Cisco APIC Layer 3 Networking Configuration Guide, Release 4.1(x)

First Published: 2019-03-28

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

Trademarks

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS REFERENCED IN THIS DOCUMENTATION ARE SUBJECT TO CHANGE WITHOUT NOTICE. EXCEPT AS MAY OTHERWISE BE AGREED BY CISCO IN WRITING, ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS DOCUMENTATION ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED.

The Cisco End User License Agreement and any supplemental license terms govern your use of any Cisco software, including this product documentation, and are located at: http://www.cisco.com/go/softwareterms.Cisco product warranty information is available at http://www.cisco.com/go/warranty. US Federal Communications Commission Notices are found here http://www.cisco.com/c/en/us/products/us-fcc-notice.html.

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 products and features described herein as in development or available at a future date remain in varying stages of development and will be offered on a when-and if-available basis. Any such product or feature roadmaps are subject to change at the sole discretion of Cisco and Cisco will have no liability for delay in the delivery or failure to deliver any products or feature roadmap items that may be set forth in this document.

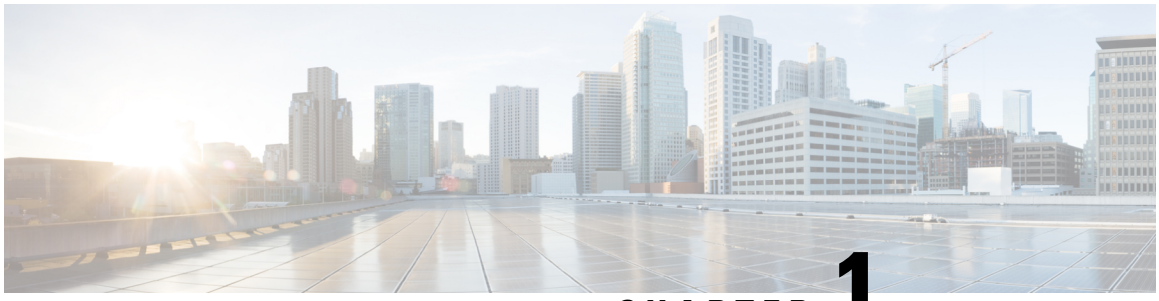
Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual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

The documentation set for this product strives to use bias-free language. For the purposes of this documentation set, bias-free is defined as language that does not imply discrimination based on age, disability, gender, racial identity, ethnic identity, sexual orientation, socioeconomic status, and intersectionality. Exceptions may be present in the documentation due to language that is hardcoded in the user interfaces of the product software, language used based on RFP documentation, or language that is used by a referenced third-party product.

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)

Trademarks

CONTENTS



CHAPTER 1

New and Changed Information

This chapter contains the following section:

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.

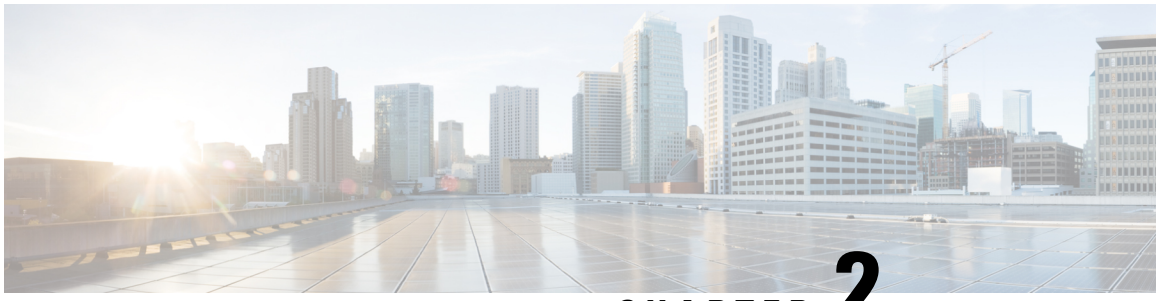
Table 1: New Features and Changed Behavior in Cisco APIC Release 4.1(2)

Feature or Change	Description	Where Documented
Direct traffic forwarding	Support is now available for direct traffic forwarding between remote leaf switches in different remote locations.	About Direct Traffic Forwarding, on page 377
--	Added a Remote Leaf restriction and limitation.	Remote Leaf Switch Restrictions and Limitations, on page 358

Table 2: New Features and Changed Behavior in Cisco APIC Release 4.1(1)

Feature or Change	Description	Where Documented
MLD snooping	Support for Multicast Listener Discovery (MLD) snooping	MLD Snooping, on page 295
Microsoft NLB	Support for Microsoft Network Load Balancing (NLB)	Microsoft NLB, on page 259
ACI IP SLAs	Support for IP SLAs in static routes	IP SLAs, on page 231

New and Changed Information



CHAPTER 2

Cisco ACI Forwarding

This chapter contains the following sections:

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.

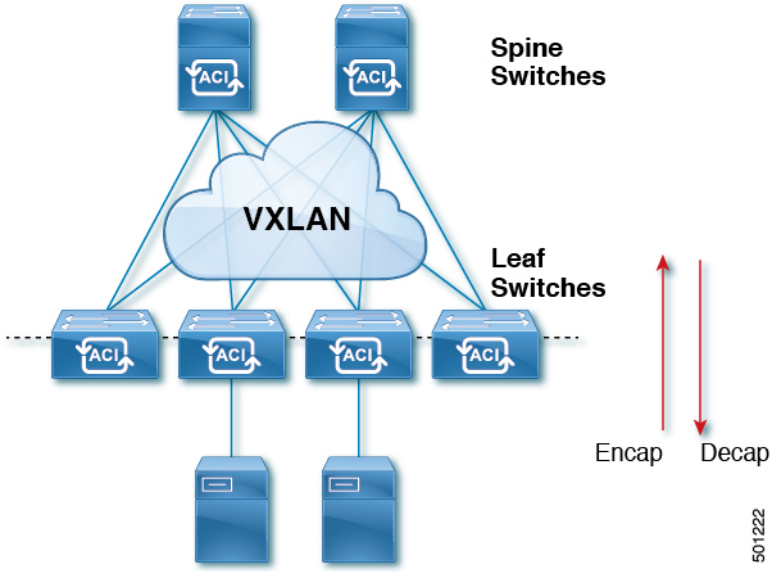
Today, application design drives east-west traffic from server to server through the data center access layer. Applications driving this shift include big data distributed processing designs like Hadoop, live virtual machine or workload migration as with VMware vMotion, server clustering, and multi-tier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

Figure 1: ACI Fabric



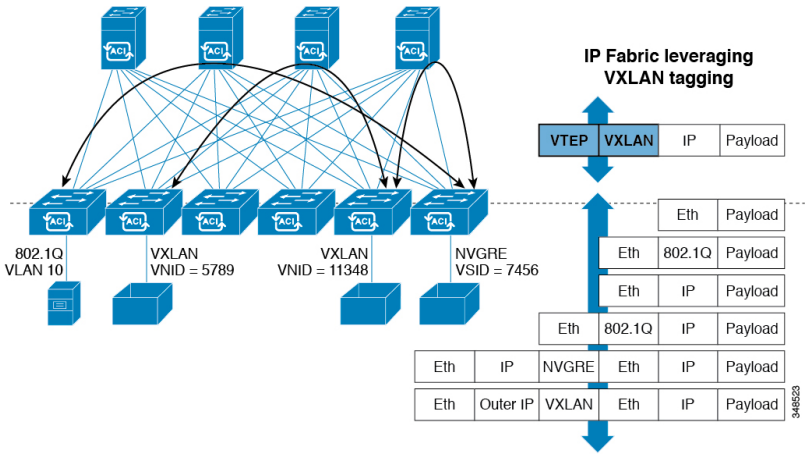
As traffic enters the fabric, ACI encapsulates and applies policy to it, forwards it as needed across the fabric through a spine switch (maximum two-hops), and de-encapsulates it upon exiting the fabric. Within the fabric, ACI uses Intermediate System-to-Intermediate System Protocol (IS-IS) and Council 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.

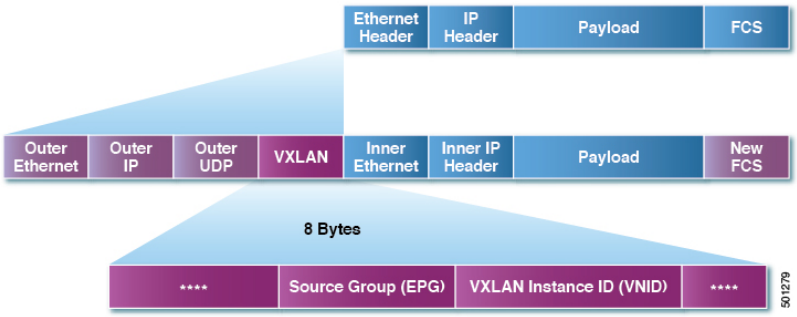
Figure 2: ACI Encapsulation Normalization



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.

Figure 3: ACI VXLAN Packet Format



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.

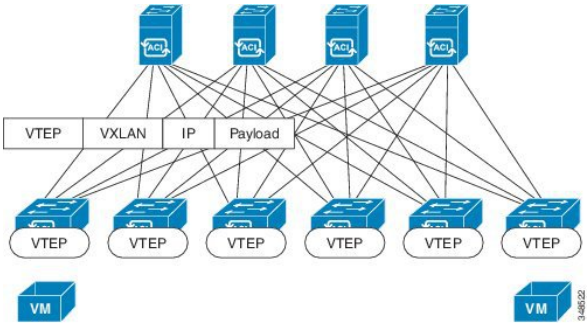
VXLAN enables ACI to deploy Layer 2 virtual networks at scale across the fabric underlay Layer 3 infrastructure. Application endpoint hosts can be flexibly placed in the data center network without concern for the Layer 3 boundary of the underlay infrastructure, while maintaining Layer 2 adjacency in 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

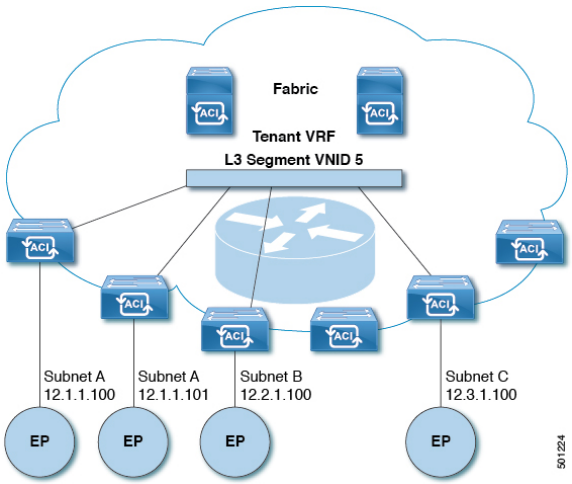
The IP interface has a unique IP address that identifies the VTEP device on the transport IP network known as the infrastructure VLAN. The VTEP device uses this IP address to encapsulate Ethernet frames and transmit the encapsulated packets to the transport network through the IP interface. A VTEP device 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.

Figure 5: Layer 3 VNIDs Transport ACI Inter-subnet Tenant Traffic



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

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. It is recommended to configure at least two spine nodes per pod as MP-BGP route reflectors for redundancy.

After route reflectors are enabled in the ACI fabric, administrators can configure connectivity to external networks through leaf nodes using a component called Layer 3 Out (L3Out). A leaf node configured with an L3Out is called a border leaf. The border leaf exchanges routes with a connected external device via a routing protocol specified in the L3Out. You can also configure static routes via L3Outs.

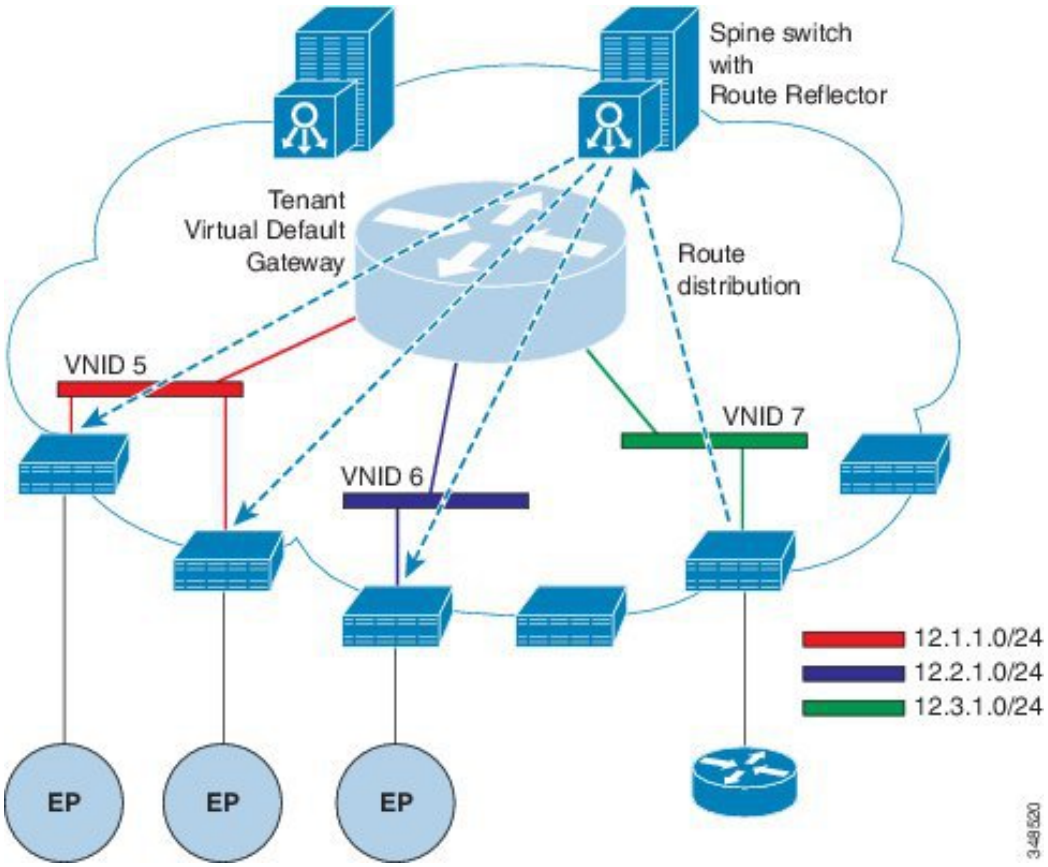
After both L3Outs and spine route reflectors are deployed, border leaf nodes learn external routes via L3Outs, and those external routes are distributed to all leaf nodes in the fabric via spine MP-BGP route reflectors.

Check the *Verified Scalability Guide for Cisco APIC* for your release to find the maximum number of routes supported by a leaf.

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.

Figure 6: Router Peering



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.

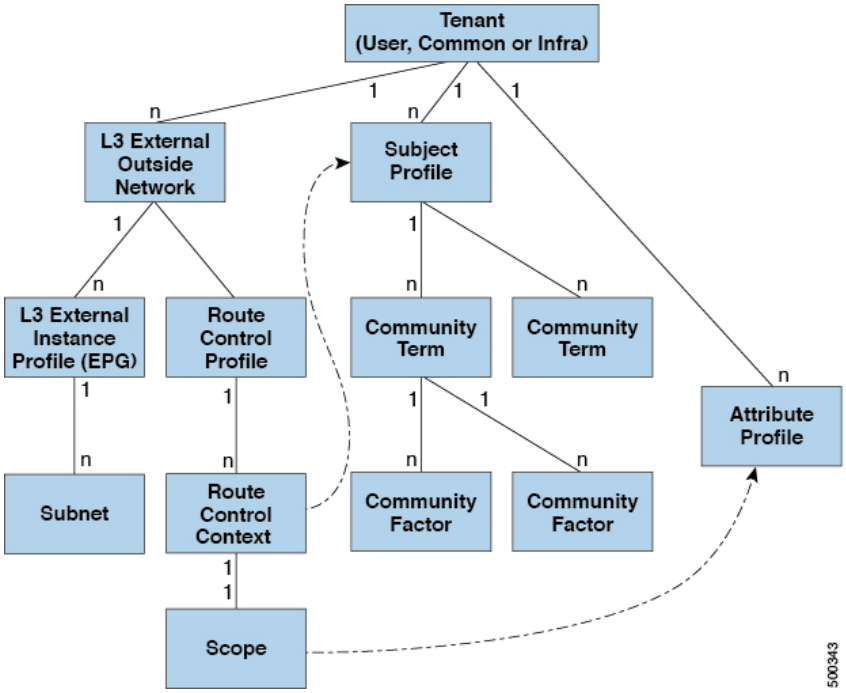
Route summarization simplifies route tables by replacing many specific addresses with an single address. For example, 10.1.1.0/24, 10.1.2.0/24, and 10.1.3.0/24 are replaced with 10.1.0.0/16. Route summarization policies enable routes to be shared efficiently among border leaf switches and their neighbor 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 `rtctrtlProfile`. A `rtctrtlProfile` can also exist under the tenant. In this case, the `l3extOut` has an interleak relation policy (`L3extRsInterleakPol`) that associates it with the `rtctrtlProfile` under the tenant. This configuration enables reusing the `rtctrtlProfile` 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 `rtctrtlProfile` defined under an L3extOut has a higher priority than one defined under the tenant.

The `rtctrtlProfile` 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 `rtctrtlProfile` provides permit-all behavior without defining explicit (0/0) subnets. A global `rtctrtlProfile` 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 `rtctrtlProfile` policies can be configured under a tenant.

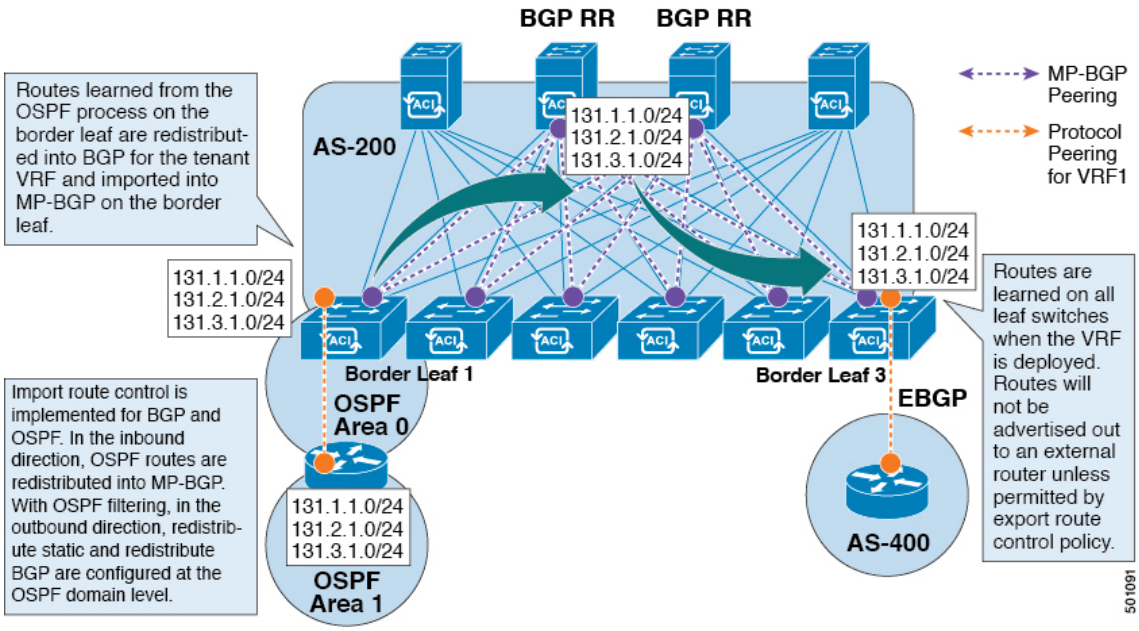
`rtctrtlProfile` policies enable enhanced default import and default export route control. Layer 3 Outside networks with aggregated import or export routes can have import/export policies that specify supported default-export and default–import, and supported 0/0 aggregation policies. To apply a `rtctrtlProfile` policy on all routes (inbound or outbound), define a global default `rtctrtlProfile` 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 `rtctrtlProfile` because a switch can have only one route map.

The protocol interleak 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 interleak policy applies to routes learned by the routing protocol in the `L3extOut`. Currently, interleak and redistribute policies are supported for OSPF v2 and v3. A route control policy `rtctrtlProfile` has to be defined as `global` when it is consumed by an interleak policy.

ACI Route Redistribution

Figure 8: 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

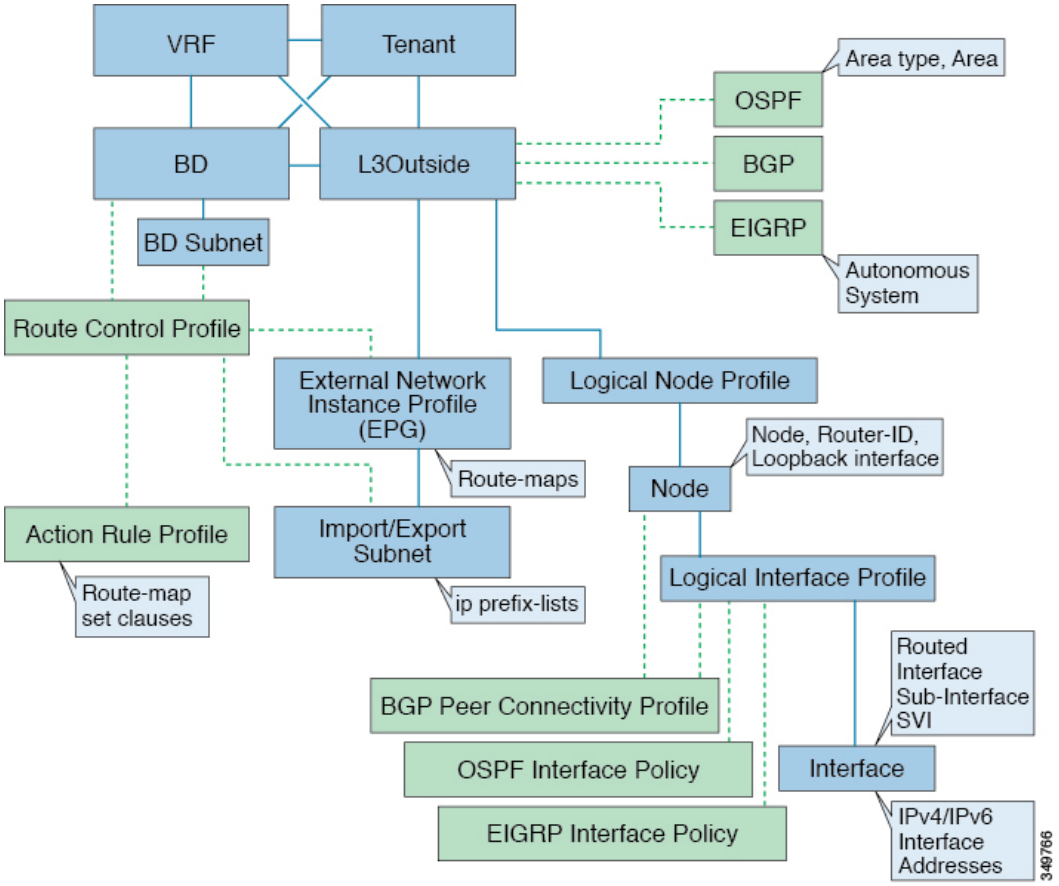
ACI supports the following External Layer 3 Outside connection options:

- 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.

Figure 9: ACI Layer 3 Managed Objects



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 **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 **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 *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-l3 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-l3 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 *Troubleshooting Unwanted _ui_ Objects* in the *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 3: Route Control Options

Route control Setting	Use	Options
Export Route Control	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.	Specific match (prefix and prefix length).
Import Route Control	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**.	Specific match (prefix and prefix length) .
Security Import Subnet	Used to permit the packets to flow between two prefix-based EPGs. Implemented with ACLs.	Uses the ACL match prefix or wildcard match rules.
Aggregate Export	Used to allow all prefixes to be advertised to the external peers. Implemented with the 0.0.0.0/ le 32 IP prefix-list.	Only supported for 0.0.0.0/0 subnet (all prefixes).
Aggregate Import	Used to allow all prefixes that are inbound from an external BGP peer. Implemented with the 0.0.0.0/0 le 32 IP prefix-list.	Only supported for the 0.0.0.0/0 subnet (all prefixes).

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.

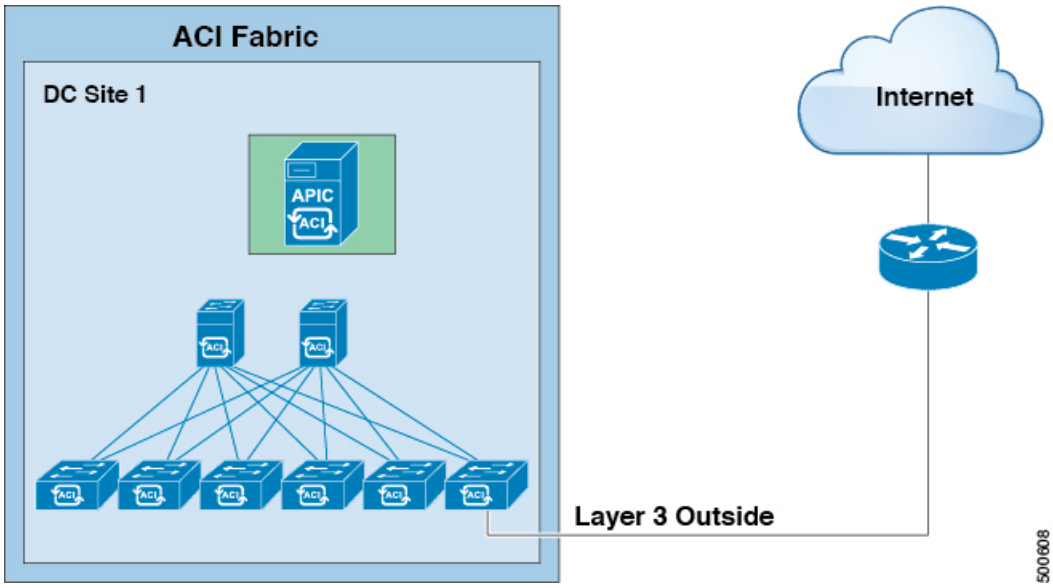
If you enable aggregate shared routes, if a route learned in one VRF must be advertised to another VRF, the routes can be shared by matching the subnet exactly, or they can be shared by using an aggregate subnet mask. Multiple subnet masks can be used to determine which specific route groups are shared 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.

Figure 10: Layer 3 outside network connection



1. Prerequisites

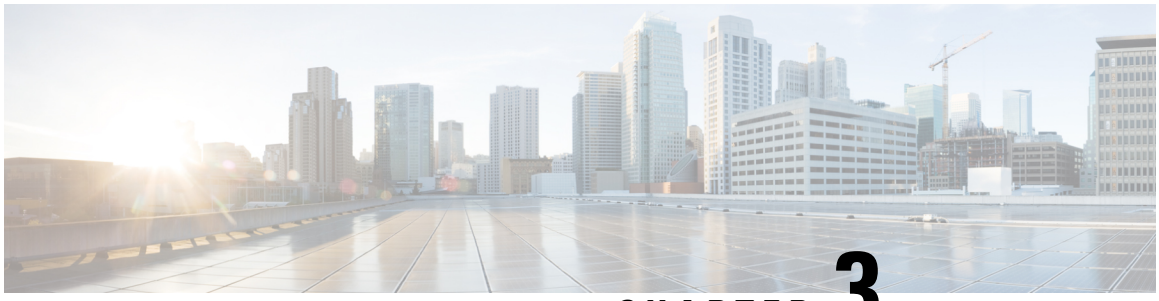
- Ensure that you have read/write access privileges to the infra security domain.
- Ensure that the target leaf switches with the necessary interfaces are available.

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.



CHAPTER 3

Prerequisites for Configuring Layer 3 Networks

This chapter contains the following sections:

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*.

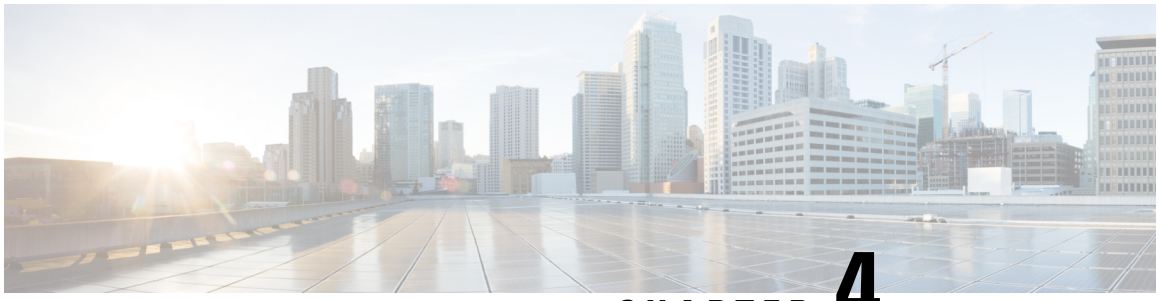
Caution If you install 1 Gigabit Ethernet (GE) or 10GE links between the leaf and spine switches in the fabric, there is risk of packets being dropped instead of forwarded, because of inadequate bandwidth. To avoid the risk, use 40GE or 100GE links between the leaf and spine switches.

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.



CHAPTER 4

Routed Connectivity to External Networks

This chapter contains the following sections:

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.

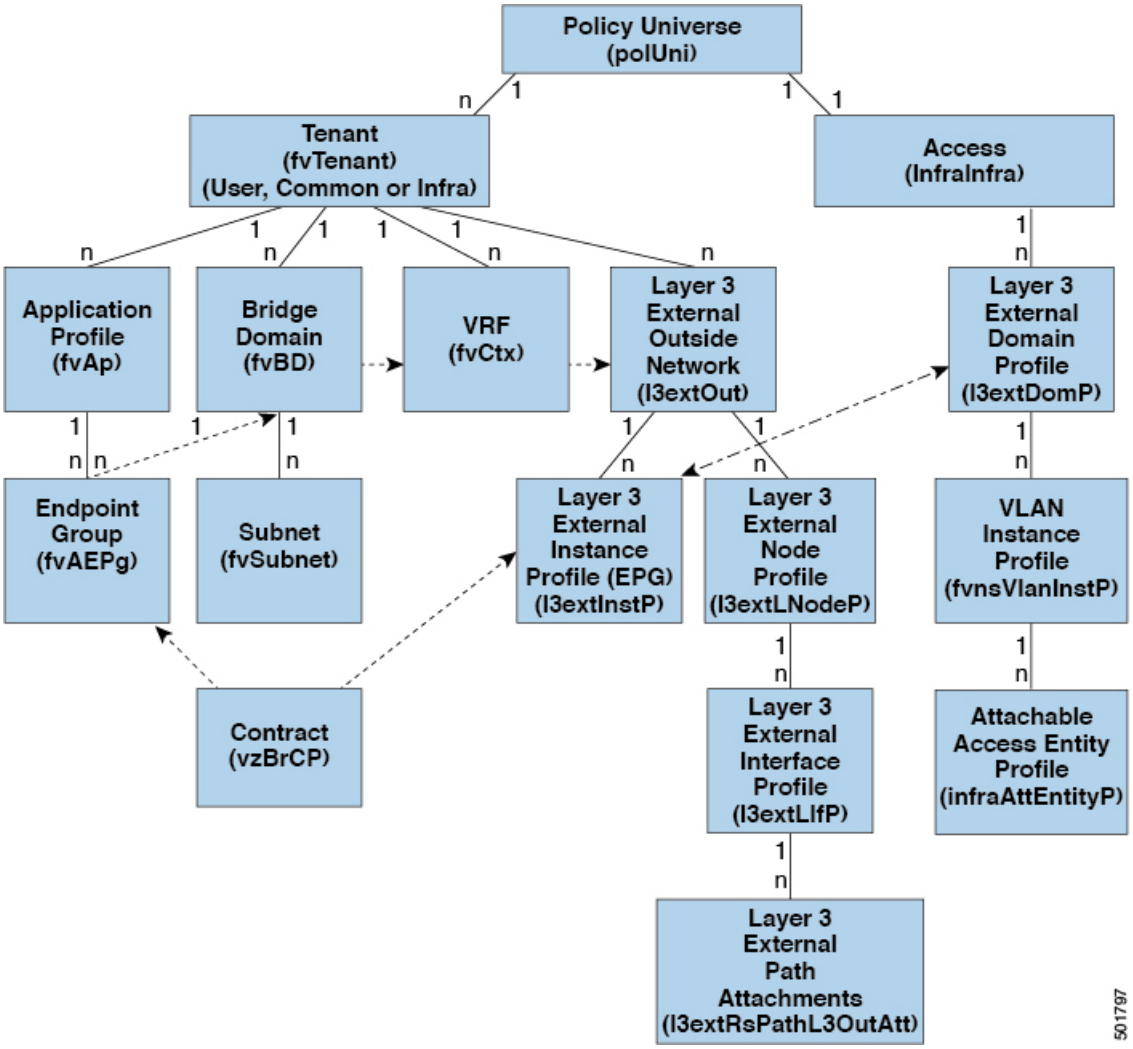
Note For guidelines and cautions for configuring and maintaining Layer 3 outside connections, see Guidelines for Routed Connectivity to Outside Networks, on page 26.

For information about the types of L3Outs, see External Layer 3 Outside Connection Types, on page 14.

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:

Figure 11: Policy Model for Layer 3 External Connections



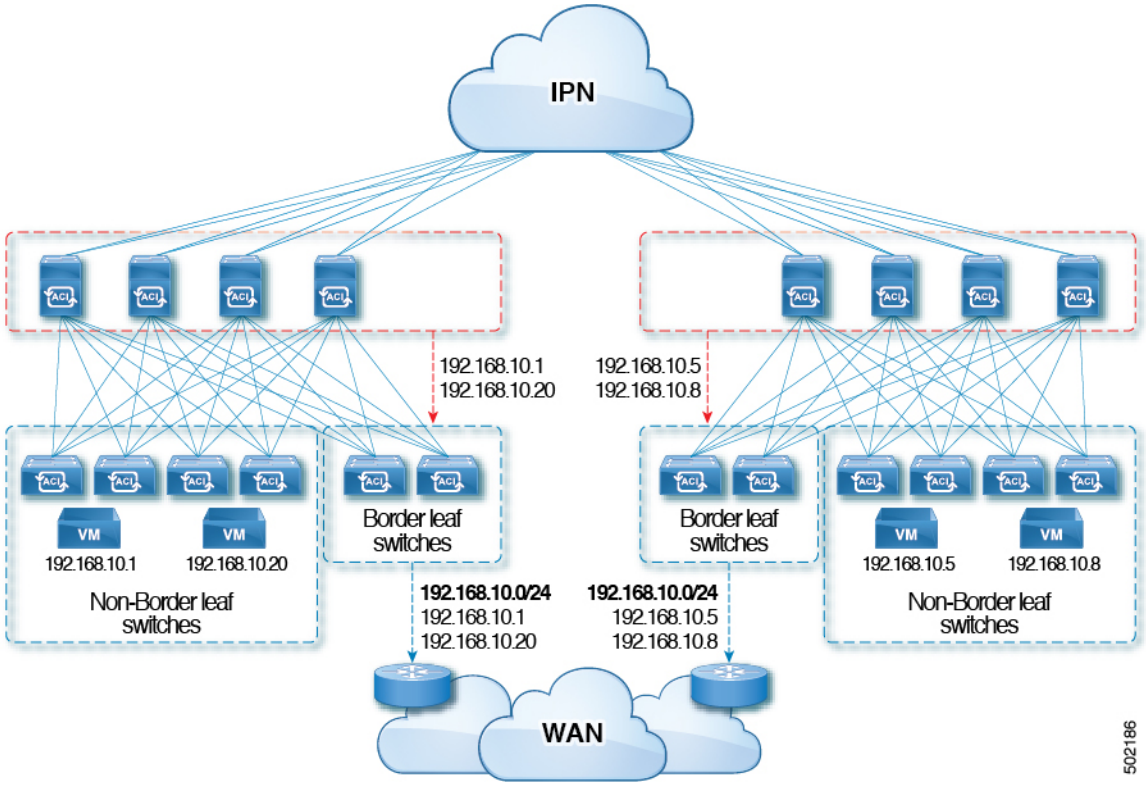
A Layer 3 external outside network (`l3extOut` object) includes the routing protocol options (BGP, OSPF, or EIGRP or supported combinations) and the switch-specific and interface-specific configurations. While the `l3extOut` contains the routing protocol (for example, OSPF with its related Virtual Routing 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:

- When host routes are advertised, the VRF Transit Route Tag is set in order to prevent them from being advertised back into the fabric and installed. In order for this loop protection to work properly, external routers must preserve this route-tag if advertising to another L3Out.
- If a bridge domain is tied to an EPG that has the same subnet configured for internal leaking, you must also enable the "Advertised Externally" flag on the EPG subnet.
- 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: epg1/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.

Topic	Caution or Guideline
Issue where a border leaf switch in a vPC pair forwards a BGP packet with an incorrect VNID to an on-peer learned endpoint	If the following conditions exist in your configuration: • Two leaf switches are part of a vPC pair • For the two leaf switches connected behind the L3Out, the destination endpoint is connected to the second (peer) border leaf switch, and the endpoint is on-peer learned on that leaf switch If the endpoint is on-peer learned on the ingress leaf switch that receives a BGP packet that is destined to the on-peer learned endpoint, an issue might arise where the transit BGP connection fails to establish between the first layer 3 switch behind the L3Out and the on-peer learned endpoint on the second leaf switch in the vPC pair. This might happen in this situation because the transit BGP packet with port 179 is forwarded incorrectly using the bridge domain VNID instead of the VRF VNID. To resolve this issue, move the endpoint to any other non-peer leaf switch in the fabric so that it is not learned on the leaf switch.
Border leaf switches and GIR (maintenance) mode	If a border leaf switch has a static route and is placed in Graceful Insertion and Removal (GIR) mode, or maintenance mode, the route from the border leaf switch might not be removed from the routing table of switches in the ACI fabric, which causes routing issues. To work around this issue, either: • Configure the same static route with the same administrative distance on the other border leaf switch, or • Use IP SLA or BFD for track reachability to the next hop of the static route
Updates through CLI	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.
Loopbacks for Layer 3 networks on same node	When configuring two Layer 3 external networks on the same node, the loopbacks need to be configured separately for both Layer 3 networks.

Topic	Caution or Guideline
Ingress-based policy enforcement	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 `Actrl` 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.
Bridge Domains with L3Outs	A bridge domain in a tenant can contain a public subnet that is advertised through an `l3extOut` provisioned in the common tenant.
Bridge domain route advertisement For OSPF and EIGRP	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.
BGP Maximum Prefix Limit	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.

Topic	Caution or Guideline
MTU	**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`.
Layer 4 to Layer 7	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.
QoS for L3Outs	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: • The VRF Policy Control Enforcement Direction must be set to**Egress**. • The VRF Policy Control Enforcement Preference must be set to **Enabled**. • 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.
ICMP settings	ICMP redirect and ICMP unreachable are disabled by default in Cisco ACI to protect the switch CPU from generating these packets.

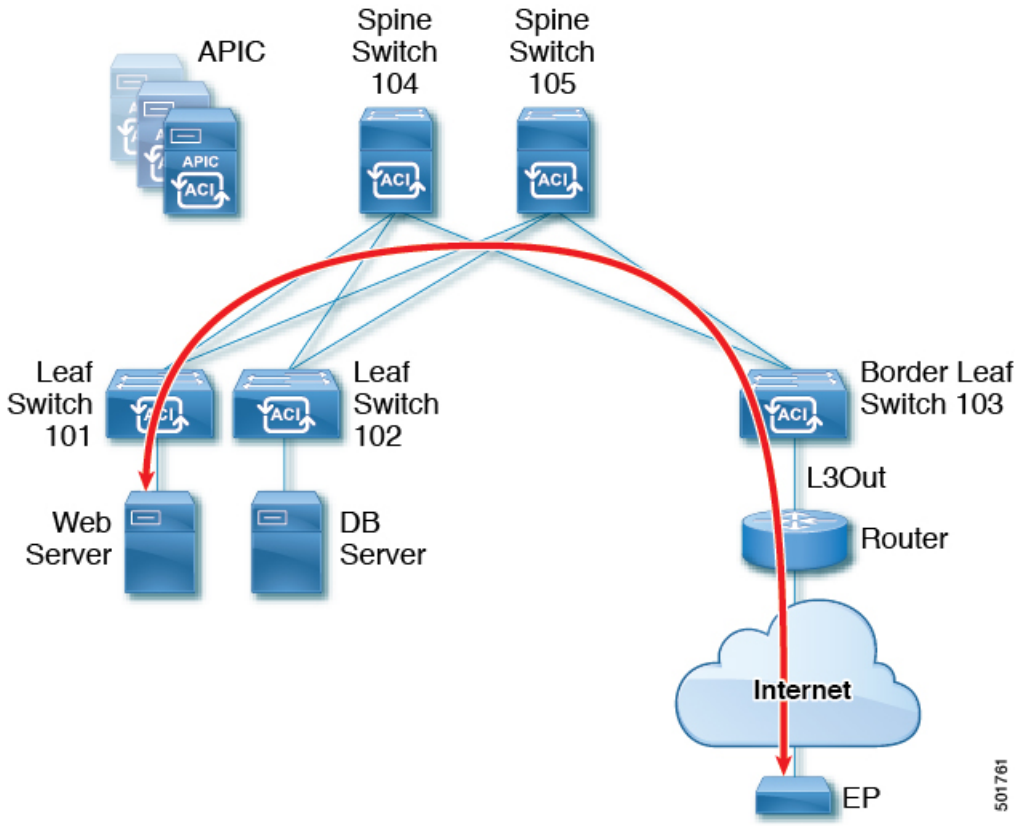
Configuring Layer 3 Outside for Tenant Networks

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:

Property	Node 103 (Border Leaf)	Node 101 (Non-Border Leaf)
Tenant	`t1`	`t1`
VRF	`v1`	`v1`
Layer 3 Outside	`l3out1`	--
Bridge domain	--	`bd1` with subnet 44.44.44.1/24
Node	Node 103, with profile `nodep1` with router ID 11.11.11.103 and path through 12.12.12.3/24	Node 101
Interface	OSPF interface `ifp1` at eth/1/3 with IP address 11.11.11.1/24	--
BGP details	Peer address 15.15.15.2/24 and ASN 100	--
OSPF details	OSPF area 0.0.0.0 and type Regular	--
EPG	External EPG `extnw1` at 20.20.20.0/24	Application `app1` with `epg1`, with `bd1`
Route Control Profile	`rp1` with a route control context `ctxp1`	--
Route map	`map1` with rule `match-rule1` with a route destination 200.3.2.0/24	--
Filter	`http-filter`	`http-filter`
Contract	`httpCtrct` provided by extnw1	`httpCtrct` consumed by epg1

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 35.

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 34.

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 `epg1`, and associates the EPG with `bd1` and the contract `httpCtrct`, as the consumer.

Example:

```
<fvAp name="app1">
    <fvAEPg name="epg1">
        <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 `eth1/3` as a routed interface (Layer 3 port), with IP address `12.12.12.1/24` and Layer 3 domain `dom1`.

Example:

```
<l3extOut name="l3out1">
    <l3extRsEctx tnFvCtxName="v1"/>
    <l3extLNodeP name="nodep1">
        <l3extRsNodeL3OutAtt rtrId="11.11.11.103" tDn="topology/pod-1/node-103"/>
        <l3extLIfP name="ifp1"/>
         <l3extRsPathL3OutAtt addr="12.12.12.3/24" ifInstT="l3-port"
tDn="topology/pod-1/paths-103/pathep-[eth1/3]"/>
        </l3extLIfP>
    </l3extLNodeP>
    <l3extRsL3DomAtt tDn="uni/l3dom-dom1"/>
</l3extOut>
```

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:

```
<l3extOut name="l3out1">
    <l3extLNodeP name="nodep1">
        <bgpPeerP addr="15.15.15.2">
            <bgpAsP asn="100"/>
        </bgpPeerP>
    </l3extLNodeP>
    <bgpExtP/>
</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:

```
<l3extOut name="l3out1">
    <ospfExtP areaId="0.0.0.0" areaType="regular"/>
    <l3extLNodeP name="nodep1">
        <l3extLIfP name="ifp1">
            <ospfIfP/>
        <l3extIfP>
    <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:

```
<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:

```
<fvTenant name="t1">
    <rtctrlSubjP name="match-rule1">
         <rtctrlMatchRtDest ip="200.3.2.0/24"/>
    </rtctrlSubjP>
    <l3extOut name="l3out1">
        <rtctrlProfile name="rp1">
            <rtctrlCtxP name="ctxp1" action="permit" order="0">
                <rtctrlScope>
                    <rtctrlRsScopeToAttrP tnRtctrlAttrPName="attrp1"/>
                </rtctrlScope>
                <rtctrlRsCtxPToSubjP tnRtctrlSubjPName="match-rule1"/>
            </rtctrlCtxP>
        </rtctrlProfile>
        <l3extInstP name="extnw1">
            <l3extSubnet ip="20.20.20.0/24" scope="import-security"/>
            <l3extRsInstPToProfile direction='export' tnRtctrlProfileName="rp1"/>
            <fvRsProv tnVzBrCPName="httpCtrct"/>
        </l3extInstP>
    </l3extOut>
</fvTenant>
```

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:

```
<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 version="1.0" encoding="UTF-8"?>
<!-- api/policymgr/mo/.xml -->
<polUni>
    <infraInfra>
        <!-- Node profile -->
        <infraNodeP name="nodeP1">
            <infraLeafS name="leafS1" type="range">
                <infraNodeBlk name="NodeBlk1" from_="101" to_="103" />
            </infraLeafS>
            <infraRsAccPortP tDn="uni/infra/accportprof-PortP1" />
        </infraNodeP>
        <!-- Port profile -->
        <infraAccPortP name="PortP1">
            <!-- 12 regular ports -->
            <infraHPortS name="PortS1" type="range">
```

```
                <infraPortBlk name="portBlk1" fromCard="1" toCard="1" fromPort="3"
toPort="32"/>
                <infraRsAccBaseGrp tDn="uni/infra/funcprof/accportgrp-default" />
            </infraHPortS>
        </infraAccPortP>
        <!-- Functional profile -->
        <infraFuncP>
            <!-- Regular port group -->
            <infraAccPortGrp name="default">
                <infraRsAttEntP tDn="uni/infra/attentp-aeP1" />
            </infraAccPortGrp>
        </infraFuncP>
        <infraAttEntityP name="aeP1">
            <infraRsDomP tDn="uni/phys-dom1"/>
            <infraRsDomP tDn="uni/l3dom-dom1/>
        </infraAttEntityP>
        <fvnsVlanInstP name="vlan-1024-2048" allocMode="static">
             <fvnsEncapBlk name="encap" from="vlan-1024" to="vlan-2048" status="created"/>
        </fvnsVlanInstP>
    </infraInfra>
    <physDomP dn="uni/phys-dom1" name="dom1">
         <infraRsVlanNs tDn="uni/infra/vlanns-[vlan-1024-2048]-static"/>
    </physDomP>
    <l3extDomP name="dom1">
        <infraRsVlanNs tDn="uni/infra/vlanns-[vlan-1024-2048]-static" />
    </l3extDomP>
</polUni>
```

The following example configures the required BGP route reflectors:

```
 <!-- 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>
    <fabricFuncP>
        <fabricPodPGrp name="bgpRRPodGrp1">
            <fabricRsPodPGrpBGPRRP tnBgpInstPolName="default"/>
        </fabricPodPGrp>
    </fabricFuncP>
    <fabricPodP name="default">
        <fabricPodS name="default" type="ALL">
            <fabricRsPodPGrp tDn="uni/fabric/funcprof/podpgrp-bgpRRPodGrp1"/>
        </fabricPodS>
    </fabricPodP>
</polUni>
```

REST API Example: L3Out

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

```
<?xml version="1.0" encoding="UTF8"?>
<!-- api/policymgr/mo/.xml -->
<polUni>
   <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>
        <fvAp name="app1">
            <fvAEPg name="epg1">
                <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>
        <l3extOut name="l3out1">
            <l3extRsEctx tnFvCtxName="v1"/>
            <l3extLNodeP name="nodep1">
                <l3extRsNodeL3OutAtt rtrId="11.11.11.103" tDn="topology/pod-1/node-103"/>
                <l3extLIfP name="ifp1">
                    <l3extRsPathL3OutAtt addr="12.12.12.3/24" ifInstT="l3-port"
tDn="topology/pod-1/paths-103/pathep-[eth1/3]"/>
                </l3extLIfP>
                <bgpPeerP addr="15.15.15.2">
                    <bgpAsP asn="100"/>
                </bgpPeerP>
             </l3extLNodeP>
             <l3extRsL3DomAtt tDn="uni/l3dom-dom1"/>
             <bgpExtP/>
             <ospfExtP areaId="0.0.0.0" areaType="regular"/>
             <l3extInstP name="extnw1" >
                 <l3extSubnet ip="20.20.20.0/24" scope="import-security"/>
                 <l3extRsInstPToProfile direction="export" tnRtctrlProfileName="rp1"/>
                 <fvRsProv tnVzBrCPName="httpCtrct"/>
             </l3extInstP>
             <rtctrlProfile name="rp1">
                 <rtctrlCtxP name="ctxp1" action="permit" order="0">
                     <rtctrlScope>
                         <rtctrlRsScopeToAttrP tnRtctrlAttrPName="attrp1"/>
                     </rtctrlScope>
                     <rtctrlRsCtxPToSubjP tnRtctrlSubjPName="match-rule1"/>
                 </rtctrlCtxP>
             </rtctrlProfile>
         </l3extOut>
         <rtctrlSubjP name="match-rule1">
             <rtctrlMatchRtDest ip="200.3.2.0/24"/>
         </rtctrlSubjP>
         <rtctrlAttrP name="attrp1">
             <rtctrlSetASPath criteria="prepend">
                 <rtctrlSetASPathASN asn="100" order="2"/>
                 <rtctrlSetASPathASN asn="200" order="1"/>
             </rtctrlSetASPath>
         </rtctrlAttrP>
         <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>
    </fvTenant>
</polUni>
```

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 40.

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 using the commands for these prerequisites, see NX-OS Style CLI Example: L3Out Prerequisites, on page 40.

Procedure

Step 1 Configure the tenant and VRF.

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

Example:

```
apic1# configure
apic1(config)# tenant t1
apic1(config-tenant)# vrf context v1
apic1(config-tenant-vrf)# exit
apic1(congig-tenant)# exit
apic1(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:

```
apic1(config)# leaf 103
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)# exit
```

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
apic1(config-leaf)# 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
```

```
apic1(config-leaf-vrf-route-map)# match route group match-rule1 order 0
apic1(config-leaf-vrf-route-map-match)# exit
apic1(config-leaf-vrf-route-map)# exit
apic1(config-leaf-vrf)# exit
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)# route-map rp1 in
apic1(config-leaf-bgp-vrf-neighbor)# exit
apic1(config-leaf-bgp-vrf)#exit
apic1(config-leaf-bgp)# exit
apic1(config-leaf)# exit
```

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
```

```
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
```

Step 11 Configure contracts and associate them with EPGs.

Example:

```
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)# application app1
apic1(config-tenant-app)# epg epg1
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)#
```

NX-OS Style CLI Example: L3Out Prerequisites

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

1. Configure a VLAN domain:

```
apic1# configure
apic1(config)# vlan-domain dom1
apic1(config-vlan)# vlan 1024-2048
apic1(config-vlan)# exit
```

2. Configure BGP route reflectors:

```
apic1(config)# bgp-fabric
apic1(config-bgp-fabric)# asn 100
apic1(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.

```
apic1# configure
apic1(config)# tenant t1
apic1(config-tenant)# vrf context v1
apic1(config-tenant-vrf)# exit
apic1(config-tenant)# exit
apic1(config)# leaf 103
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-vrf)# 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
apic(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-match)# 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)#exit
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-match)# 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
```

```
epg1
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-l3ext-epg)# vrf member v1
apic1(config-tenant-l3ext-epg)# contract provider httpCtrct
apic1(config-tenant-l3ext-epg)# exit
apic1(config-tenant)# application app1
apic1(config-tenant-app)# epg epg1
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 leafs.
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**.

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 **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**.

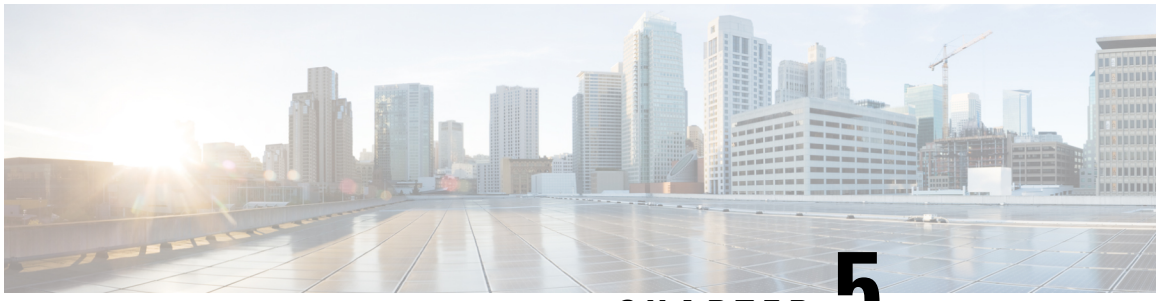
Step 22 To add a contract, use the following steps:

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**.



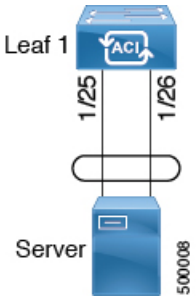
CHAPTER 5

Layer 3 Routed and Sub-Interface Port Channels

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



Note 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

Note 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*.

- 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.

Note

- 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 Tenant Routed Multicast, on page 209 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 Tenant Routed Multicast, on page 209 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 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 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**.

See "Configuring IPv6 Neighbor Discovery Duplicate Address Detection" for more information for this field.

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.

See "Configuring IPv6 Neighbor Discovery Duplicate Address Detection" for more information for the IPv6 DAD field in the Create Secondary IP Address screen.

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.

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 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

	Command or Action	Purpose
Step 1	**configure** **Example:** `apic1# configure`	Enters global configuration mode.
Step 2	**leaf** *node-id* **Example:** `apic1(config)# leaf 101`	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.
Step 3	**interface port-channel** *channel-name* **Example:** `apic1(config-leaf)# interface port-channel po1`	Enters the interface configuration mode for the specified port channel.
Step 4	**no switchport** **Example:** `apic1(config-leaf-if)# no switchport`	Makes the interface Layer 3 capable.
Step 5	**vrf member** *vrf-name* **tenant** *tenant-name* **Example:** `apic1(config-leaf-if)# vrf member v1 tenant t1`	Associates this port channel to this virtual routing and forwarding (VRF) instance and L3 outside policy, where: • *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.

	Command or Action	Purpose
Step 6	**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.
Step 7	**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.
Step 8	**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.
Step 9	**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.
Step 10	**mac-address** *mac-address* **Example:** `apic1(config-leaf-if)# mac-address 00:44:55:66:55::01`	Manually sets the interface MAC address.
Step 11	**mtu** *mtu-value* **Example:** `apic1(config-leaf-if)# mtu 1500`	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

	Command or Action	Purpose
Step 1	**configure** **Example:** `apic1# configure`	Enters global configuration mode.
Step 2	**leaf** *node-id* **Example:** `apic1(config)# leaf 101`	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.
Step 3	**vrf member** *vrf-name* **tenant** *tenant-name* **Example:** `apic1(config-leaf-if)# vrf member v1 tenant t1`	Associates this port channel to this virtual routing and forwarding (VRF) instance and L3 outside policy, where:, where: • *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.
Step 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.

	Command or Action	Purpose
Step 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.
Step 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.
Step 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.
Step 8	**mac-address** *mac-address* **Example:** `apic1(config-leaf-if)# mac-address 00:44:55:66:55::01`	Manually sets the interface MAC address.
Step 9	**mtu** *mtu-value* **Example:** `apic1(config-leaf-if)# mtu 1500`	Sets the MTU for this class of service.
Step 10	exit **Example:** `apic1(config-leaf-if)# exit`	Returns to configure mode.
Step 11	**interface port-channel** *channel-name* **Example:** `apic1(config-leaf)# interface port-channel po1`	Enters the interface configuration mode for the specified port channel.

	Command or Action	Purpose
Step 12	**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.
Step 13	exit **Example:** `apic1(config-leaf-if)# exit`	Returns to configure mode.
Step 14	**interface port-channel** *channel-name.number* **Example:** `apic1(config-leaf)# interface port-channel po1.2001`	Enters the interface configuration mode for the specified sub-interface port channel.
Step 15	**vrf member** *vrf-name* **tenant** *tenant-name* **Example:** `apic1(config-leaf-if)# vrf member v1 tenant t1`	Associates this port channel to this virtual routing and forwarding (VRF) instance and L3 outside policy, where:, where: • *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.
Step 16	exit **Example:** `apic1(config-leaf-if)# exit`	Returns to configure mode.

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
apic1(config-leaf-if)# mac-address 00:44:55:66:55::01
apic1(config-leaf-if)# mtu 1500
apic1(config-leaf-if)# exit
apic1(config-leaf)# interface port-channel po1
apic1(config-leaf-if)# vlan-domain member dom1
apic1(config-leaf-if)# exit
apic1(config-leaf)# interface port-channel po1.2001
```

```
apic1(config-leaf-if)# vrf member v1 tenant t1
apic1(config-leaf-if)# exit
```

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

	Command or Action	Purpose
Step 1	**configure** **Example:** `apic1# configure`	Enters global configuration mode.
Step 2	**leaf** *node-id* **Example:** `apic1(config)# leaf 101`	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.
Step 3	**interface Ethernet** *slot/port* **Example:** `apic1(config-leaf)# interface Ethernet 1/1-2`	Enters interface configuration mode for the interface you want to configure.
Step 4	**channel-group** *channel-name* **Example:** `apic1(config-leaf-if)# channel-group p01`	Configures the port in a channel group.

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

Note 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.

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 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="nblk" 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"/>
                   <infraRsAccBaseGrp tDn="uni/infra/funcprof/accbundle-po17_PolGrp"/>
             </infraHPortS>
        </infraAccPortP>

        <infraFuncP>
             <infraAccBndlGrp name="po17_PolGrp" lagT="link">
                  <infraRsHIfPol  tnFabricHIfPolName="default"/>
                  <infraRsCdpIfPol tnCdpIfPolName="default"/>
                  <infraRsLacpPol tnLacpLagPolName="default"/>
             </infraAccBndlGrp>
        </infraFuncP>
```

```
</infraInfra>
</polUni>
```

What to do next

Configure a Layer 3 routed port channel or sub-interface port channel using the REST API.

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".

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

Example:

```
<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"/>
        <l3extRsEctx 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>
                <l3extRsPathL3OutAtt addr="10.1.5.3/24" autostate="disabled" descr="" \
                 encap="unknown" encapScope="local" ifInstT="l3-port" llAddr="::" \
```

```
                mac="00:22:BD:F8:19:FF" mode="regular" mtu="inherit" \
                tDn="topology/pod-1/paths-101/pathep-[po17_PolGrp]" \
                targetDscp="unspecified"/>
               <l3extRsNdIfPol tnNdIfPolName=""/>
               <l3extRsIngressQosDppPol tnQosDppPolName=""/>
               <l3extRsEgressQosDppPol tnQosDppPolName=""/>
            </l3extLIfP>
        </l3extLNodeP>
        <l3extInstP descr="" floodOnEncap="disabled" matchT="AtleastOne" \
         name="accountingInst" nameAlias="" prefGrMemb="exclude" prio="unspecified" \
         targetDscp="unspecified">
            <fvRsProv matchT="AtleastOne" prio="unspecified" tnVzBrCPName="webCtrct"/>
            <l3extSubnet aggregate="export-rtctrl,import-rtctrl" descr="" ip="0.0.0.0/0" \
             name="" nameAlias="" scope="export-rtctrl,import-rtctrl,import-security"/>
            <l3extSubnet aggregate="export-rtctrl,import-rtctrl" descr="" ip="::/0" \
             name="" nameAlias="" scope="export-rtctrl,import-rtctrl,import-security"/>
            <fvRsCustQosPol tnQosCustomPolName=""/>
        </l3extInstP>
        <l3extConsLbl descr="" name="golf" nameAlias="" owner="infra" ownerKey="" \
         ownerTag="" tag="yellow-green"/>
    </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.

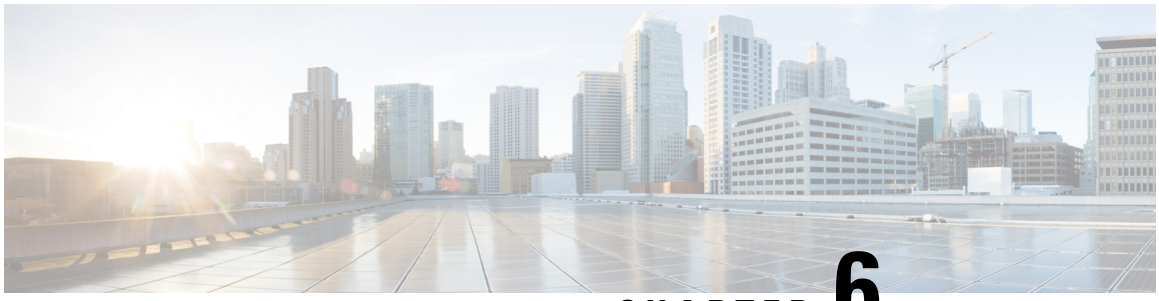
Procedure

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:

```
<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"/>
        <l3extRsEctx 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="lifp27" nameAlias="" ownerKey="" ownerTag="" \
             tag="yellow-green">
                <ospfIfP authKeyId="1" authType="none" descr="" name="" nameAlias="">
                    <ospfRsIfPol tnOspfIfPolName=""/>
                </ospfIfP>
                <l3extRsPathL3OutAtt addr="11.1.5.3/24" autostate="disabled" descr="" \
                 encap="vlan-2001" encapScope="local" ifInstT="sub-interface" \
                 llAddr="::" mac="00:22:BD:F8:19:FF" mode="regular" mtu="inherit" \
                 tDn="topology/pod-1/paths-101/pathep-[po27_PolGrp]" \
                 targetDscp="unspecified"/>
                <l3extRsNdIfPol tnNdIfPolName=""/>
                <l3extRsIngressQosDppPol tnQosDppPolName=""/>
                <l3extRsEgressQosDppPol tnQosDppPolName=""/>
            </l3extLIfP>
        </l3extLNodeP>
        <l3extInstP descr="" floodOnEncap="disabled" matchT="AtleastOne" \
         name="accountingInst" nameAlias="" prefGrMemb="exclude" prio="unspecified" \
         targetDscp="unspecified">
            <fvRsProv matchT="AtleastOne" prio="unspecified" tnVzBrCPName="webCtrct"/>
            <l3extSubnet aggregate="export-rtctrl,import-rtctrl" descr="" ip="0.0.0.0/0" \
             name="" nameAlias="" scope="export-rtctrl,import-rtctrl,import-security"/>
            <l3extSubnet aggregate="export-rtctrl,import-rtctrl" descr="" ip="::/0" \
             name="" nameAlias="" scope="export-rtctrl,import-rtctrl,import-security"/>
            <fvRsCustQosPol tnQosCustomPolName=""/>
        </l3extInstP>
        <l3extConsLbl descr="" name="golf" nameAlias="" owner="infra" ownerKey="" \
         ownerTag="" tag="yellow-green"/>
    </l3extOut>
</fvTenant>
</polUni>
```



CHAPTER 6

QoS for L3Outs

This chapter contains the following sections:

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:

```
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:

```
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:

```
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:

```
<l3extLIfP descr=""
dn="uni/tn-DT/out-L3_4_2_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:

```
<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" 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=""/>
    <l3extRsLIfPCustQosPol annotation="" tnQosCustomPolName=" vrfQos002"/>
</l3extLIfP>
```

Step 3 Configure QoS priorities for a routed outside.

Example:

```
<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 66 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:

```
<fvTenant  name="t1">
    <fvCtx name="v1" pcEnfPref="enforced" pcEnfDir="egress"/>
    <fvBD name="bd1">
        <fvRsCtx tnFvCtxName="v1"/>
        <fvSubnet 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.

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, So either we need to configure this one or the new one.

Example:

```
<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>
```

Configuring QoS Contract for L3Out Using 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 65 instead.

Procedure

Step 1 Configure the VRF for egress mode and enable policy enforcement to support QoS priority enforcement on the L3Out.

```
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(congig-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.

```
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)# contract httpCtrct
apic1(config-tenant-contract)# scope vrf
apic1(config-tenant-contract)# qos-class level1
apic1(config-tenant-contract)# subject http-subject
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
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 64 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 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**.

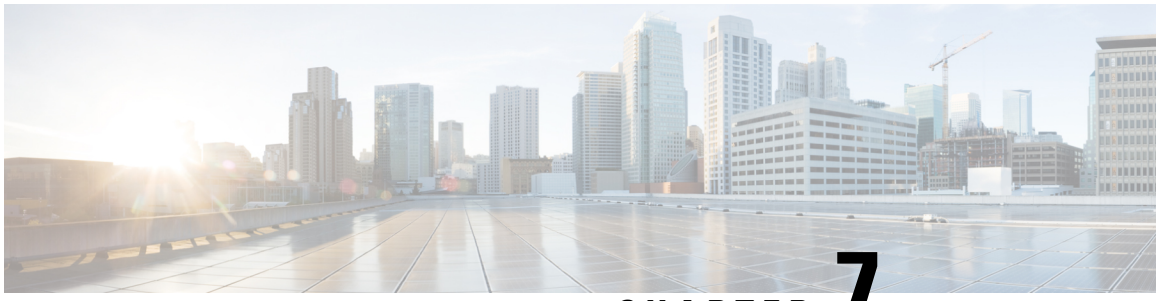
e) Expand the previously created filter and click on a filter entry.
f) Set the **Match DSCP** field to the desired DSCP level for the entry, for example, **EF**.

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**.



CHAPTER 7

Routing Protocol Support

This chapter contains the following sections:

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:

- The BGP direct route export behavior changed after release 3.2(1), where ACI does not evaluate the originating route type (such as static, direct, and so on) when matching export route map clauses. As a result, the "match direct" deny clause that is always included in the outbound neighbor route map no longer matches direct routes, and direct routes are now advertised based on whether or not a user-defined route map clause matches.

  Therefore, the direct route must be advertised explicitly through the route map. Failure to do so will implicitly deny the direct route being advertised.

- The **AS override** option in the **BGP Controls** field in the BGP Peer Connectivity Profile for an L3Out was introduced in release 3.1(2). It allows Cisco Application Centric Infrastructure (ACI) to overwrite a remote AS in the AS_PATH with ACI BGP AS. In Cisco ACI, it is typically used when performing transit routing from an eBGP L3Out to another eBGP L3Out with the same AS number.

However, an issue arises if you enable the **AS override** option when the eBGP neighbor has a different AS number. In this situation, strip the peer-as from the AS_PATH when reflecting it to a peer.

- The **Local-AS Number** option in the BGP Peer Connectivity Profile is supported only for eBGP peering. This enables Cisco ACI border leaf switches to appear to be a member of another AS in addition to its real AS assigned to the fabric MP-BGP Route Reflector Policy. This means that the local AS number must be different from the real AS number of the Cisco ACI fabric. When this feature is configured, Cisco ACI border leaf switches prepend the local AS number to the AS_PATH of the incoming updates and append the same to the AS_PATH of the outgoing updates. Prepending of the local AS number to the incoming updates can be disabled by the **no-prepend** setting in the **Local-AS Number Config**. The **no-prepend + replace-as** setting can be used to prevents the local AS number from being appended to the outgoing updates in addition to not prepending the same to the incoming updates.
- A router ID for an L3Out for any routing protocols cannot be the same IP address or the same subnet as the L3Out interfaces such as routed interface, sub-interface or SVI. However, if needed, a router ID can be the same as one of the L3Out loopback IP addresses.
- If you have multiple L3Outs of the same routing protocol on the same leaf switch in the same VRF instance, the router ID for those must be the same. If you need a loopback with the same IP address as the router ID, you can configure the loopback in only one of those L3Outs.
- There are two ways to define the BGP peer for an L3Out:
  - Through the BGP peer connectivity profile (**bgpPeerP**) at the logical node profile level (**l3extLNodeP**), which associates the BGP peer to the loopback IP address. When the BGP peer is configured at this level, a loopback address is expected for BGP connectivity, so a fault is raised if the loopback address configuration is missing.
  - Through the BGP peer connectivity profile (**bgpPeerP**) at the logical interface profile level (**l3extRsPathL3OutAtt**), which associates the BGP peer to the respective interface or sub-interface.
- You must configure an IPv6 address to enable peering over loopback using IPv6.
- 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. After the maximum 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. You can use only one option 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 Cisco Application Policy Infrastructure Controller (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:

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

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

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:

```
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
apic1(config-leaf-bgp-vrf-neighbor)# private-as-control remove-exclusive-all
apic1(config-leaf-bgp-vrf-neighbor)# private-as-control remove-exclusive-all-replace-as
apic1(config-leaf-bgp-vrf-neighbor)# exit
apic1(config-leaf-bgp-vrf)# address-family ipv4 unicast
apic1(config-leaf-bgp-vrf-af)# inherit bgp dampening damp_rp
This template will be inherited on all leaves where VRF ctx3 has been deployed
apic1(config-leaf-bgp-vrf-af)# exit
apic1(config-leaf-bgp-vrf)# address-family ipv6 unicast
apic1(config-leaf-bgp-vrf-af)# inherit bgp dampening damp_rp
This template will be inherited on all leaves where VRF ctx3 has been deployed
apic1(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:

```
<l3extOut descr="" dn="uni/tn-t1/out-l3out-bgp" enforceRtctrl="export" name="l3out-bgp"
ownerKey="" ownerTag="" targetDscp="unspecified">
 <l3extRsEctx tnFvCtxName="ctx3"/>
 <l3extLNodeP configIssues="" descr="" name="l3extLNodeP_1" ownerKey="" ownerTag=""
tag="yellow-green" targetDscp="unspecified">
  <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/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="bfd" privateASctrl="remove-all,remove-exclusive,replace-as"
ttl="1" weight="1000">
     <bgpRsPeerPfxPol tnBgpPeerPfxPolName=""/>
     <bgpAsP asn="3001" descr="" name=""/>
    </bgpPeerP>
   </l3extRsPathL3OutAtt>
  </l3extLIfP>
  <l3extLIfP descr="" name="l3extLIfP_1" ownerKey="" ownerTag="" tag="yellow-green">
   <l3extRsNdIfPol tnNdIfPolName=""/>
   <l3extRsIngressQosDppPol tnQosDppPolName=""/>
   <l3extRsEgressQosDppPol tnQosDppPolName=""/>
   <l3extRsPathL3OutAtt addr="31.0.1.2/24" 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/paths-101/pathep-[eth1/8]" targetDscp="unspecified">
    <bgpPeerP addr="31.0.1.0/24" allowedSelfAsCnt="3" ctrl="send-com,send-ext-com" descr=""
 name="" peerCtrl="" privateASctrl="remove-all,remove-exclusive,replace-as" ttl="1"
weight="100">
     <bgpRsPeerPfxPol tnBgpPeerPfxPolName=""/>
     <bgpLocalAsnP asnPropagate="none" descr="" localAsn="200" name=""/>
     <bgpAsP asn="3001" descr="" name=""/>
    </bgpPeerP>
   </l3extRsPathL3OutAtt>
  </l3extLIfP>
 </l3extLNodeP>
 <l3extRsL3DomAtt tDn="uni/l3dom-l3-dom"/>
 <l3extRsDampeningPol af="ipv6-ucast" tnRtctrlProfileName="damp_rp"/>
 <l3extRsDampeningPol af="ipv4-ucast" tnRtctrlProfileName="damp_rp"/>
 <l3extInstP descr="" matchT="AtleastOne" name="l3extInstP_1" prio="unspecified"
targetDscp="unspecified">
  <l3extSubnet aggregate="" descr="" ip="130.130.130.0/24" name=""
scope="import-rtctrl"></l3extSubnet>
  <l3extSubnet aggregate="" descr="" ip="130.130.131.0/24" name="" scope="import-rtctrl"/>
  <l3extSubnet aggregate="" descr="" ip="120.120.120.120/32" name=""
scope="export-rtctrl,import-security"/>
  <l3extSubnet aggregate="" descr="" ip="3001::130:130:130:100/120" name=""
scope="import-rtctrl"/>
 </l3extInstP>
 <bgpExtP descr=""/>
</l3extOut>
<rtctrlProfile descr="" dn="uni/tn-t1/prof-damp_rp" name="damp_rp" ownerKey="" ownerTag=""
 type="combinable">
 <rtctrlCtxP descr="" name="ipv4_rpc" order="0">
  <rtctrlScope descr="" name="">
   <rtctrlRsScopeToAttrP tnRtctrlAttrPName="act_rule"/>
```

```
  </rtctrlScope>
 </rtctrlCtxP>
</rtctrlProfile>
<rtctrlAttrP descr="" dn="uni/tn-t1/attr-act_rule" name="act_rule">
 <rtctrlSetDamp descr="" halfLife="15" maxSuppressTime="60" name="" reuse="750"
suppress="2000" type="dampening-pol"/>
</rtctrlAttrP>
```

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** > ***tenant_name*** > **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.

Refer to the *Verified Scalability Guide for Cisco APIC* on the Cisco APIC documentation page for the acceptable values for the following fields.

a) In the **Name** field, enter a name for the policy.
b) In the **eBGP Distance** field, enter a value for the administrative distance of eBGP routes.
c) In the **iBGP Distance** field, enter a value for the administrative distance of iBGP routes.
d) In the **Local Distance** field, enter a value for the local distance.
e) In the **eBGP Max ECMP** field, enter a value for the maximum number of equal-cost paths for eBGP load sharing.
f) In the **iBGP Max ECMP** field, enter a value for the maximum number of equal-cost paths for iBGP load sharing.
g) In the **Enable Host Route Leak** field, click the box to enable distributing EVPN type-2 (MAC/IP) host routes to the DCIG.
h) Click **Submit** after you have updated your entries.

Step 3 Click **Tenants** > ***tenant_name*** > **Networking** > **VRFs** > ***VRF_name***

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 ibpg 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:

```
<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:

Prepend	Appends the specified AS number to the AS path of the route matched by the route map. **Note** • 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.
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:

Prepend	1	Prepend the specified AS number.
Prepend-last-as	2	Prepend the last AS numbers to the AS path.
DEFAULT	Prepend(1)	Prepend the specified AS number.

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

```
apic1(config-leaf-vrf-template-route-profile)# set as-path {'prepend as-num [ ,... as-num ]
| prepend-last-as num}
```

Example:

```
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
apic1(config-leaf)# 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}
```

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 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 BGB 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.

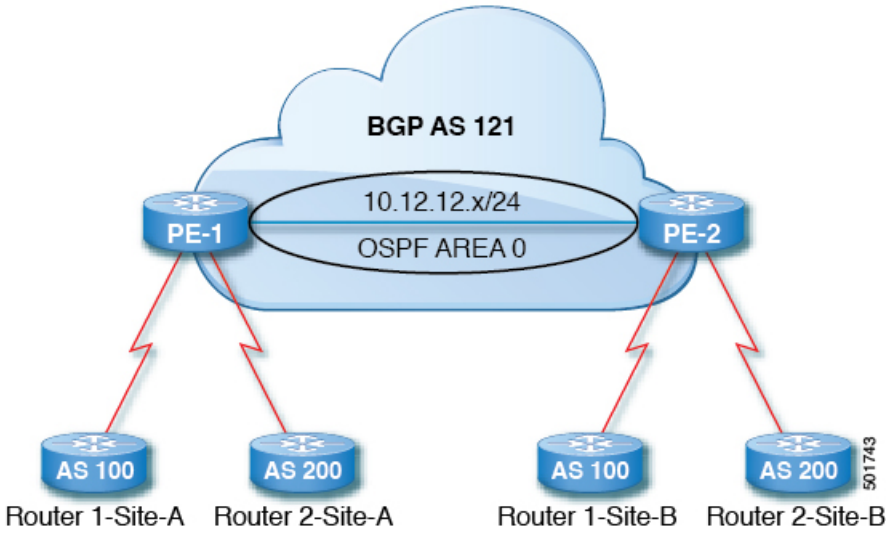
The default setting to prevent route loops from occurring could create an issue if you use the same Autonomous System number along various sites and disallow user sites with identical Autonomous System numbers to link by another Autonomous System number. In such a scenario, routing updates from one site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

Figure 14: Example Topology Illustrating the Autonomous System Override Process



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:

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:

```
<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>

   <!-- GOLF L3Out -->
   <l3extOut name="routAccounting">
      <l3extConsLbl name="golf_transit" owner="infra" status=""/>
      <bgpExtP/>
      <l3extInstP 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"

          />
          <fvRsProv tnVzBrCPName="webCtrct"/>
      </l3extInstP>

      <l3extRsEctx tnFvCtxName="coke"/>
   </l3extOut>

    <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>
      <fvAEPg name="cokeEPg2" >
          <fvRsBd tnFvBDName="cokeBD2" />
            <fvRsPathAtt tDn="topology/pod-1/paths-103/pathep-[eth1/20]" encap="vlan-110"
instrImedcy="immediate" mode="regular"/>
          <fvRsCons tnVzBrCPName="webCtrct"/>
      </fvAEPg>
    </fvAp>

    <!-- Non GOLF L3Out-->
    <l3extOut name="NonGolfOut">
       <bgpExtP/>
       <l3extLNodeP 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"/>
      </ipRouteP>


        </l3extRsNodeL3OutAtt>
          <l3extLIfP name='portIfV4'>
              <l3extRsPathL3OutAtt tDn="topology/pod-1/paths-101/pathep-[eth1/17]"
encap='vlan-1010' ifInstT='sub-interface' addr="20.1.12.2/24">

              </l3extRsPathL3OutAtt>
          </l3extLIfP>
          <l3extLIfP name='portIfV6'>
              <l3extRsPathL3OutAtt tDn="topology/pod-1/paths-101/pathep-[eth1/17]"
encap='vlan-1010' ifInstT='sub-interface' addr="64:ff9b::1401:302/120">
```

```
                    <bgpPeerP addr="64:ff9b::1401:d03" ctrl="send-com,send-ext-com" />
                </l3extRsPathL3OutAtt>
            </l3extLIfP>
            <bgpPeerP addr="2.2.2.2" ctrl="as-override,disable-peer-as-check,
send-com,send-ext-com" status=""/>
        </l3extLNodeP>
        <!--
         <bgpPeerP addr="2.2.2.2" ctrl="send-com,send-ext-com" status=""/>
         -->
        <l3extInstP name="accountingInst">
            <l3extSubnet ip="192.10.0.0/16" scope="import-security,import-rtctrl" />
            <l3extSubnet ip="192.3.3.0/24" scope="import-security,import-rtctrl" />
            <l3extSubnet ip="192.4.2.0/24" scope="import-security,import-rtctrl" />
            <l3extSubnet ip="64:ff9b::c007:200/120" scope="import-security,import-rtctrl"
/>
            <l3extSubnet ip="192.2.2.0/24" scope="export-rtctrl" />
            <l3extSubnet ip="0.0.0.0/0"
                                scope="export-rtctrl,import-rtctrl,import-security"
                                aggregate="export-rtctrl,import-rtctrl"

           />
        </l3extInstP>
       <l3extRsEctx tnFvCtxName="coke"/>
    </l3extOut>

</fvTenant>
```

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 On the menu bar, choose **Tenant** > ***Tenant_name*** > **Policies** > **Protocol** > **BGP** > **BGP Timers**, then right click **Create BGP Timers Policy**.

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 **Tenant** > ***Tenant_name*** > **Networking** > **External Routed Networks**, 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 **Tenant** > ***Tenant_name*** > **Networking** > **External Routed Networks** > ***L3Out_name*** > **Logical Node Profiles** > ***LogicalNodeProfile_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

<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" />
      <l3extLNodeP name="node1" >
        <bgpProtP name="protp1" >
            <bgpRsBgpNodeCtxPol tnBgpCtxPolName="pol2" />
        </bgpProtP>
      </l3extLNodeP>
      <l3extLNodeP name="node2" >
      </l3extLNodeP>
```

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 BPG 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

<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" />
<l3extLNodeP name="node1" >
  <bgpProtP name="protp1" status="deleted" >
      <bgpRsBgpNodeCtxPol tnBgpCtxPolName="pol2" />
  </bgpProtP>
</l3extLNodeP>
<l3extLNodeP name="node2" >
</l3extLNodeP>
```

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

	Command or Action	Purpose

Step 1	Configure BGP ASN and the route reflector before creating a timer policy.
	Example:
	`apic1(config)#`
	`apic1(config)# bgp-fabric`
	`apic1(config-bgp-fabric)# route-reflector spine 102`
	`apic1(config-bgp-fabric)# asn 42`
	`apic1(config-bgp-fabric)# exit`
	`apic1(config)# exit`
	`apic1#`	

Step 2	Create a timer policy.
	Example:
	`apic1# config`
	`apic1(config)# leaf 101`
	`apic1(config-leaf)# template bgp timers pol7 tenant tn1`
	`This template will be available on all nodes where tenant tn1 has a VRF deployment`
	`apic1(config-bgp-timers)# timers bgp 120 240`
	`apic1(config-bgp-timers)# graceful-restart stalepath-time 500`
	`apic1(config-bgp-timers)# maxas-limit 300`
	`apic1(config-bgp-timers)# exit`
	`apic1(config-leaf)# exit`
	`apic1(config)# exit`
	`apic1#`
		The specific values are provided as examples only.

Step 3	Display the configured BGP policy. **Example:**	

	Command or Action	Purpose
	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	
Step 4	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 5	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
  ctx1
  out1
    ctx1
    node1
      protp pol1

  out2
    ctx1
    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:

- Prior to APIC release 5.0, BFD echo packets received on a leaf switch from neighbor routers are classified with the default QoS class (best effort). Due to that classification, BFD drops might result when there is congestion on the interfaces.
- 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.
- Multihop 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.

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`.

Optimizing BFD on Subinterfaces

You can optimize BFD on subinterfaces. BFD creates sessions for all configured subinterfaces. BFD sets the subinterface with the lowest configured VLAN ID as the master subinterface and that subinterface uses the BFD session parameters of the parent interface. The remaining subinterfaces use the slow timer.

If the optimized subinterface session detects an error, BFD marks all subinterfaces on that physical interface as down.

You can configure the BFD echo function on one or both ends of a BFD-monitored link. The echo function slows down the required minimum receive interval, based on the configured slow timer. The *RequiredMinEchoRx* BFD session parameter is set to *zero* if the echo function is disabled. The slow timer becomes the required minimum receive interval if the echo function is enabled.

Note If one of the subinterfaces flap, subinterfaces on that physical interface are impacted and will go down for a second.

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:

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

Example:

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

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

Example:

```
apic1(config)# leaf-profile test_leaf_profile
apic1(config-leaf-profile)# leaf-group test_leaf_group
apic1(config-leaf-group)# leaf-policy-group test_leaf_policy_group
apic1(config-leaf-group)# leaf 101-102
apic1(config-leaf-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# configure
apic1(config)# spine-profile test_spine_profile
apic1(config-spine-profile)# spine-group test_spine_group
apic1(config-spine-group)# spine-policy-group test_spine_policy_group
apic1(config-spine-group)# spine 103-104
apic1(config-leaf-group)# exit
```

Configuring BFD Globally Using the REST API

Procedure

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

Example:

```
<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 Layer 3 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.

Note When a BFD interface policy is configured over a parent routed interface, by default all of its routed sub-interfaces with the same address family as that of the parent interface will inherit this policy. If any of the inherited configuration needs to be overridden, configure an explicit BFD interface policy on the sub-interfaces. However, if **Admin State** or **Echo Admin State** is disabled on the parent interface, the property cannot be overridden on the sub-interfaces.

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:

```
apic1# configure
apic1(config)# tenant t0
apic1(config-tenant)# vrf context v0
apic1(config-tenant-vrf)# exit
apic1(config-tenant)# exit
apic1(config)# leaf 101
apic1(config-leaf)# vrf context tenant t0 vrf v0
apic1(config-leaf-vrf)# exit
apic1(config-leaf)# interface Ethernet 1/18
apic1(config-leaf-if)# vrf  member tenant t0 vrf v0
apic1(config-leaf-if)# exit
apic1(config-leaf)# template bfd bfdIfPol1 tenant t0
apic1(config-template-bfd-pol)# [no] echo-mode enable
apic1(config-template-bfd-pol)# [no] echo-rx-interval 500
apic1(config-template-bfd-pol)# [no] min-rx 70
apic1(config-template-bfd-pol)# [no] min-tx 100
apic1(config-template-bfd-pol)# [no] multiplier 5
apic1(config-template-bfd-pol)# [no] optimize subinterface
apic1(config-template-bfd-pol)# 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
apic1(config-leaf-if)# bfd ipv6 inherit interface-policy bfdPol1
apic1(config-leaf-if)# bfd ipv6 authentication keyed-sha1 key 10 key password
```

Configuring BFD Interface Override Using the REST API

Procedure

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

Example:

```
<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]"
```

```
ifInstT='l3-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:

```
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# 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"/>
    <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='l3-port' addr="10.0.0.1/24" mtu="1500"/>
                <bfdIfP type="sha1" key="password">
                    <bfdRsIfPol tnBfdIfPolName='bfdIfPol'/>
                </bfdIfP>
            </l3extLIfP>


        </l3extLNodeP>
    </l3extOut>
</fvTenant>
```

Step 2 The following example shows the interface configuration for enabling BFD on OSPF and EIGRP:

Example:

BFD on leaf switch

```
<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

```
<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>

             <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"/>
                 <ospfIfP>
                     <ospfRsIfPol tnOspfIfPolName='ospf_intf_pol'/>
                 </ospfIfP>
                 <bfdIfP name="test" type="sha1" key="hello" status="created,modified">
                    <bfdRsIfPol tnBfdIfPolName='default' status="created,modified"/>
                </bfdIfP>
             </l3extLIfP>

         </l3extLNodeP>
```

Step 3 The following example shows the interface configuration for enabling BFD on BGP:

Example:

```
<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='l3-port' addr="10.0.0.1/24" mtu="1500">
                  <bgpPeerP addr="4.4.4.4/24" allowedSelfAsCnt="3" ctrl="bfd" descr=""
name="" peerCtrl="" ttl="1">
                     <bgpRsPeerPfxPol tnBgpPeerPfxPolName=""/>
                     <bgpAsP asn="3" descr="" name=""/>
                  </bgpPeerP>
               </l3extRsPathL3OutAtt>
            </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>
            </l3extRsNodeL3OutAtt>
            <l3extLIfP name='portIpv4'>
                <l3extRsPathL3OutAtt tDn="topology/pod-1/paths-101/pathep-[eth1/3]"
```

```
ifInstT='l3-port' addr="10.10.10.2/24" mtu="1500" status="created,modified" />
            </l3extLIfP>

        </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>
             </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:

Example:

```
<fabricInst>
          <l3IfPol name="testL3IfPol" bfdIsis="enabled"/>
              <fabricLeafP name="LeNode" >
  <fabricRsLePortP tDn="uni/fabric/leportp-leaf_profile" />
 <fabricLeafS name="spsw" type="range">
 <fabricNodeBlk name="node101" to_="102" from_="101" />
 </fabricLeafS>
            </fabricLeafP>

           <fabricSpineP name="SpNode" >
 <fabricRsSpPortP tDn="uni/fabric/spportp-spine_profile" />
 <fabricSpineS name="spsw" type="range">
     <fabricNodeBlk name="node103" to_="103" from_="103" />
 </fabricSpineS>
         </fabricSpineP>

          <fabricLePortP name="leaf_profile">
 <fabricLFPortS name="leafIf" type="range">
<fabricPortBlk name="spBlk" fromCard="1" fromPort="49" toCard="1" toPort="49" />
       <fabricRsLePortPGrp tDn="uni/fabric/funcprof/leportgrp-LeTestPGrp" />
 </fabricLFPortS>
        </fabricLePortP>

       <fabricSpPortP name="spine_profile">
 <fabricSFPortS name="spineIf" type="range">
      <fabricPortBlk name="spBlk" fromCard="5" fromPort="1" toCard="5" toPort="2" />
      <fabricRsSpPortPGrp tDn="uni/fabric/funcprof/spportgrp-SpTestPGrp" />
 </fabricSFPortS>
     </fabricSpPortP>
```

```
 <fabricFuncP>
                <fabricLePortPGrp name = "LeTestPGrp">
 <fabricRsL3IfPol tnL3IfPolName="testL3IfPol"/>
               </fabricLePortPGrp>

            <fabricSpPortPGrp name = "SpTestPGrp">
 <fabricRsL3IfPol tnL3IfPolName="testL3IfPol"/>
           </fabricSpPortPGrp>

</fabricFuncP>

</fabricInst>
```

OSPF External Routed Networks

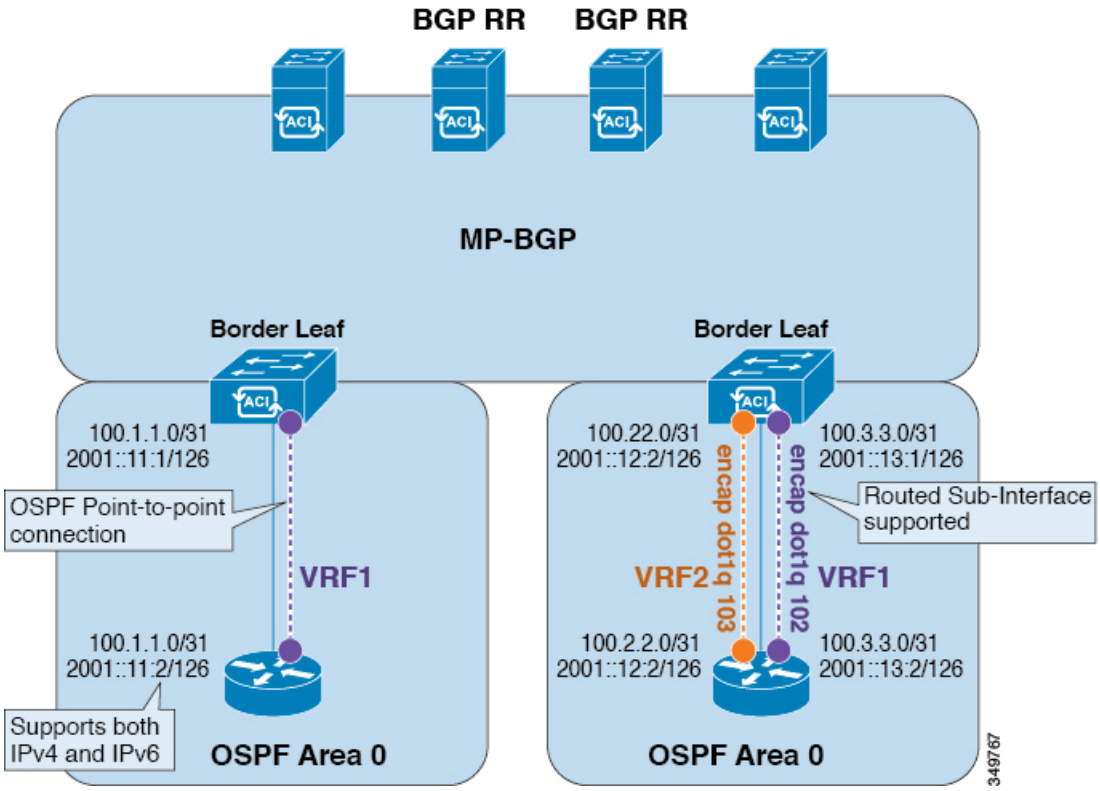
Use the procedures in the following sections to configure OSPF external routed networks.

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 Layer 2 and Layer 3 traffic. The SVIs are supported on ports, port channels, and virtual port channels (vPCs).

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 using the external bridge domain accessible through the other vPC node.
- When an OSPF time policy or a BGP, OSPF, or EIGRP address family policy is applied to an L3Out, you can observe the following behaviors:
  - If the L3Out and the policy are defined in the same tenant, then there is no change in behavior.
  - If the L3Out is configured in a user tenant other than the common tenant, the L3Out VRF instance is resolved to the common tenant, and the policy is defined in the common tenant, then only the default values are applied. Any change in the policy will not take effect.
- If a border leaf switch forms OSPF adjacency with two external switches and one of the two switches experiences a route loss while the adjacent switches does not, the Cisco ACI border leaf switch reconverges the route for both neighbors.

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 menu bar, 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:

a) In the **Name** field, enter a name (RtdOut).
b) Check the **OSPF** check box.
c) In the **OSPF Area ID** field, enter an area ID.
d) In the **OSPF Area Control** field, check the appropriate check box.
e) In the **OSPF Area Type** field, choose the appropriate area type.
f) In the **OSPF Area Cost** field, choose the appropriate value.
g) 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:

a) In the **Name** field, enter a name for the node profile. (borderLeaf).
b) In the **Nodes** field, click the **+** icon to display the **Select Node** dialog box.
c) In the **Node ID** field, from the drop-down list, choose the first node. (leaf1).
d) In the **Router ID** field, enter a unique router ID.
e) 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).

i) 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:

```
apic1(config)# vlan-domain dom_exampleCorp
apic1(config-vlan)# vlan 5-1000
apic1(config-vlan)# exit
```

Step 2 Configure the tenant VRF and enable policy enforcement on the VRF.

Example:

```
apic1(config)# tenant exampleCorp
apic1(config-tenant)# vrf context
 exampleCorp_v1
apic1(config-tenant-vrf)# contract enforce
apic1(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:

```
apic1(config-tenant)# bridge-domain exampleCorp_b1
apic1(config-tenant-bd)# vrf member exampleCorp_v1
apic1(config-tenant-bd)# exit
apic1(config-tenant)# interface bridge-domain exampleCorp_b1
apic1(config-tenant-interface)# ip address 172.1.1.1/24 scope public
apic1(config-tenant-interface)# exit
```

Step 4 Configure the VRF on a leaf.

Example:

```
apic1(config)# leaf 101
apic1(config-leaf)# vrf context tenant exampleCorp vrf exampleCorp_v1
```

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
apic1(config-leaf-ospf)# 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">
      <l3extRsL3DomAtt tDn="uni/l3dom-extdom"/>
      <l3extInstP name="extMgmt">
      </l3extInstP>
      <l3extLNodeP 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='l3-port' addr="192.168.62.1/24"/>
           <l3extRsPathL3OutAtt tDn="topology/pod-1/paths-102/pathep-[eth1/40]"
ifInstT='l3-port' addr="192.168.62.5/24"/>
            <ospfIfP/>
                                  </l3extLIfP>
      </l3extLNodeP>
      <l3extRsEctx tnFvCtxName="inb"/>
      <ospfExtP areaId="57" />
   </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`s on the same node

- Per-interface summarization (an EIGRP summary policy will apply to all interfaces configured under an L3Out)
- 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 `l3extLIfP`.
- `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.

The following configurations will cause the EIGRP neighbors to flap:

- Changing administrative distances or metric style (wide/narrow) through an EIGRP address family context in the VRF
- Setting the following configurations that will cause a table-map used internally to be updated:
  - Changing the route tag for the VRF
  - Setting configurations of import direction route control for an OSPF L3Out in the same VRF on the same border leaf switch as an EIGRP L3Out (for example, enabling or disabling the Route Control Enforcement “Import” option or changing routes that are allowed or denied for the import direction). Note that such configurations are not allowed in an EIGRP L3Out itself as the feature is not supported for EIGRP. However, the configurations in an OSPF L3Out still impacts EIGRP L3Outs in the same VRF and leaf switch. This is because the import route control for OSPF utilizes a table-map that is shared, for other purposes, with EIGRP in the same VRF on the same border leaf switch.

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
        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
    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
tenant1
# 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 tenant1
# 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
      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 l3out-L1
      switchport trunk allowed vlan 666 tenant ten_v6_cli external-svi l3out l3out_cli_L1
      switchport trunk allowed vlan 1513 tenant tenant16 external-svi l3out l3out-L1
      switchport trunk allowed vlan 1313 tenant tenant14 external-svi l3out l3out-L1
      switchport trunk allowed vlan 1413 tenant tenant15 external-svi l3out l3out-L1
      switchport trunk allowed vlan 1113 tenant tenant12 external-svi l3out 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 l3out-L1
      exit
    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
      exit
    router eigrp default
      exit
    router eigrp default
      vrf member tenant tenant1 vrf l3out
        autonomous-system 1001 l3out l3out-L1
        address-family ipv6 unicast
          inherit eigrp vrf-policy tenant1
          exit
        address-family ipv4 unicast
          inherit eigrp vrf-policy tenant1
          exit
        exit
      exit
```

Configuring EIGRP Using the REST API

Procedure

Step 1 Configure an EIGRP context policy.

Example:

```
<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=""/>
```

```
    </fvTenant>
</polUni>
```

Step 2 Configure an EIGRP interface policy.

Example:

```
<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:

```
<polUni>
    <fvTenant name="cisco_6">
        <fvCtx name="dev">
          <fvRsCtxToEigrpCtxAfPol tnEigrpCtxAfPolName="eigrp_ctx_pol_v4" af="1"/>
        </fvCtx>
    </fvTenant>
</polUni>
```

IPv6:

```
<polUni>
    <fvTenant name="cisco_6">
        <fvCtx name="dev">
          <fvRsCtxToEigrpCtxAfPol tnEigrpCtxAfPolName="eigrp_ctx_pol_v6" af="ipv6-ucast"/>
        </fvCtx>
    </fvTenant>
</polUni>
```

Step 4 Configure an EIGRP Layer3 Outside.

Example:

IPv4

```
<polUni>
    <fvTenant name="cisco_6">
        <l3extOut name="ext">
            <eigrpExtP asn="4001"/>
            <l3extLNodeP name="node1">
                <l3extLIfP name="intf_v4">
                    <l3extRsPathL3OutAtt addr="201.1.1.1/24" ifInstT="l3-port"
                      tDn="topology/pod-1/paths-101/pathep-[eth1/4]"/>
                    <eigrpIfP name="eigrp_ifp_v4">
                        <eigrpRsIfPol tnEigrpIfPolName="eigrp_if_pol_v4"/>
                    </eigrpIfP>
                </l3extLIfP>
            </l3extLNodeP>
        </l3extOut>
    </fvTenant>
</polUni>
```

IPv6

```
<polUni>
    <fvTenant name="cisco_6">
        <l3extOut name="ext">
```

```
            <eigrpExtP asn="4001"/>
            <l3extLNodeP name="node1">
                <l3extLIfP name="intf_v6">
                    <l3extRsPathL3OutAtt addr="2001::1/64" ifInstT="l3-port"
                      tDn="topology/pod-1/paths-101/pathep-[eth1/4]"/>
                    <eigrpIfP name="eigrp_ifp_v6">
                        <eigrpRsIfPol tnEigrpIfPolName="eigrp_if_pol_v6"/>
                    </eigrpIfP>
                </l3extLIfP>
            </l3extLNodeP>
        </l3extOut>
    </fvTenant>
</polUni>
```

IPv4 and IPv6

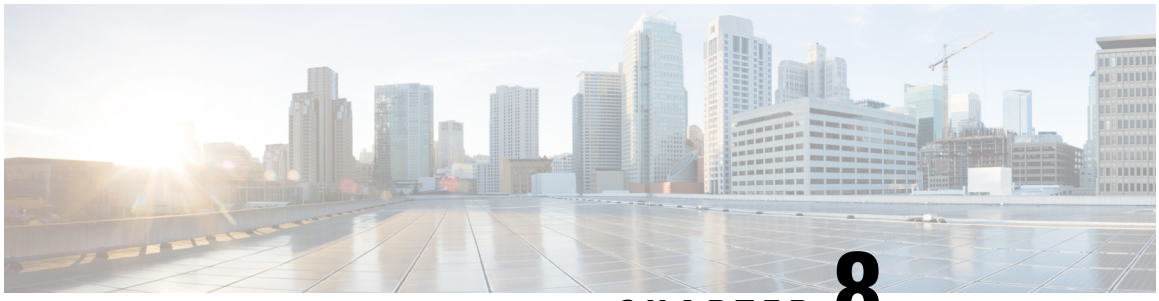
```
<polUni>
    <fvTenant name="cisco_6">
        <l3extOut name="ext">
            <eigrpExtP asn="4001"/>
            <l3extLNodeP name="node1">
                <l3extLIfP name="intf_v4">
                    <l3extRsPathL3OutAtt addr="201.1.1.1/24" ifInstT="l3-port"
                      tDn="topology/pod-1/paths-101/pathep-[eth1/4]"/>
                    <eigrpIfP name="eigrp_ifp_v4">
                        <eigrpRsIfPol tnEigrpIfPolName="eigrp_if_pol_v4"/>
                    </eigrpIfP>
                </l3extLIfP>

                <l3extLIfP name="intf_v6">
                    <l3extRsPathL3OutAtt addr="2001::1/64" ifInstT="l3-port"
                      tDn="topology/pod-1/paths-101/pathep-[eth1/4]"/>
                    <eigrpIfP name="eigrp_ifp_v6">
                        <eigrpRsIfPol tnEigrpIfPolName="eigrp_if_pol_v6"/>
                    </eigrpIfP>
                </l3extLIfP>
            </l3extLNodeP>
        </l3extOut>
    </fvTenant>
</polUni>
```

Step 5 (Optional) Configure the interface policy knobs.

Example:

```
<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".



CHAPTER 8

Route Summarization

This chapter contains the following sections:

Route Summarization

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 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.

Guidelines and Limitations

A route summarization policy configured under an external EPG will result in the summarized prefix being advertised to all of the BGP peers that are connected to the same border leaf switch and in the same VRF. This includes BGP peers that belong to different L3Outs if the same border leaf switch and VRF condition is met.

If you do not want this behavior to take place and you want to limit which BGP peers receive the aggregate route, block the routes where applicable using outbound route-maps on the respective L3Outs.

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:

```
<fvTenant name="common">
           <fvCtx 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]"
ifInstT='l3-port' addr="10.20.1.3/24/>
               </l3extLIfP>
           </l3extLNodeP>
         <bgpExtP />
           <l3extInstP name="InstP" >
         <l3extSubnet ip="10.0.0.0/8" scope="export-rtctrl">
           <l3extRsSubnetToRtSumm tDn="uni/tn-common/bgprtsum-bgp_rt_summ"/>
           <l3extRsSubnetToProfile tnRtctrlProfileName="rtprof"/>
         </l3extSubnet>
       </l3extInstP>
       <l3extRsEctx tnFvCtxName="vrf1"/>
   </l3extOut>
</fvTenant>
```

Step 2 Configure OSPF inter-area and external summarization using the following REST API:

Example:

```
<?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">
     <l3extRsNodeL3OutAtt rtrId="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>
    <ospfExtP areaCost="1" areaCtrl="redistribute,summary" areaId="backbone"
areaType="regular"/>
  </l3extOut>
  <!-- L3OUT Regular Area-->
  <l3extOut enforceRtctrl="export" name="l3_2">
    <l3extRsEctx tnFvCtxName="ctx0"/>
    <l3extLNodeP name="node-101">
     <l3extRsNodeL3OutAtt rtrId="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>
    <ospfExtP areaCost="1" areaCtrl="redistribute,summary" areaId="0.0.0.57"
areaType="regular"/>
  </l3extOut>
</fvTenant>
```

Step 3 Configure EIGRP summarization using the following REST API:

Example:

```
<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>
  <eigrpRtSummPol name="pol1" />
```

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:

```
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 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)# area 0.0.0.2 range 10.0.0.0/8 cost 20
```

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:

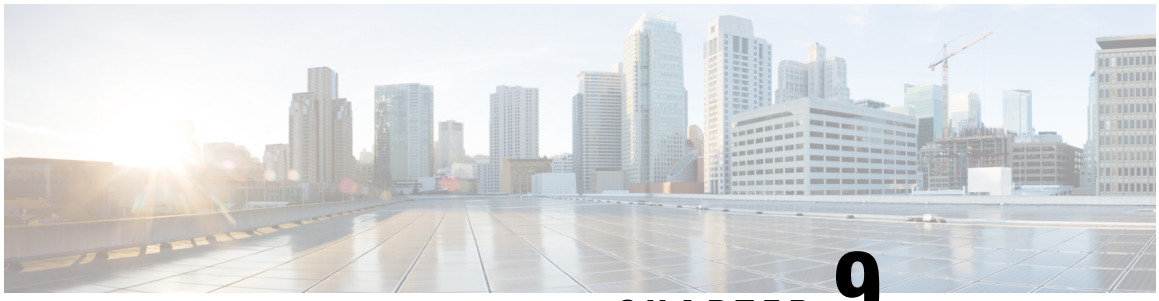
- Enter an IP address in the **IP Address** field.
- Check the check box next to **Export Route Control Subnet**.
- Check the check box next to **External Subnets for the External EPG**.
- From the **OSPF Route Summarization Policy** drop-down menu, choose either **default** for an existing (default) 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**.



CHAPTER 9

Route Control

This chapter contains the following sections:

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:

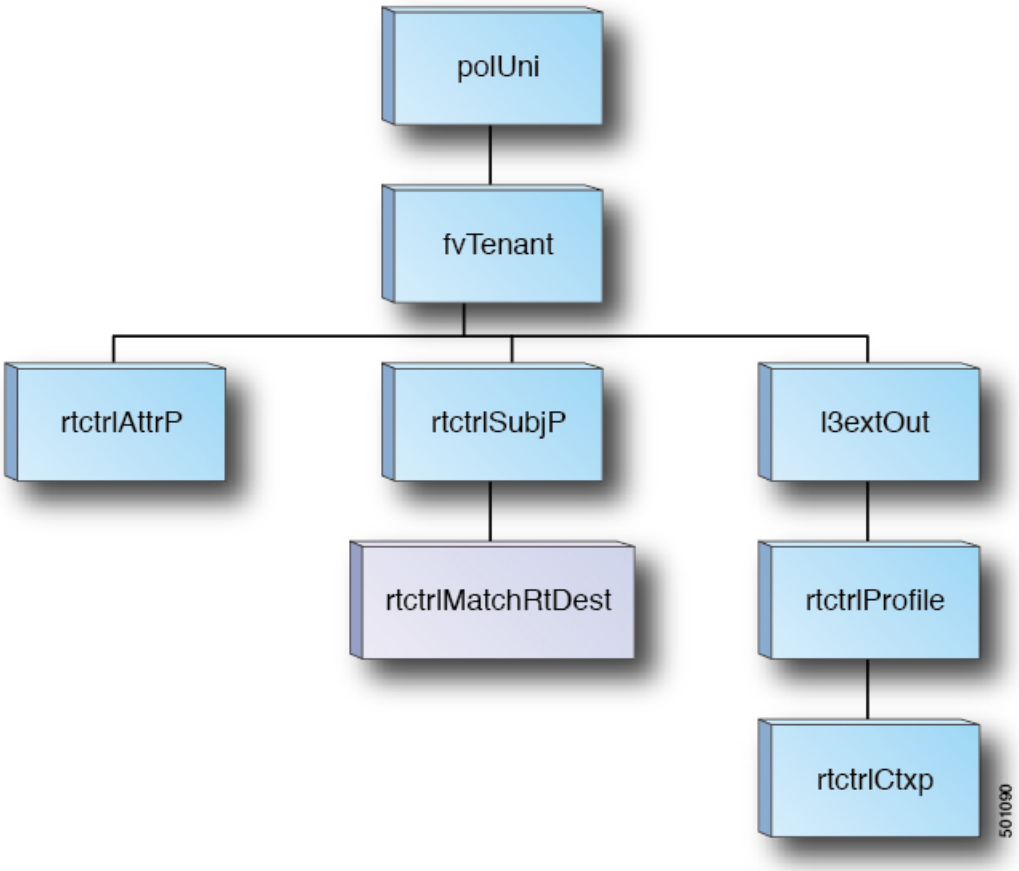
- Advertising BD subnets through BD to Layer 3 Out relation.

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.

Figure 16: External Policy Model of API



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).

Note 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:

```
<l3extOut annotation="" descr="" dn="uni/tn-t9/out-L3-out" enforceRtctrl="export"
name="L3-out" nameAlias="" ownerKey="" ownerTag="" targetDscp="unspecified">
    <rtctrlProfile annotation="" descr="" name="rpm_with_catch_all" nameAlias="" ownerKey=""
 ownerTag="" type="combinable">
        <rtctrlCtxP action="permit" annotation="" descr="" name="catch_all" nameAlias=""
order="0">
            <rtctrlScope annotation="" descr="" name="" nameAlias="">
                <rtctrlRsScopeToAttrP annotation="" tnRtctrlAttrPName="set_metric_type"/>
            </rtctrlScope>
        </rtctrlCtxP>
    </rtctrlProfile>
    <ospfExtP annotation="" areaCost="1" areaCtrl="redistribute,summary" areaId="backbone"
 areaType="regular" descr="" multipodInternal="no" nameAlias=""/>
    <l3extRsEctx annotation="" tnFvCtxName="ctx0"/>
   <l3extLNodeP annotation="" configIssues="" descr="" name="leaf" nameAlias="" ownerKey=""
 ownerTag="" tag="yellow-green" targetDscp="unspecified">
       <l3extRsNodeL3OutAtt annotation="" configIssues="" rtrId="20.2.0.2" rtrIdLoopBack="no"
 tDn="topology/pod-1/node-104">
          <l3extLoopBackIfP addr="14.1.1.1/32" annotation="" descr="" name="" nameAlias=""/>

            <l3extInfraNodeP annotation="" descr="" fabricExtCtrlPeering="no"
fabricExtIntersiteCtrlPeering="no" name="" nameAlias="" spineRole=""/>
        </l3extRsNodeL3OutAtt>
        <l3extLIfP annotation="" descr="" name="interface" nameAlias="" ownerKey=""
ownerTag="" tag="yellow-green">
            <ospfIfP annotation="" authKeyId="1" authType="none" descr="" name=""
nameAlias="">
                <ospfRsIfPol annotation="" tnOspfIfPolName=""/>
            </ospfIfP>
            <l3extRsPathL3OutAtt addr="36.1.1.1/24" annotation="" autostate="disabled"
descr="" encap="vlan-3063" encapScope="local" ifInstT="ext-svi" ipv6Dad="enabled" llAddr="::"
 mac="00:22:BD:F8:19:FF" mode="regular" mtu="inherit"
tDn="topology/pod-1/paths-104/pathep-[accBndlGrp_104_pc13]" targetDscp="unspecified"/>
            <l3extRsNdIfPol annotation="" tnNdIfPolName=""/>
            <l3extRsIngressQosDppPol annotation="" tnQosDppPolName=""/>
            <l3extRsEgressQosDppPol annotation="" tnQosDppPolName=""/>
        </l3extLIfP>
    </l3extLNodeP>
    <l3extInstP annotation="" descr="" exceptionTag="" floodOnEncap="disabled"
matchT="AtleastOne" name="epg" nameAlias="" prefGrMemb="exclude" prio="unspecified"
targetDscp="unspecified">
        <l3extRsInstPToProfile annotation="" direction="export"
tnRtctrlProfileName="rpm_with_catch_all"/>
       <l3extSubnet aggregate="" annotation="" descr="" ip="0.0.0.0/0" name="" nameAlias=""
 scope="import-security"/>
        <fvRsCustQosPol annotation="" tnQosCustomPolName=""/>
    </l3extInstP>
```

```
</l3extOut>


<rtctrlAttrP annotation="" descr="" dn="uni/tn-t9/attr-set_metric_type" name="set_metric_type"
 nameAlias="">
   <rtctrlSetRtMetricType annotation="" descr="" metricType="ospf-type1" name="" nameAlias=""
 type="metric-type"/>
</rtctrlAttrP>


<rtctrlSubjP annotation="" descr="" dn="uni/tn-t9/subj-catch_all_ip" name="catch_all_ip"
nameAlias="">
    <rtctrlMatchRtDest aggregate="yes" annotation="" descr="" ip="0.0.0.0/0" name=""
nameAlias=""/>
</rtctrlSubjP>
```

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:

```
<l3extOut annotation="" descr="" dn="uni/tn-t9/out-L3-out" enforceRtctrl="export"
name="L3-out" nameAlias="" ownerKey="" ownerTag="" targetDscp="unspecified">
    <rtctrlProfile annotation="" descr="" name="default-export" nameAlias="" ownerKey=""
ownerTag="" type="combinable">
        <rtctrlCtxP action="permit" annotation="" descr="" name="set-rule" nameAlias=""
order="0">
```

```
            <rtctrlScope annotation="" descr="" name="" nameAlias="">
                <rtctrlRsScopeToAttrP annotation="" tnRtctrlAttrPName="set_metric_type"/>
            </rtctrlScope>
        </rtctrlCtxP>
    </rtctrlProfile>
    <ospfExtP annotation="" areaCost="1" areaCtrl="redistribute,summary" areaId="backbone"
 areaType="regular" descr="" multipodInternal="no" nameAlias=""/>
    <l3extRsEctx annotation="" tnFvCtxName="ctx0"/>
   <l3extLNodeP annotation="" configIssues="" descr="" name="leaf" nameAlias="" ownerKey=""
 ownerTag="" tag="yellow-green" targetDscp="unspecified">
      <l3extRsNodeL3OutAtt annotation="" configIssues="" rtrId="20.2.0.2" rtrIdLoopBack="no"
 tDn="topology/pod-1/node-104">
          <l3extLoopBackIfP addr="14.1.1.1/32" annotation="" descr="" name="" nameAlias=""/>

            <l3extInfraNodeP annotation="" descr="" fabricExtCtrlPeering="no"
fabricExtIntersiteCtrlPeering="no" name="" nameAlias="" spineRole=""/>
        </l3extRsNodeL3OutAtt>
        <l3extLIfP annotation="" descr="" name="interface" nameAlias="" ownerKey=""
ownerTag="" tag="yellow-green">
            <ospfIfP annotation="" authKeyId="1" authType="none" descr="" name=""
nameAlias="">
                <ospfRsIfPol annotation="" tnOspfIfPolName=""/>
            </ospfIfP>
            <l3extRsPathL3OutAtt addr="36.1.1.1/24" annotation="" autostate="disabled"
descr="" encap="vlan-3063" encapScope="local" ifInstT="ext-svi" ipv6Dad="enabled" llAddr="::"
 mac="00:22:BD:F8:19:FF" mode="regular" mtu="inherit"
tDn="topology/pod-1/paths-104/pathep-[accBndlGrp_104_pc13]" targetDscp="unspecified"/>
            <l3extRsNdIfPol annotation="" tnNdIfPolName=""/>
            <l3extRsIngressQosDppPol annotation="" tnQosDppPolName=""/>
            <l3extRsEgressQosDppPol annotation="" tnQosDppPolName=""/>
        </l3extLIfP>
    </l3extLNodeP>
    <l3extInstP annotation="" descr="" exceptionTag="" floodOnEncap="disabled"
matchT="AtleastOne" name="epg" nameAlias="" prefGrMemb="exclude" prio="unspecified"
targetDscp="unspecified">
       <l3extSubnet aggregate="" annotation="" descr="" ip="0.0.0.0/0" name="" nameAlias=""
 scope="import-security"/>
        <fvRsCustQosPol annotation="" tnQosCustomPolName=""/>
    </l3extInstP>
</l3extOut>
```

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
```

```
leaf4# show ip prefix-list IPv4-st16391-2555939-exc-int-inferred-export-dst
% Policy IPv4-st16391-2555939-exc-int-inferred-export-dst not found
ifav82-leaf4# show ip prefix-list
IPv4-st16391-2555939-exc-int-out-default-export2set-rule0pfx-only-dst
ip prefix-list IPv4-st16391-2555939-exc-int-out-default-export2set-rule0pfx-only-dst: 1
entries
   seq 1 permit 209.165.201.0/27

leaf4#
```

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 L3Oout 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

	Command or Action	Purpose
Step 1	**configure** **Example:** `apic1# configure`	Enters configuration mode.
Step 2	**leaf** *node-id* **Example:** `apic1(config)# leaf 101`	Specifies the leaf to be configured.
Step 3	**template route group** *group-name* **tenant** *tenant-name* **Example:** `apic1(config-leaf)# template route group g1 tenant exampleCorp`	Creates a route group template. **Note** 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.
Step 4	**ip prefix permit** *prefix/masklen* [**le**{**32 | 128** }] **Example:** `apic1(config-route-group)# ip prefix permit 15.15.15.0/24`	Add IP prefix to the route group. **Note** 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.
Step 5	**community-list** [**standard | expanded**] *community-list-name expression* **Example:** `apic1(config-route-group)# community-list standard com1 65535:20`	This is an optional command. Add match criteria for community if community also needs to be matched along with IP prefix.

	Command or Action	Purpose

Step 6	**exit**
	Example:
	`apic1(config-route-group)# exit`
	`apic1(config-leaf)#`
		Exit template mode.

Step 7	**vrf context tenant** *tenant-name* **vrf** *vrf-name* [**l3out** {**BGP | EIGRP | OSPF | STATIC** }] **Example:** `apic1(config-leaf)# vrf context tenant exampleCorp vrf v1`	Enters a tenant VRF mode for the node. **Note** If you enter the optional **l3out** string, the L3Out must be an L3Out that you configured through the NX-OS CLI.
Step 8	**template route-profile** *profile-name* [*route-control-context-name order-value*] **Example:** `apic1(config-leaf-vrf)# template route-profile rp1 ctxl 1`	Creates a template containing set actions that should be applied to the matched routes.
Step 9	**set** *attribute value* **Example:** `apic1(config-leaf-vrf-template-route-profile)# set metric 128`	Add desired attributes (set actions) to the template.

Step 10	**exit**
	Example:
	`apic1(config-leaf-vrf-template-route-profile)# exit`
	`apic1(config-leaf-vrf)#`
		Exit template mode.

Step 11	**route-map** *map-name* **Example:** `apic1(config-leaf-vrf)# route-map bgpMap`	Create a route-map and enter the route-map configuration mode.
Step 12	**match route group** *group-name* [**order** *number*] [**deny**] **Example:** `apic1(config-leaf-vrf-route-map)# match route group g1 order 1`	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**.
Step 13	**inherit route-profile** *profile-name* **Example:** `apic1(config-leaf-vrf-route-map-match)# inherit route-profile rp1`	Inherit a route-profile (set actions). **Note** These actions will be applied to the matched routes. Alternatively, the set actions can be configured inline instead of inheriting a route-profile.

	Command or Action	Purpose

Step 14	**exit**
	Example:
	`apic1(config-leaf-vrf-route-map-match)# exit`
	`apic1(config-leaf-vrf-route-map)#`
		Exit match mode.

Step 15	**exit**
	Example:
	`apic1(config-leaf-vrf-route-map)# exit`
	`apic1(config-leaf-vrf)#`
		Exit route map configuration mode.

Step 16	**exit**
	Example:
	`apic1(config-leaf-vrf)# exit`
	`apic1(config-leaf)#`
		Exit VRF configuration mode.

Step 17	**router bgp** *fabric-asn* **Example:** `apic1(config-leaf)# router bgp 100`	Configure the leaf node.
Step 18	**vrf member tenant** *t1* **vrf** *v1* **Example:** `apic1(config-leaf-bgp)# vrf member tenant t1 vrf v1`	Set the BGP's VRF membership and the tenant for the BGP policy.
Step 19	**neighbor** *IP-address-of-neighbor* **Example:** `apic1(config-leaf-bgp-vrf)# neighbor 15.15.15.2`	Configure a BGP neighbor.
Step 20	**route-map** *map-name* {**in** | **out** } **Example:** `apic1(config-leaf-bgp-vrf-neighbor)# route-map bgpMap out`	Configure the route map for a BGP neighbor.

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 version="1.0" encoding="UTF-8"?>
<fvTenant name="PM" status="">
   <rtctrlAttrP name="set_dest">
      <rtctrlSetComm community="regular:as2-nn2:5:24" />
   </rtctrlAttrP>
   <rtctrlSubjP 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="" />
   </rtctrlSubjP>
   <rtctrlSubjP name="deny_dest">
      <rtctrlMatchRtDest ip="192.168.0.0/24" />
   </rtctrlSubjP>
   <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">
            <l3extRsPathL3OutAtt 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>
      <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>
   </l3extOut>
   <fvBD name="testBD">
      <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

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 section describes how to create a route map using the NX-OS 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:

```
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-match)# set tag 200
apic1(config-leaf-vrf-route-map-match)# set local-preference 64
apic1(config-leaf)# router bgp 100
apic1(config-bgp)# vrf member tenant exampleCorp vrf v1
```

```
apic1(config-leaf-bgp-vrf)# neighbor 3.3.3.3
apic1(config-leaf-bgp-vrf-neighbor)# route-map bgpMap in
```

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)# set metric 256
apic1(config-leaf-vrf)# template route-profile bd-rtctrl
apic1(config-leaf-vrf-template-route-profile)# set metric 128

#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)#exit
apic1(config-leaf-vrf-route-map)# match prefix-list p1
apic1(config-leaf-vrf-route-map-match)#exit
apic1(config-leaf-vrf-route-map)# match bridge-domain bd2
apic1(config-leaf-vrf-route-map-match)# 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.

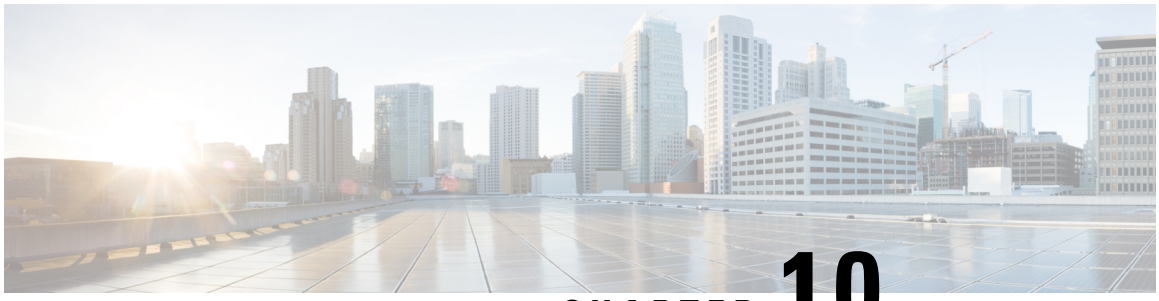
Procedure

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="" tag="yellow-green"
targetDscp="unspecified">
            <l3extRsNodeL3OutAtt rtrId="1.2.3.4" rtrIdLoopBack="yes"
```

```
tDn="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>
```



CHAPTER 10

Common Pervasive Gateway

This chapter contains the following sections:

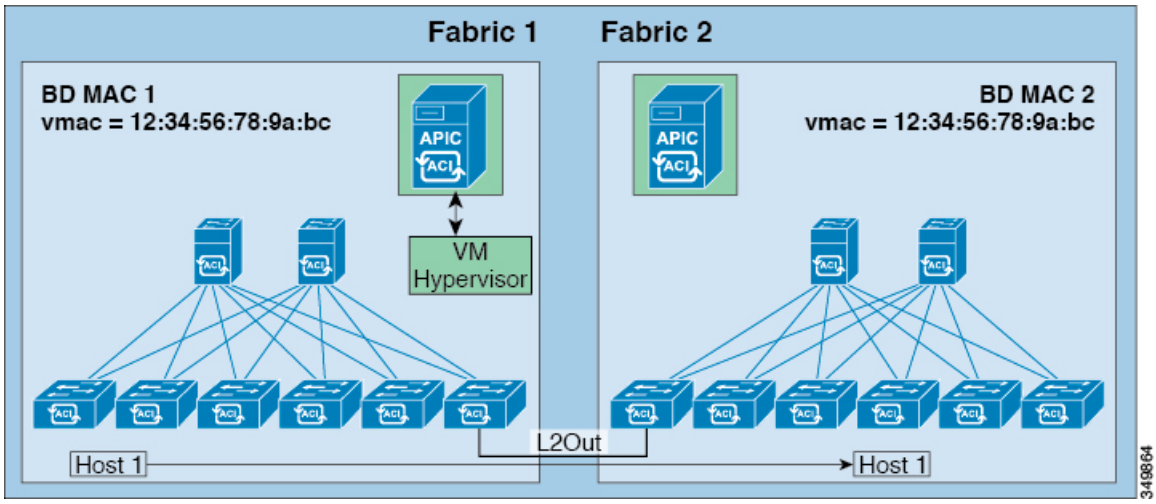
Overview

Note 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.



Note 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

Step 1 On the menu bar, click **Tenants**.

Step 2 In the **Navigation** pane, expand the *Tenant_name* > **Networking** > **Bridge Domains**.

Step 3 Right-click **Bridge Domains**, and click **Create Bridge Domain**.

Step 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

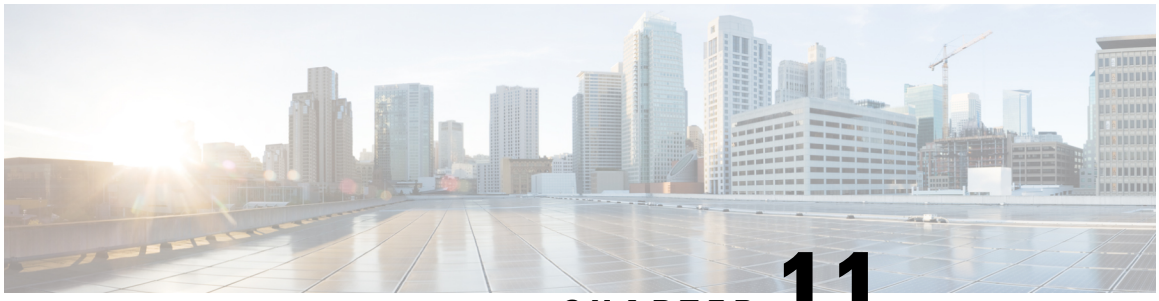
Configure common pervasive gateway.

In the following example REST API XML, the bolded text is relevant to configuring a common pervasive gateway.

Example:

```
<?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">
      <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"/>
      </fvAEPg>
    </fvAp>
  </fvTenant>
</polUni>
```



CHAPTER 11

Static Route on a Bridge Domain

This chapter contains the following sections:

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 (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 assoicated with the pervasive BD.

Before you begin

The tenant, VRF, BD, and EPG are created.

Procedure

Step 1 On the menu bar, click **Tenants** > ***tenant-name*** .

Step 2 In the Navigation pane, expand **Application Profiles** and click the application profile name.

Step 3 Click **Application EPGs** and expand the EPG for the static route.

Step 4 Expand **Subnets**, right-click the subnet for the static route, and choose **Create Endpoints Behind EPG Subnet**.

Step 5 Enter the **NextHop IP Address** for the endpoint and click **Update**.

Step 6 Click **Submit**.

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 assoicated with the pervasive BD.

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `apic1# configure`	Enters configuration mode.
Step 2	**tenant** *tenant-name* **Example:** `apic1(config)# tenant t1`	Creates a tenant or enters tenant configuration mode.
Step 3	**application** *ap-name* **Example:** `apic1(config-tenant)# application ap1`	Creates an application profile or enters application profile mode.

	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 ap1
      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.

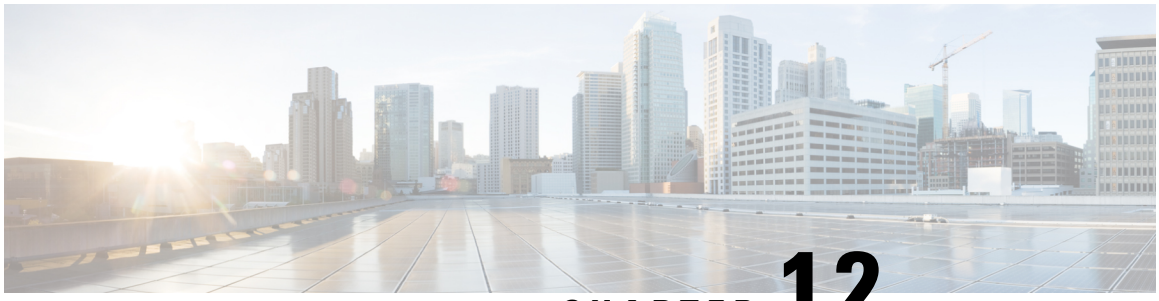
Procedure

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="2002:0db8:85a3:0000:0000:8a2e:0370:7344/128"
ctrl="no-default-gateway"   >
              <fvEpReachability>
```

```
            <ipNexthopEpP  nhAddr="2001:0db8:85a3:0000:0000:8a2e:0370:7343/128" />
        </fvEpReachability>
    </fvSubnet>
</fvAEPg>
```



CHAPTER 12

MP-BGP Route Reflectors

This chapter contains the following sections:

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.

Note ACI supports the following BGP features. NX-OS BGP features not listed below are not currently supported in ACI.

Table 4: BGP Peer Connectivity Profile Features

BGP Features	Feature Description	NX-OS Equivalent Commands
Allow Self-AS	Works with Allowed AS Number Count setting.	**allowas-in**
Disable peer AS check	Disable checking of the peer AS number when advertising.	**disable-peer-as-check**
Next-hop self	Always set the next hop attribute to the local peering address.	**next-hop-self**
Send community	Send the community attribute to the neighbor.	**send-community**
Send community extended	Send the extended community attribute to the neighbor.	**send-community extended**
Password	The BGP MD5 authentication.	**password**
Allowed AS Number Count	Works with Allow Self-AS feature.	**allowas-in**
Disable connected check	Disable connected check for the directly connected EBGP neighbors (allowing EBGP neighbor peering from the loopbacks).	
TTL	Set the TTL value for EBGP multihop connections. It is only valid for EBGP.	**ebgp-multihop <TTL>**
Autonomous System Number	Remote Autonomous System number of the peer.	**neighbor <x.x.x.x> remote-as**
Local Autonomous System Number Configuration	Options when using the Local AS feature. (No Prepend+replace-AS+dual-AS etc).	
Local Autonomous System Number	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.	**local-as xxx <no-prepend> <replace-as> <dual-as>**

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
```

```
apic1(config-bgp-fabric)# asn 100
apic1(config-bgp-fabric)# route-reflector spine 104,105
```

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

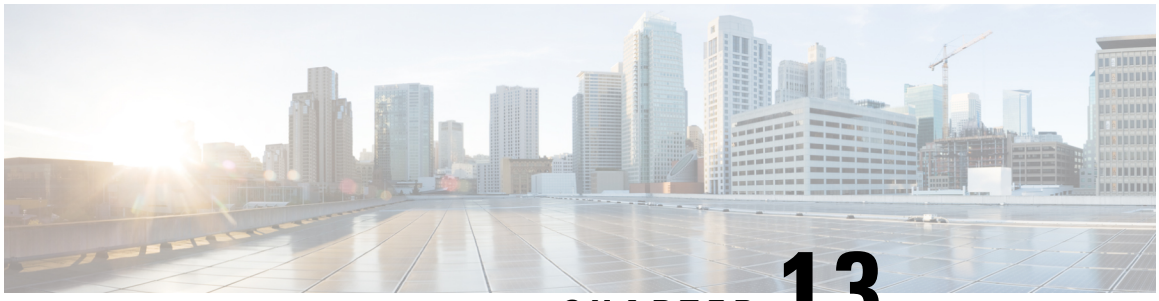
   Example:

   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.



CHAPTER 13

Switch Virtual Interface

This chapter contains the following sections:

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:

Figure 17: Local Scope Encapsulation and One Layer 3 Out

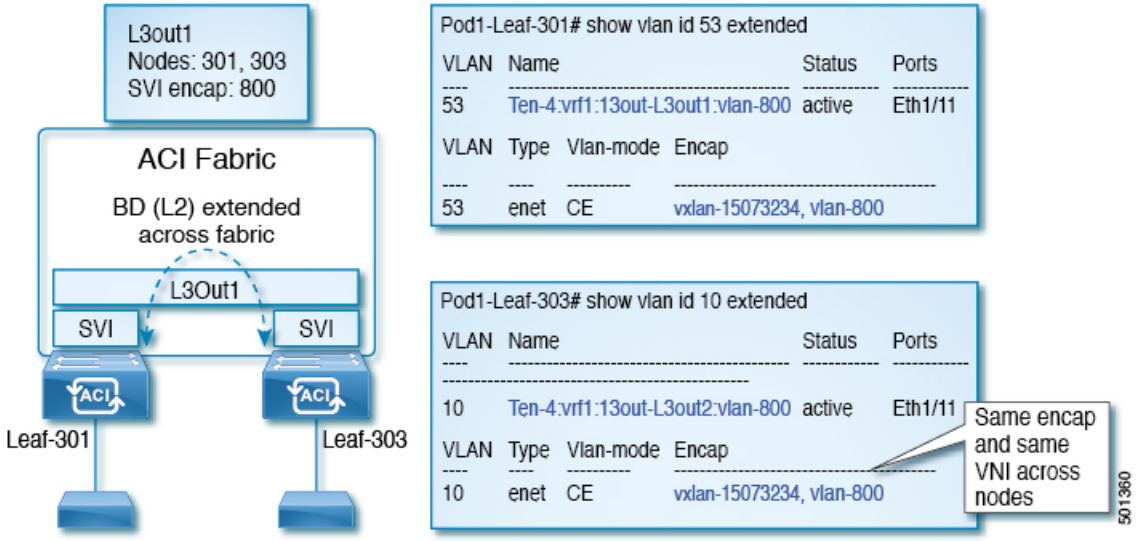
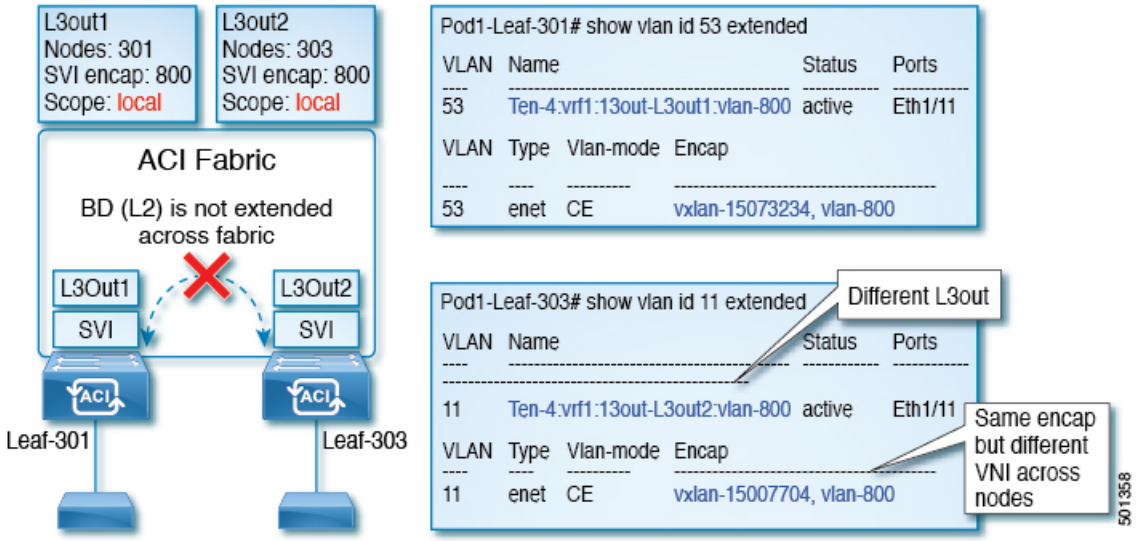


Figure 18: Local Scope Encapsulation and Two Layer 3 Outs

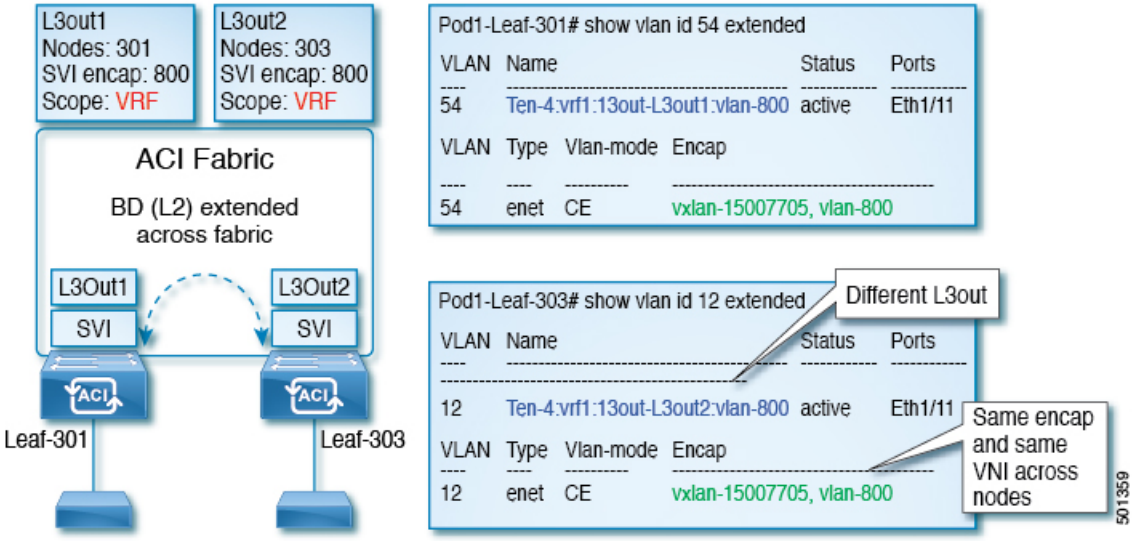


Starting with Cisco APIC release 2.3, it is now possible to choose the behavior when deploying two (or more) Layer 3 Outs using the same external encapsulation (SVI).

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*.

Figure 19: 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 5: Encapsulation Scope Syntax

CLI	API	GUI
l3out	local	Local
vrf	ctx	VRF

Note 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

	Command or Action	Purpose
Step 1	Enter the configure mode. **Example:** `apic1# configure`	Enters the configuration mode.
Step 2	Enter the switch mode. **Example:** `apic1(config)# leaf 104`	Enters the switch mode.
Step 3	Create the VLAN interface. **Example:** `apic1(config-leaf)# interface vlan 2001`	Creates the VLAN interface. The VLAN range is 1-4094.

	Command or Action	Purpose
Step 4	Specify the encapsulation scope. **Example:** `apic1(config-leaf-if)# encap scope vrf context`	Specifies the encapsulation scope.
Step 5	Exit the interface mode. **Example:** `apic1(config-leaf-if)# exit`	Exits the interface mode.

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 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="1.2.3.4/24" descr="" encap="vlan-2001" encapScope="ctx"
ifInstT="ext-svi" 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"/>
     <l3extRsNdIfPol tnNdIfPolName=""/>
     <l3extRsIngressQosDppPol tnQosDppPolName=""/>
     <l3extRsEgressQosDppPol tnQosDppPolName=""/>
    </l3extLIfP>
   </l3extLNodeP>
   <l3extInstP descr="" matchT="AtleastOne" name="epg1" nameAlias="" prefGrMemb="exclude"
prio="unspecified" targetDscp="unspecified">
    <l3extSubnet aggregate="" descr="" ip="101.10.10.1/24" name="" nameAlias=""
scope="import-security"/>
    <fvRsCustQosPol tnQosCustomPolName=""/>
   </l3extInstP>
  </l3extOut>
```

```
 </fvTenant>
</polUni>
```

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 6: SVI Auto State

SVI Auto State	Description of SVI State
Disabled	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.
Enabled	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.

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

	Command or Action	Purpose
Step 1	Enter the configure mode. **Example:** `apic1# configure`	Enters the configuration mode.
Step 2	Enter the switch mode. **Example:**	Enters the switch mode.

	Command or Action	Purpose
	`apic1(config)# leaf 104`	
Step 3	Create the VLAN interface. **Example:** `apic1(config-leaf)# interface vlan 2001`	Creates the VLAN interface. The VLAN range is 1-4094.
Step 4	Enable SVI auto state. **Example:** `apic1(config-leaf-if)# autostate`	Enables SVI auto state. By default, the SVI auto state value is not enabled.
Step 5	Exit the interface mode. **Example:** `apic1(config-leaf-if)# exit`	Exits the interface mode.

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.

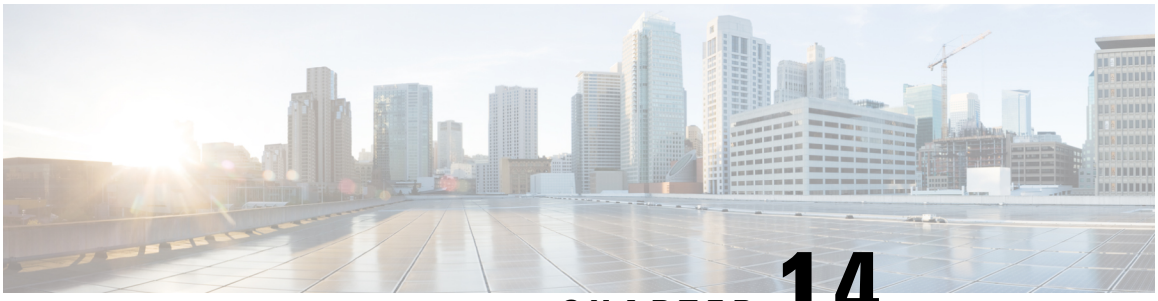
Procedure

Enable the SVI auto state value.

Example:

```
<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 14

Shared Services

This chapter contains the following sections:

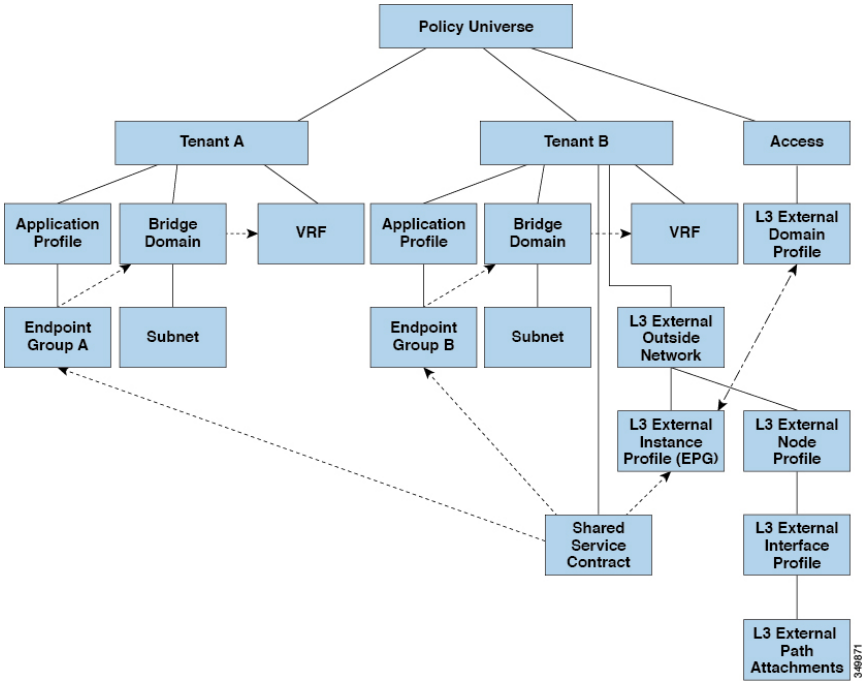
Shared Layer 3 Out

A shared Layer 3 Outside (L3Out or `l3extOut`) configuration provides routed connectivity to an external network as a shared service across VRF instances or tenants. An external EPG instance profile (external EPG or `l3extInstP`) in an L3Out provides the configurations to control which routes can be shared from both the routing perspective and contract perspective. A contract under an external EPG determines to which VRF instances or tenants those routes should be leaked.

An L3Out can be provisioned as a shared service in any tenant (*user*, `common`, `infra`, or `mgmt`). An EPG in any tenant can use a shared services contract to connect with an external EPG regardless of where in the fabric that external EPG is provisioned. This simplifies the provisioning of routed connectivity to external networks; multiple tenants can share a single external EPG for routed connectivity to external networks. Sharing an external EPG is more efficient because it consumes only one session on the switch regardless of how many EPGs use the single shared external EPG.

The figure below illustrates the major policy model objects that are configured for a shared external EPG.

Figure 20: Shared L3Out Policy Model



Take note of the following guidelines and limitations for shared L3Out network configurations:

- No tenant limitations: Tenants A and B can be any kind of tenant (*user*, `common`, `infra`, `mgmt`). The shared external 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 instance, but they are not required to do so. EPG A and EPG B are in different bridge domains and different VRF instances but still share the same external EPG.
- A subnet can be *private*, *public*, or *shared*. A subnet that is to be advertised into a consumer or provider EPG of an L3Out must be set to *shared*. A subnet that is to be exported to an L3Out must be set to *public*.
- The shared service contract is exported from the tenant that contains the external EPG that provides shared L3Out 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 L3Out; this configuration is not supported.
- The external EPG as a shared service provider is supported, but only with non-external EPG consumers (where the L3Out EPG is the same as the external EPG).
- Traffic Disruption (Flap): When an external EPG is configured with an external subnet of 0.0.0.0/0 with the scope property of the external EPG subnet set to shared route control (*shared-rctrl*), or shared security (*shared-security*), the VRF instance is redeployed with a global `pcTag`. This will disrupt all the external traffic in that VRF instance (because the VRF instance is redeployed with a global `pcTag`).
- Prefixes for a shared L3Out must to be unique. Multiple shared L3Out configurations with the same prefix in the same VRF instance will not work. Be sure that the external subnets (external prefixes) that are advertised into a VRF instance are unique (the same external subnet cannot belong to multiple external EPGs). An L3Out configuration (for example, named `L3Out1`) with prefix1 and a second L3Out

configuration (for example, named `L3Out2`) also with prefix1 belonging to the same VRF instance will not work (because only 1 pcTag is deployed).

- Different behaviors of L3Out are possible when configured on the same leaf switch under the same VRF instance. The two possible scenarios are as follows:
  - Scenario 1 has an L3Out with an SVI interface and two subnets (10.10.10.0/24 and 0.0.0.0/0) defined. If ingress traffic on the L3Out network has the matching prefix 10.10.10.0/24, then the ingress traffic uses the external EPG pcTag. If ingress traffic on the L3Out network has the matching default prefix 0.0.0.0/0, then the ingress traffic uses the external bridge pcTag.
  - Scenario 2 has an L3Out 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 L3Out network has the matching prefix 10.10.10.0/24, then the ingress traffic uses the external EPG pcTag. If ingress traffic on the L3Out network has the matching default prefix 0.0.0.0/0, then the ingress traffic uses the VRF instance pcTag.
  - As a result of these described behaviors, the following use cases are possible if the same VRF instance and same leaf switch are configured with `L3Out-A` and `L3Out-B` using an SVI interface:

    Case 1 is for `L3Out-A`: This external EPG has two subnets defined: 10.10.10.0/24 and 0.0.0.0/1. If ingress traffic on `L3Out-A` has the matching prefix 10.10.10.0/24, it uses the external EPG pcTag and `contract-A`, which is associated with `L3Out-A`. When egress traffic on `L3Out-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 pcTag and `contract-A`.

    Case 2 is for `L3Out-B`: This external EPG has one subnet defined: 0.0.0.0/0. When ingress traffic on `L3Out-B` has the matching prefix10.10.10.0/24 (which is defined under `L3Out-A`), it uses the external EPG pcTag of `L3Out-A` and the `contract-A`, which is tied with `L3Out-A`. It does not use `contract-B`, which is tied with `L3Out-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 L3Out network configuration (for example, named `L3Out-1`) with VRF1 is called `provider1`.
    - A second L3Out network configuration (for example, named `L3Out-2`) with VRF2 is called `provider2`.
    - `L3Out-1` VRF1 advertises a default route to the Internet, 0.0.0.0/0, which enables both *shared-rtctrl* and *shared-security*.
    - `L3Out-2` VRF2 advertises specific subnets to DNS and NTP, 192.0.0.0/8, which enables *shared-rtctrl*.

- `L3Out-2` VRF2 has specific subnet 192.1.0.0/16, which enables *shared-security*.
- **Variation A**: EPG traffic goes to multiple VRF instances.
  - Communications between EPG1 and `L3Out-1` is regulated by an *allow_all* contract.
  - Communications between EPG1 and `L3Out-2` is regulated by an *allow_all* contract.

  Result: Traffic from EPG1 to `L3Out-2` also goes to 192.2.x.x.
- **Variation B**: An EPG conforms to the *allow_all* contract of a second shared L3Out network.
  - Communications between EPG1 and `L3Out-1` is regulated by an *allow_all* contract.
  - Communications between EPG1 and `L3Out-2` is regulated by an *allow_icmp* contract.

  Result: Traffic from EPG1 to `L3Out-2` to 192.2.x.x conforms to the *allow_all* contract.

- **Case 2** configuration details:
  - An external EPG has one shared prefix and other non-shared prefixes.
  - Traffic coming in with `src = non-shared` is allowed to go to the EPG.
    - **Variation A**: Unintended traffic goes through an EPG.

      External EPG traffic goes through an L3Out that has these prefixes:

      Unordered List bullet 5
      - `192.0.0.0/8 = import-security, shared-rtctrl`

      Unordered List bullet 5
      - `192.1.0.0/16 = shared-security`

      Unordered List bullet 5
      - The EPG has `1.1.0.0/16 = shared`.

      Result: Traffic going from 192.2.x.x also goes through to the EPG.
    - **Variation B**: Unintended traffic goes through an EPG. Traffic coming in a shared L3Out can go through the EPG.

      Unordered List bullet 5
      - The shared L3Out VRF instance has an EPG with `pcTag = prov vrf` and a contract set to *allow_all*.

      Unordered List bullet 5
      - The EPG `<subnet> = shared`.

      Result: The traffic coming in on the L3Out can go through the EPG.

Layer 3 Out to Layer 3 Out Inter-VRF Leaking

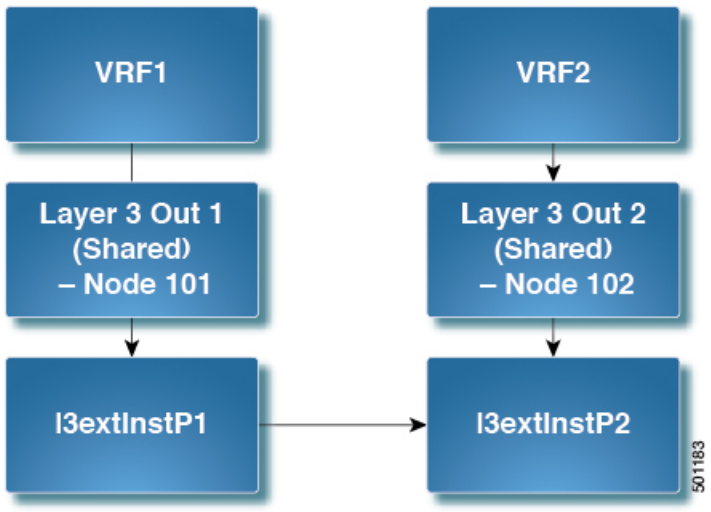
Starting with Cisco APIC release 2.2(2e) , when there are two Layer 3 Outs in two different VRFs, inter-VRF leaking is supported.

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 (l3extInstP) 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:

```
<tenant name="t1_provider">
<fvCtx name="VRF1">
<l3extOut name="T0-o1-L3OUT-1">
                <l3extRsEctx tnFvCtxName="o1"/>
                <ospfExtP areaId='60'/>
                <l3extInstP name="l3extInstP-1">
                <fvRsProv tnVzBrCPName="vzBrCP-1">
                </fvRsProv>
                <l3extSubnet ip="192.168.2.0/24" scope="shared-rtctrl, shared-security"
aggregate=""/>
                </l3extInstP>
</l3extOut>
</tenant>
```

Step 2 Configure the consumer Layer 3 Out.

Example:

```
<tenant name="t1_consumer">
<fvCtx name="VRF2">
<l3extOut name="T0-o1-L3OUT-1">
                <l3extRsEctx tnFvCtxName="o1"/>
                <ospfExtP areaId='70'/>
                <l3extInstP name="l3extInstP-2">
                <fvRsCons tnVzBrCPName="vzBrCP-1">
                </fvRsCons>
                <l3extSubnet ip="199.16.2.0/24" scope="shared-rtctrl, shared-security"
         aggregate=""/>
                </l3extInstP>
</l3extOut>
</tenant>
```

Configuring Shared Layer 3 Out Inter-VRF Leaking Using the NX-OS Style CLI - Named Example

Procedure

	Command or Action	Purpose
Step 1	Enter the configure mode. **Example:** `apic1# configure`	

	Command or Action	Purpose

Step 2	Configure the provider Layer 3 Out.
	Example:
	`apic1(config)# tenant t1_provider`
	`apic1(config-tenant)# external-l3 epg l3extInstP-1 l3out T0-o1-L3OUT-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`
	`apic1(config)# leaf 101`
	`apic1(config-leaf)# vrf context tenant t1_provider vrf VRF1 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-1 permit 192.168.2.0/24`
	`apic1(config-leaf-vrf-route-map)# match prefix-list l3extInstP-1`
	`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 3	Configure the consumer Layer 3 Out.
	Example:
	`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-tenant)# exit`
	`apic1(config)# leaf 101`
	`apic1(config-leaf)# vrf context tenant t1_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-vrf)# exit`
	`apic1(config-leaf)# exit`	

	Command or Action	Purpose
	apic1(config)#	

Configuring Shared Layer 3 Out Inter-VRF Leaking Using the NX-OS Style CLI - Implicit Example

Procedure

	Command or Action	Purpose
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**	

	Command or Action	Purpose
	apic1(config-tenant-l3ext-epg)# **exit** apic1(config-tenant)# **exit**	
Step 6	Configure the provider export map. **Example:** apic1(config)# **leaf 101** apic1(config-leaf)# **vrf context tenant t1_provider vrf VRF1** apic1(config-leaf-vrf)# **route-map map1** apic1(config-leaf-vrf-route-map)# **ip prefix-list p1 permit 192.168.2.0/24** apic1(config-leaf-vrf-route-map)# **match prefix-list p1** apic1(config-leaf-vrf-route-map-match)# **exit** apic1(config-leaf-vrf-route-map)# **exit** apic1(config-leaf-vrf)# **export map map1** apic1(config-leaf-vrf)# **exit** apic1(config-leaf)# **exit**	
Step 7	Configure the consumer external Layer 3 EPG. **Example:** apic1(config)# **tenant t1_consumer** apic1(config-tenant)# **external-l3 epg l3extInstP-2** 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-tenant)# **exit**	
Step 8	Configure the consumer export map. **Example:** apic1(config)# **leaf 101** apic1(config-leaf)# **vrf context tenant t1_consumer vrf VRF2** apic1(config-leaf-vrf)# **route-map map2** apic1(config-leaf-vrf-route-map)# **ip prefix-list p2 permit 199.16.2.0/24** apic1(config-leaf-vrf-route-map)# **match prefix-list p2** apic1(config-leaf-vrf-route-map-match)# **exit** apic1(config-leaf-vrf-route-map)# **exit** apic1(config-leaf-vrf)# **export map map2** apic1(config-leaf-vrf)# **exit** apic1(config-leaf)# **exit** apic1(config)#	

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.

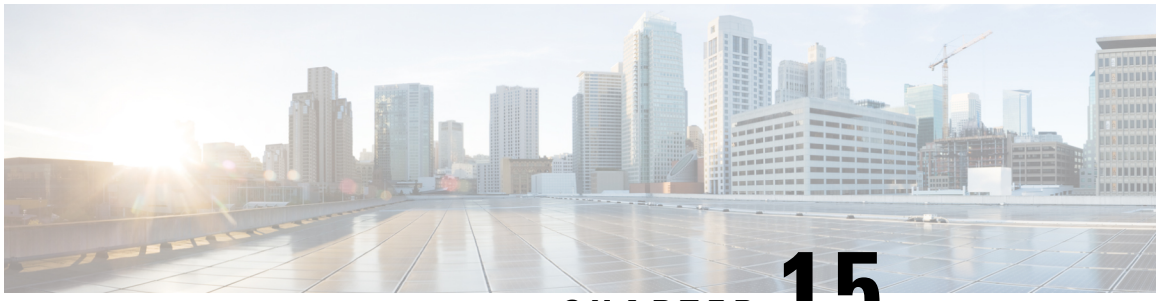
b) Expand **Subnet**, and in the **Create Subnet** dialog box, and in the **IP Address** field, enter the match IP address. Click **OK**.

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.



CHAPTER 15

Interleak Redistribution for MP-BGP

This chapter contains the following sections:

Overview Interleak Redistribution for MP-BGP

This topic provides how to configure an interleak redistribution in the Cisco Application Centric Infrastructure (ACI) fabric using Cisco Application Policy Infrastructure Controller (APIC).

In Cisco ACI, a border leaf node on which Layer 3 Outsides (L3Outs) are deployed redistributes L3Out routes to the BGP IPv4/IPv6 address family and then to the MP-BGP VPNv4/VPNv6 address family along with the VRF information so that L3Out routes are distributed from a border leaf node to other leaf nodes through the spine nodes. Interleak redistribution in the Cisco ACI fabric refers to this redistribution of L3Out routes to the BGP IPv4/IPv6 address family. By default, interleak happens for all L3Out routes, such as routes learned through dynamic routing protocols, static routes, and directly-connected subnets of L3Out interfaces, except for routes learned through BGP. Routes learned through BGP are already in the BGP IPv4/IPv6 table and are ready to be exported to MP-BGP VPNv4/VPNv6 without interleak.

Interleak redistribution allows users to apply a route-map to redistribute L3Out routes selectively into BGP to control which routes should be visible to other leaf nodes, or to set some attributes to the routes, such as BGP community, preference, metric, and so on. This redistribution enables selective transit routing to be performed on another border leaf node based on the attributes set by the ingress border leaf node or so that other leaf nodes can prefer routes from one border leaf node to another.

Applying a route map to interleak redistribution from OSPF and EIGRP routes has been available in earlier releases.

Configuring a Route Map for Interleak Redistribution Using the GUI

Route maps for interleak redistribution can be created under **Tenant** > **Policies** > **Protocol** > **Route Maps for BGP Dampening, Inter-leak**.

Before you begin

Create the tenant.

Procedure

Step 1 On the menu bar, click **Tenants**.

Step 2 In the Work pane, double click the tenant's name.

Step 3 In the **Navigation** pane, expand ***tenant_name*** > **Policies** > **Protocol** > **Route Maps for BGP Dampening, Inter-leak**.

Step 4 Right-click **Route Maps for BGP Dampening, Inter-leak** and click **Create Route Maps for BGP Dampening, Inter-leak**.

Step 5 In the **Create Route Maps for BGP Dampening, Inter-leak** dialog box, perform the following actions:

a) In the **Name** field, enter a name for the route map to control interleak (redistribution to BGP).
b) In the **Type** field, you must choose **Match Routing Policy Only**.

Step 6 In the **Contexts** area, click the **+** sign to open the **Create Route Control Context** dialog box, and perform the following actions:

a) Populate the **Order** and the **Name** fields as desired.
b) In the **Action** field, choose **Permit**.
c) In the **Match Rule** field, choose your desired match rule or create a new one.
d) In the **Set Rule** field, choose your desired set rule or create a new one.
e) Click **OK**.

Repeat this step for each route control context that you need to create.

Step 7 In the **Create Route Maps for BGP Dampening, Inter-leak** dialog box, click **Submit**.

Applying a Route Map for Interleak Redistribution Using the GUI

A route map to customize interleak redistribution from a specific L3Out must be applied through the L3Out.

Before you begin

Create the tenant, VRF, and L3Out.

Procedure

Step 1 On the menu bar, click **Tenants**.

Step 2 In the Work pane, double click the tenant's name.

Step 3 In the **Navigation** pane, expand ***tenant_name*** > **Networking** > **L3Outs** > ***L3Out_name***.

Step 4 Click the **Policy** > **Main** tab to access the **Properties** window for this L3Out.

Step 5 For the OSPF or EIGRP routes, perform the following actions:

a) In the **Route Profile for Interleak** field, choose or create a route map/profile.

b) In the Work pane, click **Submit**, then **Submit Changes**.

Configuring Interleak Redistribution Using the NX-OS-Style CLI

The following procedure describes how to configure the interleak redistribution using the NX-OS-style CLI.

Before you begin

Create the tenant, VRF, and L3Out.

Procedure

Step 1 Configure the route map for interleak redistribution for the border leaf node.

Example:

The following example configures the route map `CLI_RP` with an IP prefix-list `CLI_PFX1` for tenant `CLI_TEST` and VRF `VRF1`:

```
apic1# conf t
apic1(config)# leaf 101
apic1(config-leaf)# vrf context tenant CLI_TEST vrf VRF1
apic1(config-leaf-vrf)# route-map CLI_RP
apic1(config-leaf-vrf-route-map)# ip prefix-list CLI_PFX1 permit 192.168.1.0/24
apic1(config-leaf-vrf-route-map)# match prefix-list CLI_PFX1 [deny]
```

Step 2 Configure the interleak redistribution using the configured route-map.

Example:

The following example configures the redistribution of OSPF routes with the configured route map `CLI_RP`:

```
apic1# conf t
apic1(config)# leaf 101
apic1(config-leaf)# router bgp 65001
apic1(config-leaf-bgp)# vrf member tenant CLI_TEST vrf VRF1
apic1(config-leaf-bgp-vrf)# redistribute ospf route-map CLI_RP
```

Configuring Interleak Redistribution Using the REST API

The following procedure describes how to configure the interleak redistribution using the REST API.

Before you begin

Create the tenant, VRF, and L3Out.

Procedure

Step 1 Configure the route-map for interleak redistribution.

Example:

The following example configures a route map `INTERLEAK_RP` with two contexts (`ROUTES_A` and `ROUTES_ALL`). The first context `ROUTES_A` matches with an IP prefix-list 10.0.0.0/24 le 32 to set a community attribute via set rule `COM_A`. The second context matches with all routes.

```
POST: https://<APIC IP>/api/mo/uni.xml
BODY:
<fvTenant dn="uni/tn-SAMPLE">
    <!-- route map with two contexts (ROUTES_A and ROUTES_ALL)-->
    <rtctrlProfile type="global" name="INTERLEAK_RP">
        <rtctrlCtxP name="ROUTES_A" order="0" action="permit">
            <rtctrlRsCtxPToSubjP tnRtctrlSubjPName="PFX_10-0-0-0_24"/>
            <rtctrlScope>
                <rtctrlRsScopeToAttrP tnRtctrlAttrPName="COM_A"/>
            </rtctrlScope>
        </rtctrlCtxP>
        <rtctrlCtxP name="ROUTES_ALL" order="9" action="permit">
            <rtctrlRsCtxPToSubjP tnRtctrlSubjPName="ALL_PREFIX"/>
        </rtctrlCtxP>
    </rtctrlProfile>

    <!-- match rule with an IP prefix-list -->
    <rtctrlSubjP name="ALL_PREFIX">
        <rtctrlMatchRtDest ip="0.0.0.0/0" aggregate="yes"/>
    </rtctrlSubjP>

    <!-- match rule with an IP prefix-list -->
    <rtctrlSubjP name="PFX_10-0-0-0_24">
        <rtctrlMatchRtDest ip="10.0.0.0/24" aggregate="yes"/>
    </rtctrlSubjP>

    <!-- setu rule for community attribute -->
    <rtctrlAttrP name="COM_A">
        <rtctrlSetComm type="community" setCriteria="append"
community="regular:as2-nn2:100:200"/>
    </rtctrlAttrP>
</fvTenant>
```

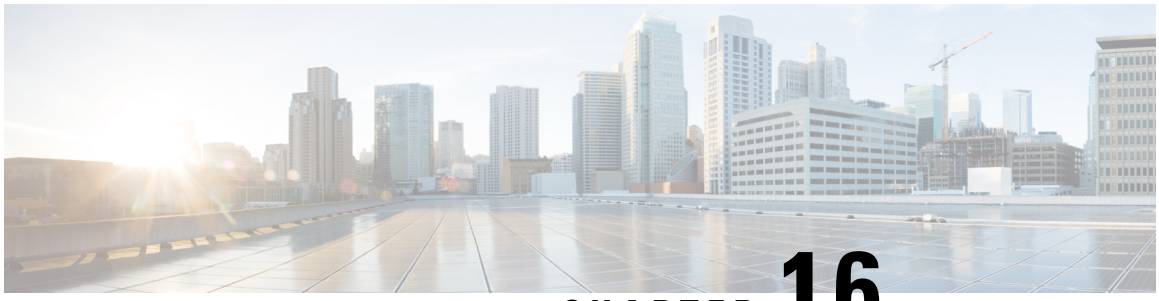
Step 2 Apply the configured route map to an L3Out.

The following example applies the route map from Step 1 to L3Out `l3out1` to customize interleak redistribution of OSPF or EIGRP routes of the given L3Out.

`L3extRsInterleakPol` is applied for dynamic routing protocol (OSPF/EIGRP) routes used by the given L3Out.

Example:

```
POST: https://<APIC IP>/api/mo/uni.xml
BODY:
<fvTenant dn="uni/tn-SAMPLE">
    <l3extOut name="l3out1">
        <!-- interleak redistribution for OSPF/EIGRP routes -->
        <l3extRsInterleakPol tnRtctrlProfileName="INTERLEAK_RP"/>
    </l3extOut>
</fvTenant>
```



CHAPTER 16

Dataplane IP Learning per VRF

This chapter contains the following sections:

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** > *tenant_name* > **Networking** > **VRFs** > *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:

```
apic1# config
```

Step 2 Enter the tenant mode for the specific tenant.

Example:

```
apic1(config)# tenant name
```

Step 3 Enter the VRF context mode.

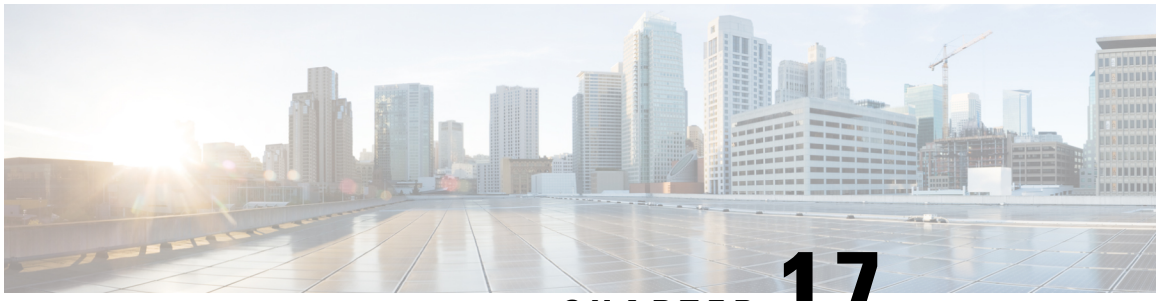
Example:

```
apic1(config-tenant)# vrf context name
```

Step 4 Disable dataplane IP learning for the VRF.

Example:

```
apic1(config-tenant-vrf)# ipdataplanelearning disabled
```



CHAPTER 17

IP Aging

This chapter contains the following sections:

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

Step 1 From the menu bar, click the **System** tab.

Step 2 From the submenu bar, click **System Settings**.

Step 3 In the navigation pane, click **Endpoint Controls**.

Step 4 In the work pane, click **Ip Aging**.
The **IP Aging Policy** appears with the **Administrative State Disabled** button selected.

Step 5 From the **Administrative State**, click one of the following options:

- **Enabled**—Enables IP aging

- **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:

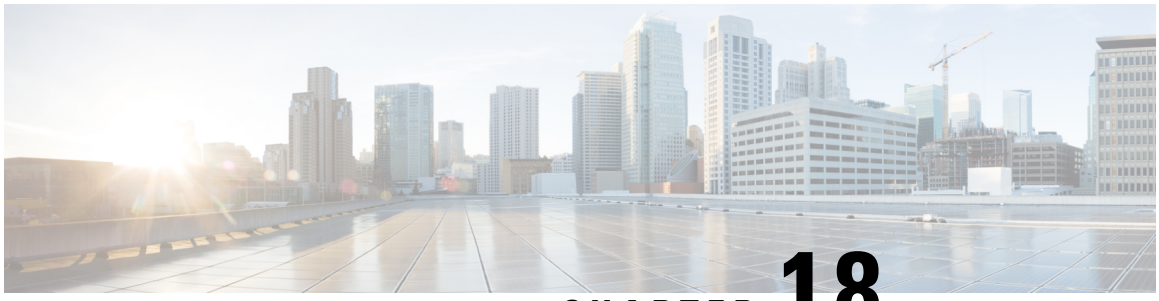
Example:

```
<epIpAgingP adminSt="disabled" descr="" dn="uni/infra/ipAgingP-default" name="default"
ownerKey="" ownerTag=""/>
```

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:

```
<fvEpRetPol bounceAgeIntvl="630" bounceTrig="protocol"
holdIntvl="350" lcOwn="local" localEpAgeIntvl="900" moveFreq="256"
name="EndpointPol1" remoteEpAgeIntvl="350"/>
```



CHAPTER 18

IPv6 Neighbor Discovery

This chapter contains the following sections:

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:

```
<fvTenant 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">
    </fvCtx>
   <fvBD arpFlood="no" 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"/>
        <fvSubnet ctrl="nd" descr="" ip="34::1/64" name="" preferred="no" scope="private">
```

```
            <fvRsNdPfxPol tnNdPfxPolName="NDPfxPol001"/>
        </fvSubnet>
        <fvSubnet ctrl="nd" descr="" ip="33::1/64" name="" preferred="no" scope="private">
            <fvRsNdPfxPol tnNdPfxPolName="NDPfxPol002"/>
        </fvSubnet>
    </fvBD>
   <ndPfxPol ctrl="auto-cfg,on-link" descr="" lifetime="1000" name="NDPfxPol001" ownerKey=""
 ownerTag="" prefLifetime="1000"/>
    <ndPfxPol ctrl="auto-cfg,on-link" descr="" lifetime="4294967295" name="NDPfxPol002"
ownerKey="" ownerTag="" prefLifetime="4294967295"/>
</fvTenant>
```

Note 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
```

```
apic1(config-tenant-interface)# ipv6 address 34::1/64
apic1(config-tenant-interface)# ipv6 address 33::1/64
apic1(config-tenant-interface)# ipv6 nd prefix 34::1/64 1000 1000
apic1(config-tenant-interface)# ipv6 nd prefix 33::1/64 4294967295 4294967295
```

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.

Note 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:

```
<fvTenant 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"/>
  <fvCtx name="pvn1" pcEnfPref="enforced">
                  </fvCtx>
  <l3extOut enforceRtctrl="export" name="l3extOut001">
    <l3extRsEctx tnFvCtxName="pvn1"/>
    <l3extLNodeP name="lnodeP001">
      <l3extRsNodeL3OutAtt rtrId="11.11.205.1" rtrIdLoopBack="yes"
tDn="topology/pod-2/node-2011"/>
      <l3extLIfP name="lifP001">
        <l3extRsPathL3OutAtt addr="2001:20:21:22::2/64" ifInstT="l3-port" llAddr="::"
```

```
mac="00:22:BD:F8:19:FF" mode="regular" mtu="inherit"
tDn="topology/pod-2/paths-2011/pathep-[eth1/1]">
          <ndPfxP>
            <ndRsPfxPToNdPfxPol tnNdPfxPolName="NDPfxPol001"/>
          </ndPfxP>
        </l3extRsPathL3OutAtt>
        <l3extRsNdIfPol tnNdIfPolName="NDPol001"/>
      </l3extLIfP>
    </l3extLNodeP>
    <l3extInstP name="instp"/>
  </l3extOut>
  <ndPfxPol ctrl="auto-cfg,on-link" descr="" lifetime="1000" name="NDPfxPol001" ownerKey=""
 ownerTag="" prefLifetime="1000"/>
</fvTenant>
```

Note 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 >
          <ndRsPfxPToNdPfxPol tnNdPfxPolName="NDPfxPol001"/>
        </ndPfxP>
      </l3extMember>
      <l3extMember addr="2001:20:25:1::2/64" descr="" llAddr="::" name=""
nameAlias="" side="B">
        <ndPfxP >
          <ndRsPfxPToNdPfxPol tnNdPfxPolName="NDPfxPol001"/>
        </ndPfxP>
      </l3extMember>
    </l3extRsPathL3OutAtt>
    <l3extRsNdIfPol tnNdIfPolName="NDPol001"/>    </l3extLIfP>
   </l3extLNodeP>
```

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.

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `apic1# configure`	Enters configuration mode.

Step 2	**tenant** *tenant_name*
	Example:
	`apic1(config)# tenant ExampleCorp`
	`apic1(config-tenant)#`
		Creates a tenant and enters the tenant mode.

Step 3	**template ipv6 nd policy** *policy_name* **Example:** `apic1(config-tenant)# template ipv6 nd policy NDPol001`	Creates an IPv6 ND policy.

Step 4	**ipv6 nd mtu** *mtu value*
	Example:
	`apic1(config-tenant-template-ipv6-nd)# ipv6 nd mtu 1500`
	`apic1(config-tenant-template-ipv6)# exit`
	`apic1(config-tenant-template)# exit`
	`apic1(config-tenant)#`
		Assigns an MTU value to the IPv6 ND policy.

Step 5	**vrf context** *VRF_name*
	Example:
	`apic1(config-tenant)# vrf context pvn1`
	`apic1(config-tenant-vrf)# exit`
		Creates a VRF.

Step 6	**l3out** *VRF_name* **Example:** `apic1(config-tenant)# l3out l3extOut001`	Creates a Layer 3 Out.
Step 7	**vrf member** *VRF_name* **Example:** `apic1(config-tenant-l3out)# vrf member pvn1`	Associates the VRF with the Layer 3 Out.

	Command or Action	Purpose
	`apic1(config-tenant-l3out)# exit`	

Step 8	**external-l3 epg instp l3out** *l3extOut001*
	Example:
	`apic1(config-tenant)# external-l3 epg instp l3out l3extOut001`
	`apic1(config-tenant-l3ext-epg)# vrf member pvn1`
	`apic1(config-tenant-l3ext-epg)# exit`
		Assigns the Layer 3 Out and the VRF to a Layer 3 interface.

Step 9	**leaf** *2011* **Example:** `apic1(config)# leaf 2011`	Enters the leaf switch mode.

Step 10	**vrf context tenant** *ExampleCorp* **vrf** *pvn1* **l3out** *l3extOut001*
	Example:
	`apic1(config-leaf)# vrf context tenant ExampleCorp vrf pvn1 l3out l3extOut001`
	`apic1(config-leaf-vrf)# exit`
		Associates the VRF to the leaf switch.

Step 11	**int** *eth 1/1*
	Example:
	`apic1(config-leaf)# int eth 1/1`
	`apic1(config-leaf-if)#`
		Enters the interface mode.

Step 12	**vrf member tenant** *ExampleCorp* **vrf** *pvn1* **l3out** *l3extOut001* **Example:** `apic1(config-leaf-if)# vrf member tenant ExampleCorp vrf pvn1 l3out l3extOut001`	Specifies the associated Tenant, VRF, Layer 3 Out in the interface.
Step 13	**ipv6 address** *2001:20:21:22::2/64* **preferred** **Example:** `apic1(config-leaf-if)# ipv6 address 2001:20:21:22::2/64 preferred`	Specifies the primary or preferred IPv6 address.

	Command or Action	Purpose
Step 14	**ipv6 nd prefix** *2001:20:21:22::2/64 1000 1000* **Example:** `apic1(config-leaf-if)# ipv6 nd prefix 2001:20:21:22::2/64 1000 1000`	Configures the IPv6 ND prefix policy under the Layer 3 interface.

Step 15	**inherit ipv6 nd** *NDPol001*
	Example:
	`apic1(config-leaf-if)# inherit ipv6 nd NDPol001`
	`apic1(config-leaf-if)# exit`
	`apic1(config-leaf)# exit`
		Configures the ND policy under the Layer 3 interface.

The configuration is complete.

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**.

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:

```
                <l3extRsPathL3OutAtt addr="2001:DB8:A::2/64" autostate="enabled" \
                 childAction="" descr="" encap="vlan-1035" encapScope="local" \
                 ifInstT="ext-svi" ipv6Dad="enabled" llAddr=": :" \
                 mac="00:22:BD:F8:19:DD"  mtu="inherit"  \
                 rn="rspathL3OutAtt-[topology/pod-1/paths-105/pathep-[eth1/1]]"  \
                 status=""  tDn="topology/pod-1/paths-105/pathep-[eth1/1]"  >
                    <l3extIp addr="2001:DB8:A::11/64" childAction="" descr="" \
                     ipv6Dad="disabled"  name="" nameAlias="" \
                     rn="addr-[2001:DB8:A::11/64]" 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"(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]
```

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:

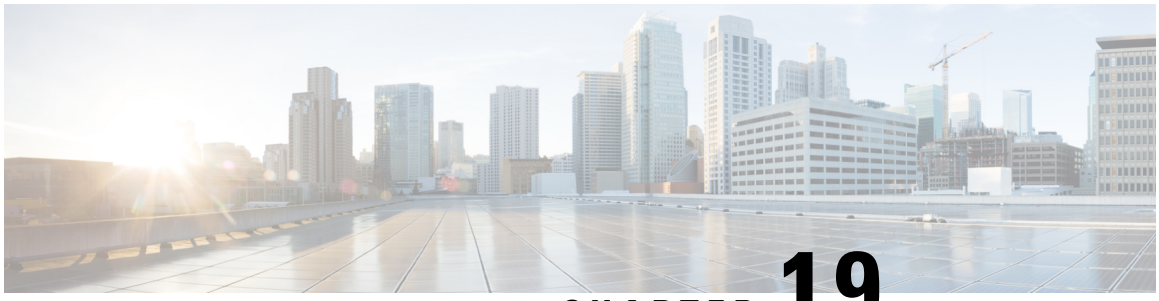
As another example, if you were configuring this setting for the routed sub-interface interface, you would:

- 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 19

Tenant Routed Multicast

This chapter contains the following sections:

Tenant Routed Multicast

Cisco Application Centric Infrastructure (ACI) Tenant Routed Multicast (TRM) enables Layer 3 multicast routing in Cisco ACI tenant VRF instances. TRM supports multicast fowarding between senders and receivers within the same or different subnets. Multicast sources and receivers can be connected to the same or different leaf switches or external to the fabric using L3Out connections.

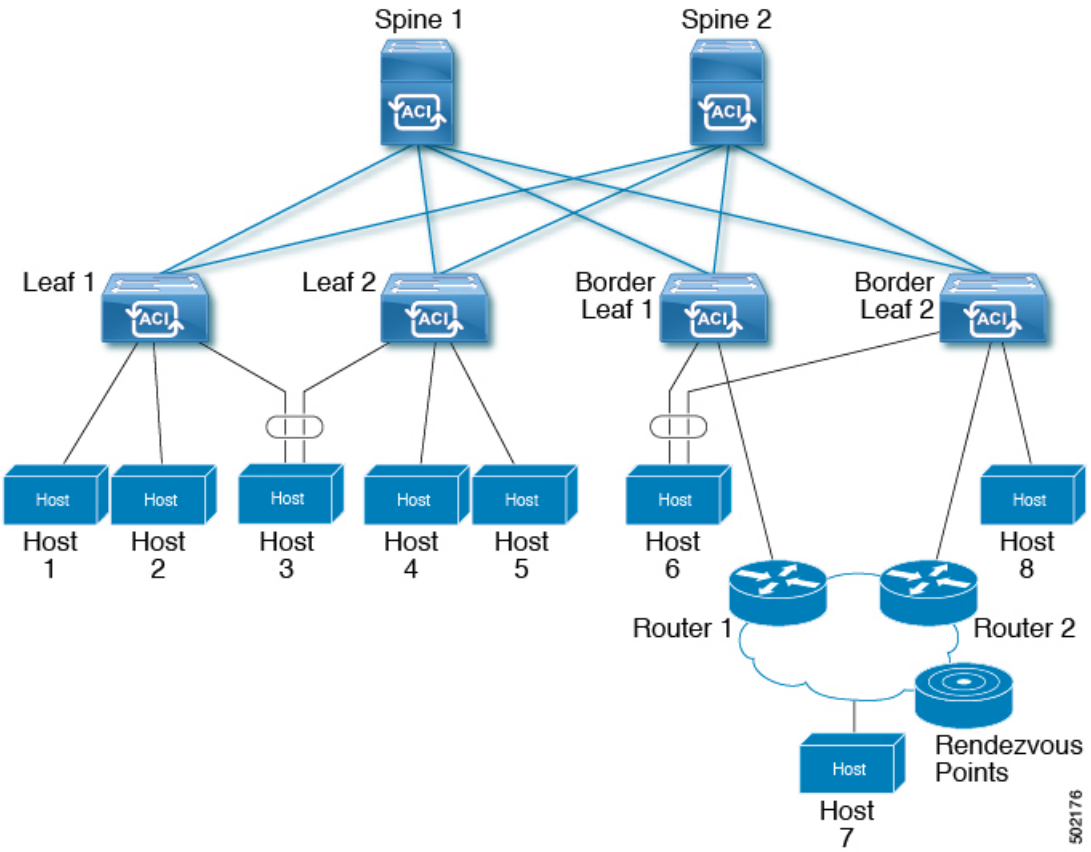
In the Cisco ACI fabric, most unicast and IPv4 multicast routing operate together on the same border leaf switches, with the IPv4 multicast protocol operating over the unicast routing protocols.

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 L3Outs and also with each other.

The following figure shows border leaf switch 1 and border leaf switch 2 connecting to router 1 and router 2 in the IPv4 multicast cloud. Each virtual routing and forwarding (VRF) instance in the fabric that requires IPv4 multicast routing will peer separately with external IPv4 multicast routers.

Figure 22: Overview of Multicast Cloud



About the Fabric Interface

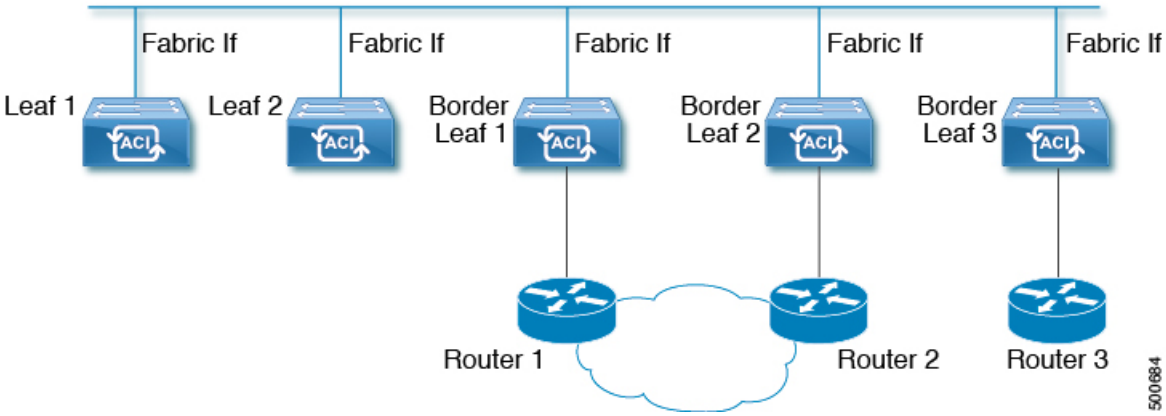
The fabric interface is a virtual interface between software modules and represents the fabric for multicast routing. The interface takes the form of a tunnel interface with the tunnel destination being the VRF GIPo (Group IP outer address)[1]. For example, if a border leaf is the designated forwarder 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).

[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.

Note Each multicast-enabled VRF requires one or more border leaf switches configured with a loopback interface. You must configure a unique IPv4 loopback address on all nodes in a PIM-enabled L3Out. The Router-ID loopback or another unique loopback address can be used.

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 for the VRF. The fabric interface source IP will be set to this loopback as the loopback interface. The following figure shows the fabric for multicast routing.

Figure 23: Fabric for Multicast routing



Enabling IPv4 Tenant Routed Multicast

The process to enable or disable multicast routing in a Cisco ACI fabric occurs at three levels:

- **VRF level**: Enable multicast routing at the VRF level.
- **L3Out level**: Enable PIM for one or more L3Outs configured in the VRF.
- **Bridge domain (BD) level**: Enable PIM for one or more bridge domains where multicast routing is needed.

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

Note 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 7: GIPo Usage

Traffic	Non-MC Routing-enabled BD	MC Routing-enabled BD
Broadcast	BD GIPo	BD GIPo
Unknown Unicast Flood	BD GIPo	BD GIPo
Multicast	BD GIPo	VRF GIPo

Multiple Border Leaf Switches as Designated Forwarder

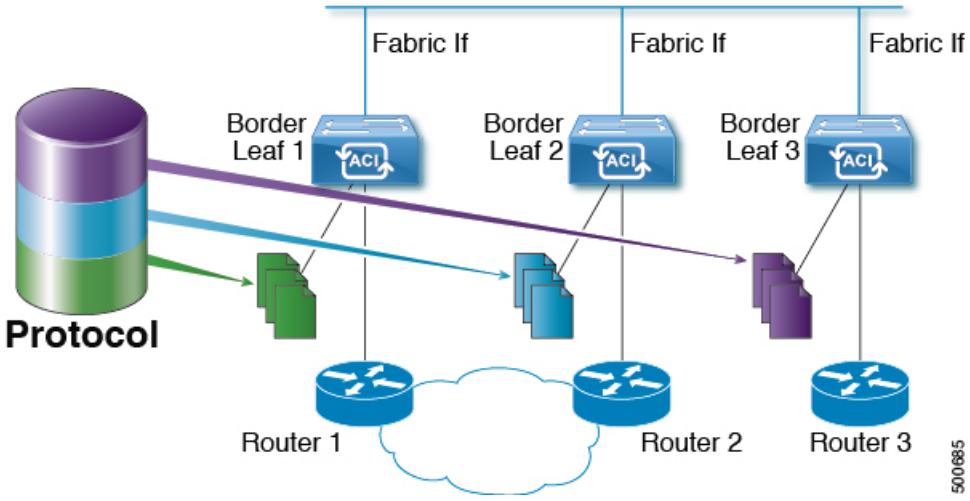
When there are multiple border leaf (BL) switches in the fabric doing multicast routing, only one of the border leafs is selected as the designated forwarder for attracting traffic from the external multicast network and forwarding it to the fabric. This prevents multiple copies of the traffic and it 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.

Figure 24: Model for Multiple Border Leafs as Designated Forwarder



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** repo[2] 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

Note 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. When multiple RPs are configured, each RP must be configured for a unique group range.

PIM enabled border leafs are required for VRFs where multicast routing is enabled. PIM is enabled for a border leaf by enabling PIM at the L3Out level. When PIM is enabled for an L3Out this will enable PIM for all nodes and interfaces configured under that L3Out.

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 a PIM anycast RP loopback interface on all PIM-enabled border leaf switches in the VRF, which is necessary for supporting inter-VRF multicast (see About Inter-VRF Multicast, on page 216). A PIM-enabled L3Out (with loopback interfaces) is required for fabric RP configuration. When configured, external routers can use the fabric RP using static RP configuration. Auto-RP and BSR are not supported with Fabric RP. Fabric RP peering with an external anycast RP member is not supported.

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 224
- Configuring Layer 3 Multicast Using the NX-OS Style CLI, on page 226
- Configuring Layer 3 Multicast Using REST API, on page 228

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.
- Inter-VRF multicast supports both shared services and share L3Out configurations. Sources and receivers can be connected to EPGs or L3Outs in different VRFs.

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.

ACI Multicast Feature List

The following sections provide a list of ACI multicast features with comparisons to similar NX-OS features.

IGMP Features

ACI Feature Name	NX-OS Feature	Description
Allow V3 ASM	ip igmp allow-v3-asm	Allow accepting IGMP version 3 source-specific reports for multicast groups outside of the SSM range. When this feature is enabled, the switch will create an (S,G) mroute entry if it receives an IGMP version 3 report that includes both the group and source even if the group is outside of the configured SSM range. This feature is not required if hosts send (*,G) reports outside of the SSM range, or send (S,G) reports for the SSM range.
Fast Leave	ip igmp immediate-leave	Option that minimizes the leave latency of IGMPv2 group memberships on a given IGMP interface because the device does not send group-specific queries. When immediate leave is enabled, the device removes the group entry from the multicast routing table immediately upon receiving a leave message for the group. The default is disabled. Note: Use this command only when there is one receiver behind the BD/interface for a given group
Report Link Local Groups	ip igmp report-link-local-groups	Enables sending reports for groups in 224.0.0.0/24. Reports are always sent for nonlink local groups. By default, reports are not sent for link local groups.
Group Timeout (sec)	ip igmp group-timeout	Sets the group membership timeout for IGMPv2. Values can range from 3 to 65,535 seconds. The default is 260 seconds.
Query Interval (sec)	ip igmp query-interval	Sets the frequency at which the software sends IGMP host query messages. Values can range from 1 to 18,000 seconds. The default is 125 seconds.
Query Response Interval (sec)	ip igmp query-max-response-time	Sets the response time advertised in IGMP queries. Values can range from 1 to 25 seconds. The default is 10 seconds.
Last Member Count	ip igmp last-member-query-count	Sets the number of times that the software sends an IGMP query in response to a host leave message. Values can range from 1 to 5. The default is 2.
Last Member Response Time (sec)	ip igmp last-member-query-response-time	Sets the query interval waited after sending membership reports before the software deletes the group state. Values can range from 1 to 25 seconds. The default is 1 second.

ACI Feature Name	NX-OS Feature	Description
Startup Query Count	ip igmp startup-query-count	Sets the query count used when the software starts up. Values can range from 1 to 10. The default is 2.
Querier Timeout	ip igmp querier-timeout	Sets the query timeout that the software uses when deciding to take over as the querier. Values can range from 1 to 65,535 seconds. The default is 255 seconds.
Robustness Variable	ip igmp robustness-variable	Sets the robustness variable. You can use a larger value for a lossy network. Values can range from 1 to 7. The default is 2.
Version	ip igmp version <2-3>	IGMP version that is enabled on the bridge domain or interface. The IGMP version can be 2 or 3. The default is 2.
Report Policy Route Map*	ip igmp report-policy <route-map>	Access policy for IGMP reports that is based on a route-map policy. IGMP group reports will only be selected for groups allowed by the route-map
Static Report Route Map*	ip igmp static-oif	Statically binds a multicast group to the outgoing interface, which is handled by the switch hardware. If you specify only the group address, the (*, G) state is created. If you specify the source address, the (S, G) state is created. You can specify a route-map policy name that lists the group prefixes, group ranges, and source prefixes. Note A source tree is built for the (S, G) state only if you enable IGMPv3.
Maximum Multicast Entries	ip igmp state-limit	Limit the mroute states for the BD or interface that are created by IGMP reports. Default is disabled, no limit enforced. Valid range is 1-4294967295.
Reserved Multicast Entries	ip igmp state-limit <limit> reserved <route-map>	Specifies to use the route-map policy name for the reserve policy and set the maximum number of (*, G) and (S, G) entries allowed on the interface.
State Limit Route Map*	ip igmp state-limit <limit> reserved <route-map>	Used with Reserved Multicast Entries feature

IGMP Snooping Features

ACI Feature Name	NX-OS Feature	Description
IGMP snooping admin state	[no] ipigmp snooping	Enables/disables the IGMP snooping feature. Cannot be disabled for PIM enabled bridge domains
Fast Leave	ip igmp snooping fast-leave	Option that minimizes the leave latency of IGMPv2 group memberships on a given IGMP interface because the device does not send group-specific queries. When immediate leave is enabled, the device removes the group entry from the multicast routing table immediately upon receiving a leave message for the group. The default is disabled. Note: Use this command only when there is one receiver behind the BD/interface for a given group

ACI Feature Name	NX-OS Feature	Description
Enable Querier	ip igmp snooping querier <ip address>	Enables the IP IGMP snooping querier feature on the Bridge Domain. Used along with the BD subnet Querier IP setting to configure an IGMP snooping querier for bridge domains. Note: Should not be used with PIM enabled bridge domains. The IGMP querier function is automatically enabled for when PIM is enabled on the bridge domain.
Query Interval	ip igmp snooping query-interval	Sets the frequency at which the software sends IGMP host query messages. Values can range from 1 to 18,000 seconds. The default is 125 seconds.
Query Response Interval	ip igmp snooping query-max-response-time	Sets the response time advertised in IGMP queries. Values can range from 1 to 25 seconds. The default is 10 seconds.
Last Member Query Interval	ip igmp snooping last-member-query-interval	Sets the query interval waited after sending membership reports before the software deletes the group state. Values can range from 1 to 25 seconds. The default is 1 second.
Start Query Count	ip igmp snooping startup-query-count	Configures snooping for a number of queries sent at startup when you do not enable PIM because multicast traffic does not need to be routed. Values can range from 1 to 10. The default is 2.
Start Query Interval (sec)	ip igmp snooping startup-query-interval	Configures a snooping query interval at startup when you do not enable PIM because multicast traffic does not need to be routed. Values can range from 1 to 18,000 seconds. The default is 31 seconds

MLD Snooping Features

ACI Feature Name	NX-OS Feature	Description
MLD snooping admin state	ipv6 mld snooping	IPv6 MLD snooping feature. Default is disabled
Fast Leave	ipv6 mld snooping fast-leave	Allows you to turn on or off the fast-leave feature on a per bridge domain basis. This applies to MLDv2 hosts and is used on ports that are known to have only one host doing MLD behind that port. This command is disabled by default.
Enable Querier	ipv6 mld snooping querier	Enables or disables IPv6 MLD snooping querier processing. MLD snooping querier supports the MLD snooping in a bridge domain where PIM and MLD are not configured because the multicast traffic does not need to be routed.
Query Interval	ipv6 mld snooping query-interval	Sets the frequency at which the software sends MLD host query messages. Values can range from 1 to 18,000 seconds. The default is 125 seconds.
Query Response Interval	ipv6 mld snooping query-interval	Sets the response time advertised in MLD queries. Values can range from 1 to 25 seconds. The default is 10 seconds.
Last Member Query Interval	ipv6 mld snooping last-member-query-interval	Sets the query response time after sending membership reports before the software deletes the group state. Values can range from 1 to 25 seconds. The default is 1 second.

PIM Features (Interface Level)

ACI Feature Name	NX-OS Feature	Description
Authentication	ip pim hello-authentication ah-md5	Enables MD5 hash authentication for PIM IPv4 neighbors
Multicast Domain Boundary	ip pim border	Enables the interface to be on the border of a PIM domain so that no bootstrap, candidate-RP, or Auto-RP messages are sent or received on the interface. The default is disabled.
Passive	ip pim passive	If the passive setting is configured on an interface, it will enable the interface for IP multicast. PIM will operate on the interface in passive mode, which means that the leaf will not send PIM messages on the interface, nor will it accept PIM messages from other devices across this interface. The leaf will instead consider that it is the only PIM device on the network and thus act as the DR. IGMP operations are unaffected by this command.
Strict RFC Compliant	ip pim strict-rfc-compliant	When configured, the switch will not process joins from unknown neighbors and will not send PIM joins to unknown neighbors
Designated Router Delay (sec)	ip pimdr-delay	Delays participation in the designated router (DR) election by setting the DR priority that is advertised in PIM hello messages to 0 for a specified period. During this delay, no DR changes occur, and the current switch is given time to learn all of the multicast states on that interface. After the delay period expires, the correct DR priority is sent in the hello packets, which retriggers the DR election. Values are from 1 to 65,535. The default value is 3. Note: This command delays participation in the DR election only upon bootup or following an IP address or interface state change. It is intended for use with multicast-access non-vPC Layer 3 interfaces only.
Designated Router Priority	ip pim dr-priority	Sets the designated router (DR) priority that is advertised in PIM hello messages. Values range from 1 to 4294967295. The default is 1.
Hello Interval (milliseconds)	ip pim hello-interval	Configures the interval at which hello messages are sent in milliseconds. The range is from 1000 to 18724286. The default is 30000.
Join-Prune Interval Policy (seconds)	ip pim jp-interval	Interval for sending PIM join and prune messages in seconds. Valid range is from 60 to 65520. Value must be divisible by 60. The default value is 60.
Interface-level Inbound Join-Prune Filter Policy*	ip pimjp-policy	Enables inbound join-prune messages to be filtered based on a route-map policy where you can specify group, group and source, or group and RP addresses. The default is no filtering of join-prune messages.
Interface-level Outbound Join-Prune Filter Policy*	ip pim jp-policy	Enables outbound join-prune messages to be filtered based on a route-map policy where you can specify group, group and source, or group and RP addresses. The default is no filtering of join-prune messages.
Interface-level Neighbor Filter Policy*	ip pim neighbor-policy	Controls which PIM neighbors to become adjacent to based on route-map policy where you specify the source address/address range of the permitted PIM neighbors

PIM Features (VRF Level)

ACI Feature Name	NX-OS Feature	Description
Static RP	ippimrp-address	Configures a PIM static RP address for a multicast group range. You can specify an optional route-map policy that lists multicast group ranges for the static RP. If no route-map is configured, the static RP will apply to all multicast group ranges excluding any configured SSM group ranges. The mode is ASM.
Fabric RP	n/a	Configures an anycast RP on all multicast enabled border leaf switches in the fabric. Anycast RP is implemented using PIM anycast RP. You can specify an optional route-map policy that lists multicast group ranges for the static RP.
Auto-RP Forward Auto-RP Updates	ip pim auto-rp forward	Enables the forwarding of Auto-RP messages. The default is disabled.
Auto-RP Listen to Auto-RP Updates	ip pim auto-rp listen	Enables the listening for Auto-RP messages. The default is disabled.
Auto-RP MA Filter *	ip pim auto-rp mapping-agent-policy	Enables Auto-RP discover messages to be filtered by the border leaf based on a route-map policy where you can specify mapping agent source addresses. This feature is used when the border leaf is configured to listen for Auto-RP messages. The default is no filtering of Auto-RP messages.
BSR Forward BSR Updates	ippimbsr forward	Enables forwarding of BSR messages. The default is disabled, which means that the leaf does not forward BSR messages.
BSR Listen to BRS Updates	ip pim bsr listen	Enables listening for BSR messages. The default is disabled, which means that the leaf does not listen for BSR messages.
BSR Filter	ip pim bsr bsr-policy	Enables BSR messages to be filtered by the border leaf based on a route-map policy where you can specify BSR source. This command can be used when the border leaf is configured to listen to BSR messages. The default is no filtering of BSR messages.
ASM Source, Group Expiry Timer Policy *	ip pim sg-expiry-timer <timer> sg-list	Applies a route map to the ASM Source, Group Expiry Timer to specify a group/range of groups for the adjusted expiry timer.
ASM Source, Group Expiry Timer Expiry (sec)	ip pim sg-expiry-timer	To adjust the (S,G) expiry timer interval for Protocol Independent Multicast sparse mode (PIM-SM) (S,G) multicast routes. This command creates persistency of the SPT (source based tree) over the default 180 seconds for intermittent sources. Range is from 180 to 604801 seconds.
Register Traffic Policy: Max Rate	ip pim register-rate-limit	Configures the rate limit in packets per second. The range is from 1 to 65,535. The default is no limit.
Register Traffic Policy: Source IP	ip pim register-source	Used to configure a source IP address of register messages. This feature can be used when the source address of register messages is routed in the network where the RP can send messages. This may happen if the bridge domain where the source is connected is not configured to advertise its subnet outside of the fabric.

ACI Feature Name	NX-OS Feature	Description
SSM Group Range Policy*	ippimssm route-map	Can be used to specify different SSM group ranges other than the default range 232.0.0.0/8. This command is not required if you want to only use the default group range. You can configure a maximum of four ranges for SSM multicast including the default range.
Fast Convergence	n/a	When fast convergence mode is enabled, every border leaf in the fabric will send PIM joins towards the root (RP for (*,G) and source (S,G)) in the external network. This allows all PIM enabled BLs in the fabric to receive the multicast traffic from external sources but only one BL will forward traffic onto the fabric. The BL that forwards the multicast traffic onto the fabric is the designated forwarder. The stripe winner BL decides on the DF. The advantage of the fast-convergence mode is that when there is a changed of the stripe winner due to a BL failure there is no latency incurred in the external network by having the new BL send joins to create multicast state. Note: Fast convergence mode can be disabled in deployments where the cost of additional bandwidth outweighs the convergence time saving.
Strict RFC Compliant	ip pim strict-rfc-compliant	When configured, the switch will not process joins from unknown neighbors and will not send PIM joins to unknown neighbors
MTU Port	ippimmtu	Enables bigger frame sizes for the PIM control plane traffic and improves the convergence. Range is from 1500 to 9216 bytes
Resource Policy Maximum Limit	ip pim state-limit	Sets the maximum (*,G)/(S,G) entries allowed per VRF. Range is from 1 to 4294967295
Resource Policy Reserved Route Map*	ip pim state-limit <limit> reserved <route-map>	Configures a route-map policy matching multicast groups or groups and sources to be applied to the Resource Policy Maximum Limit reserved entries.
Resource Policy Reserved Multicast Entries	ip pim state-limit <limit> reserved <route-map> <limit>	Maximum reserved (*, G) and (S, G) entries allowed in this VRF. Must be less than or equal to the maximum states allowed. Used with the Resource Policy Reserved Route Map policy

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.
- Any Source Multicast (ASM) and Source-Specific Multicast (SSM) are supported.
- You can configure a maximum of four ranges for SSM multicast in the route map per VRF.
- 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.

- If the multicast source is connected to Leaf-A as an orphan port and you have an L3Out on Leaf-B, and Leaf-A and Leaf-B are in a vPC pair, the EPG encapsulation VLAN tied to the multicast source will need to be deployed on Leaf-B.
- 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 preceeding 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.
- You cannot use a filter with inter-VRF multicast communication.

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:

```
apic1(config-tenant-vrf)# ip igmp
apic1(config-tenant-vrf)# exit
apic1(config-tenant)# interface bridge-domain tenant1_bd
apic1(config-tenant-interface)# ip multicast
apic1(config-tenant-interface)# ip igmp allow-v3-asm
apic1(config-tenant-interface)# ip igmp fast-leave
apic1(config-tenant-interface)# ip igmp inherit interface-policy igmp_intpol1
apic1(config-tenant-interface)# exit
```

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
apic1(config)#
apic1(config)# leaf 101
apic1(config-leaf)# interface ethernet 1/125
apic1(config-leaf-if) ip pim inherit interface-policy pim_intpol1
```

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_intpol1
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-tenant)# 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:

```
<fvTenant dn="uni/tn-PIM_Tenant" name="PIM_Tenant">
   <fvCtx knwMcastAct="permit" name="ctx1">
       <pimCtxP mtu="1500">
       </pimCtxP>
   </fvCtx>
  </fvTenant>
```

Step 2 Configure L3 Out and enable multicast (PIM, IGMP) on the L3 Out.

Example:

```
<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_pol1"/>
           </pimIfP>
           <l3extRsPathL3OutAtt addr="131.1.1.1/24" ifInstT="l3-port" mode="regular"
mtu="1500" tDn="topology/pod-1/paths-101/pathep-[eth1/46]"/>
         </l3extLIfP>
       </l3extLNodeP>
       <l3extRsL3DomAtt tDn="uni/l3dom-l3outDom"/>
       <l3extInstP name="l3out-PIM_Tenant-CTX1-1topo" >
       </l3extInstP>
       <pimExtP enabledAf="ipv4-mcast" name="pim"/>
     </l3extOut>
```

Step 3 Configure a BD under the tenant and enable multicast and IGMP on the BD.

Example:

```
<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/>
       <fvSubnet ctrl="" ip="41.1.1.254/24" preferred="no" scope="private" virtual="no"/>
     </fvBD>
 </fvTenant>
```

Step 4 Configure an IGMP policy and assign it to the BD.

Example:

```
<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"
startQueryIntvl="125" ver="v2">
     </igmpIfPol>
      <fvBD arpFlood="yes" mcastAllow="yes" name="bd2">
      <igmpIfP>
         <igmpRsIfPol tDn="uni/tn-PIM_Tenant/igmpIfPol-igmp_pol"/>
       </igmpIfP>
      </fvBD>
 </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:

```
<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:

```
<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.

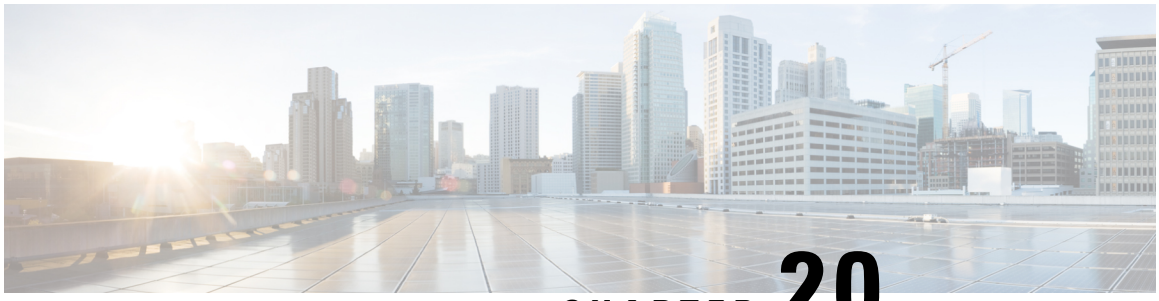
Example:

```
<fvTenant dn="uni/tn-PIM_Tenant" name="PIM_Tenant">
  <pimIfPol authKey="" authT="none" ctrl="" drDelay="60" drPrio="1" helloItvl="30000"
itvl="60" name="pim_pol1"/>
  <l3extOut enforceRtctrl="export" name="l3out-pim_l3out1" targetDscp="unspecified">
       <l3extRsEctx tnFvCtxName="ctx1"/>
       <l3extLNodeP  name="bLeaf-CTX1-101">
         <l3extRsNodeL3OutAtt rtrId="200.0.0.1" rtrIdLoopBack="yes"
tDn="topology/pod-1/node-101"/>
         <l3extLIfP name="if-SIRI_VPC_src_recv-CTX1" tag="yellow-green">
           <pimIfP>
             <pimRsIfPol tDn="uni/tn-tn-PIM_Tenant/pimifpol-pim_pol1"/>
           </pimIfP>
         </l3extLIfP>
       </l3extLNodeP>
     </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" >
                    <rtdmcRsFilterToRtMapPol tDn="uni/tn-t0/rtmap-intervrf" />
                   </pimInterVRFEntryPol>
               </pimInterVRFPol>
            </pimCtxP>
        </fvCtx>
</fvTenant>
```



CHAPTER 20

IP SLAs

This chapter contains the following sections:

About ACI IP SLAs

Many companies conduct most of their business online and any loss of service can affect their profitability. Internet service providers (ISPs) and even internal IT departments now offer a defined level of service, a service level agreement (SLA), to provide their customers with a degree of predictability.

IP SLA tracking is a common requirement in networks. IP SLA tracking allows a network administrator to collect information about network performance in real time. With the Cisco ACI IP SLA, you can track an IP address using ICMP and TCP probes. Tracking configurations can influence route tables, allowing for routes to be removed when tracking results come in negative and returning the route to the table when the results become positive again.

ACI IP SLAs are available for the following:

- Static routes:
  - New in ACI 4.1
  - Automatically remove or add a static route from/to a route table
  - Track the route using ICMP and TCP probes
- Policy-based redirect (PBR) tracking:
  - Available since ACI 3.1
  - Automatically remove or add a next -hop
  - Track the next-hop IP address using ICMP and TCP probes, or a combination using L2Ping
  - Redirect traffic to the PBR node based on the reachability of the next-hop

For more information about PBR tracking, see Configuring Policy-Based Redirect in the *Cisco APIC Layer 4 to Layer 7 Services Deployment Guide*.

Note For either feature, you can perform a network action based on the results of the probes, including configuration, using APIs, or running scripts.

ACI IP SLA Supported Topologies

The following ACI fabric topologies support IP SLA:

- **Single Fabric:** IP SLA tracking is supported for IP address reachable through both L3out and EPG/BD
- **Multi-Pod**
  - You can define a single object tracking policy across different Pods.
  - A workload can move from one Pod to another. The IP SLA policy continues to check accessibility information and detects if an endpoint has moved.
  - If an endpoint moves to another Pod, IP SLA tracking is moved to the other Pod as well, so that tracking information is not passed through the IP network.
- **Remote Leaf**
  - You can define single object tracking policies across ACI main data center and the remote leaf switch.
  - IP SLA probes on remote leaf switches track IP addresses locally without using the IP network.
  - A workload can move from one local leaf to a remote leaf. The IP SLA policy continues to check accessibility information and detects if an endpoint has moved.
  - IP SLA policies move to the remote leaf switches or ACI main data center, based on the endpoint location, for local tracking, so that tracking traffic is not passed through the IP network.

Cisco ACI IP SLA Operation

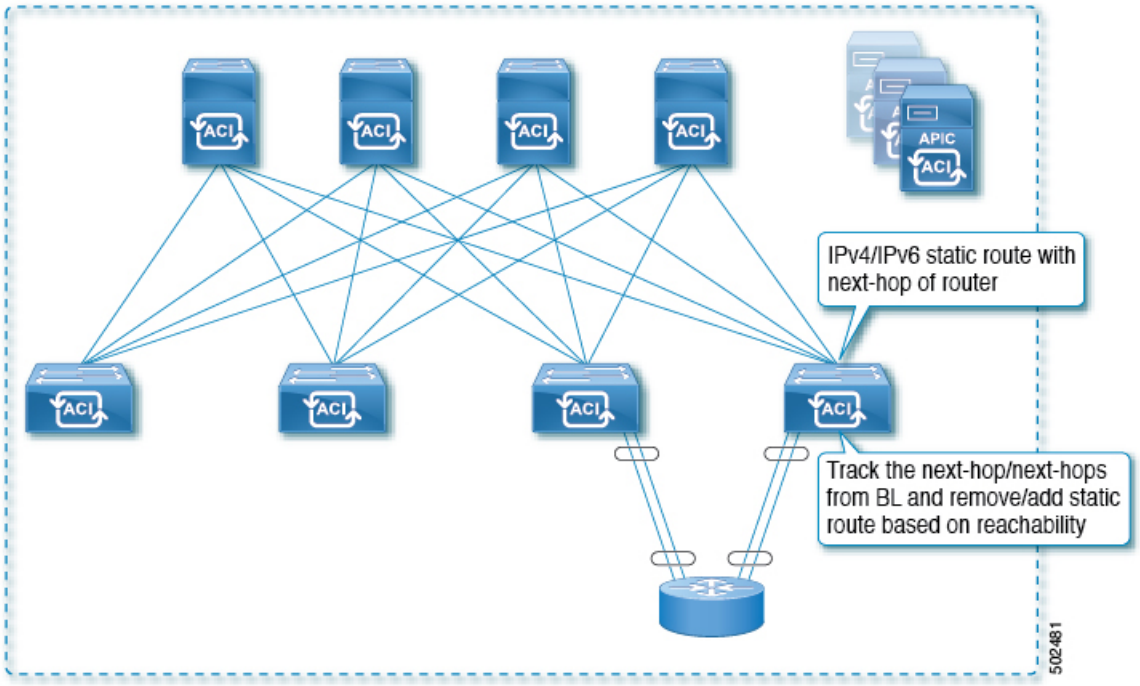
A Cisco ACI IP SLA provides monitoring capabilities on the ACI fabric allowing the SLA probing to occur across the data center network and out to the external network. This is accomplished by configuring an IP SLA monitoring policy which defines the probe type used during monitoring. The monitoring policy is then associated with monitoring probe profiles known as "track members". Once configured, track members define an endpoint or next-hop by IP address, the associated monitoring policy, and the scope (bridge domain or L3Out). One or more track members can be assigned to a "track list". Track lists configure thresholds that, if exceeded, determine if a track list is available (up) or unavailable (down).

The following four examples show the supported use cases for ACI IP SLAs in static routes.

Example 1: Static Route Availability by Tracking the Next-Hop

The following figure shows the network topology and the operation for tracking the static route availability of a router.

Figure 25: Static Route Availability by Tracking the Next-Hop



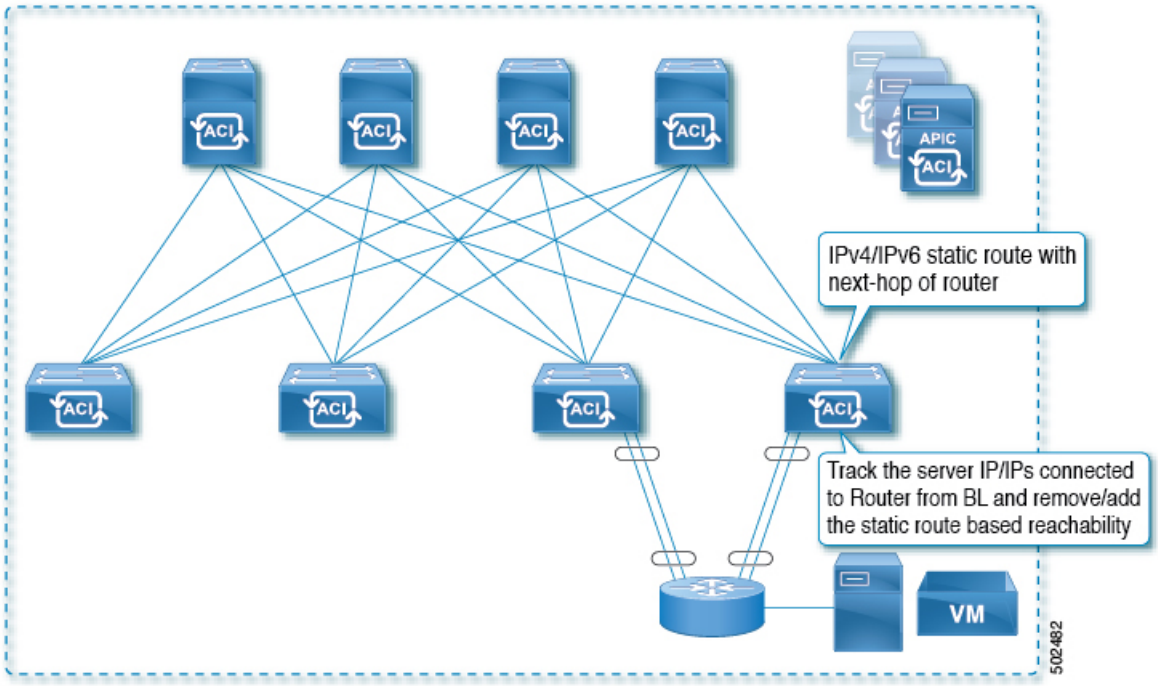
For this use case:

- The next-hop can be direct or indirect meaning that the next-hop can be a loopback IP address of the router.
- The next-hop can be accessed through a physical interface, sub-interface, port channel (PC), PC sub-interface, vPC, or switch virtual interface (SVI).
- The static route is configured under the L3out external network and can be removed or added from/to the route table based on the accessibility of the next-hop .

Example 2: Static Route Availability by Tracking an IP Address Through L3Out

The following figure shows the network topology and the operation for tracking the static route availability of a server through an L3Out external route.

Figure 26: Static Route Availability by Tracking an IP Address Through L3Out



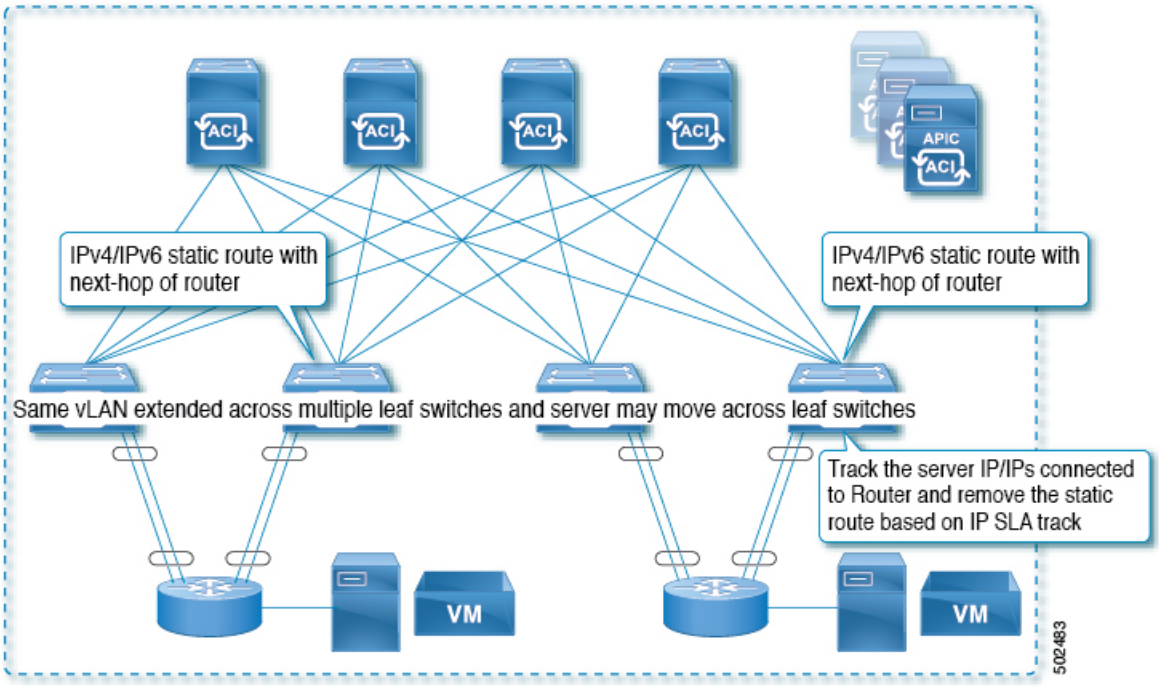
For this use case:

- Track the server IP address connected to the router from the ACI fabric (border leaf) and remove or add the static route based on accessibility of the server.
- The L3Out can be through a port channel (PC), PC sub-interface, vPC, switch virtual interface (SVI), L3 interface, or an L3 sub-interface.
- The static route is configured under L3Out and is removed or added based on the accessibility of the IP address.

Example 3: Static Route Removal by Tracking an IP Address Through L3Out

The following figure shows the network topology and the operation for tracking the static route availability of a server through an L3Out external route. The route is removed if it is not accessible through the L3Out/VRF.

Figure 27: Static Route Removal by Tracking an IP Address Through L3Out



For this use case:

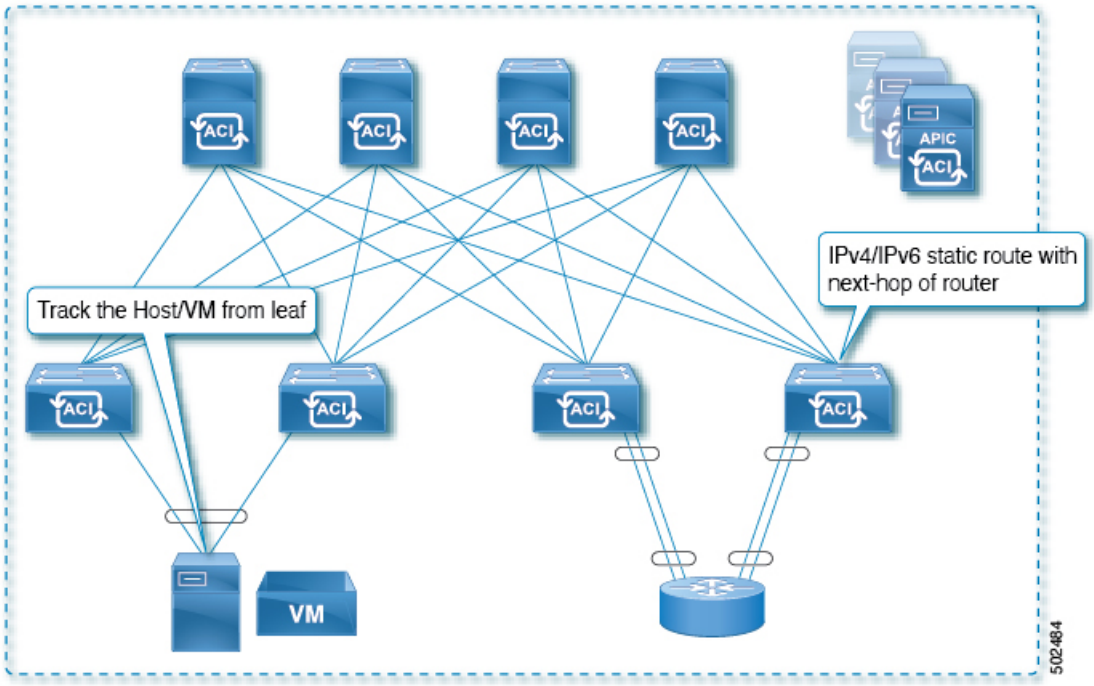
- The L3Out is configured over VLAN/SVI, and that SVI is extended across multiple leaves.
- The server IP address that is accessible through the L3Out can move across leaves.
- Track the server IP address(es) and if they are not accessible through the L3Out/VRF, then remove the static route from the route table.
- The static route is added back to the route table when server is accessible again.

Example 4: Static Route Removal by Tracking an IP Address in the ACI Fabric

Although, as shown in the previous examples, the probe IP of IP SLA for routes is typically the next-hop of the route or an external IP address that should be reachable via the route, you can also use an endpoint IP address in the ACI BD as the probe IP, even if the endpoint does not reside behind the route targeted by the IP SLA. This might be helpful when the static route is to be used solely by certain specific endpoints inside ACI. If such endpoints don’t exist, there is no use for the route.

The following figure shows the network topology and the operation for tracking an IP address in the ACI fabric.

Figure 28: Static Route Availability by Tracking an IP Address in the ACI Fabric



For this use case:

- Track the IP reachability of the endpoints that are connected through the EPG/BD.
- Based on the accessibility of the endpoints, the static route will be removed or added in the L3Out.
- Even if the endpoint moves from one location to another within the fabric, as long as there is the IP reachability to the endpoint from the same BD, IP SLA monitoring considers it accessible and there will be no impact to the validity of the static route.

IP SLA Monitoring Policy

IP Service Level Agreements (SLAs) use active traffic monitoring to generate traffic in a continuous, reliable, and predictable manner, and analyze it to measure the network performance. Measurement statistics that are provided by the IP SLA monitoring policy operations can be used for troubleshooting, problem analysis, and designing network topologies.

With Cisco ACI, the IP SLA monitoring policy is associated with:

- Service Redirect Policies: All the destinations under a service redirect policy are monitored based on the configurations and parameters that are set in the monitoring policy.
- Static Routes: Adding an IP SLA monitoring policy to a track list or track member and associated it with a static route provides the mechanism for monitoring the availability of the next hop segments of the route.

An IP SLA monitoring policy identifies the probe frequency and the type of probe.

ACI IP SLA Monitoring Operation Probe Types

Using ACI IP SLAs, you can monitor the performance between any area in the network: core, distribution, and edge. Monitoring can be done anytime, anywhere, without deploying a physical probe. ACI IP SLAs use generated traffic to measure network performance between two networking devices such as switches. The types of IP SLA operations include:

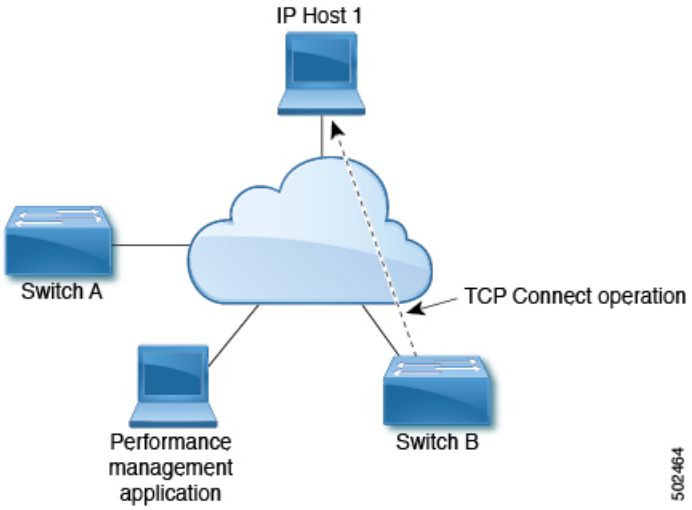
- ICMP: Echo Probes
- TCP: Connect Probes

TCP Connect Operation

The IP SLAs TCP connect operation measures the response time that is taken to perform a TCP probe between a Cisco switch and an IP device. TCP is a transport layer (Layer 4) Internet Protocol that provides reliable full-duplex data transmission. The destination device can be any device using IP.

In the following figure, Switch B is configured as the source IP SLA device based on the configured static route. A TCP connect operation is configured in the IP SLA monitoring policy (associated with the static route) with the destination device as IP Host 1.

Figure 29: TCP Connection Operation Example



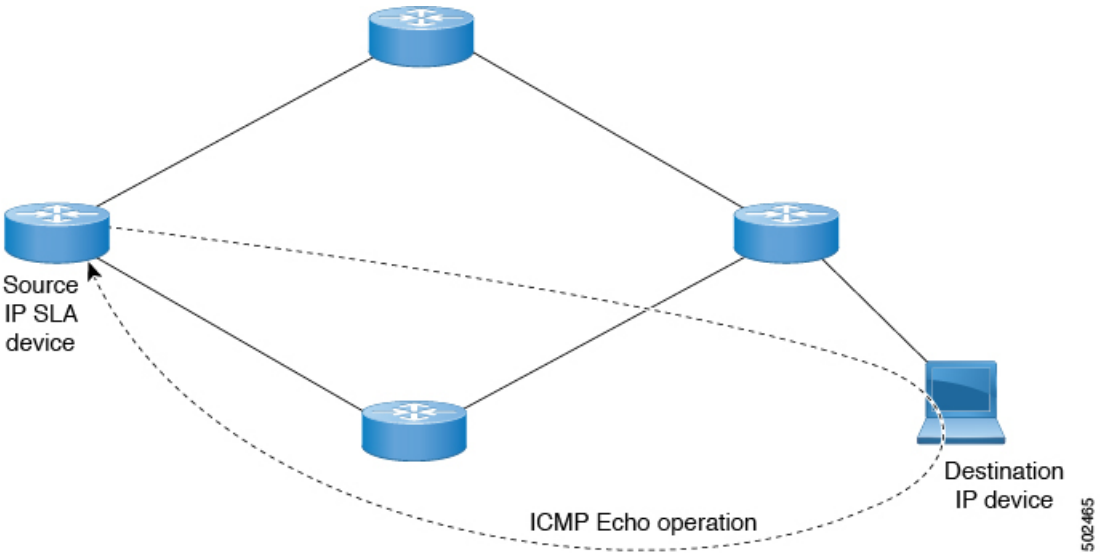
The connection response time is computed by measuring the time that is taken between sending a TCP request message from Switch B to IP Host 1 and receiving a reply from IP Host 1.

ICMP Echo Operation

The Internet Control Message Protocol (ICMP) Echo operation measures the end-to-end response time between two devices that use IPv4 or IPv6. The response time is computed by measuring the time that is taken between sending an ICMP Echo request message to the destination and receiving a reply. An ICMP Echo is useful for troubleshooting network connectivity issues. The results of the ICMP Echo operation can be displayed and analyzed to determine how the network IP connections are performing.

In the following figure, the ICMP Echo operation uses a ping-based probe to measure the response time between the source IP SLAs device and the destination IP device. Many customers use IP SLA ICMP-based operations, in-house ping testing, or ping-based dedicated probes for response time measurements.

Figure 30: ICMP Echo Operation Example



The IP SLA ICMP Echo operation conforms to the same IETF specifications for ICMP ping testing and the two methods result in the same response times.

IP SLA Track Members

An IP SLA track member identifies the:

- IP address to be tracked
- IP SLA monitoring policy (probe frequency and type)
- Scope (bridge domain or L3Out)

IP SLA Track Lists

An IP SLA track list aggregates one or more IP SLA track members representing a network segment to be monitored. The track list determines what percentage or weight of track members must be up or down for the static route to be considered available or unavailable. If the track list is up, based on the threshold percentage or weight, then the static route remains in routing table. If the track list is down, then the static route is removed from the routing table until the track list recovers.

The following is an example of configuring four track members in a track list using the threshold percentage option.

Threshold configuration:

- Set the Percentage Up parameter to 100 (percent)
- Set the Percentage Down parameter to 50 (percent)

In this track list, each of the four track members is assigned 25%. For the track list to become unreachable (down), two of the four track members must be unreachable (50%). For the track list to return to reachable (up), all four track members must be reachable (100%).

Note When a track list is associated with a static route and the track list becomes unreachable (down), the static route is removed from the routing table until the track list becomes reachable again.

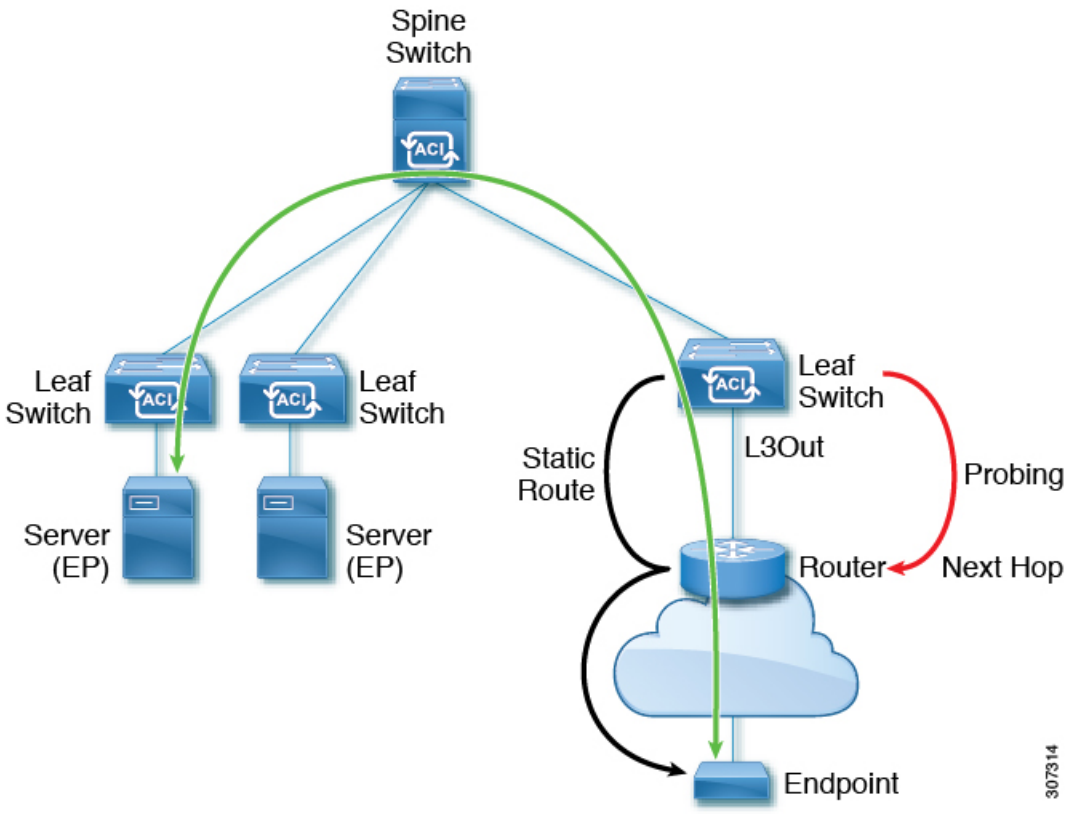
Example IP SLA Configuration Component Associations

ACI IP SLAs rely on track members and track lists to identify the types of probes to send and where to send them. Planning the configuration will help make the task easy and fast. This section uses an example to explain how to set up the IP SLA.

Cisco ACI IP SLA L3Out Example

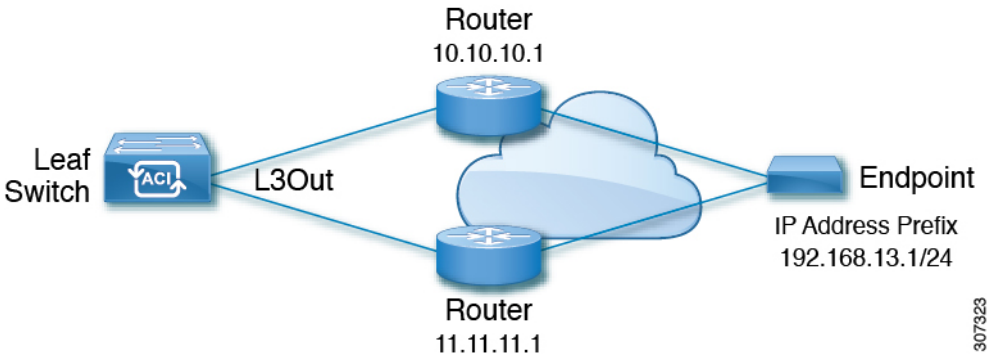
The following figure shows a Cisco ACI IP SLA providing monitoring/probing of a specific configured static route within the ACI fabric and including an external endpoint.

Figure 31: Example ACI L3Out IP SLA



The following image shows a static route for the endpoint prefix of 192.168.13.1/24. It also shows a pair of routers in a static route between an L3Out leaf switch and a consumer endpoint.

Figure 32: Example Static Route



To configure an ACI IP SLA based on the figure above, the router must be monitored to ensure connectivity to the consumer endpoint. This is accomplished by creating a static route, track members, and track lists:

- Static route for 192.168.13.1/24 with next hops of 10.10.10.1 and 11.11.11.1
- Track Member 1 (TM-1) includes the router IP address 10.10.10.1 (this is the next hop probe)
- Track Member 2 (TM-2) includes the router IP address 11.11.11.1 (this is the next hop probe)
- Track List 1 (TL-1) with TM-1 and TM-2 included (track list associated with a static route. The track list contains list of next hops through which configured prefix end points can be reached. Thresholds determining if the track list is reachable or unreachable are also configured.)
- Track List 2 (TL-2) with TM-1 included (associated with a next hop entry included in a static route)
- Track List 3 (TL-3) with TM-2 included (associated with a next hop entry included in a static route)

For a generic static route, you can associate TL-1 with the static route, associate TL-2 with the 10.10.10.1 next hop, and associate TL-3 with the 11.11.11.1 next hop. For a pair of specific static routes (both 192.168.13.1/24), you can associate TL-2 on one and TL-3 on the other. Both should also have TL-2 and TL-3 associated with the router next hops.

These options allow for one router to fail while providing a back-up route in case of the failure. See the following sections to learn more about track members and track lists.

Guidelines and Limitations for IP SLA

Consider the following guidelines and limitations when planning and configuring IP Service Level Agreements:

- IP SLA supports both IPv4 and IPv6 addresses
- IP SLA is supported in all Cisco Nexus second generation switches, which includes the -EX and -FX chassis.
- Beginning in Cisco Application Policy Infrastructure Controller (APIC) release 4.1(1), the IP SLA monitor policy validates the IP SLA port value. Because of the validation, when TCP is configured as the IP SLA type, Cisco APIC no longer accepts an IP SLA port value of 0, which was allowed in previous releases. An IP SLA monitor policy from a previous release that has an IP SLA port value of 0 becomes invalid if the Cisco APIC is upgraded to release 4.1(1) or later. This results in a failure for the configuration import or snapshot rollback.

The workaround is to configure a non-zero IP SLA port value before upgrading the Cisco APIC, and use the snapshot and configuration export that was taken after the IP SLA port change.

- You must enable global GIPo if you are supporting remote leaf switches in an IP SLA:
  1. On the menu bar, click **System** > **System Settings**.
  2. In the System Settings navigation pane, click **System Global GIPo**.
  3. In the System Global GIPo Policy work pane, click **Enabled**.
  4. In the Policy Usage Warning dialog, review the nodes and policies that may be using the GIPo policy and, if appropriate, click **Submit Changes**.
- Statistics viewed through Fabric > Inventory > Pod *number* > Leaf Node *name* > Protocols > IP SLA > ICMP Echo Operations or TCP Connect Operations can only be gathered in five minute intervals. The interval default is **15 Minute**, but this must be set to **5 Minute**.
- IP SLA policy is not supported for endpoints connected through vPod.
- IP SLA is supported for single pods, Cisco ACI Multi-Pod, and remote leaf switches.
- IP SLA is not supported when the destination IP address to be tracked is connected across Cisco ACI Multi-Site.
- If a border leaf switch has a static route that is redistributed in MP-BGP (Multiprotocol Border Gateway Protocol) for the VRF, the MP-BGP route has the same administrative distance as the static route as shown below:

```
leaf102# show ip route 10.10.10.10/32 vrf test:VRF-1
IP Route Table for VRF "test:VRF-1"
'*' denotes best ucast next-hop
'**' denotes best mcast next-hop
'[x/y]' denotes [preference/metric]
'%<string>' in via output denotes VRF <string>

10.10.10.10/32, ubest/mbest: 1/0
*via 102.0.0.2, vlan45, [1/0], 01w00d, static
```

  This route will be injected into the fabric MP-BGP routes for the VRF and are discovered by other remote leaf switches as an iBGP route as shown below:

```
leaf103# show ip route 10.10.10.10/32 vrf test:VRF-1
IP Route Table for VRF "test:VRF-1"
'*' denotes best ucast next-hop
'**' denotes best mcast next-hop
'[x/y]' denotes [preference/metric]
'%<string>' in via output denotes VRF <string>

10.10.10.10/32, ubest/mbest: 1/0
*via 10.0.200.64%overlay-1, [1/0], 01w00d, bgp-65310, internal, tag 65310
recursive next hop: 10.0.200.64/32%overlay-1
```

  However, the adminstrative distance of iBGP route is the same as the adminstrative distance of the static route instead of the adminstrative distance of iBGP AD.

  This observed in both APIC release 4.1(1) and APIC release 5.0(1).

For information on verified IP SLA numbers, refer to the appropriate *Verified Scalability Guide for Cisco APIC* on the Cisco APIC documentation page.

Configuring and Associating ACI IP SLAs for Static Routes

This section describes the tasks that are required to configure and associate the following IP SLA policies and profiles:

- IP SLA Monitoring Policies
- IP SLA Track Members
- IP SLA Track Lists

The previous components are applied to either static routes or next hop profiles.

Configuring IP SLA Monitoring Policy Using the GUI

To enable Cisco APIC to send monitoring probes for a specific SLA type using the APIC GUI, perform the following steps:

Procedure

Step 1 On the menu bar, click **Tenant** > **tenant_name**. In the navigation pane, click **Policies** > **Protocol** > **IP SLA**.

Step 2 Right-click **IP SLA Monitoring Policies**, and click **Create IP SLA Monitoring Policy**.

Step 3 In the **Create IP SLA Monitoring Policy** dialog box, perform the following actions:

a) In the **Name** field, enter a unique name for the IP SLA Monitoring policy.

b) In the **SLA Frequency** field, enter a value, in seconds, to determine the configured frequency to track a packet.

The range is from 1 to 65535. The default value is 60.

c) In the **Detect Multiplier** field, enter a value for the number of missed probes in a row that shows that a failure is detected or a track is down.

By default, failures are detected when three probes are missed in a row. Changing the value in the **Detect Multiplier** field changes the number of missed probes in a row that will determine when failures are detected or when a track is considered to be down.

Used in conjunction with the entry in the **SLA Frequency**, you can determine when a failure will be detected. For example, assume you have the following entries in these fields:

- **SLA Frequency (sec)**: 5
- **Detect Multiplier**: 30

A failure would be detected in roughly 150 seconds in this example scenario (5 seconds x 30).

d) In the **SLA Type** field, choose the SLA type.

The SLA type can be **TCP**, **ICMP**, or **L2Ping**. **ICMP** is the default value.

Note **L2Ping** is supported only for L1/L2 PBR tracking.

e) If you chose **TCP**, enter a port number in the **Destination Port** field.

f) Click **Submit**.

The IP SLA Monitoring Policy is configured.

Configuring an IP SLA Monitoring Policy Using the NX-OS-Style CLI

To configure Cisco Application Policy Infrastructure Controller (APIC) to send monitoring probes for a specific SLA type using the NX-OS-style CLI, perform the following steps:

Before you begin

Make sure a tenant is configured.

Procedure

Step 1 Enter the configuration mode.

Example:

```
apic1# configure
```

Step 2 Create a tenant and enter tenant configuration mode, or enter tenant configuration mode for an existing tenant.

Example:

```
apic1(config)# tenant t1
```

Step 3 Create an IP SLA monitoring policy and enter IP SLA policy configuration mode.

Example:

```
apic1(config-tenant)# ipsla-pol ipsla-policy-3
```

Step 4 Configure the monitoring frequency in seconds, which is the interval between sending probes.

Example:

```
apic1(config-ipsla-pol)# sla-frequency 40
```

Step 5 Configure the monitoring probe type.

The possible values for the type are:

- `icmp`
- `l2ping`
- `tcp sla-port` *number*

Only ICMP and TCP are valid for IP SLA in static routes.

Example:

```
apic1(config-ipsla-pol)# sla-type tcp sla-port 90
```

What to do next

To view the IP SLA monitoring policy you just created, enter:

show running-config all tenant *tenant-name* ipsla-pol

The following output appears:

```
# Command: show running-config all tenant 99 ipsla-pol
# Time: Tue Mar 19 19:01:06 2019
  tenant t1
    ipsla-pol ipsla-policy-3
      sla-detectmultiplier 3
      sla-frequency 40
      sla-type tcp sla-port 90
        sla-port 90
        exit
      exit
    exit
```

Configuring an IP SLA Monitoring Policy Using the REST API

To enable Cisco APIC to send monitoring probes for a specific SLA type using REST API, perform the following steps:

Procedure

Configure an IP SLA monitoring policy.

Example:

```
<?xml version="1.0" encoding="utf-8"?>
<imdata totalCount="1">
                <fvIPSLAMonitoringPol annotation="" descr=""
dn="uni/tn-t8/ipslaMonitoringPol-ICMP-Probe"
                 name="ICMP-Probe" nameAlias="" ownerKey="" ownerTag=""
slaDetectMultiplier="3" slaFrequency="5"
                 slaPort="0" slaType="icmp"/>
</imdata>
```

Configuring IP-SLA Track Members Using the GUI

Use this task to create an IP SLA track member which is one of a number added to an IP SLA track list. Track lists are applied to static routes to monitor performance from one defined next hop to another.

Before you begin

You must have created an IP SLA monitoring policy and know the destination IP address for the next hop this track member represents in a static route.

To configure an IP SLA track member using the APIC GUI, perform the following steps:

Procedure

Step 1 On the menu bar, click **Tenants** > ***tenant-name*** .

Step 2 In the Navigation pane, expand **Policies** and then expand **Protocol**.

Step 3 Expand **IP SLA**, right-click **Track Members** and choose **Create Track Member**.

Step 4 Configure the following parameters:

a) In the **Name** field, enter a unique name for the track member.
b) In the **Destination IP** field, enter the IP address of the next hop this configuration represents.
c) In the **Scope of Track Member** drop-down list, choose an existing bridge domain or external network to which this track member belongs.
d) In the **IP SLA Policy** field, select an existing or create a new IP SLA monitoring policy that defines the probe that is used during monitoring.

Step 5 Click **Submit**.

What to do next

Repeat the preceding steps to create the required number of track members for the static route to be monitored. Once all track members are configured, create a track list and add them to it.

Configuring an IP-SLA Track Member Using the NX-OS Style CLI

To configure an IP SLA track member using the NX-OS style CLI, perform the following steps:

Before you begin

Make sure a tenant and an IP SLA monitoring policy under the tenant is configured.

Procedure

Step 1 **configure**

Enters configuration mode.

Example:

```
apic1# configure
```

Step 2 **tenant** *tenant-name*

Creates a tenant or enters tenant configuration mode.

Example:

```
apic1(config)# tenant t1
```

Step 3 **track-member** *name* **dst-IpAddr** *ipv4-or-ipv6-address* **l3-out** *name*

Creates a track member with a destination IP address and enters track member configuration mode.

Example:

```
apic1(config-tenant)# )# track-member tm-1 dst-IpAddr 10.10.10.1 l3-out ext-l3-1
```

Step 4 **ipsla-monpol** *name*

Assigns an IP SLA monitoring policy to the track member.

Example:

```
apic1(config-track-member)# ipsla-monpol ipsla-policy-3
```

Example

The following example shows the commands to configure an IP SLA track member.

```
apic1# configure
      apic1(config)# tenant t1
      apic1(config-tenant)# )# track-member tm-1 dst-IpAddr 10.10.10.1 l3-out ext-l3-1
      apic1(config-track-member)# ipsla-monpol ipsla-policy-3
```

What to do next

To view the track member configuration you just created, enter:

show running-config all tenant *tenant-name* **track-member** *name*

The following output appears:

```
# Command: show running-config all tenant 99 track-member tm-1
# Time: Tue Mar 19 19:01:06 2019
  tenant t1
    track-member tm-1 10.10.10.1 l3-out ext-l3-1
      ipsla-monpol slaICMPProbe
      exit
    exit
```

Configuring an IP-SLA Track Member Using the REST API

To configure an IP SLA track member using REST API, perform the following steps:

Procedure

Configure an IP SLA track member.

Example:

```
<?xml version="1.0" encoding="utf-8"?>
<imdata totalCount="1">
                <fvTrackMember annotation="" descr="" dn="uni/tn-t8/trackmember-TM_pc_sub"

              dstIpAddr="52.52.52.1" name="TM_pc_sub" nameAlias="" ownerKey="" ownerTag=""

                scopeDn="uni/tn-t8/out-t8_l3">
                        <fvRsIpslaMonPol annotation=""
tDn="uni/tn-t8/ipslaMonitoringPol-TCP-Telnet"/>
```

```
                </fvTrackMember>
</imdata>
```

Configuring an IP-SLA Track List Using the GUI

Use this task to create an IP SLA track list which defines a group of track members representing the next hops in a static route. Track lists are applied to static routes to monitor performance from one defined next hop to another.

Before you begin

You must have created one or more IP SLA track members.

To configure an IP SLA track list using the APIC GUI, perform the following steps:

Procedure

Step 1 On the menu bar, click **Tenants** > ***tenant-name*** .

Step 2 In the Navigation pane, expand **Policies** and then expand **Protocol**.

Step 3 Expand **IP SLA**, right-click **Track Lists** and choose **Create Track List**.

The **Create Track List** dialog appears.

Step 4 Configure the following parameters:

a) In the **Name** field, enter a unique name for the track list.

b) In the **Type of Track List** field, choose **Threshold percentage** if you want the route availability to be based on the percentage of track members that are up or down. Choose **Threshold weight** if the route availability is based on a weight value that is assigned to each track member.

c) In the **Track list to track member relation** table, click the + icon in the table head to add a track member to the list. Choose an existing track member and, if the **Type of Track List** is **Threshold weight**, assign a weight value.

Step 5 Click **Submit**.

What to do next

Associate the track list with a static route or next hop IP address.

Configuring an IP-SLA Track List Using the NX-OS Style CLI

To configure an IP SLA track list using the NX-OS style CLI, perform the following steps:

Before you begin

Make sure a tenant, an IP SLA monitoring policy, and at least one track member under the tenant is configured.

Procedure

Step 1 **configure**

Enters configuration mode.

Example:

```
apic1# configure
```

Step 2 **tenant** *tenant-name*

Creates a tenant or enters tenant configuration mode.

Example:

```
apic1(config)# tenant t1
```

Step 3 **track-list** *name* **{ percentage [percentage-down | percentage-up]** *number* **| weight [weight-down | weight-up** *number* **}**

Creates a track list with percentage or weight threshold settings and enters track list configuration mode.

Example:

```
apic1(config-tenant)# )# track-list tl-1 percentage percentage-down 50 percentage-up 100
```

Step 4 **track-member** *name*

Assigns an existing track member to the track list.

Example:

```
apic1(config-track-list)# track-member tm-1
```

Example

The following example shows the commands to configure an IP SLA track list.

```
apic1# configure
      apic1(config)# tenant t1
      apic1(config-tenant)# )# track-list tl-1 percentage percentage-down 50 percentage-up
 100
      apic1(config-track-list)# track-member tm1
```

What to do next

To view the track member configuration you just created, enter:

show running-config all tenant *tenant-name* **track-member** *name*

The following output appears:

```
# Command: show running-config all tenant 99 track-list tl-1
# Time: Tue Mar 19 19:01:06 2019
  tenant t1
    track-list tl-1 percentage percentage-down 50 percentage-up 100
```

```
  track-member tm-1 weight 10
  exit
exit
```

Configuring an IP-SLA Track List Using the REST API

To configure an IP SLA track list using REST API, perform the following steps:

Procedure

Configure an IP SLA track list.

Example:

```
<?xml version="1.0" encoding="utf-8"?>
<imdata totalCount="1">
                <fvTrackList annotation="" descr="" dn="uni/tn-t8/tracklist-T8_pc_sub1"
                name="T8_pc_sub1" nameAlias="" ownerKey="" ownerTag="" percentageDown="0"
                percentageUp="1" type="weight" weightDown="5" weightUp="10">
                        <fvRsOtmListMember annotation=""
tDn="uni/tn-t8/trackmember-TM_pc_sub"
                        weight="10"/>
                </fvTrackList>
</imdata>
```

Associating a Track List with a Static Route Using the GUI

Use this task to associate a track list with a configured static route allowing the system to monitor the performance of a series of next hops.

Note The following task assumes that a next hop configuration already exists for the static route.

Before you begin

A configured routed network with a static route must be available. A configured track list must also be available.

To associate an IP SLA track list with a static route using the APIC GUI, perform the following steps:

Procedure

Step 1 On the menu bar, click **Tenants** > ***tenant-name*** .

Step 2 In the Navigation pane, expand **Networking** and then expand **External Routed Networks**.

Step 3 Expand the configured routed network (name), **Logical Node Profiles**, a configured logical node profile (name), and **Configured Nodes**.

Step 4 Click a configured node (name).

The **Node Association** work pane appears.

Step 5 In the **Static Routes** table, double-click the route entry to which you want to add the track list.

The **Static Route** dialog appears.

Step 6 In the **Track Policy** drop-down list, choose or create an IP SLA track list to associate with this static route.

Step 7 Click **Submit**.

Step 8 The **Policy Usage Warning** dialog appears.

Step 9 Verify that this change will not impact other nodes or policies using this static route and click **Submit Changes**.

Associating a Track List with a Static Route Using the NX-OS Style CLI

To associate an IP SLA track list with a static route using the NX-OS style CLI, perform the following steps:

Before you begin

Make sure a tenant, a VRF, and a track list under the tenant is configured.

Procedure

Step 1 **configure**

Enters configuration mode.

Example:

```
apic1# configure
```

Step 2 **leaf** *id or leaf-name*

Selects a leaf switch and enter the leaf switch configuration mode.

Example:

```
apic1(config)# leaf 102
```

Step 3 **vrf context tenant** *name* **vrf** *name*

Selects a VRF context and enters the VRF configuration mode.

Example:

```
apic1(config-leaf)# )# vrf context tenant 99 vrf default
```

Step 4 **ip route** *ip-address next-hop-ip-address route-prefix* **bfd ip-trackList** *name*

Assigns an existing track list to the static route.

Example:

```
apic1(config-leaf-vrf)# ip route 10.10.10.1/4 20.20.20.8 10 bfd ip-trackList tl-1
```

Example

The following example shows the commands to associate an IP SLA track list with a static route.

```
apic1# configure
      apic1(config)# leaf 102
      apic1(config-leaf)# )# vrf context tenant 99 vrf default
      apic1(config-leaf-vrf)# ip route 10.10.10.1/4 20.20.20.8 10 bfd ip-trackList tl-1
```

Associating a Track List with a Static Route Using the REST API

To associate an IP SLA track list with a static route using REST API, perform the following steps:

Procedure

Associate an IP SLA track list with a static route.

Example:

```
<?xml version="1.0" encoding="utf-8"?>
<imdata totalCount="1">
                <ipRouteP aggregate="no" annotation="" descr=""

dn="uni/tn-t8/out-t8_l3/lnodep-t8_l3_vpc1/rsnodeL3OutAtt-[topology/pod-2/node-108]/rt-[88.88.88.2/24]"

                ip="88.88.88.2/24" name="" nameAlias="" pref="1" rtCtrl="">
                        <ipRsRouteTrack annotation=""
tDn="uni/tn-t8/tracklist-T8_TL1_Static"/>
                        <ipNexthopP annotation="" descr="" name="" nameAlias=""
nhAddr="23.23.2.3"
                        pref="1" type="prefix"/>
                </ipRouteP>
</imdata>
```

Associating a Track List with a Next Hop Profile Using the GUI

Use this task to associate a track list with a configured next hop profile in a static route allowing the system to monitor the next hop performance.

Before you begin

A configured routed network with a static route and next hop profile must be available.

To associate an IP SLA track list with a next hop profile using the APIC GUI, perform the following steps:

Procedure

Step 1 On the menu bar, click **Tenants** > ***tenant-name*** .

Step 2 In the Navigation pane, expand **Networking** and then expand **External Routed Networks**.

Step 3 Expand the configured routed network (name), **Logical Node Profiles**, a configured logical node profile (name), and **Configured Nodes**.

Step 4 Click a configured node (name).

The **Node Association** work pane appears.

Step 5 In the **Static Routes** table, double-click the route entry to which you want to add the track list.

The **Static Route** dialog appears.

Step 6 In the **Next Hop Addresses** table, double-click the next hop entry to which you want to add the track list.

The **Next Hop Profile** dialog appears.

Step 7 In the **Track Policy** drop-down list, choose or create an IP SLA track list to associate with this static route.

Note If you add an IP SLA Policy to the next hop profile, a track member and track list is automatically created and associated with the profile.

Step 8 Click **Submit**.

Step 9 The **Policy Usage Warning** dialog appears.

Step 10 Verify that this change will not impact other nodes or policies using this static route and click **Submit Changes**.

Associating a Track List with a Next Hop Profile Using the NX-OS Style CLI

To associate an IP SLA track list with a next hop profile using the NX-OS style CLI, perform the following steps:

Before you begin

Make sure a tenant, a VRF, and a track list under the tenant is configured.

Procedure

Step 1 **configure**

Enters configuration mode.

Example:

```
apic1# configure
```

Step 2 **leaf** *id or leaf-name*

Selects a leaf switch and enter the leaf switch configuration mode.

Example:

```
apic1(config)# leaf 102
```

Step 3 **vrf context tenant** *name* **vrf** *name*

Selects a VRF context and enters the VRF configuration mode.

Example:

```
apic1(config-leaf)# )# vrf context tenant 99 vrf default
```

Step 4 **ip route** *ip-address next-hop-ip-address route-prefix* **bfd nh-ip-trackList** *name*

Assigns an existing track list to the next hop.

Example:

```
apic1(config-leaf-vrf)# ip route 10.10.10.1/4 20.20.20.8 10 bfd nh-trackList tl-1
```

Example

The following example shows the commands to associate an IP SLA track list with a next hop profile.

```
apic1# configure
      apic1(config)# leaf 102
      apic1(config-leaf)# )# vrf context tenant 99 vrf default
      apic1(config-leaf-vrf)# ip route 10.10.10.1/4 20.20.20.8 10 bfd nh-ip-trackList tl-1
```

Associating a Track List with a Next Hop Profile Using the REST API

To associate an IP SLA track list with a next hop profile using REST API, perform the following steps:

Procedure

Associate an IP SLA track list with a next hop profile.

Example:

```
<?xml version="1.0" encoding="utf-8"?>
<imdata totalCount="1">
                <ipRouteP aggregate="no" annotation="" descr=""

dn="uni/tn-t8/out-t8_l3/lnodep-t8_l3_vpc1/rsnodeL3OutAtt-[topology/pod-2/node-109]/rt-[86.86.86.2/24]"

                ip="86.86.86.2/24" name="" nameAlias="" pref="1" rtCtrl="">
                        <ipNexthopP annotation="" descr="" name="" nameAlias=""
nhAddr="25.25.25.3" pref="1" type="prefix">
                                <ipRsNexthopRouteTrack annotation=""
tDn="uni/tn-t8/tracklist-ctx0_25.25.25.3"/>
                                <ipRsNHTrackMember annotation=""
tDn="uni/tn-t8/trackmember-ctx0_25.25.25.3"/>
                        </ipNexthopP>
                </ipRouteP>
</imdata>
```

Viewing ACI IP SLA Monitoring Information

This section describes the tasks that are required to view IP SLA statistics, track lists, track members, and associated static routes:

- Viewing ACI IP SLA Probe Statistics Using the GUI
- Viewing Track List and Track Member Status Using the CLI

Viewing IP SLA Probe Statistics Using the GUI

ACI IP SLAs generate the following real-time statistics:

ICMP

- ICMP Echo Round Trip Time (milliseconds)
- Number of Failed ICMP Echo Probes (packets)
- Number of Successful ICMP Echo Probes (packets)
- Number of Transmitted ICMP Echo Probes (packets)

TCP

- Number of Failed TCP Connect Probes (packets)
- Number of Successful TCP Connect Probes (packets)
- Number of Transmitted TCP Connect Probes (packets)
- TCP Connect Round Trip Time (milliseconds)

Use this task to view statistics for an IP SLA track list or member currently monitoring a static route or next hop.

Before you begin

You must have created an IP SLA track list and associated it with a static route before viewing statistics.

Procedure

Step 1 On the menu bar, click **Tenants** > ***tenant-name*** .

Step 2 In the Navigation section, expand **Policies** and then expand **Protocol**.

Step 3 Expand **IP SLA** and expand either **Track Members** or **Track Lists**.

Step 4 Click an existing track member or track list you want to view.

Step 5 Click the **Stats** tab.

Step 6 Click the **Select Stats** icon to choose the probe statistic types you want to view.

Step 7 Choose a probe statistic type (chosen statistic types are highlighted in blue) and move it from **Available** to **Selected** with the arrow icon. You can move a probe statistics type from **Selected** back to **Available** with the opposite arrow icon.

Step 8 When finished selecting the probe statistic type(s) you want to view, click **Submit**.

What to do next

The statistics chosen in this task are labeled in the legend above the graph. Lines representing the selected probe statistic types should begin to appear on the graph as the counters begin to accumulate.

Viewing Track List and Track Member Status Using the CLI

You can display IP SLA track list and track member status.

Procedure

	Command or Action	Purpose
Step 1	show track brief **Example:** `switch# show track brief`	Displays the status of all track lists and track members.

Example

```
switch# show track brief
TrackId  Type     Instance    Parameter        State    Last Change
97       IP SLA   2034        reachability     up       2019-03-20T14:08:34.127-07:00
98       IP SLA   2160        reachability     up       2019-03-20T14:08:34.252-07:00
99       List     ---         percentage       up       2019-03-20T14:08:45.494-07:00
100      List     ---         percentage       down     2019-03-20T14:08:45.039-07:00
101      List     ---         percentage       down     2019-03-20T14:08:45.040-07:00
102      List     ---         percentage       up       2019-03-20T14:08:45.495-07:00
103      IP SLA   2040        reachability     up       2019-03-20T14:08:45.493-07:00
104      IP SLA   2887        reachability     down     2019-03-20T14:08:45.104-07:00
105      IP SLA   2821        reachability     up       2019-03-20T14:08:45.494-07:00
1        List     ---         percentage       up       2019-03-20T14:08:39.224-07:00
2        List     ---         weight           down     2019-03-20T14:08:33.521-07:00
3        IP SLA   2412        reachability     up       2019-03-20T14:08:33.983-07:00
26       IP SLA   2320        reachability     up       2019-03-20T14:08:33.988-07:00
27       IP SLA   2567        reachability     up       2019-03-20T14:08:33.987-07:00
28       IP SLA   2598        reachability     up       2019-03-20T14:08:33.990-07:00
29       IP SLA   2940        reachability     up       2019-03-20T14:08:33.986-07:00
30       IP SLA   2505        reachability     up       2019-03-20T14:08:38.915-07:00
31       IP SLA   2908        reachability     up       2019-03-20T14:08:33.990-07:00
32       IP SLA   2722        reachability     up       2019-03-20T14:08:33.992-07:00
33       IP SLA   2753        reachability     up       2019-03-20T14:08:38.941-07:00
34       IP SLA   2257        reachability     up       2019-03-20T14:08:33.993-07:00
```

Viewing Track List and Track Member Detail Using the CLI

You can display IP SLA track list and track member detail.

Procedure

	Command or Action	Purpose
Step 1	show track [*number*] | more **Example:** `switch# show track | more`	Displays the detail of all track lists and track members.

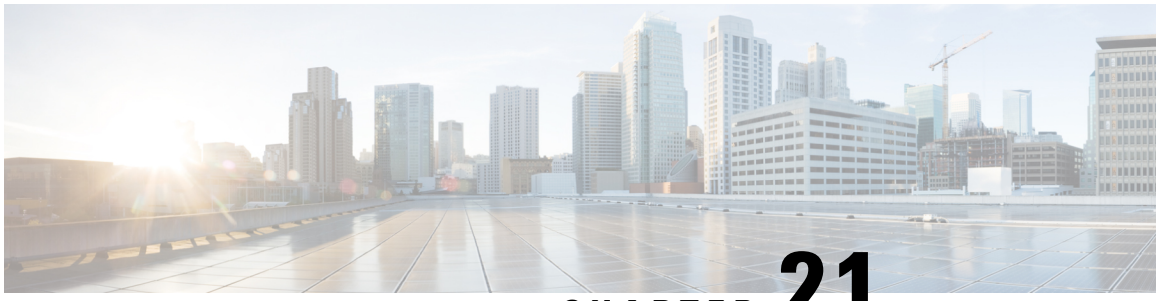
Example

```
switch# show track | more
Track 4
    IP SLA 2758
    reachability is down
    1 changes, last change 2019-03-12T21:41:34.729+00:00
    Tracked by:
        Track List 3
        Track List 5

Track 3
    List Threshold percentage
    Threshold percentage is down
    1 changes, last change 2019-03-12T21:41:34.700+00:00
    Threshold percentage up 1% down 0%
    Tracked List Members:
        Object 4 (50)% down
        Object 6 (50)% down
    Attached to:
        Route prefix 172.16.13.0/24

Track 5
    List Threshold percentage
    Threshold percentage is down
    1 changes, last change 2019-03-12T21:41:34.710+00:00
    Threshold percentage up 1% down 0%
    Tracked List Members:
        Object 4 (100)% down
    Attached to:
        Nexthop Addr 12.12.12.2/32

Track 6
    IP SLA 2788
    reachability is down
    1 changes, last change 2019-03-14T21:34:26.398+00:00
    Tracked by:
        Track List 3
        Track List 7

Track 20
    List Threshold percentage
    Threshold percentage is up
    4 changes, last change 2019-02-21T14:04:21.920-08:00
    Threshold percentage up 100% down 32%
    Tracked List Members:
    Object 4 (20)% up
    Object 5 (20)% up
    Object 6 (20)% up
    Object 3 (20)% up
    Object 9 (20)% up
    Attached to:
    Route prefix 88.88.88.0/24
    Route prefix 5000:8:1:14::/64
    Route prefix 5000:8:1:2::/64
    Route prefix 5000:8:1:1::/64
```

In this example, Track 4 is a track member identified by the IP SLA ID and by the track lists in the **Tracked by:** field.

Track 3 is a track list identified by the threshold information and the track member in the **Track List Members** field.

Track 20 is a track list that is currently reachable (up) and shows the static routes to which it is associated.



CHAPTER 21

Microsoft NLB

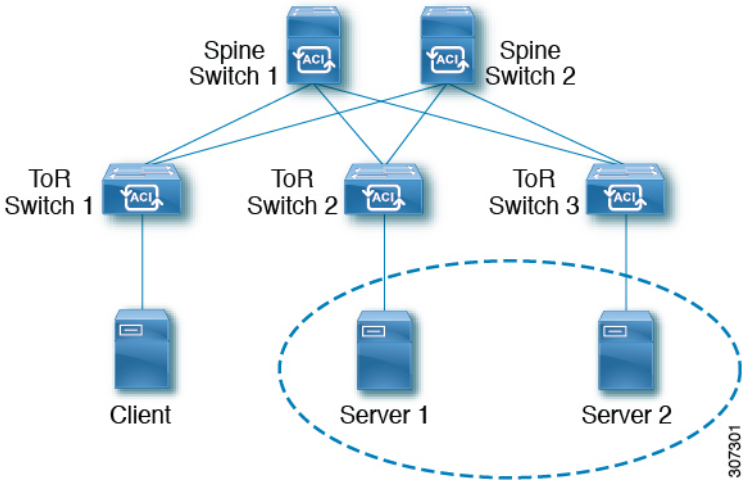
This chapter contains the following sections:

About Microsoft NLB

The Microsoft Network Load Balancing (NLB) feature distributes the client traffic across many servers, with each server running its individual copy of the application. Network Load Balancing uses Layer 2 unknown unicast or multicast to simultaneously distribute the incoming network traffic to all cluster hosts.

A group of Microsoft NLB nodes is collectively known as an NLB cluster. An NLB cluster serves one or more virtual IP (VIP) addresses. Nodes in the NLB cluster use a load-balancing algorithm to decide which individual node will service the particular traffic flow that is destined for the NLB VIP. Every node within the cluster receives every packet of traffic, but only one node services a request.

The following figure shows a graphical representation of how Microsoft NLB is implemented with Cisco APIC.



In this figure, Server 1 and Server 2 are in the MS NLB cluster. These servers appear as a single-host server to outside clients. All servers in the MS NLB cluster receive all incoming requests, then MS NLB distributes the load between the servers.

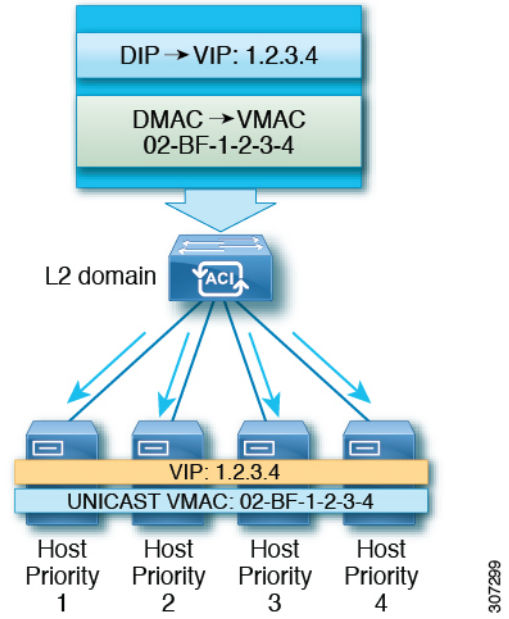
Microsoft NLB functions in three different operational modes:

- **Unicast Mode:** In this mode, each NLB cluster VIP is assigned a unicast MAC address. This mode relies on unknown unicast flooding to deliver traffic to the cluster.
- **Multicast Mode:** In this mode, each NLB cluster VIP is assigned a non-Internet Assigned Numbers Authority (IANA) multicast MAC address (03xx.xxxx.xxxx).
- **IGMP Mode:** In this mode, an NLB cluster VIP is assigned a unique IPv4 multicast group address. The multicast MAC address for this is derived from the standard MAC derivation for IPv4 multicast addresses.

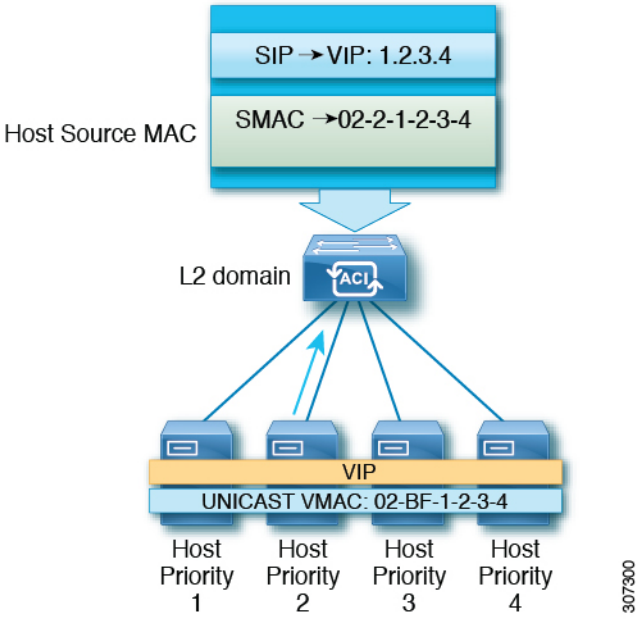
Understanding Unicast Mode

In the unicast mode of operation, Network Load Balancing reassigns the MAC address of the network adapter on which it is enabled (called the cluster adapter), and all cluster hosts are assigned the same MAC address. This MAC address is derived from the cluster's primary IP address. For example, for a primary IP address of 1.2.3.4, the unicast MAC address is set to 02-BF-1-2-3-4.

Network Load Balancing's unicast mode induces switch flooding in order to simultaneously deliver incoming network traffic to all cluster hosts, as shown in the following figure.



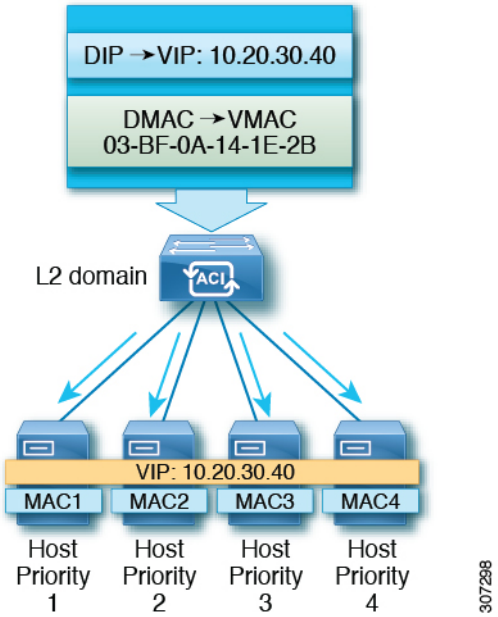
The use of a common MAC address would normally create a conflict, since Layer 2 switches expect to see unique source MAC addresses on all switch ports. To avoid this problem, Network Load Balancing uniquely modifies the source MAC address for outgoing packets. If the cluster MAC address is 02-BF-1-2-3-4, then each host's source MAC address is set to 02-*x*-1-2-3-4, where *x* is the host's priority within the cluster, as shown in the following figure.



Understanding Multicast Mode

Network Load Balancing also provides multicast mode for distributing incoming network traffic to all cluster hosts. Multicast mode assigns a Layer 2 multicast address to the cluster adapter instead of changing the

adapter's MAC address. For example, the multicast MAC address could be set to 03-BF-0A-14-1E-28 for a cluster's primary IP address of 10.20.30.40. Cluster communication doesn't require a separate adapter.

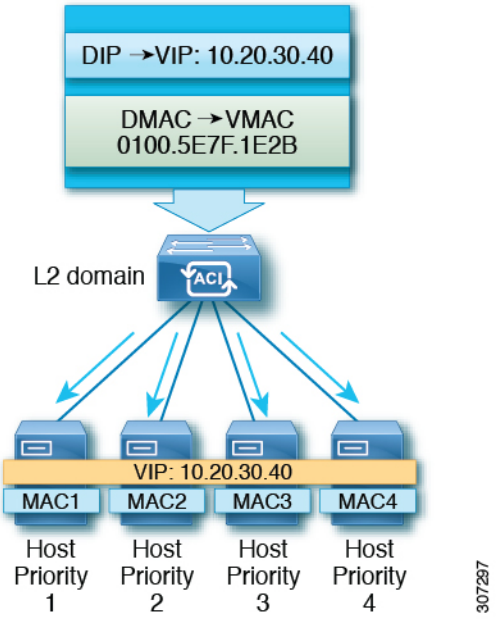


Understanding IGMP Mode

Microsoft NLB servers can also be configured to use IGMP to join the multicast group. This, combined with a querier on a switch and IGMP snooping, can optimize the scope of the flooding of multicast messages.

Microsoft NLB servers send IGMP joins to a multicast group address, where the last two octets of the multicast address correspond to the last two octets of the cluster IP. For example, in a situation where the Microsoft NLB servers send IGMP joins to a multicast address of 239.255.x.x, the following would occur:

- Cluster IP: 10.20.**30.40**
- IGMP sent to 239.255.**30.40**
- MAC used in client-to-server direction: 0100.5E7F.**1E28**
- Cluster communication doesn't require a separate adapter



Cisco ACI Configuration for Microsoft NLB Servers

Prior to Release 4.1, Microsoft NLB deployment requires the Cisco ACI fabric to be Layer 2 only and uses an external router as the Layer 3 gateway for endpoints. Starting with Release 4.1, the Cisco ACI fabric can be a Layer 3 gateway for Microsoft NLB deployment.

The following table summarizes the deployment considerations for each Microsoft NLB deployment mode.

Table 8: Cisco ACI Deployment Modes with Microsoft NLB

	Unicast Mode	Multicast Mode	IGMP Mode
Cisco ACI as a Layer 2 Network, With External Router as Layer 3 Gateway	Supported on leaf switch models with -EX, -FX, or -FX2 at the end of the switch name.	Supported on leaf switch models with -EX, -FX, or -FX2 at the end of the switch name, as well as leaf switch models that do not have a suffix at the end of the switch name.	Supported on leaf switch models with -EX, -FX, or -FX2 at the end of the switch name, as well as leaf switch models that do not have a suffix at the end of the switch name. However, Microsoft NLB traffic is not scoped by IGMP, but rather is flooded instead.
Cisco ACI as a Layer 3 Gateway	Supported on Release 4.1 and later.	Supported on Release 4.1 and later.	Supported on Release 4.1 and later.

The following table provides more information on the configuration options available for deploying Microsoft NLB using Cisco ACI as Layer 2.

Table 9: External Router and ACI Bridge Domain Configuration for the Three Microsoft NLB Modes

	Unicast Mode	Multicast Mode	IGMP Mode[3]
ACI Bridge Domain Configuration	• Bridge domain configured for unknown unicast flooding (not hw-proxy) • No IP routing	• Bridge domain configured for unknown unicast flooding (not hw-proxy) • No IP routing • Layer 3 unknown multicast: flood (even with optimized multicast flooding, Microsoft NLB traffic is flooded) • IGMP snooping configuration: Not applicable	• Bridge domain configured for unknown unicast flooding (not hw-proxy) • No IP routing • Layer3 unknown multicast: Optional, but can be configured for future compatibility • Querier configuration: Optional, but can be enabled for future compatibility; Configure subnet under the bridge domain, no need for IP routing • IGMP snooping configuration: Optional, but can be enabled for future compatibility
External Router ARP Table Configuration	• No special ARP configuration • External router learns VIP to VMAC mapping	Static ARP configuration for unicast VIP to multicast MAC	Static ARP configuration for unicast VIP to multicast MAC

[3] As of Release 3.2, using Microsoft NLB IGMP mode compared with Microsoft NLB multicast mode offers no benefits in terms of scoping of the multi-destination traffic

Beginning with Release 4.1, configuring Cisco ACI to connect Microsoft NLB servers consists of the following general tasks:

- Configuring the VRF, where you can configure the VRF in egress or ingress mode.
- Configuring a bridge domain (BD) for the Microsoft NLB servers, with L2 unknown unicast in flooding mode and not in hardware-proxy mode.
- Defining an EPG for all the Microsoft NLB servers that share the same VIP. You must associate this EPG with the previously defined BD.
- Entering the Microsoft NLB VIP as a subnet under the EPG. You can configure the Microsoft NLB in the following modes:
  - **Unicast mode:** You will enter the unicast MAC address as part of the Microsoft NLB VIP configuration. In this mode, the traffic from the client to the Microsoft NLB VIP is flooded to all the EPGs in the Microsoft NLB BD.

- **Multicast mode:** You will enter the multicast MAC address while configuring the Microsoft NLB VIP itself. You will go to the static ports under the Microsoft NLB EPG and add the Microsoft NLB multicast MAC to the EPG ports where the Microsoft NLB servers are connected. In this mode, the traffic is forwarded to the ports that have the static MAC binding.
- **IGMP mode:** You will enter a Microsoft NLB group address while configuring the Microsoft NLB VIP itself. In this mode, the traffic from the client to the Microsoft NLB VIP is forwarded to the ports where the IGMP join is received for the Microsoft NLB group address.

- Configuring a contract between the Microsoft NLB EPG and the client EPG. You must configure the Microsoft NLB EPG as the provider side of the contract and the client EPG as the consumer side of the contract.

Microsoft NLB is a route plus flood solution. Traffic from the client to the Microsoft NLB VIP is first routed at the consumer ToR switch, and is then flooded on the Microsoft NLB BD toward the provider ToR switch.

Once traffic leaves the consumer ToR switch, traffic is flooded and contracts cannot be applied to flood traffic. Therefore, the contract enforcements must be done on consumer ToR switch.

For a VRF in ingress mode, intra-VRF traffic from the L3Out to the Microsoft NLB EPG may be dropped on the consumer ToR switch because the border leaf switch (consumer ToR switch) does not have a policy. To work around this issue, use one of the following options:

- **Option 1:** Configure the VRF in egress mode. When you configure the VRF in egress mode, the policy is downloaded on the border leaf switch.
- **Option 2:** Add the Microsoft NLB EPG and L3external of the L3Out in a preferred group. Traffic will hit the default-allow policy on the consumer ToR switch.
- **Option 3:** Deploy the Microsoft NLB EPG on an unused port that is in an up state, or on a port connected to a Microsoft NLB server on the border leaf switch. By doing so, the Microsoft NLB EPG becomes a local endpoint on the border leaf switch. The policy is downloaded for local endpoints, so the border leaf switch would therefore have the policy downloaded.
- **Option 4:** Use a shared service. Deploy an L3Out in the consumer VRF, which is different from the provider Microsoft NLB VRF. For the Microsoft NLB VIP under the Microsoft NLB EPG, check the **Shared between VRFs** box. Configure a contract between L3Out from the consumer VRF and the Microsoft NLB EPG. By using a shared service, the policy is downloaded on the border leaf switch.

The following table provides more information on supported EPG and BD configurations for the Microsoft NLB modes.

Table 10: Cisco ACI EPG and BD Configurations for the Microsoft NLB Modes

	Unicast Mode	Multicast Mode	IGMP Mode
Bridge Domain Configuration	• IP routing on • Bridge domain configured for unknown unicast flooding (not hw-proxy) • Do not change the bridge domain MAC address	• IP routing on • Bridge domain configured for unknown unicast flooding (not hw-proxy) • Do not change the bridge domain MAC address	• IP routing on • Bridge domain configured for unknown unicast flooding (not hw-proxy) • Do not change the bridge domain MAC address

	Unicast Mode	Multicast Mode	IGMP Mode
EPG Configuration	• Subnet for the VIP • Unicast MAC address defined as part of the subnet	• Subnet for the VIP • Multicast MAC address defined as part of the subnet • Static binding to the ports where the servers are • Static group MAC address on each path	• Subnet for the VIP • No need to enter a MAC address • You can choose dynamic group or static group • If you choose the static group option, then enter static paths and enter the multicast group in each path
VMM Domain	You can enter a VMM domain	Multicast mode requires a static path, so you cannot use a VMM domain in this situation	In dynamic group mode, you can use a VMM domain

Guidelines and Limitations

Following are the guidelines and limitations for Microsoft NLB:

- Layer 3 multicast is not supported (you cannot enable PIM on the Microsoft NLB BD).
- For IGMP, the allowable mode group is IPv4 (IPv6 is not supported).
- Only Cisco Nexus 9000 series switches with names that end in EX and later are supported.
- Shared services and microsegment (uSeg) EPGs are supported with Microsoft NLB.
- Cisco ACI Multi-Site is currently not supported.
- You must configure Microsoft NLB in layer 2 unknown unicast flooding mode.

  If you configure the BD for hardware-proxy instead, Cisco ACI raises a fault, which is cleared by fixing the BD configuration. If you leave the BD incorrectly configured for hardware-proxy, ACI tries to get the faulty configuration up every 30 seconds, which is an unnecessary overhead for the switch.
- You should configure Microsoft NLB BD with the default SVI MAC address. Under layer 3 configurations, you should configure the bridge domain MAC address with the default setting of 00:22:BD:F8:19:FF. Do not modify this default SVI MAC address for the Microsoft NLB BD.
- There is a hardware limit of 128 Microsoft NLB VIPs per fabric.
- Virtualized servers that are configured for Microsoft NLB can connect to Cisco ACI with static binding in all modes (unicast, multicast, and IGMP).
- Virtualized servers that are configured for Microsoft NLB can connect to Cisco ACI through VMM integration in unicast mode and IGMP mode.
- Microsoft NLB unicast mode is not supported with VMM integration behind Cisco UCS B-Series Blade Servers in end-host mode.

Microsoft NLB in unicast mode relies on unknown unicast flooding for delivery of cluster-bound packets. Unicast mode will not work on Cisco UCS B-Series Blade Servers when the fabric interconnect is in end-host mode, because unknown unicast frames are not flooded as required by this mode. For more details on the layer 2 forwarding behavior of Cisco UCS B-Series Blade Servers in end-host mode, see:

https://www.cisco.com/c/en/us/solutions/collateral/data-center-virtualization/unified-computing/whitepaper_c11-701962.html

Configuring Microsoft NLB Using the GUI

Configuring Microsoft NLB in Unicast Mode Using the GUI

This task configures Microsoft NLB to flood all of the ports in the bridge domain.

Before you begin

Have the following information available before proceeding with these procedures:

- Microsoft NLB cluster VIP
- Microsoft NLB cluster MAC address

Procedure

Step 1 In the **Navigation** pane, choose **Tenant** > *tenant_name* > **Application Profiles** > *application_profile_name* > **Application EPGs** > *application_EPG_name* > **Subnets**.

Step 2 Right-click **Subnets** and select **Create EPG Subnet**.

Step 3 In the **Create EPG Subnet** dialog box, fill in the following fields:

a) In the **Default Gateway IP** field, enter the Microsoft NLB cluster VIP.

For example, `192.0.2.1/32`.

b) In the **Scope** area, for shared services, check **Shared between VRFs**.

Uncheck **Private to VRF**, if it is selected.

c) Under **Subnet Control**, check the **No Default SVI Gateway** check box.

d) In the **Type Behind Subnet** area, click **EpNlb**.

The **Mode** field appears.

e) From the **Mode** drop-down list, choose **NLB in unicast mode**.

The **MAC Address** field appears.

f) In the **MAC Address** field, enter the Microsoft NLB cluster MAC address.

For example, `00:01:02:03:04:05`.

Step 4 Click **Submit**.

Configuring Microsoft NLB in Multicast Mode Using the GUI

This task configures Microsoft NLB to flood only on certain ports in the bridge domain.

Before you begin

Have the following information available before proceeding with these procedures:

- Microsoft NLB cluster VIP
- Microsoft NLB cluster MAC address

Procedure

Step 1 In the **Navigation** pane, choose **Tenant** > ***tenant_name*** > **Application Profiles** > ***application_profile_name*** > **Application EPGs** > ***application_EPG_name*** > **Subnets**.

Step 2 Right-click **Subnets** and select **Create EPG Subnet**.

Step 3 In the **Create EPG Subnet** dialog box, fill in the following fields:

a) In the **Default Gateway IP** field, enter the Microsoft NLB cluster VIP.

For example, `192.0.2.1/32`.

b) In the **Scope** area, for shared services, check **Shared between VRFs**.

Uncheck **Private to VRF**, if it is selected.

c) Under **Subnet Control**, check the **No Default SVI Gateway** check box.

d) In the **Type Behind Subnet** area, click **EpNlb**.

The **Mode** field appears.

e) From the **Mode** drop-down list, choose **NLB in static multicast mode**.

The **MAC Address** field appears.

f) In the **MAC Address** field, enter the Microsoft NLB cluster MAC address.

For the Microsoft NLB cluster MAC address for the multicast mode, the cluster MAC address has to start with `03`.

For example, `03:BF:01:02:03:04`.

g) Copy the Microsoft NLB cluster MAC address that you entered in this field for the multicast mode.

Step 4 Click **Submit**.

Step 5 In the **Navigation** pane, choose **Tenant** ***tenant_name*** > **Application Profiles** > ***application_profile_name*** > **Application EPGs** > ***application_EPG_name*** > **Static Ports** > ***static_port*** .

Choose the static port that you want to configure Microsoft NLB to flood onto in the bridge domain.

Step 6 On the **Static Path** page for this port, fill in the following field:

a) In the **NLB Static Group** area, click **+** (Create), then paste the MAC address that you copied from 3.g, on page 268 into the **Mac Address** field.

b) Click **Update** underneath the **Mac Address** field.

Step 7 In the **Static Path** page, click **Submit**.

Any traffic to this Microsoft NLB cluster MAC address will now go out on this static port.

Configuring Microsoft NLB in IGMP Mode Using the GUI

This task configures Microsoft NLB to flood only on certain ports in the bridge domain.

Before you begin

Have the following information available before proceeding with these procedures:

- Microsoft NLB cluster VIP

Procedure

Step 1 In the **Navigation** pane, choose **Tenant** > ***tenant_name*** > **Application Profiles** > ***application_profile_name*** > **Application EPGs** > ***application_EPG_name*** > **Subnets**.

Step 2 Right-click **Subnets** and select **Create EPG Subnet**.

Step 3 In the **Create EPG Subnet** dialog box, fill in the following fields:

a) In the **Default Gateway IP** field, enter the Microsoft NLB cluster VIP.

For example, `192.0.2.1/32`.

b) In the **Scope** area, for shared services, check **Shared between VRFs**.

Uncheck **Private to VRF**, if it is selected.

c) Under **Subnet Control**, check the **No Default SVI Gateway** check box.

d) In the **Type Behind Subnet** area, click **EpNlb**.

The **Mode** field appears.

e) From the **Mode** drop-down list, choose **NLB in IGMP mode**.

The **Group Id** field appears.

f) In the **Group Id** field, enter the Microsoft NLB multicast group address.

For the Microsoft NLB multicast group address, the last two octets of the address correspond to the last two octets of the instance cluster IP address. For example, if the instance cluster IP address is `10.20.30.40`, then the Microsoft NLB multicast group address that you would enter into this field might be `239.255.30.40`.

Step 4 Click **Submit**.

Traffic to the Microsoft NLB cluster VIP will be flooded to the outgoing interface list that is either configured statically from the APIC or dynamically based on IGMP joins from the NLB cluster.

Step 5 Determine if you want to have a static join or a dynamic join.

You can have a combination of static joins and dynamic joins, where some ports can have a static join and other ports can have a dynamic join.

- **Dynamic Join:** In the dynamic join, the join is sent by the Microsoft NLB cluster on the respective ports, then the switch dynamically comes up with that outgoing interface list.
- **Static Join:** In the static join, traffic to the Microsoft NLB cluster VIP will go to the ports that you configure in the following steps.

  If you want to have a static join:

  a. Copy the Microsoft NLB multicast group address that you entered in the **Group Id** field in 3.f, on page 269.

  b. In the **Navigation** pane, choose **Tenant** > ***tenant_name*** > **Application Profiles** > ***application_profile_name*** > **Application EPGs** > ***application_EPG_name*** > **Static Ports** > ***static_port*** .

  Choose the static port that you want to configure Microsoft NLB to flood onto in the bridge domain.

  c. On the **Static Path** page for this port, fill in the following field:

  - In the **IGMP Snoop Static Group** area, click **+** (Create), then paste the MAC address that you copied from 3.f, on page 269 into the **Group Address** field.
  - Click **Update** underneath the **Group Address** field.

  d. In the **Static Path** page, click **Submit**.

IGMP snooping is enabled by default on the bridge domain because the IGMP snooping policy default that is associated with the bridge domain has **Enabled** as the administrative state of the policy. For more information, see Configuring an IGMP Snooping Policy Using the GUI, on page 281.

Configuring Microsoft NLB Using the REST API

Configuring Microsoft NLB in Unicast Mode Using the REST API

Procedure

To configure Microsoft NLB in unicast mode, send a post with XML such as the following example:

Example:

```
https://apic-ip-address/api/node/mo/uni/.xml
<polUni>
    <fvTenant name="tn2" >
        <fvCtx name="ctx1"/>
        <fvBD name="bd2">
            <fvRsCtx tnFvCtxName="ctx1" />
        </fvBD>
        <fvAp name = "ap1">
```

```
            <fvAEPg name = "ep1">
                <fvRsBd tnFvBDName = "bd2"/>
                <fvSubnet ip="10.0.1.1/32" scope="public" ctrl="no-default-gateway">
                    <fvEpNlb mac="12:21:21:35" mode="mode-uc"/>
                </fvSubnet>
            </fvAEPg>
        </fvAp>
    </fvTenant>
</polUni>
```

Configuring Microsoft NLB in Multicast Mode Using the REST API

Procedure

To configure Microsoft NLB in multicast mode, send a post with XML such as the following example:

Example:

```
https://apic-ip-address/api/node/mo/uni/.xml
<polUni>
    <fvTenant name="tn2" >
        <fvCtx name="ctx1"/>
        <fvBD name="bd2">
            <fvRsCtx tnFvCtxName="ctx1" />
        </fvBD>
        <fvAp name = "ap1">
            <fvAEPg name = "ep1">
                <fvRsBd tnFvBDName = "bd2"/>
                <fvSubnet ip="2001:0db8:85a3:0000:0000:8a2e:0370:7344/128" scope="public"
ctrl="no-default-gateway">
                    <fvEpNlb mac="03:21:21:35" mode="mode-mcast--static"/>
                </fvSubnet>
              <fvRsPathAtt tDn="topology/pod-1/paths-101/pathep-[eth1/6]" encap="vlan-911"
 >
                    <fvNlbStaticGroup mac = "03:21:21:35"  />
                </fvRsPathAtt>
            </fvAEPg>
        </fvAp>
    </fvTenant>
</polUni>
```

Configuring Microsoft NLB in IGMP Mode Using the REST API

Procedure

To configure Microsoft NLB in IGMP mode, send a post with XML such as the following example:

Example:

```
https://apic-ip-address/api/node/mo/uni/.xml
<polUni>
    <fvTenant name="tn2" >
        <fvCtx name="ctx1"/>
        <fvBD name="bd2">
            <fvRsCtx tnFvCtxName="ctx1" />
        </fvBD>
        <fvAp name = "ap1">
            <fvAEPg name = "ep1">
                <fvRsBd tnFvBDName = "bd2"/>
                <fvSubnet ip="10.0.1.3/32" scope="public" ctrl="no-default-gateway">
                    <fvEpNlb group ="224.132.18.17" mode="mode-mcast-igmp" />
                </fvSubnet>
            </fvAEPg>
        </fvAp>
    </fvTenant>
</polUni>
```

Configuring Microsoft NLB Using the NX-OS Style CLI

Configuring Microsoft NLB in Unicast Mode Using the NX-OS Style CLI

This task configures Microsoft NLB to flood all of the ports in the bridge domain.

Before you begin

Have the following information available before proceeding with these procedures:

- Microsoft NLB cluster VIP
- Microsoft NLB cluster MAC address

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `apic1# configure`	Enters configuration mode.
Step 2	**tenant** *tenant-name* **Example:** `apic1 (config)# tenant tenant1`	Creates a tenant if it does not exist or enters tenant configuration mode.
Step 3	**application** *app-profile-name* **Example:** `apic1 (config-tenant)# application app1`	Creates an application profile if it doesn't exist or enters application profile configuration mode.

	Command or Action	Purpose
Step 4	**epg** *epg-name* **Example:** `apic1 (config-tenant-app)# epg epg1`	Creates an EPG if it doesn't exist or enters EPG configuration mode.
Step 5	**[no] endpoint {ip | ipv6}** *ip-address* **epnlb mode mode-uc mac** *mac-address* **Example:** `apic1 (config-tenant-app-epg)# endpoint ip 192.0.2.2/32 epnlb mode mode-uc mac 03:BF:01:02:03:04`	Configures Microsoft NLB in unicast mode, where: • *ip-address* is the Microsoft NLB cluster VIP. • *mac-address* is the Microsoft NLB cluster MAC address.

Configuring Microsoft NLB in Multicast Mode Using the NX-OS Style CLI

This task configures Microsoft NLB to flood only on certain ports in the bridge domain.

Before you begin

Have the following information available before proceeding with these procedures:

- Microsoft NLB cluster VIP
- Microsoft NLB cluster MAC address

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `apic1# configure`	Enters configuration mode.
Step 2	**tenant** *tenant-name* **Example:** `apic1 (config)# tenant tenant1`	Creates a tenant if it does not exist or enters tenant configuration mode.
Step 3	**application** *app-profile-name* **Example:** `apic1 (config-tenant)# application app1`	Creates an application profile if it doesn't exist or enters application profile configuration mode.
Step 4	**epg** *epg-name* **Example:** `apic1 (config-tenant-app)# epg epg1`	Creates an EPG if it does not exist or enters EPG configuration mode.
Step 5	**[no] endpoint {ip | ipv6}** *ip-address* **epnlb mode mode-mcast--static mac** *mac-address*	Configures Microsoft NLB in static multicast mode, where:

	Command or Action	Purpose
	Example: `apic1 (config-tenant-app-epg)# endpoint ip 192.0.2.2/32 epnlb mode mode-mcast--static mac 03:BF:01:02:03:04`	• *ip-address* is the Microsoft NLB cluster VIP. • *mac-address* is the Microsoft NLB cluster MAC address.
Step 6	**[no] nlb static-group** *mac-address* **leaf** *leaf-num* **interface {ethernet** *slot/port* **| port-channel** *port-channel-name***} vlan** *portEncapVlan* **Example:** `apic1 (config-tenant-app-epg)# nlb static-group 03:BF:01:02:03:04 leaf 102 interface ethernet 1/12 vlan 19`	Adds Microsoft NLB multicast VMAC to the EPG ports where the Microsoft NLB servers are connected, where: • *mac-address* is the Microsoft NLB cluster MAC address that you entered in Step 5, on page 273. • *leaf-num* is the leaf switch that contains the interface to be added or removed. • *port-channel-name* is the name of the port-channel, when the port-channel option is used. • *portEncapVlan* is the encapsulation VLAN for the static member of the application EPG.

Configuring Microsoft NLB in IGMP Mode Using the NX-OS Style CLI

This task configures Microsoft NLB to flood only on certain ports in the bridge domain.

Before you begin

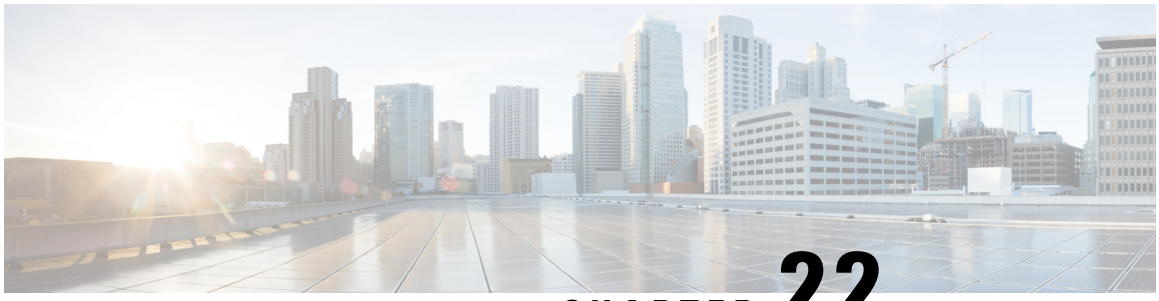
Have the following information available before proceeding with these procedures:

- Microsoft NLB cluster VIP
- Microsoft NLB cluster MAC address

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `apic1# configure`	Enters configuration mode.
Step 2	**tenant** *tenant-name* **Example:** `apic1 (config)# tenant tenant1`	Creates a tenant if it does not exist or enters tenant configuration mode.

	Command or Action	Purpose
Step 3	**application** *app-profile-name* **Example:** `apic1 (config-tenant)# application app1`	Creates an application profile if it doesn't exist or enters application profile configuration mode.
Step 4	**epg** *epg-name* **Example:** `apic1 (config-tenant-app)# epg epg1`	Creates an EPG if it doesn't exist or enters EPG configuration mode.
Step 5	**[no] endpoint {ip | ipv6}** *ip-address* **epnlb mode mode-mcast-igmp group** *multicast-IP-address* **Example:** `apic1 (config-tenant-app-epg)# endpoint ip 192.0.2.2/32 epnlb mode mode-mcast-igmp group 1.3.5.7`	Configures Microsoft NLB in IGMP mode, where: • *ip-address* is the Microsoft NLB cluster VIP. • *multicast-IP-address* is the multicast IP for the NLB endpoint group.



CHAPTER 22

IGMP Snooping

About Cisco APIC and IGMP Snooping

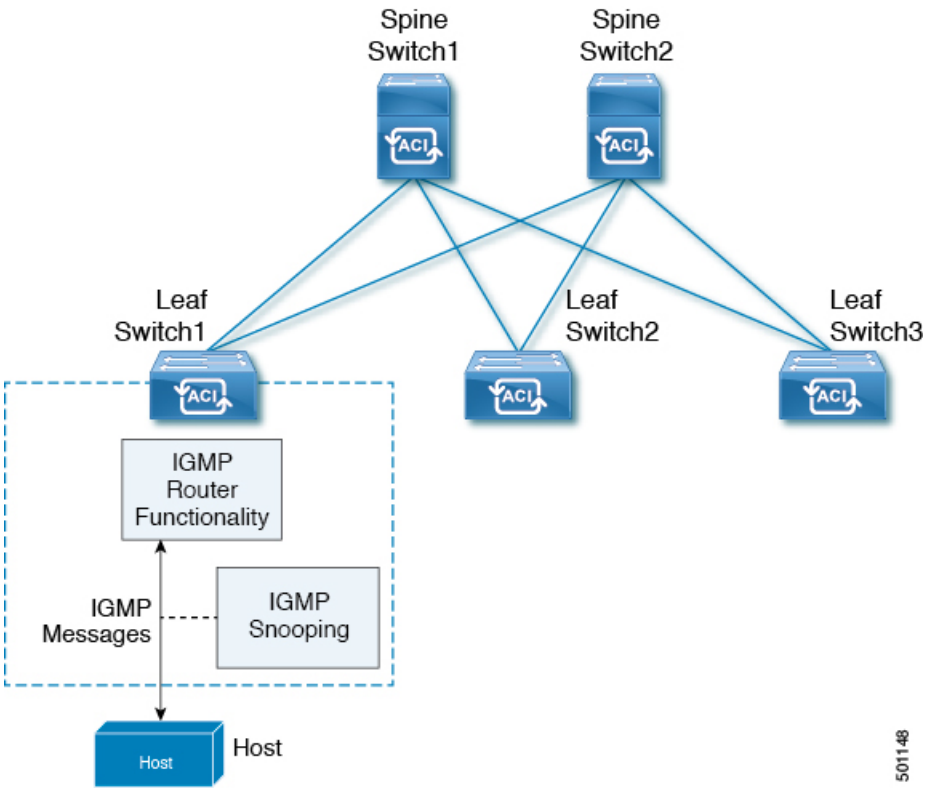
How IGMP Snooping is Implemented in the ACI Fabric

Note 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.

Figure 33: IGMP Snooping 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.

Note 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.

Note 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.
- IGMPv3 snooping will forward multicast based on the group and source entry only when PIM is enabled on the bridge domain. If PIM is not enabled, forwarding will be based on the group only.

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** enable or disable IGMP snooping for this particular policy.

c) Select or unselect **Fast Leave** to enable or disable IGMP V2 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 he IGMP Snooping policy.

Procedure

	Command or Action	Purpose
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 stqrtup-query-count 2 ip igmp snooping startup-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.

	Command or Action	Purpose

	tenant foo template ip igmp snooping policy cookieCut1
	#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 stqrtup-query-count 2
	  ip igmp snooping startup-query-interval 31
	  no description
	  exit
	  exit
	apic1(config-tenant-template-ip-igmp-snooping)#
	  exit
	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 -->
```

```
<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*

- *Deploying an EPG on a Specific Port with APIC Using the NX-OS Style CLI*
- *Deploying an EPG on a Specific Port with APIC Using the REST API*

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

	Command or Action	Purpose

Step 1	On the target interfaces enable IGMP snooping and layer 2 multicasting
	Example:
	`apic1# conf t`
	`apic1(config)# tenant tenant_A`
	`apic1(config-tenant)# application application_A`
	`apic1(config-tenant-app)# epg epg_A`
		The example sequences enable:
		• IGMP snooping on the statically-linked target interface 1/10 and associates it with a multicast IP address, 225.1.1.1
		• IGMP snooping on the statically-linked target interface 1/11 and associates it with a multicast IP address, 227.1.1.1

	Command or Action	Purpose
	apic1(config-tenant-app-epg)# ip igmp snooping static-group 225.1.1.1 leaf 101 interface ethernet 1/10 vlan 305 apic1(config-tenant-app-epg)# end apic1# conf t apic1(config)# tenant tenant_A; application application_A; epg epg_A apic1(config-tenant-app-epg)# ip igmp snooping static-group 227.1.1.1 leaf 101 interface ethernet 1/11 vlan 309 apic1(config-tenant-app-epg)# exit apic1(config-tenant-app)# exit	

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
<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 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 RouteMap Policy** to display the **Create RouteMap 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*** > **Policies** > **Protocol** > **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

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*.

Procedure

	Command or Action	Purpose

Step 1	Define the route-map "access groups."
	Example:
	`apic1# conf t`
	`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`
		The example sequences configure:
		• Route-map-access group "foobroker" linked to multicast group 225.1.1.1/24, access permited
		• Route-map-access group "foobroker" linked to multicast group 227.1.1.1/24, access denied

Step 2	Verify route map configurations.
	Example:
	`apic1(config-tenant)# show running-config tenant test route-map fooBroker`
	`# 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.
	Example:
	`apic1(config-tenant)# application application_A`
	`apic1(config-tenant-app)# epg epg_A`
	`apic1(config-tenant-app-epg)# ip igmp snooping access-group route-map fooBroker leaf 101 interface ethernet 1/10 vlan 305`
		The example sequences configure:
		• Route-map-access group "foobroker" connected through leaf switch 101, interface 1/10, and VLAN 305.
		• Route-map-access group "newbroker" connected through leaf switch 101, interface 1/10, and VLAN 305.

	Command or Action	Purpose
	`apic1(config-tenant-app-epg)# ip igmp snooping access-group route-map newBroker leaf 101 interface ethernet 1/10 vlan 305`	

Step 4	Verify the access group connections.
	Example:
	`apic1(config-tenant-app-epg)# show run`
	`# Command: show running-config tenant tenant_A application application_A epg epg_A`
	`# Time: Mon Aug 29 14:43:02 2016`
	`tenant tenent_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`
	`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.

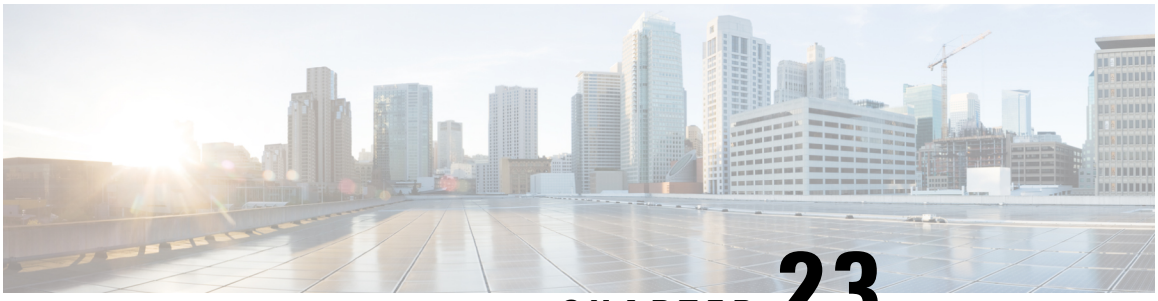
Procedure

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:

```
<!-- 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]"> <!-- IGMP snooping access group case -->
<igmpSnoopAccessGroup name="F23broker"> <igmpRsSnoopAccessGroupFilterRMap
tnPimRouteMapPolName="Rmap_A"/> </igmpSnoopAccessGroup> </fvRsPathAtt> </fvAEPg> </fvAp>
</fvTenant>
```



CHAPTER 23

MLD Snooping

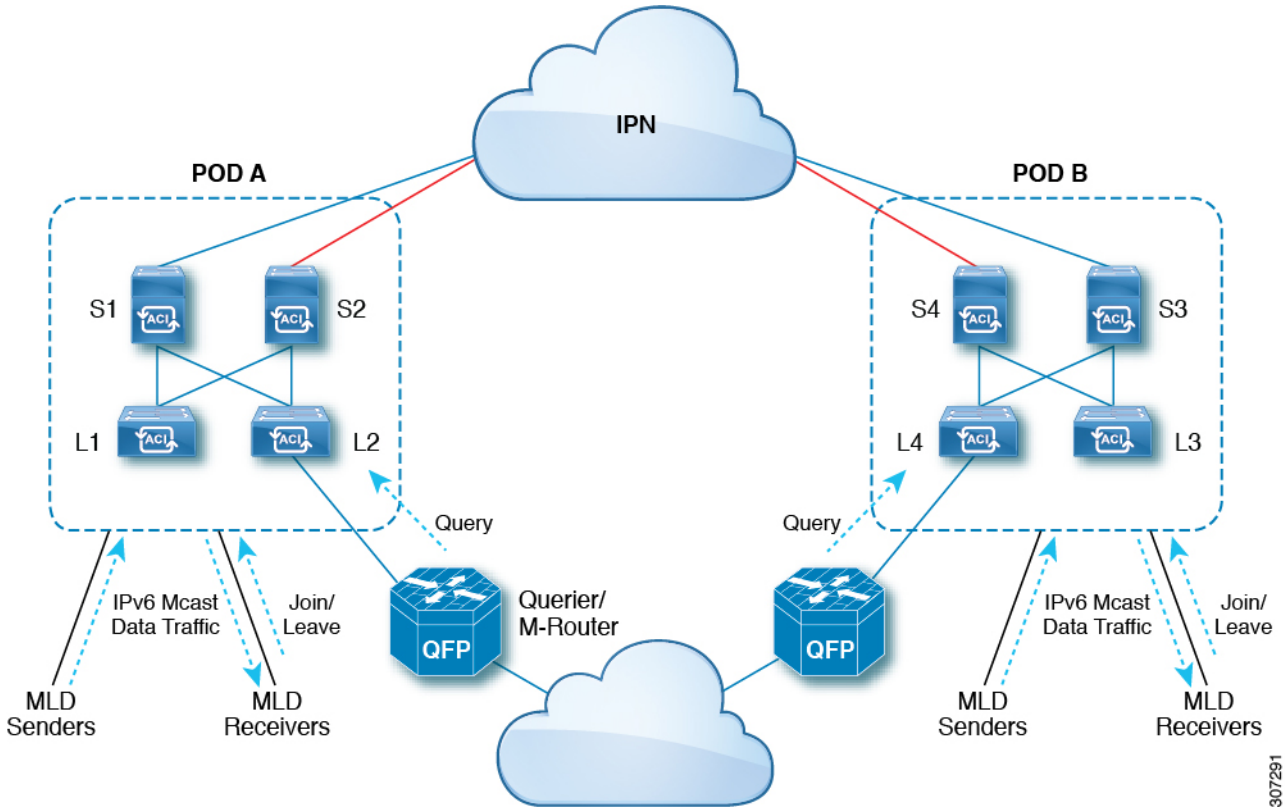
This chapter contains the following sections:

About Cisco APIC and MLD Snooping

Multicast Listener Discovery (MLD) snooping enables the efficient distribution of IPv6 multicast traffic between hosts and routers. It is a Layer 2 feature that restricts IPv6 multicast traffic within a bridge domain to a subset of ports that have transmitted or received MLD queries or reports. In this way, MLD snooping provides the benefit of conserving the bandwidth on those segments of the network where no node has expressed interest in receiving the multicast traffic. This reduces the bandwidth usage instead of flooding the bridge domain, and also helps hosts and routers save unwanted packet processing.

The MLD snooping functionality is similar to IGMP snooping, except that the MLD snooping feature snoops for IPv6 multicast traffic and operates on MLDv1 (RFC 2710) and MLDv2 (RFC 3810) control plane packets. MLD is a sub-protocol of ICMPv6, so MLD message types are a subset of ICMPv6 messages and MLD messages are identified in IPv6 packets by a preceding next header value of 58. Message types in MLDv1 include listener queries, multicast address-specific (MAS) queries, listener reports, and done messages. MLDv2 is designed to be interoperable with MLDv1 except that it has an extra query type, the multicast address and source-specific (MASS) query. The protocol level timers available in MLD are similar to those available in IGMP.

The following figure shows the different components in an MLD snooping arrangement.



Following are explanations of the components in the figure:

- **MLD Senders (sources):** Hosts that send IPv6 traffic into the fabric.
- **MLD Receivers:** Hosts interested in receiving the IPv6 multicast packets. They can choose to join or leave the sessions.
- **Querier/M-Router:** A router or switch that periodically sends queries, and maintains a group membership database. The querier will periodically send queries to determine who might be interested in joining a multicast stream. The M-Router (multicast router) is a gateway to the world outside of the fabric. If there is multicast data traffic inside the fabric, that stream can go outside of the fabric through the multicast router.

When MLD snooping is disabled, then all the multicast traffic is flooded to all the ports, whether they have an interest or not. When MLD snooping is enabled, the fabric will forward IPv6 multicast traffic based on MLD interest. Unknown IPv6 multicast traffic will be flooded based on the bridge domain's IPv6 L3 unknown multicast flood setting.

There are two modes for forwarding unknown IPv6 multicast packets:

- **Flooding mode:** All EPGs and all ports under the bridge domain will get the flooded packets.
- **OMF (Optimized Multicast Flooding) mode:** Only multicast router ports will get the packet.

Guidelines and Limitations

The MLD snooping feature has the following guidelines and limitations:

- MLD snooping is supported only on new generation ToR switches, which are switch models with "EX", "FX" or "FX2" at the end of the switch name.
- Support is enabled for up to 2000 IPv6 multicast groups to be snooped across the fabric.
- Hardware forwarding happens with the (*,G) lookup, even for the source-specific snoop entry with MLDv2.
- The following features are not supported for MLD snooping in this release:
  - Layer 3 multicast routing across bridge domains or VRFs is not supported for IPv6 multicast traffic
  - Static MLD snooping entry
  - Access filter for MLD snoop entries through a route map
  - Virtual endpoints behind the VTEPs (VL)

Configuring and Assigning an MLD Snooping Policy

Configuring and Assigning an MLD Snooping Policy to a Bridge Domain in the GUI

To implement MLD snooping functionality, you configure an MLD snooping policy then assign that policy to one or more bridge domains.

Configuring an MLD Snooping Policy Using the GUI

Create an MLD snooping policy whose MLD snooping 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 MLD snooping support.

Step 2 In the **Navigation** pane, click **Policies** > **Protocol** > **MLD Snoop**.

Step 3 Right-click **MLD Snoop** and select **Create MLD Snoop Policy**.

Step 4 In the **Create MLD 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.

The default entry for this field is **Disabled**.

c) In the **Control** field, select or unselect **Fast Leave** to enable or disable MLD v1 immediate dropping of queries through this policy.

d) In the **Control** field, select or unselect **Enable querier** to enable or disable the MLD 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** > ***bd_subnet*** .

e) Specify in seconds the **Query Interval** value for this policy.

The Query Interval is the interval between general queries sent by the querier. The default entry for this fields is 125 seconds.

f) 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 than the maximum response time. When this timer expires, the host replies with a report.

This is used to control the maximum response time for hosts to answer an MLD query message. Configuring a value less than 10 seconds enables the router to prune groups much faster, but this action results in network burstiness because hosts are restricted to a shorter response time period.

g) Specify in seconds the **Last Member Query Interval** value for this policy.

MLD uses this value when it receives an MLD 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 MLD Fast Leave and, if not, it sends out an out-of-sequence query.

If no reports are received in the interval, the group state is deleted. The software can detect the loss of the last member of a group or source more quickly when the values are smaller. Values range from 1 to 25 seconds. The default is 1 second.

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 MLD Snoop policy is listed in the **Protocol Policies - MLD Snoop** summary page.

What to do next

To put this policy into effect, assign it to any bridge domain.

Assigning an MLD Snooping Policy to a Bridge Domain Using the GUI

Assigning an MLD Snooping policy to a bridge domain configures that bridge domain to use the MLD Snooping properties specified in that policy.

Before you begin

- Configure a bridge domain for a tenant.
- Configure the MLD 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** > ***bd_subnet*** .

Procedure

Step 1 Click the APIC **Tenants** tab and select the name of the tenant whose bridge domains you intend to configure with an MLD 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 MLD Snoop configuration.

Step 3 On the main **Policy** tab, scroll down to the **MLD Snoop Policy** field and select the appropriate MLD policy from the drop-down menu.

Step 4 Click **Submit**.

The target bridge domain is now associated with the specified MLD Snooping policy.

Step 5 To configure the node forwarding parameter for Layer 3 unknown IPv6 Multicast destinations for the bridge domain:

a) Select the bridge domain that you just configured.
b) Click the **Policy** tab, then click the **General** sub-tab.
c) In the **IPv6 L3 Unknown Multicast** field, select either **Flood** or **Optimized Flood**.

Step 6 To change the Link-Local IPv6 address for the switch-querier feature:

a) Select the bridge domain that you just configured.
b) Click the **Policy** tab, then click the **L3 Configurations** sub-tab.
c) In the **Link-local IPv6 Address** field, enter a Link-Local IPv6 address, if necessary.

The default Link-Local IPv6 address for the bridge domain is internally generated. Configure a different Link-Local IPv6 address for the bridge domain in this field, if necessary.

Configuring and Assigning an MLD Snooping Policy to a Bridge Domain using the NX-OS Style CLI

Before you begin

- Create the tenant that will consume the MLD Snooping policy.
- Create the bridge domain for the tenant, where you will attach the MLD Snooping policy.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`apic1# configure terminal`
	`apic1(config)#`
		Enters configuration mode.

Step 2	**tenant** *tenant-name*
	Example:
	`apic1(config)# tenant tn1`
	`apic1(config-tenant)#`
		Creates a tenant or enters tenant configuration mode.

Step 3	**template ipv6 mld snooping policy** *policy-name*
	Example:
	`apic1(config-tenant)# template ipv6 mld snooping policy mldPolicy1`
	`apic1(config-tenant-template-ip-mld-snooping)#`
		Creates an MLD snooping policy. The example NX-OS style CLI sequence creates an MLD snooping policy named mldPolicy1.

Step 4	**[no] ipv6 mld snooping**
	Example:
	`apic1(config-tenant-template-ip-mld-snooping)# ipv6 mld snooping`
	`apic1(config-tenant-template-ip-mld-snooping)# no ipv6 mld snooping`
		Enables or disables the admin state of the MLD snoop policy. The default state is disabled.

Step 5	**[no] ipv6 mld snooping fast-leave**
	Example:
	`apic1(config-tenant-template-ip-mld-snooping)# ipv6 mld snooping fast-leave`
	`apic1(config-tenant-template-ip-mld-snooping)# no ipv6 mld snooping fast-leave`
		Enables or disables IPv6 MLD snooping fast-leave processing.

Step 6	**[no] ipv6 mld snooping querier** **Example:**	Enables or disables IPv6 MLD snooping querier processing. For the enabling querier

	Command or Action	Purpose
	apic1(config-tenant-template-ip-mld-snooping)# **ipv6 mld snooping querier** apic1(config-tenant-template-ip-mld-snooping)# **no ipv6 mld snooping querier**	option to be effectively enabled on the assigned policy, you must also enable the querier option in the subnets assigned to the bridge domains to which the policy is applied, as described in Step 14, on page 302.
Step 7	**ipv6 mld snooping last-member-query-interval** *parameter* **Example:** apic1(config-tenant-template-ip-mld-snooping)# **ipv6 mld snooping last-member-query-interval 25**	Changes the IPv6 MLD snooping last member query interval parameter. The example NX-OS style CLI sequence changes the IPv6 MLD snooping last member query interval parameter to 25 seconds. Valid options are 1-25. The default is 1 second.
Step 8	**ipv6 mld snooping query-interval** *parameter* **Example:** apic1(config-tenant-template-ip-mld-snooping)# **ipv6 mld snooping query-interval 300**	Changes the IPv6 MLD snooping query interval parameter. The example NX-OS style CLI sequence changes the IPv6 MLD snooping query interval parameter to 300 seconds. Valid options are 1-18000. The default is 125 seconds.
Step 9	**ipv6 mld snooping query-max-response-time** *parameter* **Example:** apic1(config-tenant-template-ip-mld-snooping)# **ipv6 mld snooping query-max-response-time 25**	Changes the IPv6 MLD snooping max query response time. The example NX-OS style CLI sequence changes the IPv6 MLD snooping max query response time to 25 seconds. Valid options are 1-25. The default is 10 seconds.
Step 10	**ipv6 mld snooping startup-query-count** *parameter* **Example:** apic1(config-tenant-template-ip-mld-snooping)# **ipv6 mld snooping startup-query-count 10**	Changes the IPv6 MLD snooping number of initial queries to send. The example NX-OS style CLI sequence changes the IPv6 MLD snooping number of initial queries to send to 10. Valid options are 1-10. The default is 2.
Step 11	**ipv6 mld snooping startup-query-interval** *parameter* **Example:** apic1(config-tenant-template-ip-mld-snooping)# **ipv6 mld snooping startup-query-interval 300**	Changes the IPv6 MLD snooping time for sending initial queries. The example NX-OS style CLI sequence changes the IPv6 MLD snooping time for sending initial queries to 300 seconds. Valid options are 1-18000. The default is 31 seconds.
Step 12	**exit** **Example:** apic1(config-tenant-template-ip-mld-snooping)#	Returns to configure mode.

	Command or Action	Purpose
	exit apic1(config-tenant)#	
Step 13	**interface bridge-domain** *bridge-domain-name* **Example:** apic1(config-tenant)# **interface bridge-domain bd1** apic1(config-tenant-interface)#	Configures the interface bridge-domain. The example NX-OS style CLI sequence configures the interface bridge-domain named bd1.
Step 14	**ipv6 address** *sub-bits/prefix-length* **snooping-querier** **Example:** apic1(config-tenant-interface)# **ipv6 address 2000::5/64 snooping-querier**	Configures the bridge domain as switch-querier. This will enable the querier option in the subnet assigned to the bridge domain where the policy is applied.
Step 15	**ipv6 mld snooping policy** *policy-name* **Example:** apic1(config-tenant-interface)# **ipv6 mld snooping policy mldPolicy1**	Associates the bridge domain with an MLD snooping policy. The example NX-OS style CLI sequence associates the bridge domain with an MLD snooping policy named mldPolicy1.
Step 16	**exit** **Example:** apic1(config-tenant-interface)# **exit** apic1(config-tenant)#	Returns to configure mode.

Configuring and Assigning an MLD Snooping Policy to a Bridge Domain using the REST API

Procedure

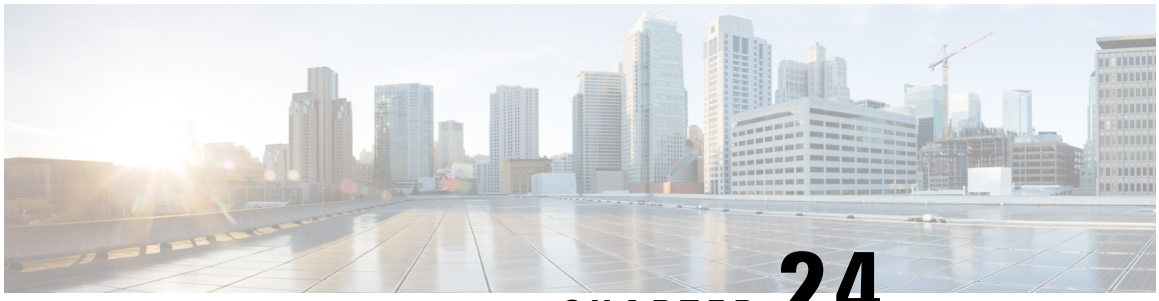
To configure an MLD Snooping policy and assign it to a bridge domain, send a post with XML such as the following example:

Example:

```
https://apic-ip-address/api/node/mo/uni/.xml
<fvTenant name="mldsn">
    <mldSnoopPol adminSt="enabled" ctrl="fast-leave,querier"
name="mldsn-it-fabric-querier-policy" queryIntvl="125"
        rspIntvl="10" startQueryCnt="2" startQueryIntvl="31" status=""/>
    <fvBD name="mldsn-bd3">
        <fvRsMldsn status="" tnMldSnoopPolName="mldsn-it-policy"/>
    </fvBD>
</fvTenant>
```

This example creates and configures the MLD Snooping policy `mldsn` with the following properties, and binds the MLD policy `mldsn-it-fabric-querier-policy` to bridge domain `mldsn-bd3`:

- Fast leave processing is enabled
- Querier processing is enabled
- Query Interval is set at 125
- Max query response time is set at 10
- Number of initial queries to send is set at 2
- Time for sending initial queries is set at 31



CHAPTER 24

HSRP

This chapter contains the following sections:

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.

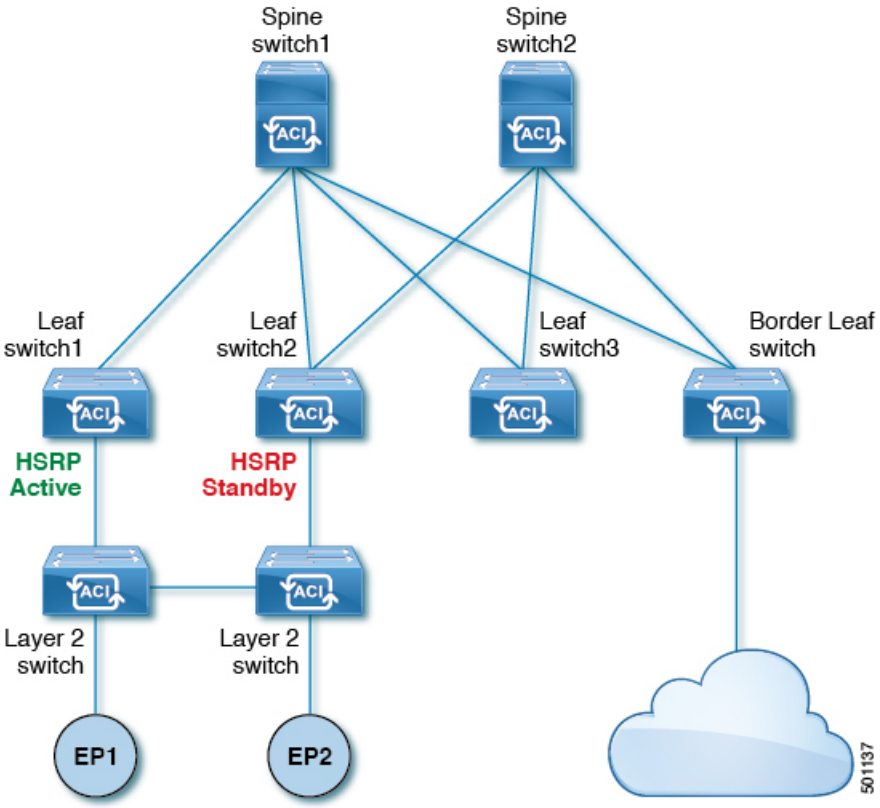
Note Packets received on a routed port destined for the HSRP virtual IP address terminate on the local router, regardless of whether that router is the active HSRP router or the standby HSRP router. This process includes ping and Telnet traffic. Packets received on a Layer 2 (VLAN) interface destined 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.

Figure 34: HSRP Deployment Topology



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

Parameters	Default Value
Version	1
Delay	0
Reload Delay	0
Interface Control	No Use-Burned-in Address (BIA)
Group ID	0
Group Af	IPv4
IP Obtain Mode	admin
Priority	100
Hello Interval	3000 msec
Hold Interval	10000 msec
Group Control	Preemption disabled
Preempt Delay	0
Authentication Type	Plain Text
Authentication Key Timeout	0
VMAC	Derived (from HSRP groupID)

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**.

Step 7 In the **Create HSRP Interface Profile** dialog box, click **Submit**.
This completes the HSRP configuration.

Step 8 To verify the HSRP interface and group policies created, in the Navigation pane, click **Networking** > **Protocol Policies** > **HSRP**.

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

	Command or Action	Purpose
Step 1	**configure** **Example:** `apic1# configure`	Enters configuration mode.

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-if-hsrp)# ip 182.16.1.2`
	`apic1(config-if-hsrp)# ip 182.16.1.3 secondary`
	`apic1(config-if-hsrp)# ip 182.16.1.4 secondary`
	`apic1(config-if-hsrp)# mac-address 5000.1000.1060`
	`apic1(config-if-hsrp)# timers 5 18`
	`apic1(config-if-hsrp)# priority 100`	

	Command or Action	Purpose
	apic1(config-if-hsrp)# **preempt** apic1(config-if-hsrp)# **preempt delay minimum 60** apic1(config-if-hsrp)# **preempt delay reload 60** apic1(config-if-hsrp)# **preempt delay sync 60** apic1(config-if-hsrp)# **authentication none** apic1(config-if-hsrp)# **authentication simple** apic1(config-if-hsrp)# **authentication md5** apic1(config-if-hsrp)# **authentication-key <mypassword>** apic1(config-if-hsrp)# **authentication-key-timeout <timeout>**	

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

	Command or Action	Purpose
Step 1	**configure** **Example:** apic1# **configure**	Enters configuration mode.
Step 2	Configure HSRP policy templates. **Example:** apic1(config)# **leaf 101** apic1(config-leaf)# **template hsrp interface-policy hsrp-intfPol1 tenant t9** apic1(config-template-hsrp-if-pol)# **hsrp use-bia** apic1(config-template-hsrp-if-pol)# **hsrp**	

	Command or Action	Purpose

	`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-if-hsrp)# ip 182.16.1.2`
	`apic1(config-if-hsrp)# ip 182.16.1.3 secondary`
	`apic1(config-if-hsrp)# ip 182.16.1.4 secondary`
	`apic1(config-if-hsrp)# mac-address 5000.1000.1060`
	`apic1(config-if-hsrp)# 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:

```
<polUni>
  <infraInfra dn="uni/infra">
    <infraNodeP name="TenantNode_101">
      <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>
        <infraRsAccBaseGrp tDn="uni/infra/funcprof/accportgrp-TenantPortGrp_101"/>
      </infraHPortS>
    </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:

```
<polUni>
  <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 tnFvCtxName="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=""
scope="import-security"/>
      </l3extInstP>
    </l3extOut>
  </fvTenant>
</polUni>
```

Step 3 Create an HSRP interface policy.

Example:

```
<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.

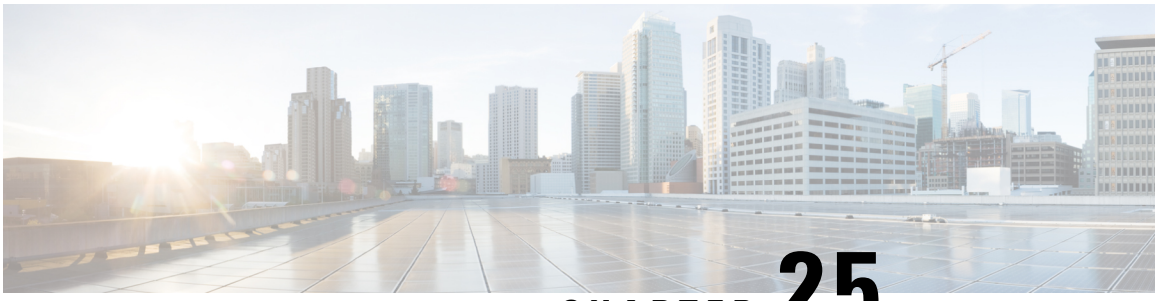
Example:

```
<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:

```
<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>
         <l3extRsPathL3OutAtt addr="2002::100/64" descr="" encap="unknown" encapScope="local"
 ifInstT="l3-port" llAddr="::" 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>
        </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>
          <l3extRsPathL3OutAtt addr="177.21.21.11/24" descr="" encap="unknown"
encapScope="local" ifInstT="l3-port" llAddr="::" mac="00:22:BD:F8:19:FF" mode="regular"
mtu="inherit" tDn="topology/pod-1/paths-101/pathep-[eth1/41]" targetDscp="unspecified">
            <l3extIp addr="177.21.23.11/24"/>
          </l3extRsPathL3OutAtt>
        </l3extLIfP>
      </l3extLNodeP>
    </l3extOut>
  </fvTenant>
</polUni>
```



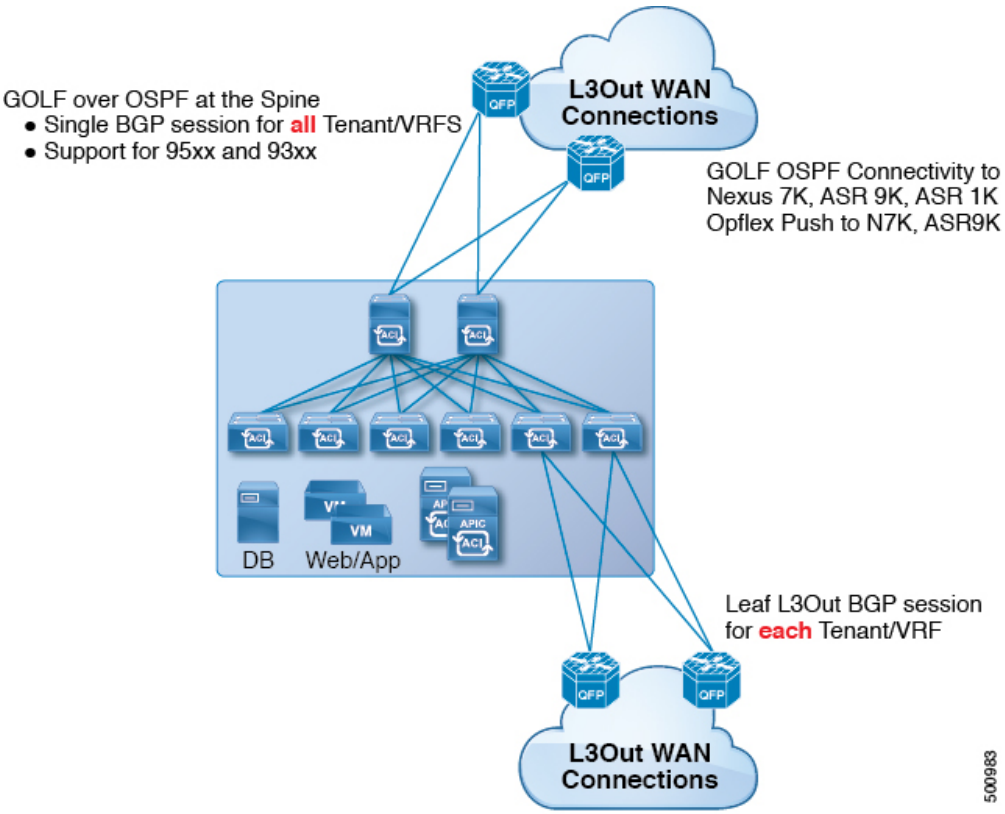
CHAPTER 25

Cisco ACI GOLF

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.

Figure 35: Cisco ACI GOLF Topology



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:

- `LNodeP` (`l3extInstP` is not required within the L3Out in the infra tenant.)
- A provider label for the `L3extOut` for GOLF in the infra tenant.
- OSPF protocol policies
- BGP protocol policies

All regular tenants use the above-defined physical connectivity. The `L3extOut` defined in regular tenants requires 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 for Cisco ACI GOLF

Observe the following Cisco ACI GOLF guidelines and limitations:

- GOLF does not support shared services.
- GOLF does not support transit routing.
- GOLF routers must advertise at least one route to Cisco Application Centric Infrastructure (ACI) 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 Cisco ACI-mode switches and all of the Cisco Nexus 9500 platform Cisco 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 Cisco APIC release 2.0(2), GOLF is not supported with Cisco ACI Multi-Pod. In release 2.0 (2), the two features are supported in the same fabric only over Cisco Nexus 9000 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 Cisco ACI Multi-Pod 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 `L3extOut`s 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, Cisco APIC will export all routes. This is the same as existing `L3extOut` on leaf switches for tenants.
- If you have an ERSPAN session that has a SPAN destination in a VRF instance, the VRF instance has GOLF enabled, and the ERSPAN source has interfaces on a spine switch, the transit prefix gets sent from a non-GOLF L3Out to the GOLF router with the wrong BGP next-hop.
- 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 Cisco 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 Cisco ACI WAN Interconnect peers, increase this as needed.
- In a deployment scenario where there are two `L3extOut`s 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 instance 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 GOLF OpFlex VRF instances, the leaking of routes cannot occur in the Cisco ACI fabric or on the GOLF device between the GOLF OpFlex VRF instance and any other VRF instance 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 GOLF Connections Between Multi-Site Sites

APIC GOLF Connections Shared by Multi-Site Sites

For APIC Sites in a Multi-Site topology, if stretched VRFs share GOLF 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 GOLF VRFs: Manual RT and Auto RT. The route target is synchronized between ACI spines and DCIs through OpFlex. Auto RT for GOLF 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.

Site 1	Site 2
ASN: 100, Fabric ID: 1	ASN: 100, Fabric ID: 1
VRF A: VNID 1000 Import/Export Route Target: 100: [1] 1000	VRF A: VNID 2000 Import/Export Route Target: 100: [1] 2000
VRF B: VNID 2000 Import/Export Route Target: 100: [1] 2000	VRF B: VNID 1000 Import/Export Route Target: 100: [1] 1000

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 GOLF spine in one site towards the DCI should not be sent to GOLF spine in another site. This can happen when the DCI to spine switches have the following types of BGP sessions:

Site1 — IBGP ---- DCI ---- EBGP ---- Site2

Site1 — EBGP ---- DCI ---- IBGP ---- Site2

Site1 — EBGP ---- DCI ---- EBGP ---- Site2

Site1 — IBGP RR client ---- DCI (RR)---- IBGP ---- Site2

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
  neighbor 11.11.11.11 remote-as 1
    update-source loopback0
    address-family l2vpn evpn
      send-community both
      route-map multi-site-in in
  neighbor 13.0.0.2 remote-as 2
    address-family l2vpn evpn
      send-community both
      route-map multi-site-out out
```

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.

Note 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.

Step 3 In the ***infra*** > **Networking** > **External Routed Networks** section of the **Navigation** pane, click to select the Golf policy just created. Enter a **Provider Label**, (for example, *golf*) and click **Submit**.

Step 4 In the **Navigation** pane for any tenant, expand the ***tenant_name*** > **Networking** and perform the following actions:

a) Right-click **External Routed Networks** and click **Create Routed Outside** to open the wizard.
b) In the **Identity** dialog box, in the **Name** field, enter a name for the policy.
c) Complete the configuration options according to the requirements of your Layer 3 connection.

Note In the protocol check boxes area, assure that both **BGP** and **OSPF** are checked. GOLF requires both BGP and OSPF.

d) Assign a **Consumer Label**. In this example, use *golf* (that was just created above.
e) Click **Next**.
f) Configure the External EPG Networks dialog box, and click **Finish** to deploy the policy.

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:

```
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
          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:

```
<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>
         <l3extLIfP descr="" name="portIf-spine1-3"
             ownerKey="" ownerTag="" tag="yellow-green">
```

```
   <ospfIfP authKeyId="1" authType="none" descr="" name="">
     <ospfRsIfPol tnOspfIfPolName="ospfIfPol"/>
   </ospfIfP>
   <l3extRsNdIfPol tnNdIfPolName=""/>
   <l3extRsIngressQosDppPol tnQosDppPolName=""/>
   <l3extRsEgressQosDppPol tnQosDppPolName=""/>
   <l3extRsPathL3OutAtt addr="7.2.1.1/24" descr=""
      encap="vlan-4"
      encapScope="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 descr="" name="portIf-spine2-1"
    ownerKey=""
    ownerTag=""
    tag="yellow-green">
    <ospfIfP authKeyId="1"
         authType="none"
         descr=""
         name="">
         <ospfRsIfPol tnOspfIfPolName="ospfIfPol"/>
    </ospfIfP>
    <l3extRsNdIfPol tnNdIfPolName=""/>
    <l3extRsIngressQosDppPol tnQosDppPolName=""/>
    <l3extRsEgressQosDppPol tnQosDppPolName=""/>
    <l3extRsPathL3OutAtt addr="7.1.0.1/24" descr=""
         encap="vlan-4"
         encapScope="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 descr="" name="portif-spine2-2"
    ownerKey=""
    ownerTag=""
    tag="yellow-green">
    <ospfIfP authKeyId="1"
         authType="none" descr=""
         name="">
         <ospfRsIfPol tnOspfIfPolName="ospfIfPol"/>
   </ospfIfP>
   <l3extRsNdIfPol tnNdIfPolName=""/>
   <l3extRsIngressQosDppPol tnQosDppPolName=""/>
   <l3extRsEgressQosDppPol tnQosDppPolName=""/>
   <l3extRsPathL3OutAtt addr="7.2.2.1/24" descr=""
         encap="vlan-4"
         encapScope="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">
        <ospfIfP authKeyId="1" authType="none" descr="" name="">
```

```
                    <ospfRsIfPol tnOspfIfPolName="ospfIfPol"/>
                </ospfIfP>
                <l3extRsNdIfPol tnNdIfPolName=""/>
                <l3extRsIngressQosDppPol tnQosDppPolName=""/>
                <l3extRsEgressQosDppPol tnQosDppPolName=""/>
                <l3extRsPathL3OutAtt addr="9.0.0.1/24" descr=""
                 encap="vlan-4"
                 encapScope="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">
                  <ospfIfP authKeyId="1" authType="none" descr="" name="">
                       <ospfRsIfPol tnOspfIfPolName="ospfIfPol"/>
                  </ospfIfP>
                  <l3extRsNdIfPol tnNdIfPolName=""/>
                  <l3extRsIngressQosDppPol tnQosDppPolName=""/>
                  <l3extRsEgressQosDppPol tnQosDppPolName=""/>
                  <l3extRsPathL3OutAtt addr="7.0.0.1/24" descr=""
                    encap="vlan-4"
                    encapScope="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"
               privateASctrl="" ttl="2" weight="0">
               <bgpRsPeerPfxPol tnBgpPeerPfxPolName=""/>
               <bgpAsP asn="150" descr="" name="aspn"/>
            </bgpInfraPeerP>
            <bgpInfraPeerP addr="10.10.4.1"
               allowedSelfAsCnt="3"
               ctrl="send-com,send-ext-com" descr="" name="" peerCtrl=""
               peerT="wan"
               privateASctrl="" ttl="1" weight="0">
               <bgpRsPeerPfxPol tnBgpPeerPfxPolName=""/>
               <bgpAsP asn="100" descr="" name=""/>
             </bgpInfraPeerP>
             <bgpInfraPeerP addr="10.10.3.1"
               allowedSelfAsCnt="3"
               ctrl="send-com,send-ext-com" descr="" name="" peerCtrl=""
               peerT="wan"
               privateASctrl="" ttl="1" weight="0">
               <bgpRsPeerPfxPol tnBgpPeerPfxPolName=""/>
               <bgpAsP asn="100" descr="" name=""/>
            </bgpInfraPeerP>
        </l3extLNodeP>
        <bgpRtTargetInstrP descr="" name="" ownerKey="" ownerTag="" rtTargetT="explicit"/>
        <l3extRsL3DomAtt tDn="uni/l3dom-l3dom"/>
        <l3extInstP descr="" matchT="AtleastOne" name="golfInstP"
                 prio="unspecified"
                 targetDscp="unspecified">
```

```
                <fvRsCustQosPol tnQosCustomPolName=""/>
          </l3extInstP>
          <bgpExtP descr=""/>
          <ospfExtP areaCost="1"
                areaCtrl="redistribute,summary"
                areaId="0.0.0.1"
                areaType="regular" descr=""/>
</l3extOut>
```

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:

```
<fvTenant descr="" dn="uni/tn-pep6" name="pep6" ownerKey="" ownerTag="">
     <vzBrCP descr="" name="webCtrct"
          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>
      <vzBrCP descr="" name="webCtrct-pod2"
           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>
       <fvCtx descr="" knwMcastAct="permit"
            name="ctx6" ownerKey="" ownerTag=""
            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>
            <fvRsCtxToExtRouteTagPol tnL3extRouteTagPolName=""/>
            <fvRsBgpCtxPol tnBgpCtxPolName=""/>
            <vzAny descr="" matchT="AtleastOne" name=""/>
            <fvRsOspfCtxPol tnOspfCtxPolName=""/>
            <fvRsCtxToEpRet tnFvEpRetPolName=""/>
            <l3extGlobalCtxName descr="" name="dci-pep6"/>
       </fvCtx>
       <fvBD arpFlood="no" descr="" epMoveDetectMode=""
```

```
     ipLearning="yes"
     limitIpLearnToSubnets="no"
     llAddr="::" mac="00:22:BD:F8:19:FF"
     mcastAllow="no"
     multiDstPktAct="bd-flood"
     name="bd107" ownerKey="" ownerTag="" type="regular"
     unicastRoute="yes"
     unkMacUcastAct="proxy"
     unkMcastAct="flood"
     vmac="not-applicable">
     <fvRsBDToNdP tnNdIfPolName=""/>
     <fvRsBDToOut tnL3extOutName="routAccounting-pod2"/>
     <fvRsCtx tnFvCtxName="ctx6"/>
     <fvRsIgmpsn tnIgmpSnoopPolName=""/>
     <fvSubnet ctrl="" descr="" ip="27.6.1.1/24"
          name="" preferred="no"
          scope="public"
          virtual="no"/>
          <fvSubnet ctrl="nd" descr="" ip="2001:27:6:1::1/64"
               name="" preferred="no"
               scope="public"
               virtual="no">
               <fvRsNdPfxPol tnNdPfxPolName=""/>
          </fvSubnet>
          <fvRsBdToEpRet resolveAct="resolve" tnFvEpRetPolName=""/>
</fvBD>
<fvBD arpFlood="no" descr="" epMoveDetectMode=""
     ipLearning="yes"
     limitIpLearnToSubnets="no"
     llAddr="::" mac="00:22:BD:F8:19:FF"
     mcastAllow="no"
     multiDstPktAct="bd-flood"
     name="bd103" ownerKey="" ownerTag="" type="regular"
     unicastRoute="yes"
     unkMacUcastAct="proxy"
     unkMcastAct="flood"
     vmac="not-applicable">
     <fvRsBDToNdP tnNdIfPolName=""/>
     <fvRsBDToOut tnL3extOutName="routAccounting"/>
     <fvRsCtx tnFvCtxName="ctx6"/>
     <fvRsIgmpsn tnIgmpSnoopPolName=""/>
     <fvSubnet ctrl="" descr="" ip="23.6.1.1/24"
          name="" preferred="no"
          scope="public"
          virtual="no"/>
     <fvSubnet ctrl="nd" descr="" ip="2001:23:6:1::1/64"
          name="" preferred="no"
          scope="public" virtual="no">
          <fvRsNdPfxPol tnNdPfxPolName=""/>
     </fvSubnet>
     <fvRsBdToEpRet resolveAct="resolve" tnFvEpRetPolName=""/>
</fvBD>
<vnsSvcCont/>
<fvRsTenantMonPol tnMonEPGPolName=""/>
<fvAp descr="" name="AP1"
     ownerKey="" ownerTag="" prio="unspecified">
     <fvAEPg descr=""
          isAttrBasedEPg="no"
          matchT="AtleastOne"
          name="epg107"
          pcEnfPref="unenforced" prio="unspecified">
          <fvRsCons prio="unspecified"
               tnVzBrCPName="webCtrct-pod2"/>
          <fvRsPathAtt descr=""
```

```
                    encap="vlan-1256"
                    instrImedcy="immediate"
                    mode="regular" primaryEncap="unknown"
                    tDn="topology/pod-2/paths-107/pathep-[eth1/48]"/>
               <fvRsDomAtt classPref="encap" delimiter=""
                    encap="unknown"
                    instrImedcy="immediate"
                    primaryEncap="unknown"
                    resImedcy="lazy" tDn="uni/phys-phys"/>
               <fvRsCustQosPol tnQosCustomPolName=""/>
               <fvRsBd tnFvBDName="bd107"/>
               <fvRsProv matchT="AtleastOne"
                    prio="unspecified"
                    tnVzBrCPName="default"/>
          </fvAEPg>
          <fvAEPg descr=""
               isAttrBasedEPg="no"
               matchT="AtleastOne"
               name="epg103"
               pcEnfPref="unenforced" prio="unspecified">
               <fvRsCons prio="unspecified" tnVzBrCPName="default"/>
               <fvRsCons prio="unspecified" tnVzBrCPName="webCtrct"/>
               <fvRsPathAtt descr="" encap="vlan-1256"
                    instrImedcy="immediate"
                    mode="regular" primaryEncap="unknown"
                    tDn="topology/pod-1/paths-103/pathep-[eth1/48]"/>
                    <fvRsDomAtt classPref="encap" delimiter=""
                         encap="unknown"
                         instrImedcy="immediate"
                         primaryEncap="unknown"
                         resImedcy="lazy" tDn="uni/phys-phys"/>
                    <fvRsCustQosPol tnQosCustomPolName=""/>
                    <fvRsBd tnFvBDName="bd103"/>
          </fvAEPg>
     </fvAp>
     <l3extOut descr=""
          enforceRtctrl="export"
          name="routAccounting-pod2"
          ownerKey="" ownerTag="" targetDscp="unspecified">
          <l3extRsEctx tnFvCtxName="ctx6"/>
          <l3extInstP descr=""
               matchT="AtleastOne"
               name="accountingInst-pod2"
               prio="unspecified" targetDscp="unspecified">
          <l3extSubnet aggregate="export-rtctrl,import-rtctrl"
               descr="" ip="::/0" name=""
               scope="export-rtctrl,import-rtctrl,import-security"/>
          <l3extSubnet aggregate="export-rtctrl,import-rtctrl"
               descr=""
               ip="0.0.0.0/0" name=""
               scope="export-rtctrl,import-rtctrl,import-security"/>
          <fvRsCustQosPol tnQosCustomPolName=""/>
          <fvRsProv matchT="AtleastOne"
               prio="unspecified" tnVzBrCPName="webCtrct-pod2"/>
          </l3extInstP>
          <l3extConsLbl descr=""
               name="golf2"
               owner="infra"
               ownerKey="" ownerTag="" tag="yellow-green"/>
     </l3extOut>
     <l3extOut descr=""
          enforceRtctrl="export"
          name="routAccounting"
          ownerKey="" ownerTag="" targetDscp="unspecified">
```

```
            <l3extRsEctx tnFvCtxName="ctx6"/>
            <l3extInstP descr=""
                 matchT="AtleastOne"
                 name="accountingInst"
                 prio="unspecified" targetDscp="unspecified">
            <l3extSubnet aggregate="export-rtctrl,import-rtctrl" descr=""
                 ip="0.0.0.0/0" name=""
                 scope="export-rtctrl,import-rtctrl,import-security"/>
            <fvRsCustQosPol tnQosCustomPolName=""/>
            <fvRsProv matchT="AtleastOne" prio="unspecified" tnVzBrCPName="webCtrct"/>
            </l3extInstP>
            <l3extConsLbl descr=""
                 name="golf"
                 owner="infra"
                 ownerKey="" ownerTag="" tag="yellow-green"/>
       </l3extOut>
</fvTenant>
```

Distributing BGP EVPN Type-2 Host Routes to a DCIG

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

Step 1 On the menu bar, click **Tenants** > **infra**,

Step 2 In the Navigation pane, expand the **External Routed Networks**, then expand **Protocol Policies** and **BGP**.

Step 3 Right-click **BGP Address Family Context**, select **Create BGP Address Family Context Policy** and perform the following steps:

a) Type a name for the policy and optionally add a description.
b) Click the **Enable Host Route Leak** check box.
c) Click **Submit**.

Step 4 Click **Tenants** > **tenant-name** (for a tenant that will consume the BGP Address Family Context Policy) and expand **Networking**.

Step 5 Expand **VRFs** and click the VRF that will include the host routes you want to distribute.

Step 6 When you configure the VRF properties, add the **BGP Address Family Context Policy** to the **BGP Context Per Address Families** for IPv4 and IPv6.

Step 7 Click **Submit**.

Enabling Distributing BGP EVPN Type-2 Host Routes to a DCIG Using the NX-OS Style CLI

Procedure

	Command or Action	Purpose

Step 1	Configure distributing EVPN type-2 host routes to a DCIG with the following commands in the BGP address family configuration mode.
	Example:
	`apic1(config)# leaf 101`
	`apic1(config-leaf)# template bgp address-family bgpAf1 tenant bgp_t1`
	`apic1(config-bgp-af)# distance 250 240 230`
	`apic1(config-bgp-af)# host-rt-enable`
		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 **no host-rt-enable** command.

	Command or Action	Purpose
	`apic1(config-bgp-af)# exit`	

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:

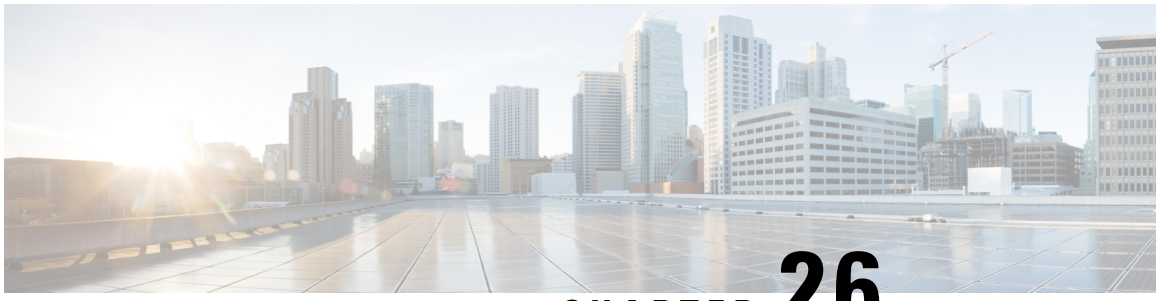
Example:

```
<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:

```
<fvCtx name="vni-10001">
<fvRsCtxToBgpCtxAfPol af="ipv4-ucast" tnBgpCtxAfPolName="bgpCtxPol_0"/>
<fvRsCtxToBgpCtxAfPol af="ipv6-ucast" tnBgpCtxAfPolName="bgpCtxPol_0"/>
</fvCtx>
```



CHAPTER 26

Multi-Pod

This chapter contains the following sections:

About Multi-Pod

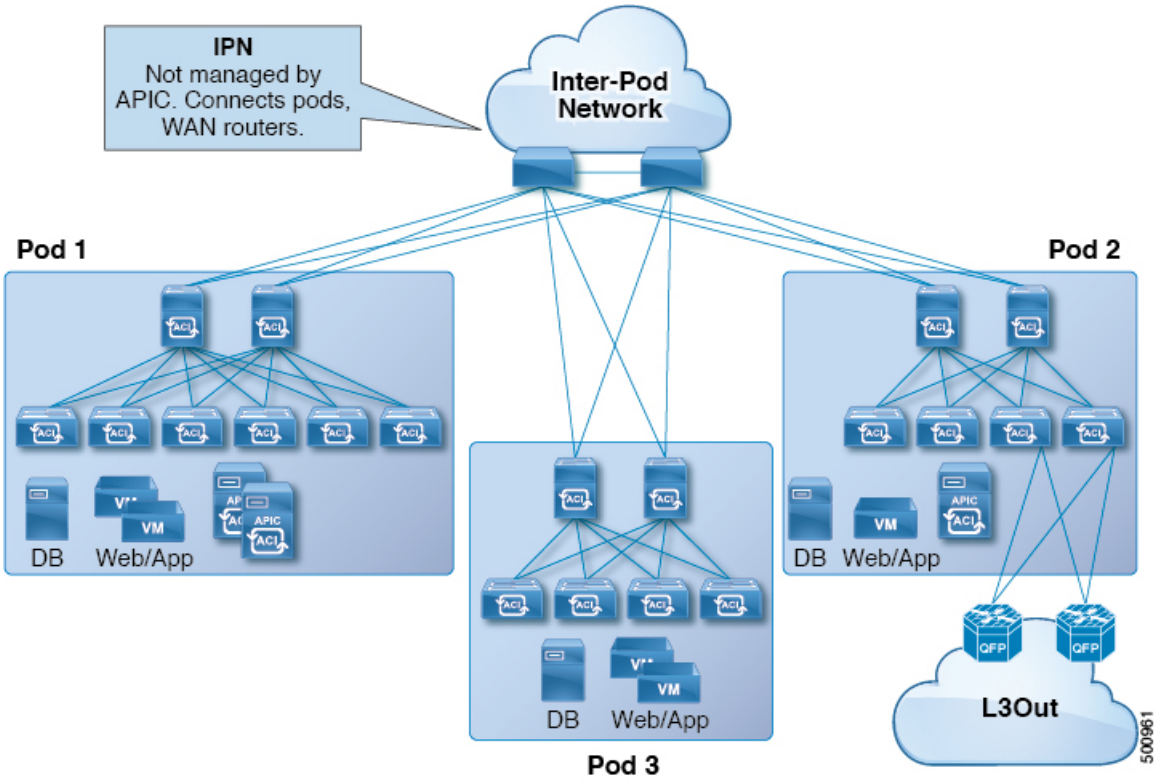
Multi-Pod enables provisioning a more fault-tolerant fabric comprised of multiple pods with isolated control plane protocols. Also, Multi-Pod provides more flexibility with regard to the full mesh cabling between leaf and spine switches. For example, if leaf switches are spread across different floors 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.

Figure 36: Multi-Pod Overview



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).

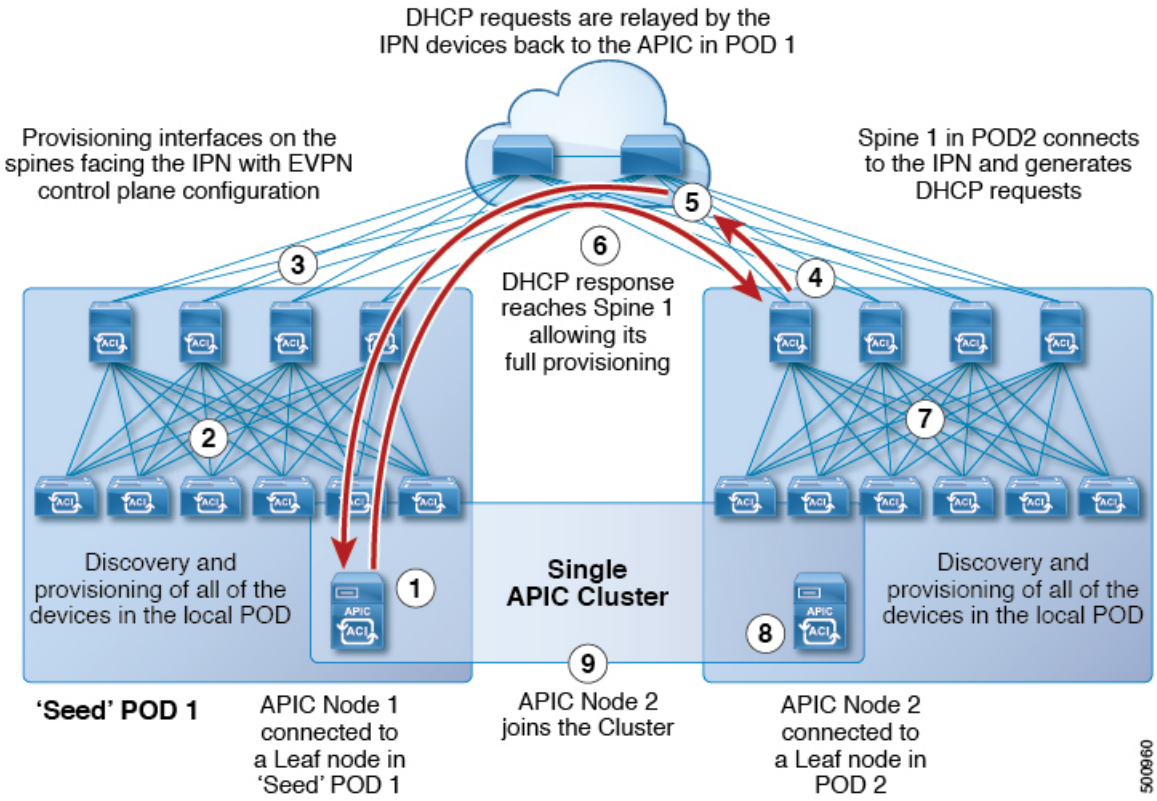
  A group in **bidir** mode has only shared tree forwarding capabilities.

- 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 37: Multi-Pod Provisioning



Guidelines for Setting Up a Multi-Pod Fabric

To configure a Multi-Pod fabric, follow these guidelines:

- Multi-Pod is supported on the following:
  - All ACI-mode spine switches
  - All Cisco Nexus 9000 Series ACI-mode leaf switches
  - All of the Cisco Nexus 9500 platform ACI-mode switch line cards and fabric modules
- 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 350 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 317.
- 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.

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`.

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 344 and Setting Up Multi-Pod Fabric Using the REST API, on page 347 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 Reflectors" 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 342 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 342 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 349 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 340 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 340 in this guide.

Procedure

Step 1 Log in to 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 340 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 340 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:

```
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 SAL1748H56D 201 ifav4-spine1 pod 1
  system switch-id SAL1803L25H 102 ifav4-leaf2 pod 1
  system switch-id SAL1819RXP4 101 ifav4-leaf1 pod 1
  system switch-id SAL1931LA3B 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:

```
ifav4-ifc1# show running-config vlan-domain l3Dom
# Command: show running-config vlan-domain l3Dom
# Time: Mon Aug  1 21:32:31 2016
  vlan-domain l3Dom
    vlan 4
    exit
ifav4-ifc1#
```

Step 3 Configure the fabric external connectivity, as in the following example:

Example:

```
ifav4-ifc1# show running-config fabric-external
# Command: show running-config fabric-external
```

```
# Time: Mon Aug  1 21:34:17 2016
  fabric-external 1
    bgp evpn peering
    pod 1
      interpod data hardware-proxy 100.11.1.1/32
      bgp evpn peering
      exit
    pod 2
      interpod data hardware-proxy 200.11.1.1/32
      bgp evpn peering
      exit
    route-map interpod-import
      ip prefix-list default permit 0.0.0.0/0
      exit
    route-target extended 5:16
    exit
ifav4-ifc1#
```

Step 4 Configure the spine switch interface and OSPF configuration as in the following example:

Example:

```
# 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
    exit
  spine 202
    vrf context tenant infra vrf overlay-1
      router-id 202.202.202.202
      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 loopback 202.202.202.202
        area 1.1.1.1 interpod peering
        exit
      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 router 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 router 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
    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 router 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
    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="SAL1934MNY0" name="ifav4-leaf3" nodeId="103" podId="1"/>
<fabricNodeIdentP serial="SAL1934MNY3" 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="1"/>
<fabricNodeIdentP serial="SAL1938PHBB" name="ifav4-leaf5" nodeId="105" podId="2"/>
<fabricNodeIdentP serial="SAL1942R857" name="ifav4-leaf6" nodeId="106" 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=''/>
      <l3extRsEctx tnFvCtxName="overlay-1"/>
      <l3extProvLbl descr="" name="prov_mp1" 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="" name=""/>
```

```
            <l3extLoopBackIfP addr="201.201.201.201/32" descr="" name=""/>
          </l3extRsNodeL3OutAtt>

          <l3extRsNodeL3OutAtt rtrId="202.202.202.202" rtrIdLoopBack="no"
tDn="topology/pod-1/node-202">
            <l3extInfraNodeP descr="" fabricExtCtrlPeering="yes" name=""/>
            <l3extLoopBackIfP addr="202::202/128" descr="" name=""/>
            <l3extLoopBackIfP addr="202.202.202.202/32" descr="" name=""/>
          </l3extRsNodeL3OutAtt>

          <l3extRsNodeL3OutAtt rtrId="203.203.203.203" rtrIdLoopBack="no"
tDn="topology/pod-2/node-203">
            <l3extInfraNodeP descr="" fabricExtCtrlPeering="yes" name=""/>
            <l3extLoopBackIfP addr="203::203/128" descr="" name=""/>
            <l3extLoopBackIfP addr="203.203.203.203/32" descr="" name=""/>
          </l3extRsNodeL3OutAtt>

          <l3extRsNodeL3OutAtt rtrId="204.204.204.204" rtrIdLoopBack="no"
tDn="topology/pod-2/node-204">
            <l3extInfraNodeP descr="" fabricExtCtrlPeering="yes" name=""/>
            <l3extLoopBackIfP addr="204::204/128" descr="" name=""/>
            <l3extLoopBackIfP addr="204.204.204.204/32" descr="" name=""/>
          </l3extRsNodeL3OutAtt>

          <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" />

            <ospfIfP>
                <ospfRsIfPol tnOspfIfPolName='ospfIfPol'/>
            </ospfIfP>

          </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=''>
       <fvPodConnP descr="" id="1" name="">
          <fvIp addr="100.11.1.1/32"/>
       </fvPodConnP>
       <fvPodConnP descr="" id="2" name="">
          <fvIp addr="200.11.1.1/32"/>
       </fvPodConnP>
      <fvPeeringP descr="" name="" ownerKey="" ownerTag="" type="automatic_with_full_mesh"/>

       <l3extFabricExtRoutingP descr="" name="ext_routing_prof_1" ownerKey="" ownerTag="">
         <l3extSubnet aggregate="" descr="" ip="100.0.0.0/8" name="" scope="import-security"/>
```

```
        <l3extSubnet aggregate="" descr="" ip="200.0.0.0/8" name="" scope="import-security"/>

         <l3extSubnet aggregate="" descr="" ip="201.1.0.0/16" name=""
scope="import-security"/>
         <l3extSubnet aggregate="" descr="" ip="201.2.0.0/16" name=""
scope="import-security"/>
         <l3extSubnet aggregate="" descr="" ip="202.1.0.0/16" name=""
scope="import-security"/>
         <l3extSubnet aggregate="" descr="" ip="203.1.0.0/16" name=""
scope="import-security"/>
         <l3extSubnet aggregate="" descr="" ip="203.2.0.0/16" name=""
scope="import-security"/>
         <l3extSubnet aggregate="" descr="" ip="204.1.0.0/16" name=""
scope="import-security"/>
      </l3extFabricExtRoutingP>
   </fvFabricExtConnP>
</fvTenant>
</polUni>
```

Sample IPN Configuration for Multi-Pod For Cisco Nexus 9000 Series Switches

Note

- 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)


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
```

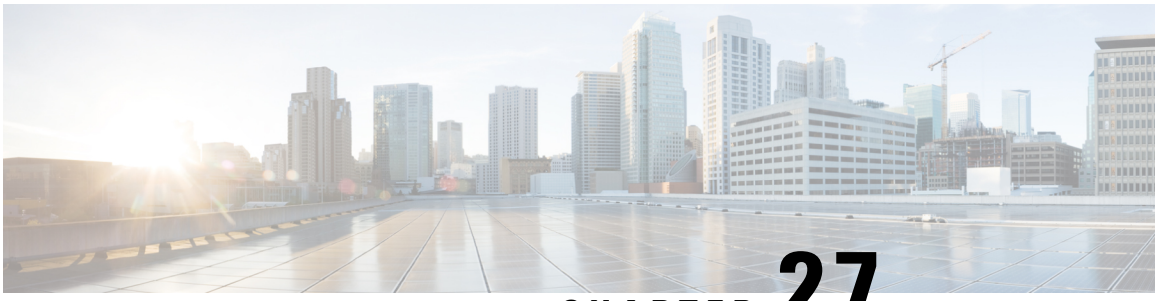
Step 4 In the APIC setup script, specify the pod ID where the APIC node has been moved.

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**.



CHAPTER 27

Remote Leaf Switches

This chapter contains the following sections:

About Remote Leaf Switches in the ACI Fabric

With an ACI fabric deployed, you can extend ACI services and APIC management to remote data centers 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 data center 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 Layer 3 multicast (bidirectional PIM) over the WAN. Any traffic that requires use of the spine switch proxy is forwarded to the main data center.

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.
- For releases prior to Release 4.1(2), before decommissioning a remote leaf switch, you must first delete the vPC.

Characteristics of Remote Leaf Switch Behavior in Release 4.0(1)

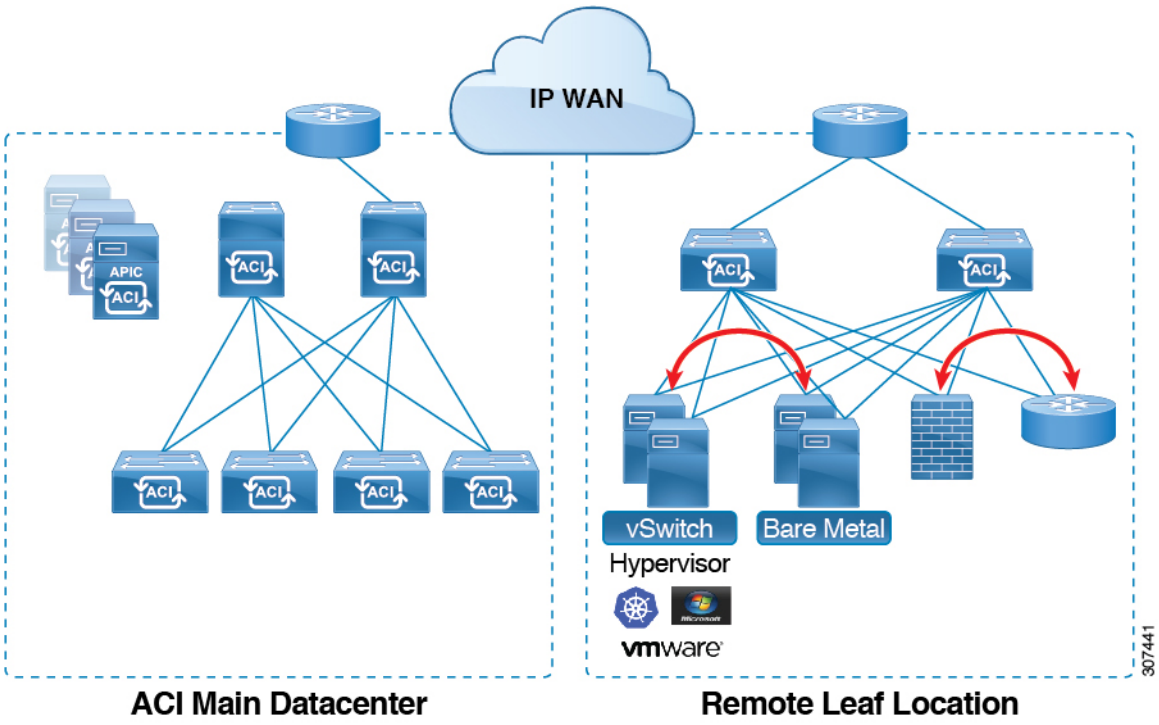
Starting in Release 4.0(1), Remote Leaf behavior takes on the following characteristics:

- 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 through 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 location when spine-proxy is not reachable.

Characteristics of Remote Leaf Switch Behavior in Release 4.1(2)

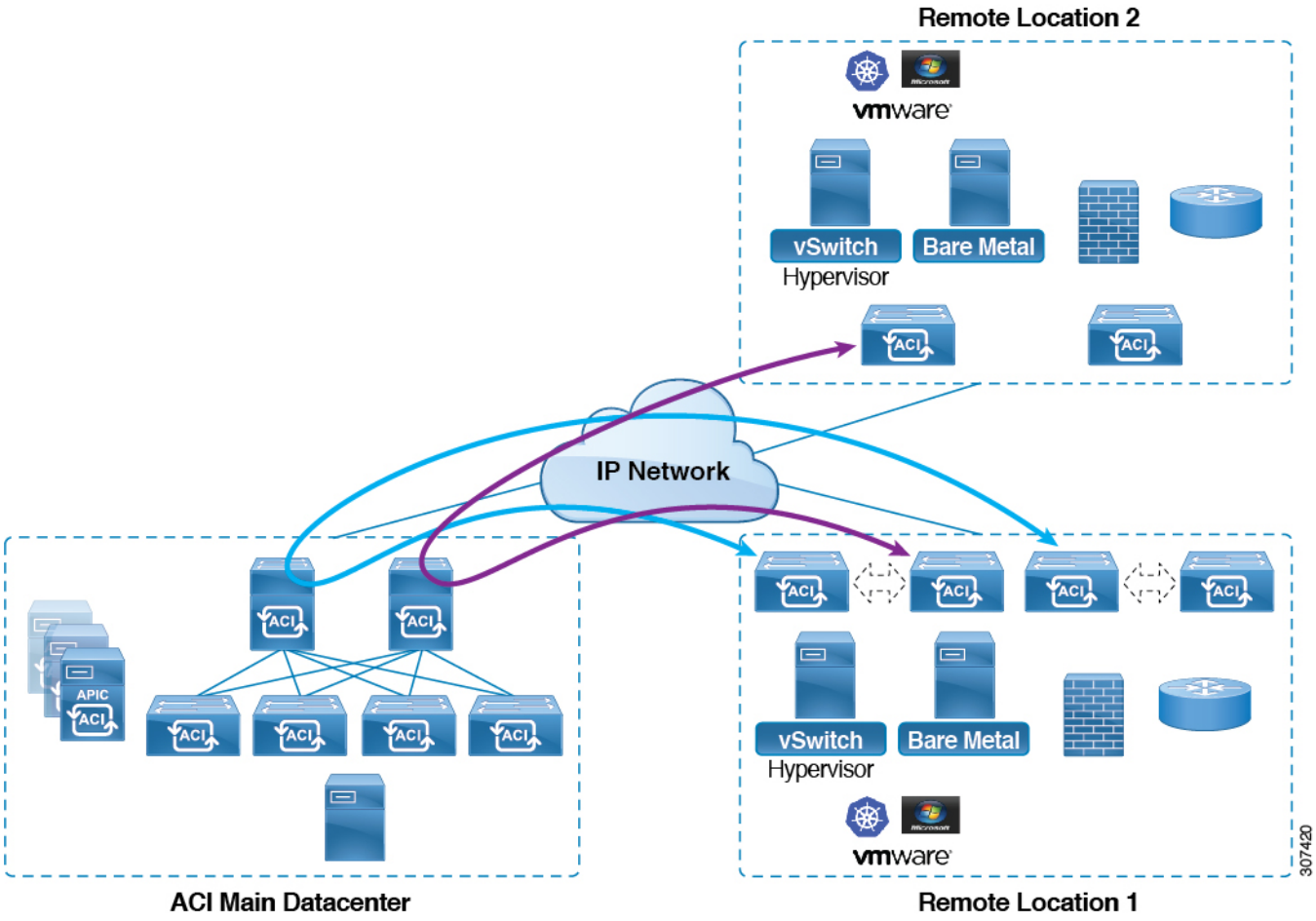
Before Release 4.1(2), all local switching (within the remote leaf vPC peer) traffic on the remote leaf location is switched directly between endpoints, whether physical or virtual, as shown in the following figure.

Figure 38: Local Switching Traffic: Prior to Release 4.1(2)



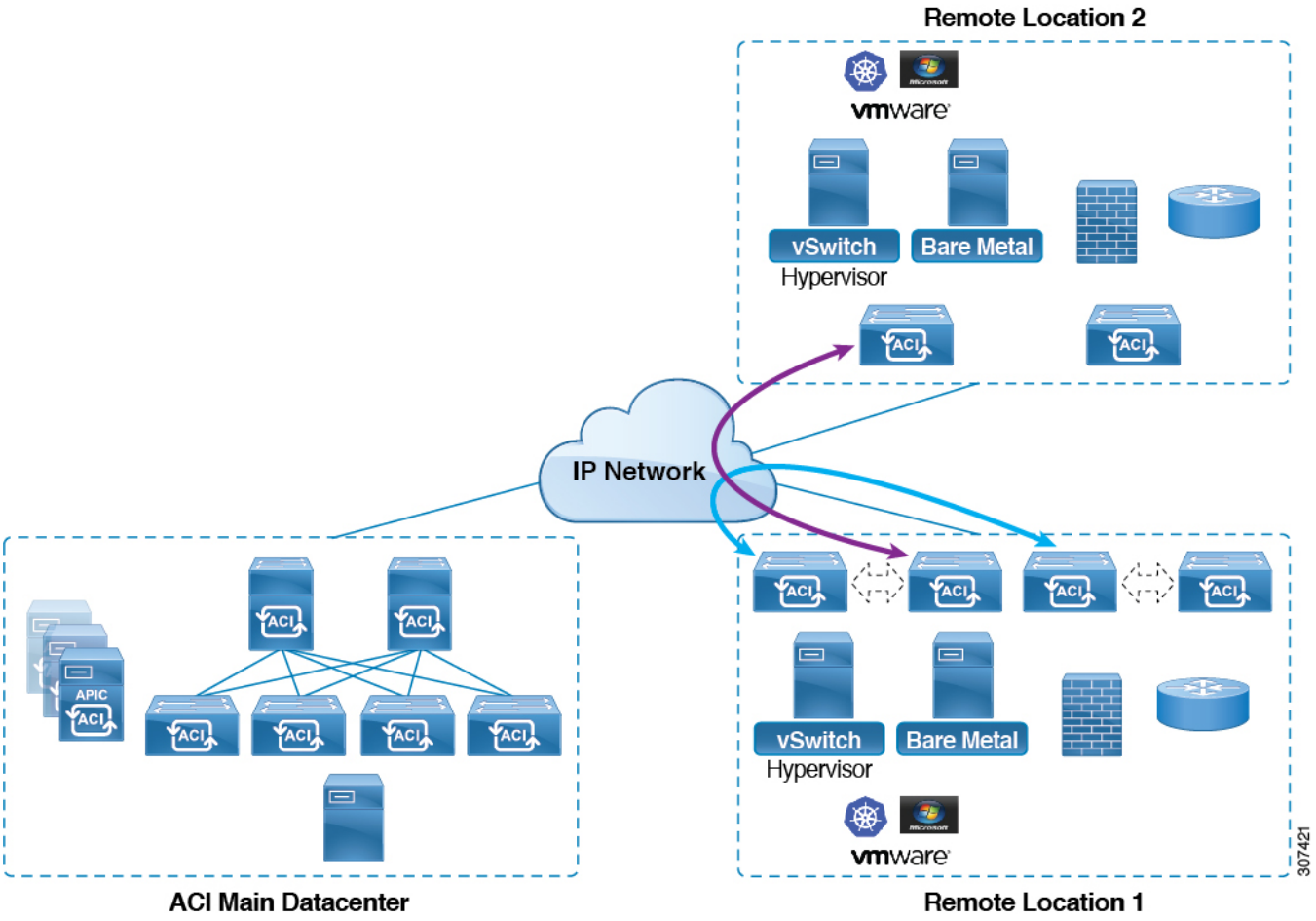
In addition, before Release 4.1(2), traffic between the remote leaf switch vPC pairs, either within a remote location or between remote locations, is forwarded to the spine switches in the ACI main data center pod, as shown in the following figure.

Figure 39: Remote Switching Traffic: Prior to Release 4.1(2)



Starting in Release 4.1(2), support is now available for direct traffic forwarding between remote leaf switches in different remote locations. This functionality offers a level of redundancy and availability in the connections between remote locations, as shown in the following figure.

Figure 40: Remote Leaf Switch Behavior: Release 4.1(2)



In addition, remote leaf switch behavior also takes on the following characteristics starting in release 4.1(2):

- Starting with Release 4.1(2), with direct traffic forwarding, when a spine switch fails within a single-pod configuration, the following occurs:
  - Local switching will continue to function for existing and new end point traffic between the remote leaf switch vPC peers, as shown in the "Local Switching Traffic: Prior to Release 4.1(2)" figure above.
  - For traffic between remote leaf switches across remote locations:
    - New end point traffic will fail because the remote leaf switch-to-spine switch tunnel would be down. From the remote leaf switch, new end point details will not get synced to the spine switch, so the other remote leaf switch pairs in the same or different locations cannot download the new end point information from COOP.
    - For uni-directional traffic, existing remote end points will age out after 300 secs, so traffic will fail after that point. Bi-directional traffic within a remote leaf site (between remote leaf VPC pairs) in a pod will get refreshed and will continue to function. Note that Bi-directional traffic to remote locations (remote leaf switches) will be affected as the remote end points will be expired by COOP after a timeout of 900 seconds.

- Bi-directional traffic within a remote leaf site (between remote leaf VPC pairs) in a pod will get refreshed and will continue to function. Note that Bi-directional traffic to remote locations (remote leaf switches) will be affected as the remote end points will be expired by COOP after a timeout of 900 seconds.
- For shared services (inter-VRF), bi-directional traffic between end points belonging to remote leaf switches attached to two different remote locations in the same pod will fail after the remote leaf switch COOP end point age-out time (900 sec). This is because the remote leaf switch-to-spine COOP session would be down in this situation. However, shared services traffic between end points belonging to remote leaf switches attached to two different pods will fail after 30 seconds, which is the COOP fast-aging time.
- L3Out-to-L3Out communication would not be able to continue because the BGP session to the spine switches would be down.

- When there is remote leaf direct uni-directional traffic, where the traffic is from remote leaf switch to remote leaf switch or from remote leaf switch to local leaf switch, there will be a milli-second traffic loss every time the remote end point (XR EP) timeout of 300 seconds occurs.

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-C9316D-GX
  - N9K-C9332C
  - N9K-C9364C
  - N9K-C9364C-GX
- 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.

Remote Leaf Switch Restrictions and Limitations

The following guidelines and restrictions apply to remote leaf switches:

- A remote leaf vPC pair has a split brain condition when the DP-TEP address of one of the switches is not reachable from the peer. In this case, both remote leaf switches are up and active in the fabric and the COOP session is also up on both of the peers. One of the remote leaf switches does not have a route to the DP-TEP address of its peer, and due to this, the vPC has a split brain condition. Both of the node roles is changed to "primary" and all the front panel links are up in both of the peers while the zero message queue (ZMQ) session is down.
- The remote leaf solution requires the /32 tunnel end point (TEP) IP addresses of the remote leaf switches and main data center leaf/spine switches to be advertised across the main data center and remote leaf switches without summarization.
- If you move a remote leaf switch to a different site within the same pod and the new site has the same node ID as the original site, you must delete and recreate the virtual port channel (vPC).
- With the Cisco N9K-C9348GC-FXP switch, you can perform the initial remote leaf switch discovery only on ports 1/53 or 1/54. Afterward, you can use the other ports for fabric uplinks to the ISN/IPN for the remote leaf switch.

The following sections provide information on what is supported and not supported with remote leaf switches:

- Supported Features, on page 358
- Unsupported Features, on page 359

Supported Features

Beginning with Cisco APIC release 4.1(2), the following features are supported:

- Remote leaf switches with ACI Multi-Site
- Traffic forwarding directly across two remote leaf vPC pairs in the same remote data center or across data centers, when those remote leaf pairs are associated to the same pod or to pods that are part of the same multipod fabric
- Transit L3Out across remote locations, which is when the main Cisco ACI data center 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)

Beginning with Cisco APIC release 4.0(1), the following features are supported:

- 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

Unsupported Features

Full fabric and tenant policies are supported on remote leaf switches in this release with the exception of the following features, which are unsupported:

- GOLF
- vPod
- Floating L3Out
- Fast-convergence mode
- Stretching of L3Out SVI between local leaf switches (ACI main data center switches) and remote leaf switches or stretching across two different vPC pairs of remote leaf switches
- Copy service is not supported when deployed on local leaf switches and when the source or destination is on the remote leaf switch. In this situation, the routable TEP IP address is not allocated for the local leaf switch. For more information, see the section "Copy Services Limitations" in the "Configuring Copy Services" chapter in the *Cisco APIC Layer 4 to Layer 7 Services Deployment Guide*, available in the APIC documentation page.
- 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
- Layer 3 Multicast routing on remote leaf switches
- Maintenance mode
- TEP to TEP atomic counters

The following scenarios are not supported when integrating remote leaf switches in a Multi-Site architecture in conjunction with the intersite L3Out functionality:

- Transit routing between L3Outs deployed on remote leaf switch pairs associated to separate sites

- Endpoints connected to a remote leaf switch pair associated to a site communicating with the L3Out deployed on the remote leaf switch pair associated to a remote site
- Endpoints connected to the local site communicating with the L3Out deployed on the remote leaf switch pair associated to a remote site
- Endpoints connected to a remote leaf switch pair associated to a site communicating with the L3Out deployed on a remote site

Note The limitations above do not apply if the different data center sites are deployed as pods as part of the same Multi-Pod fabric.

The following deployments and configurations are not supported with the remote leaf switch feature:

- It is not supported to stretch a bridge domain between remote leaf nodes associated to a given site (APIC domain) and leaf nodes part of a separate site of a Multi-Site deployment (in both scenarios where those leaf nodes are local or remote) and a fault is generated on APIC to highlight this restriction. This applies independently from the fact that BUM flooding is enabled or disabled when configuring the stretched bridge domain on the Multi-Site Orchestrator (MSO). However, a bridge domain can always be stretched (with BUM flooding enabled or disabled) between remote leaf nodes and local leaf nodes belonging to the same site (APIC domain).
- Spanning Tree Protocol across remote leaf location and main data center
- APICs 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
- /32 loopbacks advertised from the spine switch to the IPN must not be suppressed/aggregated toward the remote leaf switch. The /32 loopbacks must be advertised to the remote leaf switch.

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 that is 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:

```
<fabricSetupPol>
      <fabricSetupP tepPool="10.0.0.0/16" podId="1" >
           <fabricExtSetupP tepPool="30.0.128.0/20" extPoolId="1"/>
     </fabricSetupP>
     <fabricSetupP tepPool="10.1.0.0/16" podId="2" >
          <fabricExtSetupP tepPool="30.1.128.0/20" extPoolId="1"/>
     </fabricSetupP>
</fabricSetupPol>
```

Step 2 To define the node identity policy, send a post with XML, such as the following example:

Example:

```
<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"/>
     <fabricNodeIdentP serial="SAL17267Z7Y" name="leaf3" nodeId="201" podId="1"
extPoolId="1" nodeType="remote-leaf-wan"/>
     <fabricNodeIdentP serial="SAL17267Z7Z" name="leaf4" nodeId="201" 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 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="209.1.0.0/16" 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="210.1.0.0/16" scope="import-security"/>
               <l3extSubnet ip="203.1.0.0/16" scope="import-security"/>
               <l3extSubnet ip="208.2.0.0/16" scope="import-security"/>
          </l3extFabricExtRoutingP>
          <fvPodConnP id="1">
                <fvIp addr="100.11.1.1/32"/>
         </fvPodConnP>
         <fvPodConnP id="2">
```

```
                <fvIp addr="200.11.1.1/32"/>
            </fvPodConnP>
            <fvPeeringP type="automatic_with_full_mesh"/>
        </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 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">
      <l3extRsNodeL3OutAtt rtrId="202.202.202.202" tDn="topology/pod-1/node-101">
      </l3extRsNodeL3OutAtt>
      <l3extLIfP name="portIf">
        <l3extRsPathL3OutAtt ifInstT="sub-interface"
tDn="topology/pod-1/paths-101/pathep-[eth1/49]" addr="202.1.1.2/30" mac="AA:11:22:33:44:66"
 encap='vlan-4'/>
        <ospfIfP>
          <ospfRsIfPol tnOspfIfPolName='ospfIfPol'/>
        </ospfIfP>
      </l3extLIfP>
    </l3extLNodeP>
    <l3extLNodeP name="rlSpine-201">
      <l3extRsNodeL3OutAtt rtrId="201.201.201.201" rtrIdLoopBack="no"
tDn="topology/pod-1/node-201">
        <!--
        <l3extLoopBackIfP addr="201::201/128" descr="" name=""/>
        <l3extLoopBackIfP addr="201.201.201.201/32" descr="" name=""/>
        -->
        <l3extLoopBackIfP addr="::" />
      </l3extRsNodeL3OutAtt>
      <l3extLIfP name="portIf">
        <l3extRsPathL3OutAtt ifInstT="sub-interface"
tDn="topology/pod-1/paths-201/pathep-[eth8/36]" addr="201.1.1.1/30" mac="00:11:22:33:77:55"
 encap='vlan-4'/>
        <ospfIfP>
          <ospfRsIfPol tnOspfIfPolName='ospfIfPol'/>
        </ospfIfP>
      </l3extLIfP>
    </l3extLNodeP>
    <l3extInstP descr="" matchT="AtleastOne" name="instp1" prio="unspecified"
targetDscp="unspecified">
      <fvRsCustQosPol tnQosCustomPolName=""/>
    </l3extInstP>
  </l3extOut>
  <ospfIfPol name="ospfIfPol" nwT="bcast"/>
</fvTenant>
</polUni>
```

Step 5 For releases prior to Release 4.1(2), 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.

Note Do not enter this information if you are deploying new remote leaf switches running Release 4.1(2) or later and you are enabling direct traffic forwarding on those remote leaf switches. Configuring an OSPF instance using VLAN-5 for multipod is not needed in this case.

See About Direct Traffic Forwarding, on page 377 for more information.

Example:

```
<?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" />
      <l3extLNodeP name="bLeaf">
        <l3extRsNodeL3OutAtt rtrId="202.202.202.202" rtrIdLoopBack="no"
tDn="topology/pod-2/node-202">
            <l3extLoopBackIfP addr="202.202.202.212"/>
       </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">
          <ospfIfP authKeyId="1" authType="none">
            <ospfRsIfPol tnOspfIfPolName="ospfIfPol" />
          </ospfIfP>
         <l3extRsPathL3OutAtt addr="10.0.254.233/30" encap="vlan-5" ifInstT="sub-interface"
 tDn="topology/pod-2/paths-202/pathep-[eth5/2]"/>
         <l3extRsPathL3OutAtt addr="10.0.255.229/30" encap="vlan-5" ifInstT="sub-interface"
 tDn="topology/pod-1/paths-102/pathep-[eth5/2]"/>
        </l3extLIfP>
      </l3extLNodeP>
      <l3extInstP matchT="AtleastOne" name="ipnInstP" />
    </l3extOut>
  </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 360.
- 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 rl-wan
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)#
```

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)# vrf context tenant infra vrf overlay-1 l3out ipn-multipodInternal
apic1(config-spine-vrf)# exit
apic1(config-spine)#
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)#
```

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
apic1(config-leaf-if)# ip router ospf default area 5
apic1(config-leaf-if)# exit
```

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
remote-leaf-site 5 node-type remote-leaf-wan
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)# vrf context tenant infra vrf overlay-1 l3out ipn-multipodInternal
apic1(config-spine-vrf)# exit
apic1(config-spine)#
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: Releases Prior to 4.1(2)

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 374.

Note These procedures describe how to configure the remote leaf switches using the wizard for releases prior to 4.1(2). For instructions on configuring the remote leaf switches using the wizard for Release 4.1(2) and later, see Configure the Pod and Fabric Membership for Remote Leaf Switches Using a Wizard: Releases 4.1(2) and Later, on page 369.

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 360.
- 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 swiches is connected to the IPN router.

Procedure

Step 1 On the menu bar click **Fabric** > **Inventory**.

Step 2 In the Navigation pane, expand **Quick Start** and click **Node or Pod Setup**.

Step 3 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**.

Step 4 Follow the instructions to configure the following:

- **Pod Fabric**—Identify the pod and the TEP Pool subnet for the remote leaf switches.

  Add the comma-separated subnets for the underlay routes leading to the remote leaf switches.

  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.

Step 5 On the menu bar click **System** > **System Settings**.

Step 6 In the Navigation pane, choose **System Global GIPo**.

Step 7 For **Use Infra GIPo as System GIPo**, choose **Enabled**.

Configure the Pod and Fabric Membership for Remote Leaf Switches Using a Wizard: Releases 4.1(2) and Later

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 374.

Note These procedures describe how to configure the remote leaf switches using the wizard for Release 4.1(2) and later. For instructions on configuring the remote leaf switches using the wizard for releases prior to 4.1(2), see Configure the Pod and Fabric Membership for Remote Leaf Switches Using a Wizard: Releases Prior to 4.1(2), on page 368.

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 360.
- The remote leaf switch pair are connected with a vPC.
- The remote leaf switches are running a switch image of 14.1.x or later (aci-n9000-dk9.14.1.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 swiches is connected to the IPN router.

Procedure

Step 1 On the menu bar click **Fabric** > **Inventory**.

Step 2 In the Navigation pane, expand **Quick Start** and click **Add Remote Leaf**.

Step 3 In the **Remote Leaf** pane of the working pane, click **Add Remote Leaf**.

Step 4 Configure the interpod connectivity before adding the remote leaf switch, if necessary.

You will see the **Configure Interpod Connectivity** screen if you do not have connections configured yet between the physical Pod and the IPN connectivity. This connectivity is a prerequisite before extending ACI to another location. You will configure the IP connectivity, routing protocols, and external TEP addresses in this part of the configuration wizard in this situation.

For information on configuring interpod connectivity, see Preparing the Pod for IPN Connectivity, on page 340.

Step 5 At the end of the process for configuring interpod connectivity, click **Add Remote Leaf** in the **Summary** page.

The **Add Remote Leaf** wizard appears.

Step 6 In the **Add Remote Leaf** wizard, review the information in the **Overview** page.

This panel provides high-level information about the steps that are required for adding a remote leaf switch to a pod in the fabric. The information that is displayed in the **Overview** panel, and the areas that you will be configuring in the subsequent pages, varies depending on your existing configuration:

- If you are adding a new remote leaf switch to a single-pod or multi-pod configuration, you will typically see the following items in the **Overview** panel, and you will be configuring these areas in these subsequent pages:
  - **External TEP**
  - **Pod Selection**
  - **Routing Protocol**
  - **Remote Leafs**

  In addition, because you are adding a new remote leaf switch, it will automatically be configured with the direct traffic forwarding feature, which was introduced in Release 4.1(2).
- If you already have remote leaf switches configured and you are using the remote leaf wizard to configure these existing remote leaf switches, but the existing remote leaf switches were upgraded from a software release prior to Release 4.1(2), then those remote leaf switches might not be configured with the direct traffic forwarding feature. You will see a warning at the top of the Overview page in this case, beginning with the statement "Remote Leaf Direct Communication is not enabled."

  You have two options when adding a remote leaf switch using the wizard in this situation:
  - **Enable the direct traffic forwarding feature on these existing remote leaf switches.** This is the recommended course of action in this situation. You must first manually enable the direct traffic forwarding feature on the switches using the instructions provided in Upgrade the Remote Leaf Switches and Enable Direct Traffic Forwarding, on page 378. Once you have manually enabled the direct traffic forwarding feature using those instructions, return to this remote leaf switch wizard and follow the process in the wizard to add the remote leaf switches to a pod in the fabric.
  - **Add the remote leaf switches without enabling the direct traffic forwarding feature.** This is an acceptable option, though not recommended. To add the remote leaf switches without enabling the direct traffic forwarding feature, continue with the remote leaf switch wizard configuration without manually enabling the direct traffic forwarding feature.

Step 7 When you have finished reviewing the information in the **Overview** panel, click **Get Started** at the bottom right corner of the page.

- If you adding a new remote leaf switch, where it will be running Release 4.1(2) or above and will be automatically configured with the direct traffic forwarding feature, the **External TEP** page appears. Go to Step 8, on page 371.

- If you are adding a remote leaf switch without enabling the direct traffic forwarding feature, or if you upgraded your switches to Release 4.1(2) and you manually enabled the direct traffic forwarding feature on the switches using the instructions provided in Upgrade the Remote Leaf Switches and Enable Direct Traffic Forwarding, on page 378, then the **Pod Selection** page appears. Go to Step 9, on page 371.

Step 8 In the **External TEP** page, configure the necessary parameters.

External TEP addresses are used by the physical pod to communicate with remote locations. In this page, configure a subnet that is routable across the network connecting the different locations. The external TEP pool cannot overlap with other internal TEP pools, remote leaf TEP pools, or external TEP pools from other pods. The wizard will automatically allocate addresses for pod-specific TEP addresses and spine router IDs from the external TEP pool. You can modify the proposed addresses, if necessary.

a) Leave the **Use Defaults** checkbox checked, or uncheck it, if necessary.

When checked, the wizard automatically allocates data plane and unicast TEP addresses. Those fields are not displayed when the **Use Defaults** box is checked. Uncheck the **Use Defaults** box to view or modify the proposed addresses, if necessary.

b) In the **External TEP Pool** field, enter the external TEP for the physical pod.

The external TEP pool must not overlap the internal TEP pool.

c) In the **Unicast TEP IP** field, change the value that is automatically populated in this field, if necessary.

This address is automatically allocated by Cisco APIC from the External TEP Pool, and will be used for sending traffic from the remote leaf switch to the local leaf switches on that pod.

Cisco APIC automatically configures the unicast TEP IP address when you enter the External TEP Pool address.

d) Repeat these steps for each pod, if you have a multi-pod configuration.

e) When you have entered all of the necessary information in this page, click the **Next** button at the bottom right corner of the page.

The **Pod Selection** page appears.

Step 9 In the **Pod Selection** page, configure the necessary parameters.

The remote leaf switch logically connects to one of the pods in the Cisco ACI fabric. In this page, select the pod ID of the pod where the remote leaf switches will be associated. A remote leaf TEP pool is needed to allocate IP addresses to the remote leaf switches. Select an existing remote leaf TEP pool or enter a remote leaf TEP pool to create a new one. The remote leaf TEP pool must be different from existing TEP pools. Multiple remote leaf pairs can be part of the same remote TEP pool.

a) In the **Pod ID** field, select the pod ID of the pod where the remote leaf switches will be associated.

b) In the **Remote Leaf TEP Pool** field, select an existing remote leaf TEP pool or enter a remote leaf TEP pool to allocate IP addresses to the remote leaf switches.

Click the **View existing TEP Pools** link underneath the **Remote Leaf TEP Pool** field to see the existing TEP pools (internal TEP pools, remote leaf TEP pools, and external TEP pools). Use this information to avoid creating duplicate or overlapping pools.

c) When you have entered all of the necessary information in this page, click the **Next** button at the bottom right corner of the page.

The **Routing Protocol** page appears.

Step 10 In the **Routing Protocol** page, configure the necessary parameters.

OSPF is used in the underlay to peer between the remote leaf switches and the upstream router. Create or select an existing L3 Outside to represent the connection between the remote leaf switches and the upstream router. Multiple remote leaf pairs can use the same L3 Outside to represent their upstream connection. Configure the OSPF Area ID, an Area Type, and OSPF Interface Policy in this page. The OSPF Interface Policy contains OSPF-specific settings, such as the OSPF network type, interface cost, and timers. Configure the OSFP Authentication Key and OSPF Area Cost by unchecking the **Use Defaults** checkbox.

Note If you peer a Cisco ACI-mode switch with a standalone Cisco Nexus 9000 switch that has the default OSPF authentication key ID of 0, the OSPF session will not come up. Cisco ACI only allows an OSPF authentication key ID of 1 to 255.

a) Under the **L3 Outside Configuration** section, in the **L3 Outside** field, create or select an existing L3Out to represent the connection between the remote leaf switches and the upstream router.

For the remote leaf switch configuration, we recommend that you use or create an L3Out that is different from the L3Out used in the multi-pod configuration.

b) Under the **OSPF** section, leave the **Use Defaults** checkbox checked, or uncheck it, if necessary.

When the checkbox is checked, the Cisco APIC GUI conceals the optional fields for configuring OSPF.

The checkbox is checked by default. Uncheck it to reveal the optional fields.

c) Gather the configuration information from the IPN, if necessary.

For example, from the IPN, you might enter the following command to gather certain configuration information:

```
IPN# show running-config interface ethernet slot/chassis-number
```

For example:

```
IPN# show running-config interface ethernet 1/5.11
...
ip router ospf infra area 0.0.0.59
...
```

d) In the **Area ID** field, enter the OSPF area ID.

Looking at the OSPF area 59 information shown in the output in the previous step, you could enter a different area in the **Area ID** field (for example, `0`) and have a different L3Out. If you are using a different area for the remote leaf switch, you must create a different L3Out. You can also create a different L3Out, even if you are using the same OSPF area ID.

e) In the **Area Type** field, select the OSPF area type.

You can choose one of the following OSPF types:

- **NSSA area**
- **Regular area**

Note You might see **Stub area** as an option in the **Area Type** field; however, stub area will not advertise the routes to the IPN, so stub area is not a supported option for infra L3Outs.

Regular area is the default.

f) In the **Area Cost** field, select the appropriate OSPF value.

g) In the **Authentication Type** field, select the appropriate OSPF authentication type.
h) In the **Authentication Key** field, select the appropriate OSPF authentication key. Re-enter the OSPF authentication key in the **Confirm Key** field.
i) In the **Interface Policy** field, enter or select the OSPF interface policy.

   You can choose an existing policy or create a new one using the **Create OSPF Interface Policy** dialog box.

j) When you have entered all of the necessary information in this page, click the **Next** button at the bottom right corner of the page.

   The **Remote Leafs** page appears.

Step 11 In the **Remote Leafs** page, configure the necessary parameters.

The interpod network (IPN) connects Cisco ACI locations to provide end-to-end network connectivity. To achieve this, remote leaf switches need IP connectivity to the upstream router. For each remote leaf switch, enter a router ID that will be used to establish the control-plane communication with the upstream router and the rest of the Cisco ACI fabric. Also provide the IP configuration for at least one interface for each remote leaf switch. Multiple interfaces are supported.

a) In the **Serial** field, enter the serial number for the remote leaf switch or select a discovered remote leaf switch from the dropdown menu.
b) In the **Node ID** field, assign a node ID to the remote leaf switch.
c) In the **Name** field, assign a name to the remote leaf switch.
d) In the **Router ID** field, enter a router ID that will be used to establish the control-plane communication with the upstream router and the rest of the Cisco ACI fabric.
e) In the **Loopback Address** field, enter the IPN router loopback IP address, if necessary.

   Leave this field blank if you use a router ID address.

f) Under the **Interfaces** section, in the **Interface** field, enter interface information for this remote leaf switch.
g) Under the **Interfaces** section, in the **IPv4 Address** field, enter the IPv4 IP address for the interface.
h) Enter information on additional interfaces, if necessary.

   Click + within the Interfaces box to enter information for multiple interfaces.

i) When you have entered all of the necessary information for this remote leaf switch, enter information for additional remote leaf switches, if necessary.

   Click + to the right of the Interfaces box to enter information for multiple remote leaf switches.

j) When you have entered all of the necessary information in this page, click the **Next** button at the bottom right corner of the page.

   The **Confirmation** page appears.

Step 12 In the **Confirmation** page, review the list of policies that the wizard will create and change the names of any of the policies, if necessary, then click **Finish** at the bottom right corner of the page.

The **Remote Leaf Summary** page appears.

Step 13 In the **Remote Leaf Summary** page, click the appropriate button.

- If you want to view the API for the configuration in a JSON file, click **View JSON**. You can copy the API and store it for future use.

- If you are satisfied with the information in this page and you do not want to view the JSON file, click **OK**.

Step 14 In the Navigation pane, click **Fabric Membership**, then click the **Nodes Pending Registration** tab to view the status of the remote leaf switch configuration.

You should see `Undiscovered` in the **Status** column for the remote leaf switch that you just added.

Step 15 Log into the spine switch connected to the IPN and enter the following command:

```
switch# show nattable
```

Output similar to the following appears:

```
------- NAT TABLE ----------
Private Ip   Routeable Ip
10.0.0.1     192.0.2.100
10.0.0.2     192.0.2.101
10.0.0.3     192.0.2.102
```

Step 16 On the IPN sub-interfaces connecting the remote leaf switches, configure the DHCP relays for each interface.

For example:

```
switch# configure terminal
switch(config)# interface ethernet 1/5.11
switch(config-subif)# ip dhcp relay address 192.0.2.100
switch(config-subif)# ip dhcp relay address 192.0.2.101
switch(config-subif)# ip dhcp relay address 192.0.2.102
switch(config-subif)# exit
switch(config)# interface ethernet 1/7.11
switch(config-subif)# ip dhcp relay address 192.0.2.100
switch(config-subif)# ip dhcp relay address 192.0.2.101
switch(config-subif)# ip dhcp relay address 192.0.2.102
switch(config-subif)# exit
switch(config)# exit
switch#
```

Step 17 In the Navigation pane, click **Fabric Membership**, then click the **Registered Nodes** tab to view the status of the remote leaf switch configuration.

After a few moments, you should see `Active` in the **Status** column for the remote leaf switch that you just added.

Step 18 On the menu bar click **System** > **System Settings**.

Step 19 In the Navigation pane, choose **System Global GIPo**.

Step 20 For **Use Infra GIPo as System GIPo**, choose **Enabled**.

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: Releases Prior to 4.1(2), on page 368

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 360.
- 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 swiches 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
- 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) For releases prior to release 4.1(2), check the **Enable Remote Leaf** check box if the pod where you are adding the remote leaf switches is part of a multipod fabric.

Note Do not check the **Enable Remote Leaf** check box if you are deploying new remote leaf switches running release 4.1(2) or later and you are enabling direct traffic forwarding on those remote leaf switches. This option enables an OSPF instance using VLAN-5 for multipod, which is not needed in this case.

See About Direct Traffic Forwarding, on page 377 for more information.

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.

About Direct Traffic Forwarding

As described in Characteristics of Remote Leaf Switch Behavior in Release 4.1(2), on page 354, support for direct traffic forwarding is supported starting in Release 4.1(2). However, the method that you use to enable or disable direct traffic forwarding varies, depending on the version of software running on the remote leaf switches:

- If your remote leaf switches are currently running on Release 4.1(2) or later [if the remote leaf switches were never running on a release prior to 4.1(2)], go to Configure the Pod and Fabric Membership for Remote Leaf Switches Using a Wizard: Releases 4.1(2) and Later, on page 369.

- If your remote leaf switches are currently running on a release prior to 4.1(2), go to Upgrade the Remote Leaf Switches and Enable Direct Traffic Forwarding, on page 378 to upgrade the switches to Release 4.1(2) or later, then make the necessary configuration changes and enable direct traffic forwarding on those remote leaf switches.
- If your remote leaf switches are running on Release 4.1(2) or later and have direct traffic forwarding enabled, but you want to downgrade to a release prior to 4.1(2), go to Disable Direct Traffic Forwarding and Downgrade the Remote Leaf Switches, on page 381 to disable the direct traffic forwarding feature before downgrading those remote leaf switches.

Upgrade the Remote Leaf Switches and Enable Direct Traffic Forwarding

If your remote leaf switches are currently running on a release prior to 4.1(2), follow these procedures to upgrade the switches to Release 4.1(2) or later, then make the necessary configuration changes and enable direct traffic forwarding on those remote leaf switches.

Note When upgrading to Release 4.1(2) or later, enabling direct traffic forwarding might be optional or mandatory, depending on the release you are upgrading to:

- If you are upgrading to a release prior to Release 5.0(1), then enabling direct traffic forwarding is **optional**; you can upgrade your switches without enabling the direct traffic forwarding feature. You can enable this feature at some point after you've made the upgrade, if necessary.
- If you are upgrading to Release 5.0(1) or later, then enabling direct traffic forwarding is **mandatory**. Direct traffic forwarding is enabled by default starting in Release 5.0(1) and cannot be disabled.

If, at a later date, you have to downgrade the software on the remote leaf switches to a version that doesn't support remote leaf switch direct traffic forwarding (to a release prior to Release 4.1(2), follow the procedures provided in Disable Direct Traffic Forwarding and Downgrade the Remote Leaf Switches, on page 381 to disable the direct traffic forwarding feature before downgrading the software on the remote leaf switches.

Procedure

Step 1 Upgrade Cisco APIC and all the nodes in the fabric to Release 4.1(2) or later.

Step 2 Verify that the routes for the Routable Subnet that you wish to configure will be reachable in the Inter-Pod Network (IPN), and that the subnet is reachable from the remote leaf switches.

Step 3 Configure Routable Subnets in all the pods in the fabric:

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 configure routable subnets.
d) Access the information in the subnets or TEP table, depending on the release of your APIC software:

- For releases prior to 4.2(3), click the + on the **Routable Subnets** table.
- For 4.2(3) only, click the + on the **External Subnets** table.
- For 4.2(4) and later, click the + on the **External TEP** table.

e) Enter the IP address and Reserve Address, if necessary, and set the state to Active or Inactive.

- The IP address is the subnet prefix that you wish to configure as the routeable IP space.
- The Reserve Address is a count of addresses within the subnet that must not be allocated dynamically to the spine switches and remote leaf switches. The count always begins with the fist IP in the subnet and increments sequentially. If you wish to allocate the Unicast TEP (covered later in these procedures) from this pool, then it must be reserved.

f) On the **Fabric Setup Policy** panel, click **Submit**.

Note If you find that you have to make changes to the information in the subnets or TEP table after you've made these configurations, follow the procedures provided in "Changing the External Routable Subnet" in the *Cisco APIC Getting Started Guide* to make those changes successfully.

Step 4 Add Routable Ucast for each pod:

a) On the menu bar, click **Tenants** > **infra** > **Policies** > **Protocol** > **Fabric Ext Connection Policies** > ***intrasite-intersite_profile_name***.

b) In the properties page for this intrasite/intersite profile, click **+** in the **Pod Connection Profile** area.

The **Create Pod Connection Profile** window appears.

c) Select a pod and enter the necessary information in the **Create Pod Connection Profile** window.

In the **Unicast TEP** field, enter a routable TEP IP address, including the bit-length of the prefix, to be used for unicast traffic over the IPN. This IP address is used by the spine switches in their respective pod for unicast traffic in certain scenarios. For example, a unicast TEP is required for remote leaf switch direct deployments.

Step 5 Click **Submit**.

The following areas are configured after configuring Routable Subnets and Routable Ucast for each pod:

- On the spine switch, the Remote Leaf Multicast TEP Interface (rl-mcast-hrep) and Routable CP TEP Interface (rt-cp-etep) are created.
- On the remote leaf switches, the private Remote Leaf Multicast TEP Interface (rl-mcast-hrep) tunnel remains as-is.
- Traffic continues to use the private Remote Leaf Multicast TEP Interface (rl-mcast-hrep).
- Traffic will resume with the newly configured Routable Ucast TEP Interface. The private Remote Leaf Unicast TEP Interface (rl_ucast) tunnel is deleted from the remote leaf switch. Since traffic is converging on the newly configured Unicast TEP, expect a very brief disruption in service.
- The remote leaf switch and spine switch COOP (council of oracle protocol) session remains with a private IP address.
- The BGP route reflector switches to Routable CP TEP Interface (rt-cp-etep).

Step 6 Verify that COOP is configured correctly.

```
# show coop internal info global
# netstat -anp | grep 5000
```

Step 7 Verify that the BGP route reflector session in the remote leaf switch is configured correctly.

```
remote-leaf# show bgp vpnv4 unicast summary vrf all | grep 14.0.0
 14.0.0.227 4 100 1292 1164 395 0 0 19:00:13 52
 14.0.0.228 4 100 1296 1164 395 0 0 19:00:10 52
```

Step 8 Enable direct traffic forwarding on the remote leaf switches.

a) On the menu bar, click **System** > **System Settings**.

b) Click **Fabric Wide Setting**.

c) Click the check box on **Enable Remote Leaf Direct Traffic Forwarding**.

When this is enabled, the spine switches will install Access Control Lists (ACLs) to prevent traffic coming from remote leaf switches from being sent back, since the remote leaf switches will now send directly between each remote leaf switches' TEPs. There may be a brief disruption in service while the tunnels are built between the remote leaf switches.

d) Click **Submit**.

e) To verify that the configuration was set correctly, on the spine switch, enter the following command:

```
spine# cat /mit/sys/summary
```

You should see the following highlighted line in the output, which is verification that the configuration was set correctly (full output truncated):

```
...
podId : 1
remoteNetworkId : 0
remoteNode : no
rldirectMode : yes
rn : sys
role : spine
...
```

At this point, the following areas are configured:

- Network Address Translation Access Control Lists (NAT ACLs) are created on the data center spine switches.
- On the remote leaf switches, private Remote Leaf Unicast TEP Interface (rl_ucast) and Remote Leaf Multicast TEP Interface (rl-mcast-hrep) tunnels are removed and routable tunnels are created.
- The **rlRoutableMode** and **rldirectMode** attributes are set to **yes**, as shown in the following example:

```
remote-leaf# moquery -d sys | egrep "rlRoutableMode|rldirectMode"
rlRoutableMode : yes
rldirectMode : yes
```

Step 9 Add the Routable IP address of Cisco APIC as DHCP relay on the IPN interfaces connecting the remote leaf switches.

Each APIC in the cluster will get assigned an address from the pool. These addresses must be added as the DHCP relay address on the interfaces facing the remote leaf switches. You can find these addresses by running the following command from the APIC CLI:

```
remote-leaf# moquery -c infraWiNode | grep routable
```

Step 10 Decommission and recommission each remote leaf switch one at a time to get it discovered on the routable IP address for the Cisco APIC.

The COOP configuration changes to Routable CP TEP Interface (rt-cp-etep). After each remote leaf switch is decommissioned and recommissioned, the DHCP server ID will have the routable IP address for the Cisco APIC.

Disable Direct Traffic Forwarding and Downgrade the Remote Leaf Switches

If your remote leaf switches are running on Release 4.1(2) or later and have direct traffic forwarding enabled, but you want to downgrade to a release prior to 4.1(2), follow these procedures to disable the direct traffic forwarding feature before downgrading the remote leaf switches.

Before you begin

Procedure

Step 1 For a multipod configuration, configure a multipod-internal L3Out using VLAN-5.

Step 2 Provision back private network reachability if it was removed when you enabled the direct traffic forwarding feature on the remote leaf switches.

For example, configure the private IP route reachability in IPN and configure the private IP address of the Cisco APIC as a DHCP relay address on the layer 3 interfaces of the IPN connected to the remote leaf switches.

Step 3 Disable remote leaf switch direct traffic forwarding for all remote leaf switches by posting the following policy:

```
POST URL : https://<ip address>/api/node/mo/uni/infra/settings.xml
<imdata>
    <infraSetPol dn="uni/infra/settings" enableRemoteLeafDirect="no" />
</imdata>
```

This will post the MO to Cisco APIC, then the configuration will be pushed from Cisco APIC to all nodes in the fabric.

At this point, the following areas are configured:

- The Network Address Translation Access Control Lists (NAT ACLs) are deleted on the data center spine switches.
- The **rlRoutableMode** and **rldirectMode** attributes are set to **no**, as shown in the following example:

```
remote-leaf# moquery -d sys | egrep "rlRoutableMode|rldirectMode"
rlRoutableMode : no
rldirectMode : no
```

Step 4 Remove the Routable Subnets and Routable Ucast from the pods in the fabric.

The following areas are configured after removing the Routable Subnets and Routable Ucast from each pod:

- On the spine switch, the Remote Leaf Multicast TEP Interface (rl-mcast-hrep) and Routable CP TEP Interface (rt-cp-etep) are deleted.

- On the remote leaf switches, the tunnel to the routable Remote Leaf Multicast TEP Interface (rl-mcast-hrep) is deleted, and a private Remote Leaf Multicast TEP Interface (rl-mcast-hrep) is created. The Remote Leaf Unicast TEP Interface (rl_ucast) tunnel remains routable at this point.
- The remote leaf switch and spine switch COOP (council of oracle protocol) and route reflector sessions switch to private.
- The tunnel to the routable Remote Leaf Unicast TEP Interface (rl_ucast) is deleted, and a private Remote Leaf Unicast TEP Interface (rl_ucast) tunnel is created.

Step 5 Decommission and recommission each remote leaf switch to get it discovered on the non-routable internal IP address of the Cisco APIC.

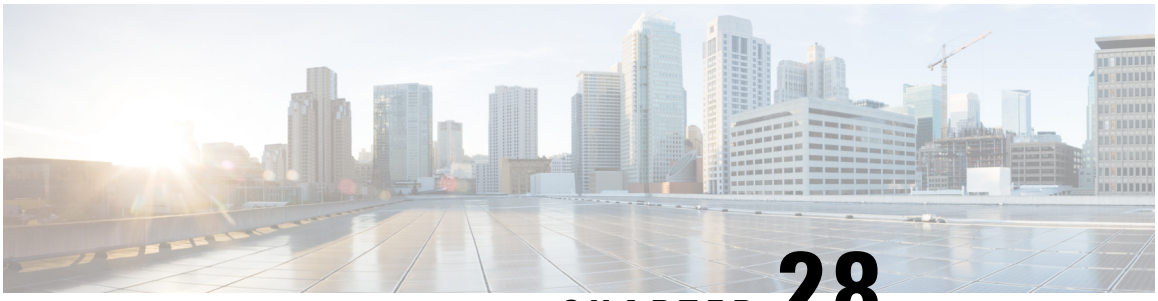
Step 6 Downgrade the Cisco APIC and all the nodes in the fabric to a release prior to 4.1(2).

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.



CHAPTER 28

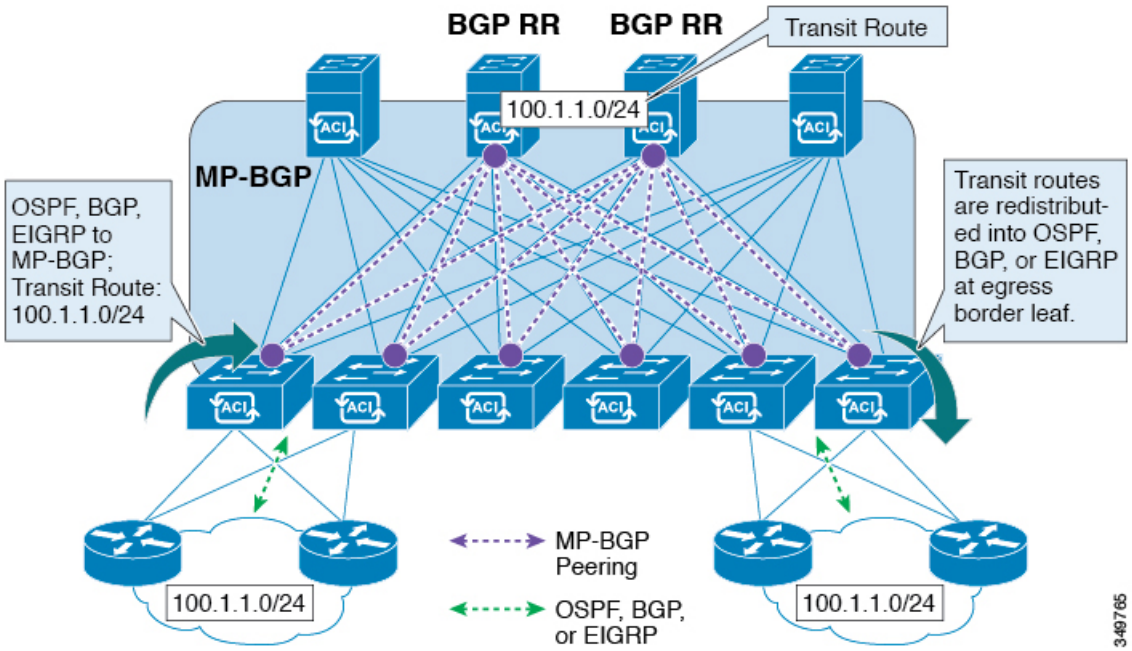
Transit Routing

This chapter contains the following sections:

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.

Figure 41: Transit Routing in the Fabric



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.

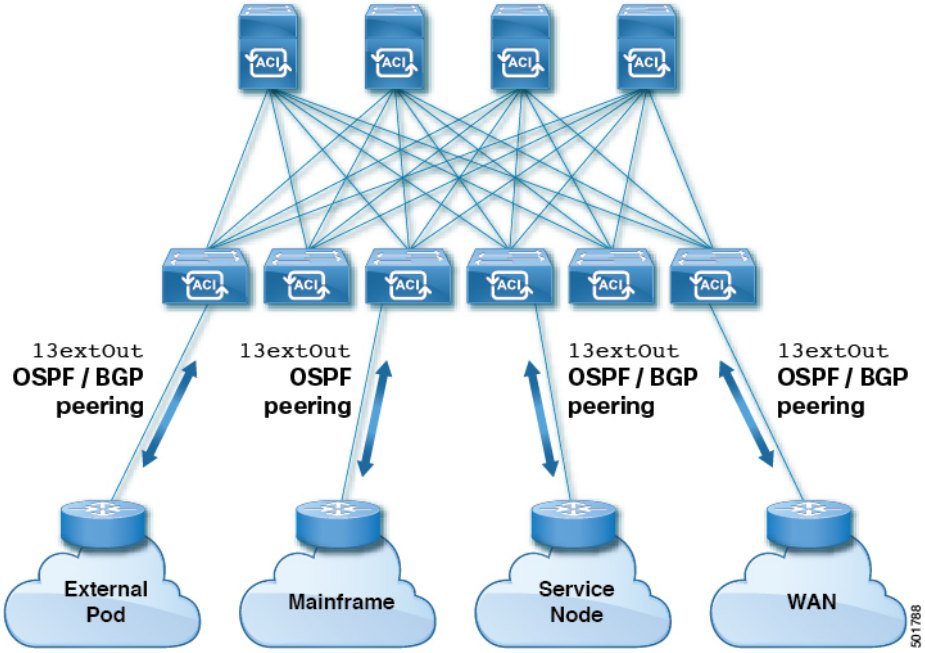
Note For cautions and guidelines for configuring transit routing, see Guidelines for Transit Routing, on page 391

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.

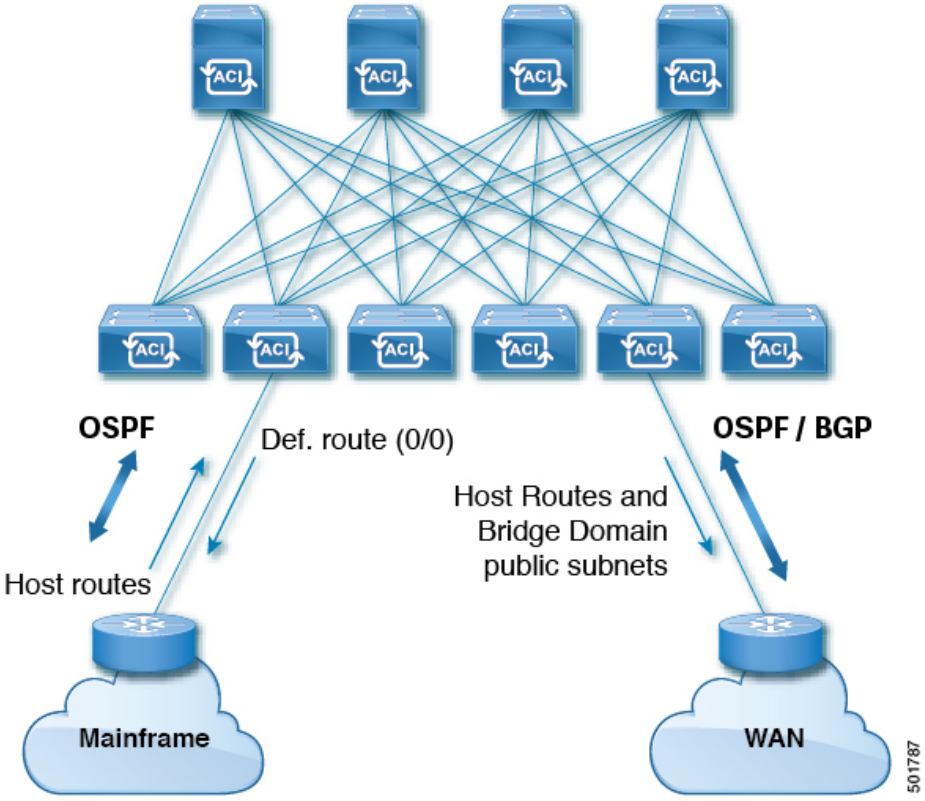
Figure 42: Transit Routing Between Layer 3 Domains



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).

Figure 43: Mainframe Transit Connectivity

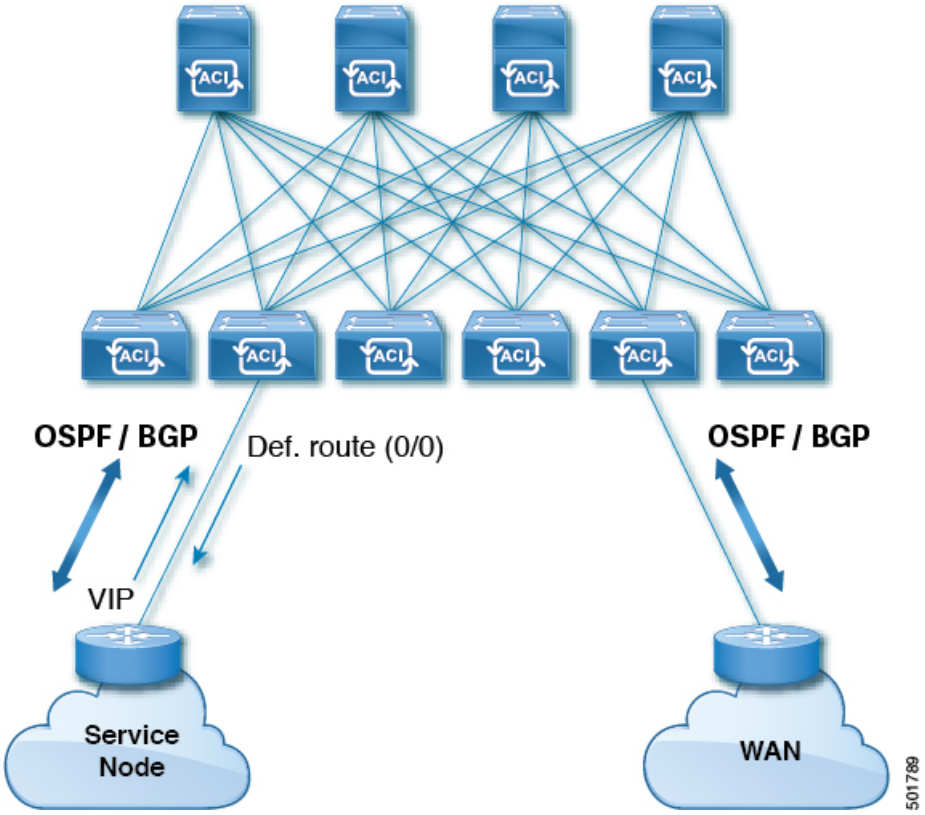


In this topology, mainframes require the ACI fabric to be a transit domain for external connectivity through a WAN router and for east-west traffic within the fabric. They push host routes to the fabric to be redistributed within the fabric and out to external interfaces.

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.

Figure 44: Service Node Transit Connectivity

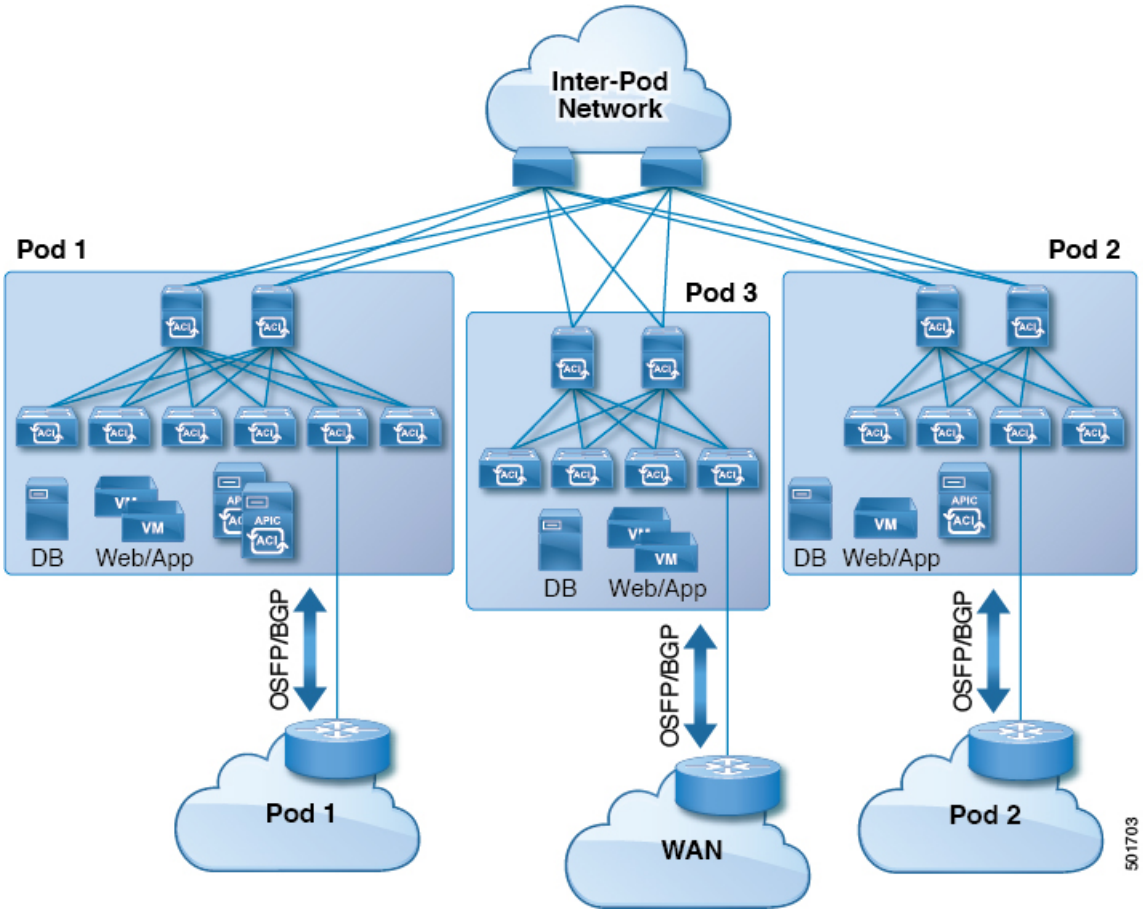


The VIP is the external facing IP address for a particular site or service. A VIP is tied to one or more servers or nodes behind a service node.

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.

Figure 45: Multiple Pods with L3Outs in a Transit-Routed Configuration



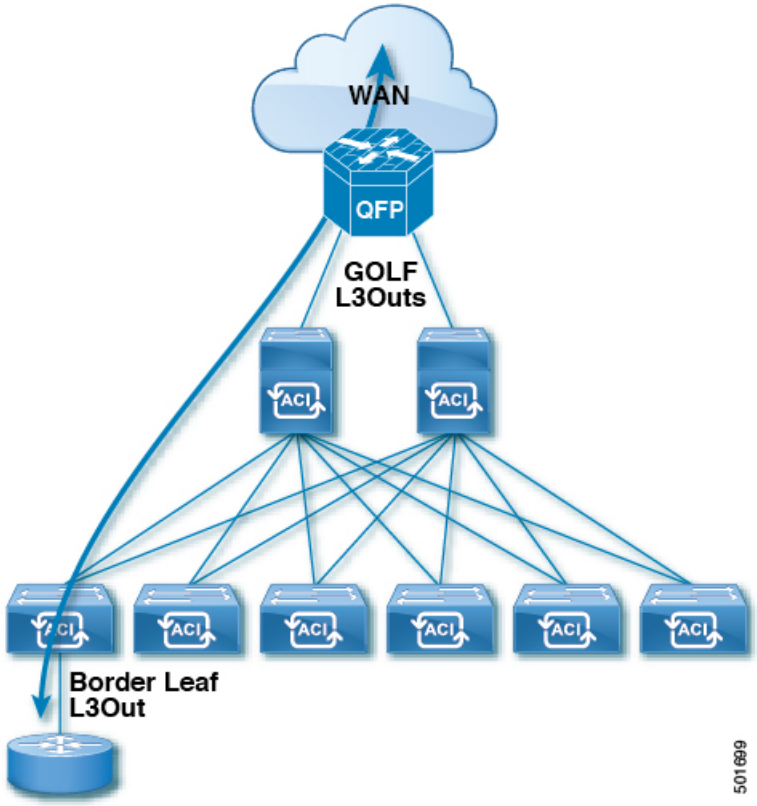
In such scenarios, the policies are administered at the demarcation points and ACI policies need not be imposed.

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.

Figure 46: GOLF L3Outs and a Border Leaf L3Out in a Transit-Routed Configuration



Supported Transit Combination Matrix

Layer 3 Outside Connection Type	OSPF	iBGP			eBGP			EIGRP v4	EIGRP v6	Static Route
		iBGP over OSPF	iBGP over Static Route	iBGP over Direct Connection	eBGP over OSPF	eBGP over Static Route	eBGP over Direct Connection			
OSPF	Yes	Yes*	Yes	Yes* (tested in APIC release 1.3c)	Yes	Yes	Yes	Yes	Yes* (tested in APIC release 1.2g)	Yes

Layer 3 Outside Connection Type		OSPF	iBGP			eBGP			EIGRP v4	EIGRP v6	Static Route
			iBGP over OSPF	iBGP over Static Route	iBGP over Direct Connection	eBGP over OSPF	eBGP over Static Route	eBGP over Direct Connection			
iBGP	iBGP over OSPF	Yes*	X	X	X	Yes* (tested in APIC release 1.3c)	X	Yes	Yes	X	Yes
	iBGP over Static Route	Yes	X	X	X	Yes* (tested in APIC release 1.2g)	X	Yes* (tested in APIC release 1.2g)	Yes	X	Yes
	iBGP over Direct Connection	Yes	X	X	X	-	X	Yes* (tested in APIC release 1.2g)	Yes	X	Yes
eBGP	eBGP over OSPF	Yes	Yes* (tested in APIC release 1.3c)	Yes* (tested in APIC release 1.3c)	Yes* (tested in APIC release 1.3c)	Yes	Yes* (tested in APIC release 1.3c)	Yes* (tested in APIC release 1.3c)	Yes	X	Yes* (tested in APIC release 1.3c)
	eBGP over Static Route	Yes	X	X	X	Yes* (tested in APIC release 1.2g)	Yes (tested in APIC release 3.0)	Yes* (tested in APIC release 1.2g)	Yes	X	Yes
	eBGP over Direct Connection	Yes	Yes	Yes	Yes* (tested in APIC release 1.3c)	Yes* (tested in APIC release 1.3c)	Yes* (tested in APIC release 1.3c)	Yes	Yes	X	Yes
EIGRPv4		Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes (tested in APIC release 1.3c)	X	Yes

Layer 3 Outside Connection Type	OSPF	iBGP			eBGP			EIGRP v4	EIGRP v6	Static Route
		iBGP over OSPF	iBGP over Static Route	iBGP over Direct Connection	eBGP over OSPF	eBGP over Static Route	eBGP over Direct Connection			
EIGRPv6	Yes (tested in APIC release 1.2g)	X	X	X	X	X	X	X	Yes (tested in APIC release 1.3c)	Yes (tested in APIC release 1.2g)
Static Route	Yes	Yes	Yes	Yes	Yes (tested in APIC release 1.3c)	Yes	Yes	Yes	Yes (tested in APIC release 1.2g)	Yes

- connec. = connection
- * = Not supported on the same leaf switch
- X = Unsupported/Untested combinations

Transit Routing Guidelines

Guidelines for Transit Routing

Use the following guidelines when creating and maintaining transit routing connections:

Topic	Caution or Guideline
OSPF/EIGRP Redistribution into ACI Fabric iBGP when Transit Routing across Multiple VRFs - Route Tags	In a transit routing scenario where external routers are used to route between multiple VRFs, and when an entry other than the default route tag (4294967295) is used to identify the policy in different VRFs, there is a risk of routing loops when there's one or more routes withdrawn from a tenant L3Out in OSPF or EIGRP. This is expected behavior. Upon the EIGRP/OSPF redistribution of routes into the ACI fabric, the default iBGP anti-routing loop mechanisms on the border leaf switches either use the specific default route tag 4294967295 or they use the same tag that is assigned in the **Transit Route Tag Policy** field in the **VRF/Policy** page. If you configure a different, specific transit route tag for each VRF, the default anti-routing loop mechanism does not work. In order to avoid this situation, use the same value for the **Transit Route Tag Policy** field across all VRFs. For additional details regarding route-maps and tags usage, see the row for "OSPF or EIGRP in Back to Back Configuration" and other information on route control profile policies in this table. **Note** The route tag policy is configured in the **Create Route Tag Policy** page, which is accessed through the **Transit Route Tag Policy** field in the **VRF/Policy** page: **Tenants** > ***tenant_name*** > **Networking** > **VRFs** > ***VRF_name***

Topic	Caution or Guideline
Transit Routing with a Single L3Out Profile	Before Cisco APIC release 2.3(1f), transit routing was not supported within a single L3Out profile. In Cisco APIC release 2.3(1f) and later, you can configure transit routing with a single L3Out profile, with the following limitations: • If the VRF instance is unenforced, you can use an external subnet (l3extSubnet) of 0.0.0.0/0 to allow traffic between the routers sharing the same Layer 3 EPG. • If the VRF instance is enforced, you cannot use an external default subnet (0.0.0.0/0) 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, Cisco APIC supports the following prefixes: • **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 You do not need a contract for intra-Layer 3 EPG forwarding. • Alternatively, you can use a single default subnet (0.0.0.0/0) when combined with a contract that has at least one other EPG (application or external). You cannot use vzAny as a replacement for this EPG. However, you do not need to deploy the other EPG anywhere. As an example, use an application EPG provided contract and a Layer 3 EPG consumed contract (matching 0.0.0.0/0) or an application EPG consumed contract and a Layer 3 EPG provided contract (matching 0.0.0.0/0).
Shared Routes: Differences in Hardware Support	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.

Topic	Caution or Guideline
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*** > **Policies** > **Protocol** > **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.
Advertising BD Subnets Outside the Fabric	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: 1. Configure the tenant subnet scope as **Public Subnet** in the subnet properties window. 2. Optional. Set the Subnet Control as **ND RA Prefix** in the subnet properties window. 3. Associate the tenant bridge domain (BD) with the external Layer 3 Outside (L3Out). 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.

Topic	Caution or Guideline
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 configured 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) and Open Shortest Path First (OSPF) when Import Route Control Enforcement is configured.

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 allowed list model where the default behavior is to drop all data plane traffic between EPGs, unless it is explicitly permitted by a contract. The allowed list 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.

Property	OSPF	EIGRP	BGP	Comments
Set Community	X	X	Yes	Supports regular and extended communities.
Route Tag	Yes	Yes	X	Supported only for BD subnets. Transit prefixes are always assigned the tag 4294967295.
Preference	X	X	Yes	Sets BGP local preference.
Metric	Yes	X	Yes	Sets MED for BGP and changes the metric for EIGRP, but you cannot specify the EIGRP composite metric.
Metric Type	Yes	X	X	OSPF Type-1 and OSPF Type-2.

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 11: Action Rule Profile Properties (route-map set clauses)

Property	OSPF	EIGRP	BGP	Comments
Set Community			Yes	Supports regular and extended communities.
Set Additional Community			Yes	Supports regular and extended communities.
Route Tag	Yes	Yes		Supported only for BD subnets. Transit prefixes are always assigned the tag 4294967295.
Preference			Yes	Sets BGP local preference.
Metric	Yes		Yes	Sets MED for BGP. Will change the metric for EIGRP but you cannot specify the EIGRP composite metric.
Metric Type	Yes			OSPF Type-1 and OSPF Type-2.

The Route Profile Polices 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 allowed list 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 allowed list 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 aggragate 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.

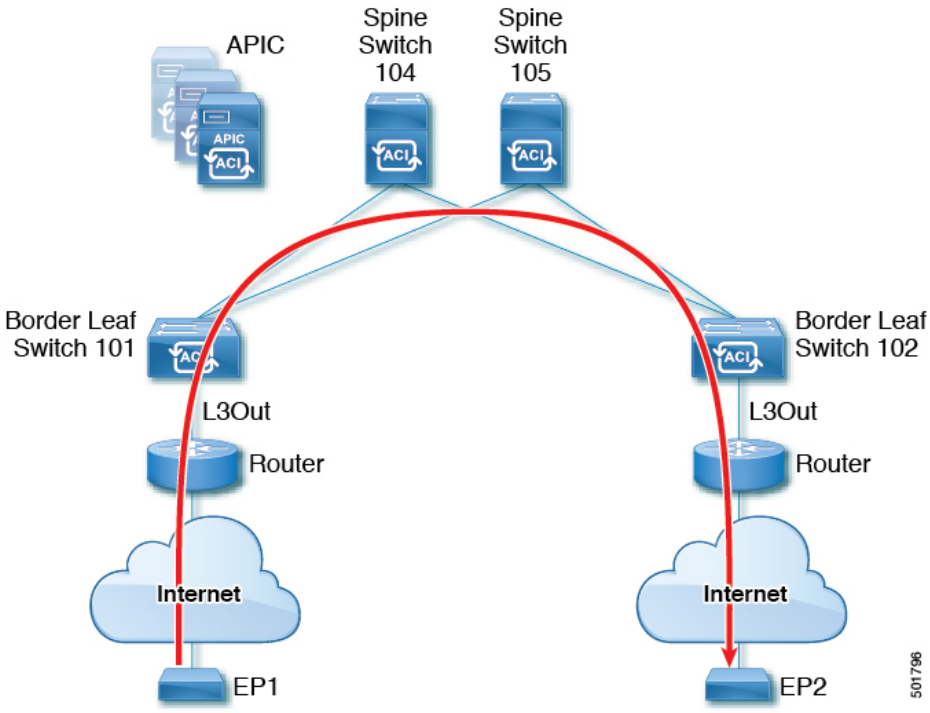
Configuring Transit Routing

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:

Figure 47:



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 `t1`, with VRF `v1`.

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

Note For transit routing cautions and guidelines, see Guidelines for Transit Routing, on page 391.

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

Property	Names for L3Out1 on Node 101	Names for L3Out2 on Node 102
Tenant	`t1`	`t1`
VRF	`v1`	`v1`
Node	`nodep1` with router ID 11.11.11.103	`nodep2` with router ID 22.22.22.203
OSPF Interface	`ifp1` at `eth/1/3`	`ifp2` at `eth/1/3`
BGP peer address	`15.15.15.2/24`	`25.25.25.2/24`
External EPG	`extnw1` at `192.168.1.0/24`	`extnw2` at `192.168.2.0/24`
Route map	`rp1` with `ctx1` and route destination `192.168.1.0/24`	`rp2` with `ctx2` and route destination `192.168.2.0/24`
Filter	`http-filter`	`http-filter`
Contract	`httpCtrct` provided by `extnw1`	`httpCtrct` consumed by `extnw2`

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 34.

Procedure

Step 1 Configure the tenant and VRF.

This example configures tenant `t1` and VRF `v1`. The VRF is not yet deployed.

Example:

```
<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:

```
<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]"/>
        </l3extLIfP>
    </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="23.23.23.3/24" ifInstT="l3-port"
tDn="topology/pod-1/paths-102/pathep-[eth1/3]"/>
```

```
        </l3extLIfP>
    </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:

```
<l3extOut name="l3out1">
    <l3extLNodeP name="nodep1">
        <bgpPeerP addr="15.15.15.2/24"
            <bgpAsP asn="100"/>
        </bgpPeerP>
    </l3extLNodeP>
</l3extOut>

<l3extOut name="l3out2">
    <l3extLNodeP name="nodep2">
        <bgpPeerP addr="25.25.25.2/24"
            <bgpAsP asn="100"/>
        </bgpPeerP>
    </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:

```
<l3extOut name="l3out1">
    <ospfExtP areaId="0.0.0.0" areaType="regular"/>
    <l3extLNodeP name="nodep1">
        <l3extLIfP name="ifp1">
            <ospfIfP/>
        <l3extIfP>
    <l3extLNodeP>
</l3extOut>
<l3extOut name="l3out2">
    <ospfExtP areaId="0.0.0.0" areaType="regular"/>
    <l3extLNodeP name="nodep2">
        <l3extLIfP name="ifp2">
            <ospfIfP/>
        <l3extIfP>
    <l3extLNodeP>
</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:

```
<l3extOut name="l3out1">
     <l3extInstP name="extnw1">
          <l3extSubnet ip="192.168.1.0/24" scope="import-security"/>
          <l3extRsInstPToProfile direction="export" tnRtctrlProfileName="rp1"/>
    </l3extInstP>
</l3extOut>
```

```
<l3extOut name="l3out2">
     <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:

```
<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">
        <rtctrlProfile name="rp1">
            <rtctrlCtxP name="ctxp1" action="permit" order="0">
                <rtctrlRsCtxPToSubjP tnRtctrlSubjPName="match-rule1" />
            </rtctrlCtxP>
        </rtctrlProfile>
        <rtctrlProfile name="rp2">
            <rtctrlCtxP name="ctxp1" action="permit" order="0">
                <rtctrlRsCtxPToSubjP tnRtctrlSubjPName="match-rule2" />
            </rtctrlCtxP>
        </rtctrlProfile>
        <l3extInstP name="extnw1">
            <l3extRsInstPToProfile direction="import" tnRtctrlProfileName="rp1" />
            <l3extRsInstPToProfile direction="export" tnRtctrlProfileName="rp2" />
        </l3extInstP>
    </l3extOut>
    <l3extOut name="l3out2">
        <rtctrlProfile name="rp1">
            <rtctrlCtxP name="ctxp1" action="permit" order="0">
                <rtctrlRsCtxPToSubjP tnRtctrlSubjPName="match-rule1" />
            </rtctrlCtxP>
        </rtctrlProfile>
        <rtctrlProfile name="rp2">
            <rtctrlCtxP name="ctxp1" action="permit" order="0">
                <rtctrlRsCtxPToSubjP tnRtctrlSubjPName="match-rule2" />
            </rtctrlCtxP>
        </rtctrlProfile>
        <l3extInstP name="extnw2">
            <l3extRsInstPToProfile direction="import" tnRtctrlProfileName="rp2" />
            <l3extRsInstPToProfile direction="export" tnRtctrlProfileName="rp1" />
        </l3extInstP>
    </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.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- api/policymgr/mo/.xml -->
<polUni>
    <fvTenant name="t1">
        <fvCtx name="v1"/>
        <l3extOut name="l3out1">
            <l3extRsEctx tnFvCtxName="v1"/>
            <l3extLNodeP name="nodep1">
                <bgpPeerP addr="15.15.15.2/24">
                    <bgpAsP asn="100"/>
                </bgpPeerP>
                <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]" />
                    <ospfIfP/>
                </l3extLIfP>
            </l3extLNodeP>
            <l3extInstP name="extnw1">
                <l3extSubnet ip="192.168.1.0/24" scope="import-security"/>
                <l3extRsInstPToProfile direction="import" tnRtctrlProfileName="rp1"/>
                <l3extRsInstPToProfile direction="export" tnRtctrlProfileName="rp2"/>
                <fvRsProv tnVzBrCPName="httpCtrct"/>
            </l3extInstP>
            <bgpExtP/>
            <ospfExtP areaId="0.0.0.0" areaType="regular"/>
            <l3extRsL3DomAtt tDn="uni/l3dom-dom1"/>
            <rtctrlProfile name="rp1">
                <rtctrlCtxP name="ctxp1" action="permit" order="0">
                    <rtctrlRsCtxPToSubjP tnRtctrlSubjPName="match-rule1"/>
                </rtctrlCtxP>
            </rtctrlProfile>
```

```
                <rtctrlProfile name="rp2">
                    <rtctrlCtxP name="ctxp1" action="permit" order="0">
                        <rtctrlRsCtxPToSubjP tnRtctrlSubjPName="match-rule2"/>
                    </rtctrlCtxP>
                </rtctrlProfile>
            </l3extOut>
            <l3extOut name="l3out2">
                <l3extRsEctx tnFvCtxName="v1"/>
                <l3extLNodeP name="nodep2">
                    <bgpPeerP addr="25.25.25.2/24">
                        <bgpAsP asn="100"/>
                    </bgpPeerP>
                    <l3extRsNodeL3OutAtt rtrId="22.22.22.203" tDn="topology/pod-1/node-102" />
                    <l3extLIfP name="ifp2">
                        <l3extRsPathL3OutAtt addr="23.23.23.3/24" ifInstT="l3-port"
tDn="topology/pod-1/paths-102/pathep-[eth1/3]" />
                        <ospfIfP/>
                    </l3extLIfP>
                </l3extLNodeP>
                <l3extInstP name="extnw2">
                    <l3extSubnet ip="192.168.2.0/24" scope="import-security"/>
                    <l3extRsInstPToProfile direction="import" tnRtctrlProfileName="rp2"/>
                    <l3extRsInstPToProfile direction="export" tnRtctrlProfileName="rp1"/>
                    <fvRsCons tnVzBrCPName="httpCtrct"/>
                </l3extInstP>
                <bgpExtP/>
                <ospfExtP areaId="0.0.0.0" areaType="regular"/>
                <l3extRsL3DomAtt tDn="uni/l3dom-dom1"/>
                <rtctrlProfile name="rp1">
                    <rtctrlCtxP name="ctxp1" action="permit" order="0">
                        <rtctrlRsCtxPToSubjP tnRtctrlSubjPName="match-rule1"/>
                    </rtctrlCtxP>
                </rtctrlProfile>
                <rtctrlProfile name="rp2">
                    <rtctrlCtxP name="ctxp1" action="permit" order="0">
                        <rtctrlRsCtxPToSubjP tnRtctrlSubjPName="match-rule2"/>
                    </rtctrlCtxP>
                </rtctrlProfile>
            </l3extOut>
            <rtctrlSubjP name="match-rule1">
                <rtctrlMatchRtDest ip="192.168.1.0/24"/>
            </rtctrlSubjP>
            <rtctrlSubjP name="match-rule2">
                <rtctrlMatchRtDest ip="192.168.2.0/24"/>
            </rtctrlSubjP>
            <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>
        </fvTenant>
</polUni>
```

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 40.

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:

- 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
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)# 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)# 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
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
apic1(config)# leaf 102
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)# exit
apic1(config-leaf-bgp-vrf)# exit
apic1(config-leaf-bgp)# exit
apic1(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
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 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)# router ospf default
apic1(config-leaf-ospf)# vrf member tenant t1 vrf v1
apic1(config-leaf-ospf-vrf)# area 0.0.0.0 loopback 60.60.60.1
apic1(config-leaf-ospf-vrf)# exit
apic1(config-leaf-ospf)# exit
apic1(config-leaf)# exit
```

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:

```
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:

Example:

```
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-match)# exit
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-match)# 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-match)# exit
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-match)# 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 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
apic1(config-tenant-l3ext-epg)# exit
apic1(config-tenant)# exit
apic1(config)#
```

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.

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

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-vrf)# 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 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)# router ospf default
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

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

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-match)# exit
```

```
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-match)# 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-rule1 order 0
apic1(config-leaf-vrf-route-map-match)# exit
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-match)# 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 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

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 http-subj
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

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
```

```
apic1(config-tenant-l3ext-epg)# exit
apic1(config-tenant)# exit
apic1(config)#
```

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 383.

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**.