Cisco Application Centric Infrastructure Fundamentals, Release 5.1(x)

First Published: 2020-10-22

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.

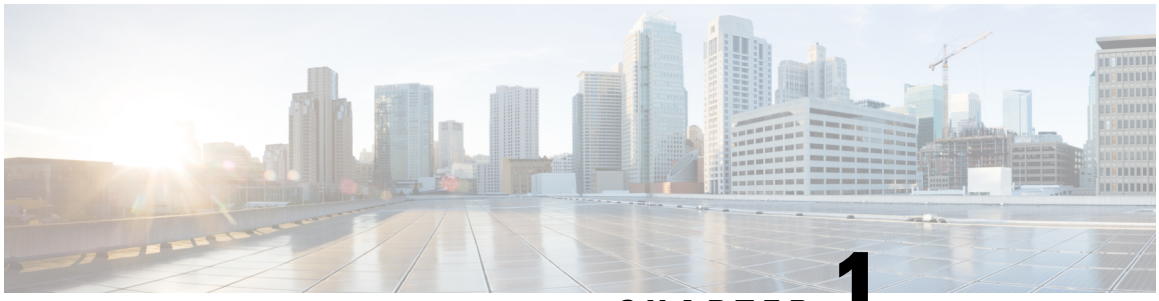
Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual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

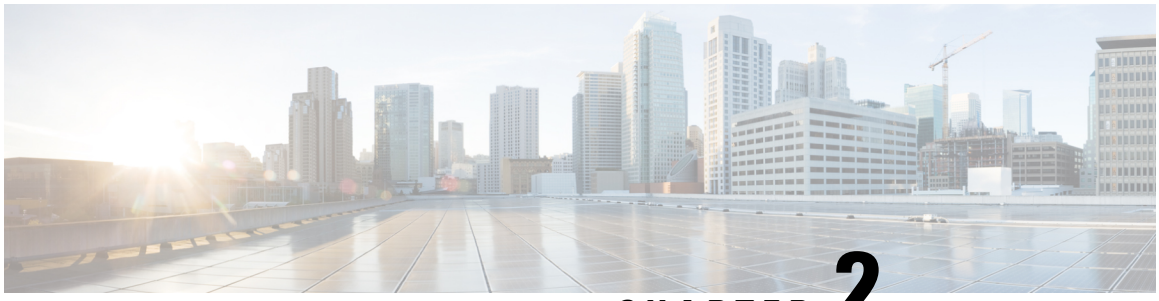
New and Changed Information

The following tables provide an overview of the significant changes to 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 Information for Cisco APIC Release 5.1(1)

Feature	Description	Where Documented
N/A	This document has no changes from the previous release.	N/A

New and Changed Information



CHAPTER 2

Cisco Application Centric Infrastructure

This chapter contains the following sections:

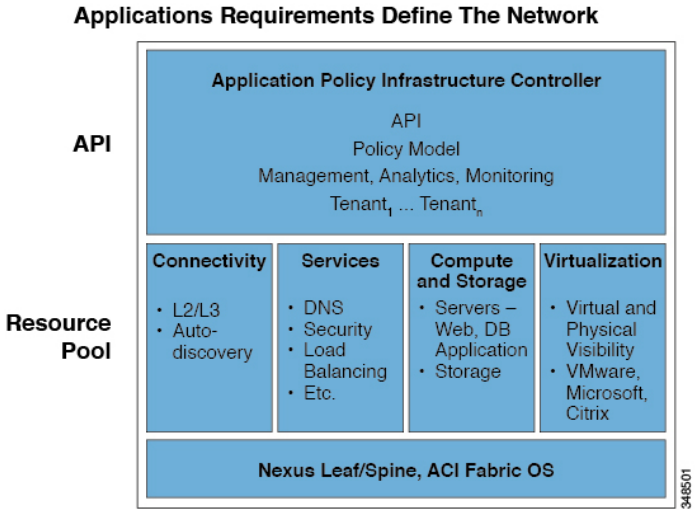
About the Cisco Application Centric Infrastructure

The Cisco Application Centric Infrastructure (ACI) allows application requirements to define the network. This architecture simplifies, optimizes, and accelerates the entire application deployment life cycle.

About the Cisco Application Policy Infrastructure Controller

The Cisco Application Policy Infrastructure Controller (APIC) API enables applications to directly connect with a secure, shared, high-performance resource pool that includes network, compute, and storage capabilities. The following figure provides an overview of the APIC.

Figure 1: APIC Overview

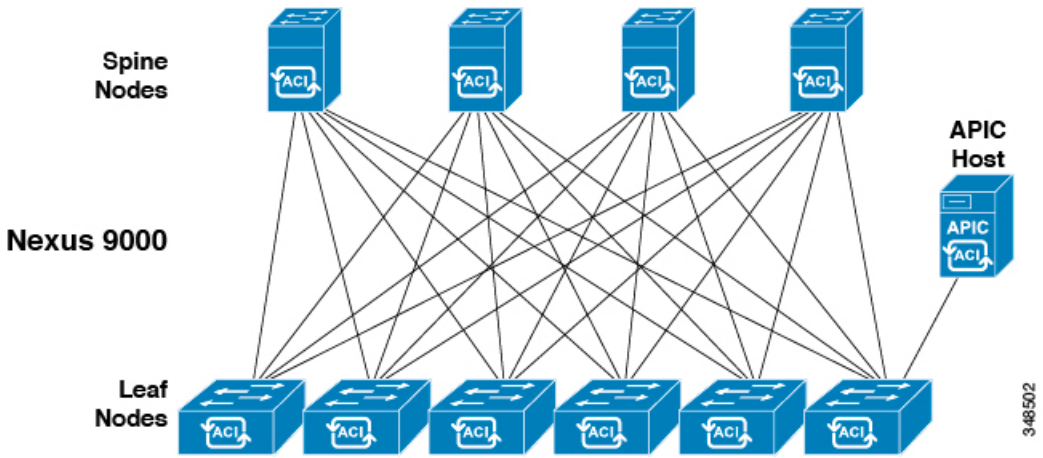


The APIC manages the scalable ACI multi-tenant fabric. The APIC provides a unified point of automation and management, policy programming, application deployment, and health monitoring for the fabric. The APIC, which is implemented as a replicated synchronized clustered controller, optimizes performance, supports any application anywhere, and provides unified operation of the physical and virtual infrastructure. The APIC enables network administrators to easily define the optimal network for applications. Data center operators can clearly see how applications consume network resources, easily isolate and troubleshoot application and infrastructure problems, and monitor and profile resource usage patterns.

Cisco Application Centric Infrastructure Fabric Overview

The Cisco Application Centric Infrastructure Fabric (ACI) fabric includes Cisco Nexus 9000 Series switches with the APIC to run in the leaf/spine ACI fabric mode. These switches form a "fat-tree" network by connecting each leaf node to each spine node; all other devices connect to the leaf nodes. The APIC manages the ACI fabric. The recommended minimum configuration for the APIC is a cluster of three replicated hosts. The APIC fabric management functions do not operate in the data path of the fabric. The following figure shows an overview of the leaf/spin ACI fabric.

Figure 2: ACI Fabric Overview

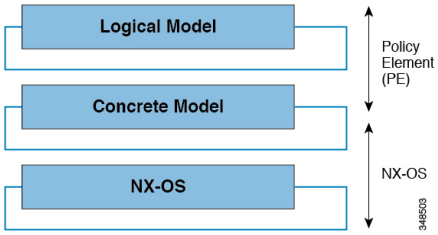


The ACI fabric provides consistent low-latency forwarding across high-bandwidth links (40 Gbps and 100-Gbps). Traffic with the source and destination on the same leaf switch is handled locally, and all other traffic travels from the ingress leaf to the egress leaf through a spine switch. Although this architecture appears as two hops from a physical perspective, it is actually a single Layer 3 hop because the fabric operates as a single Layer 3 switch.

The ACI fabric object-oriented operating system (OS) runs on each Cisco Nexus 9000 Series node. It enables programming of objects for each configurable element of the system.

The ACI fabric OS renders policies from the APIC into a concrete model that runs in the physical infrastructure. The concrete model is analogous to compiled software; it is the form of the model that the switch operating system can execute. The figure below shows the relationship of the logical model to the concrete model and the switch OS.

Figure 3: Logical Model Rendered into a Concrete Model



All the switch nodes contain a complete copy of the concrete model. When an administrator creates a policy in the APIC that represents a configuration, the APIC updates the logical model. The APIC then performs the intermediate step of creating a fully elaborated policy that it pushes into all the switch nodes where the concrete model is updated.

Note The Cisco Nexus 9000 Series switches can only execute the concrete model. Each switch has a copy of the concrete model. If the APIC goes offline, the fabric k.jpg functioning but modifications to the fabric policies are not possible.

The APIC is responsible for fabric activation, switch firmware management, network policy configuration, and instantiation. While the APIC acts as the centralized policy and network management engine for the fabric, it is completely removed from the data path, including the forwarding topology. Therefore, the fabric can still forward traffic even when communication with the APIC is lost.

The Cisco Nexus 9000 Series switches offer modular and fixed 1-, 10-, 40-, and 100-Gigabit Ethernet switch configurations that operate in either Cisco NX-OS stand-alone mode for compatibility and consistency with the current Cisco Nexus switches or in ACI mode to take full advantage of the APIC's application policy-driven services and infrastructure automation features.

Determining How the Fabric Behaves

The ACI fabric allows customers to automate and orchestrate scalable, high performance network, compute and storage resources for cloud deployments. Key players who define how the ACI fabric behaves include the following:

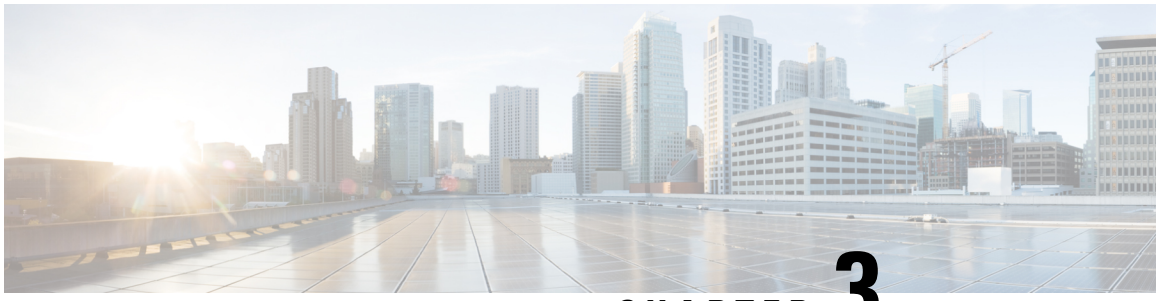
- IT planners, network engineers, and security engineers
- Developers who access the system via the APIC APIs
- Application and network administrators

The Representational State Transfer (REST) architecture is a key development method that supports cloud computing. The ACI API is REST-based. The World Wide Web represents the largest implementation of a system that conforms to the REST architectural style.

Cloud computing differs from conventional computing in scale and approach. Conventional environments include software and maintenance requirements with their associated skill sets that consume substantial operating expenses. Cloud applications use system designs that are supported by a very large scale infrastructure that is deployed along a rapidly declining cost curve. In this infrastructure type, the system administrator, development teams, and network professionals collaborate to provide a much higher valued contribution.

In conventional settings, network access for compute resources and endpoints is managed through virtual LANs (VLANs) or rigid overlays, such as Multiprotocol Label Switching (MPLS), that force traffic through rigidly defined network services, such as load balancers and firewalls. The APIC is designed for programmability and centralized management. By abstracting the network, the ACI fabric enables operators to dynamically provision resources in the network instead of in a static fashion. The result is that the time to deployment (time to market) can be reduced from months or weeks to minutes. Changes to the configuration of virtual or physical switches, adapters, policies, and other hardware and software components can be made in minutes with API calls.

The transformation from conventional practices to cloud computing methods increases the demand for flexible and scalable services from data centers. These changes call for a large pool of highly skilled personnel to enable this transformation. The APIC is designed for programmability and centralized management. A key feature of the APIC is the web API called REST. The APIC REST API accepts and returns HTTP or HTTPS messages that contain JavaScript Object Notation (JSON) or Extensible Markup Language (XML) documents. Today, many web developers use RESTful methods. Adopting web APIs across the network enables enterprises to easily open up and combine services with other internal or external providers. This process transforms the network from a complex mixture of static resources to a dynamic exchange of services on offer.



CHAPTER 3

ACI Policy Model

This chapter contains the following sections:

About the ACI Policy Model

The ACI policy model enables the specification of application requirements policies. The APIC automatically renders policies in the fabric infrastructure. When a user or process initiates an administrative change to an object in the fabric, the APIC first applies that change to the policy model. This policy model change then triggers a change to the actual managed endpoint. This approach is called a model-driven framework.

Policy Model Key Characteristics

Key characteristics of the policy model include the following:

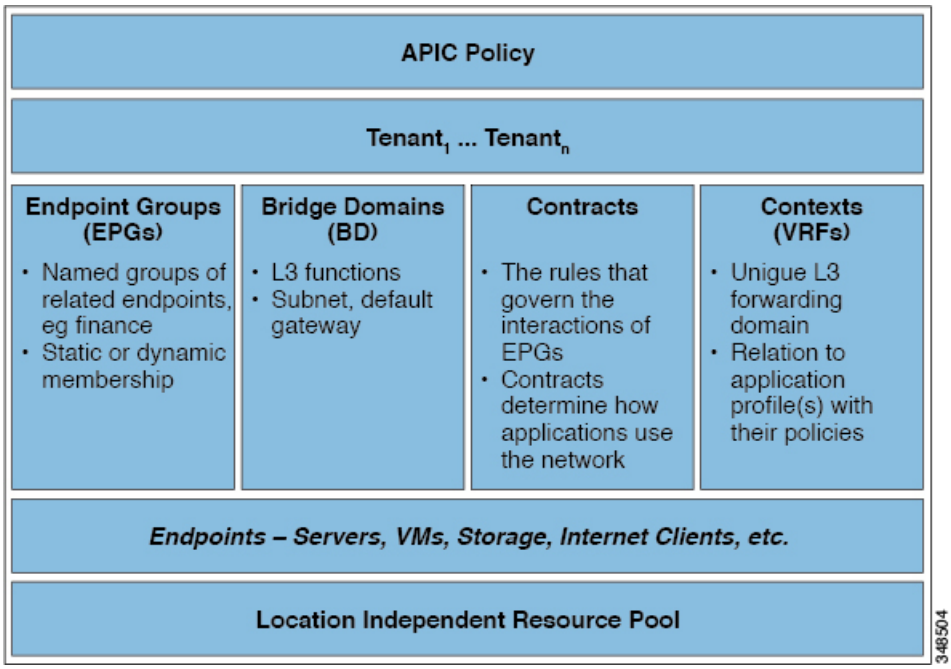
- As a model-driven architecture, the software maintains a complete representation of the administrative and operational state of the system (the model). The model applies uniformly to fabric, services, system behaviors, and virtual and physical devices attached to the network.
- The logical and concrete domains are separated; the logical configurations are rendered into concrete configurations by applying the policies in relation to the available physical resources. No configuration is carried out against concrete entities. Concrete entities are configured implicitly as a side effect of the changes to the APIC policy model. Concrete entities can be, but do not have to be, physical (such as a virtual machine or a VLAN).
- The system prohibits communications with newly connected devices until the policy model is updated to include the new device.
- Network administrators do not configure logical and physical system resources directly but rather define logical (hardware independent) configurations and APIC policies that control different aspects of the system behavior.

Managed object manipulation in the model relieves engineers from the task of administering isolated, individual component configurations. These characteristics enable automation and flexible workload provisioning that can locate any workload anywhere in the infrastructure. Network-attached services can be easily deployed, and the APIC provides an automation framework to manage the life cycle of those network-attached services.

Logical Constructs

The policy model manages the entire fabric, including the infrastructure, authentication, security, services, applications, and diagnostics. Logical constructs in the policy model define how the fabric meets the needs of any of the functions of the fabric. The following figure provides an overview of the ACI policy model logical constructs.

Figure 4: ACI Policy Model Logical Constructs Overview



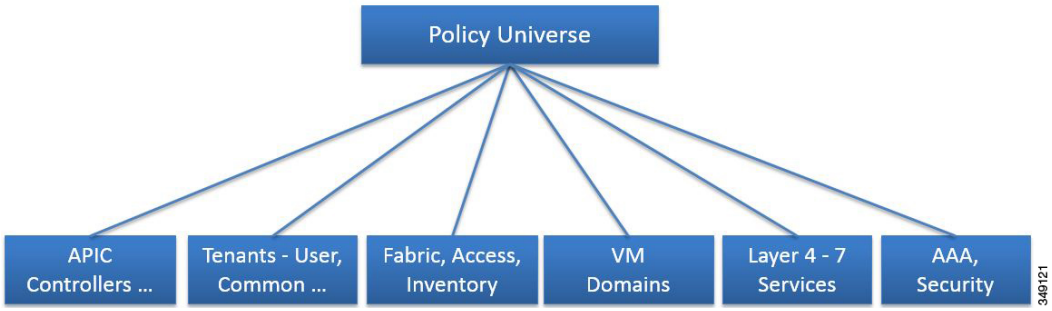
Fabric-wide or tenant administrators create predefined policies that contain application or shared resource requirements. These policies automate the provisioning of applications, network-attached services, security policies, and tenant subnets, which puts administrators in the position of approaching the resource pool in terms of applications rather than infrastructure building blocks. The application needs to drive the networking behavior, not the other way around.

The Cisco ACI Policy Management Information Model

The fabric comprises the physical and logical components as recorded in the Management Information Model (MIM), which can be represented in a hierarchical management information tree (MIT). The information model is stored and managed by processes that run on the APIC. Similar to the OSI Common Management Information Protocol (CMIP) and other X.500 variants, the APIC enables the control of managed resources by presenting their manageable characteristics as object properties that can be inherited according to the location of the object within the hierarchical structure of the MIT.

Each node in the tree represents a managed object (MO) or group of objects. MOs are abstractions of fabric resources. An MO can represent a concrete object, such as a switch, adapter, or a logical object, such as an application profile, endpoint group, or fault. The following figure provides an overview of the MIT.

Figure 5: Cisco ACI Policy Management Information Model Overview



The hierarchical structure starts with the policy universe at the top (Root) and contains parent and child nodes. Each node in the tree is an MO and each object in the fabric has a unique distinguished name (DN) that describes the object and locates its place in the tree.

The following managed objects contain the policies that govern the operation of the system:

- APIC controllers comprise a replicated synchronized clustered controller that provides management, policy programming, application deployment, and health monitoring for the multitenant fabric.
- A tenant is a container for policies that enable an administrator to exercise domain-based access control. The system provides the following four kinds of tenants:
  - User tenants are defined by the administrator according to the needs of users. They contain policies that govern the operation of resources such as applications, databases, web servers, network-attached storage, virtual machines, and so on.
  - The common tenant is provided by the system but can be configured by the fabric administrator. It contains policies that govern the operation of resources accessible to all tenants, such as firewalls, load balancers, Layer 4 to Layer 7 services, intrusion detection appliances, and so on.
  - The infrastructure tenant is provided by the system but can be configured by the fabric administrator. It contains policies that govern the operation of infrastructure resources such as the fabric VXLAN

overlay. It also enables a fabric provider to selectively deploy resources to one or more user tenants. Infrastructure tenant polices are configurable by the fabric administrator.

- The management tenant is provided by the system but can be configured by the fabric administrator. It contains policies that govern the operation of fabric management functions used for in-band and out-of-band configuration of fabric nodes. The management tenant contains a private out-of-bound address space for the APIC/fabric internal communications that is outside the fabric data path that provides access through the management port of the switches. The management tenant enables discovery and automation of communications with virtual machine controllers.

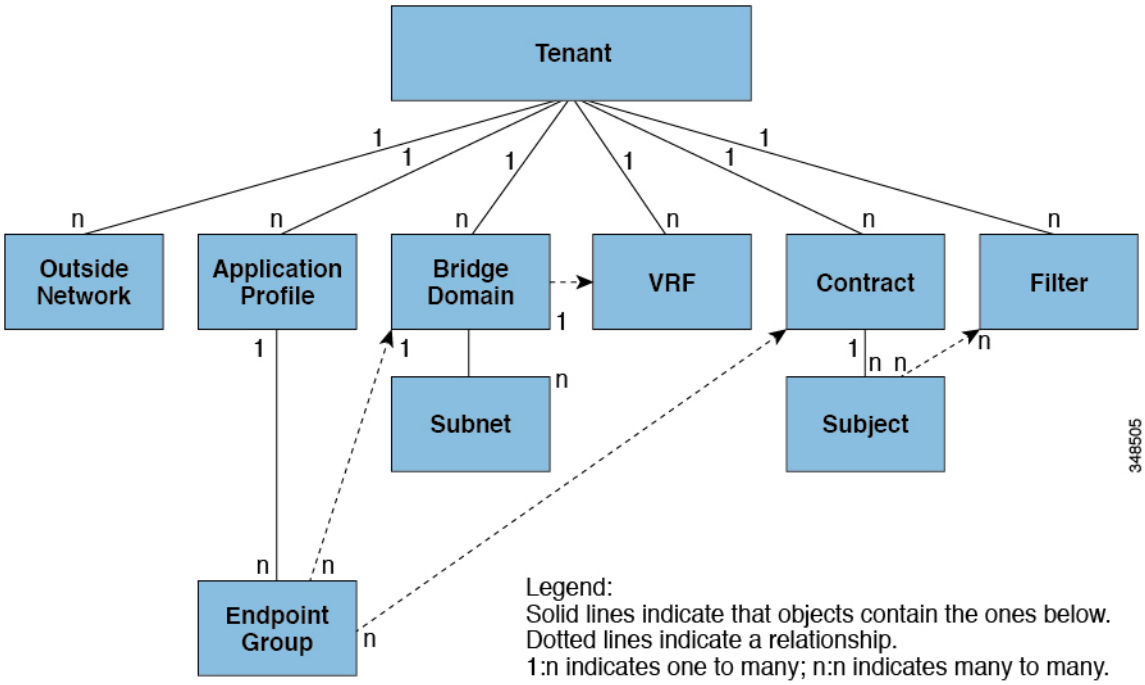
- Access policies govern the operation of switch access ports that provide connectivity to resources such as storage, compute, Layer 2 and Layer 3 (bridged and routed) connectivity, virtual machine hypervisors, Layer 4 to Layer 7 devices, and so on. If a tenant requires interface configurations other than those provided in the default link, Cisco Discovery Protocol (CDP), Link Layer Discovery Protocol (LLDP), Link Aggregation Control Protocol (LACP), or Spanning Tree, an administrator must configure access policies to enable such configurations on the access ports of the leaf switches.
- Fabric policies govern the operation of the switch fabric ports, including such functions as Network Time Protocol (NTP) server synchronization, Intermediate System-to-Intermediate System Protocol (IS-IS), Border Gateway Protocol (BGP) route reflectors, Domain Name System (DNS) and so on. The fabric MO contains objects such as power supplies, fans, chassis, and so on.
- Virtual Machine (VM) domains group VM controllers with similar networking policy requirements. VM controllers can share VLAN or Virtual Extensible Local Area Network (VXLAN) space and application endpoint groups (EPGs). The APIC communicates with the VM controller to publish network configurations such as port groups that are then applied to the virtual workloads.
- Layer 4 to Layer 7 service integration life cycle automation framework enables the system to dynamically respond when a service comes online or goes offline. Policies provide service device package and inventory management functions.
- Access, authentication, and accounting (AAA) policies govern user privileges, roles, and security domains of the Cisco ACI fabric.

The hierarchical policy model fits well with the REST API interface. When invoked, the API reads from or writes to objects in the MIT. URLs map directly into distinguished names that ide.jpgy objects in the MIT. Any data in the MIT can be described as a self-contained structured tree text document encoded in XML or JSON.

Tenants

A tenant (`fvTenant`) is a logical container for application policies that enable an administrator to exercise domain-based access control. A tenant represents a unit of isolation from a policy perspective, but it does not represent a private network. Tenants can represent a customer in a service provider setting, an organization or domain in an enterprise setting, or just a convenient grouping of policies. The following figure provides an overview of the tenant portion of the management information tree (MIT).

Figure 6: Tenants



Tenants can be isolated from one another or can share resources. The primary elements that the tenant contains are filters, contracts, outside networks, bridge domains, Virtual Routing and Forwarding (VRF) instances, and application profiles that contain endpoint groups (EPGs). Entities in the tenant inherit its policies. VRFs are also known as contexts; each VRF can be associated with multiple bridge domains.

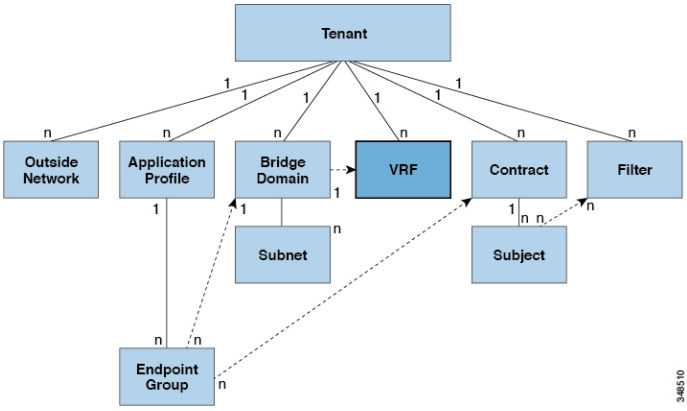
Note In the APIC GUI under the tenant navigation path, a VRF (context) is called a private network.

Tenants are logical containers for application policies. The fabric can contain multiple tenants. You must configure a tenant before you can deploy any Layer 4 to Layer 7 services. The ACI fabric supports IPv4, IPv6, and dual-stack configurations for tenant networking.

VRFs

A Virtual Routing and Forwarding (VRF) object (`fvCtx`) or context is a tenant network (called a private network in the APIC GUI). A tenant can have multiple VRFs. A VRF is a unique Layer 3 forwarding and application policy domain. The following figure shows the location of VRFs in the management information tree (MIT) and their relation to other objects in the tenant.

Figure 7: VRFs



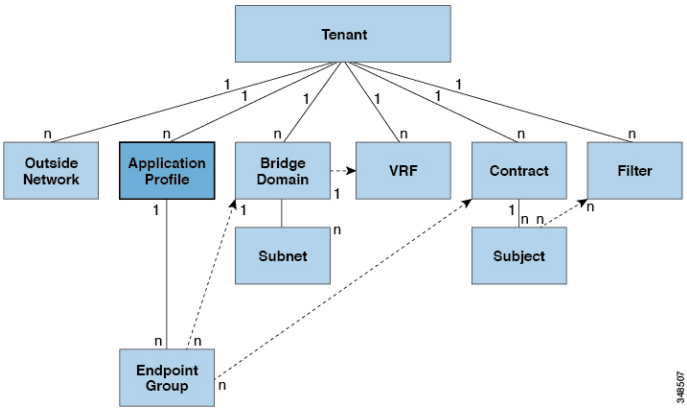
A VRF defines a Layer 3 address domain. One or more bridge domains are associated with a VRF. All of the endpoints within the Layer 3 domain must have unique IP addresses because it is possible to forward packets directly between these devices if the policy allows it. A tenant can contain multiple VRFs. After an administrator creates a logical device, the administrator can create a VRF for the logical device, which provides a selection criteria policy for a device cluster. A logical device can be selected based on a contract name, a graph name, or the function node name inside the graph.

Note In the APIC GUI, a VRF (`fvCtx`) is also called a "Context" or "Private Network."

Application Profiles

An application profile (`fvAp`) defines the policies, services and relationships between endpoint groups (EPGs). The following figure shows the location of application profiles in the management information tree (MIT) and their relation to other objects in the tenant.

Figure 8: Application Profiles



Application profiles contain one or more EPGs. Modern applications contain multiple components. For example, an e-commerce application could require a web server, a database server, data located in a storage

area network, and access to outside resources that enable financial transactions. The application profile contains as many (or as few) EPGs as necessary that are logically related to providing the capabilities of an application.

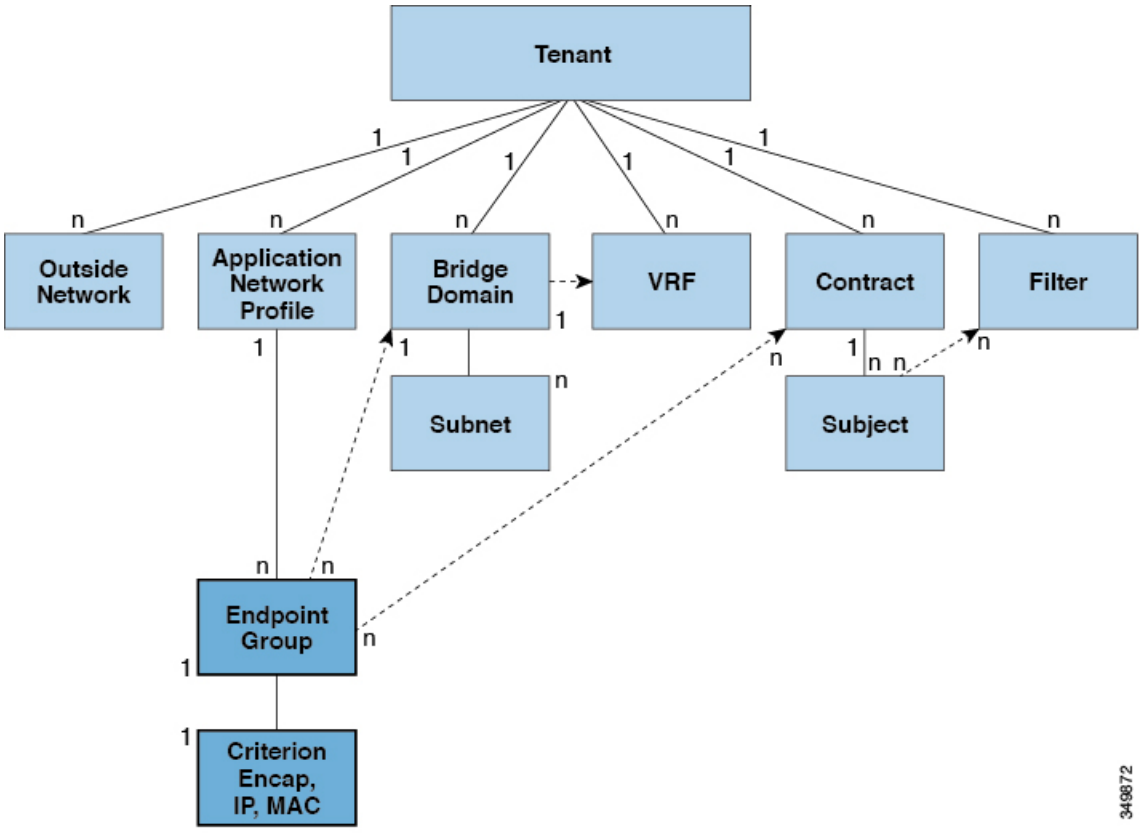
EPGs can be organized according to one of the following:

- The application they provide, such as a DNS server or SAP application (see *Tenant Policy Example* in *Cisco APIC REST API Configuration Guide*).
- The function they provide (such as infrastructure)
- Where they are in the structure of the data center (such as DMZ)
- Whatever organizing principle that a fabric or tenant administrator chooses to use

Endpoint Groups

The endpoint group (EPG) is the most important object in the policy model. The following figure shows where application EPGs are located in the management information tree (MIT) and their relation to other objects in the tenant.

Figure 9: Endpoint Groups



An EPG is a managed object that is a named logical entity that contains a collection of endpoints. Endpoints are devices that are connected to the network directly or indirectly. They have an address (identity), a location, attributes (such as version or patch level), and can be physical or virtual. Knowing the address of an endpoint

also enables access to all its other identity details. EPGs are fully decoupled from the physical and logical topology. Endpoint examples include servers, virtual machines, network-attached storage, or clients on the Internet. Endpoint membership in an EPG can be dynamic or static.

The ACI fabric can contain the following types of EPGs:

- Application endpoint group (`fvAEPg`)
- Layer 2 external outside network instance endpoint group (`l2extInstP`)
- Layer 3 external outside network instance endpoint group (`l3extInstP`)
- Management endpoint groups for out-of-band (`mgmtOoB`) or in-band (`mgmtInB`) access.

EPGs contain endpoints that have common policy requirements such as security, virtual machine mobility (VMM), QoS, or Layer 4 to Layer 7 services. Rather than configure and manage endpoints individually, they are placed in an EPG and are managed as a group.

Policies apply to EPGs, never to individual endpoints. An EPG can be statically configured by an administrator in the APIC, or dynamically configured by an automated system such as vCenter or OpenStack.

Note When an EPG uses a static binding path, the encapsulation VLAN associated with this EPG must be part of a static VLAN pool. For IPv4/IPv6 dual-stack configurations, the IP address property is contained in the `fvStIp` child property of the `fvStCEp` MO. Multiple `fvStIp` objects supporting IPv4 and IPv6 addresses can be added under one `fvStCEp` object. When upgrading ACI from IPv4-only firmware to versions of firmware that support IPv6, the existing IP property is copied to an `fvStIp` MO.

Regardless of how an EPG is configured, EPG policies are applied to the endpoints they contain.

WAN router connectivity to the fabric is an example of a configuration that uses a static EPG. To configure WAN router connectivity to the fabric, an administrator configures an `l3extInstP` EPG that includes any endpoints within an associated WAN subnet. The fabric learns of the EPG endpoints through a discovery process as the endpoints progress through their connectivity life cycle. Upon learning of the endpoint, the fabric applies the `l3extInstP` EPG policies accordingly. For example, when a WAN connected client initiates a TCP session with a server within an application (`fvAEPg`) EPG, the `l3extInstP` EPG applies its policies to that client endpoint before the communication with the `fvAEPg` EPG web server begins. When the client server TCP session ends and communication between the client and server terminate, that endpoint no longer exists in the fabric.

Note If a leaf switch is configured for *static binding (leaf switches)* under an EPG, the following restrictions apply:

- The static binding cannot be overridden with a static path.
- Interfaces in that switch cannot be used for routed external network (L3out) configurations.
- Interfaces in that switch cannot be assigned IP addresses.

Virtual machine management connectivity to VMware vCenter is an example of a configuration that uses a dynamic EPG. Once the virtual machine management domain is configured in the fabric, vCenter triggers the dynamic configuration of EPGs that enable virtual machine endpoints to start up, move, and shut down as needed.

IP-Based EPGs

Although encapsulation-based EPGs are commonly used, IP-based EPGs are suitable in networks where there is a need for large numbers of EPGs that cannot be supported by Longest Prefix Match (LPM) classification. IP-based EPGs do not require allocating a network/mask range for each EPG, unlike LPM classification. Also, a unique bridge domain is not required for each IP-based EPG. The configuration steps for an IP-based EPG are like those for configuring a virtual IP-based EPG that is used in the Cisco AVS vCenter configuration.

Observe the following guidelines and limitations of IP-based EPGs:

- IP-based EPGs are supported starting with the APIC 1.1(2x) and ACI switch 11.1(2x) releases on the following Cisco Nexus N9K switches:
  - Switches with "E" on the end of the switch name, for example, N9K-C9372PX-E.
  - Switches with "EX" on the end of the switch name, for example, N9K-93108TC-EX.

  The APIC raises a fault when you attempt to deploy IP-based EPGs on older switches that do not support them.
- IP-based EPGs can be configured for specific IP addresses or subnets, but not IP address ranges.
- IP-based EPGs are not supported in the following scenarios:
  - In combination with static EP configurations.
  - External, infrastructure tenant (infra) configurations will not be blocked, but they do not take effect, because there is no Layer 3 learning in this case.
  - In Layer 2-only bridge domains, IP-based EPG does not take effect, because there is no routed traffic in this case. If proxy ARP is enabled on Layer 3 bridge domains, the traffic is routed even if endpoints are in the same subnet. So IP-based EPG works in this case.
  - Configurations with a prefix that is used both for shared services and an IP-based EPG.

Microsegmentation

Microsegmentation associates endpoints from multiple EPGs into a microsegmented EPG according to virtual machine attributes, IP address, or MAC address. Virtual machine attributes include: VNic domain name, VM identifier, VM name, hypervisor identifier, VMM domain, datacenter, operating system, or custom attribute.

Some advantages of microsegmentation include the following:

- Stateless white list network access security with line rate enforcement.
- Per-microsegment granularity of security automation through dynamic Layer 4 - Layer 7 service insertion and chaining.
- Hypervisor agnostic microsegmentation in a broad range of virtual switch environments.
- ACI policies that easily move problematic VMs into a quarantine security zone.
- When combined with intra-EPG isolation for bare metal and VM endpoints, microsegmentation can provide policy driven automated complete endpoint isolation within application tiers.

For any EPG, the ACI fabric ingress leaf switch classifies packets into an EPG according to the policies associated with the ingress port. Microsegmented EPGs apply policies to individual virtual or physical endpoints

that are derived based on the VM attribute, MAC address, or IP address specified in the microsegmented EPG policy.

Intra-EPG Endpoint Isolation

Intra-EPG endpoint isolation policies provide full isolation for virtual or physical endpoints; no communication is allowed between endpoints in an EPG that is operating with isolation enforced. Isolation enforced EPGs reduce the number of EPG encapsulations required when many clients access a common service but are not allowed to communicate with each other.

An EPG is isolation enforced for all Cisco Application Centric Infrastructure (ACI) network domains or none. While the Cisco ACI fabric implements isolation directly to connected endpoints, switches connected to the fabric are made aware of isolation rules according to a primary VLAN (PVLAN) tag.

Note If an EPG is configured with intra-EPG endpoint isolation enforced, these restrictions apply:

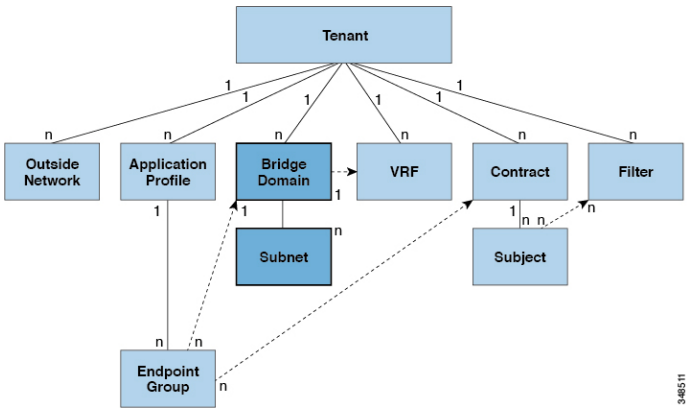
- All Layer 2 endpoint communication across an isolation enforced EPG is dropped within a bridge domain.
- All Layer 3 endpoint communication across an isolation enforced EPG is dropped within the same subnet.
- Preserving QoS CoS priority settings is not supported when traffic is flowing from an EPG with isolation enforced to an EPG without isolation enforced.

BPDUs are not forwarded through EPGs with intra-EPG isolation enabled. Therefore, when you connect an external Layer 2 network that runs spanning tree in a VLAN that maps to an isolated EPG on Cisco ACI, Cisco ACI might prevent spanning tree in the external network from detecting a Layer 2 loop. You can avoid this issue by ensuring that there is only a single logical link between Cisco ACI and the external network in these VLANs.

Bridge Domains and Subnets

A bridge domain (`fvBD`) represents a Layer 2 forwarding construct within the fabric. The following figure shows the location of bridge domains in the management information tree (MIT) and their relation to other objects in the tenant.

Figure 10: Bridge Domains



A bridge domain must be linked to a VRF instance (also known as a context or private network). With the exception of a Layer 2 VLAN, it must have at least one subnet (`fvSubnet`) associated with it. The bridge domain defines the unique Layer 2 MAC address space and a Layer 2 flood domain if such flooding is enabled. While a VRF instance defines a unique IP address space, that address space can consist of multiple subnets. Those subnets are defined in one or more bridge domains that reference the corresponding VRF instance.

The options for a subnet under a bridge domain or under an EPG are as follows:

- *Public*: The subnet can be exported to a routed connection.
- *Private*: The subnet applies only within its tenant.
- *Shared*: The subnet can be shared with and exported to multiple VRF instances in the same tenant or across tenants as part of a shared service. An example of a shared service is a routed connection to an EPG present in another VRF instance in a different tenant. This enables traffic to pass in both directions across VRF instances. An EPG that provides a shared service must have its subnet configured under that EPG (not under a bridge domain), and its scope must be set to advertised externally, and shared between VRF instances.

Note Shared subnets must be unique across the VRF instance involved in the communication. When a subnet under an EPG provides a Layer 3 external network shared service, such a subnet must be globally unique within the entire Cisco Application Centric Infrastructure (ACI) fabric.

Bridge domain packet behavior can be controlled in the following ways:

Packet Type	Mode
ARP	You can enable or disable **ARP Flooding**; without flooding, ARP packets are sent with unicast. **Note** If the `limitIpLearnToSubnets` in `fvBD` is set, endpoint learning is limited to the bridge domain only if the IP address is in a configured subnet of the bridge domain or an EPG subnet that is a shared service provider.

Packet Type	Mode
Unknown Unicast	**L2 Unknown Unicast**, which can be **Flood** or **Hardware Proxy**. **Note** When the bridge domain has **L2 Unknown Unicast** set to **Flood**, if an endpoint is deleted the system deletes it from both the local leaf switches as well as the remote leaf switches where the bridge domain is deployed, by selecting **Clear Remote MAC Entries**. Without this feature, the remote leaf continues to have this endpoint learned until the timer expires. Modifying the **L2 Unknown Unicast** setting causes traffic to bounce (go down and up) on interfaces to devices attached to EPGs associated with this bridge domain.
Unknown IP Multicast	**L3 Unknown Multicast Flooding** **Flood**: Packets are flooded on ingress and border leaf switch nodes only. With N9K-93180YC-EX, packets are flooded on all the nodes where a bridge domain is deployed. **Optimized**: Only 50 bridge domains per leaf are supported. This limitation is not applicable for N9K-93180YC-EX.
L2 Multicast, Broadcast, Unicast	**Multi-Destination Flooding**, which can be one of the following: • **Flood in BD**: Flood in bridge domain • **Flood in Encapsulation**: Flood in encapsulation • **Drop**: Drop the packets

Note Beginning with Cisco APIC release 3.1(1), on the Cisco Nexus 9000 series switches (with names ending with EX and FX and onwards), the following protocols can be flooded in encapsulation or flooded in a bridge domain: OSPF/OSPFv3, BGP, EIGRP, LACP, ISIS, IGMP, PIM, ST-BPDU, ARP/GARP, RARP, and ND.

Bridge domains can span multiple switches. A bridge domain can contain multiple subnets, but a subnet is contained within a single bridge domain. If the bridge domain (fvBD) `limitIPLearnToSubnets` property is set to `yes`, endpoint learning will occur in the bridge domain only if the IP address is within any of the configured subnets for the bridge domain or within an EPG subnet when the EPG is a shared service provider. Subnets can span multiple EPGs; one or more EPGs can be associated with one bridge domain or subnet. In hardware proxy mode, ARP traffic is forwarded to an endpoint in a different bridge domain when that endpoint has been learned as part of the Layer 3 lookup operation.

Bridge Domain Options

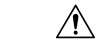
A bridge domain can be set to operate in flood mode for unknown unicast frames or in an optimized mode that eliminates flooding for these frames. When operating in flood mode, Layer 2 unknown unicast traffic is flooded over the multicast tree of the bridge domain (GIPo). For the bridge domain to operate in optimized mode you should set it to hardware-proxy. In this case, Layer 2 unknown unicast frames are sent to the spine-proxy anycast VTEP address.

Caution Changing from unknown unicast flooding mode to hw-proxy mode is disruptive to the traffic in the bridge domain.

If IP routing is enabled in the bridge domain, the mapping database learns the IP address of the endpoints in addition to the MAC address.

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.

Scaled L2 Only Mode - Legacy Mode

In Cisco Application Centric Infrastructure (ACI), the same VLAN ID can be reused for any purpose as long as the VLAN is deployed on different leaf nodes. This allows the Cisco ACI fabric to overcome the theoretical maximum number of VLANs 4094 as a fabric. However, to accomplish this, and also to hide the complexity of underlying VxLAN implementation, each individual leaf node can contain smaller number of VLANs. This may pose a problem when the density of VLANs per leaf node is required. In such a scenario, you can enable `Scaled L2 Only mode`, formerly known as legacy mode on the bridge domain. A bridge domain in scaled L2 only mode allows large number of VLANs per leaf node. However, such a bridge domain has some limitations.

For the number of VLANs or bridge domains supported per leaf node with or without scaled L2 only mode, see Verified Scalability Guide for your specific release.

Limitations for Scaled L2 Only Mode

The following are limitations for legacy mode or scaled L2 only mode.

- The bridge domain can contain only one EPG and one VLAN.
- Unicast routing is not supported.

- Contracts are not supported.
- Dynamic VLAN allocation for VMM integration is not supported.
- Service graph is not supported.
- A QoS policy is not supported.
- The bridge domain essentially behaves as a VLAN in standalone Cisco NX-OS.

Scaled L2 Only Mode Configuration

The following are considerations to configure a bridge domain in scaled L2 only mode.

- VLAN ID is configured on the bridge domain.
- VLAN IDs configured under the EPG are overridden.
- Enabling or disabling a scaled L2 only mode on an existing bridge domain will impact service.

  Cisco Application Policy Infrastructure Controller (APIC) will automatically undeploy and redeploy the bridge domain when the VLAN ID is different from what was used prior to the change.

  When the same VLAN ID is used before and after the mode change, Cisco APIC will not automatically undeploy and redeploy the bridge domain. You must manually undeploy and redeply the bridge domain, which can be performed by deleting and recreating the static port configuration under the EPG.
- When changing the VLAN ID for scaled L2 only mode, you must first disable the mode, then enable scaled L2 only mode with the new VLAN ID.

Disabling IP Learning per Bridge Domain

You can disable IP dataplane learning for a bridge domain. The MAC learning still occurs in the hardware, but the IP learning only occurs from the ARP/GARP/ND processes. This functionality was introduced in the Cisco APIC 3.1 releases primarily for service graph policy-based redirect (PBR) deployments, and it has been superseded by the ability to disable IP dataplane learning per-VRF instance (Cisco APIC release 4.0), per bridge domain subnet (Cisco APIC release 5.2), and per-EPG (Cisco APIC release 5.2). We do not recommend using this option and it is not supported except when used with PBR.

See the following guidelines and limitations for disabling IP learning per bridge domain:

- Layer 3 multicast is not supported because the source IP address is not learned to populate the S,G information in the remote leaf switches.
- As the DL bit is set in the iVXLAN header, the MAC address is also not learned from the data path in the remote leaf switches. It results in flooding of the unknown unicast traffic from the remote leaf switch to all leaf switches in the fabric where this bridge domain is deployed. We recommend that you configure the bridge domain in proxy mode to overcome this situation if endpoint dataplane learning is disabled.
- ARP should be in flood mode and GARP based detection should be enabled.
- When IP learning is disabled, Layer 3 endpoints are not flushed in the corresponding VRF instance. It may lead to the endpoints pointing to the same leaf switch forever. To resolve this issue, flush all the remote IP endpoints in this VRF on all leaf switches.

The configuration change of disabling dataplane learning on the bridge domain does not flush previously locally learned endpoints. This limits the disruption to existing traffic flows. MAC learned endpoints age as

usual if the Cisco ACI leaf switch sees no traffic with the given source MAC for longer than the endpoint retention policy.

Note Disabling IP dataplane learning means that the endpoint IP information is not updated as a result of traffic forwarding, but Cisco ACI can refresh the endpoint IP information with ARP/ND. This means that the aging of the local endpoints (whether they were learned before the configuration change, or they are learned after the configuration change) differs slightly from the normal aging and it depends also from `System > System Settings > Endpoint Controls > IP Aging`.

If `IP Aging` is disabled, traffic from a source MAC that matches an already learned endpoint MAC, refreshes the MAC addresses information in the endpoint table, and as a result also refreshes the IP information (this is the same as IP dataplane learning enabled).

If `IP Aging` is enabled, Cisco ACI ages out endpoint IP addresses individually (this is no different from what happens with IP dataplane learning enabled), but differently from configurations with IP dataplane learning enabled, traffic from a known source MAC and IP that matches an already learned endpoint, refreshes the MAC address information in the endpoint table, but not the IP information.

Attachable Entity Profile

The ACI fabric provides multiple attachment points that connect through leaf ports to various external entities such as bare metal servers, virtual machine hypervisors, Layer 2 switches (for example, the Cisco UCS fabric interconnect), or Layer 3 routers (for example Cisco Nexus 7000 Series switches). These attachment points can be physical ports, FEX ports, port channels, or a virtual port channel (vPC) on leaf switches.

Note When creating a VPC domain between two leaf switches, both switches must be in the same switch generation, one of the following:

- Generation 1 - Cisco Nexus N9K switches without “EX” or "FX" on the end of the switch name; for example, N9K-9312TX
- Generation 2 – Cisco Nexus N9K switches with “EX” or "FX" on the end of the switch model name; for example, N9K-93108TC-EX

Switches such as these two are not compatible VPC peers. Instead, use switches of the same generation.

An Attachable Entity Profile (AEP) represents a group of external entities with similar infrastructure policy requirements. The infrastructure policies consist of physical interface policies that configure various protocol options, such as Cisco Discovery Protocol (CDP), Link Layer Discovery Protocol (LLDP), or Link Aggregation Control Protocol (LACP).

An AEP is required to deploy VLAN pools on leaf switches. Encapsulation blocks (and associated VLANs) are reusable across leaf switches. An AEP implicitly provides the scope of the VLAN pool to the physical infrastructure.

The following AEP requirements and dependencies must be accounted for in various configuration scenarios, including network connectivity, VMM domains, and multipod configuration:

- The AEP defines the range of allowed VLANS but it does not provision them. No traffic flows unless an EPG is deployed on the port. Without defining a VLAN pool in an AEP, a VLAN is not enabled on the leaf port even if an EPG is provisioned.
- A particular VLAN is provisioned or enabled on the leaf port that is based on EPG events either statically binding on a leaf port or based on VM events from external controllers such as VMware vCenter or Microsoft Azure Service Center Virtual Machine Manager (SCVMM).
- Attached entity profiles can be associated directly with application EPGs, which deploy the associated application EPGs to all those ports associated with the attached entity profile. The AEP has a configurable generic function (infraGeneric), which contains a relation to an EPG (infraRsFuncToEpg) that is deployed on all interfaces that are part of the selectors that are associated with the attachable entity profile.

A virtual machine manager (VMM) domain automatically derives physical interface policies from the interface policy groups of an AEP.

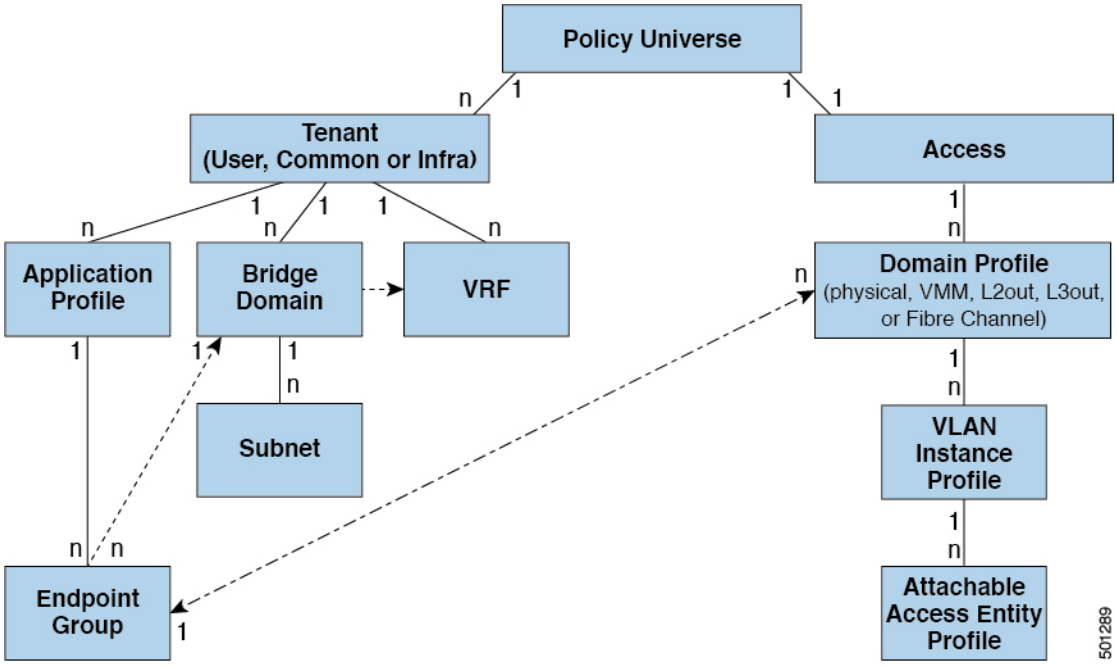
An override policy at the AEP can be used to specify a different physical interface policy for a VMM domain. This policy is useful in scenarios where a VM controller is connected to the leaf switch through an intermediate Layer 2 node, and a different policy is desired at the leaf switch and VM controller physical ports. For example, you can configure LACP between a leaf switch and a Layer 2 node. At the same time, you can disable LACP between the VM controller and the Layer 2 switch by disabling LACP under the AEP override policy.

VLANs and EPGs

Access Policies Automate Assigning VLANs to EPGs

While tenant network policies are configured separately from fabric access policies, tenant policies are not activated unless their underlying access policies are in place. Fabric access external-facing interfaces connect to external devices such as virtual machine controllers and hypervisors, hosts, routers, or Fabric Extenders (FEXs). Access policies enable an administrator to configure port channels and virtual port channels, protocols such as LLDP, CDP, or LACP, and features such as monitoring or diagnostics.

Figure 11: Association of Endpoint Groups with Access Policies



In the policy model, EPGs are tightly coupled with VLANs. For traffic to flow, an EPG must be deployed on a leaf port with a VLAN in a physical, VMM, L2out, L3out, or Fibre Channel domain. For more information, see Networking Domains, on page 137.

In the policy model, the domain profile associated to the EPG contains the VLAN instance profile. The domain profile contains both the VLAN instance profile (VLAN pool) and the attacheable Access Entity Profile (AEP), which are associated directly with application EPGs. The AEP deploys the associated application EPGs to all the ports to which it is attached, and automates the task of assigning VLANs. While a large data center could easily have thousands of active virtual machines provisioned on hundreds of VLANs, the ACI fabric can automatically assign VLAN IDs from VLAN pools. This saves a tremendous amount of time, compared with trunking down VLANs in a traditional data center.

VLAN Guidelines

Use the following guidelines to configure the VLANs where EPG traffic will flow.

- Multiple domains can share a VLAN pool, but a single domain can only use one VLAN pool.
- To deploy multiple EPGs with same VLAN encapsulation on a single leaf switch, see Per Port VLAN, on page 26.

Native 802.1p and Tagged EPGs on Interfaces

When assigning Access (802.1p or Untagged) modes, follow these guidelines to ensure that devices that require untagged or 802.1p packets operate as expected when they are connected to access ports of an ACI leaf switch.

These guidelines apply to EPGs deployed on ports on a single leaf switch. When EPGs are deployed on different switches, these restrictions do not apply.

- In the APIC GUI, when you assign VLANs on ports to EPGs, you can assign one of the following VLAN modes: **Trunk**, **Access (802.1p)**, or **Access (Untagged)**.
- Only one 802.1p VLAN or one untagged VLAN is allowed on a port. It can be one or the other but not both.
- For generation 1 switches, if an EPG deployed on any port on a leaf switch is configured with Access (Untagged) mode, all the ports used by the EPG should be untagged on the same leaf switch and its VPC peer (if there is one). You can have a combination of untagged and tagged ports on generation 2 switches (with -EX, -FX, or -FX2 suffixes).
- You can deploy different EPGs using (tagged) VLAN numbers in **Trunk** mode on the same port, with an EPG deployed on the port in **Access (Untagged)** mode.

There are some differences in traffic handling, depending on the switch, when a leaf switch port is associated with a single EPG that is configured as **Access (802.1p)** or **Access (Untagged)** modes.

Generation 1 Switches

- If the port is configured in **Access (802.1p)** mode:
  - On egress, if the access VLAN is the only VLAN deployed on the port, then traffic will be untagged.
  - On egress, if the port has other (tagged) VLANs deployed along with an untagged EPG, then traffic from that EPG is zero tagged.
  - On egress, for all FEX ports, traffic is untagged, irrespective of one or more VLAN tags configured on the port.
  - The port accepts ingress traffic that is untagged, tagged, or in 802.1p mode.
- If a port is configured in **Access (Untagged)** mode:
  - On egress, the traffic from the EPG is untagged.
  - The port accepts ingress traffic that is untagged, tagged, or 802.1p.

Generation 2 Switches

Generation 2 switches, or later, do not distinguish between the **Access (Untagged)** and **Access (802.1p)** modes. When EPGs are deployed on Generation 2 ports configured with either Untagged or 802.1p mode:

- On egress, traffic is always untagged on a node where this is deployed.
- The port accepts ingress traffic that is untagged, tagged, or in 802.1p mode.

VLAN Mode Combinations on Ports: First Generation and Second Generation Hardware Running Cisco APIC Releases Prior to 3.2(3i)

VLAN Mode Combinations Supported for One EPG

EPG 1 on Port 1, with VLAN mode:	EPG 1 on different ports, the following VLAN modes are allowed:
Trunk	Trunk or 802.1p

EPG 1 on Port 1, with VLAN mode:	EPG 1 on different ports, the following VLAN modes are allowed:
Untagged	Untagged
802.1p	Trunk or 802.1p

VLAN Mode Combinations Supported for Multiple EPGs

EPG 1 on port 1 with VLAN mode:	EPG 1 on port 2, the following modes are allowed:	EPG 2 on port 1, the following modes are allowed:
Untagged	Untagged	Trunk
802.1p	Trunk or 802.1p	Trunk
Trunk	802.1p or Trunk	Trunk or 802.1p or untagged

VLAN Mode Combinations on Ports: Second Generation Hardware Running Cisco APIC Release 3.2(3i) or Later

VLAN Mode Combinations Supported for One EPG

EPG 1 on Port 1, with VLAN mode:	EPG 1 on different ports, the following VLAN modes are allowed:
Trunk	Trunk (tagged) or untagged or 802.1p
Untagged	Untagged or 802.1p or trunk (tagged)
802.1p	Trunk (tagged) or 802.1p or untagged

VLAN Mode Combinations Supported for Multiple EPGs

EPG 1 on port 1 with VLAN mode:	EPG 1 on port 2, the following modes are allowed:	EPG 2 on port 1, the following modes are allowed:
Untagged	Untagged or 802.1p or trunk (tagged)	Trunk (tagged)
802.1p	Trunk (tagged) or 802.1p or untagged	Trunk (tagged)
Trunk	802.1p or trunk (tagged) or untagged	Trunk (tagged) or 802.1p or untagged

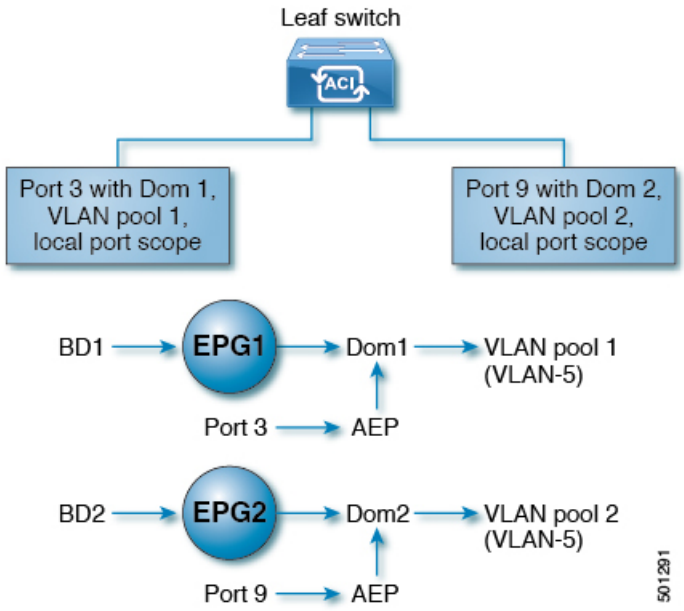
Note Certain older network interface cards (NICs) that send traffic on the native VLAN untagged, drop return traffic that is tagged as VLAN 0. This is normally only a problem on interfaces configured as trunk ports. However, if an Attachable Entity Profile (AEP) for an access port is configured to carry the infra VLAN, then it is treated as a trunk port, even though it is configured as an access port. In these circumstances, packets sent on the native VLAN from the switch with Network Flow Engine (NFE) cards will be tagged as VLAN 0, and older switch NICs may drop them. Options to address this issue include:

- Removing the infra VLAN from the AEP.
- Configuring "port local scope" on the port. This enables per-port VLAN definition and allows the switch equipped with NFE to send packets on the native VLAN, untagged.

Per Port VLAN

In ACI versions prior to the v1.1 release, a given VLAN encapsulation maps to only a single EPG on a leaf switch. If there is a second EPG which has the same VLAN encapsulation on the same leaf switch, the ACI raises a fault.

Starting with the v1.1 release, you can deploy multiple EPGs with the same VLAN encapsulation on a given leaf switch (or FEX), in the Per Port VLAN configuration, similar to the following diagram:



To enable deploying multiple EPGs using the same encapsulation number, on a single leaf switch, use the following guidelines:

- EPGs must be associated with different bridge domains.
- EPGs must be deployed on different ports.
- Both the port and EPG must be associated with the same domain that is associated with a VLAN pool that contains the VLAN number.

- Ports must be configured with `portLocal` VLAN scope.

For example, with Per Port VLAN for the EPGs deployed on ports 3 and 9 in the diagram above, both using VLAN-5, port 3 and EPG1 are associated with Dom1 (pool 1) and port 9 and EPG2 are associated with Dom2 (pool 2).

Traffic coming from port 3 is associated with EPG1, and traffic coming from port 9 is associated with EPG2.

This does not apply to ports configured for Layer 3 external outside connectivity.

When an EPG has more than one physical domain with overlapping VLAN pools, avoid adding more than one domain to the AEP that is used to deploy the EPG on the ports. This avoids the risk of traffic forwarding issues.

When an EPG has only one physical domain with overlapping VLAN pool, you can associate multiple domains with single AEP.

Only ports that have the `vlanScope` set to `portlocal` allow allocation of separate (Port, VLAN) translation entries in both ingress and egress directions. For a given port with the `vlanScope` set to `portGlobal` (the default), each VLAN used by an EPG must be unique on a given leaf switch.

Note Per Port VLAN is not supported on interfaces configured with Multiple Spanning Tree (MST), which requires VLAN IDs to be unique on a single leaf switch, and the VLAN scope to be global.

Reusing VLAN Numbers Previously Used for EPGs on the Same Leaf Switch

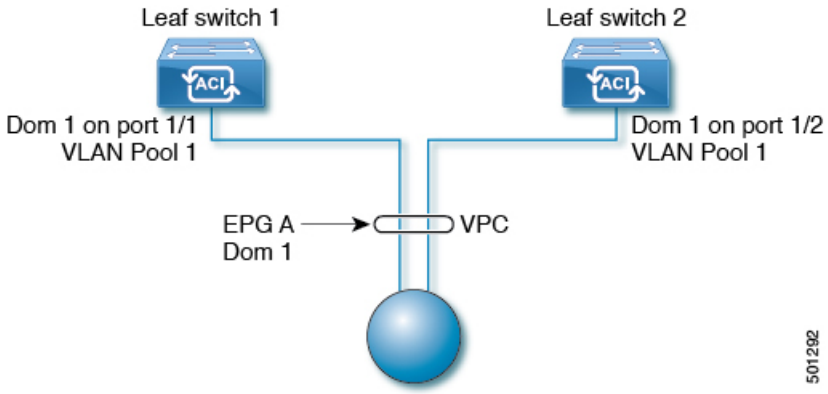
If you have previously configured VLANs for EPGs that are deployed on a leaf switch port, and you want to reuse the same VLAN numbers for different EPGs on different ports on the same leaf switch, use a process, such as the following example, to set them up without disruption:

In this example, EPGs were previously deployed on a port associated with a domain including a VLAN pool with a range of 9-100. You want to configure EPGs using VLAN encapsulations from 9-20.

1. Configure a new VLAN pool on a different port (with a range of, for example, 9-20).
2. Configure a new physical domain that includes leaf ports that are connected to firewalls.
3. Associate the physical domain to the VLAN pool you configured in step 1.
4. Configure the VLAN Scope as `portLocal` for the leaf port.
5. Associate the new EPGs (used by the firewall in this example) to the physical domain you created in step 2.
6. Deploy the EPGs on the leaf ports.

VLAN Guidelines for EPGs Deployed on vPCs

Figure 12: VLANs for Two Legs of a vPC



When an EPG is deployed on a vPC, it must be associated with the same domain (with the same VLAN pool) that is assigned to the leaf switch ports on the two legs of the vPC.

In this diagram, EPG A is deployed on a vPC that is deployed on ports on Leaf switch 1 and Leaf switch 2. The two leaf switch ports and the EPG are all associated with the same domain, containing the same VLAN pool.

Configuring Flood in Encapsulation for All Protocols and Proxy ARP Across Encapsulations

Cisco Application Centric Infrastructure (ACI) uses the bridge domain as the Layer 2 broadcast boundary. Each bridge domain can include multiple endpoint groups (EPGs), and each EPG can be mapped to multiple virtual or physical domains. Each EPG can also use different VLAN encapsulation pools in each domain. Each EPG can also use different VLAN or VXLAN encapsulation pools in each domain.

Ordinarily, when you put multiple EPGs within bridge domains, broadcast flooding sends traffic to all the EPGs in the bridge domain. Because EPGs are used to group endpoints and manage traffic to fulfill specific functions, sending the same traffic to all the EPGs in the bridge domain is not always practical.

The flood in encapsulation feature helps to consolidate bridge domains in your network. The feature does so by enabling you to control broadcast flooding to endpoints within the bridge domain based on the encapsulation of the virtual or physical domain that the EPGs are associated with.

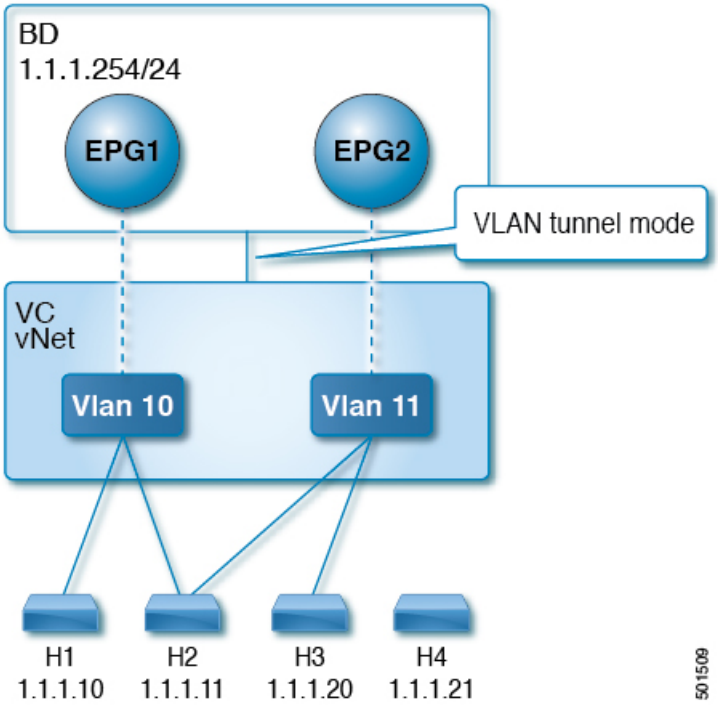
Flood in encapsulation requires the bridge domain to be configured with a subnet and with IP routing because in order to allow communication between endpoints of different EPGs in the same bridge domain Cisco ACI performs proxy ARP.

Example of Flood in Encapsulation Use Case with VLAN Encapsulation

Flood in encapsulation is often used when the external device is using Virtual Connect Tunnel mode where one MAC address is maintained per vNet because of VLAN-agnostic MAC learning.

Using multiple VLANs in tunnel mode can introduce a few challenges. In a typical deployment using Cisco ACI with a single tunnel, as illustrated in the following figure, there are multiple EPGs under one bridge domain. In this case, certain traffic is flooded within the bridge domain (and thus in all the EPGs), with the risk of MAC learning ambiguities that can cause forwarding errors.

Figure 13: Challenges of Cisco ACI with VLAN Tunnel Mode



In this topology, the blade switch (virtual connect in this example) has a single tunnel network defined that uses one uplink to connect with the Cisco ACI leaf node. Two user VLANs, VLAN 10 and VLAN 11 are carried over this link. The bridge domain is set in flooding mode as the servers' gateways are outside the Cisco ACI cloud. ARP negotiations occur in the following process:

- The server sends one ARP broadcast request over the VLAN 10 network.
- The ARP packet travels through the tunnel network to the external server, which records the source MAC address, learned from its downlink.
- The server then forwards the packet out its uplink to the Cisco ACI leaf switch.
- The Cisco ACI fabric sees the ARP broadcast packet entering on access port VLAN 10 and maps it to EPG1.
- Because the bridge domain is set to flood ARP packets, the packet is flooded within the bridge domain and thus to the ports under both EPGs as they are in the same bridge domain.
- The same ARP broadcast packet comes back over the same uplink.
- The blade switch sees the original source MAC address from this uplink.

Result: The blade switch has the same MAC address learned from both the downlink port and uplink port within its single MAC forwarding table, causing traffic disruptions.

Recommended Solution

The flood in encapsulation option is used to limit flooding traffic inside the bridge domain to a single encapsulation. When EPG1/VLAN X and EPG2/VLAN Y share the same bridge domain and flood in encapsulation is enabled, the encapsulation flooding traffic does not reach the other EPG/VLAN.

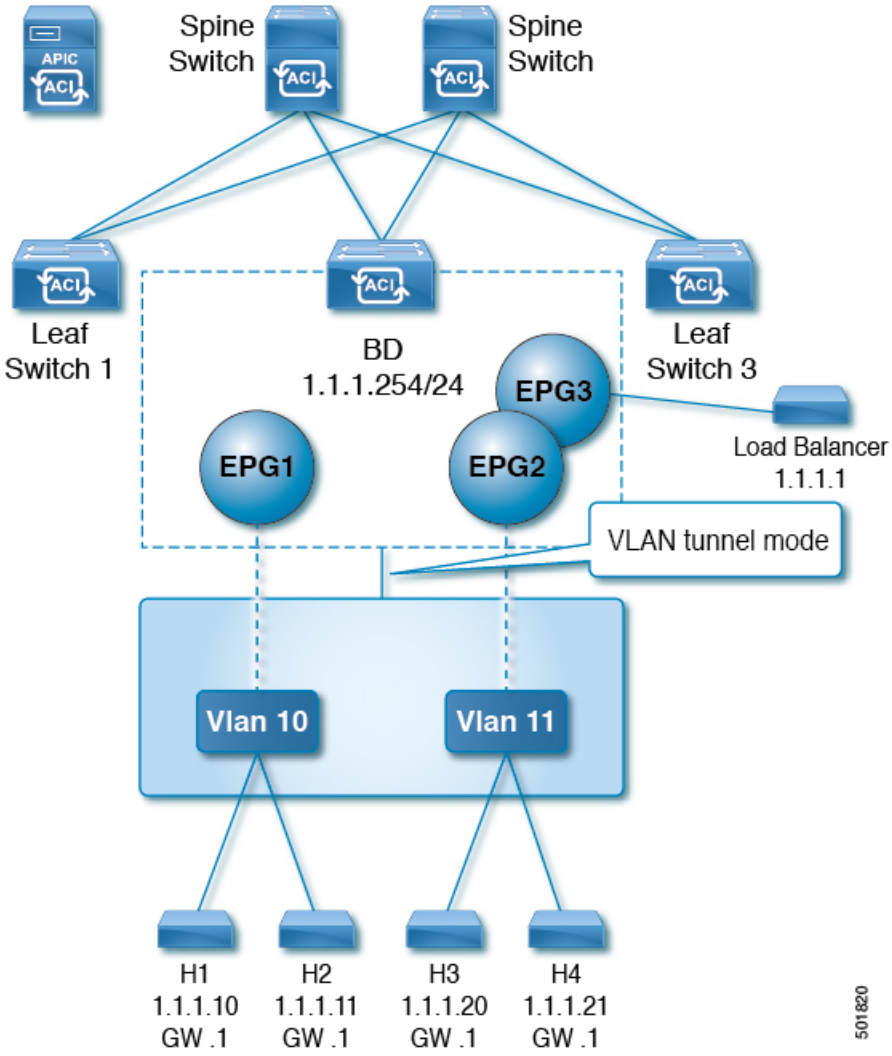
Beginning with Cisco Application Policy Infrastructure Controller (APIC) release 3.1(1), on the Cisco Nexus 9000 series switches (with names ending with EX and FX and onwards), all protocols are flooded in encapsulation. Also, when flood in encapsulation is enabled under the bridge domain for any inter-VLAN traffic, Proxy ARP ensures that the MAC flap issue does not occur. It also limits all flooding (ARP, GARP, and BUM) to the encapsulation. The restriction applies for all EPGs under the bridge domain where it is enabled.

Note Before Cisco APIC release 3.1(1), these features are not supported (proxy ARP and all protocols being included when flooding within encapsulation). In an earlier Cisco APIC release or earlier generation switches (without EX or FX on their names), if you enable flood in encapsulation it does not function, no informational fault is generated, but Cisco APIC decreases the health score by 1.

Note Beginning with Cisco APIC release 3.2(5), you can configure flood in encapsulation for EPGs associated with VXLAN encapsulation. Previously, only VLANs were supported for flood in encapsulation for virtual domains. You configure flood in encapsulation when you create or modify a bridge domain or an EPG.

The recommended solution is to support multiple EPGs under one bridge domain by adding an external switch. This design with multiple EPGs under one bridge domain with an external switch is illustrated in the following figure.

Figure 14: Design with Multiple EPGs Under one Bridge Domain with an External Switch



Within the same bridge domain, some EPGs can be service nodes and other EPGs can have flood in encapsulation configured. A load balancer resides on a different EPG. The load balancer receives packets from the EPGs and sends them to the other EPGs (There is no Proxy ARP and flood within encapsulation does not take place).

Multi-Destination Protocol Traffic

The EPG/bridge domain level broadcast segmentation is supported for the following network control protocols:

- OSPF
- EIGRP
- LACP
- IS-IS
- BGP

- IGMP
- PIM
- STP-BPDU (flooded within EPG)
- ARP/GARP (controlled by ARP Proxy)
- ND

Flood in Encapsulation Limitations

The following limitations apply when using flood in encapsulation for all protocols:

- Flood in encapsulation does not work in ARP unicast mode.
- Neighbor Solicitation (Proxy NS/ND) is not supported for this release.
- Because proxy Address Resolution Protocol (ARP) is enabled implicitly, ARP traffic can go to the CPU for communication between different encapsulations.

  To ensure even distribution to different ports to process ARP traffic, enable per-port Control Plane Policing (CoPP) for ARP with flood in encapsulation.
- Flood in encapsulation is supported only in bridge domain in flood mode and ARP in flood mode. Bridge domain spine proxy mode is not supported.
- IPv4 Layer 3 multicast is not supported.
- IPv6 NS/ND proxy is not supported when flood in encapsulation is enabled. As a result, the connection between two endpoints that are under same IPv6 subnet but resident in EPGs with different encapsulation may not work.
- Virtual machine migration to a different VLAN has momentary issues (60 seconds). Virtual machine migration to a different VLAN or VXLAN has momentary issues (60 seconds).
- Setting up communication between virtual machines through a firewall, as a gateway, is not recommended because if the virtual machine IP address changes to the gateway IP address instead of the firewall IP address, then the firewall can be bypassed.
- Prior releases are not supported (even interoperating between prior and current releases).
- A mixed-mode topology with older-generation Application Leaf Engine (ALE) and Application Spine Engine (ASE) is not recommended and is not supported with flood in encapsulation. Enabling them together can prevent QoS priorities from being enforced.
- Flood in encapsulation is not supported for EPG and bridge domains that are extended across Cisco ACI fabrics that are part of the same Multi-Site domain. However, flood in encapsulation is still working and fully supported, and works for EPGs or bridge domains that are locally defined in Cisco ACI fabrics, independently from the fact those fabrics may be configured for Multi-Site. The same considerations apply for EPGs or bridge domains that are stretched between Cisco ACI fabric and remote leaf switches that are associated to that fabric.
- Flood in encapsulation is not supported on EPGs where microsegmentation is configured.
- Flood in encapsulation is not supported for Common Pervasive Gateway. See the chapter "Common Pervasive Gateway" in the Cisco APIC Layer 3 Networking Configuration Guide.

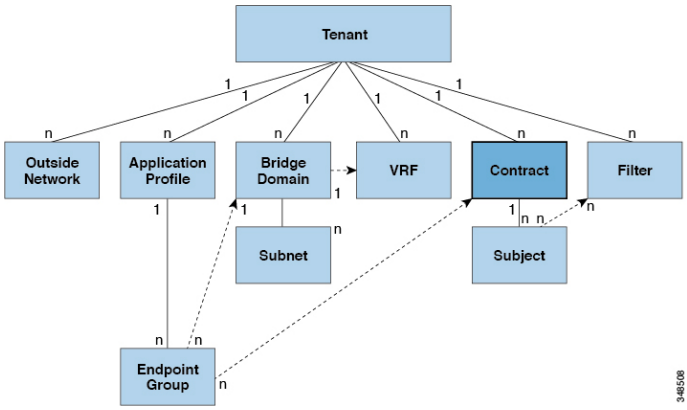
- If you configure the flood in encapsulation on all EPGs of a bridge domain, ensure that you configure the flood in encapsulation on the bridge domain as well.
- IGMP snooping is not supported with flood in encapsulation.
- There is a condition that causes Cisco ACI to flood in the bridge domain (instead of the encapsulation) packets that are received on an EPG that is configured for flood in encapsulation. This happens regardless of whether the administrator configured flood in encapsulation directly on the EPG or on the bridge domain. The condition for this forwarding behavior is if the ingress leaf node has a remote endpoint for the destination MAC address while the egress leaf node does not have a corresponding local endpoint. This can happen due to reasons such as an interface flapping, an endpoint flush due to STP TCN, learning being disabled on the bridge domain due to an excessive amount of moves, and so on.

  In the 4.2(6o) and later 4.2(6) releases, 4.2(7m) and later 4.2(7) releases, and 5.2(1g) and later releases, this behavior was enhanced. If the administrator enables flood in encapsulation on the bridge domain (instead of the EPG), Cisco ACI does not send out such packets on any encapsulations from downlinks facing external devices on the non-ingress (egress and transit) leaf nodes. This new behavior prevents the packets from leaking to unexpected encapsulations. When flood in encapsulation is enabled only at an EPG level, the non-ingress leaf node may still flood packets in the bridge domain instead of the encapsulation. For more information, see the enhancement bug CSCvx83364.
- A Layer 3 gateway must be in the Cisco ACI fabric.

Contracts

In addition to EPGs, contracts (`vzBrCP`) are key objects in the policy model. EPGs can only communicate with other EPGs according to contract rules. The following figure shows the location of contracts in the management information tree (MIT) and their relation to other objects in the tenant.

Figure 15: Contracts



An administrator uses a contract to select the type(s) of traffic that can pass between EPGs, including the protocols and ports allowed. If there is no contract, inter-EPG communication is disabled by default. There is no contract required for intra-EPG communication; intra-EPG communication is always implicitly allowed.

You can also configure contract preferred groups that enable greater control of communication between EPGs in a VRF. If most of the EPGs in the VRF should have open communication, but a few should only have limited communication with the other EPGs, you can configure a combination of a contract preferred group and contracts with filters to control communication precisely.

Contracts govern the following types of endpoint group communications:

- Between ACI fabric application EPGs (`fvAEPg`), both intra-tenant and inter-tenant

Note In the case of a shared service mode, a contract is required for inter-tenant communication. A contract is used to specify static routes across VRFs, even though the tenant VRF does not enforce a policy.

- Between ACI fabric application EPGs and Layer 2 external outside network instance EPGs (`l2extInstP`)
- Between ACI fabric application EPGs and Layer 3 external outside network instance EPGs (`l3extInstP`)
- Between ACI fabric out-of-band (`mgmtOoB`) or in-band (`mgmtInB`) management EPGs

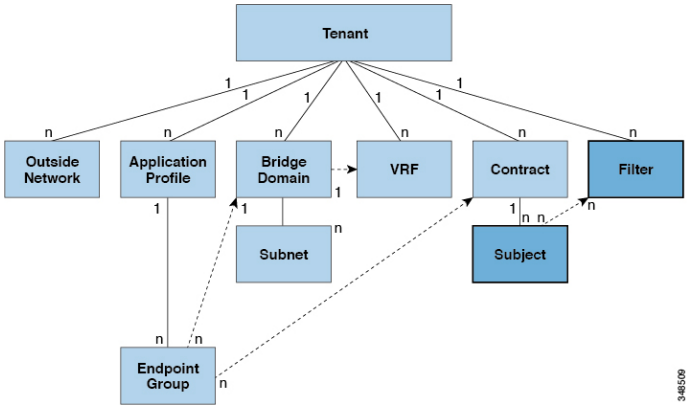
Contracts govern the communication between EPGs that are labeled providers, consumers, or both. EPG providers expose contracts with which a would-be consumer EPG must comply. The relationship between an EPG and a contract can be either a provider or consumer. When an EPG provides a contract, communication with that EPG can be initiated from other EPGs as long as the communication complies with the provided contract. When an EPG consumes a contract, the endpoints in the consuming EPG may initiate communication with any endpoint in an EPG that is providing that contract.

Note An EPG can both provide and consume the same contract. An EPG can also provide and consume multiple contracts simultaneously.

Labels, Filters, Aliases, and Subjects Govern EPG Communications

Label, subject, alias and filter managed-objects enable mixing and matching among EPGs and contracts so as to satisfy various applications or service delivery requirements. The following figure shows the location of application subjects and filters in the management information tree (MIT) and their relation to other objects in the tenant.

Figure 16: Labels, Subjects, and Filters



Contracts can contain multiple communication rules and multiple EPGs can both consume and provide multiple contracts. Labels control which rules apply when communicating between a specific pair of EPGs. A policy

designer can compactly represent complex communication policies and re-use these policies across multiple instances of an application. For example, the sample policy in the *Cisco Application Centric Infrastructure Fundamentals* "Contract Scope Examples" chapter shows how the same contract uses labels, subjects, and filters to differentiate how communications occur among different EPGs that require HTTP or HTTPS.

Labels, subjects, aliases and filters define EPG communications according to the following options:

- Labels are managed objects with only one property: a name. Labels enable classifying which objects can and cannot communicate with one another. Label matching is done first. If the labels do not match, no other contract or filter information is processed. The label match attribute can be one of these values: at least one (the default), all, none, or exactly one. The *Cisco Application Centric Infrastructure Fundamentals* "Label Matching" chapter shows simple examples of all the label match types and their results.

Note Labels can be applied to a variety of provider and consumer managed objects, including EPGs, contracts, bridge domains, DHCP relay policies, and DNS policies. Labels do not apply across object types; a label on an application EPG has no relevance to a label on a bridge domain.

  Labels determine which EPG consumers and EPG providers can communicate with one another. Label matching determines which subjects of a contract are used with a given EPG provider or EPG consumer of that contract.

  The two types of labels are as follows:

  - Subject labels that are applied to EPGs. Subject label matching enables EPGs to choose a subset of the subjects in a contract.
  - Provider/consumer labels that are applied to EPGs. Provider/consumer label matching enables consumer EPGs to choose their provider EPGs and vice versa.

- Aliases are alternative names you can apply to objects, which can be changed, unlike the name.
- Filters are Layer 2 to Layer 4 fields, TCP/IP header fields such as Layer 3 protocol type, Layer 4 ports, and so forth. According to its related contract, an EPG provider dictates the protocols and ports in both the in and out directions. Contract subjects contain associations to the filters (and their directions) that are applied between EPGs that produce and consume the contract.

Note When a contract filter match type is `All`, best practice is to use the VRF unenforced mode. Under certain circumstances, failure to follow these guidelines results in the contract not allowing traffic among EPGs in the VRF.

- Subjects are contained in contracts. One or more subjects within a contract use filters to specify the type of traffic that can be communicated and how it occurs. For example, for HTTPS messages, the subject specifies the direction and the filters that specify the IP address type (for example, IPv4), the HTTP protocol, and the ports allowed. Subjects determine if filters are unidirectional or bidirectional. A unidirectional filter is used in one direction. Unidirectional filters define in or out communications but not the same for both. Bidirectional filters are the same for both; they define both in and out communications.

Configuring Contract or Subject Exceptions for Contracts

In Cisco APIC Release 3.2(1), contracts between EPGs are enhanced to enable denying a subset of contract providers or consumers from participating in the contract. Inter-EPG contracts and Intra-EPG contracts are supported with this feature.

You can enable a provider EPG to communicate with all consumer EPGs except those that match criteria configured in a subject or contract exception. For example, if you want to enable an EPG to provide services to all EPGs for a tenant, except a subset, you can enable those EPGs to be excluded. To configure this, you create an exception in the contract or one of the subjects in the contract. The subset is then denied access to providing or consuming the contract.

Labels, counters, and permit and deny logs are supported with contracts and subject exceptions.

To apply an exception to all subjects in a contract, add the exception to the contract. To apply an exception only to a single subject in the contract, add the exception to the subject.

When adding filters to subjects, you can set the action of the filter (to permit or deny objects that match the filter criteria). Also for **Deny** filters, you can set the priority of the filter. **Permit** filters always have the default priority. Marking the subject-to-filter relation to deny automatically applies to each pair of EPGs where there is a match for the subject. Contracts and subjects can include multiple subject-to-filter relationships that can be independently set to permit or deny the objects that match the filters.

Exception Types

Contract and subject exceptions can be based on the following types and include regular expressions, such as the * wildcard:

Exception criteria exclude these objects as defined in the Consumer Regex and Provider Regex fields	Example	Description
Tenant	`<vzException consRegex= “common” field= “Tenant” name= “excep03” provRegex= “t1” />`	This example, excludes EPGs using the `common` tenant from consuming contracts provided by the `t1` tenant.
VRF	`<vzException consRegex= “ctx1” field= “Ctx” name= “excep05” provRegex= “ctx1” />`	This example excludes members of `ctx1` from consuming the services provided by the same VRF.
EPG	`<vzException consRegex= “EPgPa.*” field= “EPg” name= “excep03” provRegex= “EPg03” />`	The example assumes that multiple EPGs exist, with names starting with `EPGPa`, and they should all be denied as consumers for the contract provided by `EPg03`
Dn	`<vzException consRegex= “uni/tn-t36/ap-customer/epg-epg193” field= “Dn” name=“excep04” provRegex= “uni/tn-t36/ap-customer/epg-epg200” />`	This example excludes `epg193` from consuming the contract provided by `epg200`.

Exception criteria exclude these objects as defined in the Consumer Regex and Provider Regex fields	Example	Description
Tag	`<vzException consRegex= "red" field= "Tag" name= "excep01" provRegex= "green" />`	The example excludes objects marked with the `red` tag from consuming and those marked with the `green` tag from participating in the contract.

Taboos

While the normal processes for ensuring security still apply, the ACI policy model aids in assuring the integrity of whatever security practices are employed. In the ACI policy model approach, all communications must conform to these conditions:

- Communication is allowed only based on contracts, which are managed objects in the model. If there is no contract, inter-EPG communication is disabled by default.
- No direct access to the hardware; all interaction is managed through the policy model.

Taboo contracts can be used to deny specific traffic that is otherwise allowed by contracts. The traffic to be dropped matches a pattern (such as, any EPG, a specific EPG, or traffic matching a filter). Taboo rules are unidirectional, denying any matching traffic coming toward an EPG that provides the contract.

With Cisco APIC Release 3.2(x) and switches with names that end in EX or FX, you can alternatively use a subject Deny action or Contract or Subject Exception in a standard contract to block traffic with specified patterns.

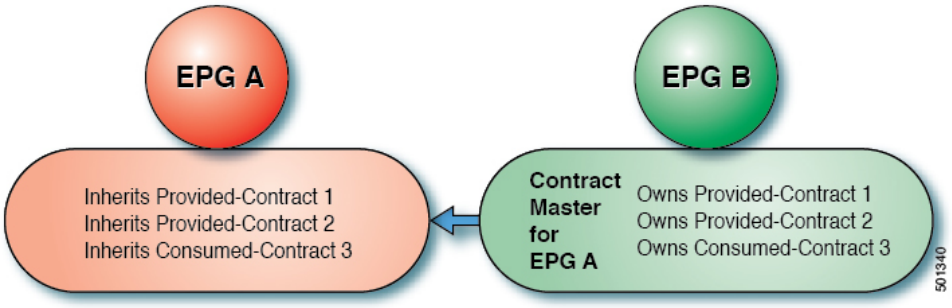
About Contract Inheritance

To streamline associating contracts to new EPGs, you can now enable an EPG to inherit all the (provided and consumed) contracts associated directly to another EPG in the same tenant. Contract inheritance can be configured for application, microsegmented, L2Out, and L3Out EPGs.

With Release 3.x, you can also configure contract inheritance for Inter-EPG contracts, both provided and consumed. Inter-EPG contracts are supported on Cisco Nexus 9000 Series switches with EX or FX at the end of their model name or later models.

You can enable an EPG to inherit all the contracts associated directly to another EPG, using the APIC GUI, NX-OS style CLI, and the REST API.

Figure 17: Contract Inheritance



In the diagram above, EPG A is configured to inherit Provided-Contract 1 and 2 and Consumed-Contract 3 from EPG B (contract master for EPG A).

Use the following guidelines when configuring contract inheritance:

- Contract inheritance can be configured for application, microsegmented (uSeg), external L2Out EPGs, and external L3Out EPGs. The relationships must be between EPGs of the same type.
- Both provided and consumed contracts are inherited from the contract master when the relationship is established.
- Contract masters and the EPGs inheriting contracts must be within the same tenant.
- Changes to the masters' contracts are propagated to all the inheritors. If a new contract is added to the master, it is also added to the inheritors.
- An EPG can inherit contracts from multiple contract masters.
- Contract inheritance is only supported to a single level (cannot be chained) and a contract master cannot inherit contracts.
- Labels with contract inheritance is supported. When EPG A inherits a contract from EPG B, if different subject labels are configured under EPG A and EPG B, APIC uses the label configured under EPG B for the contract inherited from EPG B. APIC uses the label configured under EPG A for the contract where EPG A is directly involved.
- Whether an EPG is directly associated to a contract or inherits a contract, it consumes entries in TCAM. So contract scale guidelines still apply. For more information, see the *Verified Scalability Guide* for your release.
- vzAny security contracts and taboo contracts are not supported.
- Beginning in Cisco APIC releases 5.0(1) and 4.2(6), contract inheritance with a service graph is supported if the contract and EPGs are in the same tenant.

For information about configuring Contract Inheritance and viewing inherited and standalone contracts, see *Cisco APIC Basic Configuration Guide*.

About Contract Preferred Groups

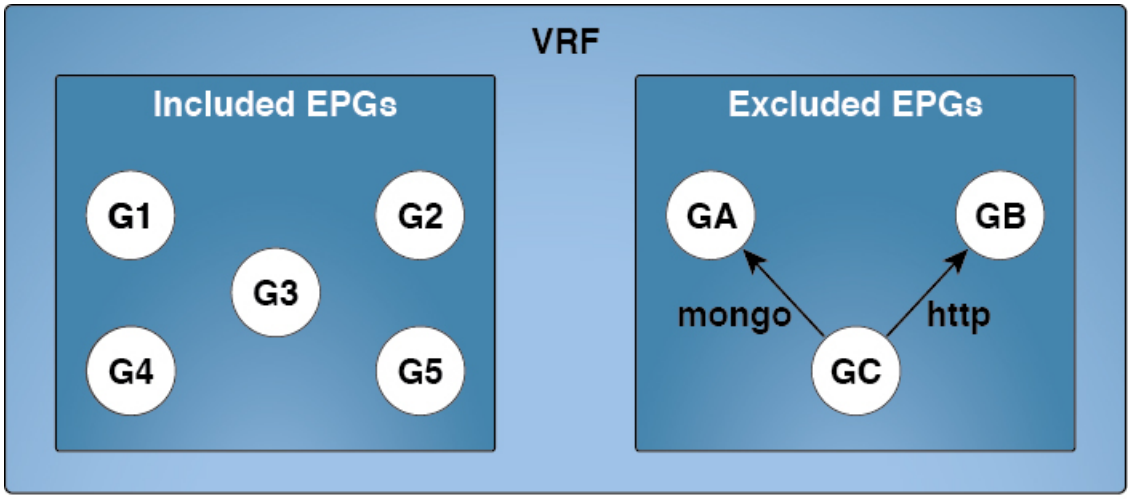
There are two types of policy enforcements available for EPGs in a VRF with a contract preferred group configured:

- Included EPGs: EPGs can freely communicate with each other without contracts, if they have membership in a contract preferred group. This is based on the source-any-destination-any-permit default rule.
- Excluded EPGs: EPGs that are not members of preferred groups require contracts to communicate with each other. Otherwise, the default source-any-destination-any-deny rule applies.

The contract preferred group feature enables greater control of communication between EPGs in a VRF. If most of the EPGs in the VRF should have open communication, but a few should only have limited communication with the other EPGs, you can configure a combination of a contract preferred group and contracts with filters to control inter-EPG communication precisely.

EPGs that are excluded from the preferred group can only communicate with other EPGs if there is a contract in place to override the source-any-destination-any-deny default rule.

Figure 18: Contract Preferred Group Overview



Source	Destination	Filter	Action
any	any	implicit	permit
GA	any	implicit	deny
any	GA	implicit	deny
GB	any	implicit	deny
any	GB	implicit	deny
GC	any	implicit	deny
any	GC	implicit	deny

Source	Destination	Filter	Action
GC	GA	mongo	permit
GA	GC	mongo	permit
GC	GB	http	permit
GB	GC	http	permit

Service Graph Support

As of APIC release 4.0(1), EPGs created by service graphs can be included in contract preferred groups. A new policy (Service EPG Policy) is available for defining the preferred group membership type (include or exclude). Once configured, it can be applied through the device selection policy or through the application of a service graph template.

Also, shadow EPGs can now be configured to be included or excluded in preferred groups.

Limitations

The following limitations apply to contract preferred groups:

- In topologies where an L3Out and application EPG are configured in a Contract Preferred Group, and the EPG is deployed only on a VPC, you may find that only one leaf switch in the VPC has the prefix entry for the L3Out. In this situation, the other leaf switch in the VPC does not have the entry, and therefore drops the traffic.

  To workaround this issue, you can do one of the following:

  - Disable and reenable the contract group in the VRF
  - Delete and recreate the prefix entries for the L3Out EPG

- Also, where the provider or consumer EPG in a service graph contract is included in a contract group, the shadow EPG can not be excluded from the contract group. The shadow EPG will be permitted in the contract group, but it does not trigger contract group policy deployment on the node where the shadow EPG is deployed. To download the contract group policy to the node, you deploy a dummy EPG within the contract group .
- Due to CSCvm63145, an EPG in a Contract Preferred Group can consume a shared service contract, but cannot be a provider for a shared service contract with an L3Out EPG as consumer.

Optimize Contract Performance

Starting with Cisco APIC, Release 3.2, you can configure bidirectional contracts that support more efficient hardware TCAM storage of contract data. With optimization enabled, contract statistics for both directions are aggregated.

TCAM Optimization is supported on the second generation Cisco Nexus 9000 Series top of rack (TOR) switches, which are those with suffixes of EX, FX, and FX2, and later (for example, N9K-C93180LC-EX or N9K-C93180YC-FX).

To configure efficient TCAM contract data storage, you enable the following options:

- Mark the contracts to be applied in both directions between the provider and consumer.
- For filters with IP TCP or UDP protocols, enable the reverse port option.
- When configuring the contract subjects, select the **Enable Policy Compression** directive, which adds the `no_stats` option to the `action` attribute of the `actrl:Rule` managed object.

Limitations

With the **Enable Policy Compression** (`no_stats`) option selected, per-rule statistics are lost. However, combined rule statistics for both directions are present in the hardware statistics.

After upgrading to Cisco APIC 3.2(1), to add the `no_stats` option to a pre-upgrade contract subject (with filters or filter entries), you must delete the contract subject and reconfigure it with the **Enable Policy Compression** directive. Otherwise, compression does not occur.

For each contract with a bi-directional subject filter, Cisco NX-OS creates 2 rules:

- A rule with an `sPcTag` and `dPcTag` that is marked `direction=bi-dir`, which is programmed in hardware
- A rule marked with `direction=uni-dir-ignore` which is not programmed

Rules with the following settings are not compressed:

- Rules with priority other than `fully_qual`
- Opposite rules (`bi-dir` and `uni-dir-ignore` marked) with non-identical properties, such as **action** including **directives**, **prio, qos** or **markDscp**
- Rule with `Implicit` or `implarp` filters
- Rules with the actions `Deny`, `Redir`, `Copy`, or `Deny-log`

The following MO query output shows the two rules for a contract, that is considered for compression:

```
apic1# moquery -c actrlRule
Total Objects shown: 2

# actrl.Rule
scopeId            : 2588677
sPcTag             : 16388
dPcTag             : 49156
fltId              : 67
action             : no_stats,permit
actrlCfgFailedBmp  :
actrlCfgFailedTs   : 00:00:00:00.000
actrlCfgState      : 0
childAction        :
ctrctName          :
descr              :
direction          : bi-dir
dn                 : sys/actrl/scope-2588677/rule-2588677-s-16388-d-49156-f-67
id                 : 4112
lcOwn              : implicit
markDscp           : unspecified
modTs              : 2019-04-27T09:01:33.152-07:00
monPolDn           : uni/tn-common/monepg-default
name               :
nameAlias          :
operSt             : enabled
operStQual         :
prio               : fully_qual
qosGrp             : unspecified
rn                 : rule-2588677-s-16388-d-49156-f-67
status             :
type               : tenant


# actrl.Rule
scopeId            : 2588677
sPcTag             : 49156
dPcTag             : 16388
fltId              : 64
action             : no_stats,permit
actrlCfgFailedBmp  :
actrlCfgFailedTs   : 00:00:00:00.000
actrlCfgState      : 0
childAction        :
ctrctName          :
descr              :
direction          : uni-dir-ignore
dn                 : sys/actrl/scope-2588677/rule-2588677-s-49156-d-16388-f-64
id                 : 4126
lcOwn              : implicit
```

```
markDscp            : unspecified
modTs               : 2019-04-27T09:01:33.152-07:00
monPolDn            : uni/tn-common/monepg-default
name                :
nameAlias           :
operSt              : enabled
operStQual          :
prio                : fully_qual
qosGrp              : unspecified
rn                  : rule-2588677-s-49156-d-16388-f-64
status              :
type                : tenant
```

Table 2: Compression Matrix

Reverse Filter Port Enabled	TCP or UDP Source Port	TCP or UCP Destination Port	Compressed
Yes	Port A	Port B	Yes
Yes	Unspecified	Port B	Yes
Yes	Port A	Unspecified	Yes
Yes	Unspecified	Unspecified	Yes
No	Port A	Port B	No
No	Unspecified	Port B	No
No	Port A	Unspecified	No
No	Unspecified	Unspecified	Yes

What vzAny Is

The `vzAny` managed object provides a convenient way of associating all endpoint groups (EPGs) in a Virtual Routing and Forwarding (VRF) instance to one or more contracts (`vzBrCP`), instead of creating a separate contract relation for each EPG.

In the Cisco ACI fabric, EPGs can only communicate with other EPGs according to contract rules. A relationship between an EPG and a contract specifies whether the EPG provides the communications defined by the contract rules, consumes them, or both. By dynamically applying contract rules to all EPGs in a VRF, `vzAny` automates the process of configuring EPG contract relationships. Whenever a new EPG is added to a VRF, vzAny contract rules automatically apply. The `vzAny` one-to-all EPG relationship is the most efficient way of applying contract rules to all EPGs in a VRF.

Note In the APIC GUI under tenants, a VRF is also known as a private network (a network within a tenant) or a context.

In the case of shared services, you *must* define the provider EPG shared subnet under the EPG in order to properly derive the pcTag (classification) of the destination from the consumer (vzAny) side. If you are migrating from a BD-to-BD shared services configuration, where both the consumer and provider subnets

are defined under bridge domains, to vzAny acting as a shared service consumer, you must take an extra configuration step where you add the provider subnet to the EPG with the shared flags at minimum.

Note If you add the EPG subnet as a duplicate of the defined BD subnet, ensure that both definitions of the subnet always have the same flags defined. Failure to do so can result in unexpected fabric forwarding behavior.

To use vzAny, navigate to **Tenants** > ***tenant-name*** > **Networking** > **VRFs** > ***vrf-name*** > **EPG Collection for VRF**.

About Copy Services

Unlike SPAN that duplicates all of the traffic, the Cisco Application Centric Infrastructure (ACI) copy services feature enables selectively copying portions of the traffic between endpoint groups, according to the specifications of the contract. Broadcast, unknown unicast and multicast (BUM), and control plane traffic that are not covered by the contract are not copied. In contrast, SPAN copies everything out of endpoint groups, access ports or uplink ports. Unlike SPAN, copy services do not add headers to the copied traffic. Copy service traffic is managed internally in the switch to minimize impact on normal traffic forwarding.

A copy service is configured as part of a Layer 4 to Layer 7 service graph template that specifies a copy cluster as the destination for the copied traffic. A copy service can tap into different hops within a service graph. For example, a copy service could select traffic between a consumer endpoint group and a firewall provider endpoint group, or between a server load balancer and a firewall. Copy clusters can be shared across tenants.

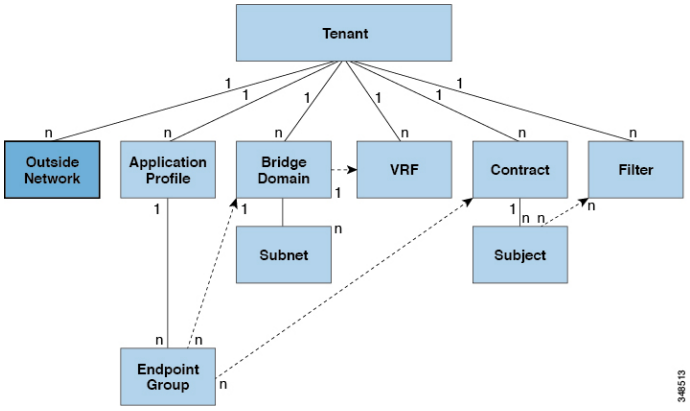
Copy services require you to do the following tasks:

- Identify the source and destination endpoint groups.
- Configure the contract that specifies what to copy according to the subject and what is allowed in the contract filter.
- Configure Layer 4 to Layer 7 copy devices that identify the target devices and specify the ports where they attach.
- Use the copy service as part of a Layer 4 to Layer 7 service graph template.
- Configure a device selection policy that specifies which device will receive the traffic from the service graph. When you configure the device selection policy, you specify the contract, service graph, copy cluster, and cluster logical interface that is in copy device.

Outside Networks

Outside network policies control connectivity to the outside. A tenant can contain multiple outside network objects. The following figure shows the location of outside networks in the management information tree (MIT) and their relation to other objects in the tenant.

Figure 19: Outside Networks



Outside network policies specify the relevant Layer 2 (`l2extOut`) or Layer 3 (`l3extOut`) properties that control communications between an outside public or private network and the ACI fabric. External devices, such as routers that connect to the WAN and enterprise core, or existing Layer 2 switches, connect to the front panel interface of a leaf switch. The leaf switch that provides such connectivity is known as a border leaf. The border leaf switch interface that connects to an external device can be configured as either a bridged or routed interface. In the case of a routed interface, static or dynamic routing can be used. The border leaf switch can also perform all the functions of a normal leaf switch.

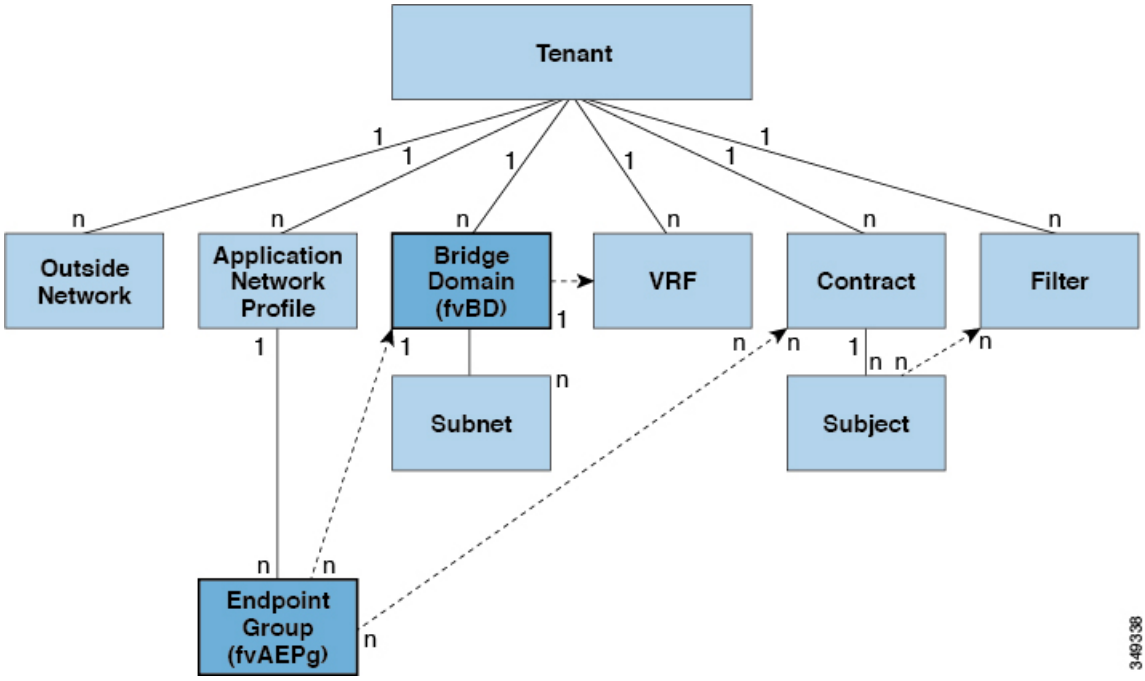
Managed Object Relations and Policy Resolution

Relationship managed objects express the relation between managed object instances that do not share containment (parent-child) relations. MO relations are established between the source MO and a target MO in one of the following two ways:

- An explicit relation (`fvRsPathAtt`) defines a relationship based on the target MO distinguished name (DN).
- A named relation defines a relationship based on the target MO name.

The dotted lines in the following figure shows several common MO relations.

Figure 20: MO Relations



For example, the dotted line between the EPG and the bridge domain defines the relation between those two MOs. In this figure, the EPG (`fvAEPg`) contains a relationship MO (`fvRsBD`) that is named with the name of the target bridge domain MO (`fvBD`). For example, if production is the bridge domain name (`tnFvBDName=production`), then the relation name would be production (`fvRsBdName=production`).

In the case of policy resolution based on named relations, if a target MO with a matching name is not found in the current tenant, the ACI fabric tries to resolve in the common tenant. For example, if the user tenant EPG contained a relationship MO targeted to a bridge domain that did not exist in the tenant, the system tries to resolve the relationship in the common tenant. If a named relation cannot be resolved in either the current tenant or the common tenant, the ACI fabric attempts to resolve to a default policy. If a default policy exists in the current tenant, it is used. If it does not exist, the ACI fabric looks for a default policy in the common tenant. Bridge domain, VRF, and contract (security policy) named relations do not resolve to a default.

Default Policies

The initial values of the APIC default policies values are taken from the concrete model that is loaded in the switch. A fabric administrator can modify default policies.

Warning Default policies can be modified or deleted. Deleting a default policy can result in a policy resolution process to complete abnormally.

The ACI fabric includes default policies for many of its core functions. Examples of default policies include the following:

- Bridge domain (in the common tenant)

- Layer 2 and Layer 3 protocols
- Fabric initialization, device discovery, and cabling detection
- Storm control and flooding
- Virtual port channel
- Endpoint retention for caching and aging of learned endpoints in switch buffers
- Loop detection
- Monitoring and statistics

Note To avoid confusion when implementing configurations that use default policies, document changes made to default policies. Be sure there are no current or future configurations that rely on a default policy before deleting a default policy. For example, deleting a default firmware update policy could result in a problematic future firmware update.

When the ACI fabric is upgraded, the existing policy default values persist, even if the default value changes in the newer release. When the node connects to the APIC for the first time, the node registers itself with APIC which pushes all the default policies to the node. Any change in the default policy is pushed to the node.

A default policy serves multiple purposes:

- Allows a fabric administrator to override the default values in the model.
- If an administrator does not provide an explicit policy, the APIC applies the default policy. An administrator can create a default policy and the APIC uses that unless the administrator provides any explicit policy.

For example, according to actions the administrator does or does not take, the APIC will do the following:

- Because the administrator does not specify the LLDP policy for the selected ports, the APIC applies the default LLDP interface policy for the ports specified in the port selector.
- If the administrator removes a port from a port selector, the APIC applies the default policies to that port. In this example, if the administrator removes port 1/15 from the port selector, the port is no longer part of the port channel and the APIC applies all the default policies to that port.

The following scenarios describe common policy resolution behavior:

- A configuration explicitly refers to the default policy: if a default policy exists in the current tenant, it is used. Otherwise, the default policy in tenant **common** is used.
- A configuration refers to a named policy (not default) that does not exist in the current tenant or in tenant common: if the current tenant has a default policy, it is used. Otherwise, the default policy in tenant **common** is used.

Note This does not apply to a bridge domain or a VRF (private network) in a tenant.

- A configuration does not refer to any policy name: if a default policy exists in the current tenant, it is used. Otherwise, the default policy in tenant **common** is used.

Note For bridge domains and VRFs, this only applies if the connectivity instrumentation policy (`fvConnInstrPol`) in the **common** tenant has the appropriate bridge domain or VRF flag set. This prevents unintended EPGs from being deployed in tenant **common** subnets.

The policy model specifies that an object is using another policy by having a relation managed object (MO) under that object and that relation MO refers to the target policy by name. If this relation does not explicitly refer to a policy by name, then the system will try to resolve a policy called default. Bridge domains (BD) and VRFs (Ctx) are exceptions to this rule.

An endpoint group (EPG) has a relation to a BD (`fvRsBd`) that has a property called `tnFvBDName`. If this is not set (`tnVfBDName=""`), the connectivity instrumentation policy (`fvConnInstrPol`) derives the behavior for this case. This policy applies for all EPG cases (VMM, baremetal, l2ext, l3ext). The instrumentation policy uses the `bdctrl` property to control whether the default BD policy will be used and the `ctxCtrl` property to control whether the default VRF (Ctx) policy will be used. The following options are the same for both:

- *do not instrument*: the leaf switch will not use the default policy.
- *instrument-and-no-route*: instrument the policy and do not enable routing.
- *instrument-and-route*: instrument the policy and enable routing.

Trans Tenant EPG Communications

EPGs in one tenant can communicate with EPGs in another tenant through a contract interface contained in a shared tenant. The contract interface is an MO that can be used as a contract consumption interface by the EPGs that are contained in different tenants. By associating to an interface, an EPG consumes the subjects represented by the interface to a contract contained in the shared tenant. Tenants can participate in a single contract, which is defined at some third place. More strict security requirements can be satisfied by defining the tenants, contract, subjects, and filter directions so that tenants remain completely isolated from one another.

Follow these guidelines when configuring shared services contracts:

- When a contract is configured between in-band and out-of-band EPGs, the following restrictions apply:
  - Both EPGs should be in the same VRF (context).
  - Filters apply in the incoming direction only.
  - Layer 2 filters are not supported.
  - QoS does not apply to in-band Layer 4 to Layer 7 services.
    - Management statistics are not available.
    - Shared services for CPU-bound traffic are not supported.
- Contracts are needed for inter-bridge domain traffic when a private network is unenforced.

- Prefix-based EPGs are not supported. Shared Services are not supported for a Layer 3 external outside network. Contracts provided or consumed by a Layer 3 external outside network need to be consumed or provided by EPGs that share the same Layer 3 VRF.
- A shared service is supported only with non-overlapping and non-duplicate subnets. When configuring subnets for shared services, follow these guidelines:
  - Configure the subnet for a shared service provider under the EPG, not under the bridge domain.
  - Subnets configured under an EPG that share the same VRF must be disjointed and must not overlap.
  - Subnets leaked from one VRF to another must be disjointed and must not overlap.
  - Subnets advertised from multiple consumer networks into a VRF or vice versa must be disjointed and must not overlap.

Note If two consumers are mistakenly configured with the same subnet, recover from this condition by removing the subnet configuration for both, then reconfigure the subnets correctly.

- Do not configure a shared service with `AnyToProv` in the provider VRF. The APIC rejects this configuration and raises a fault.
- The private network of a provider cannot be in unenforced mode while providing a shared service.

Tags

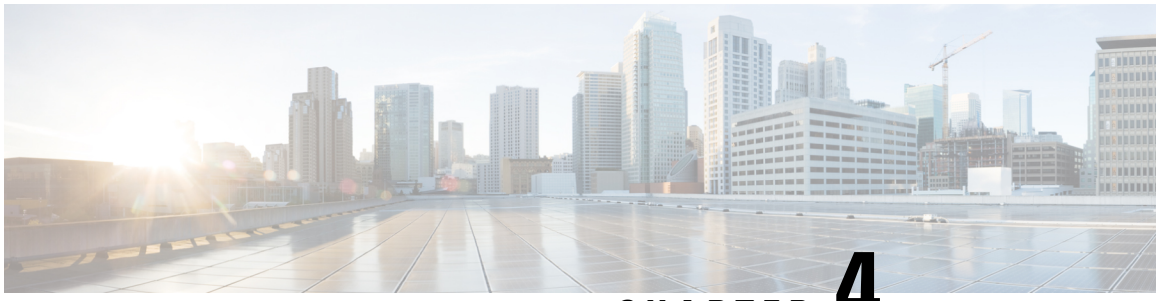
Object tags simplify API operations. In an API operation, an object or group of objects can be referenced by the tag name instead of by the distinguished name (DN). Tags are child objects of the item they tag; besides the name, they have no other properties.

Use a tag to assign a descriptive name to a group of objects. The same tag name can be assigned to multiple objects. Multiple tag names can be assigned to an object. For example, to enable easy searchable access to all web server EPGs, assign a web server tag to all such EPGs. Web server EPGs throughout the fabric can be located by referencing the web server tag.

About APIC Quota Management Configuration

Starting in the Cisco Application Policy Infrastructure Controller (APIC) Release 2.3(1), there are limits on number of objects a tenant admin can configure. This enables the admin to limit what managed objects that can be added under a given tenant or globally across tenants.

This feature is useful when you want to limit any tenant or group of tenants from exceeding ACI maximums per leaf or per fabric or unfairly consuming a majority of available resources, potentially affecting other tenants on the same fabric.



CHAPTER 4

Fabric Provisioning

This chapter contains the following sections:

• Geolocation, on page 96

Fabric Provisioning

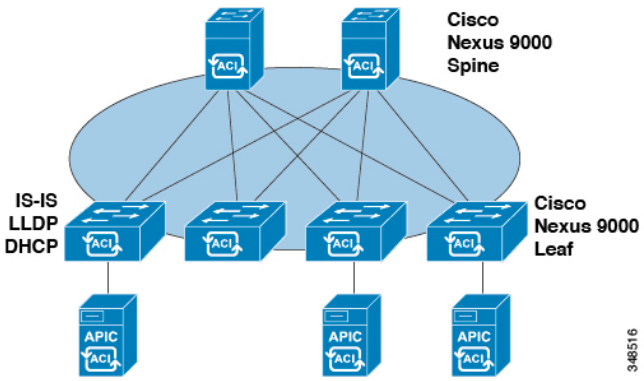
Cisco Application Centric Infrastructure (ACI) automation and self-provisioning offers these operation advantages over the traditional switching infrastructure:

- A clustered logically centralized but physically distributed APIC provides policy, bootstrap, and image management for the entire fabric.
- The APIC startup topology auto discovery, automated configuration, and infrastructure addressing uses these industry-standard protocols: Intermediate System-to-Intermediate System (IS-IS), Link Layer Discovery Protocol (LLDP), and Dynamic Host Configuration Protocol (DHCP).
- The APIC provides a simple and automated policy-based provisioning and upgrade process, and automated image management.
- APIC provides scalable configuration management. Because ACI data centers can be very large, configuring switches or interfaces individually does not scale well, even using scripts. APIC pod, controller, switch, module and interface selectors (all, range, specific instances) enable symmetric configurations across the fabric. To apply a symmetric configuration, an administrator defines switch profiles that associate interface configurations in a single policy group. The configuration is then rapidly deployed to all interfaces in that profile without the need to configure them individually.

Startup Discovery and Configuration

The clustered APIC controller provides DHCP, bootstrap configuration, and image management to the fabric for automated startup and upgrades. The following figure shows startup discovery.

Figure 21: Startup Discovery Configuration



The Cisco Nexus ACI fabric software is bundled as an ISO image, which can be installed on the Cisco APIC server through the KVM interface on the Cisco Integrated Management Controller (CIMC). The Cisco Nexus ACI Software ISO contains the Cisco APIC image, the firmware image for the leaf node, the firmware image for the spine node, default fabric infrastructure policies, and the protocols required for operation.

The ACI fabric bootstrap sequence begins when the fabric is booted with factory-installed images on all the switches. The Cisco Nexus 9000 Series switches that run the ACI firmware and APICs use a reserved overlay

for the boot process. This infrastructure space is hard-coded on the switches. The APIC can connect to a leaf through the default overlay, or it can use a locally significant identifier.

The ACI fabric uses an infrastructure space, which is securely isolated in the fabric and is where all the topology discovery, fabric management, and infrastructure addressing is performed. ACI fabric management communication within the fabric takes place in the infrastructure space through internal private IP addresses. This addressing scheme allows the APIC to communicate with fabric nodes and other Cisco APIC controllers in the cluster. The APIC discovers the IP address and node information of other Cisco APIC controllers in the cluster using the Link Layer Discovery Protocol (LLDP)-based discovery process.

The following describes the APIC cluster discovery process:

- Each APIC in the Cisco ACI uses an internal private IP address to communicate with the ACI nodes and other APICs in the cluster. The APIC discovers the IP address of other APIC controllers in the cluster through the LLDP-based discovery process.
- APICs maintain an appliance vector (AV), which provides a mapping from an APIC ID to an APIC IP address and a universally unique identifier (UUID) of the APIC. Initially, each APIC starts with an AV filled with its local IP address, and all other APIC slots are marked as unknown.
- When a switch reboots, the policy element (PE) on the leaf gets its AV from the APIC. The switch then advertises this AV to all of its neighbors and reports any discrepancies between its local AV and neighbors' AVs to all the APICs in its local AV.

Using this process, the APIC learns about the other APIC controllers in the ACI through switches. After validating these newly discovered APIC controllers in the cluster, the APIC controllers update their local AV and program the switches with the new AV. Switches then start advertising this new AV. This process continues until all the switches have the identical AV and all APIC controllers know the IP address of all the other APIC controllers.

Note Prior to initiating a change to the cluster, always verify its health. When performing planned changes to the cluster, all controllers in the cluster should be healthy. If one or more of the APIC controllers in the cluster is not healthy, remedy that situation before proceeding with making changes to the cluster. Also, assure that cluster controllers added to the APIC are running the same version of firmware as the other controllers in the APIC cluster. See the KB: Cisco ACI APIC Cluster Management article for guidelines that must be followed to assure that making changes the APIC cluster complete normally.

The ACI fabric is brought up in a cascading manner, starting with the leaf nodes that are directly attached to the APIC. LLDP and control-plane IS-IS convergence occurs in parallel to this boot process. The ACI fabric uses LLDP- and DHCP-based fabric discovery to automatically discover the fabric switch nodes, assign the infrastructure VXLAN tunnel endpoint (VTEP) addresses, and install the firmware on the switches. Prior to this automated process, a minimal bootstrap configuration must be performed on the Cisco APIC controller. After the APIC controllers are connected and their IP addresses assigned, the APIC GUI can be accessed by entering the address of any APIC controller into a web browser. The APIC GUI runs HTML5 and eliminates the need for Java to be installed locally.

Fabric Inventory

The policy model contains a complete real-time inventory of the fabric, including all nodes and interfaces. This inventory capability enables automation of provisioning, troubleshooting, auditing, and monitoring.

For Cisco ACI fabric switches, the fabric membership node inventory contains policies that identify the node ID, serial number, and name. Third-party nodes are recorded as unmanaged fabric nodes. Cisco ACI switches can be automatically discovered, or their policy information can be imported. The policy model also maintains fabric member node state information.

Node States	Condition
Unknown	No policy. All nodes require a policy; without a policy, a member node state is unknown.
Discovering	A transient state showing that the node is being discovered and waiting for host traffic.
Undiscovered	The node has policy but has never been brought up in the fabric.
Unsupported	The node is a Cisco switch but it is not supported. For example, the firmware version is not compatible with ACI fabric.
Decommissioned	The node has a policy, was discovered, but a user disabled it. The node can be reenabled. **Note** Specifying the wipe option when decommissioning a leaf switch results in the APIC attempting to remove all the leaf switch configurations on both the leaf switch and on the APIC. If the leaf switch is not reachable, only the APIC is cleaned. In this case, the user must manually wipe the leaf switch by resetting it.
Inactive	The node is unreachable. It had been discovered but currently is not accessible. For example, it may be powered off, or its cables may be disconnected.
Active	The node is an active member of the fabric.

Disabled interfaces can be ones blacklisted by an administrator or ones taken down because the APIC detects anomalies. Examples of link state anomalies include the following:

- A wiring mismatch, such as a spine connected to a spine, a leaf connected to a leaf, a spine connected to a leaf access port, a spine connected to a non-ACI node, or a leaf fabric port connected to a non-ACI device.
- A fabric name mismatch. The fabric name is stored in each ACI node. If a node is moved to another fabric without resetting it to a back to factory default state, it will retain the fabric name.
- A UUID mismatch causes the APIC to disable the node.

Note If an administrator uses the APIC to disable all the leaf nodes on a spine, a spine reboot is required to recover access to the spine.

Provisioning

The APIC provisioning method automatically brings up the ACI fabric with the appropriate connections. The following figure shows fabric provisioning.

Figure 22: Fabric Provisioning

After Link Layer Discovery Protocol (LLDP) discovery learns all neighboring connections dynamically, these connections are validated against a loose specification rule such as "LEAF can connect to only SPINE-L1-*" or "SPINE-L1-* can connect to SPINE-L2-* or LEAF." If a rule mismatch occurs, a fault occurs and the connection is blocked because a leaf is not allowed to be connected to another leaf, or a spine connected to a spine. In addition, an alarm is created to indicate that the connection needs attention. The Cisco ACI fabric administrator can import the names and serial numbers of all the fabric nodes from a text file into the APIC or allow the fabric to discover the serial numbers automatically and then assign names to the nodes using the APIC GUI, command-line interface (CLI), or API. The APIC is discoverable via SNMP. It has the following asysobjectId: `ciscoACIController OBJECT IDENTIFIER ::= { ciscoProducts 2238 }`

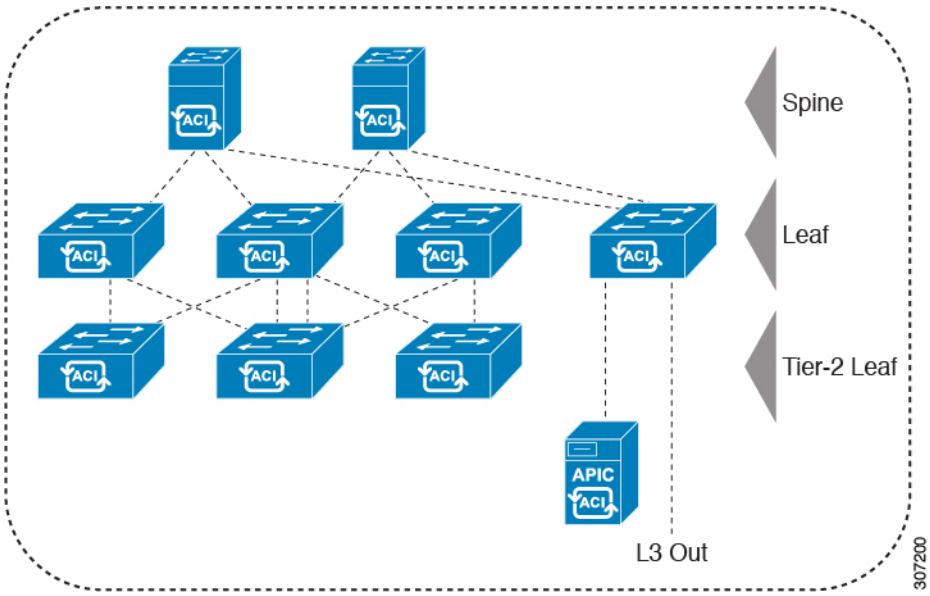
Multi-Tier Architecture

3-tier Core-Aggregation-Access architectures are common in data center network topologies. As of the Cisco APIC Release 4.1(1), you can create a multi-tier ACI fabric topology that corresponds to the Core-Aggregation-Access architecture, thus mitigating the need to upgrade costly components such as rack space or cabling. The addition of a tier-2 leaf layer makes this topology possible. The tier-2 leaf layer supports connectivity to hosts or servers on the downlink ports and connectivity to the leaf layer (aggregation) on the uplink ports.

In the multi-tier topology, the leaf switches initially have uplink connectivity to the spine switches and downlink connectivity to the tier-2 leaf switches. To make the entire topology an ACI fabric, all ports on the leaf switches connecting to tier-2 leaf fabric ports must be configured as fabric ports (if not already using the default fabric ports). After APIC discovers the tier-2 leaf switch, you can change the downlink port on the tier-2 leaf to a fabric port and connect to an uplink port on the middle layer leaf.

The following figure shows an example of a multi-tier fabric topology.

Figure 23: Multi-Tier Fabric Topology Example



While the topology in the above image shows the Cisco APIC and L3Out/EPG connected to the leaf aggregation layer, the tier-2 leaf access layer also supports connectivity to APICs and L3Out/EPGs.

APIC Cluster Management

Cluster Management Guidelines

The Cisco Application Policy Infrastructure Controller (APIC) cluster comprises multiple Cisco APICs that provide operators a unified real time monitoring, diagnostic, and configuration management capability for the Cisco Application Centric Infrastructure (ACI) fabric. To assure optimal system performance, use the following guidelines when making changes to the Cisco APIC cluster:

- Prior to initiating a change to the cluster, always verify its health. When performing planned changes to the cluster, all controllers in the cluster should be healthy. If one or more of the Cisco APICs' health status in the cluster is not "fully fit," remedy that situation before proceeding. Also, assure that cluster controllers added to the Cisco APIC are running the same version of firmware as the other controllers in the Cisco APIC cluster.
- We recommend that you have at least 3 active Cisco APICs in a cluster, along with additional standby Cisco APICs. In most cases, we recommend a cluster size of 3, 5, or 7 Cisco APICs. We recommend 4 Cisco APICs for a two site multi-pod fabric that has between 80 to 200 leaf switches.
- Disregard cluster information from Cisco APICs that are not currently in the cluster; they do not provide accurate cluster information.
- Cluster slots contain a Cisco APIC `ChassisID`. Once you configure a slot, it remains unavailable until you decommission the Cisco APIC with the assigned `ChassisID`.
- If a Cisco APIC firmware upgrade is in progress, wait for it to complete and the cluster to be fully fit before proceeding with any other changes to the cluster.
- When moving a Cisco APIC, first ensure that you have a healthy cluster. After verifying the health of the Cisco APIC cluster, choose the Cisco APIC that you intend to shut down. After the Cisco APIC has shut down, move the Cisco APIC, re-connect it, and then turn it back on. From the GUI, verify that the all controllers in the cluster return to a fully fit state.

Note Only move one Cisco APIC at a time.

- When moving a Cisco APIC that is connected to a set of leaf switches to another set of leaf switches or when moving a Cisco APIC to different port within the same leaf switch, first ensure that you have a healthy cluster. After verifying the health of the Cisco APIC cluster, choose the Cisco APIC that you intend to move and decommission it from the cluster. After the Cisco APIC is decomissioned, move the Cisco APIC and then commission it.
- Before configuring the Cisco APIC cluster, ensure that all of the Cisco APICs are running the same firmware version. Initial clustering of Cisco APICs running differing versions is an unsupported operation and may cause problems within the cluster.
- Unlike other objects, log record objects are stored only in one shard of a database on one of the Cisco APICs. These objects get lost forever if you decommission or replace that Cisco APIC.

- When you decommission a Cisco APIC, the Cisco APIC loses all fault, event, and audit log history that was stored in it. If you replace all Cisco APICs, you lose all log history. Before you migrate a Cisco APIC, we recommend that you manually backup the log history.

About Cold Standby for a Cisco APIC Cluster

The Cold Standby functionality for a Cisco Application Policy Infrastructure Controller (APIC) cluster enables you to operate the Cisco APICs in a cluster in an Active/Standby mode. In a Cisco APIC cluster, the designated active Cisco APICs share the load and the designated standby Cisco APICs can act as a replacement for any of the Cisco APICs in the active cluster.

As an admin user, you can set up the Cold Standby functionality when the Cisco APIC is launched for the first time. We recommend that you have at least three active Cisco APICs in a cluster, and one or more standby Cisco APICs. As an admin user, you can initiate the switch over to replace an active Cisco APIC with a standby Cisco APIC.

Important Notes

- The standby Cisco APICs are automatically updated with firmware updates to keep the backup Cisco APIC at same firmware version as the active cluster.
- During an upgrade process, after all the active Cisco APICs are upgraded, the standby Cisco APICs are also upgraded automatically.
- Temporary IDs are assigned to the standby Cisco APICs. After a standby Cisco APIC is switched over to an active Cisco APIC, a new ID is assigned.
- The admin login is not enabled on the standby Cisco APICs. To troubleshoot a Cold Standby Cisco APIC, you must log in to the standby using SSH as *rescue-user*.
- During the switch over, the replaced active Cisco APIC is powered down to prevent connectivity to the replaced Cisco APIC.
- Switch over fails under the following conditions:
  - If there is no connectivity to the standby Cisco APIC.
  - If the firmware version of the standby Cisco APIC is not the same as that of the active cluster.
- After switching over a standby Cisco APIC to be active, if it was the only standby, you must configure a new standby.
- The following limitations are observed for retaining out of band address for the standby Cisco APIC after a fail over:
  - The standby (new active) Cisco APIC may not retain its out of band address if more than 1 active Cisco APICs are down or unavailable.
  - The standby (new active) Cisco APIC may not retain its out of band address if it is in a different subnet than the active Cisco APIC. This limitation is only applicable for Cisco APIC release 2.x.
  - The standby (new active) Cisco APIC may not retain its IPv6 out of band address. This limitation is not applicable starting from Cisco APIC release 3.1x.
  - The standby (new active) Cisco APIC may not retain its out of band address if you have configured a non-static OOB management IP address policy for the replacement (old active) Cisco APIC.

- The standby (new active) Cisco APIC may not retain its out of band address if it is not in a pod that has an active Cisco APIC.

Note If you want to retain the standby Cisco APIC's out of band address despite the limitations, you must manually change the OOB policy for the replaced Cisco APIC after the replace operation had completed successfully.

- There must be three active Cisco APICs to add a standby Cisco APIC.
- The standby Cisco APIC does not participate in policy configuration or management.
- No information is replicated to the standby Cisco APICs, not even the administrator credentials.

Maintenance Mode

Following are terms that are helpful to understand when using maintenance mode:

- **Maintenance mode**: Used to isolate a switch from user traffic for debugging purposes. You can put a switch in **maintenance mode** by enabling the **Maintenance (GIR)** field in the **Fabric Membership** page in the APIC GUI, located at **Fabric** > **Inventory** > **Fabric Membership** (right-click on a switch and choose **Maintenance (GIR)**).

  If you put a switch in **maintenance mode**, that switch is not considered as a part of the operational ACI fabric infra and it will not accept regular APIC communications.

You can use maintenance mode to gracefully remove a switch and isolate it from the network in order to perform debugging operations. The switch is removed from the regular forwarding path with minimal traffic disruption.

In graceful removal, all external protocols are gracefully brought down except the fabric protocol (IS-IS) and the switch is isolated from the network. During maintenance mode, the maximum metric is advertised in IS-IS within the Cisco Application Centric Infrastructure (Cisco ACI) fabric and therefore the leaf switch in maintenance mode does not attract traffic from the spine switches. In addition, all front-panel interfaces on the switch are shutdown except for the fabric interfaces. To return the switch to its fully operational (normal) mode after the debugging operations, you must recommission the switch. This operation will trigger a stateless reload of the switch.

In graceful insertion, the switch is automatically decommissioned, rebooted, and recommissioned. When recommissioning is completed, all external protocols are restored and maximum metric in IS-IS is reset after 10 minutes.

The following protocols are supported:

- Border Gateway Protocol (BGP)
- Enhanced Interior Gateway Routing Protocol (EIGRP)
- Intermediate System-to-Intermediate System (IS-IS)
- Open Shortest Path First (OSPF)
- Link Aggregation Control Protocol (LACP)

Protocol Independent Multicast (PIM) is not supported.

Important Notes

- If a border leaf switch has a static route and is placed in 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

- While the switch is in maintenance mode, the Ethernet port module stops propagating the interface related notifications. As a result, if the remote switch is rebooted or the fabric link is flapped during this time, the fabric link will not come up afterward unless the switch is manually rebooted (using the **acidiag touch clean** command), decommissioned, and recommissioned.
- While the switch is in maintenance mode, CLI 'show' commands on the switch show the front panel ports as being in the up state and the BGP protocol as up and running. The interfaces are actually shut and all other adjacencies for BGP are brought down, but the displayed active states allow for debugging.
- For multi-pod / multi-site, **IS-IS metric for redistributed routes** should be set to less than 63 to minimize the traffic disruption when bringing the node back into the fabric. To set the **IS-IS metric for redistributed routes**, choose **Fabric** > **Fabric Policies** > **Pod Policies** > **IS-IS Policy**.
- Existing GIR supports all Layer 3 traffic diversion. With LACP, all the Layer 2 traffic is also diverted to the redundant node. Once a node goes into maintenance mode, LACP running on the node immediately informs neighbors that it can no longer be aggregated as part of port-channel. All traffic is then diverted to the vPC peer node.
- The following operations are not allowed in maintenance mode:
  - **Upgrade**: Upgrading the network to a newer version
  - **Stateful Reload**: Restarting the GIR node or its connected peers
  - **Stateless Reload**: Restarting with a clean configuration or power-cycle of the GIR node or its connected peers
  - **Link Operations**: Shut / no-shut or optics OIR on the GIR node or its peer node
  - **Configuration Change**: Any configuration change (such as clean configuration, import, or snapshot rollback)
  - **Hardware Change**: Any hardware change (such as adding, swapping, removing FRU's or RMA)

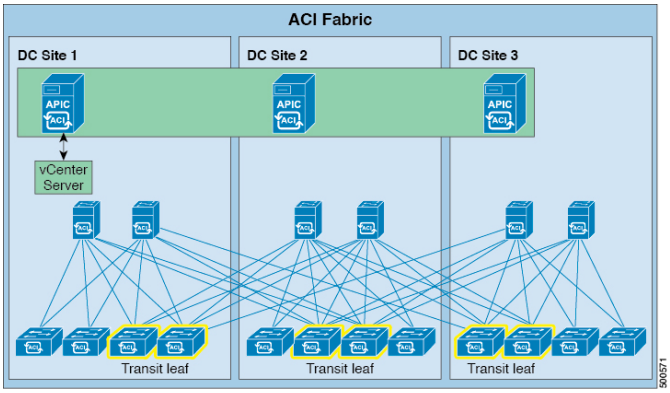
Stretched ACI Fabric Design Overview

Stretched ACI fabric is a partially meshed design that connects ACI leaf and spine switches distributed in multiple locations. Typically, an ACI fabric implementation is a single site where the full mesh design connects each leaf switch to each spine switch in the fabric, which yields the best throughput and convergence. In

multi-site scenarios, full mesh connectivity may be not possible or may be too costly. Multiple sites, buildings, or rooms can span distances that are not serviceable by enough fiber connections or are too costly to connect each leaf switch to each spine switch across the sites.

The following figure illustrates a stretched fabric topology.

Figure 24: ACI Stretched Fabric Topology



The stretched fabric is a single ACI fabric. The sites are one administration domain and one availability zone. Administrators are able to manage the sites as one entity; configuration changes made on any APIC controller node are applied to devices across the sites. The stretched ACI fabric preserves live VM migration capability across the sites. The ACI stretched fabric design has been validated, and is hence supported, on up to three interconnected sites.

An ACI stretched fabric essentially represents a "stretched pod" extended across different locations. A more solid, resilient (and hence recommended) way to deploy an ACI fabric in a distributed fashion across different locations is offered since ACI release 2.0(1) with the ACI Multi-Pod architecture. For more information, refer to the following white paper:

https://www.cisco.com/c/en/us/solutions/collateral/data-center-virtualization/application-centric-infrastructure/white-paper-c11-737855.html

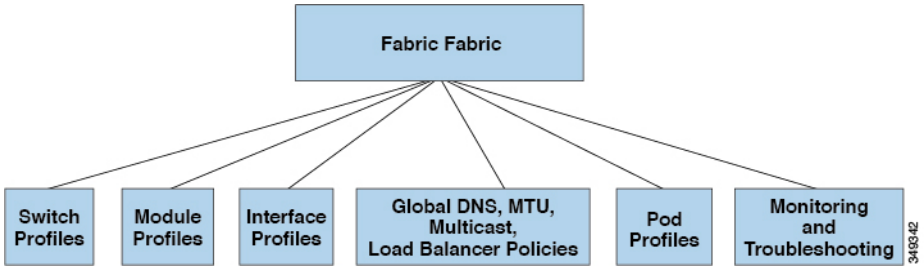
Stretched ACI Fabric Related Documents

The KB Stretched ACI Fabric Design Overview technical note provides design guidelines regarding traffic flow, APIC cluster redundancy and operational considerations for implementing an ACI fabric stretched across multiple sites.

Fabric Policies Overview

Fabric policies govern the operation of internal fabric interfaces and enable the configuration of various functions, protocols, and interfaces that connect spine and leaf switches. Administrators who have fabric administrator privileges can create new fabric policies according to their requirements. The APIC enables administrators to select the pods, switches, and interfaces to which they will apply fabric policies. The following figure provides an overview of the fabric policy model.

Figure 25: Fabric Polices Overview



Fabric policies are grouped into the following categories:

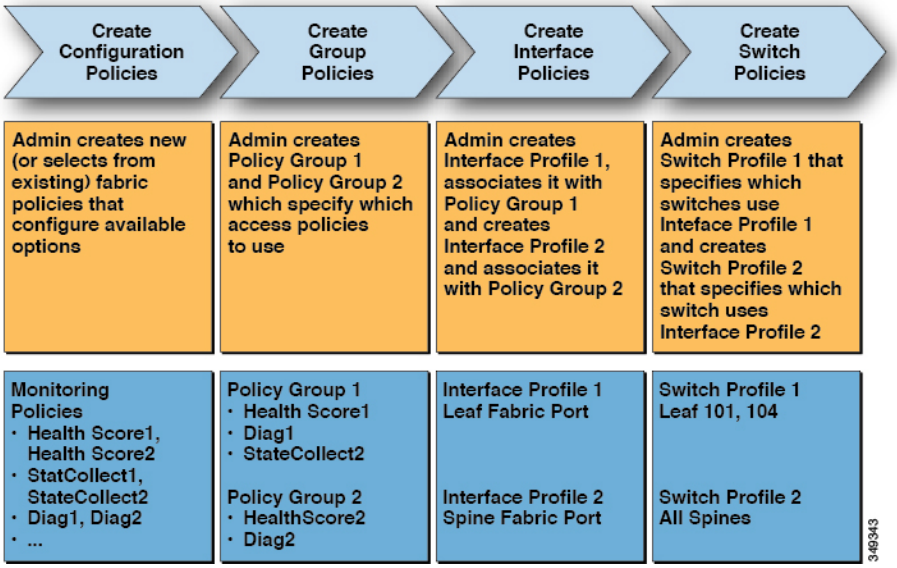
- Switch profiles specify which switches to configure and the switch configuration policy.
- Module profiles specify which spine switch modules to configure and the spine switch configuration policy.
- Interface profiles specify which fabric interfaces to configure and the interface configuration policy.
- Global policies specify DNS, fabric MTU default, multicast tree, and load balancer configurations to be used throughout the fabric.
- Pod profiles specify date and time, SNMP, council of oracle protocol (COOP), IS-IS and Border Gateway Protocol (BGP) route reflector policies.
- Monitoring and troubleshooting policies specify what to monitor, thresholds, how to handle faults and logs, and how to perform diagnostics.

Fabric Policy Configuration

Fabric policies configure interfaces that connect spine and leaf switches. Fabric policies can enable features such as monitoring (statistics collection and statistics export), troubleshooting (on-demand diagnostics and SPAN), IS-IS, council of oracle protocol (COOP), SNMP, Border Gateway Protocol (BGP) route reflectors, DNS, or Network Time Protocol (NTP).

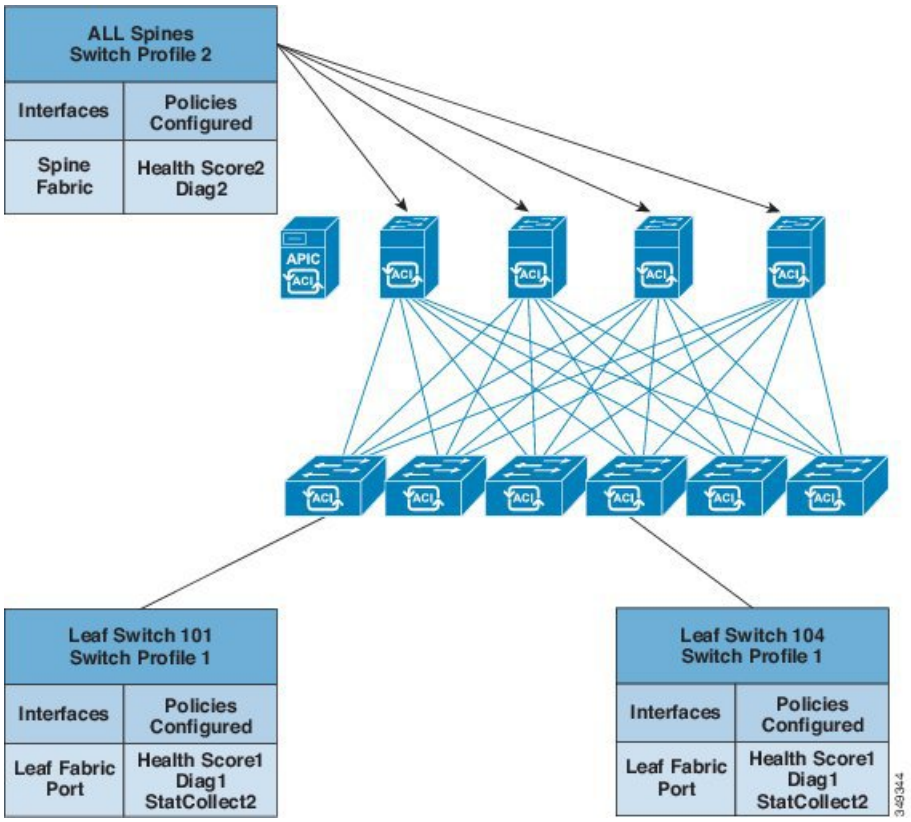
To apply a configuration across the fabric, an administrator associates a defined group of policies to interfaces on switches in a single step. In this way, large numbers of interfaces across the fabric can be configured at once; configuring one port at a time is not scalable. The following figure shows how the process works for configuring the ACI fabric.

Figure 26: Fabric Policy Configuration Process



The following figure shows the result of applying Switch Profile 1 and Switch Profile 2 to the ACI fabric.

Figure 27: Application of a Fabric Switch Policy



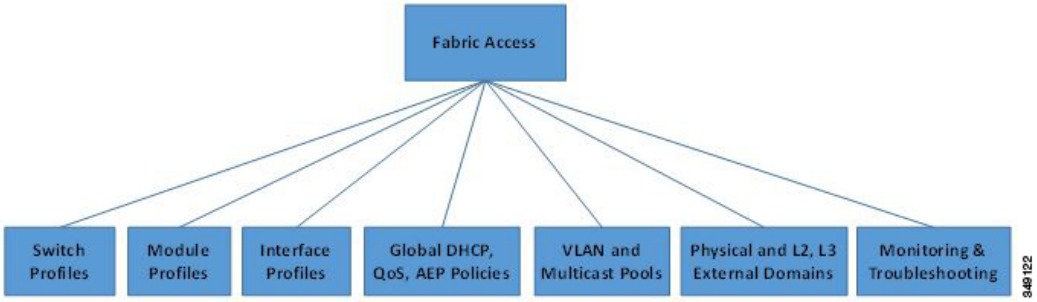
This combination of infrastructure and scope enables administrators to manage fabric configuration in a scalable fashion. These configurations can be implemented using the REST API, the CLI, or the GUI. The Quick Start Fabric Interface Configuration wizard in the GUI automatically creates the necessary underlying objects to implement such policies.

Access Policies Overview

Access policies configure external-facing interfaces that connect to devices such as virtual machine controllers and hypervisors, hosts, network attached storage, routers, or Fabric Extender (FEX) interfaces. Access policies enable the configuration of port channels and virtual port channels, protocols such as Link Layer Discovery Protocol (LLDP), Cisco Discovery Protocol (CDP), or Link Aggregation Control Protocol (LACP), and features such as statistics gathering, monitoring, and diagnostics.

The following figure provides an overview of the access policy model.

Figure 28: Access Policy Model Overview



Access policies are grouped into the following categories:

- Switch profiles specify which switches to configure and the switch configuration policy.
- Module profiles specify which leaf switch access cards and access modules to configure and the leaf switch configuration policy.
- Interface profiles specify which access interfaces to configure and the interface configuration policy.
- Global policies enable the configuration of DHCP, QoS, and attachable access entity (AEP) profile functions that can be used throughout the fabric. AEP profiles provide a template to deploy hypervisor policies on a large set of leaf ports and associate a Virtual Machine Management (VMM) domain and the physical network infrastructure. They are also required for Layer 2 and Layer 3 external network connectivity.
- Pools specify VLAN, VXLAN, and multicast address pools. A pool is a shared resource that can be consumed by multiple domains such as VMM and Layer 4 to Layer 7 services. A pool represents a range of traffic encapsulation ide.jpgiers (for example, VLAN IDs, VNIDs, and multicast addresses).
- Physical and external domains policies include the following:
  - External bridged domain Layer 2 domain profiles contain the port and VLAN specifications that a bridged Layer 2 network connected to the fabric uses.
  - External routed domain Layer 3 domain profiles contain the port and VLAN specifications that a routed Layer 3 network connected to the fabric uses.

- Physical domain policies contain physical infrastructure specifications, such as ports and VLAN, used by a tenant or endpoint group.

- Monitoring and troubleshooting policies specify what to monitor, thresholds, how to handle faults and logs, and how to perform diagnostics.

Access Policy Configuration

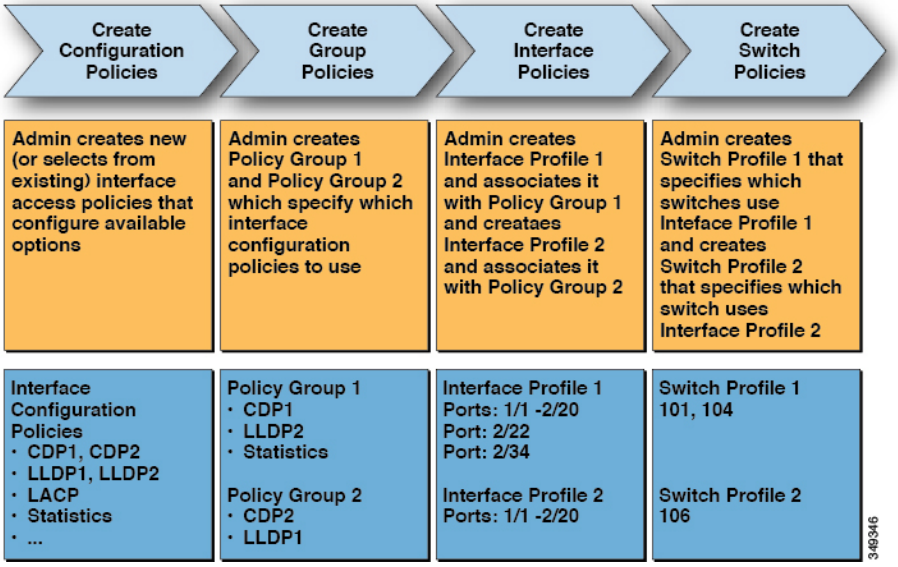
Access policies configure external-facing interfaces that do not connect to a spine switch. External-facing interfaces connect to external devices such as virtual machine controllers and hypervisors, hosts, routers, or Fabric Extenders (FEXs). Access policies enable an administrator to configure port channels and virtual port channels, protocols such as LLDP, CDP, or LACP, and features such as monitoring or diagnostics.

Sample XML policies for switch interfaces, port channels, virtual port channels, and change interface speeds are provided in *Cisco APIC Rest API Configuration Guide*.

Note While tenant network policies are configured separately from fabric access policies, tenant policies are not activated unless the underlying access policies they depend on are in place.

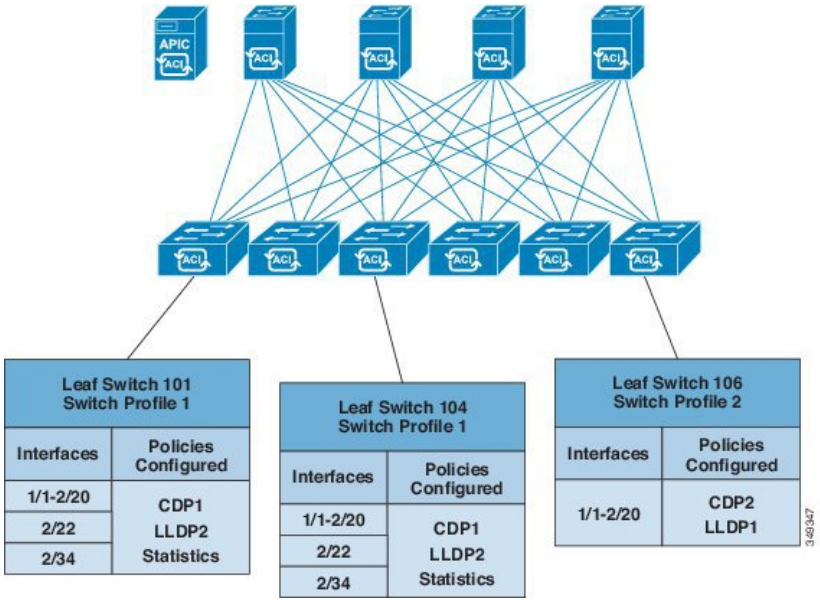
To apply a configuration across a potentially large number of switches, an administrator defines switch profiles that associate interface configurations in a single policy group. In this way, large numbers of interfaces across the fabric can be configured at once. Switch profiles can contain symmetric configurations for multiple switches or unique special purpose configurations. The following figure shows the process for configuring access to the ACI fabric.

Figure 29: Access Policy Configuration Process



The following figure shows the result of applying Switch Profile 1 and Switch Profile 2 to the ACI fabric.

Figure 30: Applying an Access Switch Policy



This combination of infrastructure and scope enables administrators to manage fabric configuration in a scalable fashion. These configurations can be implemented using the REST API, the CLI, or the GUI. The Quick Start Interface, PC, VPC Configuration wizard in the GUI automatically creates the necessary underlying objects to implement such policies.

Port Channel and Virtual Port Channel Access

Access policies enable an administrator to configure port channels and virtual port channels. Sample XML policies for switch interfaces, port channels, virtual port channels, and change interface speeds are provided in *Cisco APIC Rest API Configuration Guide*.

FEX Virtual Port Channels

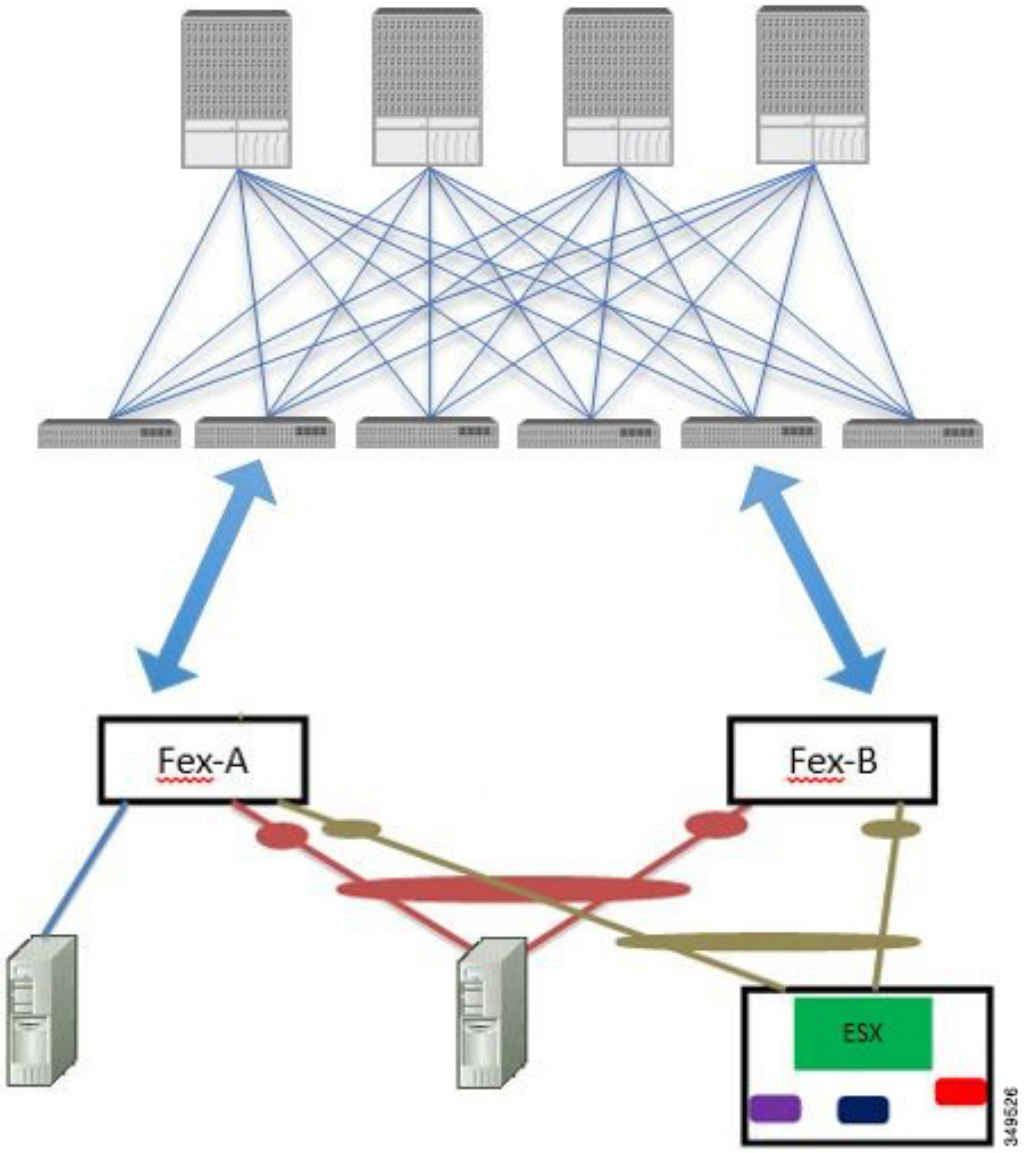
The ACI fabric supports Cisco Fabric Extender (FEX) server-side virtual port channels (vPC), also known as an FEX straight-through vPC.

Note When creating a vPC domain between two leaf switches, both switches must be in the same switch generation, one of the following:

- Generation 1 - Cisco Nexus N9K switches without “EX” or "FX" on the end of the switch name; for example, N9K-9312TX
- Generation 2 – Cisco Nexus N9K switches with “EX” or "FX" on the end of the switch model name; for example, N9K-93108TC-EX

Switches such as these two are not compatible vPC peers. Instead, use switches of the same generation.

Figure 31: Supported FEX vPC Topologies



Supported FEX vPC port channel topologies include the following:

- Both VTEP and non-VTEP hypervisors behind a FEX.
- Virtual switches (such as AVS or VDS) connected to two FEXs that are connected to the ACI fabric (vPCs directly connected on physical FEX ports is not supported - a vPC is supported only on port channels).

Note When using GARP as the protocol to n.jpgy of IP to MAC binding changes to different interfaces on the same FEX you must set the bridge domain mode to **ARP Flooding** and enable **EP Move Detection Mode**: **GARP-based Detection**, on the **L3 Configuration** page of the bridge domain wizard. This workaround is only required with Generation 1 switches. With Generation 2 switches or later, this is not an issue.

Fibre Channel and FCoE

For Fibre Channel and FCoE configuration information, see the *Cisco APIC Layer 2 Networking Configuration Guide*.

Supporting Fibre Channel over Ethernet Traffic on the Cisco ACI Fabric

Cisco Application Centric Infrastructure (ACI) enables you to configure and manage support for Fibre Channel over Ethernet (FCoE) traffic on the Cisco ACI fabric.

FCoE is a protocol that encapsulates Fibre Channel packets within Ethernet packets, thus enabling storage traffic to move seamlessly between a Fibre Channel SAN and an Ethernet network.

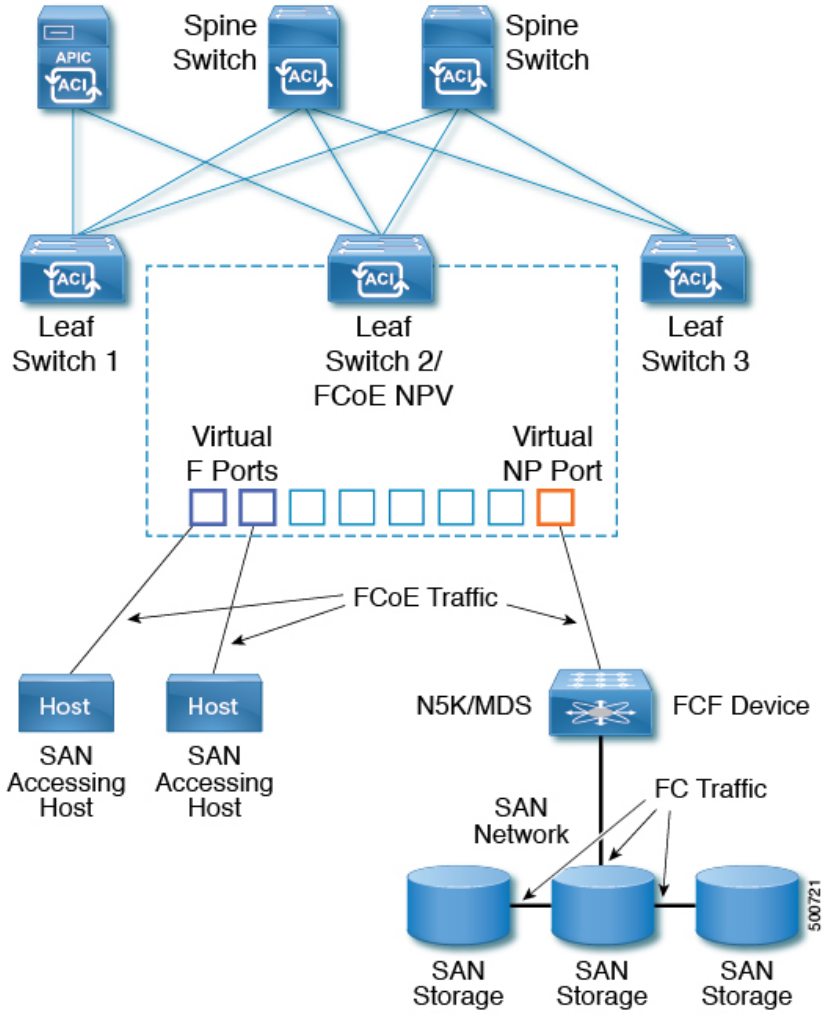
A typical implementation of FCoE protocol support on the Cisco ACI fabric enables hosts located on the Ethernet-based Cisco ACI fabric to communicate with SAN storage devices located on a Fibre Channel network. The hosts are connecting through virtual F ports deployed on an Cisco ACI leaf switch. The SAN storage devices and Fibre Channel network are connected through a Fibre Channel Forwarding (FCF) bridge to the Cisco ACI fabric through a virtual NP port, deployed on the same Cisco ACI leaf switch as is the virtual F port. Virtual NP ports and virtual F ports are also referred to generically as virtual Fibre Channel (vFC) ports.

Note In the FCoE topology, the role of the Cisco ACI leaf switch is to provide a path for FCoE traffic between the locally connected SAN hosts and a locally connected FCF device. The leaf switch does not perform local switching between SAN hosts, and the FCoE traffic is not forwarded to a spine switch.

Topology Supporting FCoE Traffic Through Cisco ACI

The topology of a typical configuration supporting FCoE traffic over the Cisco ACI fabric consists of the following components:

Figure 32: Cisco ACI Topology Supporting FCoE Traffic



- One or more Cisco ACI leaf switches configured through Fibre Channel SAN policies to function as an NPV backbone.
- Selected interfaces on the NPV-configured leaf switches configured to function as virtual F ports, which accommodate FCoE traffic to and from hosts running SAN management or SAN-consuming applications.
- Selected interfaces on the NPV-configured leaf switches configured to function as virtual NP ports, which accommodate FCoE traffic to and from a Fibre Channel Forwarding (FCF) bridge.

The FCF bridge receives Fibre Channel traffic from Fibre Channel links typically connecting SAN storage devices and encapsulates the Fibre Channel packets into FCoE frames for transmission over the Cisco ACI fabric to the SAN management or SAN Data-consuming hosts. It receives FCoE traffic and repackages it back to the Fibre Channel for transmission over the Fibre Channel network.

Note In the above Cisco ACI topology, FCoE traffic support requires direct connections between the hosts and virtual F ports and direct connections between the FCF device and the virtual NP port.

Cisco Application Policy Infrastructure Controller (APIC) servers enable an operator to configure and monitor the FCoE traffic through the Cisco APIC GUI, or NX-OS-style CLI, or through application calls to the REST API.

Topology Supporting FCoE Initialization

In order for FCoE traffic flow to take place as described, you must also set up separate VLAN connectivity over which SAN Hosts broadcast FCoE Initialization protocol (FIP) packets to discover the interfaces enabled as F ports.

vFC Interface Configuration Rules

Whether you set up the vFC network and EPG deployment through the Cisco APIC GUI, NX-OS-style CLI, or the REST API, the following general rules apply across platforms:

- F port mode is the default mode for vFC ports. NP port mode must be specifically configured in the Interface policies.
- The load balancing default mode is for leaf-switch or interface level vFC configuration is src-dst-ox-id.
- One VSAN assignment per bridge domain is supported.
- The allocation mode for VSAN pools and VLAN pools must always be static.
- vFC ports require association with a VSAN domain (also called Fibre Channel domain) that contains VSANs mapped to VLANs.

Fibre Channel Connectivity Overview

Cisco ACI supports Fibre Channel (FC) connectivity on a leaf switch using N-Port Virtualization (NPV) mode. NPV allows the switch to aggregate FC traffic from locally connected host ports (N ports) into a node proxy (NP port) uplink to a core switch.

A switch is in NPV mode after enabling NPV. NPV mode applies to an entire switch. Each end device connected to an NPV mode switch must log in as an N port to use this feature (loop-attached devices are not supported). All links from the edge switches (in NPV mode) to the NPV core switches are established as NP ports (not E ports), which are used for typical inter-switch links.

Note In the FC NPV application, the role of the ACI leaf switch is to provide a path for FC traffic between the locally connected SAN hosts and a locally connected core switch. The leaf switch does not perform local switching between SAN hosts, and the FC traffic is not forwarded to a spine switch.

FC NPV Benefits

FC NPV provides the following:

- Increases the number of hosts that connect to the fabric without adding domain IDs in the fabric. The domain ID of the NPV core switch is shared among multiple NPV switches.
- FC and FCoE hosts connect to SAN fabrics using native FC interfaces.

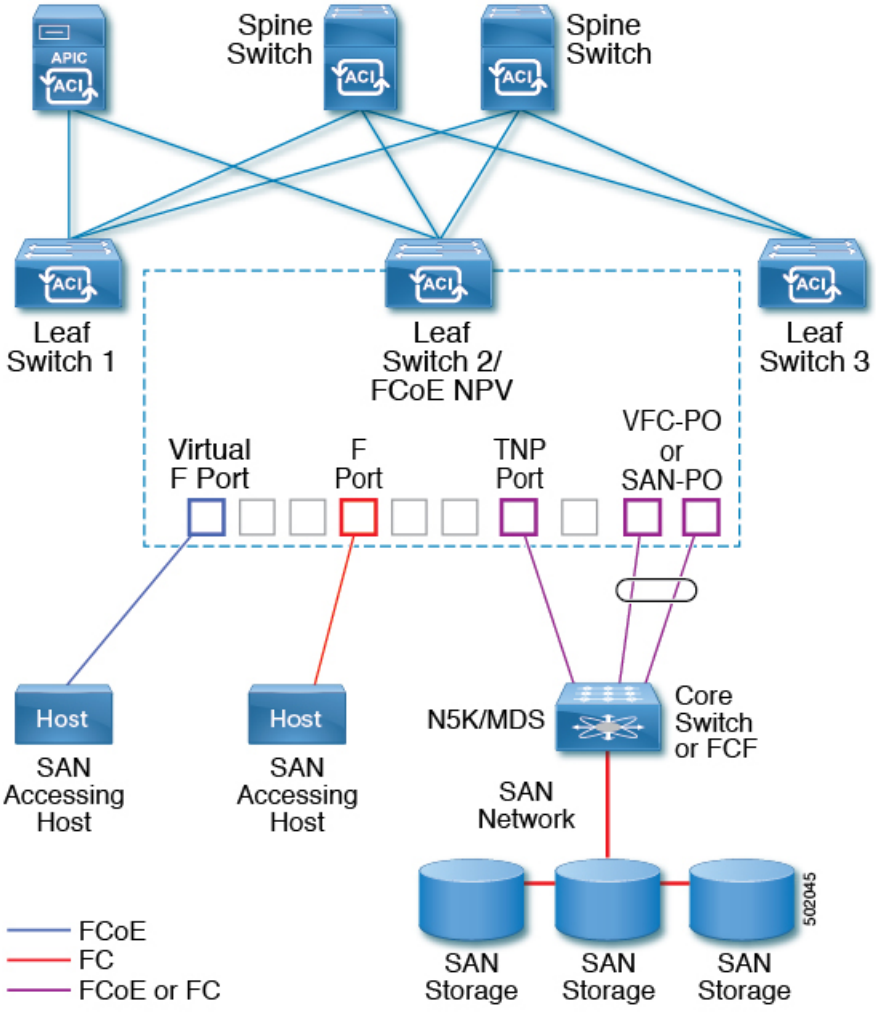
- Automatic traffic mapping for load balancing. For newly added servers connected to NPV, traffic is automatically distributed among the external uplinks based on current traffic loads.
- Static traffic mapping. A server connected to NPV can be statically mapped to an external uplink.

FC NPV Mode

Feature-set fcoe-npv in ACI will be enabled automatically by default when the first FCoE/FC configuration is pushed.

FC Topology

The topology of various configurations supporting FC traffic over the ACI fabric is shown in the following figure:



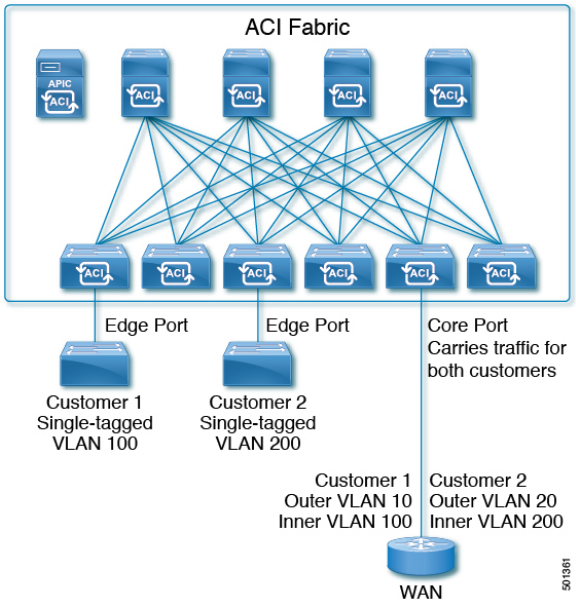
- Server/storage host interfaces on the ACI leaf switch can be configured to function as either native FC ports or as virtual FC (FCoE) ports.
- An uplink interface to a FC core switch can be configured as any of the following port types:

  - native FC NP port
  - SAN-PO NP port
- An uplink interface to a FCF switch can be configured as any of the following port types:
  - virtual (vFC) NP port
  - vFC-PO NP port
- N-Port ID Virtualization (NPIV) is supported and enabled by default, allowing an N port to be assigned multiple N port IDs or Fibre Channel IDs (FCID) over a single link.
- Trunking can be enabled on an NP port to the core switch. Trunking allows a port to support more than one VSAN. When trunk mode is enabled on an NP port, it is referred to as a TNP port.
- Multiple FC NP ports can be combined as a SAN port channel (SAN-PO) to the core switch. Trunking is supported on a SAN port channel.
- FC F ports support 4/16/32 Gbps and auto speed configuration, but 8Gbps is not supported for host interfaces. The default speed is "auto."
- FC NP ports support 4/8/16/32 Gbps and auto speed configuration. The default speed is "auto."
- Multiple FDISC followed by Flogi (nested NPIV) is supported with FC/FCoE host and FC/FCoE NP links.
- An FCoE host behind a FEX is supported over an FCoE NP/uplink.
- Starting in the APIC 4.1(1) release, an FCoE host behind a FEX is supported over the Fibre Channel NP/uplink.
- All FCoE hosts behind one FEX can either be load balanced across multiple vFC and vFC-PO uplinks, or through a single Fibre Channel/SAN port channel uplink.
- SAN boot is supported on a FEX through an FCoE NP/uplink.
- Starting in the APIC 4.1(1) release, SAN boot is also supported over a FC/SAN-PO uplink.
- SAN boot is supported over vPC for FCoE hosts that are connected through FEX.

802.1Q Tunnels

About ACI 802.1Q Tunnels

Figure 33: ACI 802.1Q Tunnels



You can configure 802.1Q tunnels on edge (tunnel) ports to enable point-to-multi-point tunneling of Ethernet frames in the fabric, with Quality of Service (QoS) priority settings. A Dot1q tunnel transports untagged, 802.1Q tagged, and 802.1ad double-tagged frames as-is across the fabric. Each tunnel carries the traffic from a single customer and is associated with a single bridge domain. Cisco Application Centric Infrastructure (ACI) front panel ports can be part of a Dot1q tunnel. Layer 2 switching is done based on the destination MAC (DMAC) and regular MAC learning is done in the tunnel. Edge port Dot1q tunnels are supported on Cisco Nexus 9000 series switches with "EX" or later suffixes in the switch model name.

You can configure multiple 802.1Q tunnels on the same core port to carry double-tagged traffic from multiple customers, each distinguished with an access encapsulation configured for each 802.1Q tunnel. You can also disable MAC address learning on 802.1Q tunnels. Both edge ports and core ports can belong to an 802.1Q tunnel with access encapsulation and disabled MAC address learning. Both edge ports and core ports in Dot1q tunnel are supported on Cisco Nexus 9000 series switches with "FX" or later suffixes in the switch model name.

IGMP and MLD packets can be forwarded through 802.1Q tunnels.

Terms used in this document may be different in the **Cisco Nexus 9000 Series** documents.

Table 3: 802.1Q Tunnel Terminology

ACI Documents	Cisco Nexus 9000 Series Documents
Edge Port	Tunnel Port

ACI Documents	Cisco Nexus 9000 Series Documents
Core Port	Trunk Port

The following guidelines and restrictions apply:

- Layer 2 tunneling of VTP, CDP, LACP, LLDP, and STP protocols is supported with the following restrictions:
  - Link Aggregation Control Protocol (LACP) tunneling functions as expected only with point-to-point tunnels using individual leaf interfaces. It is not supported on port channels (PCs) or virtual port channels (vPCs).
  - CDP and LLDP tunneling with PCs or vPCs is not deterministic; it depends on the link it chooses as the traffic destination.
  - To use VTP for Layer 2 protocol tunneling, CDP must be enabled on the tunnel.
  - STP is not supported in an 802.1Q tunnel bridge domain when Layer 2 protocol tunneling is enabled and the bridge domain is deployed on Dot1q tunnel core ports.
  - Cisco ACI leaf switches react to STP TCN packets by flushing the end points in the tunnel bridge domain and flooding them in the bridge domain.
  - CDP and LLDP tunneling with more than two interfaces flood packets on all interfaces.
  - The destination MAC address of Layer 2 protocol packets tunneled from edge to core ports is rewritten as 01-00-0c-cd-cd-d0 and the destination MAC address of Layer 2 protocol packets tunneled from core to edge ports is rewritten with the standard default MAC address for the protocol.
- If a PC or vPC is the only interface in a Dot1q tunnel and it is deleted and reconfigured, remove the association of the PC/VPC to the Dot1q tunnel and reconfigure it.
- For 802.1Q tunnels deployed on switches that have EX in the product ID, Ethertype combinations of 0x8100+0x8100, 0x8100+0x88a8, 0x88a8+0x8100, and 0x88a8+0x88a8 for the first two VLAN tags are not supported.

  If the tunnels are deployed on a combination of EX and FX or later switches, then this restriction still applies.

  If the tunnels are deployed only on switches that have FX or later in the product ID, then this restriction does not apply.
- For core ports, the Ethertypes for double-tagged frames must be 0x8100 followed by 0x8100.
- You can include multiple edge ports and core ports (even across leaf switches) in a Dot1q tunnel.
- An edge port may only be part of one tunnel, but a core port can belong to multiple Dot1q tunnels.
- Regular EPGs can be deployed on core ports that are used in 802.1Q tunnels.
- L3Outs are not supported on interfaces enabled for Dot1q tunnel.
- FEX interfaces are not supported as members of a Dot1q tunnel.
- Interfaces configured as breakout ports do not support 802.1Q tunnels.
- Interface-level statistics are supported for interfaces in Dot1q tunnel, but statistics at the tunnel level are not supported.

Dynamic Breakout Ports

Configuration of Dynamic Breakout Ports

Breakout cables are suitable for very short links and offer a cost effective way to connect within racks and across adjacent racks.

Breakout enables a 40 Gigabit (Gb) port to be split into four independent and logical 10Gb ports or a 100Gb port to be split into four independent and logical 25Gb ports.

Before you configure breakout ports, connect a 40Gb port to four 10Gb ports or a 100Gb port to four 25Gb ports with one of the following cables:

- Cisco QSFP-4SFP10G
- Cisco QSFP-4SFP25G
- Cisco QSFP-4X10G-AOC
- MPO to breakout splitter cable with QSFP-40G-SR4 and 4 x SFP-10G-SR on the ends
- MPO to breakout splitter cable with QSFP-100G-SR4-S and 4 x SFP-25G-SR-S on the ends

Note For the supported optics and cables, see the *Cisco Optics-to-Device Compatibility Matrix*:

https://tmgmatrix.cisco.com/

The 40Gb to 10Gb dynamic breakout feature is supported on the access facing ports of the following switches:

- N9K-C93180LC-EX
- N9K-C93180YC-FX
- N9K-C9336C-FX2
- N9K-C93360YC-FX2
- N9K-C93216TC-FX2
- N9K-C93108TC-FX3P (beginning in the 5.1(3) release)
- N9K-C93180YC-FX3 (beginning in the 5.1(3) release)
- N9K-C93600CD-GX (beginning in the 5.1(3) release)
- N9K-C9364C-GX (beginning in the 5.1(3) release)

The 100Gb to 25Gb breakout feature is supported on the access facing ports of the following switches:

- N9K-C93180LC-EX
- N9K-C9336C-FX2
- N9K-C93180YC-FX

- N9K-C93360YC-FX2
- N9K-C93216TC-FX2
- N9K-C93108TC-FX3P (beginning in the 5.1(3) release)
- N9K-C93180YC-FX3 (beginning in the 5.1(3) release)
- N9K-C93600CD-GX (beginning in the 5.1(3) release)
- N9K-C9364C-GX (beginning in the 5.1(3) release)

Observe the following guidelines and limitations:

- Breakout ports are supported only on downlinks and converted downlinks.
- The following switches support dynamic breakouts (both 100Gb and 40Gb) on profiled QSFP ports:
  - Cisco N9K-C93180YC-FX
  - Cisco N9K-C93216TC-FX2
  - Cisco N9K-C93360YC-FX2
  - Cisco N9K-C93600CD-GX

    This applies only to ports 1/25 to 34. Ports 1/29 to 34 can be used for dynamic breakouts if the ports are converted to down links.
  - Cisco N9K-C9336C-FX2

    You can configure up to 34 dynamic breakouts.
  - Cisco N9K-C9364C-GX (beginning in the 5.1(3) release)

    You can configure up to 30 dynamic breakouts on odd-numbered profiled QSFP ports from 1/1 to 59.
  - Cisco N9K-93600CD-GX (beginning in the 5.1(3) release)

    You can configure up to 12 dynamic breakouts out of 24 40/100G ports and up to 10 dynamic breakouts out of ports 25 to 34. Ports 29 to 34 can be breakouts if the ports are converted to down links. The last 2 ports (ports 35 and 36) are reserved for fabric links.
- The Cisco N9K-C9336C-FX2 switch supports LACP fast hello on the breakout sub-port.
- Breakout ports cannot be used for Cisco Application Policy Infrastructure Controller (APIC) connectivity.
- Fast Link Failover policies are not supported on the same port with the dynamic breakout feature.
- Breakout subports can be used in the same way other port types in the policy model are used.
- When a port is enabled for dynamic breakout, other policies (expect monitoring policies) on the parent port are no longer valid.
- When a port is enabled for dynamic breakout, other EPG deployments on the parent port are no longer valid.
- A breakout sub-port can not be further broken out using a breakout policy group.

- A breakout sub-port supports LACP. The LACP transmit rate configuration defined in the "default" port channel member policy is used by default. The LACP transmit rate can be changed by either changing the "default" port channel member policy or using an override policy group on each PC/vPC interface policy group.
- If the LACP transmit rate on port channels that have breakout sub-ports need to be changed, then all the port channels that include breakout sub-ports need to use the same LACP transmit rate configuration. You can configure an override policy to set the transmit rate as follows:
  1. Configure/change the default port channel member policy to include Fast Transmit Rate (**Fabric** > **Access Policies** > **Policies** > **Interface** > **Port Channel Member**).
  2. Configure all the PC/vPC interface policy groups to include the above default port channel member policy under the override policy groups (**Fabric** > **Access Policies** > **Interfaces** > **Leaf Interfaces** > **Policy Groups** > **PC/vPC Interface**).
- Following guidelines and limitations apply to the Cisco N9K-C9364C-GX switch:
  - Odd numbered ports (on rows 1 and row 3) support breakout. Adjacent even numbered ports (on row 2 or row 4) will be disabled ("hw-disabled"). This is applicable to ports 1/1 to 60.
  - The last 2 ports (1/63 and 64) are reserved for fabric links.
  - Ports 1/61 and 62 can be converted to down links, but breakout is not supported. Breakout ports and 40/100G non-breakout ports cannot be mixed in a set of 4 ports starting from 1/1, such as 1/1 to 4 or 1/5 to 8.

    For example, if port 1/1 is breakout enabled, port 1/3 can be used with breakout enabled or native 10G. Port 1/3 will be error-disabled if it is 40/100G.
  - The max number of downlinks are 30 x 4 ports 10/25 (breakout) + 2 ports (1/61 and 62) = 122 ports. Ports 1/63 and 64 are reserved for fabric links and even numbered ports from 1/2 to 60 are error-disabled.
  - This switch supports 10G with QSA on all ports. Native 10G requires QSA.
- Following guidelines and limitations apply to the Cisco N9K-93600CD-GX switch:
  - Odd numbered ports (all ports on row 1) support breakout. Even numbered ports on row 2 will be disabled ("hw-disabled"). This is applicable only to ports 1 to 24.
  - Breakout and 40/100G non-breakout cannot be mixed in a set of 4 ports starting from 1/1 until 1/24, such as 1/1 to 4 or 1/5 to 8. For example:
    - For ports 1/1 to 24, you can have 4 ports per set.

      For example, if port 1/1 is breakout enabled, port 1/3 can be used with breakout enabled or native 10G. Port 1/3 will be error-disabled if it is 40/100G.
    - For ports 1/25 to 28, you can have 2 ports per set.

      For example, even if port 1/25 is breakout enabled, port 1/27 can be used with 40/100G.
  - The maximum number of downlinks are 12 x 4 ports 10/25G (breakout) + 10 x 4 ports 10/25G (breakout) = 88 ports. Ports 35 and 36 are reserved for fabric links and 12 ports are disabled.
  - This switch supports 10G with QSA on all ports. Native 10G requires QSA.

Configuring Port Profiles

Uplink and downlink conversion is supported on Cisco Nexus 9000 series switches with names that end in EX or FX, and later (for example, N9K-C9348GC-FXP or N9K-C93240YC-FX2). A FEX connected to converted downlinks is also supported.

For information about the supported Cisco switches, see Port Profile Configuration Summary, on page 78.

When an uplink port is converted to a downlink port, it acquires the same capabilities as any other downlink port.

Restrictions

- Fast Link Failover policies and port profiles are not supported on the same port. If port profile is enabled, Fast Link Failover cannot be enabled or vice versa.
- The last 2 uplink ports of supported leaf switches cannot be converted to downlink ports (they are reserved for uplink connections.)
- Dynamic breakouts (both 100Gb and 40Gb) are supported on profiled QSFP ports on the N9K-C93180YC-FX switch. Breakout and port profile are supported together for conversion of uplink to downlink on ports 49-52. Breakout (both **10g-4x** and **25g-4x** options) is supported on downlink profiled ports.
- The N9K-C9348GC-FXP does not support FEX.
- Breakout is supported only on downlink ports, and not on fabric ports that are connected to other switches.
- A Cisco ACI leaf switch cannot have more than 56 fabric links.
- Reloading a switch after changing a switch's port profile configuration interrupts traffic through the data plane.

Guidelines

In converting uplinks to downlinks and downlinks to uplinks, consider the following guidelines.

Subject	Guideline
Decommissioning nodes with port profiles	If a decommissioned node has the Port Profile feature deployed on it, the port conversions are not removed even after decommissioning the node. It is necessary to manually delete the configurations after decommission, for the ports to return to the default state. To do this, log onto the switch, run the `setup-clean-config.sh` script, and wait for it to run. Then, enter the `reload` command. Optionally, you can specify `-k` with the `setup-clean-config.sh` script to allow the port-profile setting to persist across the reload, making an additional reboot unnecessary.

Subject	Guideline
Maximum uplink port limit	When the maximum uplink port limit is reached and ports 25 and 27 are converted from uplink to downlink and back to uplink on Cisco 93180LC-EX switches: On Cisco N9K-93180LC-EX switches, ports 25 and 27 are the original uplink ports. Using the port profile, if you convert port 25 and 27 to downlink ports, ports 29, 30, 31, and 32 are still available as four original uplink ports. Because of the threshold on the number of ports (which is maximum of 12 ports) that can be converted, you can convert 8 more downlink ports to uplink ports. For example, ports 1, 3, 5, 7, 9, 13, 15, 17 are converted to uplink ports and ports 29, 30, 31 and 32 are the 4 original uplink ports (the maximum uplink port limit on Cisco 93180LC-EX switches). When the switch is in this state and if the port profile configuration is deleted on ports 25 and 27, ports 25 and 27 are converted back to uplink ports, but there are already 12 uplink ports on the switch (as mentioned earlier). To accommodate ports 25 and 27 as uplink ports, 2 random ports from the port range 1, 3, 5, 7, 9, 13, 15, 17 are denied the uplink conversion and this situation cannot be controlled by the user. Therefore, it is mandatory to clear all the faults before reloading the leaf node to avoid any unexpected behavior regarding the port type. It should be noted that if a node is reloaded without clearing the port profile faults, especially when there is a fault related to limit-exceed, the port might not be in an expected operational state.

Breakout Limitations

Switch	Releases	Limitations
N9K-C93180LC-EX	Cisco APIC 3.1(1) and later	• 40Gb and 100Gb dynamic breakouts are supported on ports 1 through 24 on odd numbered ports. • When the top ports (odd ports) are broken out, then the bottom ports (even ports) are error disabled. • Port profiles and breakouts are not supported on the same port. However, you can apply a port profile to convert a fabric port to a downlink, and then apply a breakout configuration.

Switch	Releases	Limitations
N9K-C9336C-FX2	Cisco APIC 4.2(4) and later	• 40Gb and 100Gb dynamic breakouts are supported on ports 1 through 34. • A port profile cannot be applied to a port with breakout enabled. However, you can apply a port profile to convert a fabric port to a downlink, and then apply a breakout configuration. • All 34 ports can be configured as breakout ports. • If you want to apply a breakout configuration on 34 ports, you must configure a port profile on the ports to have 34 downlink ports, then you must reboot the leaf switch. • If you apply a breakout configuration to a leaf switch for multiple ports at the same time, it can take up to 10 minutes for the hardware of 34 ports to be programmed. The ports remain down until the programming completes. The delay can occur for a new configuration, after a clean reboot, or during switch discovery.
N9K-C9336C-FX2	Cisco APIC 3.2(1) up through, but not including, 4.2(4)	• 40Gb and 100Gb dynamic breakouts are supported on ports 1 through 30. • Port profiles and breakouts are not supported on the same port. However, you can apply a port profile to convert a fabric port to a downlink, and then apply a breakout configuration. • A maximum of 20 ports can be configured as breakout ports.

Switch	Releases	Limitations
N9K-C93180YC-FX	Cisco APIC 3.2(1) and later	• 40Gb and 100Gb dynamic breakouts are supported on ports 49 though 52, when they are on profiled QSFP ports. To use them for dynamic breakout, perform the following steps: • Convert ports 49-52 to front panel ports (downlinks). • Perform a port-profile reload, using one of the following methods: • In the Cisco APIC GUI, navigate to **Fabric** > **Inventory** > **Pod** > **Leaf**, right-click **Chassis** and choose **Reload**. • In the iBash CLI, enter the **reload** command. • Apply breakouts on the profiled ports 49-52. • Ports 53 and 54 do not support either port profiles or breakouts.
N9K-C93240YC-FX2	Cisco APIC 4.0(1) and later	Breakout is not supported on converted downlinks.

Port Profile Configuration Summary

The following table summarizes supported uplinks and downlinks for the switches that support port profile conversions from uplink to downlink and downlink to uplink.

Switch Model	Default Links	Max Uplinks (Fabric Ports)	Max Downlinks (Server Ports)	Release Supported
N9K-C9348GC-FXP[1]	48 x 100M/1G BASE-T downlinks 4 x 10/25-Gbps SFP28 downlinks 2 x 40/100-Gbps QSFP28 uplinks	48 x 100M/1G BASE-T downlinks 4 x 10/25-Gbps SFP28 uplinks 2 x 40/100-Gbps QSFP28 uplinks	Same as default port configuration	3.1(1)

Switch Model	Default Links	Max Uplinks (Fabric Ports)	Max Downlinks (Server Ports)	Release Supported
N9K-C93180LC-EX	24 x 40 Gbps QSFP28 downlinks (ports 1-24) 2 x 40/100 Gbps QSFP28 uplinks (ports 25, 27) 4 x 40/100 Gbps QSFP28 uplinks (ports 29-32) Or 12 x 100 Gbps QSFP28 downlinks (odd number ports from 1-24) 2 x 40/100 Gbps QSFP28 uplinks (ports 25, 27) 4 x 40/100 Gbps QSFP28 uplinks (ports 29-32)	18 x 40-Gbps QSFP28 downlinks (from 1-24) 6 x 40-Gbps QSFP28 uplinks(from 1-24) 2 x 40/100-Gbps QSFP28 uplinks(25, 27) 4 x 40/100-Gbps QSFP28 uplinks(29-32) Or 6 x 100-Gbps QSFP28 downlinks(odd number from 1-24) 6 x 100-Gbps QSFP28 uplinks(odd number from 1-24) 2 x 40/100-Gbps QSFP28 uplinks(25, 27) 4 x 40/100-Gbps QSFP28 uplinks(29-32)	24 x 40-Gbps QSFP28 downlinks(1-24) 2 x 40/100-Gbps QSFP28 downlinks(25, 27) 4 x 40/100-Gbps QSFP28 uplinks(29-32) Or 12 x 100-Gbps QSFP28 downlinks(odd number from 1-24) 2 x 40/100-Gbps QSFP28 downlinks (25, 27) 4 x 40/100-Gbps QSFP28 uplinks(29-32)	3.1(1)
N9K-C93180YC-EX N9K-C93180YC-FX N9K-C93180YC-FX3	48 x 10/25-Gbps fiber downlinks 6 x 40/100-Gbps QSFP28 uplinks	Same as default port configuration	48 x 10/25-Gbps fiber downlinks 4 x 40/100-Gbps QSFP28 downlinks 2 x 40/100-Gbps QSFP28 uplinks	3.1(1)
		48 x 10/25-Gbps fiber uplinks 6 x 40/100-Gbps QSFP28 uplinks		4.0(1) 5.1(3)
N9K-C93108TC-EX[2] N9K-C93108TC-FX[2] N9K-C93108TC-FX3	48 x 10GBASE-T downlinks 6 x 40/100-Gbps QSFP28 uplinks	Same as default port configuration	48 x 10/25-Gbps fiber downlinks 4 x 40/100-Gbps QSFP28 downlinks 2 x 40/100-Gbps QSFP28 uplinks	3.1(1) 4.0(1) 5.1(3)

Switch Model	Default Links	Max Uplinks (Fabric Ports)	Max Downlinks (Server Ports)	Release Supported
N9K-C9336C-FX2	30 x 40/100-Gbps QSFP28 downlinks 6 x 40/100-Gbps QSFP28 uplinks	18 x 40/100-Gbps QSFP28 downlinks 18 x 40/100-Gbps QSFP28 uplinks	Same as default port configuration	3.2(1)
		18 x 40/100-Gbps QSFP28 downlinks 18 x 40/100-Gbps QSFP28 uplinks	34 x 40/100-Gbps QSFP28 downlinks 2 x 40/100-Gbps QSFP28 uplinks	3.2(3)
		36 x 40/100-Gbps QSFP28 uplinks	34 x 40/100-Gbps QSFP28 downlinks 2 x 40/100-Gbps QSFP28 uplinks	4.1(1)
N9K-93240YC-FX2	48 x 10/25-Gbps fiber downlinks 12 x 40/100-Gbps QSFP28 uplinks	Same as default port configuration	48 x 10/25-Gbps fiber downlinks 10 x 40/100-Gbps QSFP28 downlinks 2 x 40/100-Gbps QSFP28 uplinks	4.0(1)
		48 x 10/25-Gbps fiber uplinks 12 x 40/100-Gbps QSFP28 uplinks		4.1(1)
N9K-C93216TC-FX2	96 x 10G BASE-T downlinks 12 x 40/100-Gbps QSFP28 uplinks	Same as default port configuration	96 x 10G BASE-T downlinks 10 x 40/100-Gbps QSFP28 downlinks 2 x 40/100-Gbps QSFP28 uplinks	4.1(2)
N9K-C93360YC-FX2	96 x 10/25-Gbps SFP28 downlinks 12 x 40/100-Gbps QSFP28 uplinks	44 x 10/25Gbps SFP28 downlinks 52 x 10/25Gbps SFP28 uplinks 12 x 40/100Gbps QSFP28 uplinks	96 x 10/25-Gbps SFP28 downlinks 10 x 40/100-Gbps QSFP28 downlinks 2 x 40/100-Gbps QSFP28 uplinks	4.1(2)
N9K-C93600CD-GX	28 x 40/100 Gbps QSFP28 downlinks (ports 1-28) 8 x 40/100/400 Gbps QSFP-DD uplinks (ports 29-36)	28 x 40/100-Gbps QSFP28 uplinks 8 x 40/100/400-Gbps QSFP-DD uplinks	28 x 40/100-Gbps QSFP28 downlinks 6 x 40/100/400-Gbps QSFP-DD downlinks 2 x 40/100/400-Gbps QSFP-DD uplinks	4.2(2)

Switch Model	Default Links	Max Uplinks (Fabric Ports)	Max Downlinks (Server Ports)	Release Supported
N9K-C9364C-GX	48 x 40/100 Gbps QSFP28 downlinks (ports 1-48) 16 x 40/100 Gbps QSFP28 uplinks (ports 49-64)	64 x 40/100-Gbps QSFP28 uplinks	62 x 40/100-Gbps QSFP28 downlinks 2 x 40/100-Gbps QSFP28 uplinks	4.2(3)
N9K-C9316D-GX	12 x 40/100/400 Gbps QSFP-DD downlinks (ports 1-12) 4 x 40/100/400 Gbps QSFP-DD uplinks (ports 13-16)	16 x 40/100/400 Gbps QSFP-DD uplinks	14 x 40/100/400 Gbps QSFP-DD downlinks	5.1(4)

1 Does not support FEX.

2 Only uplink to downlink conversion is supported.

Port Tracking Policy for Fabric Port Failure Detection

Fabric port failure detection can be enabled in the port tracking system settings. The port tracking policy monitors the status of fabric ports between leaf switches and spine switches, and ports between tier-1 leaf switches and tier-2 leaf switches. When an enabled port tracking policy is triggered, the leaf switches take down all access interfaces on the switch that have EPGs deployed on them.

If you enabled the **Include APIC ports when port tracking is triggered** option, port tracking disables Cisco Application Policy Infrastructure Controller (APIC) ports when the leaf switch loses connectivity to all fabric ports (that is, there are 0 fabric ports). Enable this feature only if the Cisco APICs are dual- or multihomed to the fabric. Bringing down the Cisco APIC ports helps in switching over to the secondary port in the case of a dual-homed Cisco APIC.

Note Port tracking is located under **System** > **System Settings** > **Port Tracking**.

The port tracking policy specifies the number of fabric port connections that trigger the policy, and a delay timer for bringing the leaf switch access ports back up after the number of specified fabric ports is exceeded.

The following example illustrates how a port tracking policy behaves:

- The port tracking policy specifies that the threshold of active fabric port connections each leaf switch that triggers the policy is 2.
- The port tracking policy triggers when the number of active fabric port connections from the leaf switch to the spine switches drops to 2.
- Each leaf switch monitors its fabric port connections and triggers the port tracking policy according to the threshold specified in the policy.

- When the fabric port connections come back up, the leaf switch waits for the delay timer to expire before bringing its access ports back up. This gives the fabric time to reconverge before allowing traffic to resume on leaf switch access ports. Large fabrics may need the delay timer to be set for a longer time.

Note Use caution when configuring this policy. If the port tracking setting for the number of active spine ports that triggers port tracking is too high, all leaf switch access ports will be brought down.

Q-in-Q Encapsulation Mapping for EPGs

Using Cisco Application Policy Infrastructure Controller (APIC), you can map double-tagged VLAN traffic ingressing on a regular interface, PC, or vPC to an EPG. When this feature is enabled, when double-tagged traffic enters the network for an EPG, both tags are processed individually in the fabric and restored to double-tags when egressing the Cisco Application Centric Infrastructure (ACI) switch. Ingressing single-tagged and untagged traffic is dropped.

The following guidelines and limitations apply:

- This feature is only supported on Cisco Nexus 9300-FX platform switches.
- Both the outer and inner tag must be of EtherType 0x8100.
- MAC learning and routing are based on the EPG port, sclass, and VRF instance, not on the access encapsulations.
- QoS priority settings are supported, derived from the outer tag on ingress, and rewritten to both tags on egress.
- EPGs can simultaneously be associated with other interfaces on a leaf switch, that are configured for single-tagged VLANs.
- Service graphs are supported for provider and consumer EPGs that are mapped to Q-in-Q encapsulated interfaces. You can insert service graphs, as long as the ingress and egress traffic on the service nodes is in single-tagged encapsulated frames.
- When vPC ports are enabled for Q-in-Q encapsulation mode, VLAN consistency checks are not performed.

The following features and options are not supported with this feature:

- Per-port VLAN feature
- FEX connections
- Mixed mode

  For example, an interface in Q-in-Q encapsulation mode can have a static path binding to an EPG with double-tagged encapsulation only, not with regular VLAN encapsulation.
- STP and the "Flood in Encapsulation" option
- Untagged and 802.1p mode
- Multi-pod and Multi-Site

- Legacy bridge domain
- L2Out and L3Out connections
- VMM integration
- Changing a port mode from routed to Q-in-Q encapsulation mode
- Per-VLAN mis-cabling protocol on ports in Q-in-Q encapsulation mode

Layer 2 Multicast

About Cisco APIC and IGMP Snooping

IGMP snooping is the process of listening to Internet Group Management Protocol (IGMP) network traffic. The feature allows a network switch to listen in on the IGMP conversation between hosts and routers and filter multicasts links that do not need them, thus controlling which ports receive specific multicast traffic.

Cisco APIC provides support for the full IGMP snooping feature included on a traditional switch such as the N9000 standalone.

- Policy-based IGMP snooping configuration per bridge domain

  APIC enables you to configure a policy in which you enable, disable, or customize the properties of IGMP Snooping on a per bridge-domain basis. You can then apply that policy to one or multiple bridge domains.

- Static port group implementation

  IGMP static port grouping enables you to pre-provision ports, already statically-assigned to an application EPG, as 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 (also called, *static-binding ports*) assigned to an application EPG.

- Access group configuration for application EPGs

  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.

Note You can use **vzAny** to enable protocols such as IGMP Snooping for all the EPGs in a VRF. For more information about **vzAny**, see Use vzAny to Automatically Apply Communication Rules to all EPGs in a VRF.

To use **vzAny**, navigate to **Tenants** > ***tenant-name*** > **Networking** > **VRFs** > ***vrf-name*** > **EPG Collection for VRF**.

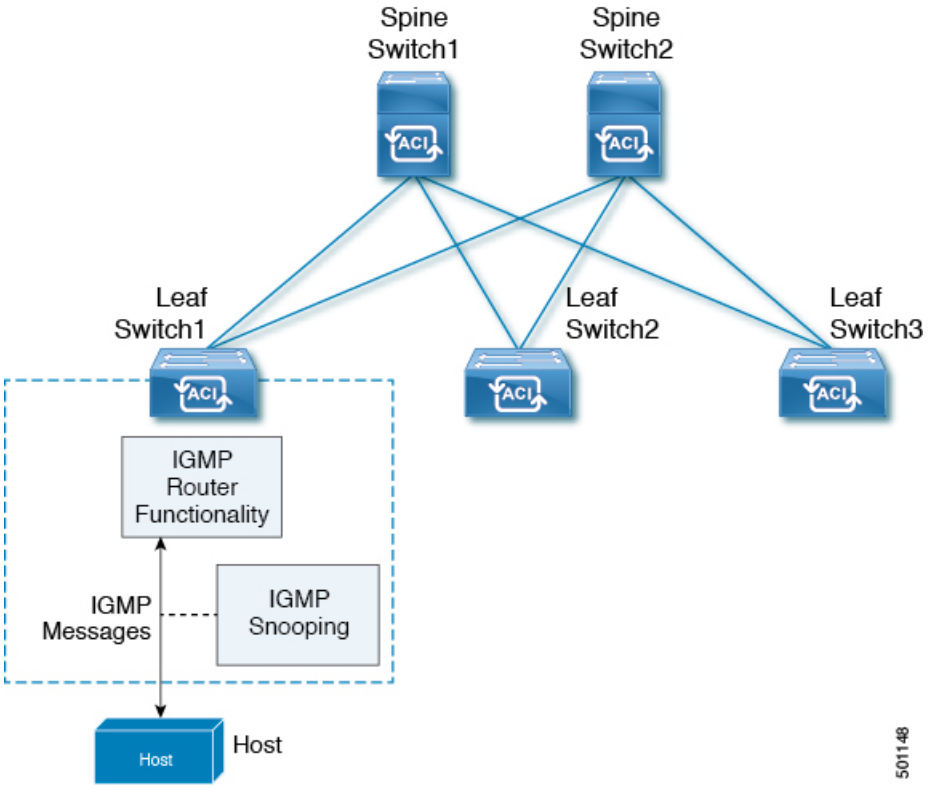
How IGMP Snooping is Implemented in the ACI Fabric

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 34: 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 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.

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.

Fabric Secure Mode

Fabric secure mode prevents parties with physical access to the fabric equipment from adding a switch or APIC controller to the fabric without manual authorization by an administrator. Starting with release 1.2(1x), the firmware checks that switches and controllers in the fabric have valid serial numbers associated with a valid Cisco digitally signed certificate. This validation is performed upon upgrade to this release or during an initial installation of the fabric. The default setting for this feature is permissive mode; an existing fabric continues to run as it has after an upgrade to release 1.2(1) or later. An administrator with fabric-wide access rights must enable strict mode. The following table summarizes the two modes of operation:

Permissive Mode (default)	Strict Mode
Allows an existing fabric to operate normally even though one or more switches have an invalid certificate.	Only switches with a valid Cisco serial number and SSL certificate are allowed.
Does not enforce serial number based authorization .	Enforces serial number authorization.
Allows auto-discovered controllers and switches to join the fabric without enforcing serial number authorization.	Requires an administrator to manually authorize controllers and switches to join the fabric.

Configuring Fast Link Failover Policy

Fast Link Failover policy is applicable to uplinks on switch models with -EX, -FX, and -FX2 suffixes. It efficiently load balances the traffic based on the uplink MAC status. With this functionality, the switch performs Layer 2 or Layer 3 lookup and it provides an output Layer 2 interface (uplinks) based on the packet hash algorithm by considering the uplink status. This functionality reduces the data traffic convergence to less than 200 milliseconds.

See the following limitations on configuring Fast Link Failover:

- Fast Link Failover and port profiles are not supported on the same interface. If port profile is enabled, Fast Link Failover cannot be enabled or vice versa.
- Configuring remote leaf does not work with Fast Link Failover. In this case, Fast Link Failover policies will not work and no fault will be generated.
- When Fast Link Failover policy is enabled, configuring SPAN on individual uplinks will not work. No fault will be generated while attempting to enable SPAN on individual uplinks but Fast Link Failover policy can be enabled on all uplinks together or it can be enabled on an individual downlink.

Note Fast Link Failover is located under **Fabric** > **Access Policies** > **Policies** > **Switch** > **Fast Link Failover**.

About Port Security and ACI

The port security feature protects the ACI fabric from being flooded with unknown MAC addresses by limiting the number of MAC addresses learned per port. The port security feature support is available for physical ports, port channels, and virtual port channels.

Port Security and Learning Behavior

For non-vPC ports or port channels, whenever a learn event comes for a new endpoint, a verification is made to see if a new learn is allowed. If the corresponding interface has a port security policy not configured or disabled, the endpoint learning behavior is unchanged with what is supported. If the policy is enabled and the limit is reached, the current supported action is as follows:

- Learn the endpoint and install it in the hardware with a drop action.
- Silently discard the learn.

If the limit is not reached, the endpoint is learned and a verification is made to see if the limit is reached because of this new endpoint. If the limit is reached, and the learn disable action is configured, learning will be disabled in the hardware on that interface (on the physical interface or on a port channel or vPC). If the limit is reached and the learn disable action is not configured, the endpoint will be installed in hardware with a drop action. Such endpoints are aged normally like any other endpoints.

When the limit is reached for the first time, the operational state of the port security policy object is updated to reflect it. A static rule is defined to raise a fault so that the user is alerted. A syslog is also raised when the limit is reached.

In case of vPC, when the MAC limit is reached, the peer leaf switch is also notified so learning can be disabled on the peer. As the vPC peer can be rebooted any time or vPC legs can become unoperational or restart, this state will be reconciled with the peer so vPC peers do not go out of sync with this state. If they get out of sync, there can be a situation where learning is enabled on one leg and disabled on the other leg.

By default, once the limit is reached and learning is disabled, it will be automatically re-enabled after the default timeout value of 60 seconds.

Protect Mode

The protect mode prevents further port security violations from occurring. Once the MAC limit exceeds the maximum configured value on a port, all traffic from excess MAC addresses will be dropped and further learning is disabled.

Port Security at Port Level

In the APIC, the user can configure the port security on switch ports. Once the MAC limit has exceeded the maximum configured value on a port, all traffic from the exceeded MAC addresses is forwarded. The following attributes are supported:

- **Port Security Timeout**—The current supported range for the timeout value is from 60 to 3600 seconds.
- **Violation Action**—The violation action is available in protect mode. In the protect mode, MAC learning is disabled and MAC addresses are not added to the CAM table. Mac learning is re-enabled after the configured timeout value.
- **Maximum Endpoints**—The current supported range for the maximum endpoints configured value is from 0 to 12000. If the maximum endpoints value is 0, the port security policy is disabled on that port.

Port Security Guidelines and Restrictions

The guidelines and restrictions are as follows:

- Port security is available per port.
- Port security is supported for physical ports, port channels, and virtual port channels (vPCs).
- Static and dynamic MAC addresses are supported.

- MAC address moves are supported from secured to unsecured ports and from unsecured ports to secured ports.
- The MAC address limit is enforced only on the MAC address and is not enforced on a MAC and IP address.
- Port security is not supported with the Fabric Extender (FEX).

About First Hop Security

First-Hop Security (FHS) features enable a better IPv4 and IPv6 link security and management over the layer 2 links. In a service provider environment, these features closely control address assignment and derived operations, such as Duplicate Address Detection (DAD) and Address Resolution (AR).

The following supported FHS features secure the protocols and help build a secure endpoint database on the fabric leaf switches, that are used to mitigate security threats such as MIM attacks and IP thefts:

- ARP Inspection—allows a network administrator to intercept, log, and discard ARP packets with invalid MAC address to IP address bindings.
- ND Inspection—learns and secures bindings for stateless autoconfiguration addresses in Layer 2 neighbor tables.
- DHCP Inspection—validates DHCP messages received from untrusted sources and filters out invalid messages.
- RA Guard—allows the network administrator to block or reject unwanted or rogue router advertisement (RA) guard messages.
- IPv4 and IPv6 Source Guard—blocks any data traffic from an unknown source.
- Trust Control—a trusted source is a device that is under your administrative control. These devices include the switches, routers, and servers in the Fabric. Any device beyond the firewall or outside the network is an untrusted source. Generally, host ports are treated as untrusted sources.

FHS features provide the following security measures:

- **Role Enforcement**—Prevents untrusted hosts from sending messages that are out the scope of their role.
- **Binding Enforcement**—Prevents address theft.
- **DoS Attack Mitigations**—Prevents malicious end-points to grow the end-point database to the point where the database could stop providing operation services.
- **Proxy Services**—Provides some proxy-services to increase the efficiency of address resolution.

FHS features are enabled on a per tenant bridge domain (BD) basis. As the bridge domain, may be deployed on a single or across multiple leaf switches, the FHS threat control and mitigation mechanisms cater to a single switch and multiple switch scenarios.

About MACsec

MACsec is an IEEE 802.1AE standards based Layer 2 hop-by-hop encryption that provides data confidentiality and integrity for media access independent protocols.

MACsec, provides MAC-layer encryption over wired networks by using out-of-band methods for encryption keying. The MACsec Key Agreement (MKA) Protocol provides the required session keys and manages the required encryption keys.

The 802.1AE encryption with MKA is supported on all types of links, that is, host facing links (links between network access devices and endpoint devices such as a PC or IP phone), or links connected to other switches or routers.

MACsec encrypts the entire data except for the Source and Destination MAC addresses of an Ethernet packet. The user also has the option to skip encryption up to 50 bytes after the source and destination MAC address.

To provide MACsec services over the WAN or Metro Ethernet, service providers offer Layer 2 transparent services such as E-Line or E-LAN using various transport layer protocols such as Ethernet over Multiprotocol Label Switching (EoMPLS) and L2TPv3.

The packet body in an EAP-over-LAN (EAPOL) Protocol Data Unit (PDU) is referred to as a MACsec Key Agreement PDU (MKPDU). When no MKPDU is received from a participants after 3 hearbeats (each hearbeat is of 2 seconds), peers are deleted from the live peer list. For example, if a client disconnects, the participant on the switch continues to operate MKA until 3 heartbeats have elapsed after the last MKPDU is received from the client.

APIC Fabric MACsec

The APIC will be responsible for the MACsec keychain distribution to all the nodes in a Pod or to particular ports on a node. Below are the supported MACsec keychain and MACsec policy distribution supported by the APIC.

- A single user provided keychain and policy per Pod
- User provided keychain and user provided policy per fabric interface
- Auto generated keychain and user provided policy per Pod

A node can have multiple policies deployed for more than one fabric link. When this happens, the per fabric interface keychain and policy are given preference on the affected interface. The auto generated keychain and associated MACsec policy are then given the least preference.

APIC MACsec supports two security modes. The MACsec **must secure** only allows encrypted traffic on the link while the **should secure** allows both clear and encrypted traffic on the link. Before deploying MACsec in **must secure** mode, the keychain must be deployed on the affected links or the links will go down. For example, a port can turn on MACsec in **must secure** mode before its peer has received its keychain resulting in the link going down. To address this issue the recommendation is to deploy MACsec in **should secure** mode and once all the links are up then change the security mode to **must secure**.

Note Any MACsec interface configuration change will result in packet drops.

MACsec policy definition consists of configuration specific to keychain definition and configuration related to feature functionality. The keychain definition and feature functionality definitions are placed in separate policies. Enabling MACsec per Pod or per interface involves deploying a combination of a keychain policy and MACsec functionality policy.

Note Using internal generated keychains do not require the user to specify a keychain.

APIC Access MACsec

MACsec is used to secure links between leaf switch L3out interfaces and external devices. APIC provides GUI and CLI to allow users to program the MACsec keys and MacSec configuration for the L3Out interfaces on the fabric on a per physical/pc/vpc interface basis. It is the responsibility of the user to make sure that the external peer devices are programmed with the correct MacSec information.

Data Plane Policing

Use data plane policing (DPP) to manage bandwidth consumption on ACI fabric access interfaces. DPP policies can apply to egress traffic, ingress traffic, or both. DPP monitors the data rates for a particular interface. When the data rate exceeds user-configured values, marking or dropping of packets occurs immediately. Policing does not buffer the traffic; therefore, the transmission delay is not affected. When traffic exceeds the data rate, the ACI fabric can either drop the packets or mark QoS fields in them.

Note Egress data plane policers are not supported on switched virtual interfaces (SVI).

DPP policies can be single-rate, dual-rate, and color-aware. Single-rate policies monitor the committed information rate (CIR) of traffic. Dual-rate policers monitor both CIR and peak information rate (PIR) of traffic. In addition, the system monitors associated burst sizes. Three colors, or conditions, are determined by the policer for each packet depending on the data rate parameters supplied: conform (green), exceed (yellow), or violate (red).

Typically, DPP policies are applied to physical or virtual layer 2 connections for virtual or physical devices such as servers or hypervisors, and on layer 3 connections for routers. DPP policies applied to leaf switch access ports are configured in the fabric access (`infraInfra`) portion of the ACI fabric, and must be configured by a fabric administrator. DPP policies applied to interfaces on border leaf switch access ports (`l3extOut` or `l2extOut`) are configured in the tenant (`fvTenant`) portion of the ACI fabric, and can be configured by a tenant administrator.

Only one action can be configured for each condition. For example, a DPP policy can to conform to the data rate of 256000 bits per second, with up to 200 millisecond bursts. The system applies the conform action to traffic that falls within this rate, and it would apply the violate action to traffic that exceeds this rate. Color-aware policies assume that traffic has been previously marked with a color. This information is then used in the actions taken by this type of policer.

Scheduler

A schedule allows operations, such as configuration import/export or tech support collection, to occur during one or more specified windows of time.

A schedule contains a set of time windows (occurrences). These windows can be one time only or can recur at a specified time and day each week. The options defined in the window, such as the duration or the maximum number of tasks to be run, determine when a scheduled task executes. For example, if a change cannot be deployed during a given maintenance window because the maximum duration or number of tasks has been reached, that deployment is carried over to the next maintenance window.

Each schedule checks periodically to see whether the APIC has entered one or more maintenance windows. If it has, the schedule executes the deployments that are eligible according to the constraints specified in the maintenance policy.

A schedule contains one or more occurrences, which determine the maintenance windows associated with that schedule. An occurrence can be one of the following:

- One-time Window—Defines a schedule that occurs only once. This window continues until the maximum duration of the window or the maximum number of tasks that can be run in the window has been reached.
- Recurring Window—Defines a repeating schedule. This window continues until the maximum number of tasks or the end of the day specified in the window has been reached.

After a schedule is configured, it can then be selected and applied to the following export and firmware policies during their configuration:

- Tech Support Export Policy
- Configuration Export Policy -- Daily AutoBackup
- Firmware Download

Firmware Upgrade

Policies on the APIC manage the following aspects of the firmware upgrade processes:

- What version of firmware to use.
- Downloading firmware images from Cisco to the APIC repository.
- Compatibility enforcement.
- What to upgrade:
  - Switches
  - The APIC
  - The compatibility catalog
- When the upgrade will be performed.
- How to handle failures (retry, pause, ignore, and so on).

Each firmware image includes a compatibility catalog that identifies supported types and switch models. The APIC maintains a catalog of the firmware images, switch types, and models that are allowed to use that firmware image. The default setting is to reject a firmware update when it does not conform to the compatibility catalog.

The APIC, which performs image management, has an image repository for compatibility catalogs, APIC controller firmware images, and switch images. The administrator can download new firmware images to the APIC image repository from an external HTTP server or SCP server by creating an image source policy.

Firmware Group policies on the APIC define what firmware version is needed.

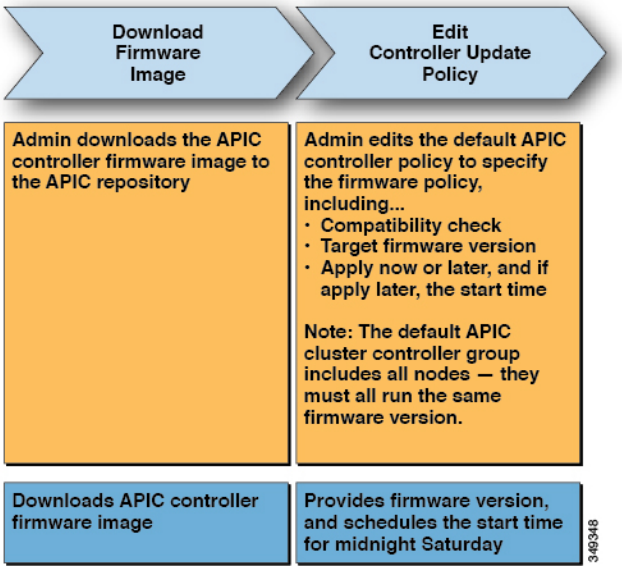
Maintenance Group policies define when to upgrade firmware, which nodes to upgrade, and how to handle failures. In addition, maintenance Group policies define groups of nodes that can be upgraded together and assign those maintenance groups to schedules. Node group options include all leaf nodes, all spine nodes, or sets of nodes that are a portion of the fabric.

The APIC controller firmware upgrade policy always applies to all nodes in the cluster, but the upgrade is always done one node at a time. The APIC GUI provides real-time status information about firmware upgrades.

Note If a recurring or one-time upgrade schedule is set with a date and time in the past, the scheduler triggers the upgrade immediately.

The following figure shows the APIC cluster nodes firmware upgrade process.

Figure 35: APIC Cluster Controller Firmware Upgrade Process



The APIC applies this controller firmware upgrade policy as follows:

- Because the administrator configured the controller update policy with a start time of midnight Saturday, the APIC begins the upgrade at midnight on Saturday.
- The system checks for compatibility of the existing firmware to upgrade to the new version according to the compatibility catalog provided with the new firmware image.

- The upgrade proceeds one node at a time until all nodes in the cluster are upgraded.

Note Because the APIC is a replicated cluster of nodes, disruption should be minimal. An administrator should be aware of the system load when considering scheduling APIC upgrades, and should plan for an upgrade during a maintenance window.

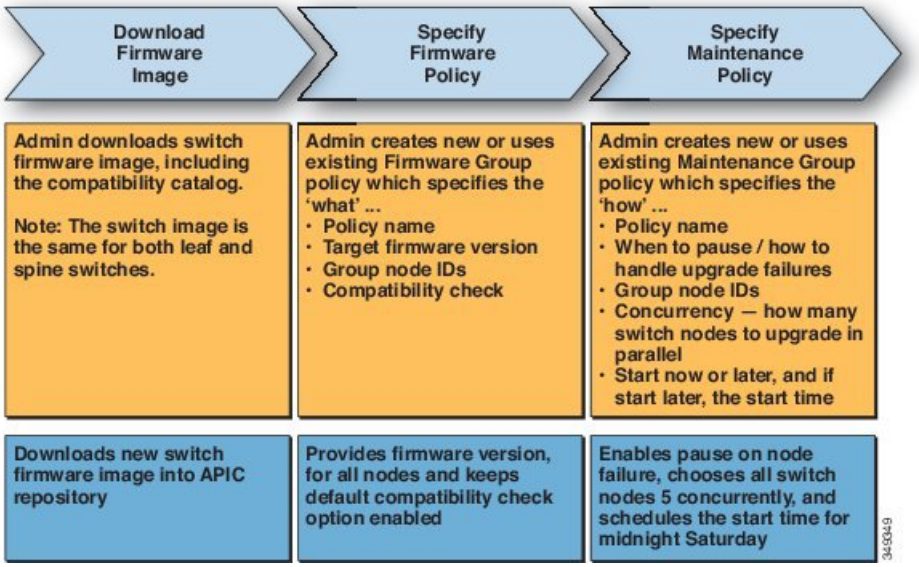
- The ACI fabric, including the APIC, continues to run while the upgrade proceeds.

Note The controllers upgrade in random order. Each APIC controller takes about 10 minutes to upgrade. Once a controller image is upgraded, it drops from the cluster, and it reboots with the newer version while the other APIC controllers in the cluster remain operational. Once the controller reboots, it joins the cluster again. Then the cluster converges, and the next controller image starts to upgrade. If the cluster does not immediately converge and is not fully fit, the upgrade will wait until the cluster converges and is fully fit. During this period, a Waiting for Cluster Convergence message is displayed.

- If a controller node upgrade fails, the upgrade pauses and waits for manual intervention.

The following figure shows how this process works for upgrading all the ACI fabric switch nodes firmware.

Figure 36: Switch Firmware Upgrade Process



The APIC applies this switch upgrade policy as follows:

- Because the administrator configured the controller update policy with a start time of midnight Saturday, the APIC begins the upgrade at midnight on Saturday.
- The system checks for compatibility of the existing firmware to upgrade to the new version according to the compatibility catalog provided with the new firmware image.

- The upgrade proceeds five nodes at a time until all the specified nodes are upgraded.

Note A firmware upgrade causes a switch reboot; the reboot can disrupt the operation of the switch for several minutes. Schedule firmware upgrades during a maintenance window.

- If a switch node fails to upgrade, the upgrade pauses and waits for manual intervention.

Refer to the *Cisco APIC Management, Installation, Upgrade, and Downgrade Guide* for detailed step-by-step instructions for performing firmware upgrades.

Configuration Zones

Configuration zones divide the ACI fabric into different zones that can be updated with configuration changes at different times. This limits the risk of deploying a faulty fabric-wide configuration that might disrupt traffic or even bring the fabric down. An administrator can deploy a configuration to a non-critical zone, and then deploy it to critical zones when satisfied that it is suitable.

The following policies specify configuration zone actions:

- `infrazone:ZoneP` is automatically created upon system upgrade. It cannot be deleted or modified.
- `infrazone:Zone` contains one or more pod groups (`PodGrp`) or one or more node groups (`NodeGrp`).

Note You can only choose `PodGrp` or `NodeGrp`; both cannot be chosen.

A node can be part of only one zone (`infrazone:Zone`). `NodeGrp` has two properties: name, and deployment mode. The deployment mode property can be:

- `enabled` - Pending updates are sent immediately.
- `disabled` - New updates are postponed.

Note

- Do not upgrade, downgrade, commission, or decommission nodes in a disabled configuration zone.
- Do not do a clean reload or an uplink/downlink port conversion reload of nodes in a disabled configuration zone.

- `triggered` - pending updates are sent immediately, and the deployment mode is automatically reset to the value it had before the change to `triggered`.

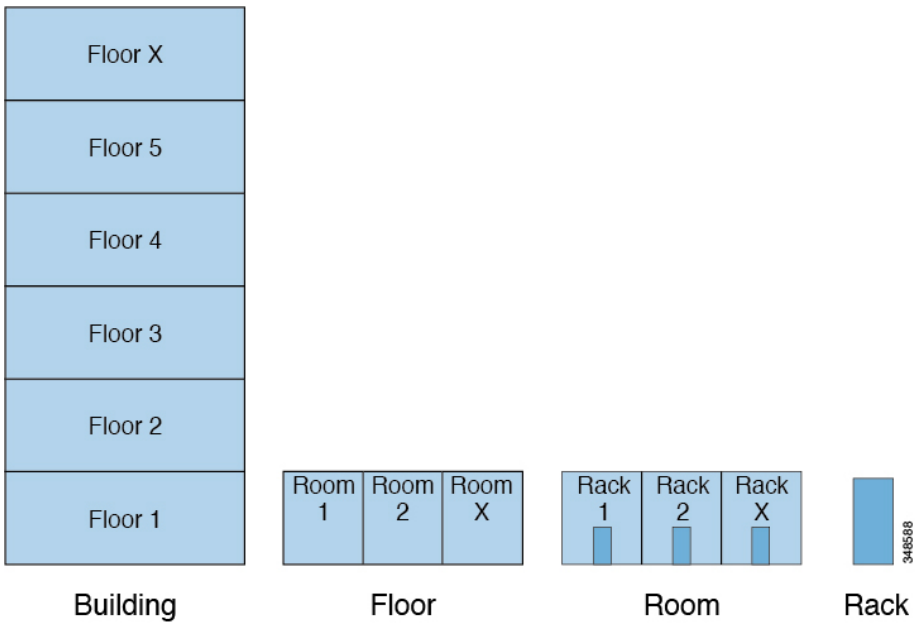
When a policy on a given set of nodes is created, modified, or deleted, updates are sent to each node where the policy is deployed. Based on policy class and `infrazone` configuration the following happens:.

- For policies that do not follow `infrazone` configuration, the APIC sends updates immediately to all the fabric nodes.
- For policies that follow `infrazone` configuration, the update proceeds according to the `infrazone` configuration:
  - If a node is part of an `infrazone:Zone`, the update is sent immediately if the deployment mode of the zone is set to enabled; otherwise the update is postponed.
  - If a node is not part of an`infrazone:Zone`, the update is done immediately, which is the ACI fabric default behavior.

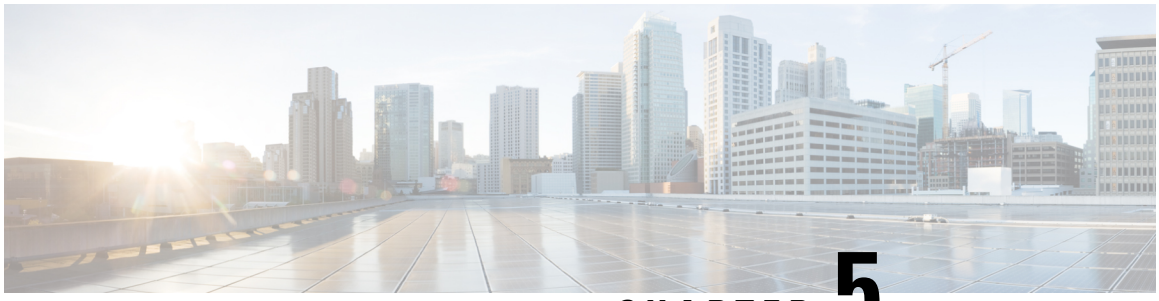
Geolocation

Administrators use geolocation policies to map the physical location of ACI fabric nodes in data center facilities. The following figure shows an example of the geolocation mapping feature.

Figure 37: Geolocation



For example, for fabric deployment in a single room, an administrator would use the default room object, and then create one or more racks to match the physical location of the switches. For a larger deployment, an administrator can create one or more site objects. Each site can contain one or more buildings. Each building has one or more floors. Each floor has one or more rooms, and each room has one or more racks. Finally each rack can be associated with one or more switches.



CHAPTER 5

Forwarding Within the ACI Fabric

This chapter contains the following sections:

About Forwarding Within the ACI Fabric

The ACI fabric supports more than 64,000 dedicated tenant networks. A single fabric can support more than one million IPv4/IPv6 endpoints, more than 64,000 tenants, and more than 200,000 10G ports. The ACI fabric enables any service (physical or virtual) anywhere with no need for additional software or hardware gateways to connect between the physical and virtual services and normalizes encapsulations for Virtual Extensible Local Area Network (VXLAN) / VLAN / Network Virtualization using Generic Routing Encapsulation (NVGRE).

The ACI fabric decouples the endpoint identity and associated policy from the underlying forwarding graph. It provides a distributed Layer 3 gateway that ensures optimal Layer 3 and Layer 2 forwarding. The fabric supports standard bridging and routing semantics without standard location constraints (any IP address anywhere), and removes flooding requirements for the IP control plane Address Resolution Protocol (ARP) / Gratuitous Address Resolution Protocol (GARP). All traffic within the fabric is encapsulated within VXLAN.

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.

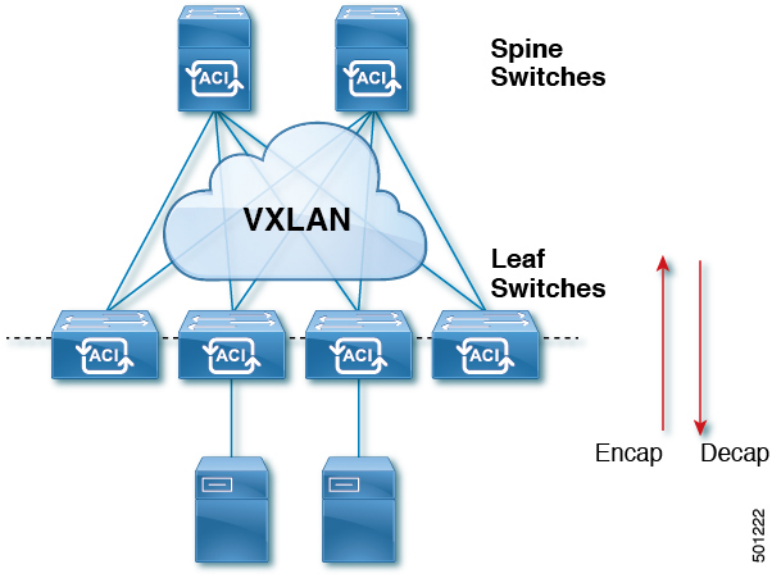
Today, application design drives east-west traffic from server to server through the data center access layer. Applications driving this shift include big data distributed processing designs like Hadoop, live virtual machine or workload migration as with VMware vMotion, server clustering, and multi-tier 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 38: 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

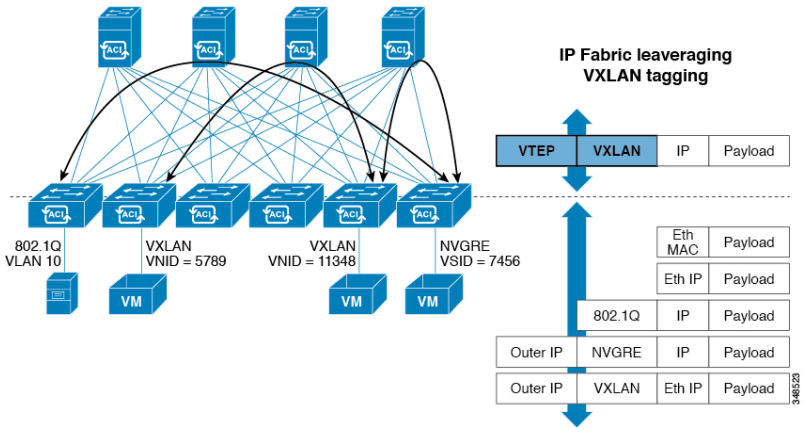
multipath (ECMP) forwarding in the fabric, and fast-reconverging. For propagating routing information between software defined networks within the fabric and routers external to the fabric, ACI uses the Multiprotocol Border Gateway Protocol (MP-BGP).

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.

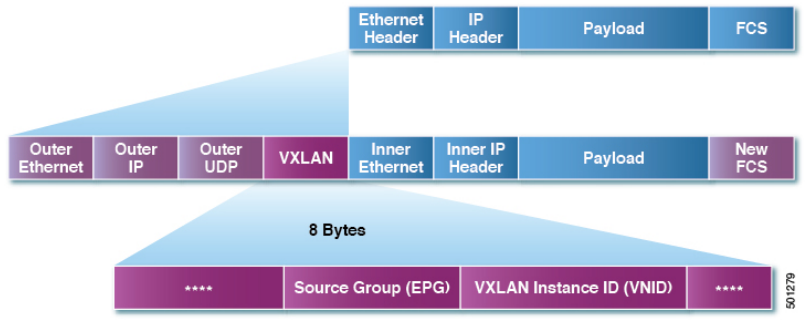
Figure 39: ACI Encapsulation Normalization



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 40: 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.

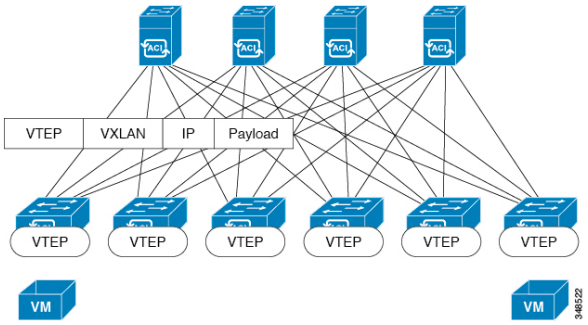
VXLAN enables ACI to deploy Layer 2 virtual networks at scale across the fabric underlay Layer 3 infrastructure. Application endpoint hosts can be flexibly placed in the data center network without concern for the Layer 3 boundary of the underlay infrastructure, while maintaining Layer 2 adjacency in 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 41: 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.

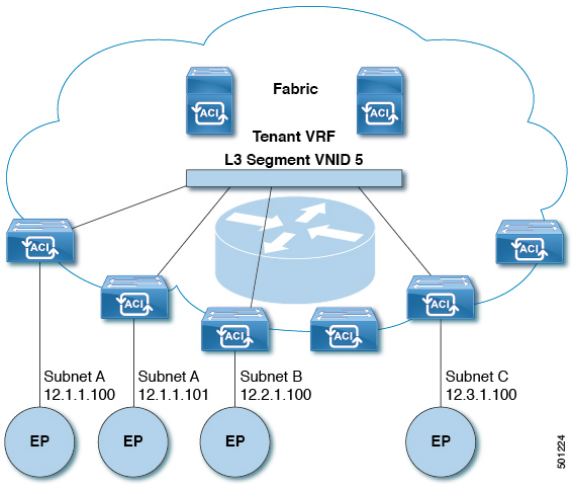
The VTEP in ACI maps the internal tenant MAC or IP address to a location using a distributed mapping database. After the VTEP completes a lookup, the VTEP sends the original data packet encapsulated in

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 42: 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.

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.

IGP Protocol Packets (EIGRP, OSPFv3) are constructed by components based on the Interface MTU size. In Cisco ACI, if the CPU MTU size is less than the Interface MTU size and if the constructed packet size is greater than the CPU MTU, then the packet is dropped by the kernal, especially in IPv6. To avoid such control packet drops always configure the same MTU values on both the control plane and on the interface.

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.

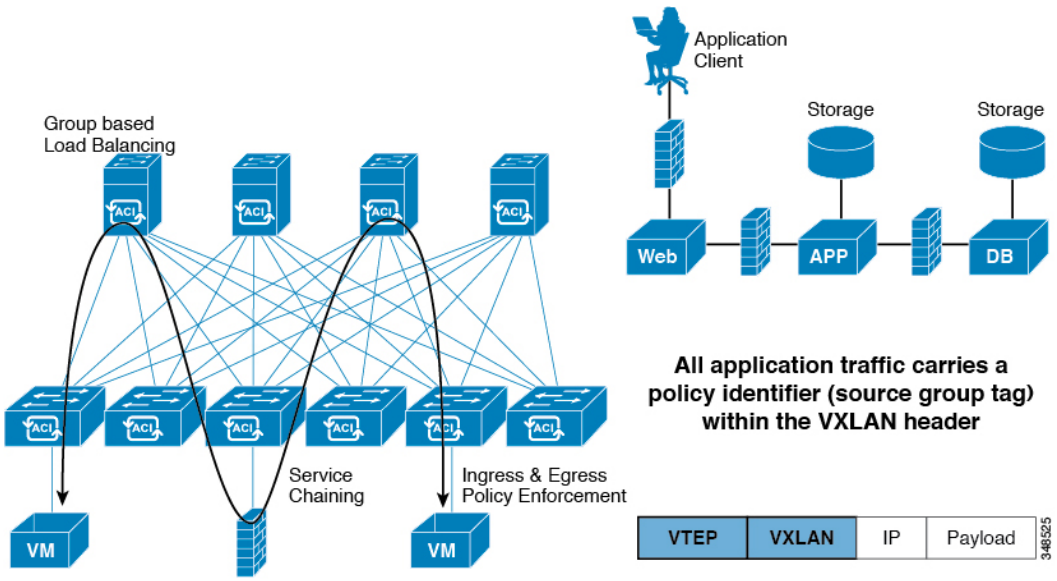
For the appropriate MTU values for each platform, see the relevant configuration guides.

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.

Policy Identification and Enforcement

An application policy is decoupled from forwarding by using a distinct tagging attribute that is carried in the VXLAN packet. Policy identification is carried in every packet in the ACI fabric, which enables consistent enforcement of the policy in a fully distributed manner. The following figure shows policy identification.

Figure 43: Policy Identification and Enforcement



Fabric and access policies govern the operation of internal fabric and external access interfaces. The system automatically creates default fabric and access policies. Fabric administrators (who have access rights to the entire fabric) can modify the default policies or create new policies according to their requirements. Fabric

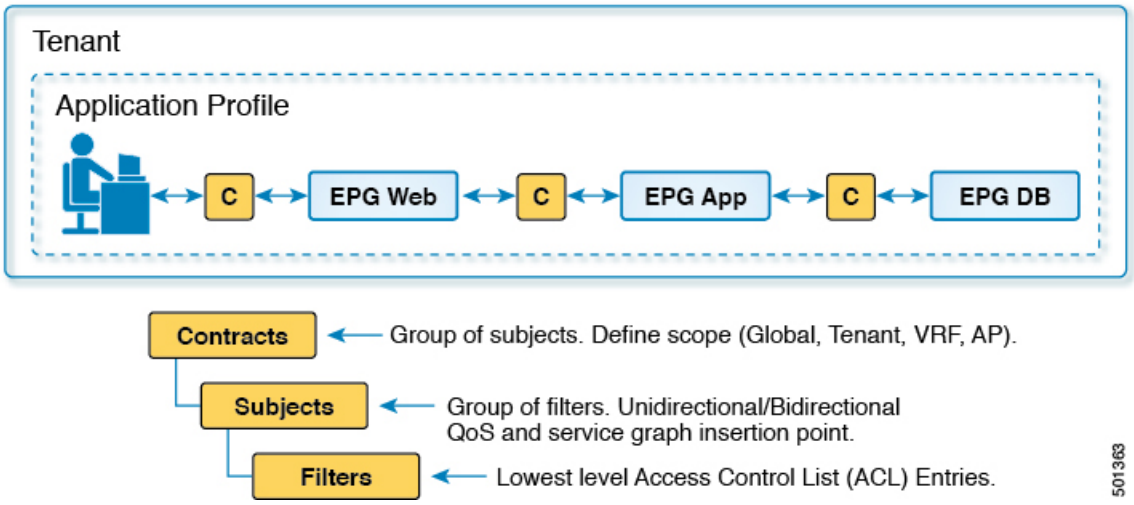
and access policies can enable various functions or protocols. Selectors in the APIC enable fabric administrators to choose the nodes and interfaces to which they will apply policies.

ACI Fabric Network Access Security Policy Model (Contracts)

The ACI fabric security policy model is based on contracts. This approach addresses limitations of traditional access control lists (ACLs). Contracts contain the specifications for security policies that are enforced on traffic between endpoint groups.

The following figure shows the components of a contract.

Figure 44: Contract Components



EPG communications require a contract; EPG to EPG communication is not allowed without a contract. The APIC renders the entire policy model, including contracts and their associated EPGs, into the concrete model in each switch. Upon ingress, every packet entering the fabric is marked with the required policy details. Because contracts are required to select what types of traffic can pass between EPGs, contracts enforce security policies. While contracts satisfy the security requirements handled by access control lists (ACLs) in conventional network settings, they are a more flexible, manageable, and comprehensive security policy solution.

Access Control List Limitations

Traditional access control lists (ACLs) have a number of limitations that the ACI fabric security model addresses. The traditional ACL is very tightly coupled with the network topology. They are typically configured per router or switch ingress and egress interface and are customized to that interface and the traffic that is expected to flow through those interfaces. Due to this customization, they often cannot be reused across interfaces, much less across routers or switches.

Traditional ACLs can be very complicated and cryptic because they contain lists of specific IP addresses, subnets, and protocols that are allowed as well as many that are specifically not allowed. This complexity means that they are difficult to maintain and often simply just grow as administrators are reluctant to remove any ACL rules for fear of creating a problem. Their complexity means that they are generally only deployed at specific demarcation points in the network such as the demarcation between the WAN and the enterprise or the WAN and the data center. In this case, the security benefits of ACLs are not exploited inside the enterprise or for traffic that is contained within the data center.

Another issue is the possible huge increase in the number of entries in a single ACL. Users often want to create an ACL that allows a set of sources to communicate with a set of destinations by using a set of protocols. In the worst case, if *N* sources are talking to *M* destinations using *K* protocols, there might be *N*M*K* lines in the ACL. The ACL must list each source that communicates with each destination for each protocol. It does not take many devices or protocols before the ACL gets very large.

The ACI fabric security model addresses these ACL issues. The ACI fabric security model directly expresses the intent of the administrator. Administrators use contract, filter, and label managed objects to specify how groups of endpoints are allowed to communicate. These managed objects are not tied to the topology of the network because they are not applied to a specific interface. They are simply rules that the network must enforce irrespective of where these groups of endpoints are connected. This topology independence means that these managed objects can easily be deployed and reused throughout the data center not just as specific demarcation points.

The ACI fabric security model uses the endpoint grouping construct directly so the idea of allowing groups of servers to communicate with one another is simple. A single rule can allow an arbitrary number of sources to communicate with an equally arbitrary number of destinations. This simplification dramatically improves their scale and maintainability which also means they are easier to use throughout the data center.

Contracts Contain Security Policy Specifications

In the ACI security model, contracts contain the policies that govern the communication between EPGs. The contract specifies what can be communicated and the EPGs specify the source and destination of the communications. Contracts link EPGs, as shown below.

EPG 1 ---------- CONTRACT ---------- EPG 2

Endpoints in EPG 1 can communicate with endpoints in EPG 2 and vice versa if the contract allows it. This policy construct is very flexible. There can be many contracts between EPG 1 and EPG 2, there can be more than two EPGs that use a contract, and contracts can be reused across multiple sets of EPGs, and more.

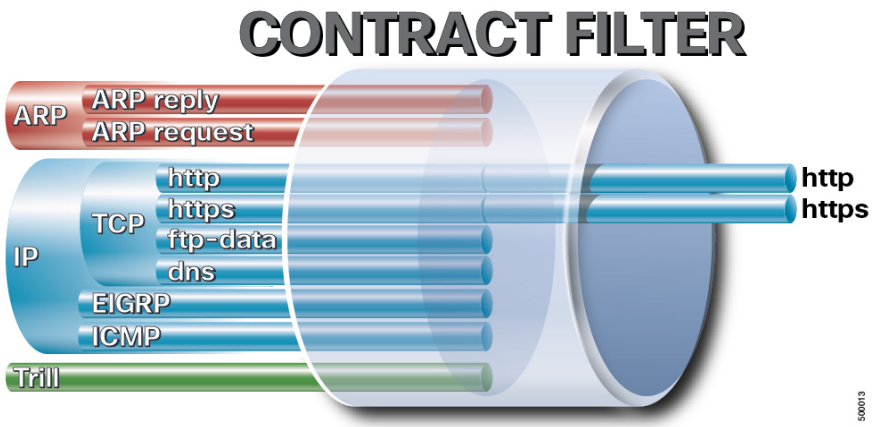
There is also directionality in the relationship between EPGs and contracts. EPGs can either provide or consume a contract. An EPG that provides a contract is typically a set of endpoints that provide a service to a set of client devices. The protocols used by that service are defined in the contract. An EPG that consumes a contract is typically a set of endpoints that are clients of that service. When the client endpoint (consumer) tries to connect to a server endpoint (provider), the contract checks to see if that connection is allowed. Unless otherwise specified, that contract would not allow a server to initiate a connection to a client. However, another contract between the EPGs could easily allow a connection in that direction.

This providing/consuming relationship is typically shown graphically with arrows between the EPGs and the contract. Note the direction of the arrows shown below.

EPG 1 <-------consumes-------- CONTRACT <-------provides-------- EPG 2

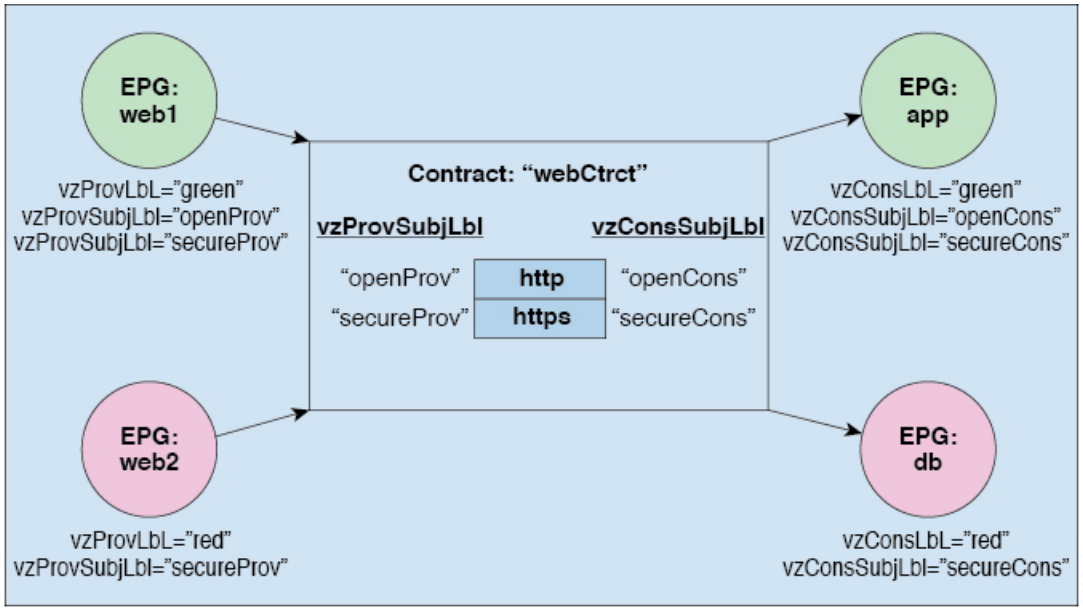
The contract is constructed in a hierarchical manner. It consists of one or more subjects, each subject contains one or more filters, and each filter can define one or more protocols.

Figure 45: Contract Filters



The following figure shows how contracts govern EPG communications.

Figure 46: Contracts Determine EPG to EPG Communications



For example, you may define a filter called HTTP that specifies TCP port 80 and port 8080 and another filter called HTTPS that specifies TCP port 443. You might then create a contract called webCtrct that has two sets of subjects. openProv and openCons are the subjects that contain the HTTP filter. secureProv and secureCons are the subjects that contain the HTTPS filter. This webCtrct contract can be used to allow both secure and non-secure web traffic between EPGs that provide the web service and EPGs that contain endpoints that want to consume that service.

These same constructs also apply for policies that govern virtual machine hypervisors. When an EPG is placed in a virtual machine manager (VMM) domain, the APIC downloads all of the policies that are associated with the EPG to the leaf switches with interfaces connecting to the VMM domain. For a full explanation of VMM domains, see the *Virtual Machine Manager Domains* chapter of *Application Centric Infrastructure Fundamentals*. When this policy is created, the APIC pushes it (pre-populates it) to a VMM domain that specifies which switches allow connectivity for the endpoints in the EPGs. The VMM domain defines the set

of switches and ports that allow endpoints in an EPG to connect to. When an endpoint comes on-line, it is associated with the appropriate EPGs. When it sends a packet, the source EPG and destination EPG are derived from the packet and the policy defined by the corresponding contract is checked to see if the packet is allowed. If yes, the packet is forwarded. If no, the packet is dropped.

Contracts consist of 1 or more subjects. Each subject contains 1 or more filters. Each filter contains 1 or more entries. Each entry is equivalent to a line in an Access Control List (ACL) that is applied on the Leaf switch to which the endpoint within the endpoint group is attached.

In detail, contracts are comprised of the following items:

- Name—All contracts that are consumed by a tenant must have different names (including contracts created under the common tenant or the tenant itself).
- Subjects—A group of filters for a specific application or service.
- Filters—Used to classify traffic based upon layer 2 to layer 4 attributes (such as Ethernet type, protocol type, TCP flags and ports).
- Actions—Action to be taken on the filtered traffic. The following actions are supported:
  - Permit the traffic (regular contracts, only)
  - Mark the traffic (DSCP/CoS) (regular contracts, only)
  - Redirect the traffic (regular contracts, only, through a service graph)
  - Copy the traffic (regular contracts, only, through a service graph or SPAN)
  - Block the traffic (taboo contracts)

    With Cisco APIC Release 3.2(x) and switches with names that end in EX or FX, you can alternatively use a subject Deny action or Contract or Subject Exception in a standard contract to block traffic with specified patterns.
  - Log the traffic (taboo contracts and regular contracts)
- Aliases—(Optional) A changeable name for an object. Although the name of an object, once created, cannot be changed, the Alias is a property that can be changed.

Thus, the contract allows more complex actions than just allow or deny. The contract can specify that traffic that matches a given subject can be re-directed to a service, can be copied, or can have its QoS level modified. With pre-population of the access policy in the concrete model, endpoints can move, new ones can come on-line, and communication can occur even if the APIC is off-line or otherwise inaccessible. The APIC is removed from being a single point of failure for the network. Upon packet ingress to the ACI fabric, security policies are enforced by the concrete model running in the switch.

Security Policy Enforcement

As traffic enters the leaf switch from the front panel interfaces, the packets are marked with the EPG of the source EPG. The leaf switch then performs a forwarding lookup on the packet destination IP address within the tenant space. A hit can result in any of the following scenarios:

1. A unicast (/32) hit provides the EPG of the destination endpoint and either the local interface or the remote leaf switch VTEP IP address where the destination endpoint is present.

2. A unicast hit of a subnet prefix (not /32) provides the EPG of the destination subnet prefix and either the local interface or the remote leaf switch VTEP IP address where the destination subnet prefix is present.
3. A multicast hit provides the local interfaces of local receivers and the outer destination IP address to use in the VXLAN encapsulation across the fabric and the EPG of the multicast group.

Note Multicast and external router subnets always result in a hit on the ingress leaf switch. Security policy enforcement occurs as soon as the destination EPG is known by the ingress leaf switch.

A miss result in the forwarding table causes the packet to be sent to the forwarding proxy in the spine switch. The forwarding proxy then performs a forwarding table lookup. If it is a miss, the packet is dropped. If it is a hit, the packet is sent to the egress leaf switch that contains the destination endpoint. Because the egress leaf switch knows the EPG of the destination, it performs the security policy enforcement. The egress leaf switch must also know the EPG of the packet source. The fabric header enables this process because it carries the EPG from the ingress leaf switch to the egress leaf switch. The spine switch preserves the original EPG in the packet when it performs the forwarding proxy function.

On the egress leaf switch, the source IP address, source VTEP, and source EPG information are stored in the local forwarding table through learning. Because most flows are bidirectional, a return packet populates the forwarding table on both sides of the flow, which enables the traffic to be ingress filtered in both directions.

Multicast and EPG Security

Multicast traffic introduces an interesting problem. With unicast traffic, the destination EPG is clearly known from examining the packet's destination. However, with multicast traffic, the destination is an abstract entity: the multicast group. Because the source of a packet is never a multicast address, the source EPG is determined in the same manner as in the previous unicast examples. The derivation of the destination group is where multicast differs.

Because multicast groups are somewhat independent of the network topology, static configuration of the (S, G) and (*, G) to group binding is acceptable. When the multicast group is placed in the forwarding table, the EPG that corresponds to the multicast group is also put in the forwarding table.

Note This document refers to multicast stream as a multicast group.

The leaf switch always views the group that corresponds to the multicast stream as the destination EPG and never the source EPG. In the access control matrix shown previously, the row contents are invalid where the multicast EPG is the source. The traffic is sent to the multicast stream from either the source of the multicast stream or the destination that wants to join the multicast stream. Because the multicast stream must be in the forwarding table and there is no hierarchical addressing within the stream, multicast traffic is access controlled at the ingress fabric edge. As a result, IPv4 multicast is always enforced as ingress filtering.

The receiver of the multicast stream must first join the multicast stream before it receives traffic. When sending the IGMP Join request, the multicast receiver is actually the source of the IGMP packet. The destination is defined as the multicast group and the destination EPG is retrieved from the forwarding table. At the ingress point where the router receives the IGMP Join request, access control is applied. If the Join request is denied, the receiver does not receive any traffic from that particular multicast stream.

The policy enforcement for multicast EPGs occurs on the ingress by the leaf switch according to contract rules as described earlier. Also, the multicast group to EPG binding is pushed by the APIC to all leaf switches that contain the particular tenant (VRF).

Multicast Tree Topology

The ACI fabric supports forwarding of unicast, multicast, and broadcast traffic from access ports. All multidestination traffic from the endpoint hosts is carried as multicast traffic in the fabric.

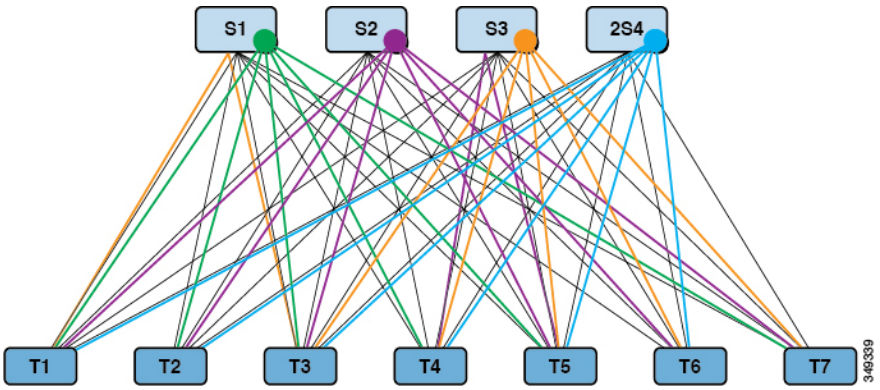
The ACI fabric consists of spine and leaf switches that are connected in a Clos topology (named after Charles Clos) where traffic that enters an ingress interface can be routed through any of the available middle stage spine switches, to the relevant egress switch. The leaf switches have two types of ports: fabric ports for connecting to spine switches and access ports for connecting servers, service appliances, routers, Fabric Extender (FEX), and so forth.

The leaf switches (also known as "top of rack" or "ToR" switches) are attached to the spine switches (also known as "end of row" or "EoR" switches). The leaf switches are not connected to each other and spine switches connect only to the leaf switches. In this Clos topology, every lower-tier switch is connected to each of the top-tier switches in a full-mesh topology. A spine switch failure only slightly degrades the performance through the ACI fabric. The data path is chosen so that the traffic load is evenly distributed between the spine switches.

The ACI fabric uses Forwarding Tag (FTAG) trees to load balance multi-destination traffic. All multi-destination traffic is forwarded in the form of encapsulated IP multicast traffic within the fabric. The ingress leaf assigns an FTAG to the traffic when forwarding it to the spine. The FTAG is assigned in the packet as part of the destination multicast address. In the fabric, the traffic is forwarded along the specified FTAG tree. Spine and any intermediate leaf switches forward traffic based on the FTAG ID. One forwarding tree is built per FTAG ID. Between any two nodes, only one link forwards per FTAG. Because of the use of multiple FTAGs, parallel links can be used with each FTAG choosing a different link for forwarding. The larger the number of FTAG trees in the fabric means the better the load balancing potential is. The ACI fabric supports up to 12 FTAGs.

The following figure shows a topology with four FTAGs. Every leaf switch in the fabric is connected to each FTAG either directly or through transit nodes. One FTAG is rooted on each of the spine nodes.

Figure 47: Multicast Tree Topology



If a leaf switch has direct connectivity to the spine, it uses the direct path to connect to the FTAG tree. If there is no direct link, the leaf switch uses transit nodes that are connected to the FTAG tree, as shown in the figure

above. Although the figure shows each spine as the root of one FTAG tree, multiple FTAG tree roots could be on one spine.

As part of the ACI Fabric bring-up discovery process, the FTAG roots are placed on the spine switches. The APIC configures each of the spine switches with the FTAGs that the spine anchors. The identity of the roots and the number of FTAGs is derived from the configuration. The APIC specifies the number of FTAG trees to be used and the roots for each of those trees. FTAG trees are recalculated every time there is a topology change in the fabric.

Root placement is configuration driven and is not re-rooted dynamically on run-time events such as a spine switch failure. Typically, FTAG configurations are static. An FTAG can be re-anchored from one spine to another when a spine switch is added or removed because the administrator might decide to redistribute the FTAG across the remaining or expanded set of spine switches.

About Traffic Storm Control

A traffic storm occurs when packets flood the LAN, creating excessive traffic and degrading network performance. You can use traffic storm control policies to prevent disruptions on Layer 2 ports by broadcast, unknown multicast, or unknown unicast traffic storms on physical interfaces.

By default, storm control is not enabled in the ACI fabric. ACI bridge domain (BD) Layer 2 unknown unicast flooding is enabled by default within the BD but can be disabled by an administrator. In that case, a storm control policy only applies to broadcast and unknown multicast traffic. If Layer 2 unknown unicast flooding is enabled in a BD, then a storm control policy applies to Layer 2 unknown unicast flooding in addition to broadcast and unknown multicast traffic.

Traffic storm control (also called traffic suppression) allows you to monitor the levels of incoming broadcast, multicast, and unknown unicast traffic over a one second interval. During this interval, the traffic level, which is expressed either as percentage of the total available bandwidth of the port or as the maximum packets per second allowed on the given port, is compared with the traffic storm control level that you configured. When the ingress traffic reaches the traffic storm control level that is configured on the port, traffic storm control drops the traffic until the interval ends. An administrator can configure a monitoring policy to raise a fault when a storm control threshold is exceeded.

Storm Control Guidelines and Limitations

Configure traffic storm control levels according to the following guidelines and limitations:

- Typically, a fabric administrator configures storm control in fabric access policies on the following interfaces:
  - A regular trunk interface.
  - A direct port channel on a single leaf switch.
  - A virtual port channel (a port channel on two leaf switches).
- Beginning with release 4.2(1), support is now available for triggering SNMP traps from Cisco Application Centric Infrastructure (ACI) when storm control thresholds are met, with the following restrictions:
  - There are two actions associated with storm control: drop and shutdown. With the shutdown action, interface traps will be raised, but the storm control traps to indicate that the storm is active or clear

is not determined by the shutdown action. Storm control traps with the shutdown action on the policy should therefore be ignored.

- If the ports flap with the storm control policy on, clear and active traps are seen together when the stats are collected. Clear and active traps are typically not seen together, but this is expected behavior in this case.

- For port channels and virtual port channels, the storm control values (packets per second or percentage) apply to all individual members of the port channel.

Note For switch hardware, beginning with Cisco Application Policy Infrastructure Controller (APIC) release 1.3(1) and switch release 11.3(1), for port channel configurations, the traffic suppression on the aggregated port may be up to two times the configured value. The new hardware ports are internally subdivided into these two groups: slice-0 and slice-1. To check the slicing map, use the `vsh_lc` command `show platform internal hal l2 port gpd` and look for `slice 0` or `slice 1` under the `Sl` column. If port channel members fall on both slice-0 and slice-1, allowed storm control traffic may become twice the configured value because the formula is calculated based on each slice.

- When configuring by percentage of available bandwidth, a value of 100 means no traffic storm control and a value of 0.01 suppresses all traffic.
- Due to hardware limitations and the method by which packets of different sizes are counted, the level percentage is an approximation. Depending on the sizes of the frames that make up the incoming traffic, the actual enforced level might differ from the configured level by several percentage points. Packets-per-second (PPS) values are converted to percentage based on 256 bytes.
- Maximum burst is the maximum accumulation of rate that is allowed when no traffic passes. When traffic starts, all the traffic up to the accumulated rate is allowed in the first interval. In subsequent intervals, traffic is allowed only up to the configured rate. The maximum supported is 65535 KB. If the configured rate exceeds this value, it is capped at this value for both PPS and percentage.
- The maximum burst that can be accumulated is 512 MB.
- On an egress leaf switch in optimized multicast flooding (OMF) mode, traffic storm control will not be applied.
- On an egress leaf switch in non-OMF mode, traffic storm control will be applied.
- On a leaf switch for FEX, traffic storm control is not available on host-facing interfaces.
- Traffic storm control unicast/multicast differentiation is not supported on Cisco Nexus C93128TX, C9396PX, C9396TX, C93120TX, C9332PQ, C9372PX, C9372TX, C9372PX-E, or C9372TX-E switches.
- SNMP traps for traffic storm control are not supported on Cisco Nexus C93128TX, C9396PX, C9396TX, C93120TX, C9332PQ, C9372PX, C9372TX, C9372PX-E, C9372TX-E switches.
- Traffic storm control traps is not supported on Cisco Nexus C93128TX, C9396PX, C9396TX, C93120TX, C9332PQ, C9372PX, C9372TX, C9372PX-E, or C9372TX-E switches.
- Storm Control Action is supported only on physical Ethernet interfaces and port channel interfaces.

Beginning with release 4.1(1), Storm Control **Shutdown** option is supported. When the **shutdown** action is selected for an interface with the default Soak Instance Count, the packets exceeding the threshold are dropped for 3 seconds and the port is shutdown on the 3rd second. The default action is **Drop**. When **Shutdown** action is selected, the user has the option to specify the soaking interval. The default soaking interval is 3 seconds. The configurable range is from 3 to 10 seconds.

- If the data plane policing (DPP) policer that is configured for the interface has a value that is lower than storm policer's value, the DPP policer will take the precedence. The lower value that is configured between the DPP policer and storm policer is honored on the configured interface.
- Beginning with release 4.2(6), the storm policer is enforced for all forwarded control traffic in the leaf switch for the DHCP, ARP, ND, HSRP, PIM, IGMP, and EIGRP protocols regardless of whether the bridge domain is configured for **Flood in BD** or **Flood in Encapsulation**. This behavior change applies only to EX and later leaf switches.
  - With EX switches, you can configure both the supervisor policer and storm policer for one of the protocols. In this case, if a server sends traffic at a rate higher than the configured supervisor policer rate (Control Plane Policing, CoPP), then the storm policer will allow more traffic than what is configured as the storm policer rate. If the incoming traffic rate is equal to or less than supervisor policer rate, then the storm policer will correctly allow the configured storm traffic rate. This behavior is applicable irrespective of the configured supervisor policer and storm policer rates.
  - One side effect of the storm policer now being enforced for all forwarded control traffic in the leaf switch for the specified protocols is that control traffic that gets forwarded in the leaf switch will now get subjected to storm policer drops. In previous releases, no such storm policer drops occur for the protocols that are affected by this behavior change.
- Traffic storm control cannot police multicast traffic in a bridge domain or VRF instance that has PIM enabled.
- When the storm control policer is applied on a port channel interface, the allowed rate may be more than the configured rate. If the member links of the port channel span across multiple slices, then the allowed traffic rate will be equal to the configured rate multiplied by the number of slices across which the member links span.

  The port-to-slice mapping depends on the switch model.

  As an example, assume that there is a port channel that has member links port1, port2, and port3 with a storm policer rate of 10Mbps.
  - If port1, port2, and port3 belong to slice1, then traffic is policed to 10Mbps.
  - If port1 and port2 belong to slice1 and port3 belongs to slice2, then traffic is policed to 20Mbps.
  - If port1 belongs to slice1, port2 belongs to slice2, and port3 belongs to slice3, then traffic is policed to 30Mbps.

Fabric Load Balancing

The ACI fabric provides several load balancing options for balancing the traffic among the available uplink links. This topic describes load balancing for leaf to spine switch traffic.

Static hash load balancing is the traditional load balancing mechanism used in networks where each flow is allocated to an uplink based on a hash of its 5-tuple. This load balancing gives a distribution of flows across the available links that is roughly even. Usually, with a large number of flows, the even distribution of flows results in an even distribution of bandwidth as well. However, if a few flows are much larger than the rest, static load balancing might give suboptimal results.

ACI fabric Dynamic Load Balancing (DLB) adjusts the traffic allocations according to congestion levels. It measures the congestion across the available paths and places the flows on the least congested paths, which results in an optimal or near optimal placement of the data.

DLB can be configured to place traffic on the available uplinks using the granularity of flows or flowlets. Flowlets are bursts of packets from a flow that are separated by suitably large gaps in time. If the idle interval between two bursts of packets is larger than the maximum difference in latency among available paths, the second burst (or flowlet) can be sent along a different path than the first without reordering packets. This idle interval is measured with a timer called the flowlet timer. Flowlets provide a higher granular alternative to flows for load balancing without causing packet reordering.

DLB modes of operation are aggressive or conservative. These modes pertain to the timeout value used for the flowlet timer. The aggressive mode flowlet timeout is a relatively small value. This very fine-grained load balancing is optimal for the distribution of traffic, but some packet reordering might occur. However, the overall benefit to application performance is equal to or better than the conservative mode. The conservative mode flowlet timeout is a larger value that guarantees packets are not to be re-ordered. The tradeoff is less granular load balancing because new flowlet opportunities are less frequent. While DLB is not always able to provide the most optimal load balancing, it is never worse than static hash load balancing.

Note Although all Nexus 9000 Series switches have hardware support for DLB, the DLB feature is not enabled in the current software releases for second generation platforms (switches with EX, FX, and FX2 suffixes).

The ACI fabric adjusts traffic when the number of available links changes due to a link going off-line or coming on-line. The fabric redistributes the traffic across the new set of links.

In all modes of load balancing, static or dynamic, the traffic is sent only on those uplinks or paths that meet the criteria for equal cost multipath (ECMP); these paths are equal and the lowest cost from a routing perspective.

Dynamic Packet Prioritization (DPP), while not a load balancing technology, uses some of the same mechanisms as DLB in the switch. DPP configuration is exclusive of DLB. DPP prioritizes short flows higher than long flows; a short flow is less than approximately 15 packets. Because short flows are more sensitive to latency than long ones, DPP can improve overall application performance.

For intra-leaf switch traffic, all DPP-prioritized traffic is marked CoS 0 regardless of a custom QoS configuration. For inter-leaf switch traffic, all DPP-prioritized traffic is marked CoS 3 regardless of a custom QoS configuration.

GPRS tunneling protocol (GTP) is used mainly to deliver data on wireless networks. Cisco Nexus switches are places in Telcom Datacenters. When packets are being sent through Cisco Nexus 9000 switches in a datacenter, traffic needs to be load-balanced based on the GTP header. When the fabric is connected with an external router through link bundling, the traffic is required to be distributed evenly between all bundle members (For example, Layer 2 port channel, Layer 3 ECMP links, Layer 3 port channel, and L3Out on the port channel). GTP traffic load balancing is performed within the fabric as well.

To achieve GTP load balancing, Cisco Nexus 9000 Series switches use 5-tuple load balancing mechanism. The load balancing mechanism takes into account the source IP, destination IP, protocol, Layer 4 resource

and destination port (if traffic is TCP or UDP) fields from the packet. In the case of GTP traffic, a limited number of unique values for these fields restrict the equal distribution of traffic load on the tunnel.

To avoid polarization for GTP traffic in load balancing, a tunnel endpoint identifier (TEID) in the GTP header is used instead of a UDP port number. Because the TEID is unique per tunnel, traffic can be evenly load balanced across multiple links in the bundle.

The GTP load balancing feature overrides the source and destination port information with the 32-bit TEID value that is present in GTPU packets.

GTP tunnel load balancing feature adds support for:

- GTP with IPv4/IPv6 transport header on physical interface
- GTPU with UDP port 2152

The ACI fabric default configuration uses a traditional static hash. A static hashing function distributes the traffic between uplinks from the leaf switch to the spine switch. When a link goes down or comes up, traffic on all links is redistributed based on the new number of uplinks.

Leaf/Spine Switch Dynamic Load Balancing Algorithms

The following table provides the default non-configurable algorithms used in leaf/spine switch dynamic load balancing:

Table 4: ACI Leaf/Spine Switch Dynamic Load Balancing

Traffic Type	Hashing Data Points
Leaf/Spine IP unicast	• Source MAC address • Destination MAC address • Source IP address • Destination IP address • Protocol type • Source Layer 4 port • Destination Layer 4 port • Segment ID (VXLAN VNID) or VLAN ID
Leaf/Spine Layer 2	• Source MAC address • Destination MAC address • Segment ID (VXLAN VNID) or VLAN ID

Endpoint Retention

Retaining cached endpoint MAC and IP addresses in the switch improves performance. The switch learns about endpoints as they become active. Local endpoints are on the local switch. Remote endpoints are on

other switches but are cached locally. The leaf switches store location and policy information about endpoints that are attached directly to them (or through a directly attached Layer 2 switch or Fabric Extender), local endpoints, and endpoints that are attached to other leaf switches on the fabric (remote endpoints in the hardware). The switch uses a 32-Kb entry cache for local endpoints and a 64-Kb entry cache for remote endpoints.

Software that runs on the leaf switch actively manages these tables. For the locally attached endpoints, the software ages out entries after a retention timer for each entry has expired. Endpoint entries are pruned from the switch cache as the endpoint activity ceases, the endpoint location moves to another switch, or the life cycle state changes to offline. The default value for the local retention timer is 15 minutes. Before removing an inactive entry, the leaf switch sends three ARP requests to the endpoint to see if it really has gone away. If the switch receives no ARP response, the entry is pruned. For remotely attached endpoints, the switch ages out the entries after five minutes of inactivity. The remote endpoint is immediately reentered in the table if it becomes active again.

Note Version 1.3(1g) adds silent host tracking that will be triggered for any virtual and local hosts.

There is no performance penalty for not having the remote endpoint in the table other than policies are enforced at the remote leaf switch until the endpoint is cached again.

When subnets of a bridge domain are configured to be *enforced*, the endpoint retention policy operates in the following way:

- New endpoints with IP addresses not contained in the subnets of the bridge domain are not learned.
- Already learned endpoints age out of the endpoint retention cache if the device does not respond for tracking.

This enforcement process operates in the same way regardless of whether the subnet is defined under a bridge domain or if the subnet is defined under and EPG.

The endpoint retention timer policy can be modified. Configuring a static endpoint MAC and IP address enables permanently storing it in the switch cache by setting its retention timer to zero. Setting the retention timer to zero for an entry means that it will not be removed automatically. Care must be taken when doing so. If the endpoint moves or its policy changes, the entry must be refreshed manually with the updated information through the APIC. When the retention timer is nonzero, this information is checked and updated instantly on each packet without APIC intervention.

The endpoint retention policy determines how pruning is done. Use the default policy algorithm for most operations. Changing the endpoint retention policy can affect system performance. In the case of a switch that communicates with thousands of endpoints, lowering the aging interval increases the number of cache windows available to support large numbers of active endpoints. When the endpoint count exceeds 10,000, we recommend distributing endpoints across multiple switches.

Observe the following guidelines regarding changing the default endpoint retention policy:

- Remote Bounce Interval = (Remote Age * 2) + 30 seconds
    - Recommended default values:
        - Local Age = 900 seconds
        - Remote Age = 300 seconds
        - Bounce Age = 630 seconds

- Upgrade considerations: When upgrading to any ACI version older than release 1.0(1k), assure that the default values of endpoint retention policy (`epRetPol`) under tenant common are as follows: Bounce Age = 660 seconds.

IP Endpoint Learning Behavior

When an ACI bridge domain is configured with unicast routing enabled, not only does it learn MAC addresses, but it also learns IP addresses associated with the MAC addresses.

ACI tracks and requires MAC addresses to be unique per bridge domain. In ACI, endpoints are based on a single MAC address, but any number of IP addresses can be tied to a single MAC address in a bridge domain. ACI links these IP addresses to a MAC address. It is possible that a MAC address represents an endpoint that only has an IP address.

Therefore ACI may learn and store local endpoints as follows:

- Only a MAC address
- MAC address with a single IP address
- MAC address with multiple IP addresses

The third case occurs if a server has multiple IP addresses on the same MAC address, such as primary and secondary IP addresses. It could also occur if the ACI fabric learns a server's MAC and IP addresses on the fabric, but the server's IP address is subsequently changed. When this occurs, ACI stores and links the MAC address with both the old and new IP addresses. The old IP address is not removed until the ACI fabric flushes the endpoint with the base MAC address.

There are two primary types of local endpoint moves in ACI:

- Where the MAC address moves to a different interface
- Where the IP address moves to a different MAC address

When the MAC address moves to a different interface, all IP addresses linked to the MAC address in the bridge domain move with it. The ACI fabric also tracks moves, when only the IP address moves (and receives a new MAC address). This might occur, for example, if a virtual server's MAC address is changed and it is moved to a new ESXI server (port).

If an IP address is seen to exist across multiple MAC addresses within a VRF, this indicates that an IP flap has occurred (which can be detrimental to fabric forwarding decisions). This is similar to MAC flapping on two separate interfaces in a legacy network or MAC flaps on a bridge domain.

One scenario that can produce IP flaps is when a server Network Information Card (NIC) pair is set to active/active, but the two are not connected in a single logical link (such as a Port-Channel or Virtual Port-Channel). This type of setup can cause a single IP address, for example a virtual machine’s IP address, to constantly move between two MAC addresses in the fabric.

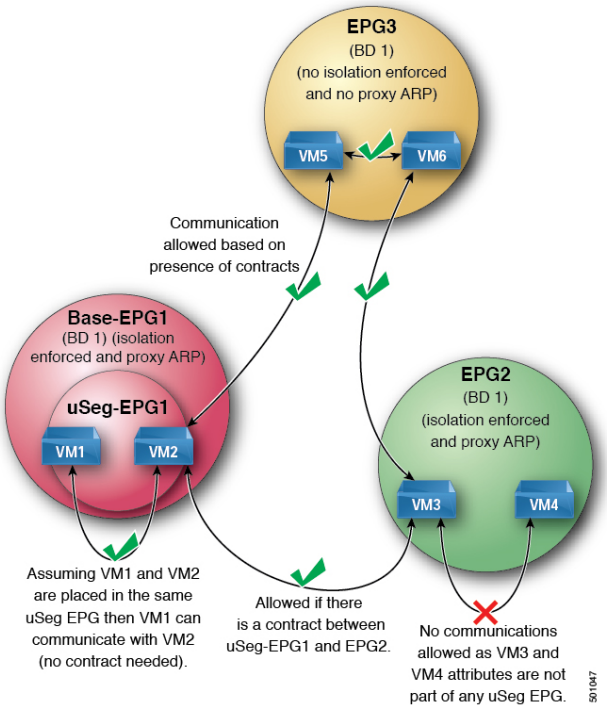
To address this type of behavior, we recommend configuring the NIC pair as the two legs of a VPC to achieve an Active/Active setup. If the server hardware does not support the Active/Active configuration (for example a blade chassis), then an active/standby type of NIC pair configuration will also prevent the IP flapping from occurring.

About Proxy ARP

Proxy ARP in Cisco ACI enables endpoints within a network or subnet to communicate with other endpoints without knowing the real MAC address of the endpoints. Proxy ARP is aware of the location of the traffic destination, and offers its own MAC address as the final destination instead.

To enable Proxy ARP, intra-EPG endpoint isolation must be enabled on the EPG see the following figure for details. For more information about intra-EPG isolation and Cisco ACI, see the *Cisco ACI Virtualization Guide*.

Figure 48: Proxy ARP and Cisco APIC



Proxy ARP within the Cisco ACI fabric is different from the traditional proxy ARP. As an example of the communication process, when proxy ARP is enabled on an EPG, if an endpoint A sends an ARP request for endpoint B and if endpoint B is learned within the fabric, then endpoint A will receive a proxy ARP response from the bridge domain (BD) MAC. If endpoint A sends an ARP request for endpoint B, and if endpoint B is not learned within the ACI fabric already, then the fabric will send a proxy ARP request within the BD. Endpoint B will respond to this proxy ARP request back to the fabric. At this point, the fabric does not send a proxy ARP response to endpoint A, but endpoint B is learned within the fabric. If endpoint A sends another ARP request to endpoint B, then the fabric will send a proxy ARP response from the BD MAC.

The following example describes the proxy ARP resolution steps for communication between clients VM1 and VM2:

1. VM1 to VM2 communication is desired.

Figure 49: VM1 to VM2 Communication is Desired.

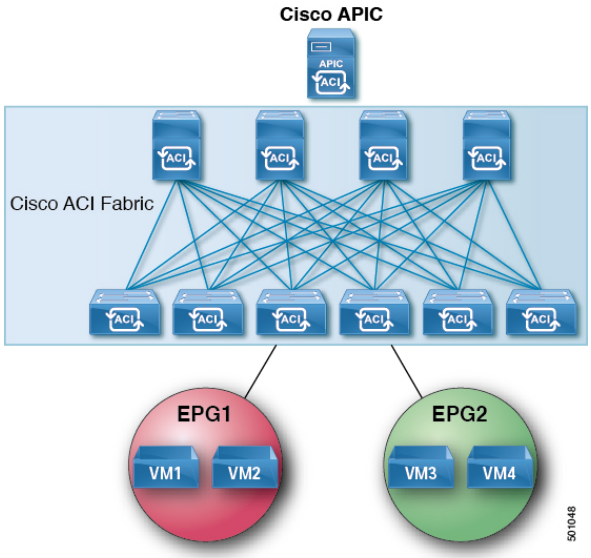


Table 5: ARP Table State

Device	State
VM1	IP = * MAC = *
ACI fabric	IP = * MAC = *
VM2	IP = * MAC = *

2. VM1 sends an ARP request with a broadcast MAC address to VM2.

Figure 50: VM1 sends an ARP Request with a Broadcast MAC address to VM2

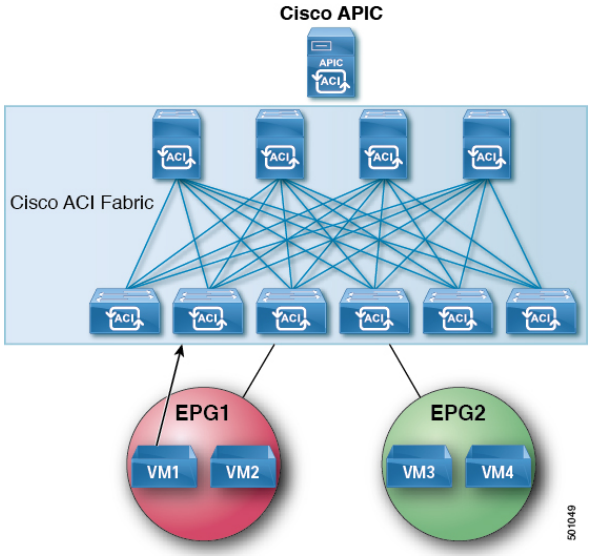


Table 6: ARP Table State

Device	State
VM1	IP = VM2 IP; MAC = ?
ACI fabric	IP = VM1 IP; MAC = VM1 MAC
VM2	IP = * MAC = *

3. The ACI fabric floods the proxy ARP request within the bridge domain (BD).

Figure 51: ACI Fabric Floods the Proxy ARP Request within the BD

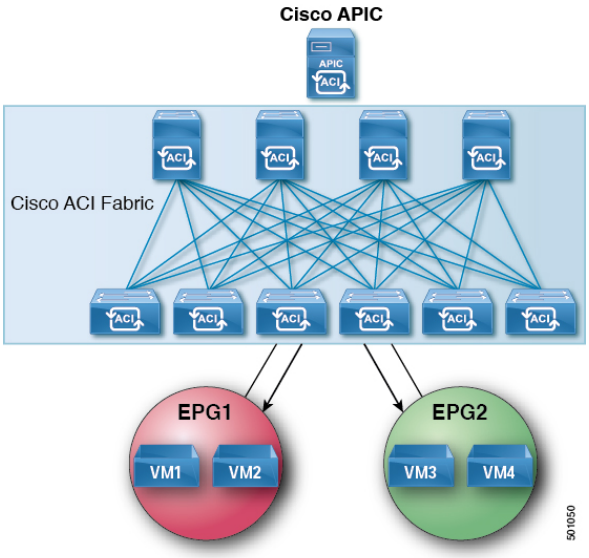


Table 7: ARP Table State

Device	State
VM1	IP = VM2 IP; MAC = ?
ACI fabric	IP = VM1 IP; MAC = VM1 MAC
VM2	IP = VM1 IP; MAC = BD MAC

4. VM2 sends an ARP response to the ACI fabric.

Figure 52: VM2 Sends an ARP Response to the ACI Fabric

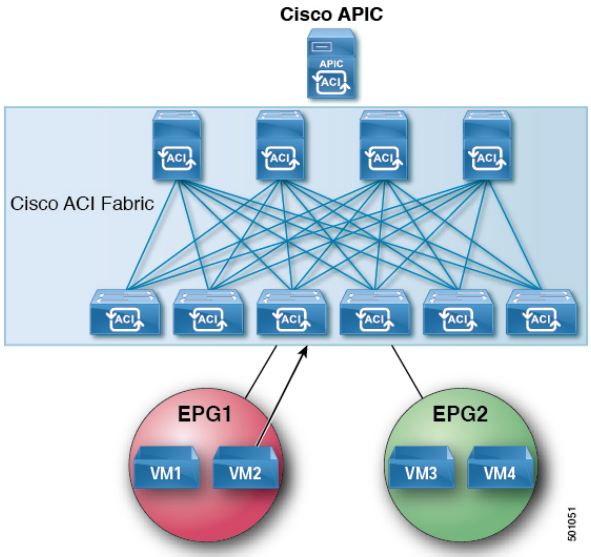


Table 8: ARP Table State

Device	State
VM1	IP = VM2 IP; MAC = ?
ACI fabric	IP = VM1 IP; MAC = VM1 MAC
VM2	IP = VM1 IP; MAC = BD MAC

5. VM2 is learned.

Figure 53: VM2 is Learned

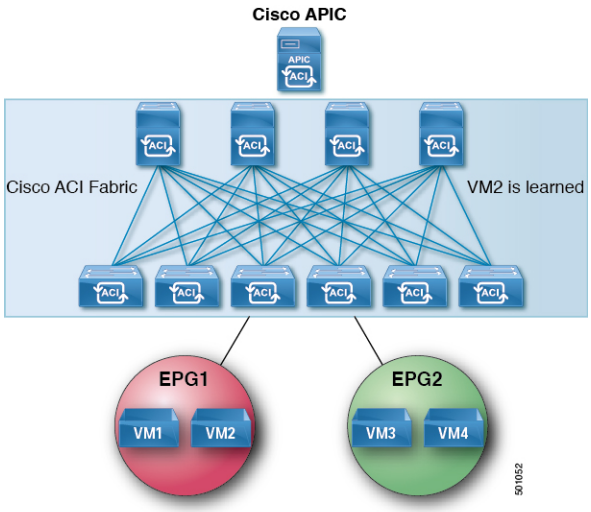


Table 9: ARP Table State

Device	State
VM1	IP = VM2 IP; MAC = ?
ACI fabric	IP = VM1 IP; MAC = VM1 MAC IP = VM2 IP; MAC = VM2 MAC
VM2	IP = VM1 IP; MAC = BD MAC

6. VM1 sends an ARP request with a broadcast MAC address to VM2.

Figure 54: VM1 Sends an ARP Request with a Broadcast MAC Address to VM2

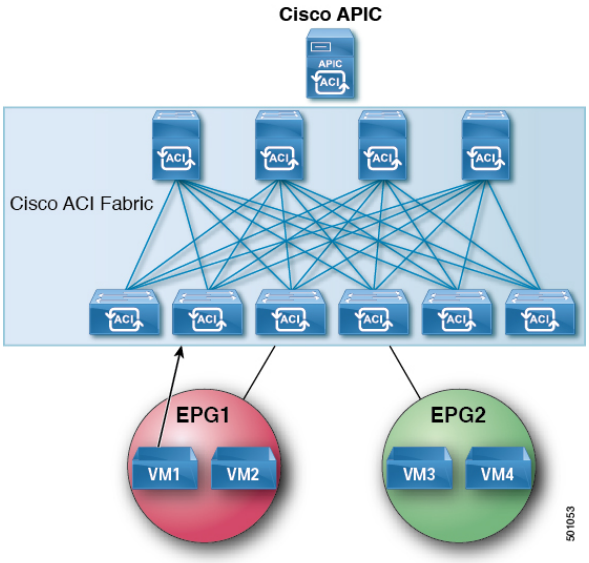


Table 10: ARP Table State

Device	State
VM1	IP = VM2 IP MAC = ?
ACI fabric	IP = VM1 IP; MAC = VM1 MAC IP = VM2 IP; MAC = VM2 MAC
VM2	IP = VM1 IP; MAC = BD MAC

7. The ACI fabric sends a proxy ARP response to VM1.

Figure 55: ACI Fabric Sends a Proxy ARP Response to VM1

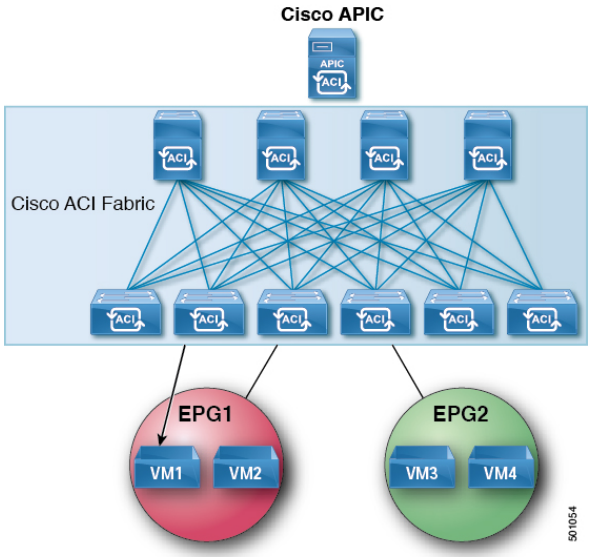


Table 11: ARP Table State

Device	State
VM1	IP = VM2 IP; MAC = BD MAC
ACI fabric	IP = VM1 IP; MAC = VM1 MAC IP = VM2 IP; MAC = VM2 MAC
VM2	IP = VM1 IP; MAC = BD MAC

Loop Detection

The ACI fabric provides global default loop detection policies that can detect loops in Layer 2 network segments which are connected to ACI access ports. These global policies are disabled by default but the port level policies are enabled by default. Enabling the global policies means they are enabled on all access ports, virtual ports, and virtual port channels unless they are disabled at the individual port level.

The ACI fabric does not participate in the Spanning Tree Protocol (STP). Instead, it implements the mis-cabling protocol (MCP) to detect loops. MCP works in a complementary manner with STP that is running on external Layer 2 networks, and handles bridge protocol data unit (BPDU) packets that access ports receive.

Note Interfaces from an external switch running spanning tree and connected to ACI fabric with a VPC can go to loop_inc status. Flapping the port-channel from the external switch resolves the problem. Enabling BDPU filter or disabling loopguard on the external switch will prevent the issue.

A fabric administrator provides a key that MCP uses to identify which MCP packets are initiated by the ACI fabric. The administrator can choose how the MCP policies identify loops and how to act upon the loops: syslog only, or disable the port.

While endpoint moves such as VM moves are normal, they can be symptomatic of loops if the frequency is high, and the interval between moves is brief. A separate global default endpoint move loop detection policy is available but is disabled by default. An administrator can choose how to act upon move detection loops.

Also, an error disabled recovery policy can enable ports that loop detection and BPDU policies disabled after an interval that the administrator can configure.

The MCP runs in native VLAN mode where the MCP BPDUs sent are not VLAN tagged, by default. MCP can detect loops due to mis-cabling if the packets sent in native VLAN are received by the fabric, but if there is a loop in non-native VLANs in EPG VLANs then it is not detected. Starting with release 2.0(2), APIC supports sending MCP BPDUs in all VLANs in the EPGs configured therefore any loops in those VLANs are detected. A new MCP configuration mode allows you to configure MCP to operate in a mode where MCP PDUs are sent in all EPG VLANs that a physical port belongs to by adding 802.1Q header with each of the EPG VLAN id to the PDUs transmitted.

Starting 3.2.1 release, the ACI fabric provides faster loop detection with transmit frequencies from 100 millisecond to 300 seconds.

Note Per-VLAN MCP will only run on 256 VLANs per interface. If there are more than 256 VLANs, then the first numerical 256 VLANs are chosen.

MCP is not supported on fabrix extender (FEX) host interface (HIF) ports.

Rogue Endpoint Detection

About the Rogue Endpoint Control Policy

A rogue endpoint attacks leaf switches through frequently, repeatedly injecting packets on different leaf switch ports and changing 802.1Q tags (thus, emulating endpoint moves) causing learned class and EPG port changes. Misconfigurations can also cause frequent IP and MAC address changes (moves).

Such rapid movement in the fabric causes significant network instability, high CPU usage, and in rare instances, endpoint mapper (EPM) and EPM client (EPMC) crashes due to significant and prolonged messaging and transaction service (MTS) buffer consumption. Also, such frequent moves may result in the EPM and EPMC logs rolling over very quickly, hampering debugging for unrelated endpoints.

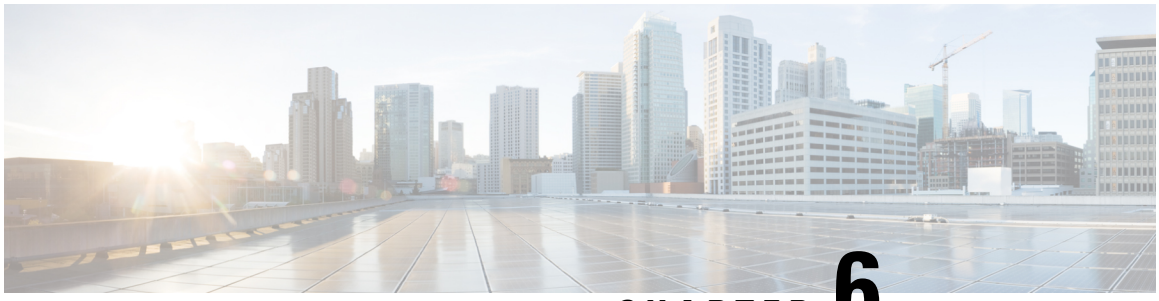
The rogue endpoint control feature addresses this vulnerability by quickly:

- Identifying such rapidly moving MAC and IP endpoints.
- Stopping the movement by temporarily making endpoints static, thus quarantining the endpoint.
- Prior to 3.2(6) release: Keeping the endpoint static for the **Rogue EP Detection Interval** and dropping the traffic to and from the rogue endpoint. After this time expires, deleting the unauthorized MAC or IP address.

- In the 3.2(6) release and later: Keeping the endpoint static for the **Rogue EP Detection Interval** (this feature no longer drops the traffic). After this time expires, deleting the unauthorized MAC or IP address.
- Generating a host tracking packet to enable the system to re-learn the impacted MAC or IP address.
- Raising a fault to enable corrective action.

The rogue endpoint control policy is configured globally and, unlike other loop prevention methods, functions at the level of individual endpoints (IP and MAC addresses). It does not distinguish between local or remote moves; any type of interface change is considered a move in determining if an endpoint should be quarantined.

The rogue endpoint control feature is disabled by default.



CHAPTER 6

Networking and Management Connectivity

This chapter contains the following sections:

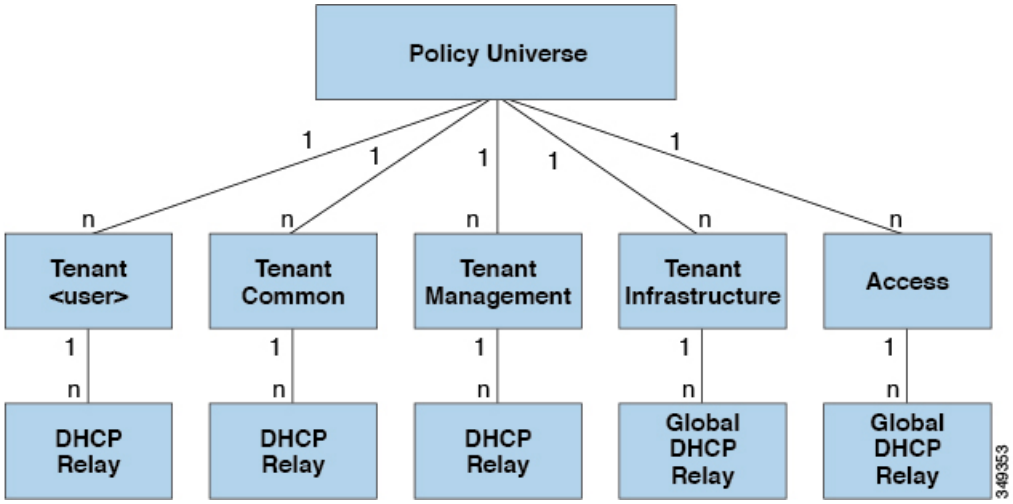
DHCP Relay

Although ACI fabric-wide flooding is disabled by default, flooding within a bridge domain is enabled by default. Because flooding within a bridge domain is enabled by default, clients can connect to DHCP servers within the same EPG. However, when the DHCP server is in a different EPG or Virtual Routing and Forwarding (VRF) instance than the clients, DHCP Relay is required. Also, when Layer 2 flooding is disabled, DHCP Relay is required.

Note When the ACI fabric acts as a DHCP relay, it inserts the DHCP Option 82 (the DHCP Relay Agent Information Option) in DHCP requests that it proxies on behalf of clients. If a response (DHCP offer) comes back from a DHCP server without Option 82, it is silently dropped by the fabric. When ACI acts as a DHCP relay, DHCP servers providing IP addresses to compute nodes attached to the ACI fabric must support Option 82. Windows 2003 and 2008 do not support option 82 but Windows 2012 does.

The figure below shows the managed objects in the management information tree (MIT) that can contain DHCP relays: user tenants, the `common` tenant, the `infra` tenant, the `mgmt` tenant, and fabric access.

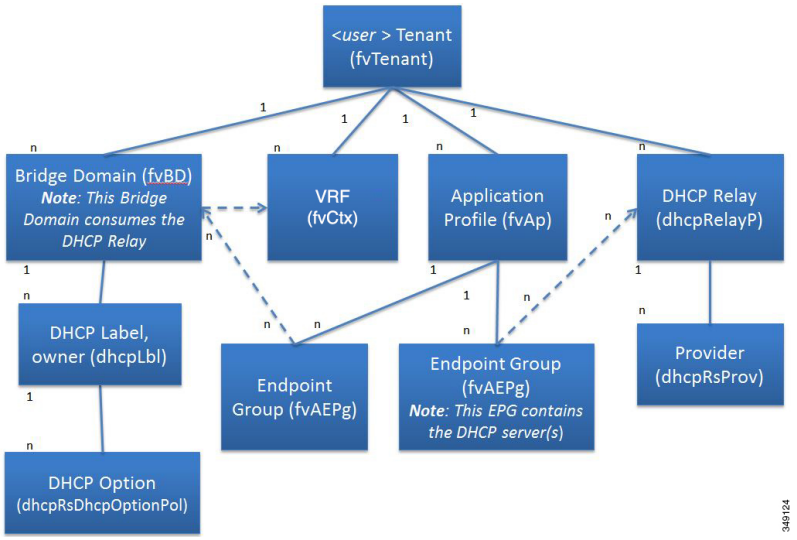
Figure 56: DHCP Relay Locations in the MIT



Note DHCP relay is limited to a single subnet per bridge domain.

The figure below shows the logical relationships of the DHCP relay objects within a user tenant.

Figure 57: Tenant DHCP Relay



The DHCP Relay profile contains one or more providers. An EPG contains one or more DHCP servers, and the relation between the EPG and DHCP Relay specifies the DHCP server IP address. The consumer bridge domain contains a DHCP label that associates the provider DHCP server with the bridge domain. Label matching enables the bridge domain to consume the DHCP Relay.

Note The bridge domain DHCP label must match the DHCP Relay name.

The DHCP label object also specifies the owner. The owner can be a tenant or the access infrastructure. If the owner is a tenant, the ACI fabric first looks within the tenant for a matching DHCP Relay. If there is no match within a user tenant, the ACI fabric then looks in the common tenant.

DHCP Relay operates in the Visable mode as follows: `Visible`—the provider's IP and subnet are leaked into the consumer's VRF. When the DHCP Relay is visible, it is exclusive to the consumer's VRF.

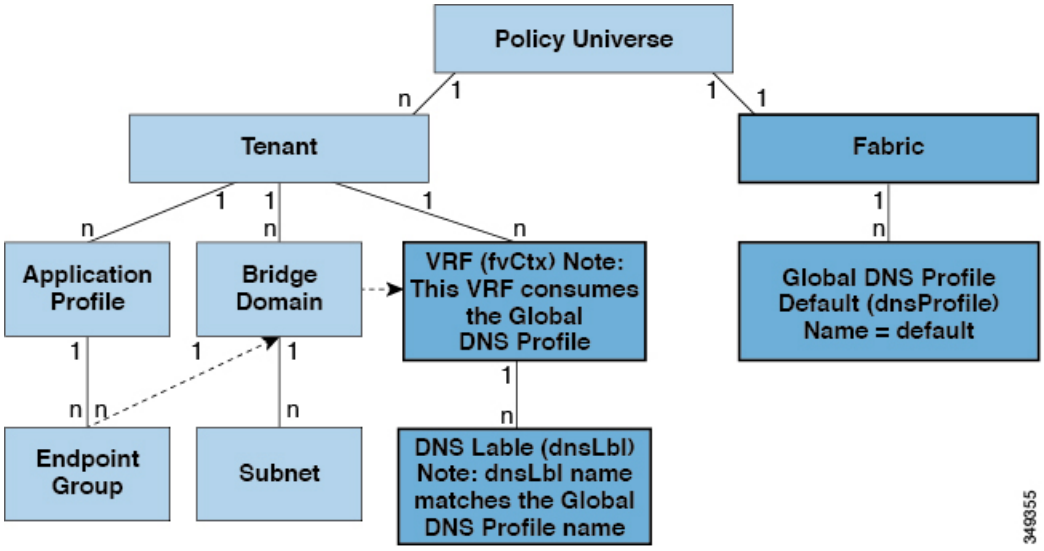
While the tenant and access DHCP Relays are configured in a similar way, the following use cases vary accordingly:

- Common tenant DHCP Relays can be used by any tenant.
- Infra tenant DHCP Relays are exposed selectively by the ACI fabric service provider to other tenants.
- Fabric Access (`infraInfra`) DHCP Relays can be used by any tenant and allow more granular configuration of the DHCP servers. In this case, it is possible to provision separate DHCP servers within the same bridge domain for each leaf switch in the node profile.

DNS

The ACI fabric DNS service is contained in the fabric managed object. The fabric global default DNS profile can be accessed throughout the fabric. The figure below shows the logical relationships of the DNS-managed objects within the fabric.

Figure 58: DNS



A VRF (context) must contain a `dnsLBL` object in order to use the global default DNS service. Label matching enables tenant VRFs to consume the global DNS provider. Because the name of the global DNS profile is “default,” the VRF label name is "default" (`dnsLBL name = default`).

In-Band and Out-of-Band Management Access

The mgmt tenant provides a convenient means to configure access to fabric management functions. While fabric management functions are accessible through the APIC, they can also be accessed directly through in-band and out-of-band network policies.

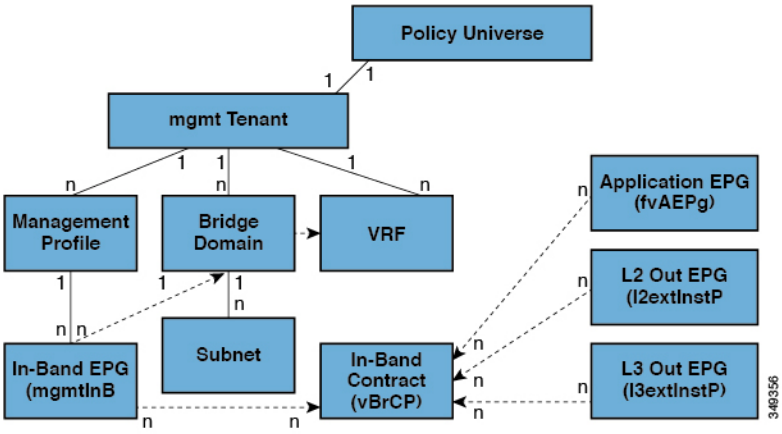
Static and Dynamic Management Access

APIC supports both static and dynamic management access. For simple deployments where users manage the IP addresses of a few leaf and spine switches, configuring static in-band and out-of-band management connectivity is simpler. For more complex deployments, where you might have a large number of leaf and spine switches that require managing many IP addresses, static management access is not recommended. For detailed information about static management access, see *Cisco APIC and Static Management Access*.

In-Band Management Access

The following figure shows an overview of the mgmt tenant in-band fabric management access policy.

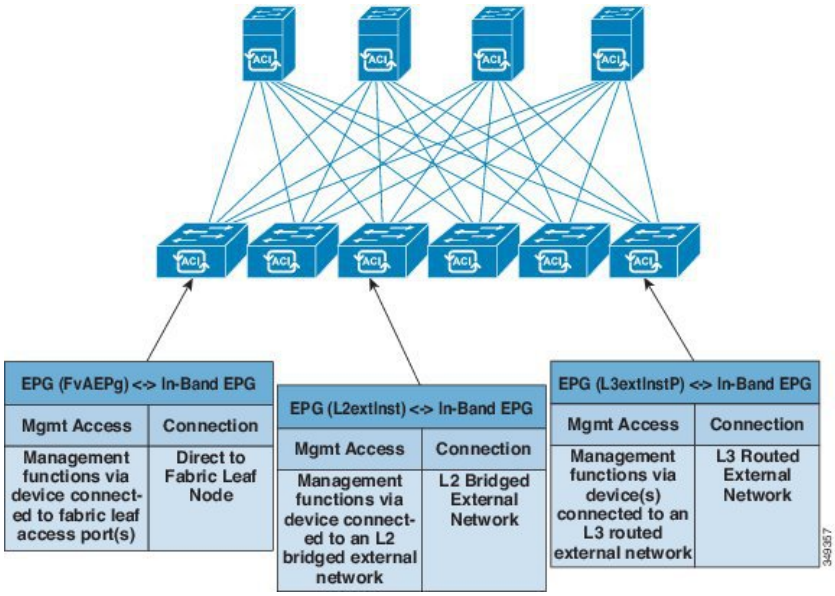
Figure 59: In-Band Management Access Policy



The management profile includes the in-band EPG MO that provides access to management functions via the in-band contract (`vzBrCP`). The `vzBrCP` enables `fvAEPg`, `l2extInstP`, and`l3extInstP` EPGs to consume the in-band EPG. This exposes the fabric management to locally connected devices, as well as devices connected over Layer 2 bridged external networks, and Layer 3 routed external networks. If the consumer and provider EPGs are in different tenants, they can use a bridge domain and VRF from the **common** tenant. Authentication, access, and audit logging apply to these connections; any user attempting to access management functions through the in-band EPG must have the appropriate access privileges.

The figure below shows an in-band management access scenario.

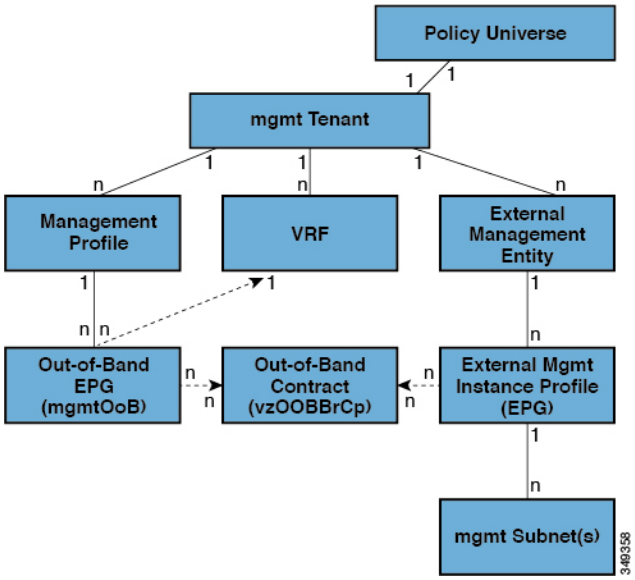
Figure 60: In-Band Management Access Scenario



Out-of-Band Management Access

The following figure shows an overview of the mgmt tenant out-of-band fabric management access policy.

Figure 61: Out-of-Band Management Access Policy

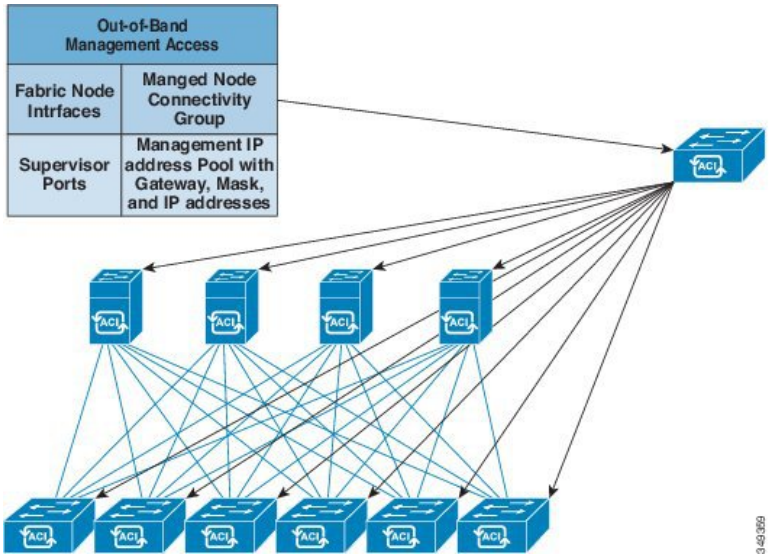


The management profile includes the out-of-band EPG MO that provides access to management functions via the out-of-band contract (`vzOOBBrCP`). The `vzOOBBrCP` enables the external management instance profile (`mgmtExtInstP`) EPG to consume the out-of-band EPG. This exposes the fabric node supervisor ports to locally or remotely connected devices, according to the preference of the service provider. While the bandwidth of the supervisor ports will be lower than the in-band ports, the supervisor ports can provide direct access to

the fabric nodes when access through the in-band ports is unavailable. Authentication, access, and audit logging apply to these connections; any user attempting to access management functions through the out-of-band EPG must have the appropriate access privileges. When an administrator configures an external management instance profile, it specifies a subnet range for devices that are allowed out-of-band access. Any device not in this range will not have out-of-band access.

The figure below shows how out-of-band management access can be consolidated through a dedicated switch.

Figure 62: Out-of-Band Access Scenario



While some service providers choose to restrict out-of-band connectivity to local connections, others can choose to enable routed or bridged connections from external networks. Also, a service provider can choose to configure a set of policies that include both in-band and out-of-band management access for local devices only, or both local and remote devices.

Note Starting with APIC release 1.2(2), when a contract is provided on an out-of-band node management EPG, the default APIC out-of-band contract source address is the local subnet that is configured on the out-of-band node management address. Previously, any address was allowed to be the default APIC out-of-band contract source address.

IPv6 Support

The ACI fabric supports the following IPv6 features for in-band and out-of-band interfaces, tenant addressing, contracts, shared services, routing, Layer 4 - Layer 7 services, and troubleshooting:

- IPv6 address management, pervasive software virtual interface (SVI) bridge domain subnets, outside network external interface addresses, and routes for shared services such as load balancers or intrusion detection.
- Neighbor Discovery using ICMPv6 messages known as router advertisements (RA) and router solicitations (RS), and Duplicate Address Detection (DAD),

- Stateless Address Auto configuration (SLAAC) and DHCPv6.
- Bridge domain forwarding.
- Troubleshooting (see the atomic counters, SPAN, iping6, and traceroute topics in the Troubleshooting Chapter).
- IPv4 only, IPv6 only, or dual stack configuration of in-band and out-of-band interfaces.

Limitations of the current ACI fabric IPv6 implementation include the following:

- Multicast Listener Discovery (MLD) snooping is not supported.
- For IPv6 management, only static addresses are permitted; dynamic IPv6 pools are not supported for IPv6 management.
- IPv6 tunnel interfaces (Intra-Site Automatic Tunnel Addressing Protocol, 6to4 and so forth) are not supported within the fabric; IPv6 tunnel traffic run over the fabric is transparent to the fabric.

ACI fabric interfaces can be configured with link local, global unicast, and multicast IPv6 addresses.

Note While many of the examples provided in this manual use IPv4 addresses, IPv6 addresses could also be used.

A global unicast address can be routed across the public Internet; it is globally unique within a routing domain. A Link Local Address (LLA) has link-local scope and is unique on the link (subnet). The LLA cannot be routed across subnets. These are used by control protocols such as neighbor discovery or OSPF. Multicast addresses are used by IPv6 control protocols such as Neighbor Discovery to deliver packets to more than one endpoint. These are not configurable; they are automatically generated by the protocol components.

Global Unicast Addresses

An administrator can manually specify one or more complete 128-bit IPv6 global unicast addresses on an interface in compressed or uncompressed format. For example, the administration can specify the address in one of the following formats: '2001:0000:0000:0001:0000:0000:0000:0003', '2001:0:0:1:0:0:0:3', '2001:0:0:1::3'. In the ACI fabric naming property, an IPv6 address is always represented in the compressed format. In the above example, the Relative Name is: 2001:0:0:1::3. The administrator can choose any mask length as appropriate for the address.

An administrator can also specify an ACI fabric IPv6 global unicast address in EUI-64 format. As specified in RFC2373, Extended Unique Identifier (EUI) enables a host to assign itself a unique 64-bit IPv6 interface identifier (EUI-64). The IPv6 EUI-64 format address is obtained by incorporating the switch MAC address within the 128-bit IPv6 global unicast address. This feature of IPv6 eliminates the need for manual configuration or DHCP. An IPv6 address for a bridge domain or Layer 3 interface specified in the EUI-64 format is formed this way: <IPv6 prefix>::/<mask>/eui64 where the mask is <=64. For example, 2002::/64/eui64 is what the administrator specifies, and the switch assigns the address as 2002::222:bdff:fef8:19ff/64. The switch uses the switch MAC address to create the EUI-64 address. The formed IPv6 address is contained in the `operAddr` field of the `ipv6If` object.

Note The EUI-64 format can only be used for pervasive bridge domain and Layer 3 interface addresses. It cannot be used for other IP fields in the fabric such as an external server address or for DHCP relay.

Bridge domain subnets and Layer 3 external interface IP addresses can be IPv6 global addresses with a mask ranging from /1 to /127. A bridge domain can contain multiple IPv4 and IPv6 subnets. To support IPv4 and IPv6 address on the same L3 external interface, the administrator creates multiple interface profiles. When an EPG or external EpP gets deployed on the switch, the presence of a manually configured link-local address for the equivalent bridge domain/L3 Interface or an IPv6 address for the subnet/address field results in the creation of `ipv6If` interface in the switch.

Link-Local Addresses

One Link-Local Address (LLA) can be assigned to an interface. The LLA can be autogenerated or configured by an administrator. By default, an ACI LLA is autogenerated by the switch in EUI-64 format. An administrator must configure at least one global address on the interface for an autogenerated LLA to be generated on the switch. The autogenerated address is saved in the `operllAddr` field of the `ipv6If` MO. For pervasive SVIs the MAC address used is the same as the configured interface MAC address. For other kinds of interfaces the switch MAC address is used. An administrator has the option to manually specify a complete 128-bit IPv6 link-local address on an interface in compressed or uncompressed format.

Note The switch hardware tables are limited to one LLA per Virtual Routing and Forwarding (VRF) instance.

Each pervasive bridge domain can have a single IPv6 LLA. This LLA can be set by an administrator, or can be automatically configured by the switch when one isn't provided. When automatically configured, the switch forms the LLA in the modified EUI-64 format where the MAC address is encoded in the IPv6 address to form a unique address. A pervasive bridge domain uses one LLA on all the leaf nodes.

Follow these guidelines for setting LLAs:

- For external SVI and VPC members, the LLA is unique for every leaf node.
- LLAs can be changed to manual (non-zero manually specified link-local addresses) or auto (by manually setting the specified link-local address to zero) anytime in the lifecycle of the interface.
- LLAs specified by an administrator must conform to the IPv6 link-local format (FE80:/10).
- The IPv6 interface MO (`ipv6If`) is created on the switch upon the creation of the first global address on the interface, or when an administrator manually configures an LLA, whichever happens first.
- An administrator-specified LLA is represented in the `llAddr` property in the bridge domain and Layer 3 interface objects in the logical model.
- The LLA used by the switch (either from `llAddr` or autogenerated when `llAddr` is zero is represented in the `operLlAddr` property in the corresponding ipv6If object.
- Operational LLA-related errors like duplicate LLAs are detected by the switch during Duplicate Address Detection process and recorded in `operStQual` field in the `ipv6If` object or raise faults as appropriate.
- Apart from the `llAddr` fields, an LLA (FE80:/10) cannot be a valid address in any other IP address field in the APIC (such as external server addresses or bridge domain subnets) as these addresses cannot be routed.

Static Routes

ACI IPv6 static routes are similar to what is supported in the IPv4, except for the address and prefix format differences in the configurations. The following types of static routes are typically handled by IPv6 static route module:

- Local Routes: Any /128 address configured on an interface leads to a local route that points to the CPU.
- Direct routes: For any configured address on a pervasive BD, the policy element pushes a subnet route pointing to an IPv4 proxy tunnel destination on the spine. For any configured address on a non-pervasive Layer 3 external interface, the IPv6 manager module automatically pushes a subnet route pointing to the CPU.
- Static routes pushed from PE: Used for external connectivity. The next hop IPv6 address for such routes can be on a directly connected subnet on the external router or a recursive next hop that can be resolved to a real next hop on a directly connected subnet. Note that the interface model does not allow an interface as a next hop (though it is supported in the switch). Used to enable shared services across tenants, the next hop for shared-services static routes is located in the shared services Virtual Routing and Forwarding (VRF) instance, which is different from the tenant VRF, where the route is installed on the ingress leaf switches.

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)

Duplicate Address Detection

Duplicate address detection (DAD) discovers any other node on the link that is already using the address being configured. DAD is performed for both link-local and global addresses. Each configured address maintains the following DAD states:

- `NONE`—This is the state when the address is initially created before attempting the DAD.
- `VALID`—This is the state that represents the address has successfully passed the DAD process without detecting the address as a duplicate address.
- `DUP`—This is the state that represents the address is found as duplicate on the link.

Any configured address is usable for sending and receiving IPv6 traffic only if its DAD state is VALID.

Stateless Address Autoconfiguration (SLAAC) and DHCPv6

The following host configurations are supported:

- SLAAC only
- DHCPv6 only
- SLAAC and DHCPv6 stateless used together use SLAAC for address configuration only, but uses DHCPv6 for DNS resolution and other functions.

IPv6 addresses are supported for DHCP relay. DHCPv6 relay applies across Virtual Routing and Forwarding (VRF) instances. DHCP relay over VLAN and VXLAN are also supported. DHCPv4 works in conjunction with DHCPv6.

Routing Within the Tenant

The Application Centric Infrastructure (ACI) fabric provides tenant default gateway functionality and routes between the fabric virtual extensible local area (VXLAN) networks. For each tenant, the fabric provides a virtual default gateway or Switched Virtual Interface (SVI) whenever a subnet is created on the APIC. This spans any switch that has a connected endpoint for that tenant subnet. Each ingress interface supports the default gateway interface and all of the ingress interfaces across the fabric share the same router IP address and MAC address for a given tenant subnet.

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.

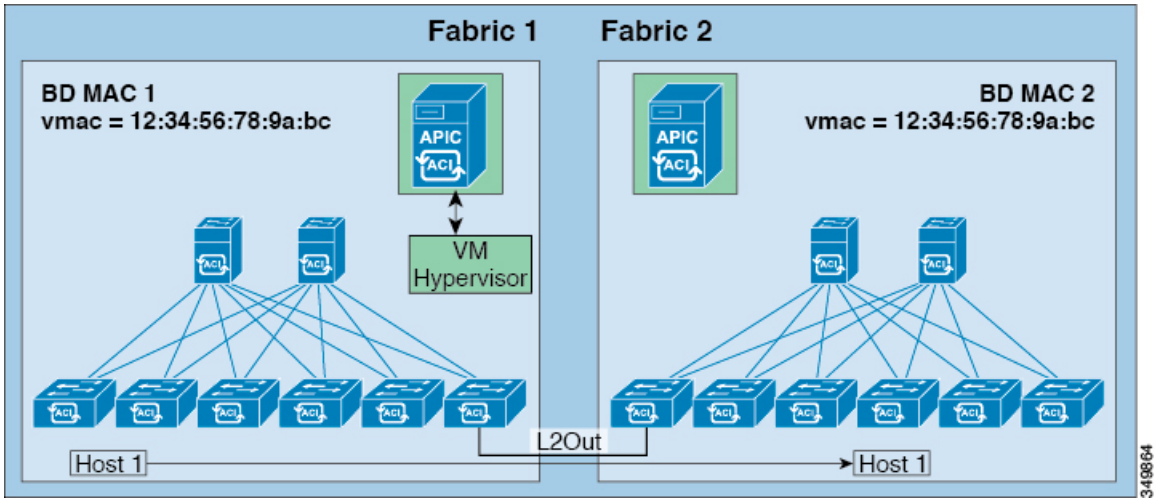
After both L3Outs and spine route reflectors are deployed, border leaf nodes learn external routes via L3Outs, and those external routes are distributed to all leaf nodes in the fabric via spine MP-BGP route reflectors.

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

Common Pervasive Gateway

Multiple ACI fabrics can be configured with an IPv4 common gateway on a per bridge domain basis. Doing so enables moving one or more virtual machines (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 routed WAN link. The following figure illustrates the basic common pervasive gateway topology.

Figure 63: ACI Multi-Fabric Common Pervasive Gateway



The per-bridge domain common pervasive gateway configuration requirements are as follows:

- The bridge domain MAC (*mac*) values for each fabric must be unique.

Note The default bridge domain MAC (*mac*) address values are the same for all ACI fabrics. The common pervasive gateway requires an administrator to configure the bridge domain MAC (*mac*) values to be unique for each ACI fabric.

- The bridge domain virtual MAC (*vmac*) 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.

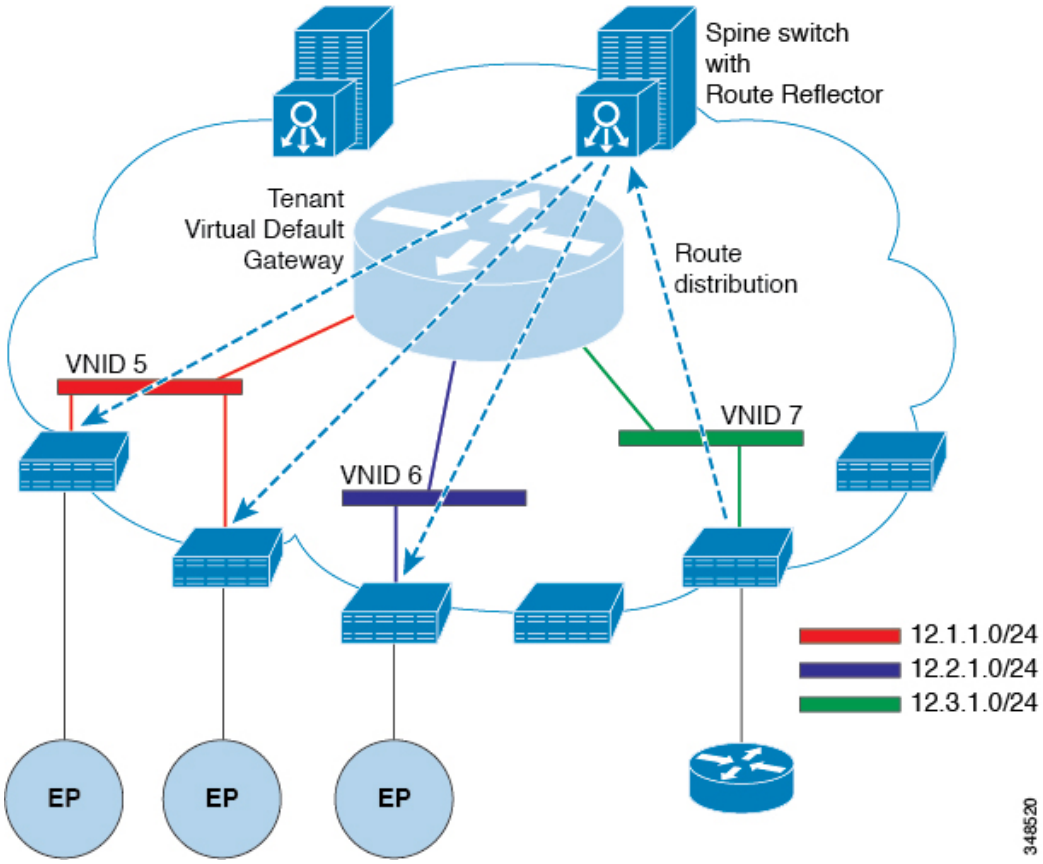
WAN and Other External Networks

External routers that connect to the WAN and the enterprise core connect to the front panel interfaces of the leaf switch. The leaf switch interface that connects to the external router can be configured as a bridged interface or a routing peer.

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 64: 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.

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.

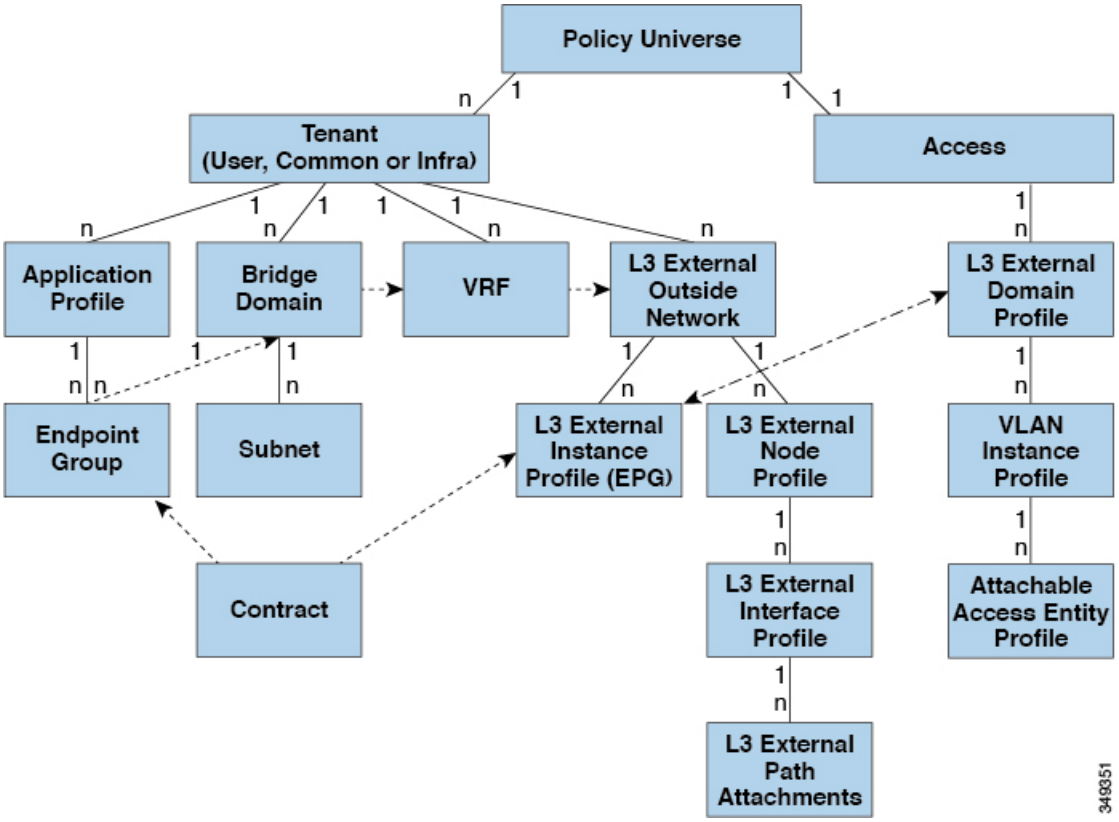
Bridged and Routed Connectivity to External Networks

Outside network managed objects enable Layer 2 and Layer 3 tenant connectivity to external networks. The GUI, CLI, or REST API can be used to configure tenant connectivity to external networks. To easily locate the external network access points in the fabric, Layer 2 and Layer 3 external leaf nodes can be tagged as "Border Leaf Nodes."

Layer 2 Out for Bridged Connectivity to External Networks

Tenant Layer 2 bridged connectivity to external networks is enabled by associating a fabric access (`infraInfra`) external bridged domain (`L2extDomP`) with the Layer 2 external instance profile (`l2extInstP`) EPG of a Layer 2 external outside network (`l2extOut`) as shown in the figure below.

Figure 65: Tenant Bridged Connectivity to External Networks

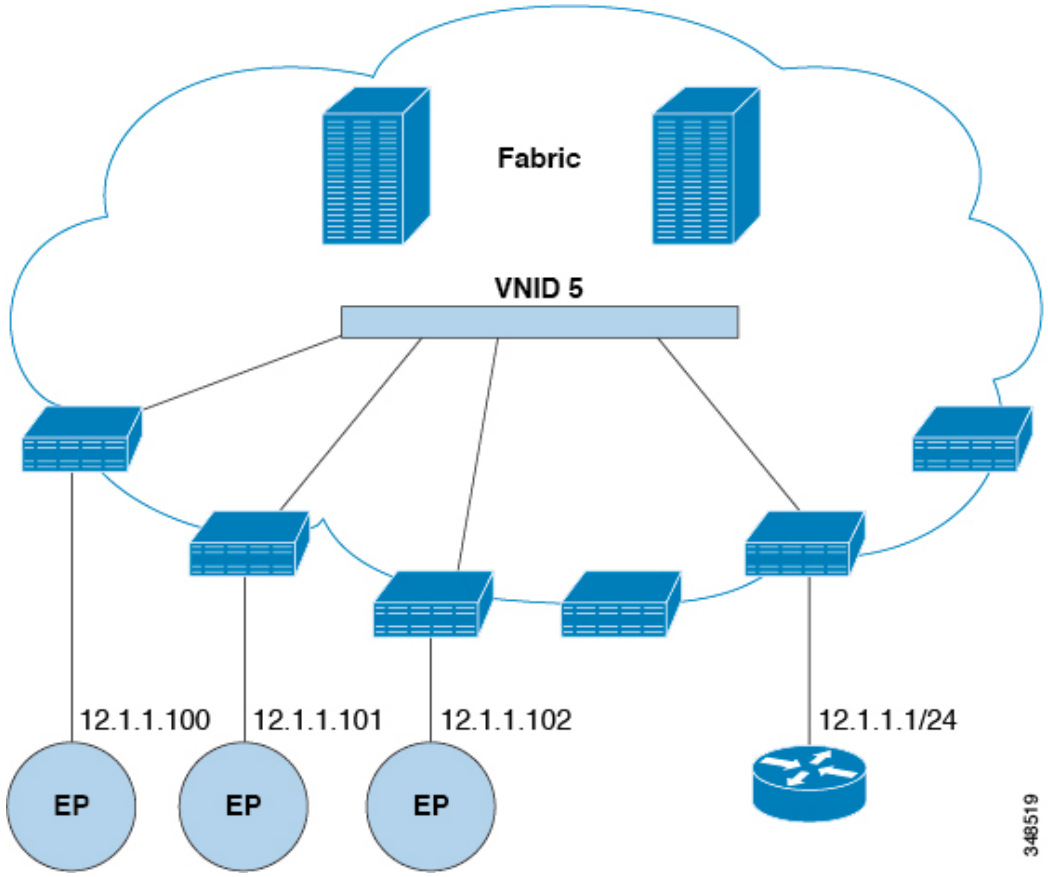


The `l2extOut` includes the switch-specific configuration and interface-specific configuration. The `l2extInstP` EPG exposes the external network to tenant EPGs through a contract. For example, a tenant EPG that contains a group of network-attached storage devices could communicate through a contract with the `l2extInstP` EPG according to the network configuration contained in the Layer 2 external outside network. Only one outside network can be configured per leaf switch. However, the outside network configuration can easily be reused for multiple nodes by associating multiple nodes with the Layer 2 external node profile. Multiple nodes that use the same profile can be configured for fail-over or load balancing.

Bridged Interface to an External Router

As shown in the figure below, when the leaf switch interface is configured as a bridged interface, the default gateway for the tenant VNID is the external router.

Figure 66: Bridged External Router

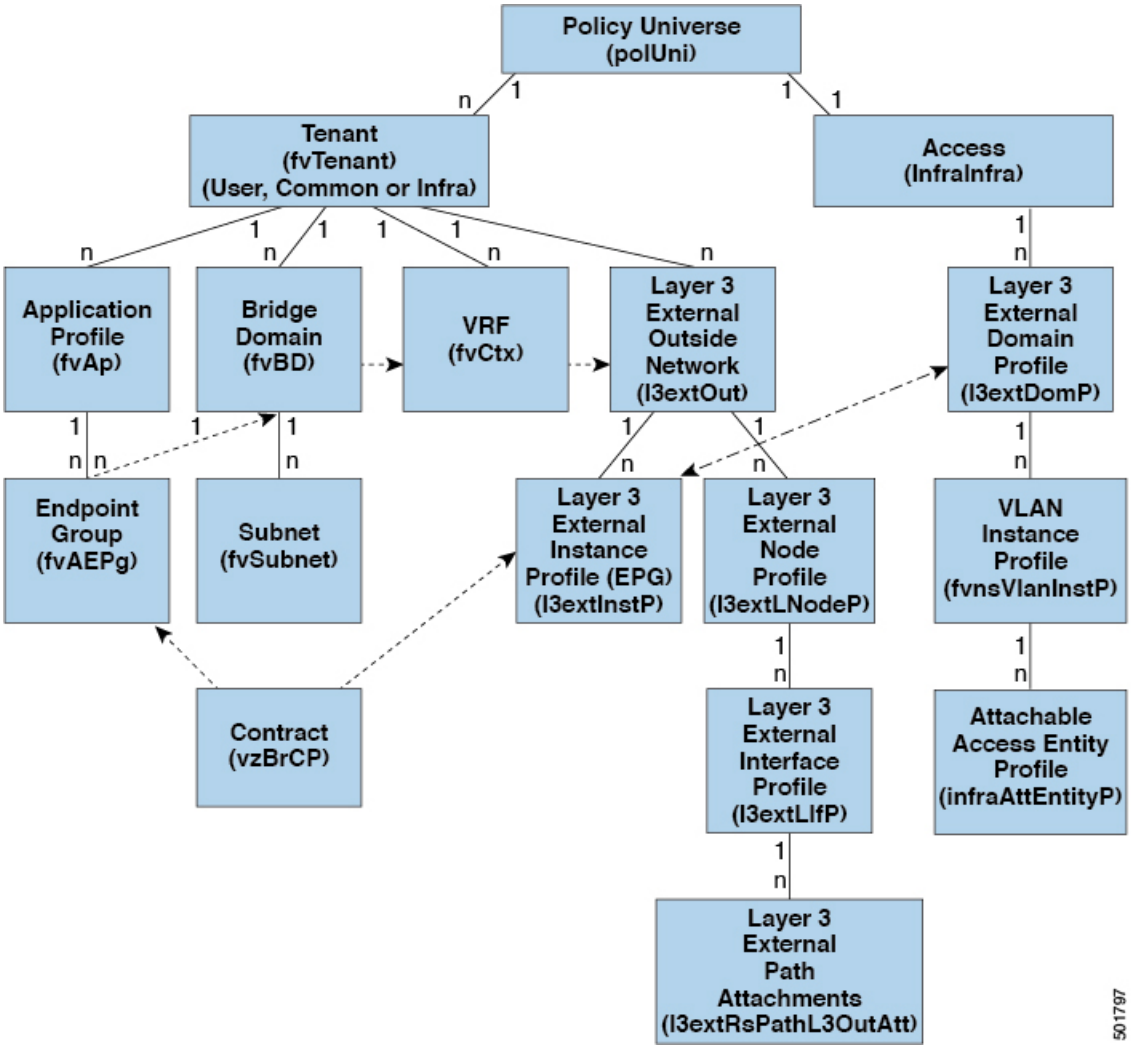


The ACI fabric is unaware of the presence of the external router and the APIC statically assigns the leaf switch interface to its EPG.

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 67: Policy Model for Layer 3 External Connections



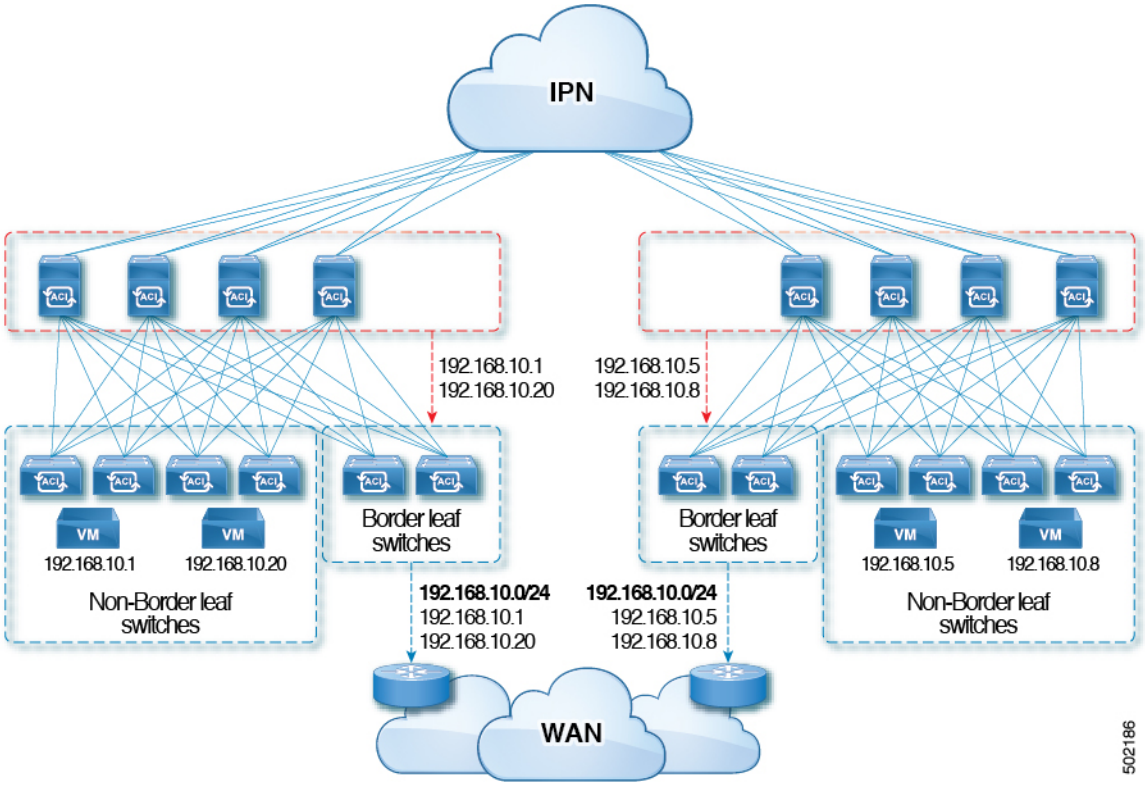
A Layer 3 external outside network (`l3extOut` object) includes the routing protocol options (BGP, OSPF, or EIGRP or supported combinations) and the switch-specific and interface-specific configurations. While the `l3extOut` contains the routing protocol (for example, OSPF with its related Virtual Routing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

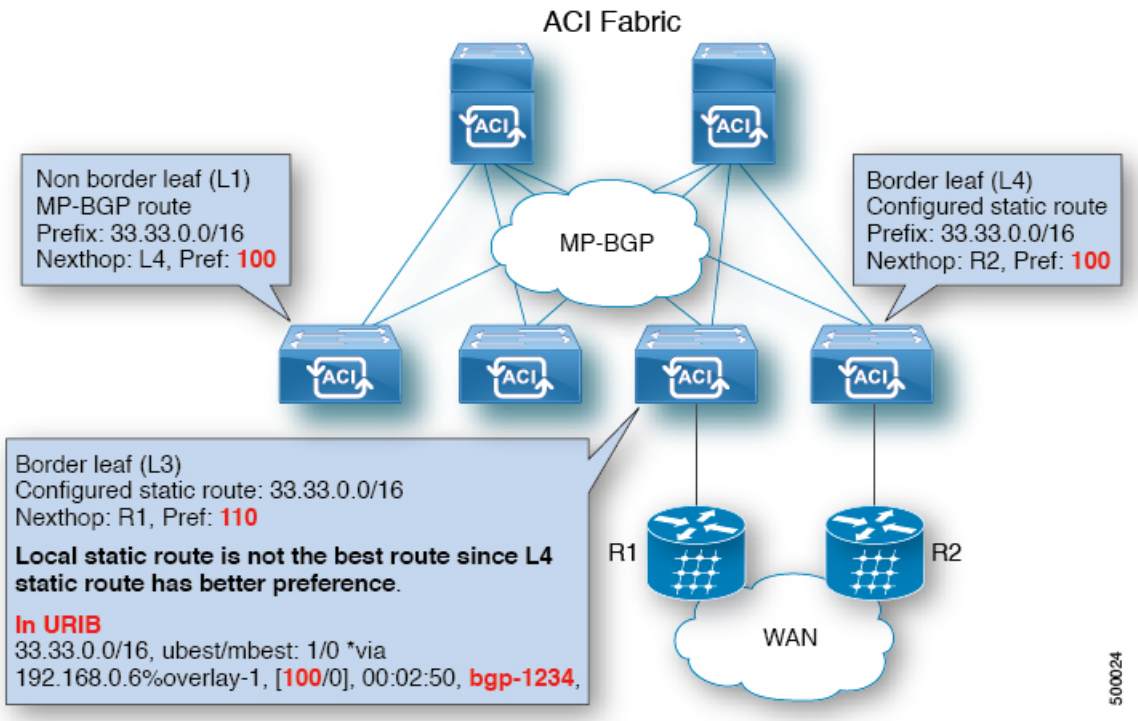
- Advertise Host Route is supported for shared services. For example: epg1/BD1 deployed is in VRF-1 and L3out in another VRF-2. By providing shared contract between EPG and L3out host routes are pulled from one VRF-1 to another VRF-2.
- When Advertise Host Route is enabled on BD custom tag cannot be 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.

Static Route Preference

Static route preference within the ACI fabric is carried in MP-BGP using cost extended community.

The following figure illustrates how the ACI fabric keeps static route preferences intact across leaf switches so that route selection happens based on this preference.

Figure 68: Static Route Preference



This figure shows a MP-BGP route coming to leaf switch 3 (L3) from leaf switch 4 (L4) that wins over a local static route. A static route is installed in the Unicast Routing Information Base (URIB) with the preference configured by an administrator. On an ACI non-border leaf switch, a static route is installed with leaf switch 4 (L4) as its nexthop. When nexthop on L4 is not available, the L3 static route becomes the best route in fabric.

Note If a static route in a leaf switch is defined with `next hop Null 0`, MP-BGP does not advertise that route to other leaf switches in 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 L3Out EPG (`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.

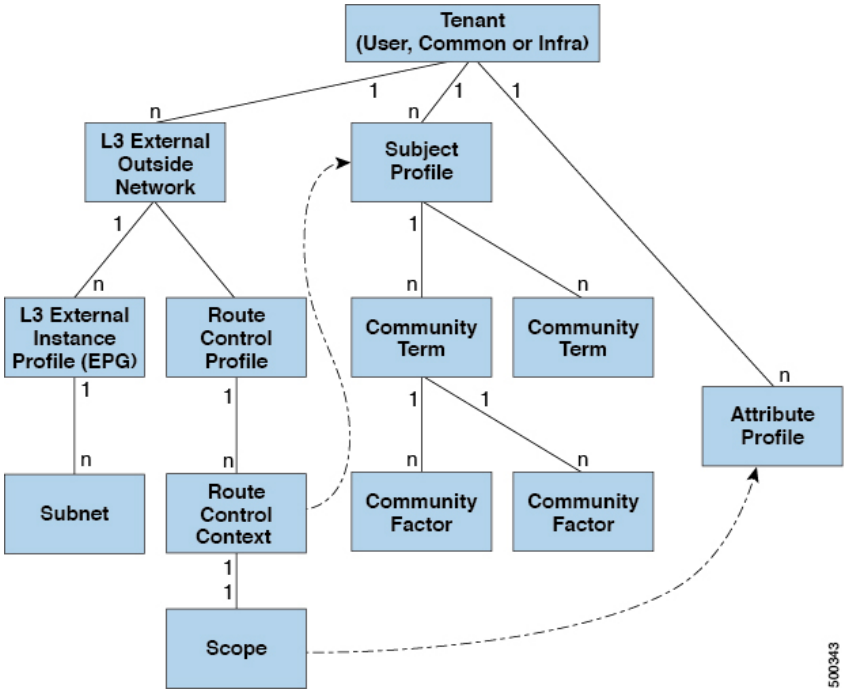
Route summarization simplifies route tables by replacing many specific addresses with an single address. For example, 10.1.1.0/24, 10.1.2.0/24, and 10.1.3.0/24 are replaced with 10.1.0.0/16. Route summarization policies enable routes to be shared efficiently among border leaf switches and their neighbor 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 69: 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.

Shared Services Contracts Usage

Shared services enable communications across tenants while preserving the isolation and security policies of the tenants. A routed connection to an external network is an example of a shared service that multiple tenants use.

Follow these guidelines when configuring shared services contracts.

- For shared services that export subnets to different Virtual Routing and Forwarding (VRF) instances (also known as contexts or private networks), the subnet must be configured under an EPG, and the scope must be set to **Advertised Externally** and **Shared Between VRFs**.
- Contracts are not needed for inter-bridge domain traffic when a VRF is unenforced.
- Contracts are needed for shared service inter-VRF traffic, even when a VRF is unenforced.
- The VRF of a provider EPG cannot be in unenforced mode while providing a shared service.
- A shared service is supported only with non-overlapping and non-duplicate subnets. When configuring subnets for shared services, follow these guidelines:

- Configure the subnet for a shared service provider under the EPG, not under the bridge domain.
- Subnets configured under an EPG that share the same VRF must be disjointed and must not overlap.
- Subnets leaked from one VRF to another must be disjointed and must not overlap.
- Subnets leaked from multiple consumer networks into a VRF or vice versa must be disjointed and must not overlap.

Note If two consumers are mistakenly configured with the same subnet, recover from this condition by removing the subnet configuration for both then reconfigure the subnets correctly.

- Do not configure a shared service with `AnyToProv` in the provider VRF. The APIC rejects this configuration and raises a fault.
- When a contract is configured between in-band and out-of-band EPGs, the following restrictions apply:
  - Both EPGs should be in the same VRF.
  - Ffilters apply only in the incoming direction.
  - Layer 2 filters are not supported.
  - QoS does not apply to in-band Layer 4 to Layer 7 services.
  - Management statistics are not available.
  - Shared services for CPU-bound traffic are not supported.

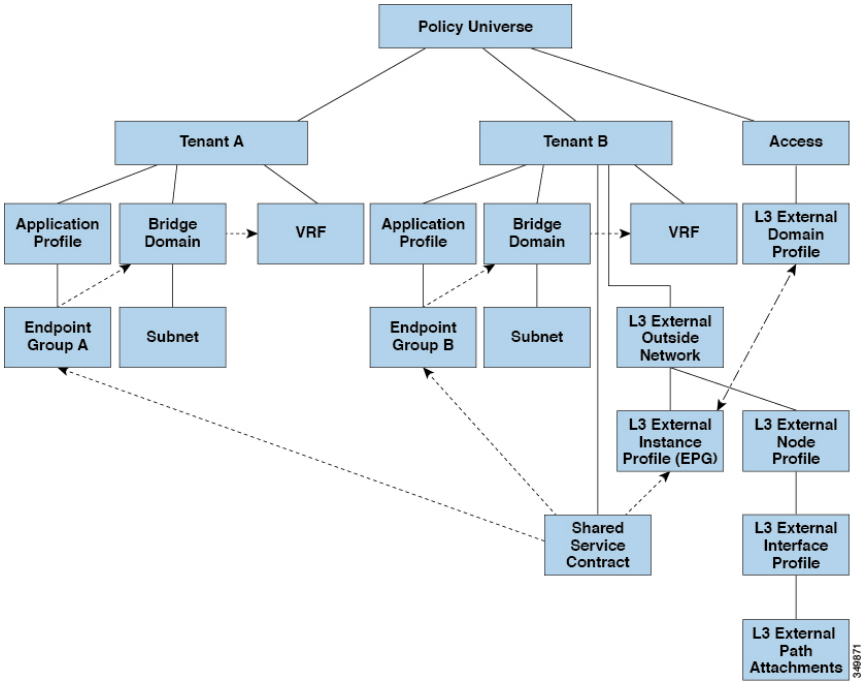
Shared Layer 3 Out

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 70: 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.

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:

- Beginning with 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.
- Beginning with APIC Release 5.0(1), BFD multihop is supported on leaf switches. The maximum number of BFD sessions is unchanged, as BFD multihop sessions are now included in the total.
- Beginning with APIC Release 5.0(1), ACI supports C-bit-aware BFD. The C-bit on incoming BFD packets determines whether BFD is dependent or independent of the control plane.
- 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`.

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.

Tenant Routed Multicast

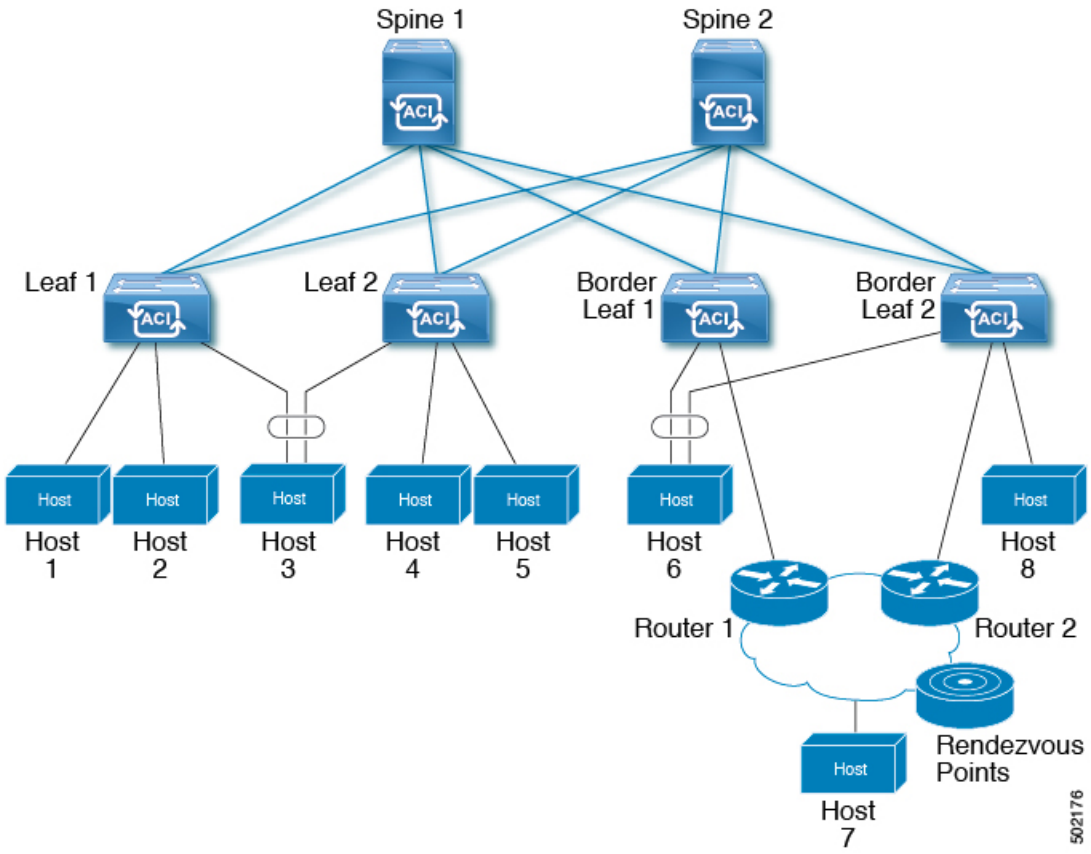
Cisco Application Centric Infrastructure (ACI) Tenant Routed Multicast (TRM) enables Layer 3 multicast routing in Cisco ACI tenant VRF instances. TRM supports multicast fowarding between senders and receivers within the same or different subnets. Multicast sources and receivers can be connected to the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

In the Cisco ACI fabric, most unicast and IPv4/IPv6 multicast routing operate together on the same border leaf switches, with the IPv4/IPv6 multicast protocol operating over the unicast routing protocols.

In this architecture, only the border leaf switches run the full Protocol Independent Multicast (PIM) or PIM6 protocol. Non-border leaf switches run PIM/PIM6 in a passive mode on the interfaces. They do not peer with any other PIM/PIM6 routers. The border leaf switches peer with other PIM/PIM6 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/IPv6 multicast cloud. Each virtual routing and forwarding (VRF) instance in the fabric that requires IPv4/IPv6 multicast routing will peer separately with external IPv4/IPv6 multicast routers.

Figure 71: Overview of Multicast Cloud



About the Fabric Interface

The fabric interface is a virtual interface between software modules and represents the fabric for IPv4/IP6 multicast routing. The interface takes the form of a tunnel interface with the tunnel destination being the VRF GIPo (Group IP outer address)[1]. PIM6 shares the same tunnel that PIM4 uses. 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 state published by the intermediate system-to-intermediate system (IS-IS).

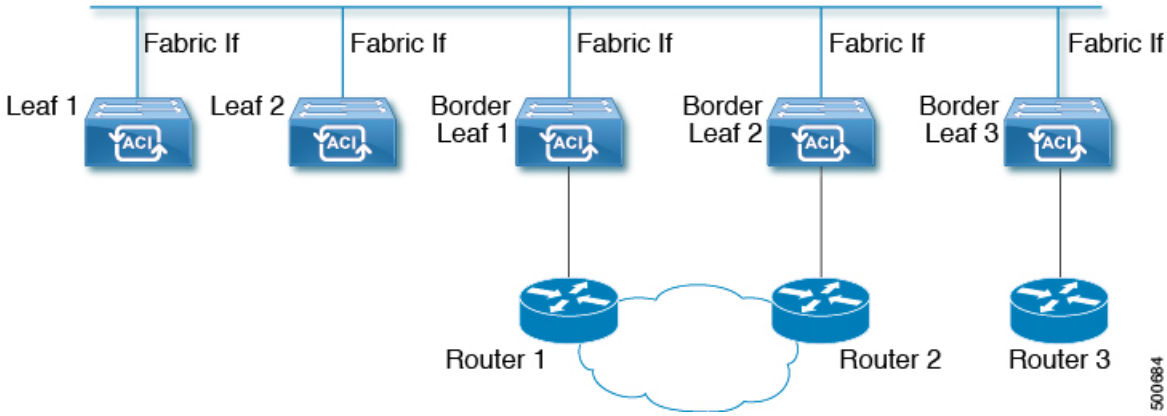
Note Each multicast-enabled VRF requires one or more border leaf switches configured with a loopback interface. You must configure a unique IPv4 loopback address on all nodes in a PIM-enabled L3Out. The Router-ID loopback or another unique loopback address can be used.

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 MP-BGP (Multiprotocol Border Gateway Protocol) routes

[1] The GIPo (Group IP outer address) is the destination multicast IP 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.

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 IPv4/IP6 multicast routing.

Figure 72: Fabric for IPv4/IP6 Multicast Routing



Enabling IPv4/IPv6 Tenant Routed Multicast

The process to enable or disable IPv4 or IPv6 multicast routing in a Cisco ACI fabric occurs at three levels:

- **VRF level**: Enable multicast routing at the VRF level.
- **L3Out level**: Enable PIM/PIM6 for one or more L3Outs configured in the VRF.
- **Bridge domain level**: Enable PIM/PIM6 for one or more bridge domains where multicast routing is needed.

At the top level, IPv4/IPv6 multicast routing must be enabled on the VRF that has any multicast routing-enabled bridge domains. On an IPv4/IPv6 multicast routing-enabled VRF, there can be a combination of IPv4/IPv6 multicast routing-enabled bridge domains and bridge domains where IPv4/IPv6 multicast routing is disabled. A bridge domain with IPv4/IPv6 multicast routing disabled will not show on the VRF IPv4/IPv6 multicast panel. An L3Out with IPv4/IPv6 multicast routing-enabled will show up on the panel, but any bridge domain that has IPv4/IPv6 multicast routing enabled will always be a part of a VRF that has IPv4/IPv6 multicast routing enabled.

IPv4/IPv6 multicast routing is not supported on the leaf switches such as Cisco Nexus 93128TX, 9396PX, and 9396TX. All the IPv4/IPv6 multicast routing and any IPv4/IPv6 multicast-enabled VRF should be deployed only on the switches with -EX and -FX in their product IDs.

Note Layer 3 Out ports and sub-interfaces are supported while external SVIs are not supported. Since external SVIs are not supported, PIM/PIM6 cannot be enabled in L3-VPC.

Guidelines, Limitations, and Expected Behaviors for Configuring Layer 3 IPv4/IPv6 Multicast

See the following guidelines and restrictions:

• Guidelines and Limitations for IPv4 and IPv6 Multicast, on page 157
• Guidelines and Limitations for IPv4 Multicast, on page 158
• Guidelines and Limitations for IPv6 Multicast, on page 159

Guidelines and Limitations for IPv4 and IPv6 Multicast

The following restrictions apply for both IPv4 and IPv6 multicast:

- The Layer 3 IPv4/IPv6 multicast feature is supported on second generation leaf switches. A second generation switch is one with -EX, -FX, -FX2, -FX3, -GX, or any later suffix in the product ID.
- Custom QoS policy is not supported for Layer 3 multicast traffic sourced from outside the Cisco Application Centric Infrastructure (ACI) fabric (received from L3Out).
- Enabling PIMv4/PIM6 and Advertise Host routes on a bridge domain is supported.
- Layer 3 multicast is enabled at the VRF level and the multicast protocols will function within the VRF instance. Each VRF instance can have multicast enabled or disabled independently.
- After a VRF instance is enabled for multicast, the individual bridge domains and L3Outs under the enabled VRF instance can be enabled for multicast configuration. By default, multicast is disabled in all bridge domains and L3Outs.
- Bidirectional PIMv4/PIM6 is currently not supported.
- Multicast routers are not supported in pervasive bridge domains.
- The supported route scale is 2,000. The multicast scale number is a combined scale that includes both IPv4 and IPv6. The total route limit is defined as route counts. Each IPv4 route is counted as 1, and each IPv6 route is counted as 4. Even with node profiles that support more multicast scales, the IPv6 route scale will remain at 2,000.
- PIMv4/PIM6 is supported on Layer 3 Out routed interfaces, routed subinterfaces including Layer 3 port-channel interfaces, and SVI interfaces.
- Enabling PIMv4/PIM6 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.
- The behavior of an ingress leaf switch receiving a packet from a source that is attached to a bridge domain differs for Layer 3 IPv4 or IPv6 multicast support:
    - For Layer 3 IPv4 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 IPv4 multicast routing, the ingress leaf switch sends only a routed VRF instance 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.

- For Layer 3 IPv6 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 IPv6 multicast routing, the ingress leaf switch sends only a routed VRF instance 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. The egress leaf also decrements the TTL in the packet by 1. This results in TTL being decremented two times. Also, for ASM the multicast group must have a valid RP configured.

- You cannot use a filter with inter-VRF multicast communication.
- Do not use the “clear ip mroute” command. This command is used for internal debugging and is not supported in a production network.

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`.

Guidelines and Limitations for IPv4 Multicast

The following restrictions apply specifically for IPv4 multicast:

- If the border leaf switches in your Cisco 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).
- Any Source Multicast (ASM) and Source-Specific Multicast (SSM) are supported for IPv4.
- You can configure a maximum of four ranges for SSM multicast in the route map per VRF instance.
- IGMP snooping cannot be disabled on pervasive bridge domains with multicast routing enabled.
- 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. Multicast sources or receivers that are connected to FEX ports are not supported.

Guidelines and Limitations for IPv6 Multicast

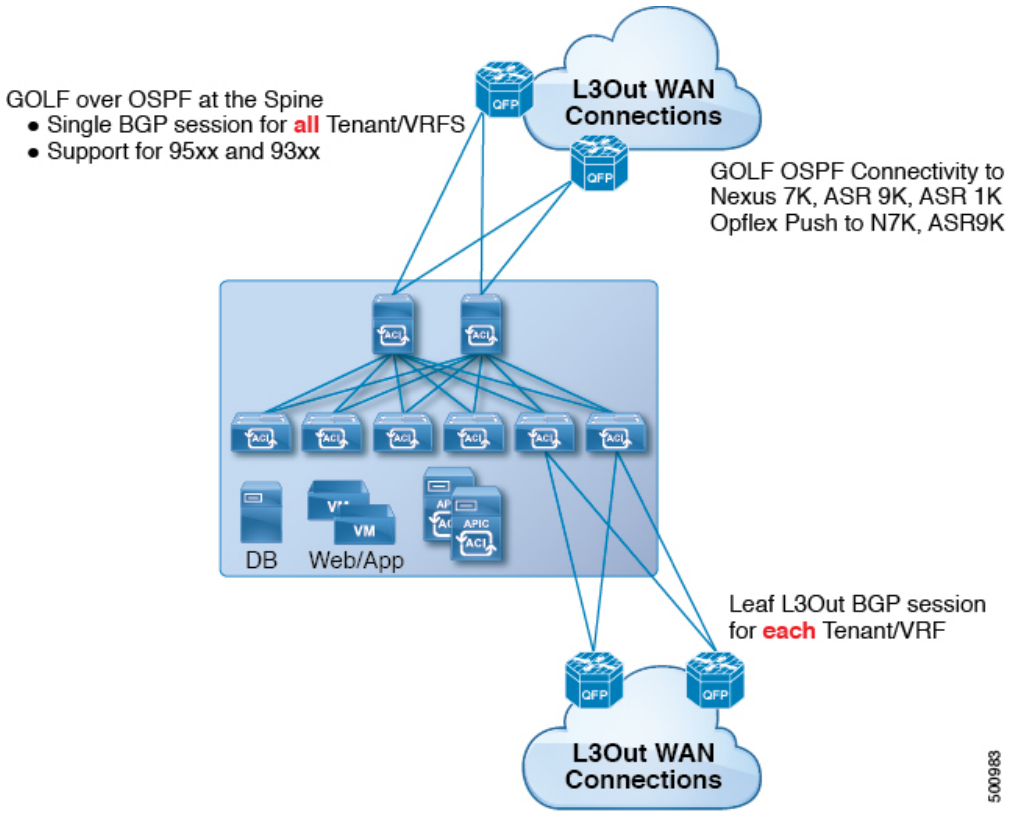
The following restrictions apply specifically for IPv6 multicast:

- Source Specific Multicast (SSM) is supported, but *RFC 3306 - Unicast-Prefix-based IPv6 Multicast Addresses* specifies a fixed SSM range. Therefore, the SSM range cannot be changed in IPv6.
- You can configure a maximum of four ranges for SSM multicast in the route map per VRF instance.
- Any Source Multicast (ASM) is supported for IPv6.
- OIF and VRF scale numbers for IPv6 are the same as they are for IPv4.
- For PIM6 only static RP configuration is supported. Auto-RP and BSR are not supported for PIM6.
- Receivers inside the fabric are not supported. MLD Snoop Policy must be disabled when enabling IPv6 multicast. MLD snooping and PIM6 cannot be enabled in the same VRF instance.
- Currently, Layer 3 Multicast Listener Discovery (MLD) is not supported with Cisco ACI.
- Fabric Rendezvous Point (RP) is not supported for IPv6 multicast.
- Cisco Multi-Site Orchestrator support is not available.

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 73: 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`.

Route Target filtering

Route target filtering is the practice of optimizing BGP routing tables by filtering the routes that are stored on them. This action can be accomplished by explicit route target policy or by automated algorithm.

Route Target Policy

A route target policy explicitly defines the BGP routes that can be shared between VRFs. It specifies which local routes can be exported from the local VRF to another and specifies which routes can be imported into the local VRF from external VRFs.

Within APIC, route target policies can be specified during creation or configuration of a VRF. which can in turn be associated with an L3 Out policy to define BGP route sharing associated with that policy.

Auto Route Target filtering

Auto route target filtering implements an automated algorithm for optimizing BGP routing tables for maximum overall efficiency, conserving memory by filtering out storage of all imported BGP route targets except for those associated with directly connected VPNs.

When a VRF receives a BGP VPN-IPv4 or VPN-IPv6 route target from another Policy Element (PE) router, BGP stores that route target in its local routing table only if at least one VRF imports a route target of that route. If no VRF imports any of the route targets of the route, BGP discards the route target; The intention is that BGP keeps track of route targets only for directly connected VPNs, and discards all other VPN-IPv4 or VPN-IPv6 route targets to conserve memory.

If a new VPN is connected to the router (that is, if the import route-target list of a VRF changes), BGP automatically sends a route-refresh message to obtain the routes that it previously discarded.

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.

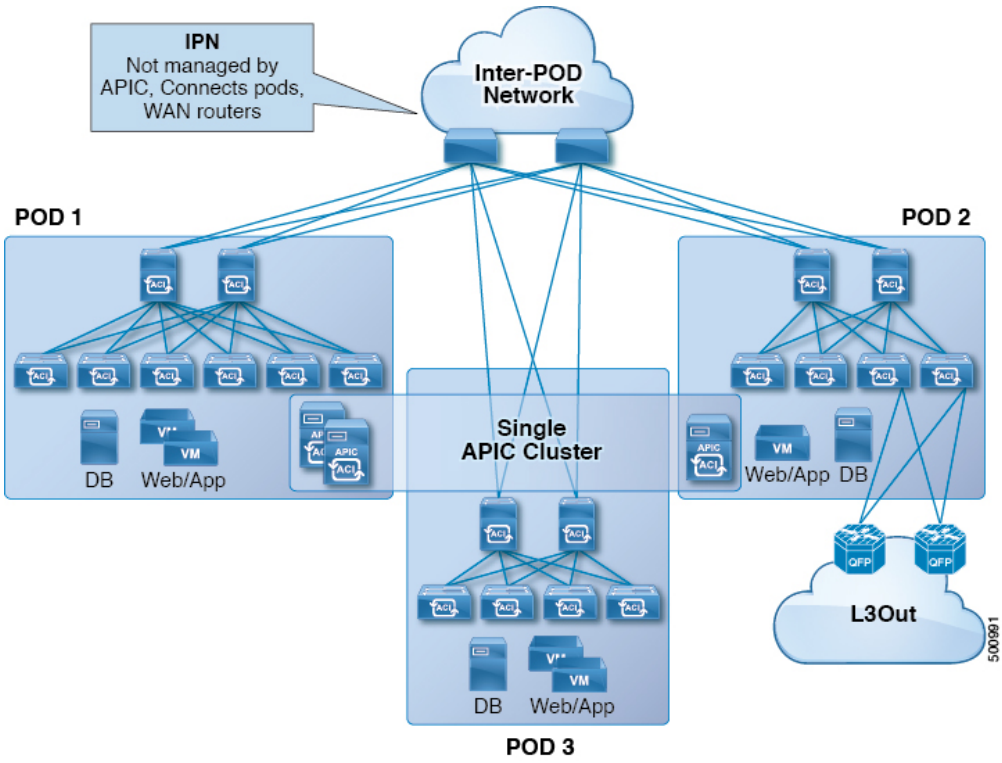
b. Configure BGP Route Target Profiles that identify routes that can be imported or exported from the VRF.

Multipod

Multipod enables provisioning a more fault tolerant fabric comprised of multiple pods with isolated control plane protocols. Also, multipod 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, multipod enables provisioning multiple pods per floor or building and providing connectivity between pods through spine switches.

Multipod uses MP-BGP EVPN as the control-plane communication protocol between the ACI spines in different Pods. WAN routers can be provisioned in the IPN, directly connected to spine switches, or connected to border leaf switches. Multipod 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 74: Multipod Overview



For control plane isolation, IS-IS and COOP are not extended across pods. Endpoints synchronize across pods using BGP EVPN over the IPN between the pods. Two spines in each pod are configured to have BGP EVPN sessions with spines of other pods. The spines connected to the IPN get the endpoints and multicast groups from COOP within a pod, but they advertise them over the IPN EVPN sessions between the pods. On the receiving side, BGP gives them back to COOP and COOP synchs them across all the spines in the pod. WAN routes are exchanged between the pods using BGP VPNv4/VPNv6 address families; they are not exchanged using the EVPN address family.

There are two modes of setting up the spine switches for communicating across pods as peers and route reflectors:

- **Automatic**
  - Automatic mode is a route reflector based mode that does not support a full mesh where all spines peer with each other. The administrator must post an existing BGP route reflector policy and select IPN aware (EVPN) route reflectors. All the peer/client settings are automated by the APIC.
  - The administrator does not have an option to choose route reflectors that don't belong to the fabric (for example, in the IPN).
- **Manual**
  - The administrator has the option to configure full mesh where all spines peer with each other without route reflectors.
  - In manual mode, the administrator must post the already existing BGP peer policy.

Observe the following multipod guidelines and limitations:

- When adding a pod to the ACI fabric, wait for the control plane to converge before adding another pod.
- OSPF is deployed on ACI spine switches and IPN switches to provide reachability between PODs. Layer 3 subinterfaces are created on spines to connect to IPN switches. OSPF is enabled on these Layer 3 subinterfaces and per POD TEP prefixes are advertised over OSPF. There is one subinterface created on each external spine link. Provision many external links on each spine if the expectation is that the amount of east-west traffic between PODs will be large. Currently, ACI spine switches support up to 64 external links on each spine, and each subinterface can be configured for OSPF. Spine proxy TEP addresses are advertised in OSPF over all the subinterfaces leading to a maximum of 64 way ECMP on the IPN switch for proxy TEP addresses. Similarly, spines would receive proxy TEP addresses of other PODs from IPN switches over OSPF and the spine can have up to 64 way ECMP for remote pod proxy TEP addresses. In this way, traffic between PODs spread over all these external links provides the desired bandwidth.
- When the all fabric links of a spine switch are down, OSPF advertises the TEP routes with the maximum metric. This will force the IPN switch to remove the spine switch from ECMP which will prevent the IPN from forwarding traffic to the down spine switch. Traffic is then received by other spines that have up fabric links.
- Up to APIC release 2.0(2), multipod is not supported with GOLF. In release 2.0 (2) the two features are supported in the same fabric only over Cisco Nexus N9000K switches without “EX” on the end of the switch name; for example, N9K-9312TX. Since the 2.1(1) release, the two features can be deployed together over all the switches used in the multipod and EVPN topologies.
- In a multipod fabric, if a spine in POD1 uses the infra tenant L3extOut-1, the TORs for the other pods (POD2, POD3) cannot use the same infra L3extOut (L3extOut-1) for Layer 3 EVPN control plane connectivity. Each POD must use their own spine switch and infra L3extOut, because it is not supported to use a pod as a transit for WAN connectivity of other pods.
- No filtering is done for limiting the routes exchanged across pods. All end-point and WAN routes present in each pod are exported to other pods.
- Inband management across pods is automatically configured by a self tunnel on every spine.
- The maximum latency supported between pods is 10 msec RTT, which roughly translates to a geographical distance of up to 500 miles.

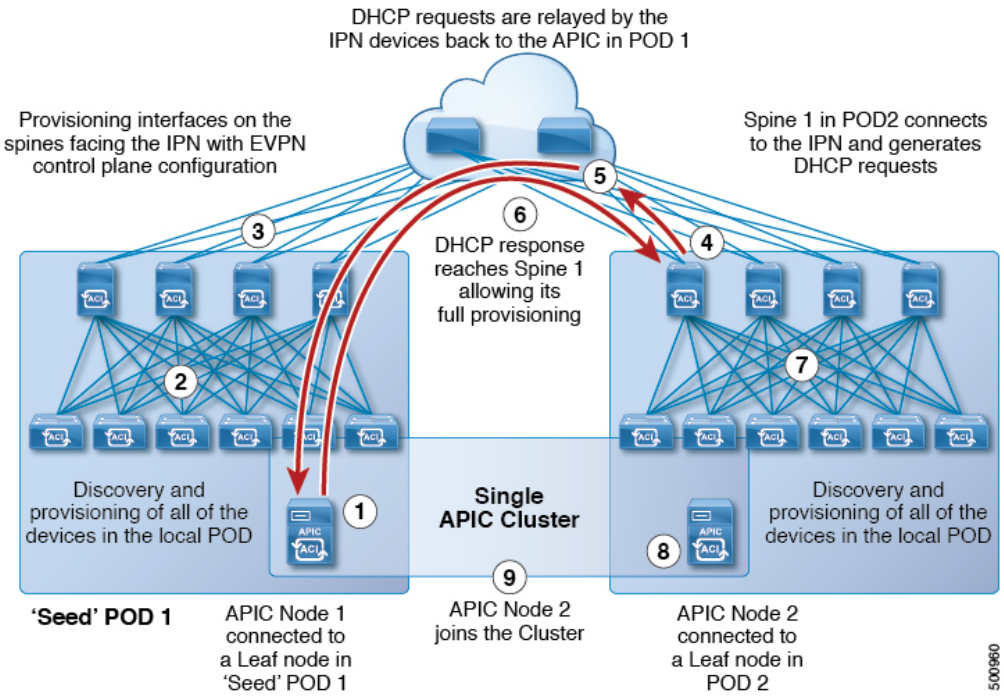
Multipod Provisioning

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 the VLAN-4 or VLAN-5 and MTU of 9150 and the associated correct IP addresses. Use VLAN-5 for the multipod interfaces/sub-interfaces, if any of the pods have connections to remote leaf switches.
- 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 Bidir group range (default is 225.0.0.0/8).
- Add 239.255.255.240/28 as PIM bidir group range.

- Enable PIM and IGMP on the interface connected to all spines.

Figure 75: Multipod Provisioning



The multipod discovery process follows the following sequence:

1. APIC1 connected to POD1 starts the discovery process.

2. Spine and leaf switches in the POD that are directly connected to the APIC1 are discovered same way as the single pod fabric discovery.

3. APIC1 pushes the L3out policies to the spines in POD1. The spine L3out policy provisions the IPN connected interfaces on spines and IP connectivity to the IPN is established.

4. POD2 spine sends DHCP request to the IPN.

5. The IPN relays the DHCP request to the APIC.

6. The APIC sends DHCP response with the sub-interface IP from the spine L3Out configuration. Upon receiving the DHCP response, the spine configures the IP address on the IPN interface, creates the static rout to the APIC using the relay address in the DHCP response as the gateway address, downloads the L3Out configuration from the spine which enables OSPG, removes the APIC static route, configures the infra DHCP relay, enables the DHCP client for all fabric and spine L3Out ports, and then the spine comes up according to the normal bringup sequence.

7. All other nodes in POD2 come up as usual.

8. The APIC controller in POD2 is discovered as usual.

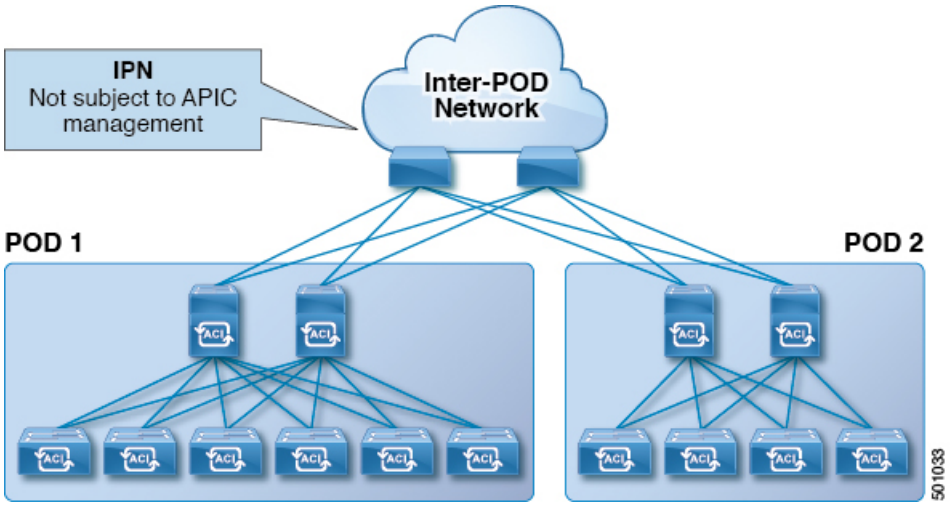
9. The APIC controller in POD2 joins the APIC cluster.

Multi-Pod QoS and DSCP Translation Policy

When traffic is sent and received within the Cisco ACI fabric, the QoS Level is determined based on the CoS value of the VXLAN packet's outer header. In Multi-Pod topologies, where devices that are not under Cisco APIC's management may modify the CoS values in the transiting packets, you can preserve the QoS Level setting by creating a mapping between the Cisco ACI and the DSCP value within the packet.

If you are not concerned with preserving the QoS settings in the IPN traffic between pods, but would like to preserve the original CoS values of the packets ingressing and egressing the fabric, see Class of Service (CoS) Preservation for Ingress and Egress Traffic, on page 176 instead.

Figure 76: Multi-Pod Topology



As illustrated in this figure, traffic between pods in a Multi-Pod topology passes through an IPN, which may contain devices that are not under Cisco APIC's management. When a network packet is sent from a spine or a leaf switch in POD1, the devices in the IPN may modify the 802.1p value in the packet. In this case, when the frame reaches a spine or a leaf switch in POD2, it would have an 802.1p value that was assigned by the IPN device, instead of the Cisco ACI QoS Level value assigned at the source in POD1.

In order to preserve the proper QoS Level of the packet and avoid high priority packets from being delayed or dropped, you can use a DSCP translation policy for traffic that goes between multiple PODs connected by an IPN. When a DSCP translation policy is enabled, Cisco APIC converts the QoS Level value (represented by the CoS value of the VXLAN packet) to a DSCP value according to the mapping rules you specify. When a packet sent from POD1 reaches POD2, the mapped DSCP value is translated back into the original CoS value for the appropriate QoS Level.

About Anycast Services

Anycast services are supported in the Cisco ACI fabric. A typical use case is to support Cisco Adaptive Security Appliance (ASA) firewalls in the pods of a multipod fabric, but Anycast could be used to enable other services, such as DNS servers or printing services. In the ASA use case, a firewall is installed in every pod and Anycast is enabled, so the firewall can be offered as an Anycast service. One instance of a firewall going down does not affect clients, as the requests are routed to the next, nearest instance available. You install ASA firewalls in each pod, then enable Anycast and configure the IP address and MAC addresses to be used.

APIC deploys the configuration of the Anycast MAC and IP addresses to the leaf switches where the VRF is deployed or where there is a contract to allow an Anycast EPG.

Initially, each leaf switch installs the Anycast MAC and IP addresses as a proxy route to the spine switch. When the first packet from the Anycast Service is received, the destination information for the service is installed on the leaf switch behind which the service is installed. All other leaf switches continue to point to the spine proxy. When the Anycast service has been learned, located behind a leaf in a pod, COOP installs the entry on the spine switch to point to the service that is local to the pod.

When the Anycast service is running in one pod, the spine receives the route information for the Anycast service present in the pod through BGP-EVPN. If the Anycast service is already locally present, then COOP caches the Anycast service information of the remote pod. This route through the remote pod is only installed when the local instance of the service goes down.

Remote Leaf Switches

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 switch 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.

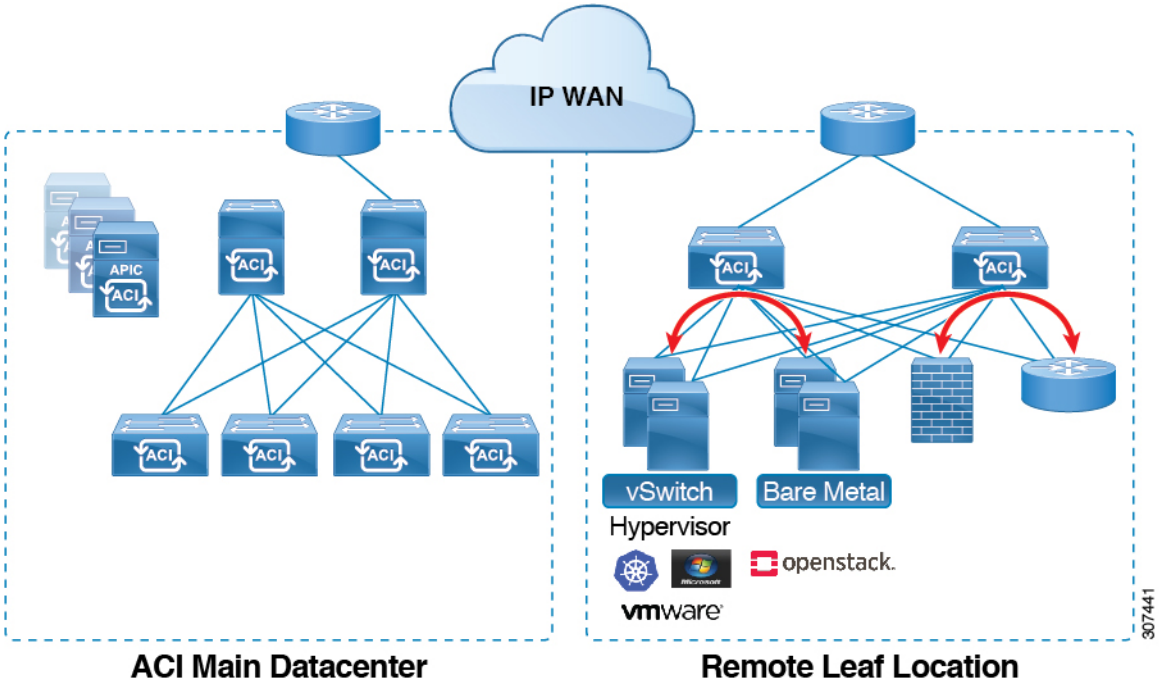
- Shared Services: Packets do not use spine-proxy path reducing WAN bandwidth consumption.
- Inter-VRF traffic is forwarded through an upstream router and not placed on the spine.
- This enhancement is only applicable for a remote leaf vPC pair. For communication across remote leaf pairs, a spine proxy 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)

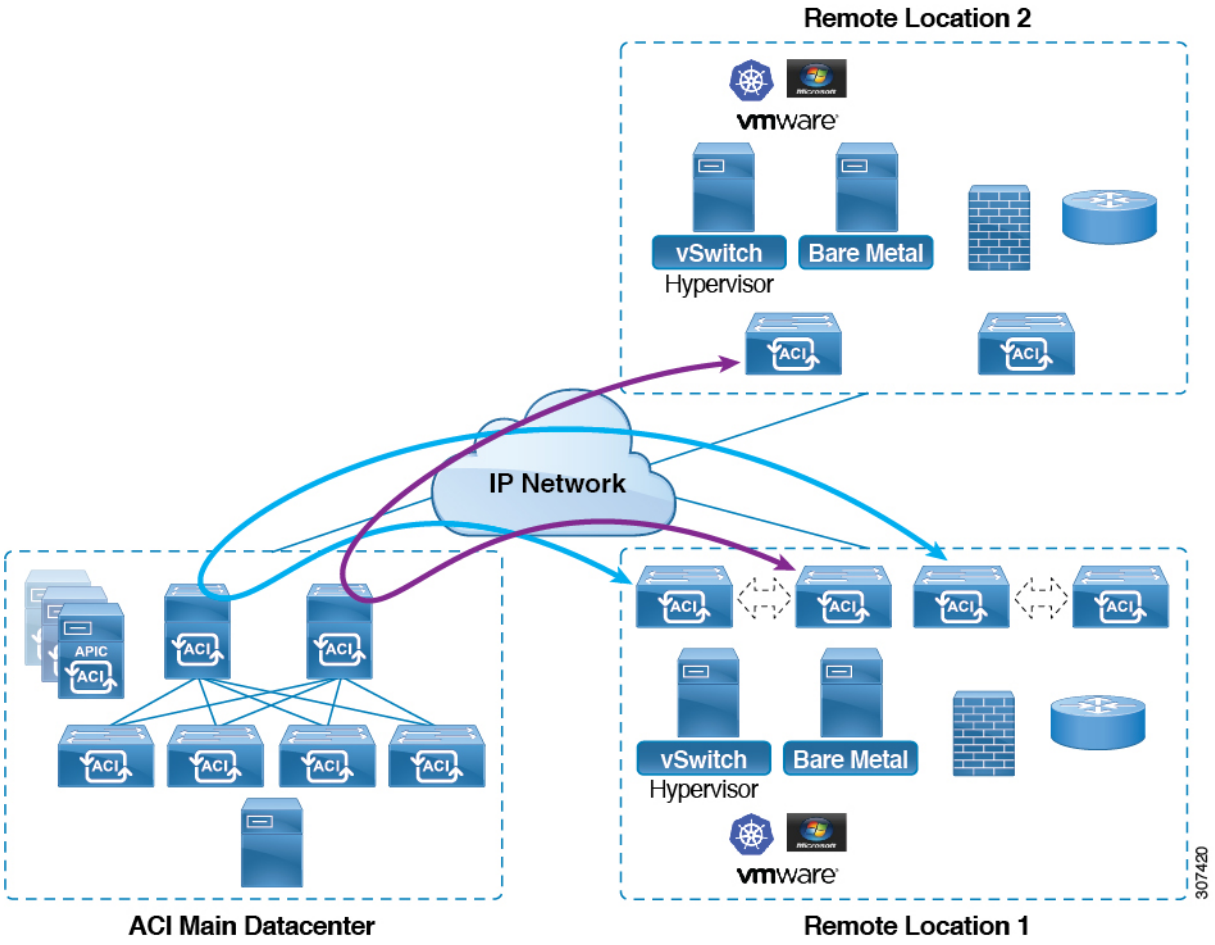
Before Release 4.1(2), all local switching (within the remote leaf vPC peer) traffic on the remote leaf location is switched directly between endpoints, whether physical or virtual, as shown in the following figure.

Figure 77: Local Switching Traffic: Prior to Release 4.1(2)



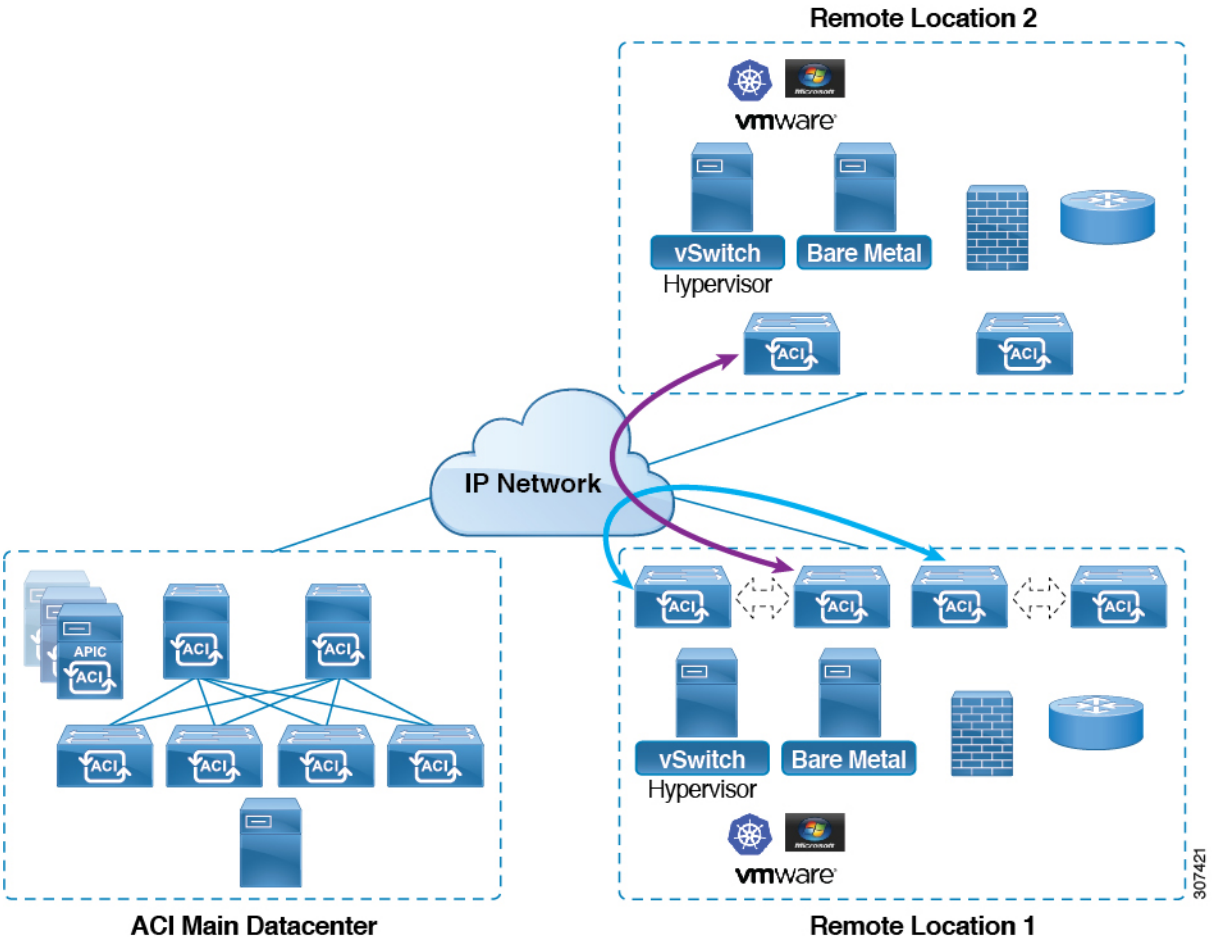
In addition, before Release 4.1(2), traffic between the remote leaf switch vPC pairs, either within a remote location or between remote locations, is forwarded to the spine switches in the ACI main data center pod, as shown in the following figure.

Figure 78: 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 79: 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.

- 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 sourced from one remote leaf switch and destined to another remote leaf switch (which is not the vPC peer of the source), there will be a milli-second traffic loss every time the remote end point (XR EP) timeout of 300 seconds occurs.
- With a remote leaf switches with ACI Multi-Site configuration, all traffic continues from the remote leaf switch to the other pods and remote locations, even with a spine switch failure, because traffic will flow through an alternate available pod in this situation.

10 Mbps Bandwidth Support in IPN for Remote Leaf Switches

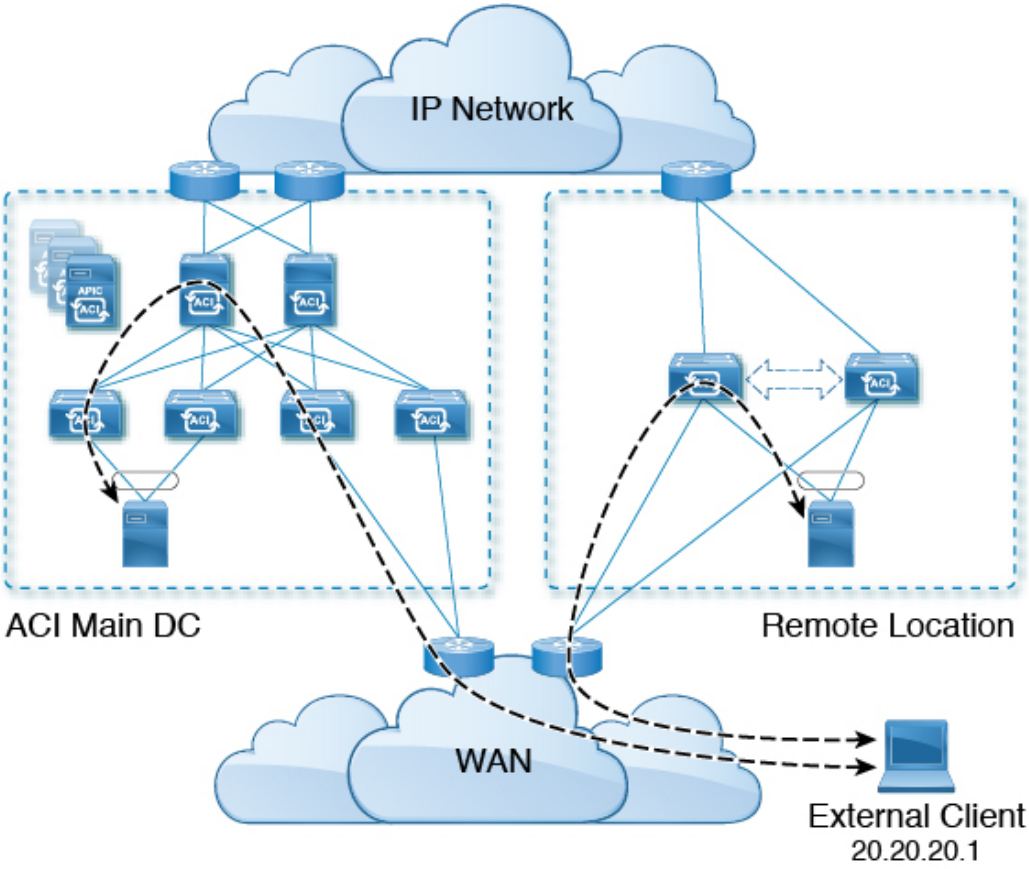
You might have situations where most of the data traffic from the remote leaf switches is local and the Inter-Pod Network (IPN) is needed only for management purposes. In these situations, you may not need a 100 Mbps IPN. To support these environments, starting with Release 4.2(4), support is now available for 10 Mbps as a minimum bandwidth in the IPN.

To support this, the following requirements should be met:

- The IPN path is only used for managing remote leaf switches (management functions such as upgrades and downgrades, discovery, COOP, and policy pushes).
- Configure IPN with the QoS configuration in order to prioritize control and management plane traffic between the Cisco ACI datacenter and remote leaf switch pairs based on the information provided in the section "Creating DSCP Translation Policy Using Cisco APIC GUI".
- All traffic from the Cisco ACI datacenter and remote leaf switches is through the local L3Out.
- The EPG or bridge domain are not stretched between the remote leaf switch and the ACI main datacenter.
- You should pre-download software images on the remote leaf switches to reduce upgrade time.

The following figure shows a graphical representation of this feature.

Figure 80: Remote Leaf Switch Behavior, Release 4.2(4): Remote Leaf Switch Management through IPN

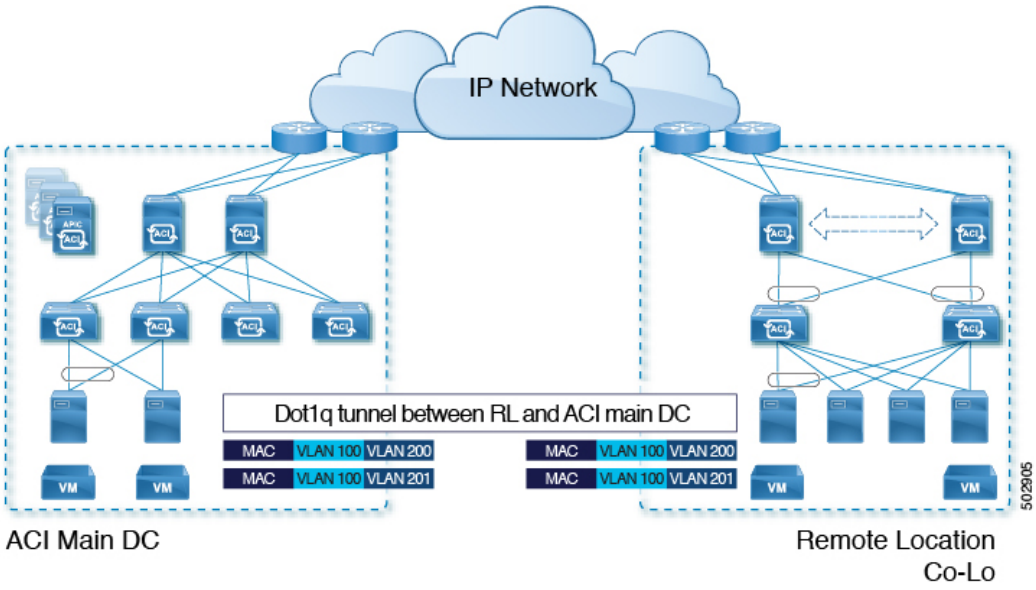


Dot1q Tunnel Support on Remote Leaf Switches

In some situations, a co-location provider might be hosting multiple customers, where each customer is using thousands of VLANs per remote leaf switch pair. Starting with Release 4.2(4), support is available to create an 802.1Q tunnel between the remote leaf switch and the ACI main datacenter, which provides the flexibility to map multiple VLANs into a single 802.1Q tunnel, thereby reducing the EPG scale requirement.

The following figure shows a graphical representation of this feature.

Figure 81: Remote Leaf Switch Behavior, Release 4.2(4): 802.1Q Tunnel Support on Remote Leaf Switches



Create this 802.1Q tunnel between the remote leaf switch and the ACI main datacenter using the instructions provided in the "802.1Q Tunnels" chapter in the *Cisco APIC Layer 2 Networking Configuration Guide*, located in the Cisco APIC documentation landing page.

You can configure remote leaf switches in the APIC GUI, either with and without a wizard, or use the REST API or the NX-OS style CLI.

Remote Leaf Switch Restrictions and Limitations

The following guidelines and restrictions apply to remote leaf switches:

- 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 173
- Unsupported Features, on page 174
- Changes For Release 5.0(1), on page 176

Supported Features

Stretching of an L3Out SVI within a vPC remote leaf switch pair is supported.

Beginning with Cisco APIC release 4.2(4), the 802.1Q (Dot1q) tunnels feature is supported.

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)
- 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.

Changes For Release 5.0(1)

Beginning with Cisco APIC release 5.0(1), the following changes have been applied for remote leaf switches:

- The direct traffic forwarding feature is enabled by default and cannot be disabled.
- A configuration without direct traffic forwarding for remote leaf switches is no longer supported. If you have remote leaf switches and you are upgrading to Cisco APIC release 5.0(1), review the information provided in the section "About Direct Traffic Forwarding" and enable direct traffic forwarding using the instructions in that section.

QoS

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.

Class of Service (CoS) Preservation for Ingress and Egress Traffic

When traffic enters the Cisco ACI fabric, each packet's priority is mapped to a Cisco ACI QoS level. These QoS levels are then stored in the CoS field and DE bit of the packet's outer header while the original headers are discarded.

If you want to preserve the original CoS values of the ingressing packets and restore it when the packet leaves the fabric, you can enable the 802.1p Class of Service (CoS) preservation using a global fabric QoS policy as described in this section.

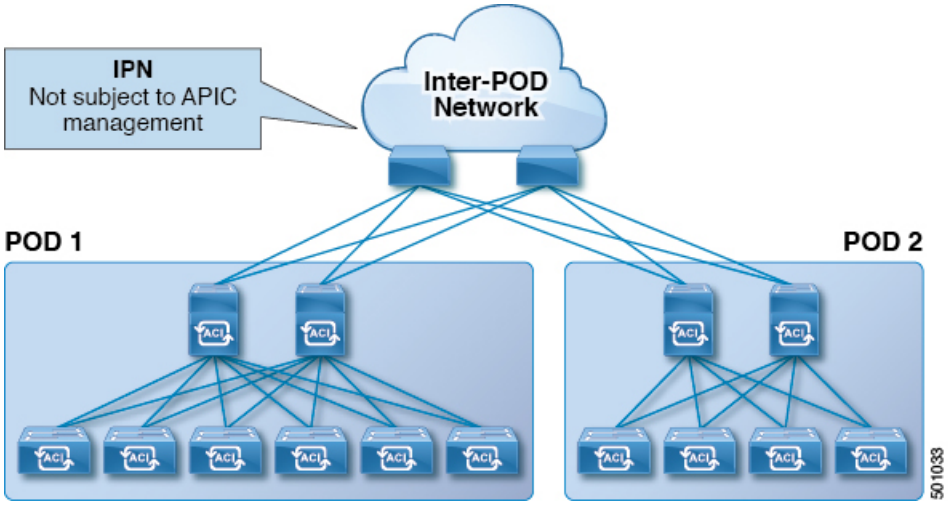
The CoS preservation is supported in single pod and multipod topologies, however in multipod topologies, CoS preservation can be used only when you are not concerned with preserving the settings in the IPN between pods. To preserve the CoS values of the packets as they are transiting the IPN, use the DSCP translation policy as described in Multi-Pod QoS and DSCP Translation Policy, on page 166.

Multi-Pod QoS and DSCP Translation Policy

When traffic is sent and received within the Cisco ACI fabric, the QoS Level is determined based on the CoS value of the VXLAN packet's outer header. In Multi-Pod topologies, where devices that are not under Cisco APIC's management may modify the CoS values in the transiting packets, you can preserve the QoS Level setting by creating a mapping between the Cisco ACI and the DSCP value within the packet.

If you are not concerned with preserving the QoS settings in the IPN traffic between pods, but would like to preserve the original CoS values of the packets ingressing and egressing the fabric, see Class of Service (CoS) Preservation for Ingress and Egress Traffic, on page 176 instead.

Figure 82: Multi-Pod Topology



As illustrated in this figure, traffic between pods in a Multi-Pod topology passes through an IPN, which may contain devices that are not under Cisco APIC's management. When a network packet is sent from a spine or a leaf switch in POD1, the devices in the IPN may modify the 802.1p value in the packet. In this case, when the frame reaches a spine or a leaf switch in POD2, it would have an 802.1p value that was assigned by the IPN device, instead of the Cisco ACI QoS Level value assigned at the source in POD1.

In order to preserve the proper QoS Level of the packet and avoid high priority packets from being delayed or dropped, you can use a DSCP translation policy for traffic that goes between multiple PODs connected by an IPN. When a DSCP translation policy is enabled, Cisco APIC converts the QoS Level value (represented by the CoS value of the VXLAN packet) to a DSCP value according to the mapping rules you specify. When a packet sent from POD1 reaches POD2, the mapped DSCP value is translated back into the original CoS value for the appropriate QoS Level.

Translating Ingress to Egress QoS Markings

Cisco APIC enables translating the DSCP and CoS values of the ingressing traffic to a QoS Level to be used inside the Cisco ACI fabric. Translation is supported only if the DSCP values are present in the IP packet and CoS values are present in the Ethernet frames.

For example, this functionality allows the Cisco ACI fabric to classify the traffic for devices that classify the traffic based only on the CoS value, such as Layer-2 packets, which do not have an IP header.

CoS Translation Guidelines and Limitations

You must enable the global fabric CoS preservation policy, as described in Class of Service (CoS) Preservation for Ingress and Egress Traffic, on page 176.

CoS translation is not supported on external L3 interfaces.

CoS translation is supported only if the egress frame is 802.1Q encapsulated.

CoS translation is not supported when the following configuration options are enabled:

- Contracts are configured that include QoS.
- The outgoing interface is on a FEX.
- Multipod QoS using a DSCP policy is enabled.
- Dynamic packet prioritization is enabled.
- If an EPG is configured with intra-EPG endpoint isolation enforced.
- If an EPG is configured with allow-microsegmentation enabled.

HSRP

About HSRP

HSRP is a first-hop redundancy protocol (FHRP) that allows a transparent failover of the first-hop IP router. HSRP provides first-hop routing redundancy for IP hosts on Ethernet networks configured with a default router IP address. You use HSRP in a group of routers for selecting an active router and a standby router. In a group of routers, the active router is the router that routes packets, and the standby router is the router that takes over when the active router fails or when preset conditions are met.

Many host implementations do not support any dynamic router discovery mechanisms but can be configured with a default router. Running a dynamic router discovery mechanism on every host is not practical for many reasons, including administrative overhead, processing overhead, and security issues. HSRP provides failover services to such hosts.

When you use HSRP, you configure the HSRP virtual IP address as the default router of the host (instead of the IP address of the actual router). The virtual IP address is an IPv4 or IPv6 address that is shared among a group of routers that run HSRP.

When you configure HSRP on a network segment, you provide a virtual MAC address and a virtual IP address for the HSRP group. You configure the same virtual address on each HSRP-enabled interface in the group. You also configure a unique IP address and MAC address on each interface that acts as the real address. HSRP selects one of these interfaces to be the active router. The active router receives and routes packets destined for the virtual MAC address of the group.

HSRP detects when the designated active router fails. At that point, a selected standby router assumes control of the virtual MAC and IP addresses of the HSRP group. HSRP also selects a new standby router at that time.

HSRP uses a priority designator to determine which HSRP-configured interface becomes the default active router. To configure an interface as the active router, you assign it with a priority that is higher than the priority of all the other HSRP-configured interfaces in the group. The default priority is 100, so if you configure just one interface with a higher priority, that interface becomes the default active router.

Interfaces that run HSRP send and receive multicast User Datagram Protocol (UDP)-based hello messages to detect a failure and to designate active and standby routers. When the active router fails to send a hello message within a configurable period of time, the standby router with the highest priority becomes the active router. The transition of packet forwarding functions between the active and standby router is completely transparent to all hosts on the network.

You can configure multiple HSRP groups on an interface. The virtual router does not physically exist but represents the common default router for interfaces that are configured to provide backup to each other. You do not need to configure the hosts on the LAN with the IP address of the active router. Instead, you configure

them with the IP address of the virtual router (virtual IP address) as their default router. If the active router fails to send a hello message within the configurable period of time, the standby router takes over, responds to the virtual addresses, and becomes the active router, assuming the active router duties. From the host perspective, the virtual router remains the same.

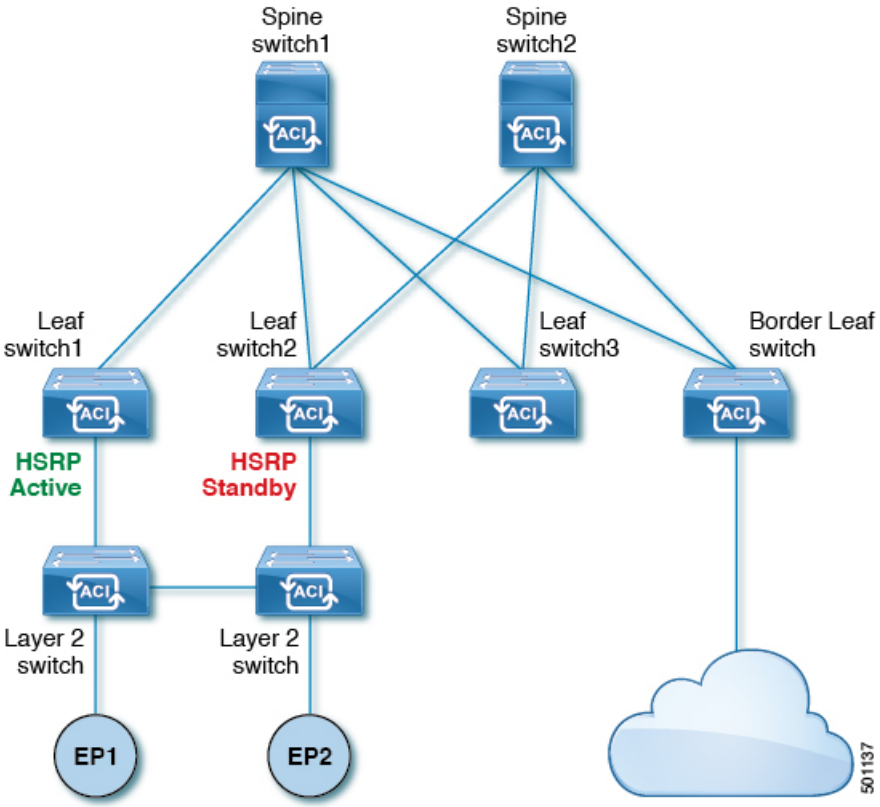
Note Packets received on a routed port destined for the HSRP virtual IP address terminate on the local router, regardless of whether that router is the active HSRP router or the standby HSRP router. This process includes ping and Telnet traffic. Packets received on a Layer 2 (VLAN) interface destined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

About Cisco APIC and HSRP

HSRP in Cisco ACI is supported only on routed-interface or sub-interface. Therefore HSRP can only be configured under Layer 3 Out. Also there must be Layer 2 connectivity provided by external device(s) such as a Layer 2 switch between ACI leaf switches running HSRP because HSRP operates on leaf switches by exchanging Hello messages over external Layer 2 connections. An HSRP hello message does not pass through the spine switch.

The following is an example topology of an HSRP deployment in Cisco APIC.

Figure 83: HSRP Deployment Topology



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

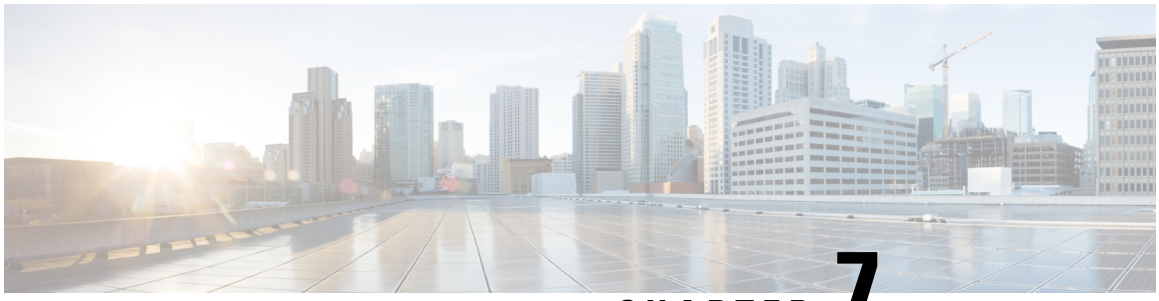
sub-interfaces must be modified to prevent MAC address flapping on the Layer 2 external device. This is because Cisco APIC assigns the same MAC address (00:22:BD:F8:19:FF) to every logical interface under the interface logical profiles.

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.



CHAPTER 7

ACI Transit Routing, Route Peering, and EIGRP Support

This chapter contains the following sections:

ACI Transit Routing

The ACI fabric supports transit routing, which enables border routers to perform bidirectional redistribution with other routing domains. Unlike the stub routing domains of earlier ACI releases, that block transit redistribution, bidirectional redistribution passes routing information from one routing domain to another. Such redistribution lets the ACI fabric provide full IP connectivity between different routing domains. Doing so can also provide redundant connectivity by enabling backup paths between routing domains.

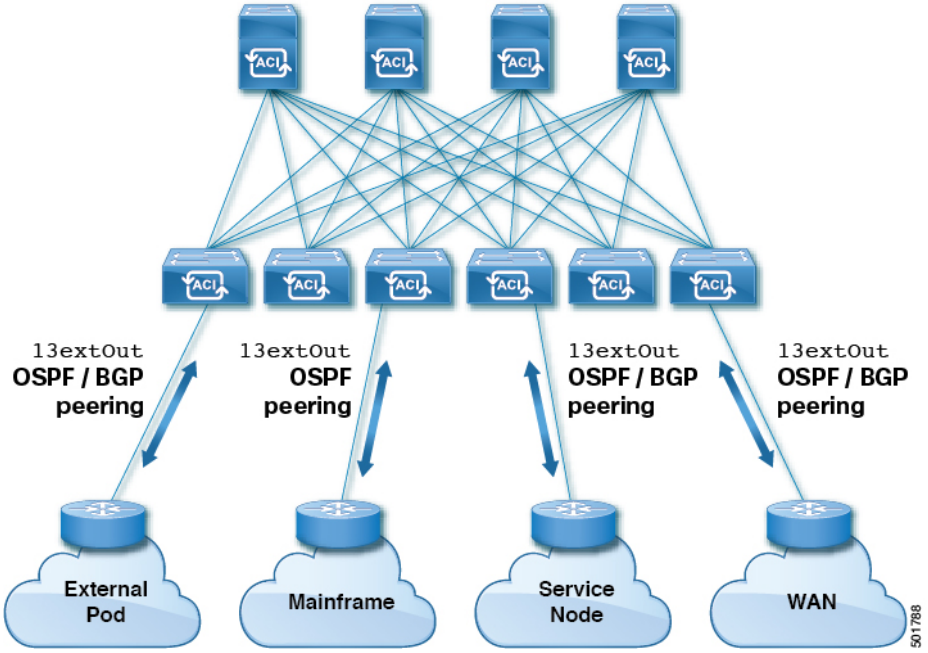
Design transit redistribution policies that avoid suboptimal routing or the more serious problem of routing loops. Typically, transit redistribution does not preserve the original topology and link-state information and redistributes external routes in distance-vector fashion (routes are advertised as vector prefixes and associated distances even with link-state protocols). Under these circumstances, the routers can inadvertently form routing loops that fail to deliver packets to their destination.

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.

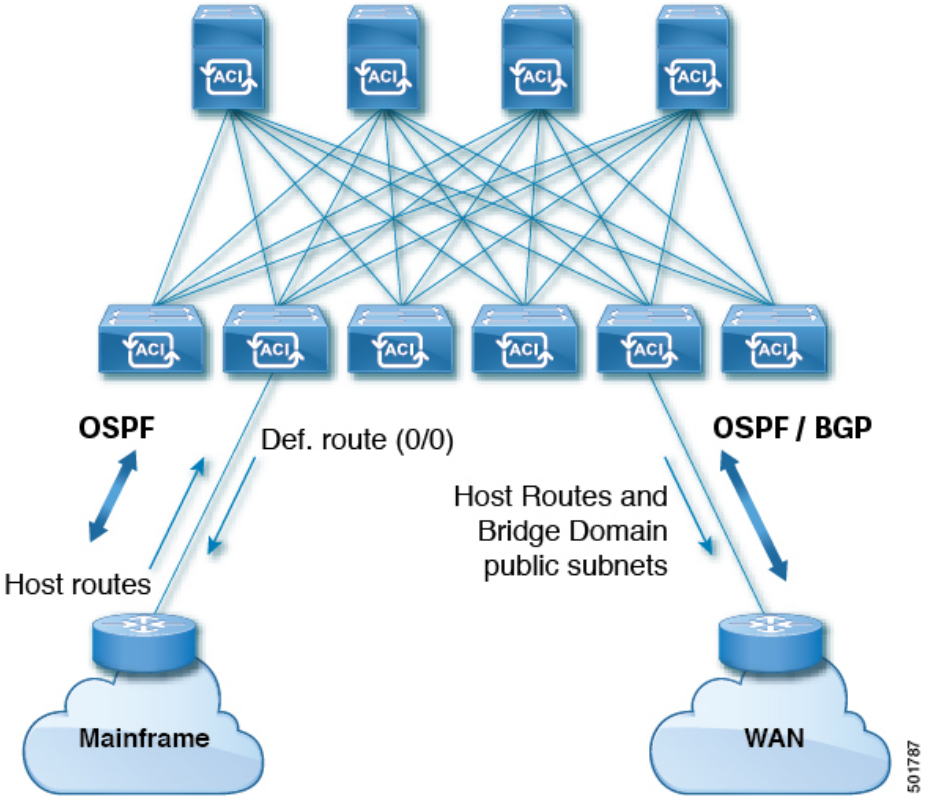
Figure 84: Transit Routing Between Layer 3 Domains



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 85: Mainframe Transit Connectivity

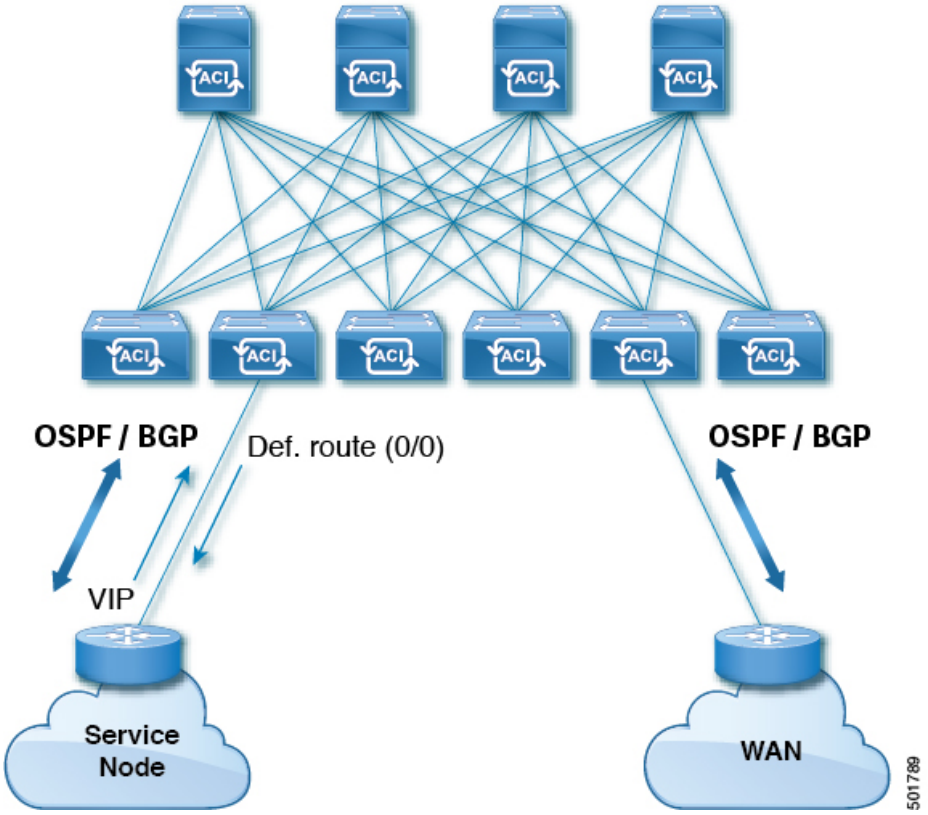


In this topology, mainframes require the ACI fabric to be a transit domain for external connectivity through a WAN router and for east-west traffic within the fabric. They push host routes to the fabric to be redistributed within the fabric and out to external interfaces.

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 86: Service Node Transit Connectivity

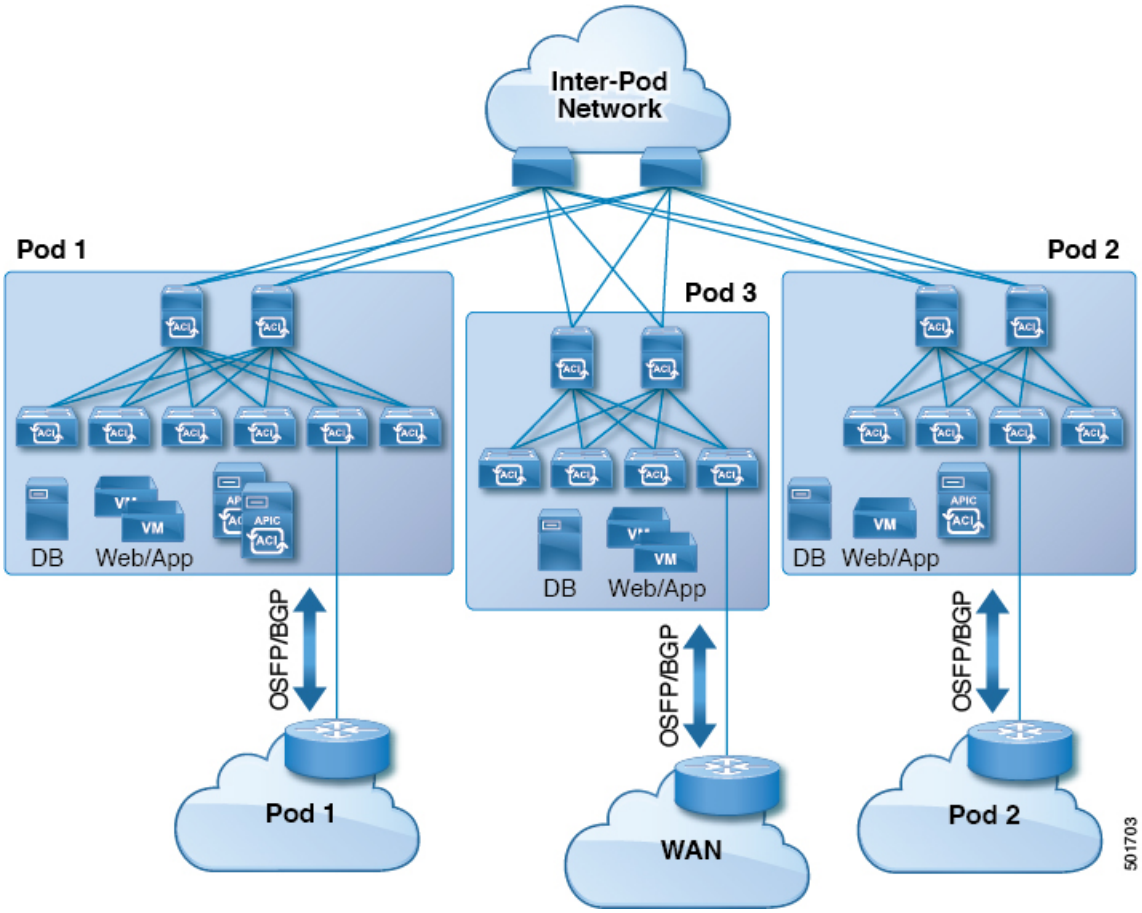


The VIP is the external facing IP address for a particular site or service. A VIP is tied to one or more servers or nodes behind a service node.

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 87: Multiple Pods with L3Outs in a Transit-Routed Configuration



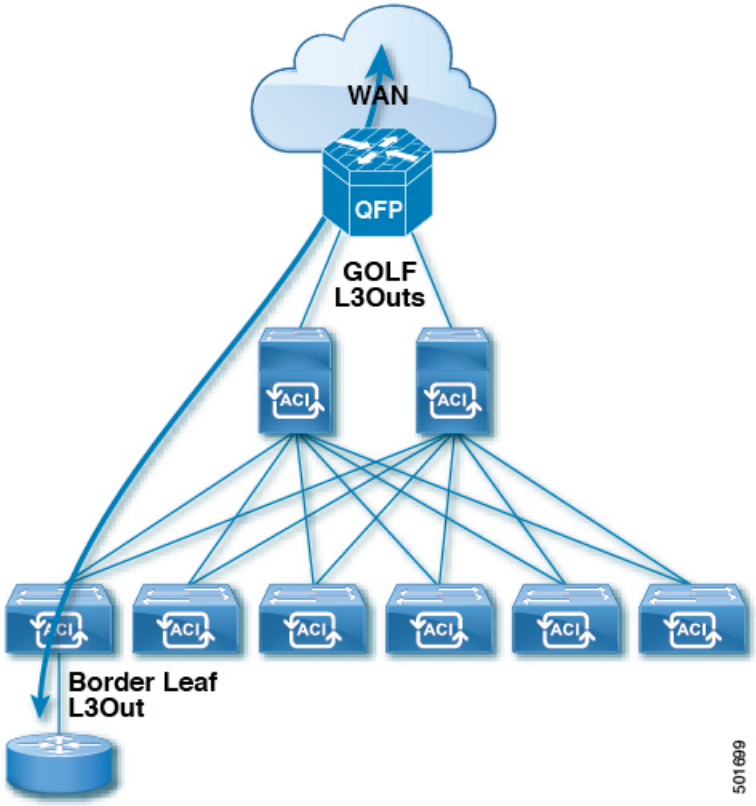
In such scenarios, the policies are administered at the demarcation points and ACI policies need not be imposed.

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 88: GOLF L3Outs and a Border Leaf L3Out in a Transit-Routed Configuration



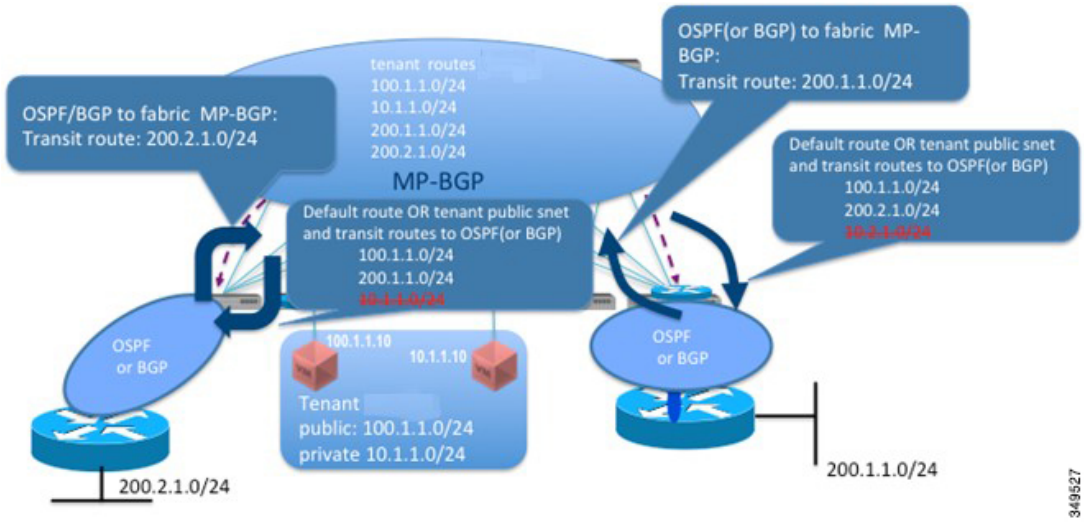
ACI Fabric Route Peering

Layer 3 connectivity and peering with the fabric is configured using a Layer 3 external outside network (`l3extOut`) interface. The peering protocol configuration along with route redistribution and inbound/outbound-filtering rules is associated with an `l3extOut`. The ACI fabric does not appear as a giant router to the external peers, but rather as a transit between separate Layer 3 domains. The peering considerations on one `l3extOut` need not affect the peering considerations on other `l3extOut` policies. The ACI fabric uses MP-BGP for distributing external routes inside the fabric.

Route Redistribution

Inbound routes from external peers are redistributed into the ACI fabric using MP-BGP, subject to inbound filtering rules. These can be transit routes or external routes in the case of WAN connectivity. MP-BGP distributes the routes to all the leaves (including other border leaves) where the tenant is deployed.

Figure 89: Route Redistribution



Inbound route filtering rules select a subset of routes advertised by the external peers to the fabric on the `l3extOut` interfaces. The import filter route-map is generated by using the prefixes in the prefix based EPG. The import filter list is associated only with MP-BGP to restrict the prefixes distributed into the fabric. Set actions can also be associated with import route-maps.

In the outbound direction, an administrator has the option to advertise default routes or transit routes and bridge domain public subnets. If default route advertisement is not enabled, outbound route filtering selectively advertises routes as configured by the administrator.

Currently, route-maps are created with a prefix-list on a per-tenant basis to indicate the bridge domain public subnets to be advertised to external routers. In addition, a prefix-list has to be created to allow all transit routes to be advertised to an external router. The prefix-list for transit routes are configured by an administrator. The default behavior is to deny all transit route advertisement to an external router.

The following options are available for the route-maps associated with transit routes:

- *Permit-all*: Allow all transit routes to be redistributed and advertised outside.
- *Match prefix-list*: Only a subset of transit routes are redistributed and advertised outside.
- *Match prefix-list* and *set action*: A set action can be associated with a subset of transit routes to tag routes with a particular attribute.

The bridge domain public subnets and transit route prefixes can be different prefix-lists but combined into a single route-map with different sequence numbers. Transit routes and bridge domain public subnets are not expected to have the same prefixes, so prefix-list matches are mutually exclusive.

Route Peering by Protocol

Route peering can be configured per protocol when combining BGP and OSPF with static routes.

OSPF	BGP
Various host types require OSPF to enable connectivity and provide redundancy. These include mainframes, external pods, and service nodes that use ACI as a layer-3 transit within the fabric and to the WAN. Such external devices peer with the fabric through a nonborder leaf running OSPF. Ideally, the OSPF areas are configured as a Not-So-Stubby Area (NSSA) or totally stub area to enable them to receive a default route, so they do not participate in full area routing. For existing deployments where the administrator prefers not to change routing configurations, a stub area configuration is not mandated. Two fabric leaf switches do not establish OSPF adjacency with each other unless they share the same external SVI interface.	External pods and service nodes can use BGP peering with the fabric. BGP peers are associated with an `l3extOut` and multiple BGP peers can be configured per `l3extOut`. BGP peer reachability can be through OSPF, EIGRP, connected interfaces, static routes, or loopback. iBGP or eBGP is used for peering with external routers. BGP route attributes from the external router are preserved since MP-BGP is used for distributing the external routes in the fabric. A configuration that contains a match for both transitive and nontransitive BGP extended communities with the same value is not supported; the APIC rejects this configuration.
OSPF Route Redistribution The default-information originate policy in OSPF generates a default route to an external router. Enabling the policy is recommended when peering with mainframes, external pods, and service nodes. When the default-information originate policy is not enabled, configure redistribute-static and redistribute-BGP in the OSPF domain to advertise static bridge domain (BD) public subnets and transit routes respectively. Associate a route-map with the redistribution policy for outbound filtering. It is recommended to not enable the default-information originate option, when peering with external WAN routers. In the inbound direction, OSPF routes are redistributed into the ACI fabric using MP-BGP.	**BGP Route Redistribution** In the outbound direction, a default route is generated by BGP on a per-peer basis by the default-originate policy. The default route is injected to the peer by BGP even in the absence of a default route in the local routing table. If a default-originate policy is not configured, then static redistribution is enabled for bridge domain public subnets. Transit routes from MP-BGP are available to BGP for advertising. These routes are conditionally advertised outside, subject to outbound filtering policies. In the inbound direction, the advertised routes are available to MP-BGP for redistribution in the fabric, subject to inbound filtering rules. If BGP is used for external peering, then all the BGP attributes of the route are preserved across the fabric.

OSPF	BGP
OSPF Route filtering You can configure OSPF to limit the number of Link-State Advertisements (LSAs) accepted from an external peer to avoid over consumption of the route table, caused by a rogue external router. Inbound route filtering is supported for Layer 3 external outside tenant networks using OSPF. It is applied using a route-map in the in-direction, to filter transit routes allowed in the fabric. In the outbound direction, configure redistribute-static and redistribute-BGP at the OSPF domain level. Configure a route-map to filter the bridge domain public subnets and transit routes. Optionally, some prefixes in the route-map can also be configured with a set action to add route tags. Inter-area prefixes are also filtered using the outbound filter list and associating it with an OSPF area.	**BGP Route filtering** Inbound route filtering in BGP is applied using a route-map on a per-peer basis. A route-map is configured at the *peer-af* level in the in-direction, to filter the transit routes to be allowed in the fabric. In the outbound direction, static routes are redistributed into BGP at the *dom-af* level. Transit routes from MP-BGP are available to the external BGP peering sessions. A route-map is configured at the *peer-af* level in the out-direction to allow only public subnets and selected transit routes outside. Optionally, a set action to advertise a community value for selected prefixes is configured on the route-map. The bridge domain public subnets and transit route prefixes can be different prefix-lists but combined into a single route-map at the *peer-af* level with different sequence numbers.
OSPF Name Lookup, Prefix Suppression, and Type 7 Translation OSPF can be configured to enable name lookup for router IDs and suppress prefixes. The APIC system performs OSPF Forwarding Address Suppression in the Translated Type-5 LSAs feature, which causes an NSSA ABR to translate Type-7 LSAs to Type-5 LSAs. To avoid this, use the 0.0.0.0 subnet as the forwarding address instead of the one that is specified in the Type-7 LSA. This feature causes routers that are configured not to advertise forwarding addresses into the backbone to direct forwarded traffic to the translating NSSA ASBRs.	**BGP Dynamic Neighbor Support and Private AS Control** Instead of providing a specific neighbor address, a dynamic neighbor range of addresses can be provided. Private Autonomous System (AS) numbers can be in the range from 64512 to 65535. They cannot be advertised to a global BGP table. Private AS numbers can be removed from the AS path on a per peer basis and can only be used for eBGP peers according to the following options: • `Remove Private AS` – Remove if the AS path only has private AS numbers. • `Remove All` – Remove if the AS path has both private and public AS numbers. • `Replace AS` – Replace the private AS with the local AS number. **Note** `Remove all` and `replace AS` can only be set if `remove private AS` is set.

BGP dampening minimizes propagation into the fabric of flapping e-BGP routes that were received from external routers that are connected to border leaf switches (BLs). Frequently flapping routes from external routers are suppressed on BLs based on criteria you configure. They are then prohibited from redistribution to iBGP peers (ACI spine switches). Suppressed routes are reused after a configured time. Each flap penalizes the e-BGP route with a penalty of 1000. When the flap penalty reaches a defined suppress-limit threshold

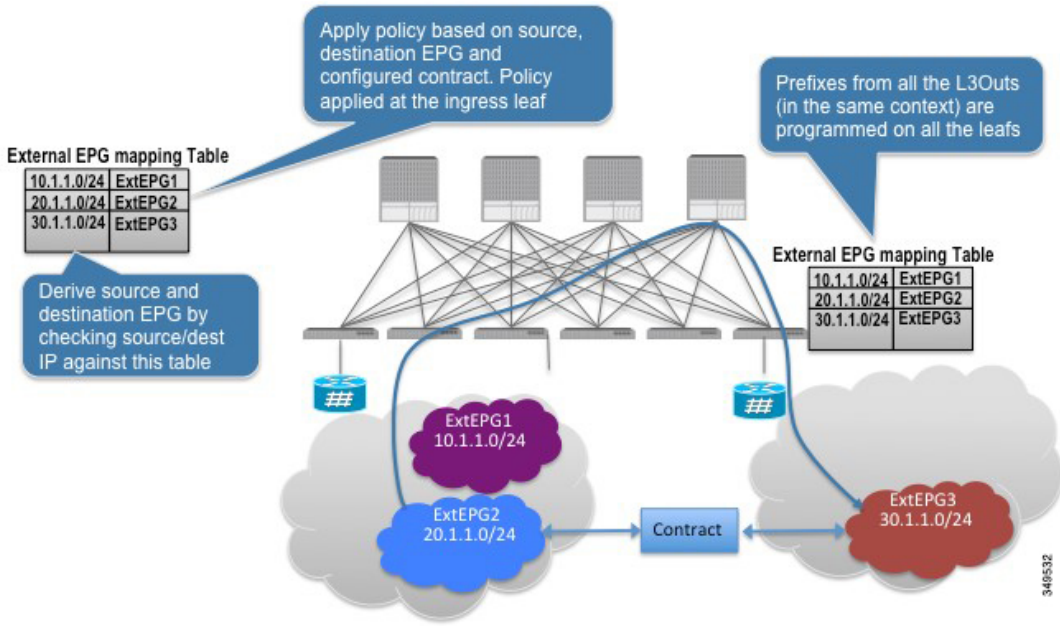
(default 2000), the e-BGP route is marked as dampened. Dampened routes are not advertised to other BGP peers. The penalty is decremented to half after every half-life interval (the default is 15 minutes). A dampened route is reused if the penalty falls below a specified reuse-limit (the default is 750). A dampened route is suppressed at most for a specified maximum suppress time (maximum of 45 minutes).

Use the BGP weight attribute to select a best path. The weight (from 0 to 65,535) is assigned locally to a specific router. The value is not propagated or carried through any of the route updates. By default, paths that the router originates have a weight of 32,768, and other paths have a weight of 0. Routes with a higher weight value have preference when there are multiple routes to the same destination. Set the weight under the BGP neighbor or under the route map.

BGP peering is typically configured to the neighbor's loopback address. In such cases, loopback reachability is statically configured or advertised (more commonly) through OSPF. The loopback interface is configured as a passive interface and added into the OSPF area. There are no redistribution policies that are attached to OSPF. The route redistribution implementation is through BGP. Route filtering can be configured in L3Outs for tenant networks that use either BGP or OSPF.

External routes can also be programmed as static routes on the border leaf in the respective tenants. A peering protocol is not required if the external routes are programmed as static routes on the border leaf. External static routes are redistributed to other leaf switches in the fabric through MP-BGP, subject to import filtering. Starting with release 1.2(1x), static route preference incoming within the ACI fabric is carried in MP-BGP using a cost extended community. On an L3Out connection, an MP-BGP route coming from Layer 4 wins over a local static route. A route is installed in the Unicast Routing Information Base (URIB) with the preference that is specified by an administrator. On an ACI nonborder leaf switch, a route is installed with Layer 4 as nexthop. When nexthop on Layer 4 is not available, the Layer 3 static route becomes the best route in fabric.

Figure 90: Static Route Policy Model for Transit



For `l3extOut` connections, external endpoints are mapped to an external EPG based on IP prefixes. For each `l3extOut` connection, you can create one or more external EPGs, based on whether different endpoints require different policies.

Each external EPG is associated with a class-id. Each prefix in the external EPG is programmed in the hardware to derive the corresponding class-id. The prefixes are only qualified VRF instance and not by the `l3extOut` interface on which they are deployed.

The union of prefixes from all the `l3extOut` policies in the same VRF is programmed on all the leaf switches where the `l3extOut` policies are deployed. The source and destination class-ids corresponding to the source and destination IP address in a packet are derived at the ingress leaf and the policy is applied on the ingress leaf itself, based on the configured contract. If a contract allows traffic between two prefixes on two L3Out interfaces, then packets with any combination of the source and destination IP address (belonging to the configured prefixes) is allowed between the L3Out interfaces. If there is no contract between the EPGs, the traffic is dropped at the ingress leaf.

Since prefixes are programmed on every leaf switch where `l3extOut` policies are deployed, the total number of prefixes APIC supports for prefix-based EPGs is limited to 1000 for the fabric.

Overlapping or equal subnets cannot be configured on different `l3extOut` interfaces in the same VRF. If overlapping or equal subnets are required, then a single `l3extOut` is used for transit with appropriate export prefixes.

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.

Default Policy Behavior

When there are no contracts between two prefix-based EPGs, traffic between unknown-source and unknown-destination prefixes are dropped. These drops are achieved by implicitly programming different class-ids for unknown source and destination prefixes. Since the class-ids are different, they are impacted by the class-unequal rule and packets are denied. The class-unequal drop rule also causes packets to be dropped from known source and destination IP addresses to unknown source and destination IP addresses and vice versa.

Due to this change in the default behavior, the class-id programming for catch-all (0/0) entries has been changed as illustrated in the example below:

- Unknown source IP address is EPG1.
- Unknown destination IP address is EPG2.
- Unknown source IP <—> Unknown destination IP => class-unequal rule => DROP.
- User-configured default prefixes (0/0) = EPG3 and (10/8) = EPG4. Contract between EPG3 and EPG4 is set to ALLOW.
- Programmed rules:
  - EPG1 <—> EPG4 => class-unequal rule => DROP
  - EPG4 <—> EPG2 => class-unequal rule => DROP

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)

EIGRP L3extOut Configuration

EIGRP is the main protocol used for advertising the fabric public subnets, connect routes, static routes, and transit routes configured on the leaf switches.

There is an enable/disable flag for EIGRP for a given Layer 3 external outside network (`l3extOut`) routed domain.

Note The autonomous system number that is a tag used for EIGRP and is not the same as the fabric ASN used by BGP.

EIGRP cannot be enabled with BGP and OSPF on the same `L3extOut`.

The following EIGRP transit scenarios are supported:

- EIGRP running in an `L3extOut` on one node and OSPF running in another `L3extOut` on a different node.

Note Multiple EIGRP `L3extOut`s are not supported on the same node in the same Virtual Routing and Forwarding (VRF).

- EIGRP to static route transit.

EIGRP Interface Profile

To enable EIGRP on an interface, an EIGRP profile needs to be configured under the interface profile in the `L3extOut`->Node->Interface hierarchy. An EIGRP profile has a relation to an EIGRP interface policy enabled in the tenant. In the absence of a relation or interface policy in the tenant, the default EIGRP interface policy in the `common` tenant is used. EIGRP is enabled on all interfaces contained in the interface profile. This includes L3-Port, Sub-interfaces, External SVI on ports, port-channels, and VPCs contained in the interface profile.

Route-map infrastructure and settings in the policy model are common across all the protocols. The route-map set actions are a superset of actions to cover BGP, OSPF, and EIGRP. The EIGRP protocol supports the *set tag* option in route maps used for interleak/redistribution. These route maps are configured on a per-VRF basis. If the `L3extOut` has both IPv4 and IPv6 interfaces, then an interleak policy is applied on both IPv4 and IPv6 address families for that VRF.

Note At this time, VRF-level route maps are supported, but interface route maps are not supported.

The default route leak policy on the `L3extOut` is protocol agnostic in terms of the configuration. Properties enabled in the default route leak policy are a superset of the individual protocols. Supported configurations in the default route leak are as follows:

- *Scope*: VRF is the only scope supported for EIGRP.
- **`Always`**: The switch advertises the default route only if present in the routing table or advertises it regardless.
- *Criteria*: only or in-addition. With the only option, only the default route is advertised by EIGRP. With in-addition, the public subnets and transit routes are advertised along with the default route.

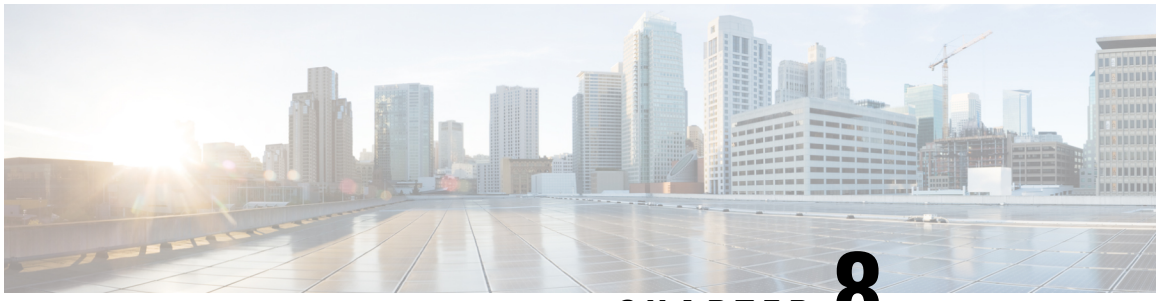
The default route leak policy is enabled in the domain per VRF per address family.

By default, the protocol redistribution interleak policies with appropriate route maps are set up for all valid configurations. The administrator enables transit routing purely by virtue of creating `l3extInstP` subnets with *scope=export-route control* to allow certain routes to be transmitted between two `L3extOut`s in the same VRF. Apart from the scope of `l3extInstP` subnets, there are no special protocol specific configurations for covering transit cases. Apart from the scope, which is protocol-specific, other parameters of the default route leak policy are common across all the protocols.

The OSPF on another `L3extOut` on a different node transit scenario is supported with EIGRP.

Observe the following EIGRP guidelines and limitations:

- At this time, multiple EIGRP L3Outs are not supported on the same leaf switch.
- All routes are imported on an `L3extOut` that uses EIGRP. Import subnet scope is disabled in the GUI if EIGRP is the protocol on the `L3extOut`.



CHAPTER 8

User Access, Authentication, and Accounting

This chapter contains the following sections:

User Access, Authorization, and Accounting

Application Policy Infrastructure Controller (APIC) policies manage the authentication, authorization, and accounting (AAA) functions of the Cisco Application Centric Infrastructure (ACI) fabric. The combination of user privileges, roles, and domains with access rights inheritance enables administrators to configure AAA functions at the managed object level in a granular fashion. These configurations can be implemented using the REST API, the CLI, or the GUI.

Note There is a known limitation where you cannot have more than 32 characters for the login domain name. In addition, the combined number of characters for the login domain name and the user name cannot exceed 64 characters.

Multiple Tenant Support

A core Application Policy Infrastructure Controller (APIC) internal data access control system provides multitenant isolation and prevents information privacy from being compromised across tenants. Read/write restrictions prevent any tenant from seeing any other tenant's configuration, statistics, faults, or event data.

Unless the administrator assigns permissions to do so, tenants are restricted from reading fabric configuration, policies, statistics, faults, or events.

User Access: Roles, Privileges, and Security Domains

The APIC provides access according to a user's role through role-based access control (RBAC). An Cisco Application Centric Infrastructure (ACI) fabric user is associated with the following:

- A predefined or custom role, which is a set of one or more privileges assigned to a user
- A set of privileges, which determine the managed objects (MOs) to which the user has access
- For each role, a privilege type: no access, read-only, or read-write
- One or more security domain tags that identify the portions of the management information tree (MIT) that a user can access

Roles and Privileges

A privilege controls access to a particular function within the system. The ACI fabric manages access privileges at the managed object (MO) level. Every object holds a list of the privileges that can read from it and a list of the privileges that can write to it. All objects that correspond to a particular function will have the privilege for that function in its read or write list. Because an object might correspond to additional functions, its lists might contain multiple privileges. When a user is assigned a role that contains a privilege, the user is given read access to the associated objects whose read list specifies read access, and write access to those whose write list specifies write access.

As an example, 'fabric-equipment' is a privilege that controls access to all objects that correspond to equipment in the physical fabric. An object corresponding to equipment in the physical fabric, such as 'eqptBoard,' will have 'fabric-equipment' in its list of privileges. The 'eqptBoard' object allows read-only access for the 'fabric-equipment' privilege. When a user is assigned a role such as 'fabric-admin' that contains the privilege 'fabric-equipment,' the user will have access to those equipment objects, including read-only access to the 'eqptBoard' object.

Note Some roles contain other roles. For example, '-admin' roles such as tenant-admin, fabric-admin, access-admin are groupings of roles with the same base name. For example, 'access-admin' is a grouping of 'access-connectivity', 'access-equipment', 'access-protocol', and 'access-qos.' Similarly, tenant-admin is a grouping of roles with a 'tenant' base, and fabric-admin is a grouping of roles with a 'fabric' base.

The 'admin' role contains all privileges.

For more details about roles and privileges see APIC Roles and Privileges Matrix.

Security Domains

A security domain is a tag associated with a certain subtree in the ACI MIT object hierarchy. For example, the default tenant "common" has a domain tag `common`. Similarly, the special domain tag `all` includes the entire MIT object tree. An administrator can assign custom domain tags to the MIT object hierarchy. For example, an administrator could assign the "solar" domain tag to the tenant named solar. Within the MIT, only certain objects can be tagged as security domains. For example, a tenant can be tagged as a security domain but objects within a tenant cannot.

Note Security Domain password strength parameters can be configured by creating **Custom Conditions** or by selecting **Any Three Conditions** that are provided.

Creating a user and assigning a role to that user does not enable access rights. It is necessary to also assign the user to one or more security domains. By default, the ACI fabric includes two special pre-created domains:

- `All`—allows access to the entire MIT
- `Infra`— allows access to fabric infrastructure objects/subtrees, such as fabric access policies

Note For read operations to the managed objects that a user's credentials do not allow, a "DN/Class Not Found" error is returned, not "DN/Class Unauthorized to read." For write operations to a managed object that a user's credentials do not allow, an HTTP 401 Unauthorized error is returned. In the GUI, actions that a user's credentials do not allow, either they are not presented, or they are grayed out.

A set of predefined managed object classes can be associated with domains. These classes should not have overlapping containment. Examples of classes that support domain association are as follows:

- Layer 2 and Layer 3 network managed objects
- Network profiles (such as physical, Layer 2, Layer 3, management)
- QoS policies

When an object that can be associated with a domain is created, the user must assign domain(s) to the object within the limits of the user's access rights. Domain assignment can be modified at any time.

If a virtual machine management (VMM) domain is tagged as a security domain, the users contained in the security domain can access the correspondingly tagged VMM domain. For example, if a tenant named solar is tagged with the security domain called sun and a VMM domain is also tagged with the security domain called sun, then users in the solar tenant can access the VMM domain according to their access rights.

Accounting

Cisco Application Centric Infrastructure (ACI) fabric accounting is handled by these two managed objects that are processed by the same mechanism as faults and events:

- The `aaaSessionLR` managed object tracks user account login and logout sessions on the Cisco Application Policy Infrastructure Controller (APIC) and switches, and token refresh. The Cisco ACI fabric session alert feature stores information such as the following:
  - Username
  - IP address initiating the session
  - Type (telnet, HTTPS, REST, and so on)
  - Session time and length

- Token refresh: A user account login event generates a valid active token which is required in order for the user account to exercise its rights in the Cisco ACI fabric.

Note Token expiration is independent of login; a user could log out but the token expires according to the duration of the timer value it contains.

- The `aaaModLR` managed object tracks the changes users make to objects and when the changes occurred.
- If the AAA server is not pingable, it is marked unavailable and a fault is seen.

Both the `aaaSessionLR` and `aaaModLR` event logs are stored in Cisco APIC shards. After the data exceeds the pre-set storage allocation size, it overwrites records on a first-in first-out basis.

Note In the event of a destructive event such as a disk crash or a fire that destroys a Cisco APIC cluster node, the event logs are lost; event logs are not replicated across the cluster.

The `aaaModLR` and `aaaSessionLR` managed objects can be queried by class or by distinguished name (DN). A class query provides all the log records for the whole fabric. All `aaaModLR` records for the whole fabric are available from the GUI at the **Fabric** > **Inventory** > **POD** > **History** > **Audit Log** section, The Cisco APIC GUI **History** > **Audit Log** options enable viewing event logs for a specific object identified in the GUI.

The standard syslog, callhome, REST query, and CLI export mechanisms are fully supported for `aaaModLR` and `aaaSessionLR` managed object query data. There is no default policy to export this data.

There are no pre-configured queries in the Cisco APIC that report on aggregations of data across a set of objects or for the entire system. A fabric administrator can configure export policies that periodically export `aaaModLR` and `aaaSessionLR` query data to a syslog server. Exported data can be archived periodically and used to generate custom reports from portions of the system or across the entire set of system logs.

Routed Connectivity to External Networks as a Shared Service Billing and Statistics

The Cisco Application Policy Infrastructure Controller (APIC) can be configured to collect byte count and packet count billing statistics from a port configured for routed connectivity to external networks as a shared service. The external networks are represented as external L3Out endpoint group (l3extInstP managed object) in Cisco Application Centric Infrastructure (ACI). Any EPG in any tenant can share an external L3Out EPG for routed connectivity to external networks. Billing statistics can be collected for each EPG in any tenant that uses an external L3Out EPG as a shared service. The leaf switch where the external L3Out EPG is provisioned forwards the billing statistics to the Cisco APIC where they are aggregated. Accounting policies can be configured to export these billing statics periodically to a server.

Custom RBAC Rules

RBAC rules enable a fabric-wide administrator to provide access across security domains that would otherwise be blocked. Use RBAC rules to expose physical resources or share services that otherwise are inaccessible because they are in a different security domain. RBAC rules provide read access only to their target resources. The GUI RBAC rules page is located under Admin => AAA => Security Management. RBAC rules can be created prior to the existence of a resource. Descriptions of RBAC rules, roles, and privileges (and their dependencies) are documented in the Management Information Model reference.

Used for viewing the policies configured including troubleshooting policies.

Note that ops role cannot be used for creating new monitoring and troubleshooting policies. They need to be done using the admin privilege, just like any other configurations in the APIC.

Selectively Expose Physical Resources across Security Domains

A fabric-wide administrator uses RBAC rules to selectively expose physical resources to users that otherwise are inaccessible because they are in a different security domain.

For example, if a user in tenant Solar needs access to a virtual machine management (VMM) domain, the fabric-wide admin could create an RBAC rule to allow this. The RBAC rule is comprised of these two parts: the distinguished name (DN) that locates the object to be accessed plus the name of the security domain that contains the user who will access the object. So, in this example, when designated users in the security domain Solar are logged in, this rule gives them access to the VMM domain as well as all its child objects in the tree. To give users in multiple security domains access to the VMM domain, the fabric-wide administrator would create an RBAC rule for each security domain that contains the DN for the VMM domain plus the security domain.

Note While an RBAC rule exposes an object to a user in a different part of the management information tree, it is not possible to use the CLI to navigate to such an object by traversing the structure of the tree. However, as long as the user knows the DN of the object included in the RBAC rule, the user can use the CLI to locate it via an MO find command.

Enable Sharing of Services across Security Domains

A fabric-wide administrator uses RBAC rules to provision trans-tenant EPG communications that enable shared services across tenants.

APIC Local Users

An administrator can choose not to use external AAA servers but rather configure users on the APIC itself. These users are called APIC-local users.

At the time a user sets their password, the APIC validates it against the following criteria:

- Minimum password length is 8 characters.

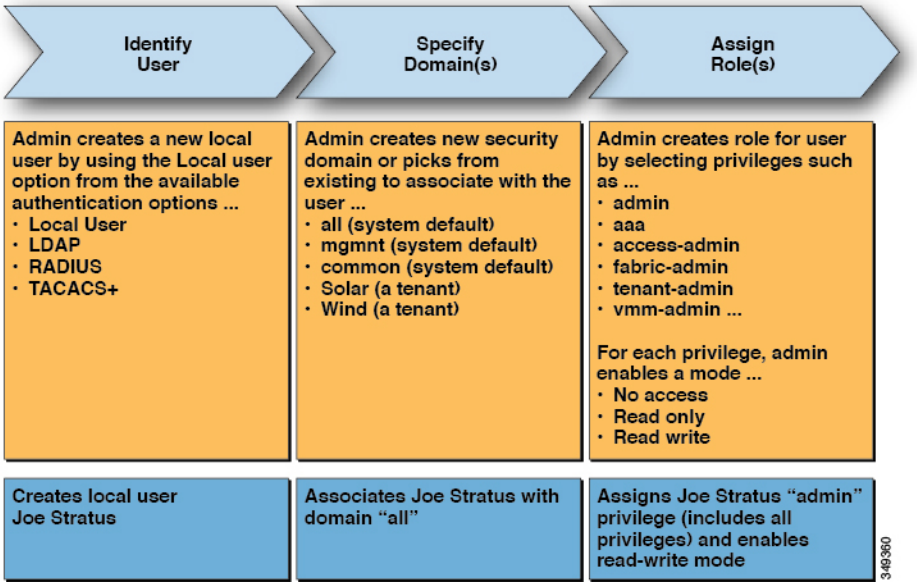
- Maximum password length is 64 characters.
- Has fewer than three consecutive repeated characters.
- Must have characters from at least three of the following characters types: lowercase, uppercase, digit, symbol.
- Does not use easily guessed passwords.
- Cannot be the username or the reverse of the username.
- Cannot be any variation of cisco, isco or any permutation of these characters or variants obtained by changing the capitalization of letters therein.

Cisco ACI uses a crypt library with a SHA256 one-way hash for storing passwords. At rest hashed passwords are stored in an encrypted filesystem. The key for the encrypted filesystem is protected using the Trusted Platform Module (TPM).

The APIC also enables administrators to grant access to users configured on externally managed authentication Lightweight Directory Access Protocol (LDAP), RADIUS, TACACS+, or SAML servers. Users can belong to different authentication systems and can log in simultaneously to the APIC.

The following figure shows how the process works for configuring an admin user in the local APIC authentication database who has full access to the entire ACI fabric.

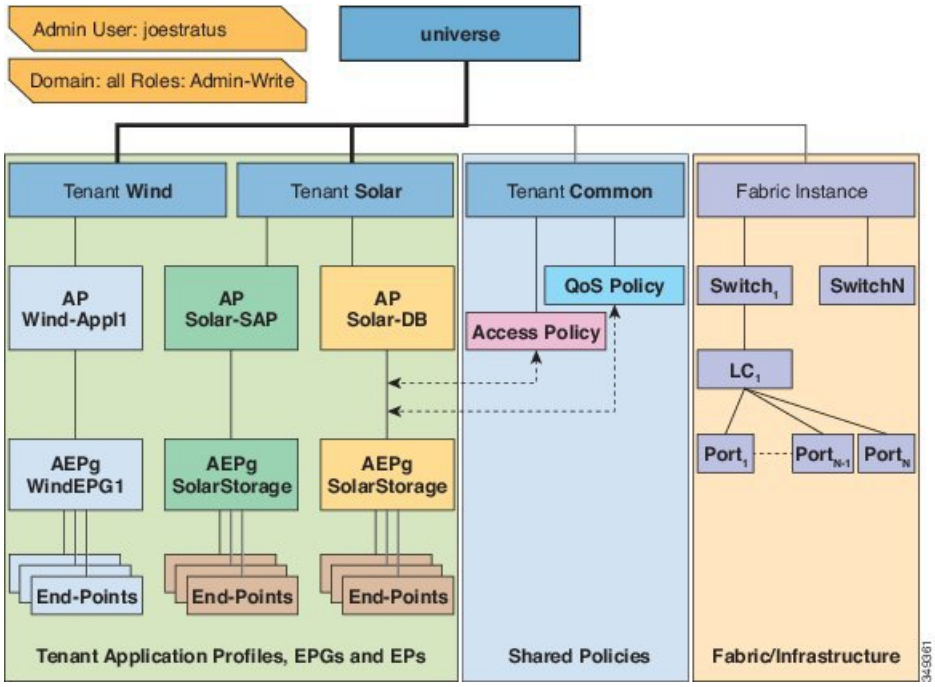
Figure 91: APIC Local User Configuration Process



Note The security domain "all" represents the entire Managed Information Tree (MIT). This domain includes all policies in the system and all nodes managed by the APIC. Tenant domains contain all the users and managed objects of a tenant. Tenant administrators should not be granted access to the "all" domain.

The following figure shows the access that the admin user Joe Stratus has to the system.

Figure 92: Result of Configuring Admin User for "all" Domain

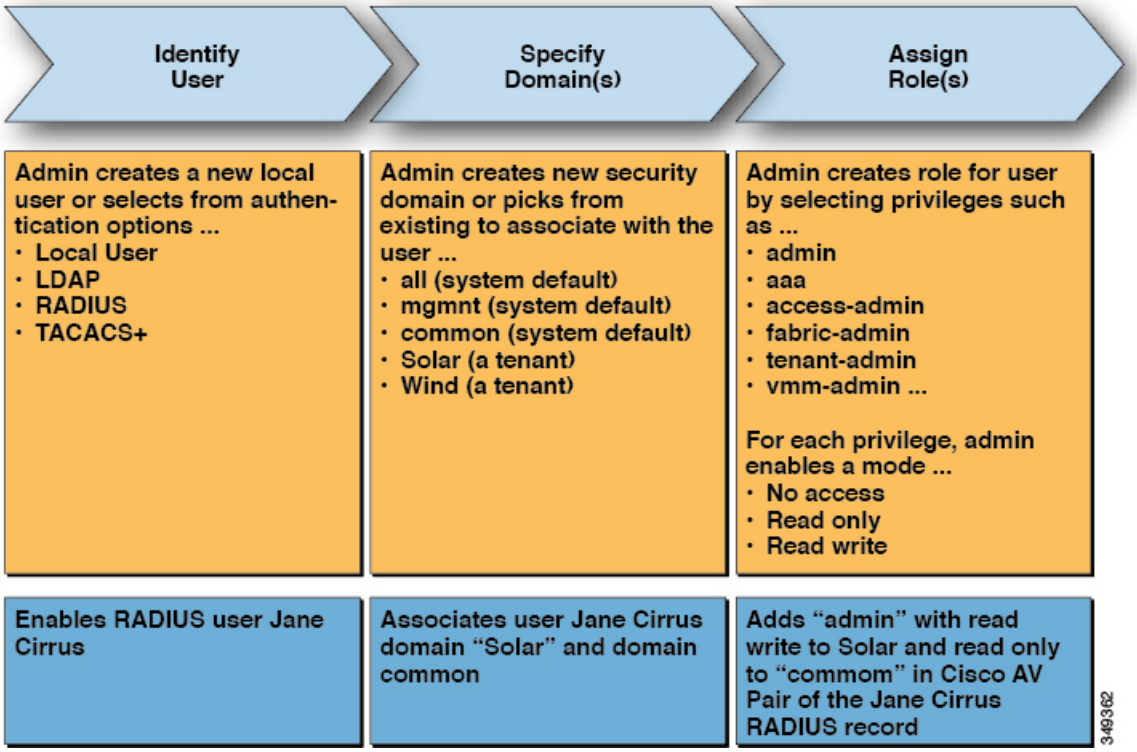


The user Joe Stratus with read-write "admin" privileges is assigned to the domain "all" which gives him full access to the entire system.

Externally Managed Authentication Server Users

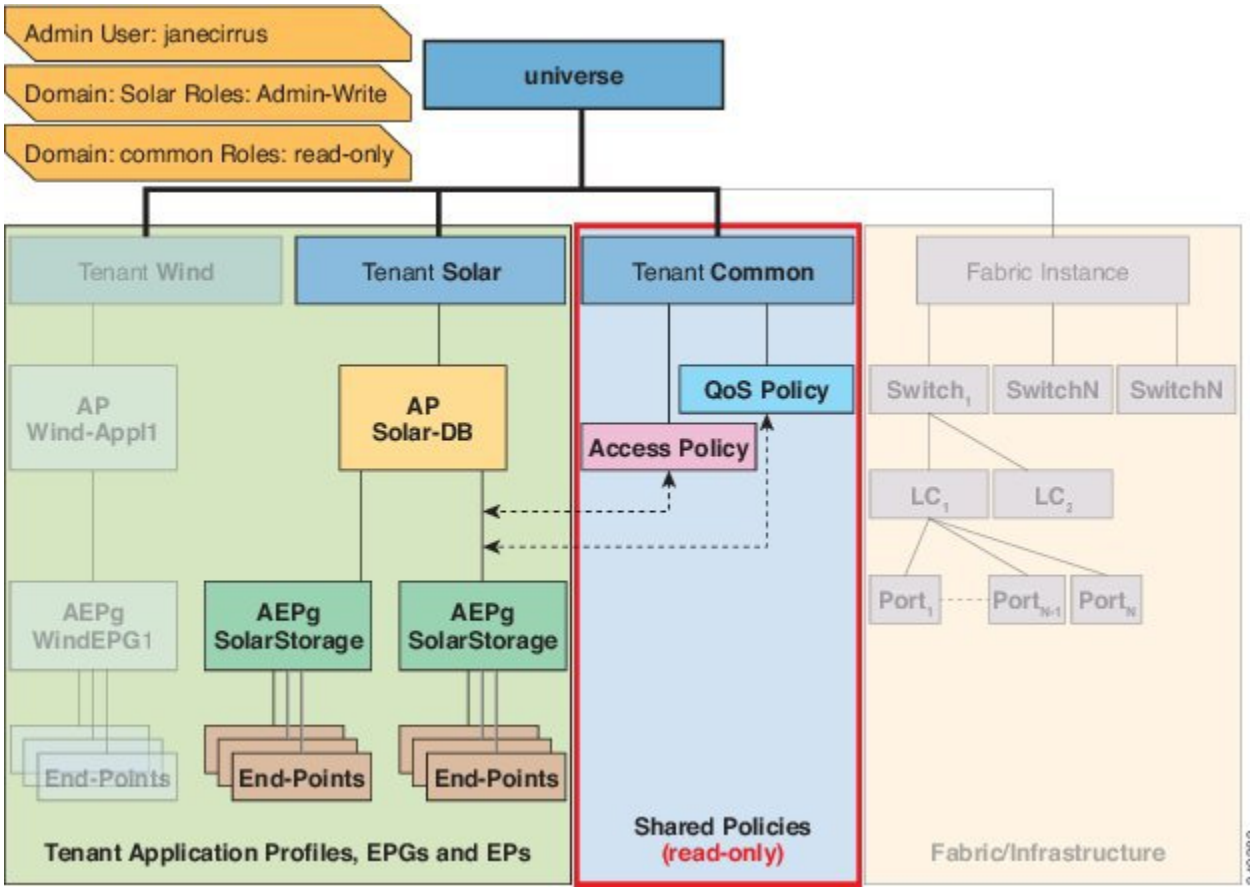
The following figure shows how the process works for configuring an admin user in an external RADIUS server who has full access to the tenant Solar.

Figure 93: Process for Configuring Users on External Authentication Servers



The following figure shows the access the admin user Jane Cirrus has to the system.

Figure 94: Result of Configuring Admin User for Tenant Solar



In this example, the Solar tenant administrator has full access to all the objects contained in the Solar tenant as well as read-only access to the tenant Common. Tenant admin Jane Cirrus has full access to the tenant Solar, including the ability to create new users in tenant Solar. Tenant users are able to modify configuration parameters of the ACI fabric that they own and control. They also are able to read statistics and monitor faults and events for the entities (managed objects) that apply to them such as endpoints, endpoint groups (EPGs) and application profiles.

In the example above, the user Jane Cirrus was configured on an external RADIUS authentication server. To configure an AV Pair on an external authentication server, add a Cisco AV Pair to the existing user record. The Cisco AV Pair specifies the Role-Based Access Control (RBAC) roles and privileges for the user on the APIC. The RADIUS server then propagates the user privileges to the APIC controller.

In the example above, the configuration for an open radius server (/etc/raddb/users) is as follows:

```
janecirrus Cleartext-Password := "<password>"
Cisco-avpair = "shell:domains = solar/admin/,common//read-all(16001)"
```

This example includes the following elements:

- `janecirrus` is the tenant administrator
- `solar` is the tenant
- `admin` is the role with write privileges

- `common` is the tenant-common subtree that all users should have read-only access to
- `read-all` is the role with read privileges

Cisco AV Pair Format

The Cisco APIC requires that an administrator configure a Cisco AV Pair on an external authentication server and only looks for one AV pair string. To do so, an administrator adds a Cisco AV pair to the existing user record. The Cisco AV pair specifies the APIC required RBAC roles and privileges for the user.

In order for the AV pair string to work, it must be formatted as follows:

```
shell:domains =
ACI_Security_Domain_1/ACI_Write_Role_1|ACI_Write_Role_2|ACI_Write_Role_3/ACI_Read_Role_1|ACI_Read_Role_2,
ACI_Security_Domain_2/ACI_Write_Role_1|ACI_Write_Role_2|ACI_Write_Role_3/ACI_Read_Role_1|ACI_Read_Role_2,
ACI_Security_Domain_3/ACI_Write_Role_1|ACI_Write_Role_2|ACI_Write_Role_3/ACI_Read_Role_1|ACI_Read_Role_2
```

- **shell:domains=** - Required so that ACI reads the string correctly. This must always prepend the shell string.
- **ACI_Security_Domain_1//admin** - Grants admin read only access to the tenants in this security domain.
- **ACI_Security_Domain_2/admin** - Grants admin write access to the tenants in this security domain.
- **ACI_Security_Domain_3/read-all** - Grants read-all write access to the tenants in this security domain.

Note /'s separate the security domain, write, read sections of the string. |'s separate multiple write or read roles within the same security domain.

Note Starting with Cisco APIC release 2.1, if no UNIX ID is provided in AV Pair, the APIC allocates the unique UNIX user ID internally.

The APIC supports the following regexes:

```
shell:domains\\s*[=:]\\s*((\\S+?/\\S*?/\\S*?),\\S+?/\\S*?/\\S*?){0,31})(\\(\\d+\\))$
shell:domains\\s*[=:]\\s*((\\S+?/\\S*?/\\S*?),\\S+?/\\S*?/\\S*?){0,31})$
```

Examples:

- Example 1: A Cisco AV Pair that contains a single Login domain with only writeRoles:

```
shell:domains=ACI_Security_Domain_1/Write_Role_1|Write_Role_2/
```

- Example 2: A Cisco AV Pair that contains a single Login domain with only readRoles:

```
shell:domains=Security_Domain_1//Read_Role_1|Read_Role_2
```

Note The "/" character is a separator between writeRoles and readRoles per Login domain and is required even if only one type of role is to be used.

The Cisco AVpair string is case sensitive. Although a fault may not be seen, using mismatching cases for the domain name or roles could lead to unexpected privileges being given.

AV Pair GUI Configuration

The security domain is defined in the ACI GUI under **Admin** > **AAA** > **Security Management** > **Security Domains** and assigned to a tenant under **Tenants** > **Tenant_Name** > **Policy**.

A security domain must have either a read or write role. These roles are defined in **APIC** > **Admin** > **Security Management** > **Roles**. If a role is input into the write section it automatically grants read privileges of the same level so there is no need to have ACI_Security_Domain_1/admin/admin.

RADIUS

To configure users on RADIUS servers, the APIC administrator must configure the required attributes (`shell:domains`) using the `cisco-av-pair` attribute. The default user role is network-operator.

The SNMPv3 authentication protocol options are SHA and MD5. The privacy protocol options are AES-128 and DES. If these options are not specified in the `cisco-av-pair` attribute, MD5 and DES are the default authentication protocols.

For example, SNMPv3 authentication and privacy protocol attributes can be specified as follows:

```
snmpv3:auth=SHA priv=AES-128
```

Similarly, the list of domains would be as follows:

```
shell:domains="domainA domainB …"
```

TACACS+ Authentication

Terminal Access Controller Access Control System Plus (TACACS+) is another remote AAA protocol that is supported by Cisco devices. TACACS+ has the following advantages over RADIUS authentication:

- Provides independent AAA facilities. For example, the Cisco Application Policy Infrastructure Controller (APIC) can authorize access without authenticating.
- Uses TCP to send data between the AAA client and server, enabling reliable transfers with a connection-oriented protocol.
- Encrypts the entire protocol payload between the switch and the AAA server to ensure higher data confidentiality. RADIUS encrypts passwords only.
- Uses the av-pairs that are syntactically and configurationally different than RADIUS but the Cisco APIC supports `shell:domains`.

The following XML example configures the Cisco Application Centric Infrastructure (ACI) fabric to work with a TACACS+ provider at IP address 10.193.208.9:

```
<aaaTacacsPlusProvider name="10.193.208.9"
           key="test123"
           authProtocol="pap"/>
```

Note While the examples provided here use IPv4 addresses, IPv6 addresses could also be used.

The following guidelines and limitations apply when using TACACS+:

- The TACACS server and TACACs ports must be reachable by ping.
- The TACACS server with the highest priority is considered first to be the primary server.

LDAP/Active Directory Authentication

Similar to RADIUS and TACACS+, LDAP allows a network element to retrieve AAA credentials that can be used to authenticate and then authorize the user to perform certain actions. An added certificate authority configuration can be performed by an administrator to enable LDAPS (LDAP over SSL) trust and prevent man-in-the-middle attacks.

The XML example below configures the ACI fabric to work with an LDAP provider at IP address 10.30.12.128.

Note While the examples provided here use IPv4 addresses, IPv6 addresses could also be used.

```
<aaaLdapProvider name="10.30.12.128"
           rootdn="CN=Manager,DC=ifc,DC=com"
           basedn="DC=ifc,DC=com"
           SSLValidationLevel="strict"
           attribute="CiscoAVPair"
           enableSSL="yes"
           filter="cn=$userid"
           port="636" />
```

Note For LDAP configurations, best practice is to use **`CiscoAVPair`** as the attribute string. If customer faces the issue using Object ID 1.3.6.1.4.1.9.22.1, an additional Object ID 1.3.6.1.4.1.9.2742.1-5 can also be used in the LDAP server.

Instead of configuring the Cisco AVPair, you have the option to create LDAP group maps in the APIC.

User IDs in the APIC Bash Shell

User IDs on the APIC for the Linux shell are generated within the APIC for local users. Users whose authentication credential is managed on external servers, the user ID for the Linux shell can be specified in the cisco-av-pair. Omitting the (16001) in the above cisco-av-pair is legal, in which case the remote user gets a default Linux user ID of 23999. Linux User IDs are used during bash sessions, allowing standard Linux permissions enforcement. Also, all managed objects created by a user are marked as created-by that user's Linux user ID.

The following is an example of a user ID as seen in the APIC Bash shell:

```
admin@ifav17-ifc1:~> touch myfile
admin@ifav17-ifc1:~> ls -l myfile
-rw-rw-r-- 1 admin admin 0 Apr 13 21:43 myfile
admin@ifav17-ifc1:~> ls -ln myfile
-rw-rw-r-- 1 15374 15374 0 Apr 13 21:43 myfile
admin@ifav17-ifc1:~> id
uid=15374(admin) gid=15374(admin) groups=15374(admin)
```

Login Domains

A login domain defines the authentication domain for a user. Login domains can be set to the Local, LDAP, RADIUS, or TACACS+ authentication mechanisms. When accessing the system from REST, the CLI, or the GUI, the APIC enables the user to select the correct authentication domain.

For example, in the REST scenario, the username is prefixed with a string so that the full login username looks as follows:

```
apic:<domain>\<username>
```

If accessing the system from the GUI, the APIC offers a drop-down list of domains for the user to select. If no `apic: domain` is specified, the default authentication domain servers are used to look up the username.

Starting in ACI version 1.0(2x), the login domain fallback of the APIC defaults local. If the default authentication is set to a non-local method and the console authentication method is also set to a non-local method and both non-local methods do not automatically fall back to local authentication, the APIC can still be accessed via local authentication.

To access the APIC fallback local authentication, use the following strings:

- From the GUI, use *apic:fallback\\username*.
- From the REST API, use *apic#fallback\\username*.

Note Do not change the fallback login domain. Doing so could result in being locked out of the system.

About SAML

SAML is an XML-based open standard data format that enables administrators to access a defined set of Cisco collaboration applications seamlessly after signing into one of those applications. SAML describes the exchange of security related information between trusted business partners. It is an authentication protocol used by service providers to authenticate a user. SAML enables exchange of security authentication information between an Identity Provider (IdP) and a service provider.

SAML SSO uses the SAML 2.0 protocol to offer cross-domain and cross-product single sign-on for Cisco collaboration solutions. SAML 2.0 enables SSO across Cisco applications and enables federation between Cisco applications and an IdP. SAML 2.0 allows Cisco administrative users to access secure web domains to exchange user authentication and authorization data, between an IdP and a Service Provider while maintaining high security levels. The feature provides secure mechanisms to use common credentials and relevant information across various applications.

The authorization for SAML SSO Admin access is based on Role-Based Access Control (RBAC) configured locally on Cisco collaboration applications.

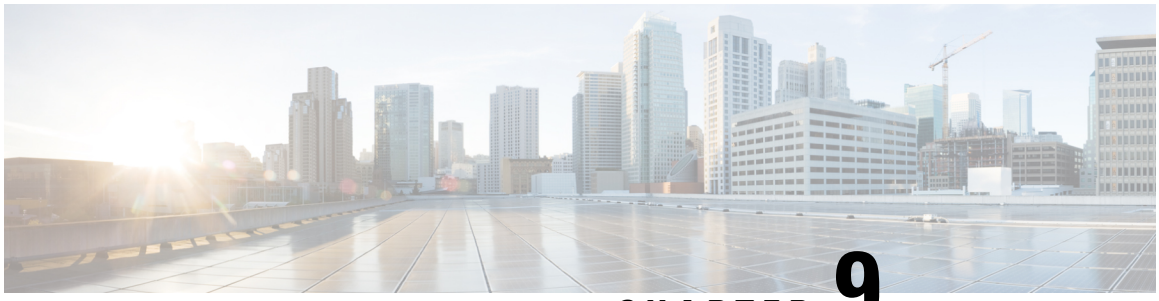
SAML SSO establishes a Circle of Trust (CoT) by exchanging metadata and certificates as part of the provisioning process between the IdP and the Service Provider. The Service Provider trusts the IdP's user information to provide access to the various services or applications.

Note Service providers are no longer involved in authentication. SAML 2.0 delegates authentication away from the service providers and to the IdPs.

The client authenticates against the IdP, and the IdP grants an Assertion to the client. The client presents the Assertion to the Service Provider. Since there is a CoT established, the Service Provider trusts the Assertion and grants access to the client.

Enabling SAML SSO results in several advantages:

- It reduces password fatigue by removing the need for entering different user name and password combinations.
- It transfers the authentication from your system that hosts the applications to a third party system. UsingSAML SSO, you can create a circle of trust between an IdP and a service provider. The service provider trusts and relies on the IdP to authenticate the users.
- It protects and secures authentication information. It provides encryption functions to protect authentication information passed between the IdP, service provider, and user. SAML SSO can also hide authentication messages passed between the IdP and the service provider from any external user.
- It improves productivity because you spend less time re-entering credentials for the same identity.
- It reduces costs as fewer help desk calls are made for password reset, thereby leading to more savings.



CHAPTER 9

Virtual Machine Manager Domains

This chapter contains the following sections:

Cisco ACI VM Networking Support for Virtual Machine Managers

Benefits of ACI VM Networking

Cisco Application Centric Infrastructure (ACI) virtual machine (VM) networking supports hypervisors from multiple vendors. It provides the hypervisors programmable and automated access to high-performance scalable virtualized data center infrastructure.

Programmability and automation are critical features of scalable data center virtualization infrastructure. The Cisco ACI open REST API enables virtual machine integration with and orchestration of the policy model-based Cisco ACI fabric. Cisco ACI VM networking enables consistent enforcement of policies across both virtual and physical workloads that are managed by hypervisors from multiple vendors.

Attachable entity profiles easily enable VM mobility and placement of workloads anywhere in the Cisco ACI fabric. The Cisco Application Policy Infrastructure Controller (APIC) provides centralized troubleshooting, application health score, and virtualization monitoring. Cisco ACI multi-hypervisor VM automation reduces or eliminates manual configuration and manual errors. This enables virtualized data centers to support large numbers of VMs reliably and cost effectively.

Supported Products and Vendors

Cisco ACI supports virtual machine managers (VMMs) from the following products and vendors:

- **Cisco Unified Computing System Manager (UCSM)**

  Integration of Cisco UCSM is supported beginning in Cisco Cisco APIC Release 4.1(1). For information, see the chapter "Cisco ACI with Cisco UCSM Integration in the *Cisco ACI Virtualization Guide, Release 4.1(1)*.

- **Cisco Application Centric Infrastructure (ACI) Virtual Pod (vPod)**

  Cisco ACI vPod is in general availability beginning in Cisco APIC Release 4.0(2). For information, see the Cisco ACI vPod documentation on Cisco.com.

- **Cisco ACI Virtual Edge**

  For information, see the Cisco ACI Virtual Edge documentation on Cisco.com.

- **Cloud Foundry**

  Cloud Foundry integration with Cisco ACI is supported beginning with Cisco APIC Release 3.1(2). For information, see the knowledge base article, Cisco ACI and Cloud Found Integration on Cisco.com.

- **Kubernetes**

  For information, see the knowledge base article, Cisco ACI and Kubernetes Integration on Cisco.com.

- **Microsoft System Center Virtual Machine Manager (SCVMM)**

  For information, see the chapters "Cisco ACI with Microsoft SCVMM" and "Cisco ACI with Microsoft Windows Azure Pack" in the *Cisco ACI Virtualization Guide* on Cisco.com

- **OpenShift**

  For information, see the OpenShift documentation on Cisco.com.

- **OpenStack**

  For information, see the OpenStack documentation on Cisco.com.

- **Red Hat Virtualization (RHV)**

  For information, see the knowledge base article, Cisco ACI and Red Hat Integration on Cisco.com.

- **VMware Virtual Distributed Switch (VDS)**

  For information, see the chapter "Cisco "ACI with VMware VDS Integration" in the Cisco ACI Virtualization Guide.

See the Cisco ACI Virtualization Compatibility Matrix for the most current list of verified interoperable products.

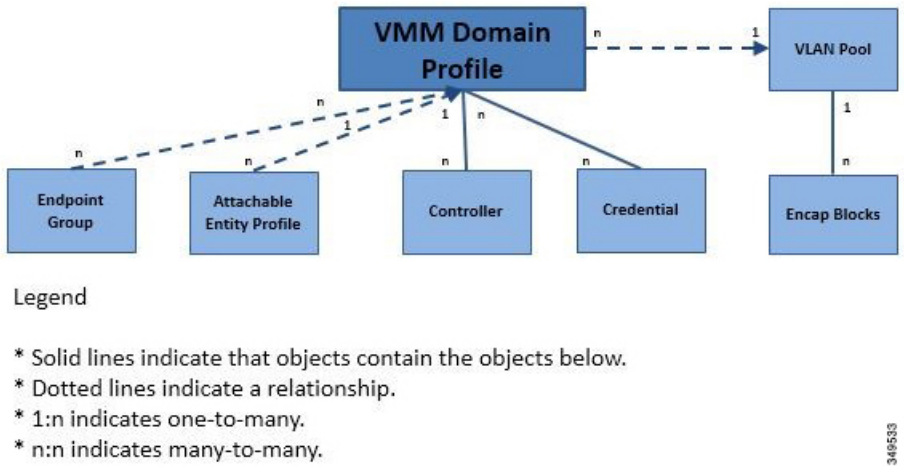
Note Beginning with Cisco APIC Release 5.0(1), Cisco Application Virtual Switch (AVS) is no longer supported. If you use Cisco AVS and upgrade to Cisco APIC Release 5.0(1), in case of issues, the fabric will not be supported. Also, a fault will be raised for the Cisco AVS domain.

If you use Cisco AVS, we recommend that you migrate to Cisco Application Centric Infrastructure (ACI) Virtual Edge Virtual Edge. See the *Cisco ACI Virtual Edge Installation Guide, Release 3.0(x)* on Cisco.com.

VMM Domain Policy Model

VMM domain profiles (`vmmDomP`) specify connectivity policies that enable virtual machine controllers to connect to the ACI fabric. The figure below provides an overview of the `vmmDomP` policy.

Figure 95: VMM Domain Policy Model Overview



Virtual Machine Manager Domain Main Components

ACI fabric virtual machine manager (VMM) domains enable an administrator to configure connectivity policies for virtual machine controllers. The essential components of an ACI VMM domain policy include the following:

- **Virtual Machine Manager Domain Profile**—Groups VM controllers with similar networking policy requirements. For example, VM controllers can share VLAN pools and application endpoint groups (EPGs). The APIC communicates with the controller to publish network configurations such as port groups that are then applied to the virtual workloads. The VMM domain profile includes the following essential components:
  - **Credential**—Associates a valid VM controller user credential with an APIC VMM domain.
  - **Controller**—Specifes how to connect to a VM controller that is part of a policy enforcement domain. For example, the controller specifies the connection to a VMware vCenter that is part a VMM domain.

Note A single VMM domain can contain multiple instances of VM controllers, but they must be from the same vendor (for example, from VMware or from Microsoft.

- **EPG Association**—Endpoint groups regulate connectivity and visibility among the endpoints within the scope of the VMM domain policy. VMM domain EPGs behave as follows:
  - The APIC pushes these EPGs as port groups into the VM controller.

    - An EPG can span multiple VMM domains, and a VMM domain can contain multiple EPGs.

- **Attachable Entity Profile Association**—Associates a VMM domain with the physical network infrastructure. An attachable entity profile (AEP) is a network interface template that enables deploying VM controller policies on a large set of leaf switch ports. An AEP specifies which switches and ports are available, and how they are configured.
- **VLAN Pool Association**—A VLAN pool specifies the VLAN IDs or ranges used for VLAN encapsulation that the VMM domain consumes.

Virtual Machine Manager Domains

An APIC VMM domain profile is a policy that defines a VMM domain. The VMM domain policy is created in APIC and pushed into the leaf switches.

VMM domains provide the following:

- A common layer in the ACI fabric that enables scalable fault-tolerant support for multiple VM controller platforms.
- VMM support for multiple tenants within the ACI fabric.

VMM domains contain VM controllers such as VMware vCenter or Microsoft SCVMM Manager and the credential(s) required for the ACI API to interact with the VM controller. A VMM domain enables VM mobility within the domain but not across domains. A single VMM domain can contain multiple instances of VM controllers but they must be the same kind. For example, a VMM domain can contain many VMware vCenters managing multiple controllers each running multiple VMs but it may not also contain SCVMM Managers. A VMM domain inventories controller elements (such as pNICs, vNICs, VM names, and so forth) and pushes policies into the controller(s), creating port groups, and other necessary elements. The ACI VMM domain listens for controller events such as VM mobility and responds accordingly.

VMM Domain VLAN Pool Association

VLAN pools represent blocks of traffic VLAN identifiers. A VLAN pool is a shared resource and can be consumed by multiple domains such as VMM domains and Layer 4 to Layer 7 services.

Each pool has an allocation type (static or dynamic), defined at the time of its creation. The allocation type determines whether the identifiers contained in it will be used for automatic assignment by the Cisco APIC (dynamic) or set explicitly by the administrator (static). By default, all blocks contained within a VLAN pool have the same allocation type as the pool but users can change the allocation type for encapsulation blocks contained in dynamic pools to static. Doing so excludes them from dynamic allocation.

A VMM domain can associate with only one dynamic VLAN pool. By default, the assignment of VLAN identifiers to EPGs that are associated with VMM domains is done dynamically by the Cisco APIC. While dynamic allocation is the default and preferred configuration, an administrator can statically assign a VLAN identifier to an endpoint group (EPG) instead. In that case, the identifiers used must be selected from encapsulation blocks in the VLAN pool associated with the VMM domain, and their allocation type must be changed to static.

The Cisco APIC provisions VMM domain VLAN on leaf ports based on EPG events, either statically binding on leaf ports or based on VM events from controllers such as VMware vCenter or Microsoft SCVMM.

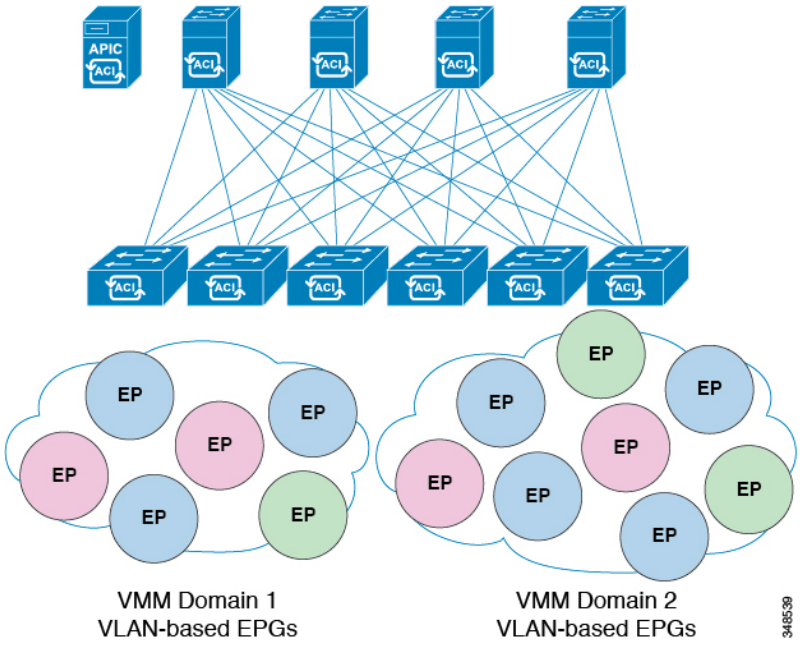
Note In dynamic VLAN pools, if a VLAN is disassociated from an EPG, it is automatically reassociated with the EPG in five minutes.

Note Dynamic VLAN association is not a part of configuration rollback, that is, in case an EPG or tenant was initially removed and then restored from the backup, a new VLAN is automatically allocated from the dynamic VLAN pools.

VMM Domain EPG Association

The Cisco Application Centric Infrastructure (ACI) fabric associates tenant application profile endpoint groups (EPGs) to virtual machine manager (VMM) domains, The Cisco ACI does so either automatically by an orchestration component such as Microsoft Azure, or by a Cisco Application Policy Infrastructure Controller (APIC) administrator creating such configurations. An EPG can span multiple VMM domains, and a VMM domain can contain multiple EPGs.

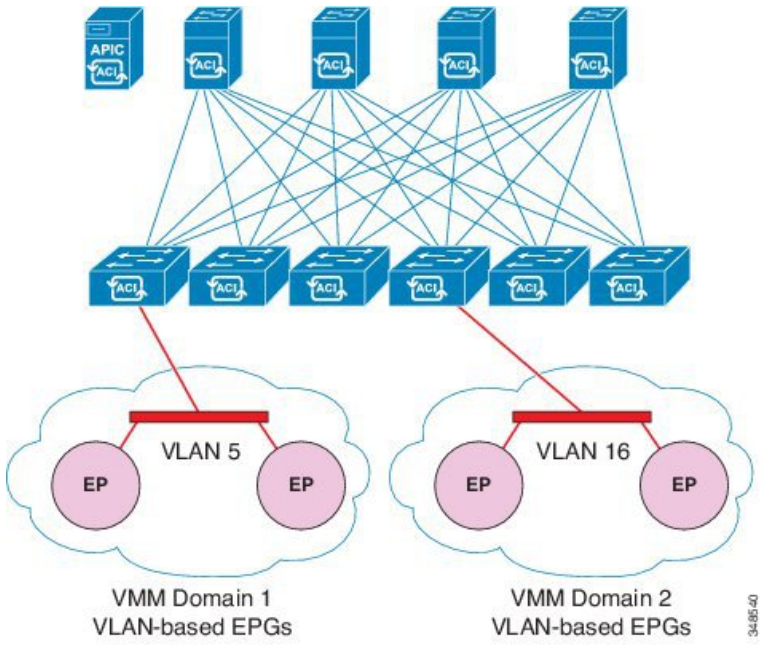
Figure 96: VMM Domain EPG Association



In the preceding illustration, end points (EPs) of the same color are part of the same EPG. For example, all the green EPs are in the same EPG although they are in two different VMM domains.

See the latest *Verified Scalability Guide for Cisco ACI* for virtual network and VMM domain EPG capacity information.

Figure 97: VMM Domain EPG VLAN Consumption



Note When multiple VMM domains with an overlapping VLAN ID range are connected to the same leaf switch, those domains should use the same VLAN pool. With the same VLAN pool, Cisco APIC can make sure to pick a different VLAN ID for each domain-to-EPG association. Otherwise, Cisco APIC might pick a VLAN ID that is already used on the switch for another domain-to-EPG association, which causes the VLAN deployment fail.

When multiple VMM domains with an overlapping VLAN ID range are connected to the same leaf switch and those domains use the same VLAN pool, you can have multiple VMM domains associated with the same EPG. However, each domain-to-EPG association deploys a different VLAN ID, respectively, even though the VLANs are for the same EPG and potentially are on the same port. If using VLAN IDs in this manner is suboptimal to your requirements, you can use the same VMM domain with multiple VMM controllers instead of having multiple VMM domains.

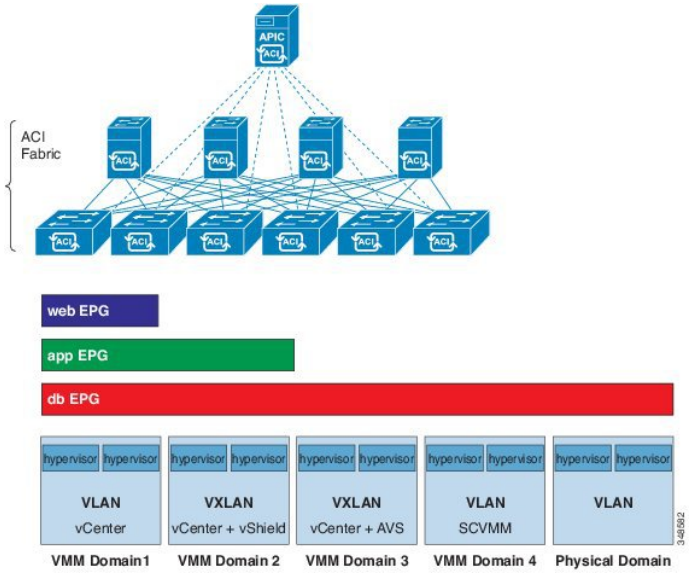
EPGs can use multiple VMM domains in the following ways:

- An EPG within a VMM domain is identified by using an encapsulation identifier. Cisco APIC can manage the identifier automatically, or the administrator can statically select it. An example is a VLAN, a Virtual Network ID (VNID).
- An EPG can be mapped to multiple physical (for baremetal servers) or virtual domains. It can use different VLAN or VNID encapsulations in each domain.

Note By default, the Cisco APIC dynamically manages the allocatiion of a VLAN for an EPG. VMware DVS administrators have the option to configure a specific VLAN for an EPG. In that case, the VLAN is chosen from a static allocation block within the pool that is associated with the VMM domain.

Applications can be deployed across VMM domains.

Figure 98: Multiple VMM Domains and Scaling of EPGs in the Fabric



While live migration of VMs within a VMM domain is supported, live migration of VMs across VMM domains is not supported.

Note When you change the VRF on a bridge domain that is linked to an EPG with an associated VMM domain, the port-group is deleted and then added back on vCenter. This results in the EPG being undeployed from the VMM domain. This is expected behavior.

Trunk Port Group

You use a trunk port group to aggregate the traffic of endpoint groups (EPGs) for VMware virtual machine manager (VMM) domains. Unlike regular port groups, which are configured under the Tenants tab in the Cisco Application Policy Infrastructure Controller (APIC) GUI, trunk port groups are configured under the VM Networking tab. Regular port groups follow the *T*|*A*|*E* format of EPG names.

The aggregation of EPGs under the same domain is based on a VLAN range, which is specified as encapsulation blocks contained in the trunk port group. Whenever the encapsulation of an EPG is changed or the encapsulation block of a trunk port group is changed, the aggregation is re-evaluated to determine if the EGP should be aggregated.

A trunk port group controls the leaf deployment of network resources, such as VLANs, that allocated to the EPGs being aggregated. The EPGs include both base EPG and microsegmented (uSeg) EPGs. In the case of a uSeg EPG, the VLAN ranges of the trunk port group are needed to include both the primary and secondary VLANs.

EPG Policy Resolution and Deployment Immediacy

Whenever an endpoint group (EPG) associates to a virtual machine manager (VMM) domain, the administrator can choose the resolution and deployment preferences to specify when a policy should be pushed into leaf switches.

Resolution Immediacy

- Pre-provision: Specifies that a policy (for example, VLAN, VXLAN binding, contracts, or filters) is downloaded to a leaf switch even before a VM controller is attached to the virtual switch (for example, VMware vSphere Distributed Switch (VDS). This pre-provisions the configuration on the switch.

  This helps the situation where management traffic for hypervisors/VM controllers is also using the virtual switch associated to the Cisco Application Policy Infrastructure Controller (APIC) VMM domain (VMM switch).

  Deploying a VMM policy such as VLAN on a Cisco Application Centric Infrastructure (ACI) leaf switch requires Cisco APICto collect CDP/LLDP information from both hypervisors through the VM controller and Cisco ACI leaf switch. However, if the VM controller is supposed to use the same VMM policy (VMM switch) to communicate with its hypervisors or even Cisco APIC, the CDP/LLDP information for hypervisors can never be collected because the policy that is required for VM controller/hypervisor management traffic is not deployed yet.

  When using pre-provision immediacy, policy is downloaded to Cisco ACI leaf switch regardless of CDP/LLDP neighborship. Even without a hypervisor host that is connected to the VMM switch.

- Immediate: Specifies that EPG policies (including contracts and filters) are downloaded to the associated leaf switch software upon ESXi host attachment to a DVS. LLDP or OpFlex permissions are used to resolve the VM controller to leaf node attachments.

  The policy will be downloaded to leaf when you add host to the VMM switch. CDP/LLDP neighborship from host to leaf is required.

- On Demand: Specifies that a policy (for example, VLAN, VXLAN bindings, contracts, or filters) is pushed to the leaf node only when an ESXi host is attached to a DVS and a VM is placed in the port group (EPG).

  The policy will be downloaded to the leaf when host is added to the VMM switch. The VM needs to be placed into a port group (EPG). CDP/LLDP neighborship from host to leaf is required.

  With both immediate and on demand, if host and leaf lose LLDP/CDP neighborship the policies are removed.

Note In OpFlex-based VMM domains, an OpFlex agent on the hypervisor reports a VM/EP virtual network interface card (vNIC) attachment to an EPG to the leaf OpFlex process. When using On Demand Resolution Immediacy, the EPG VLAN/VXLAN is programmed on **all** leaf port channel ports, virtual port channel ports, or both when the following are true:

- Hypervisors are connected to leafs on port channel or virtual port channel attached directly or through blade switches.
- A VM or instance vNIC is attached to an EPG.
- Hypervisors are attached as part of the EPG or VMM domain.

Opflex-based VMM domains are Microsoft Security Center Virtual Machine Manager (SCVMM) and HyperV, Cisco ACI Virtual Edge, and Cisco Application Virtual Switch (AVS).

Deployment Immediacy

Once the policies are downloaded to the leaf software, deployment immediacy can specify when the policy is pushed into the hardware policy content-addressable memory (CAM).

- Immediate: Specifies that the policy is programmed in the hardware policy CAM as soon as the policy is downloaded in the leaf software.
- On demand: Specifies that the policy is programmed in the hardware policy CAM only when the first packet is received through the data path. This process helps to optimize the hardware space.

Note When you use on demand deployment immediacy with MAC-pinned VPCs, the EPG contracts are not pushed to the leaf ternary content-addressble memory (TCAM) until the first endpoint is learned in the EPG on each leaf. This can cause uneven TCAM utilization across VPC peers. (Normally, the contract would be pushed to both peers.)

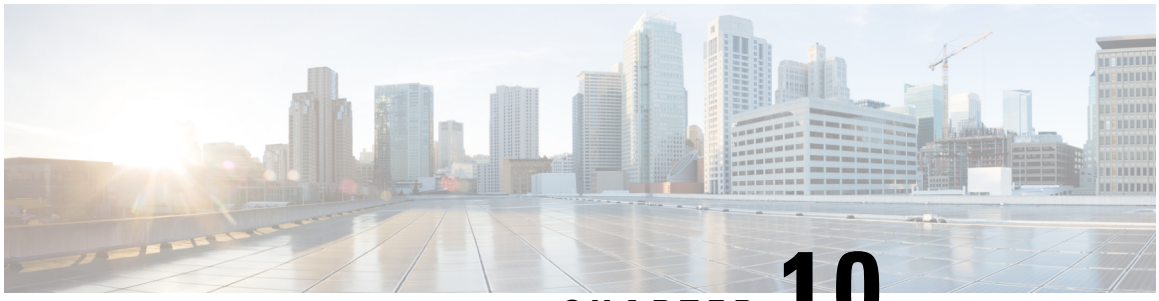
Guidelines for Deleting VMM Domains

Follow the sequence below to assure that the Cisco Application Policy Infrastructure Controller (APIC) request to delete a VMM domain automatically triggers the associated VM controller (for example VMware vCenter or Microsoft SCVMM) to complete the process normally, and that no orphan EPGs are stranded in the Cisco Application Centric Infrastructure (ACI) fabric.

1. The VM administrator must detach all the VMs from the port groups (in the case of VMware vCenter) or VM networks (in the case of SCVMM), created by the Cisco APIC.
2. The Cisco ACI administrator deletes the VMM domain in the Cisco APIC. The Cisco APIC triggers deletion of VMware VDS or SCVMM logical switch and associated objects.

Note The VM administrator should not delete the virtual switch or associated objects (such as port groups or VM networks); allow the Cisco APIC to trigger the virtual switch deletion upon completion of step 2 above. EPGs could be orphaned in the Cisco APIC if the VM administrator deletes the virtual switch from the VM controller before the VMM domain is deleted in the Cisco APIC.

If this sequence is not followed, the VM controller does delete the virtual switch associated with the Cisco APIC VMM domain. In this scenario, the VM administrator must manually remove the VM and vtep associations from the VM controller, then delete the virtual switch(es) previously associated with the Cisco APIC VMM domain.



CHAPTER 10

Layer 4 to Layer 7 Service Insertion

This chapter contains the following sections:

Layer 4 to Layer 7 Service Insertion

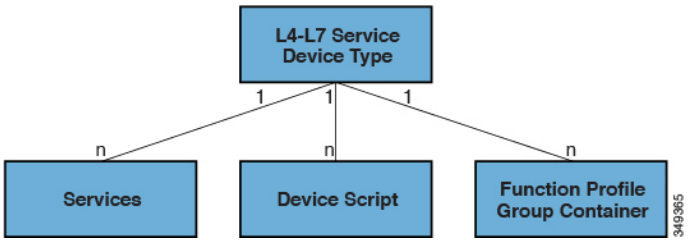
The Cisco Application Policy Infrastructure Controller (APIC) manages network services. Policies are used to insert services. APIC service integration provides a life cycle automation framework that enables the system to dynamically respond when a service comes online or goes offline. Shared services that are available to the entire fabric are administered by the fabric administrator. Services that are for a single tenant are administered by the tenant administrator.

The APIC provides automated service insertion while acting as a central point of policy control. APIC policies manage both the network fabric and services appliances. The APIC can configure the network automatically so that traffic flows through the services. Also, the APIC can automatically configure the service according to the application's requirements. This approach allows organizations to automate service insertion and eliminate the challenge of managing all of the complex traffic-steering techniques of traditional service insertion.

Layer 4 to Layer 7 Policy Model

The Layer 4 to Layer 7 service device type policies includes key managed objects such as services supported by the package and device scripts. The following figure shows the objects of the Layer 4 to Layer 7 service device type policy model.

Figure 99: Layer 4 to Layer 7 Policy Model



Layer 4 to Layer 7 service policies contain the following:

- **Services**—Contains metadata for all the functions provided by a device such as SSL offloading and load-balancing. This MO contains the connector names, encapsulation type, such as VLAN and VXLAN, and any interface labels.
- **Device Script**—Represents a device script handler that contains meta information about the related attributes of the script handler including its name, package name, and version.
- **Function Profile Group Container**—Objects that contain the functions available to the service device type. Function profiles contain all the configurable parameters supported by the device organized into folders.

About Service Graphs

The Cisco Application Centric Infrastructure (ACI) treats services as an integral part of an application. Any services that are required are treated as a service graph that is instantiated on the ACI fabric from the Cisco Application Policy Infrastructure Controller (APIC). Users define the service for the application, while service graphs identify the set of network or service functions that are needed by the application.

A service graph represents the network using the following elements:

- Function node—A function node represents a function that is applied to the traffic, such as a transform (SSL termination, VPN gateway), filter (firewalls), or terminal (intrusion detection systems). A function within the service graph might require one or more parameters and have one or more connectors.
- Terminal node—A terminal node enables input and output from the service graph.
- Connector—A connector enables input and output from a node.
- Connection—A connection determines how traffic is forwarded through the network.

After the graph is configured in the APIC, the APIC automatically configures the services according to the service function requirements that are specified in the service graph. The APIC also automatically configures the network according to the needs of the service function that is specified in the service graph, which does not require any change in the service device.

A service graph is represented as two or more tiers of an application with the appropriate service function inserted between.

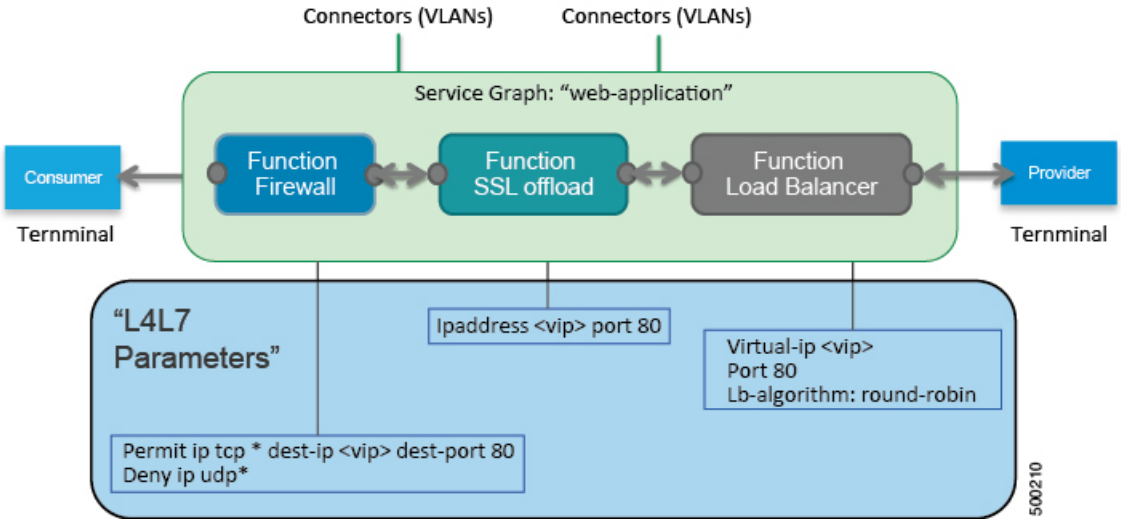
A service appliance (device) performs a service function within the graph. One or more service appliances might be required to render the services required by a graph. One or more service functions can be performed by a single-service device.

Service graphs and service functions have the following characteristics:

- Traffic sent or received by an endpoint group can be filtered based on a policy, and a subset of the traffic can be redirected to different edges in the graph.
- Service graph edges are directional.
- Taps (hardware-based packet copy service) can be attached to different points in the service graph.
- Logical functions can be rendered on the appropriate (physical or virtual) device, based on the policy.
- The service graph supports splits and joins of edges, and it does not restrict the administrator to linear service chains.
- Traffic can be reclassified again in the network after a service appliance emits it.
- Logical service functions can be scaled up or down or can be deployed in a cluster mode or 1:1 active-standby high-availability mode, depending on the requirements.

The following figure provides an example of a service graph deployment:

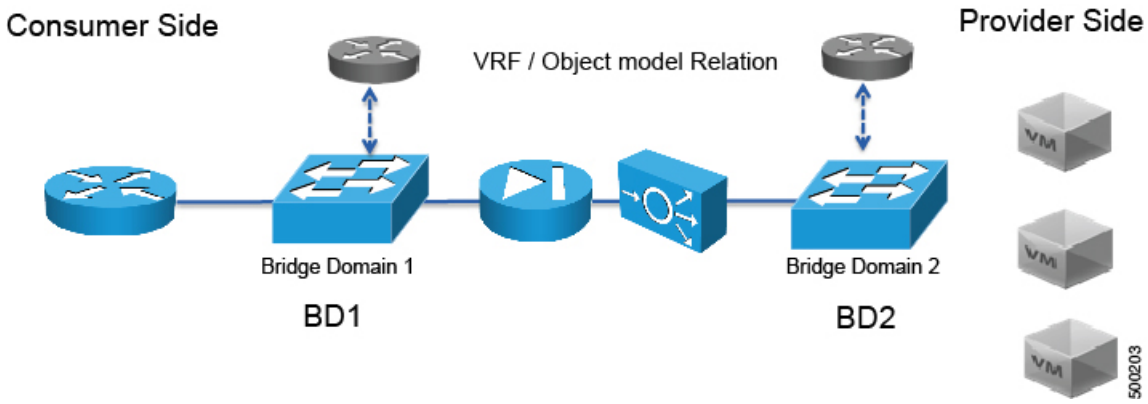
Figure 100: Example Service Graph Deployment



By using a service graph, you can install a service, such as an ASA firewall, once and deploy it multiple times in different logical topologies. Each time the graph is deployed, ACI takes care of changing the configuration on the firewall to enable the forwarding in the new logical topology.

Deploying a service graph requires bridge domains and VRFs, as shown in the following figure:

Figure 101: Bridge Domains and VRFs of a Service Graph



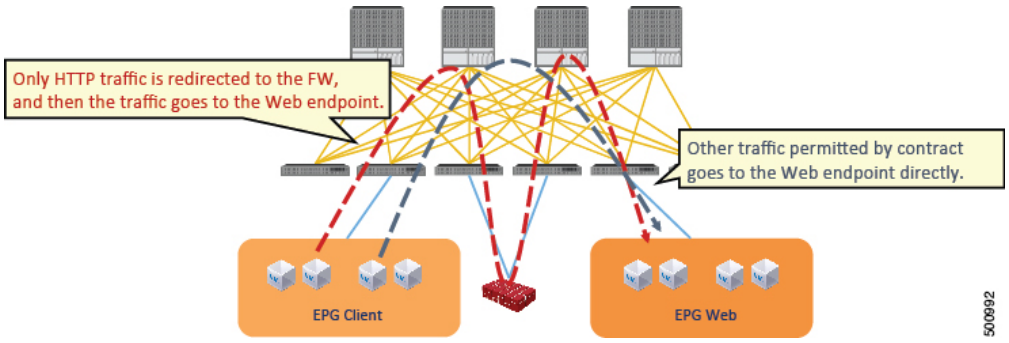
Note If you have some of the legs of a service graph that are attached to endpoint groups in other tenants, when you use the **Remove Related Objects of Graph Template** function in the GUI, the APIC does not remove contracts that were imported from tenants other than where the service graph is located. The APIC also does not clean endpoint group contracts that are located in a different tenant than the service graph. You must manually remove these objects that are in different tenants.

About Policy-Based Redirect

Cisco Application Centric Infrastructure (ACI) policy-based redirect (PBR) enables provisioning service appliances, such as firewalls or load balancers, as managed or unmanaged nodes without needing a Layer 4 to Layer 7 package. Typical use cases include provisioning service appliances that can be pooled, tailored to application profiles, scaled easily, and have reduced exposure to service outages. PBR simplifies the deployment of service appliances by enabling the provisioning consumer and provider endpoint groups to be all in the same virtual routing and forwarding (VRF) instance. PBR deployment consists of configuring a route redirect policy and a cluster redirect policy, and creating a service graph template that uses the route and cluster redirect policies. After the service graph template is deployed, use the service appliance by enabling endpoint groups to consume the service graph provider endpoint group. This can be further simplified and automated by using vzAny. While performance requirements may dictate provisioning dedicated service appliances, virtual service appliances can also be deployed easily using PBR.

The following figure illustrates the use case of redirecting specific traffic to the firewall:

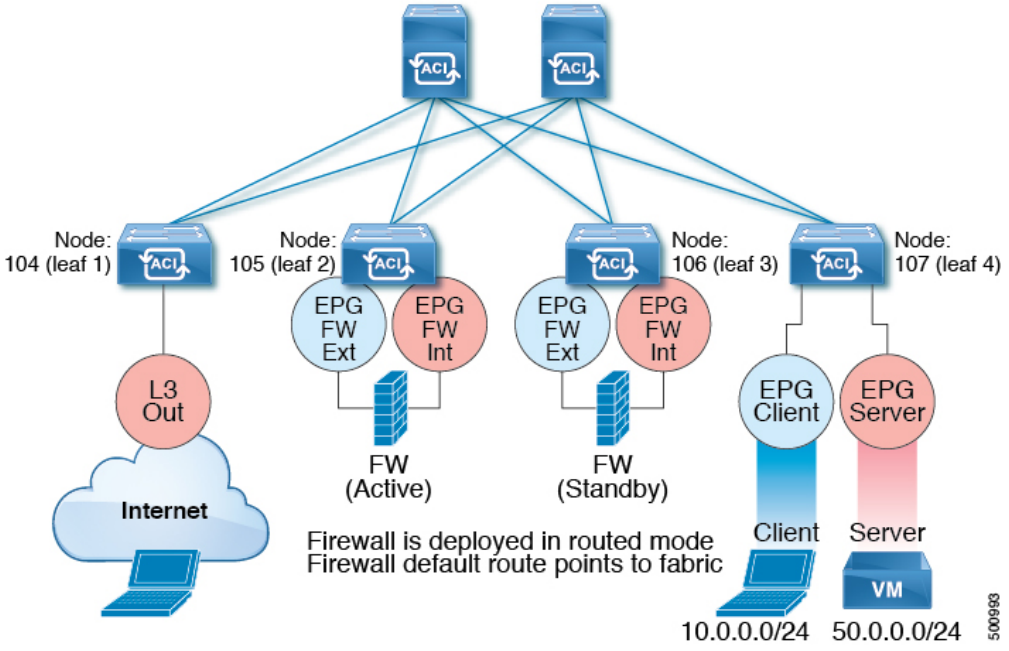
Figure 102: Use Case: Redirecting Specific Traffic to the Firewall



In this use case, you must create two subjects. The first subject permits HTTP traffic, which then gets redirected to the firewall. After the traffic passes through the firewall, it goes to the Web endpoint. The second subject permits all traffic, which captures traffic that is not redirected by the first subject. This traffic goes directly to the Web endpoint.

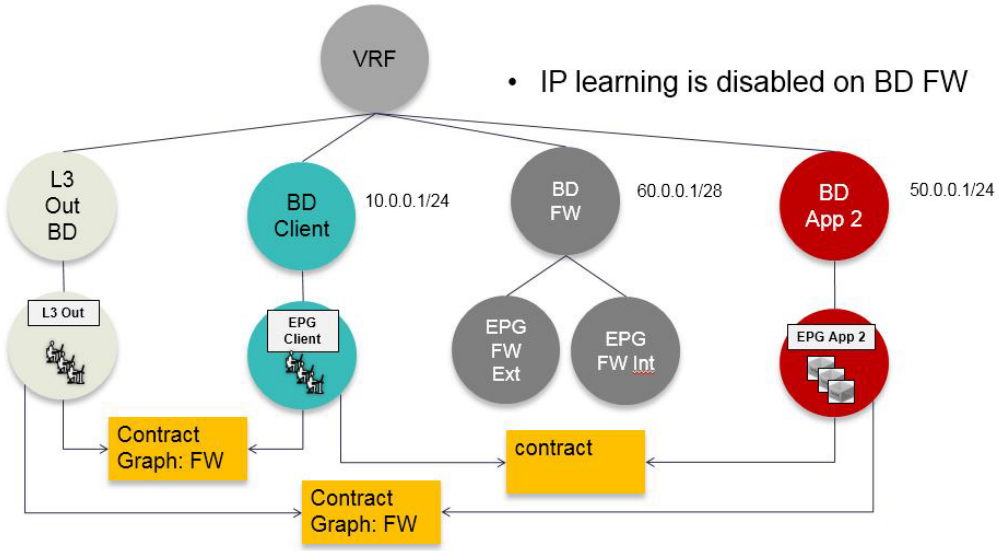
The following figure illustrates a sample ACI PBR physical topology:

Figure 103: Sample ACI PBR Physical Topology



The following figure illustrates a sample ACI PBR logical topology:

Figure 104: Sample ACI PBR Logical Topology



While these examples illustrate simple deployments, ACI PBR enables scaling up mixtures of both physical and virtual service appliances for multiple services, such as firewalls and server load balancers.

About Symmetric Policy-Based Redirect

Symmetric policy-based redirect (PBR) configurations enable provisioning a pool of service appliances so that the consumer and provider endpoint groups traffic is policy-based. The traffic is redirected to one of the service nodes in the pool, depending on the source and destination IP equal-cost multi-path routing (ECMP) prefix hashing.

Note Symmetric PBR configurations require 9300-EX hardware.

Sample symmetric PBR REST posts are listed below:

```
Under fvTenant svcCont

<vnsSvcRedirectPol name="LoadBalancer_pool">
          <vnsRedirectDest name="lb1" ip="1.1.1.1" mac="00:00:11:22:33:44"/>
          <vnsRedirectDest name="lb2" ip="2.2.2.2" mac="00:de:ad:be:ef:01"/>
          <vnsRedirectDest name="lb3" ip="3.3.3.3" mac="00:de:ad:be:ef:02"/>
</vnsSvcRedirectPol>

<vnsLIfCtx name="external">
          <vnsRsSvcRedirectPol tnVnsSvcRedirectPolName="LoadBalancer_pool"/>
          <vnsRsLIfCtxToBD tDn="uni/tn-solar/bd-fwBD">
</vnsLIfCtx>

<vnsAbsNode name="FW" routingMode="redirect">
```

Sample symmetric PBR NX-OS-style CLI commands are listed below.

The following commands under the tenant scope create a service redirect policy:

```
apic1(config-tenant)# svcredir-pol fw-external
apic1(svcredir-pol)# redir-dest 2.2.2.2 00:11:22:33:44:56
```

The following commands enable PBR:

```
apic1(config-tenant)# l4l7 graph FWOnly contract default
apic1(config-graph)# service FW svcredir enable
```

The following commands set the redirect policy under the device selection policy connector:

```
apic1(config-service)# connector external
apic1(config-connector)# svcredir-pol tenant solar name fw-external
```

Automated Service Insertion

Although VLAN and virtual routing and forwarding (VRF) stitching is supported by traditional service insertion models, the Application Policy Infrastructure Controller (APIC) can automate service insertion and the provisioning of network services, such as Secure Sockets Layer (SSL) offload, server load balancing (SLB), Web Application firewall (WAF), and firewall, while acting as a central point of policy control. The network services are typically rendered by service appliances, such as Application Delivery Controllers (ADCs) and firewalls. The APIC policies manage both the network fabric and services appliances. The APIC can configure the network automatically so that traffic flows through the services. The APIC can also automatically configure the service according to the application's requirements, which allows organizations to automate service insertion and eliminate the challenge of managing the complex techniques of traditional service insertion.

About Device Packages

The Application Policy Infrastructure Controller (APIC) requires a device package to configure and monitor service devices. You add service functions to the APIC through the device package. A device package manages a single class of service devices and provides the APIC with information about the device and its capabilities. A device package is a `zip` file that contains the following parts:

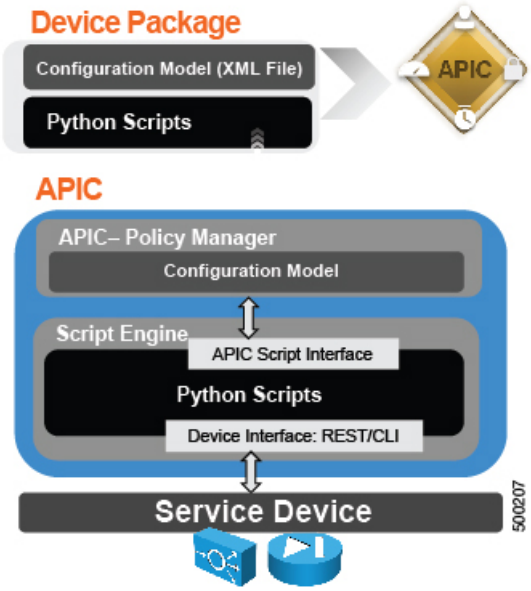
Device specification	An XML file that defines the following:
	• Device properties:
	  • **Model**—Model of the device.
	  • **Vendor**—Vendor of the device.
	  • **Version**—Software version of the device.
	• Functions provided by a device, such as load balancing, content switching, and SSL termination.
	• Interfaces and network connectivity information for each function.
	• Device configuration parameters.
	• Configuration parameters for each function.

Device script	A Python script that interacts with the device from the APIC. APIC events are mapped to function calls that are defined in the device script. A device package can contain multiple device scripts. A device script can interface with the device by using REST, SSH, or any similar mechanism.
Function profile	Function parameters with default values that are specified by the vendor. You can configure a function to use these default values.
Device-level configuration parameters	A configuration file that specifies parameters that are required by a device. This configuration can be shared by one or more graphs using a device.

You can create a device package or it can be provided by a device vendor or Cisco.

The following figure illustrates the interaction of a device package and the APIC:

Figure 105: Device Package and the APIC

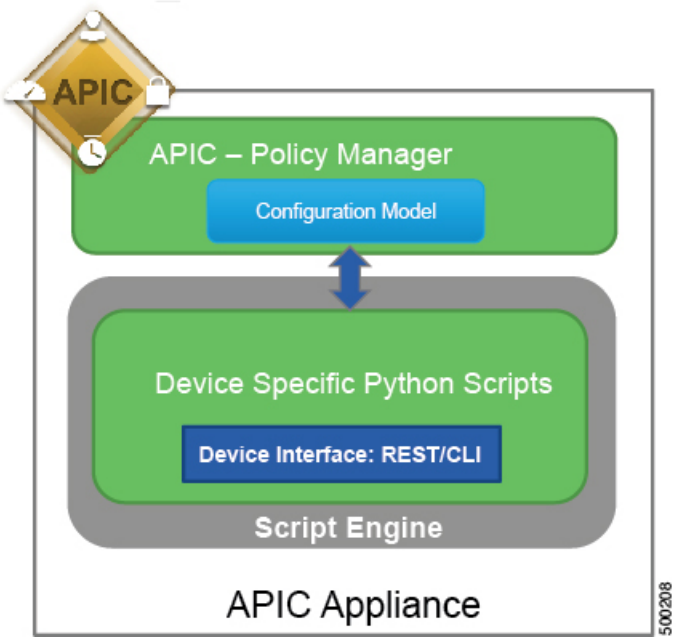


The functions in a device script are classified into the following categories:

- Device/Infrastructure—For device level configuration and monitoring
- Service Events—For configuring functions, such as a server load balancer or Secure Sockets Layer, on the device
- Endpoint/Network Events—For handling endpoint and network attach/detach events

The APIC uses the device configuration model that is provided in the device package to pass the appropriate configuration to the device scripts. The device script handlers interface with the device using its REST or CLI interface.

Figure 106: How the Device Scripts Interface with a Service Device



The device package enables an administrator to automate the management of the following services:

- Device attachment and detachment
- Endpoint attachment and detachment
- Service graph rendering
- Health monitoring
- Alarms, notifications, and logging
- Counters

For more information about device packages and how to develop a device package, see *Cisco APIC Layer 4 to Layer 7 Device Package Development Guide*

About Device Clusters

A device cluster (also known as a logical device) is one or more concrete devices that act as a single device. A device cluster has cluster (logical) interfaces, which describe the interface information for the device cluster. During service graph template rendering, function node connectors are associated with cluster (logical) interfaces. The Application Policy Infrastructure Controller (APIC) allocates the network resources (VLAN or Virtual Extensible Local Area Network [VXLAN]) for a function node connector during service graph template instantiation and rendering and programs the network resources onto the cluster (logical) interfaces.

The service graph template uses a specific device that is based on a device selection policy (called a *logical device context*) that an administrator defines.

An administrator can set up a maximum of two concrete devices in active-standby mode.

To set up a device cluster, you must perform the following tasks:

1. Connect the concrete devices to the fabric.
2. Assign the management IP address to the device cluster.
3. Register the device cluster with the APIC. The APIC validates the device using the device specifications from the device package.

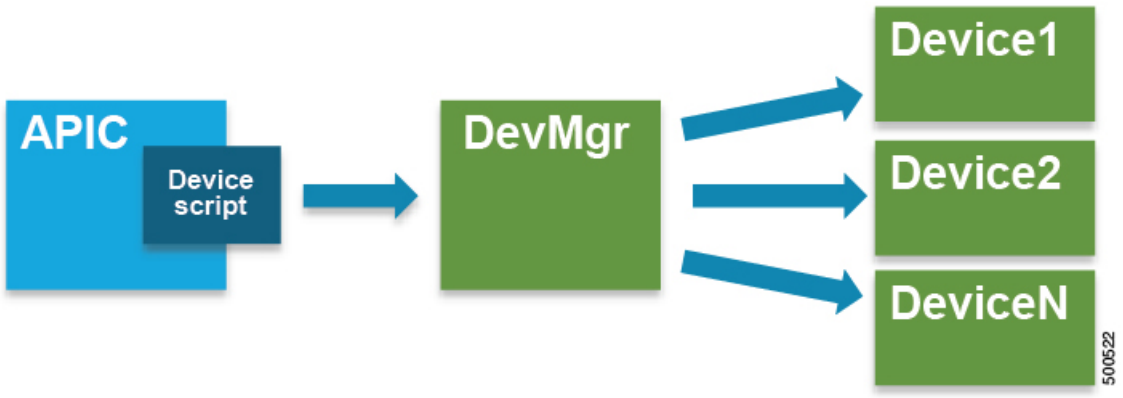
Note The APIC does not validate a duplicate IP address that is assigned to two device clusters. The APIC can provision the wrong device cluster when two device clusters have the same management IP address. If you have duplicate IP addresses for your device clusters, delete the IP address configuration on one of the devices and ensure there are no duplicate IP addresses that are provisioned for the management IP address configuration.

About Device Managers and Chassis Managers

A device manager serves as a single point of configuration for a set of clusters in a Cisco Application Centric Infrastructure (ACI) fabric. The administration or operational state is presented in the native GUI of the devices. A device manager handles configuration on individual devices and enables you to simplify the configuration in the Application Policy Infrastructure Controller (APIC). You create a template in device manager, then populate the device manager with instance-specific values from the APIC, which requires only a few values.

The following figure illustrates a device manager controlling multiple devices in a cluster:

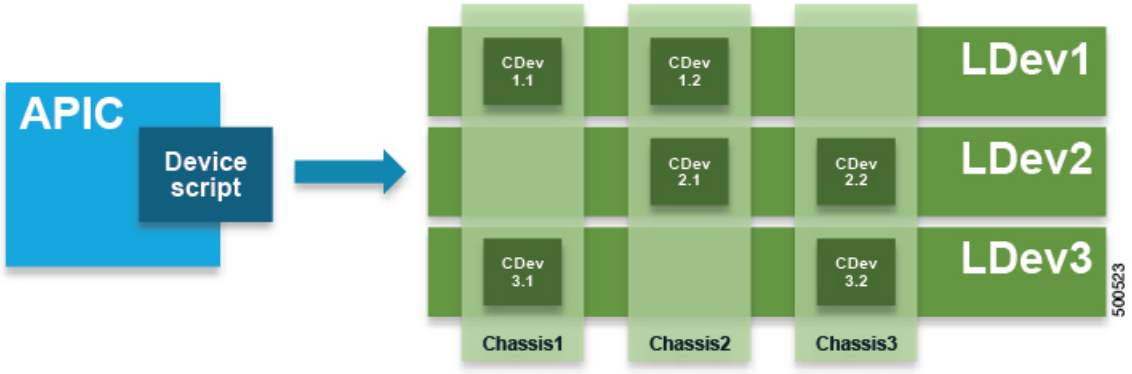
Figure 107: Controlling Devices with a Device Manager



A chassis manager is a physical or virtual "container" of processing resources. A chassis manager supports a number of virtual service devices that are represented as `CDev` objects. A chassis manager handles networking, while `CDev` handles processing. A chassis manager enables the on-demand creation of virtual processing nodes. For a virtual device, some parts of a service (specifically the VLANs) must be applied to the chassis rather than to the virtual machine. To accomplish this, the chassis management IP address and credentials must be included in callouts.

The following figure illustrates a chassis manager acting as a container of processing resources:

Figure 108: Controlling Devices with a Device Manager

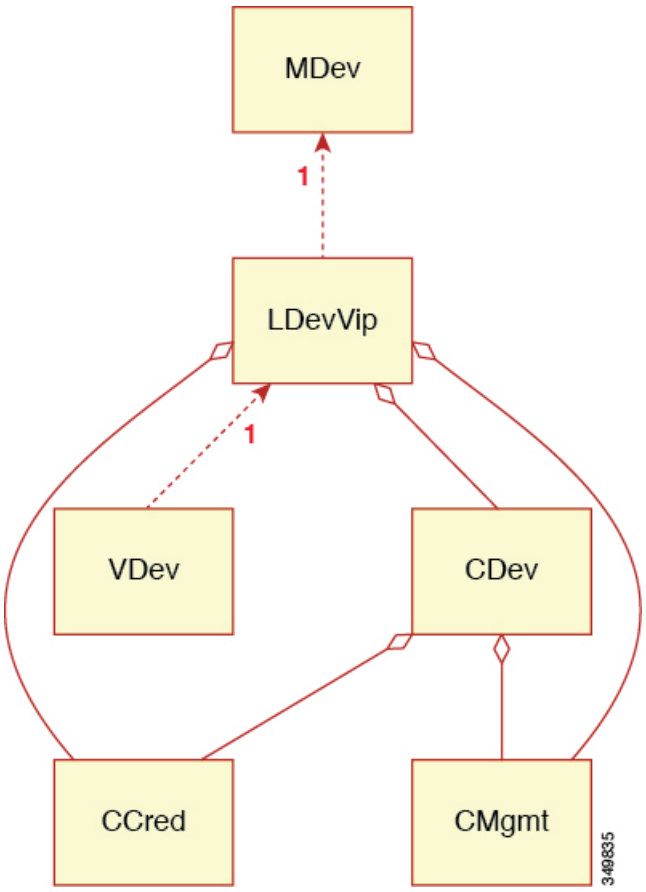


Without a device manager or chassis manager, the model for service devices contains the following key managed objects:

- `MDev`—Represents a device type (vendor, model, version).
- `LDevVIP`—Represents a cluster, a set of identically configured devices for Cold Standby. Contains `CMgmt` and `CCred` for access to the device.
- `CDev`—Represents a member of a cluster, either physical or virtual. Contains `CMgmt` and `CCred` for access to the device.
- `VDev`—Represents a context on a cluster, similar to a virtual machine on a server.

The following figure illustrates the model for the key managed objects, with `CMgmt` (management connectivity) and `CCred` (credentials) included:

Figure 109: Managed Object Model Without a Device Manager or Chassis Manager



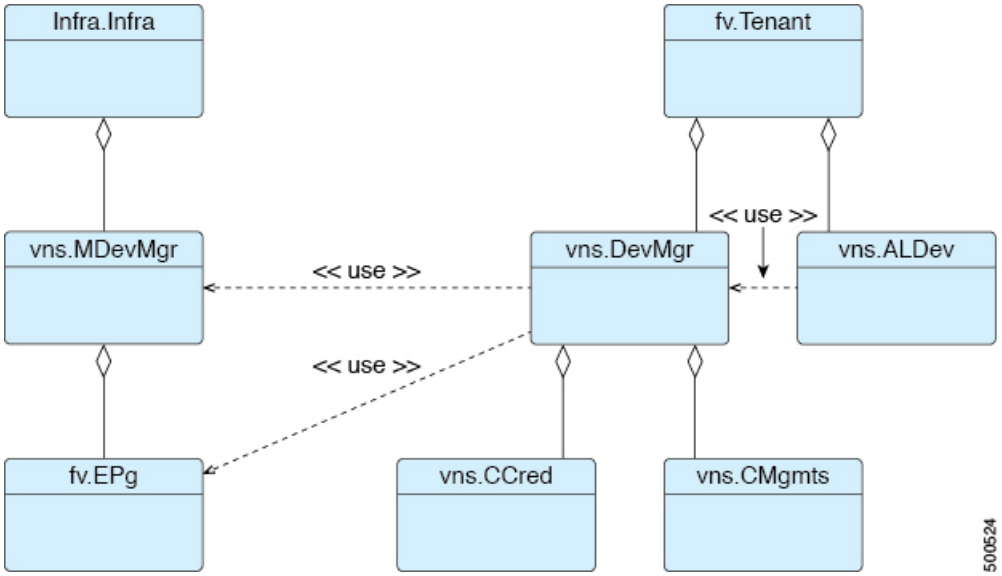
`CMgmt` (host + port) and `CCred` (username + password) allow the script to access the device and cluster.

A device manager and chassis manager adds the ability to control the configuration of clusters and devices from a central management station. The chassis adds a parallel hierarchy to the `MDev` object and `ALDev` object to allow a `CDev` object to be tagged as belonging to a specific chassis. The following managed objects are added to the model to support the device and chassis manager concept:

- `MDevMgr`—Represents a type of device manager. An `MDevMgr` can manage a set of different `MDev`s, which are typically different products from the same vendor.
- `DevMgr`—Represents a device manager. Access to the manager is provided using the contained `CMgmt` and `CCred` managed objects. Each cluster can be associated with only one `DevMgr`.
- `MChassis`—Represents a type of chassis. This managed object is typically included in the package.
- `Chassis`—Represents a chassis instance. It contains the `CMgmt` and `CCred`[Secret] managed objects to provide connectivity to the chassis.

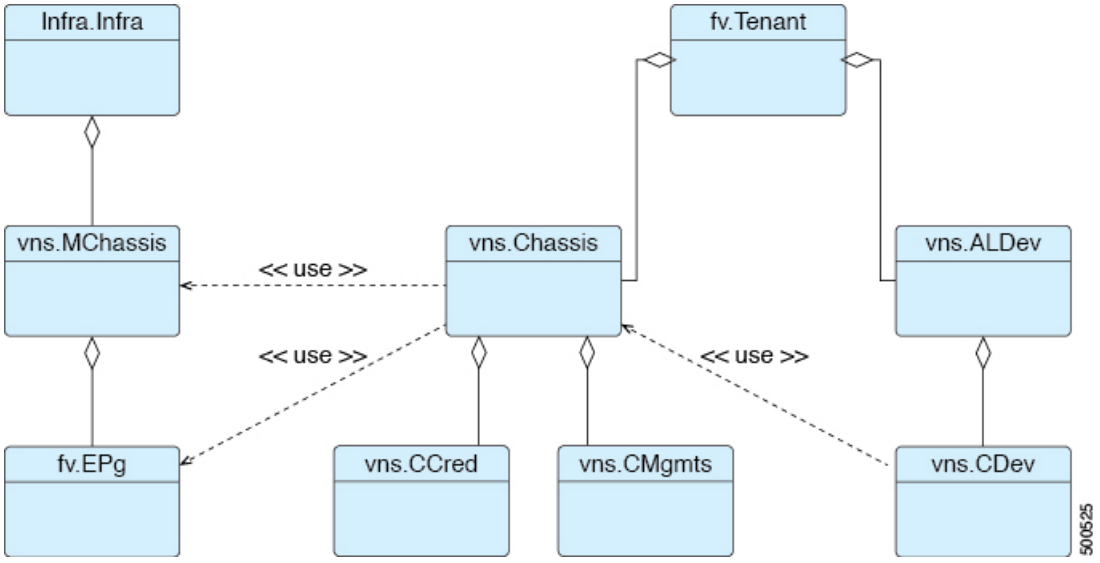
The following figure illustrates the device manager object model:

Figure 110: Device Manager Object Model



The following figure illustrates the chassis manager object model:

Figure 111: Chassis Manager Object Model



About Concrete Devices

A concrete device can be either physical or virtual. If the device is virtual, you must choose the controller (vCenter or SCVMM controller) and the virtual machine name. A concrete device has concrete interfaces. When a concrete device is added to a logical device, the concrete interfaces are mapped to the logical interfaces. During service graph template instantiation, VLANs and VXLANs are programmed on concrete interfaces that are based on their association with logical interfaces.

About Function Nodes

A function node represents a single service function. A function node has function node connectors, which represent the network requirement of a service function.

A function node within a service graph can require one or more parameters. The parameters can be specified by an endpoint group (EPG), an application profile, or a tenant VRF. Parameters can also be assigned at the time that you define a service graph. The parameter values can be locked to prevent any additional changes.

Note For Multi-Site configuration, up to 2 nodes can be deployed in a service graph. For non-Multi-Site configuration, up to 5 nodes can be deployed in a service graph.

About Function Node Connectors

A function node connector connects a function node to the service graph and is associated with the appropriate bridge domain and connections based on the graph's connector's subset. Each connector is associated with a VLAN or Virtual Extensible LAN (VXLAN). Each side of a connector is treated as an endpoint group (EPG), and whitelists are downloaded to the switch to enable communication between the two function nodes.

About Terminal Nodes

Terminal nodes connect a service graph with the contracts. You can insert a service graph for the traffic between two application endpoint groups (EPGs) by connecting the terminal node to a contract. Once connected, traffic between the consumer EPG and provider EPG of the contract is redirected to the service graph.

About Privileges

An administrator can grant privileges to the roles in the APIC. Privileges determine what tasks a role is allowed to perform. Administrators can grant the following privileges to the administrator roles:

Privilege	Description
nw-svc-connectivity	• Create a management EPG • Create management connectivity to other objects
nw-svc-policy	• Create a service graph • Attach a service graph to an application EPG and a contract • Monitor a service graph
nw-svc-device	• Create a device cluster • Create a concrete device • Create a device context

Note Only an infrastructure administrator can upload a device package to the APIC.

Service Automation and Configuration Management

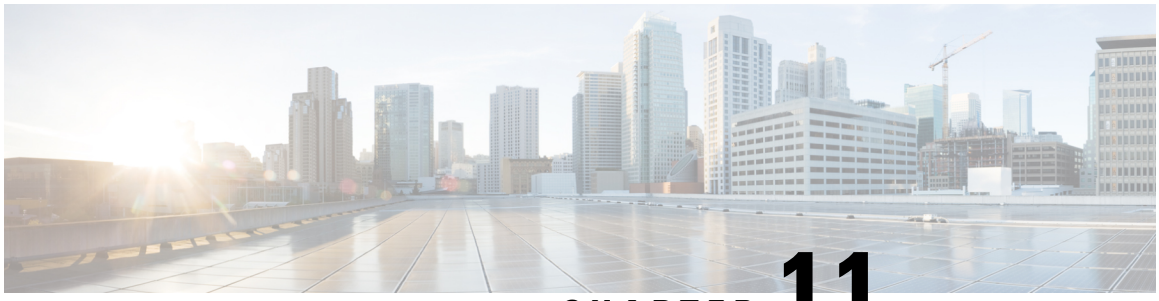
The Cisco APIC can optionally act as a point of configuration management and automation for service devices and coordinate the service devices with the network automation. The Cisco APIC interfaces with a service device by using Python scripts and calls device-specific Python script functions on various events.

The device scripts and a device specification that defines functions supported by the service device are bundled as a device package and installed on the Cisco APIC. The device script handlers interface with the device by using its REST interface (preferred) or CLI based on the device configuration model.

Service Resource Pooling

The Cisco ACI fabric can perform nonstateful load distribution across many destinations. This capability allows organizations to group physical and virtual service devices into service resource pools, which can be further grouped by function or location. These pools can offer high availability by using standard high-availability mechanisms or they can be used as simple stateful service engines with the load redistributed to the other members if a failure occurs. Either option provides horizontal scale out that far exceeds the current limitations of the equal-cost multipath (ECMP), port channel features, and service appliance clustering, which requires a shared state.

Cisco ACI can perform a simple version of resource pooling with any service devices if the service devices do not have to interact with the fabric, and it can perform more advanced pooling that involves coordination between the fabric and the service devices.



CHAPTER 11

Management Tools

This chapter contains the following sections:

Management Tools

Cisco Application Centric Infrastructure (ACI) tools help fabric administrators, network engineers, and developers to develop, configure, debug, and automate the deployment of tenants and applications.

About the Management GUI

The following management GUI features provide access to the fabric and its components (leaves and spines):

- Based on universal web standards (HTML5). No installers or plugins are required.
- Access to monitoring (statistics, faults, events, audit logs), operational and configuration data.
- Access to the APIC and spine and leaf switches through a single sign-on mechanism.
- Communication with the APIC using the same RESTful APIs that are available to third parties.

About the CLI

The CLI features an operational and configuration interface to the APIC, leaf, and spine switches:

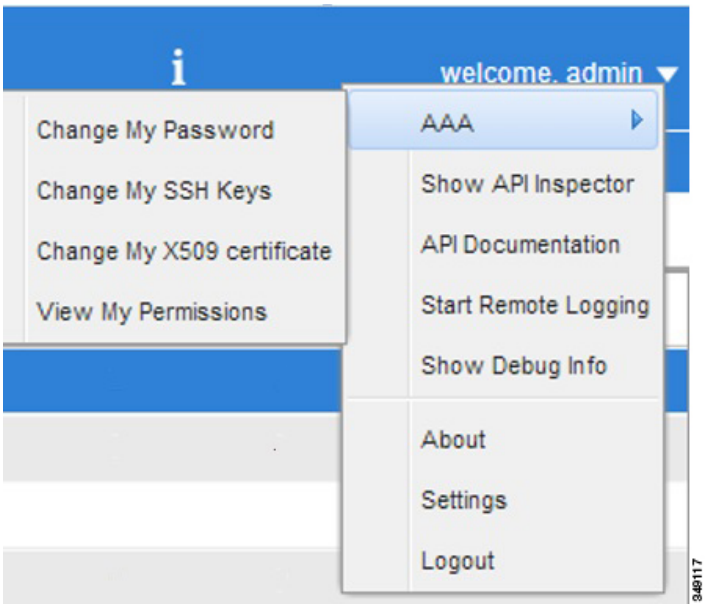
- Implemented from the ground up in Python; can switch between the Python interpreter and CLI
- Plugin architecture for extensibility

- Virtual Routing and Forwarding (VRF)-based access to monitoring, operation, and configuration data
- Automation through Python commands or batch scripting

User Login Menu Options

The user login drop-down menu provides several configuration, diagnostic, reference, and preference options. The figure below shows this drop-down menu.

Figure 112: User Login Menu Options



The options include the following:

- AAA options for changing the user password, SSH Keys, X509 Ce.jpgicate, and viewing the permissions of the logged-on user.

Note The ACI fabric must be configured with an active Network Time Protocol (NTP) policy to assure that the system clocks on all devices are correct. Otherwise, a ce.jpgicate could be rejected on nodes with out-of-sync time.

- Show API Inspector opens the API Inspector.
- API Documentation opens the Management Information Model reference.
- Remote Logging.
- Debug information.
- About the current version number of the software.
- Settings preferences for using the GUI.

- Logout to exit the system.

Customizing the GUI and CLI Banners

GUI and CLI banners can be in the Admin > AAA > Security management section of the GUI. The CLI banner displays before user login authentication. The CLI banner is a text based string printed as-is to the console. The GUI banner displays before user login authentication. The GUI banner is a URL. The URL must allow the being placed in an iFrame. If the URL `x-frame-option` is set to `deny` or `sameorigin`, the URL will not appear before user login authentication.

REST API

About the REST API

The Application Policy Infrastructure Controller (APIC) REST API is a programmatic interface that uses REST architecture. The API accepts and returns HTTP (not enabled by default) or HTTPS messages that contain JavaScript Object Notation (JSON) or Extensible Markup Language (XML) documents. You can use any programming language to generate the messages and the JSON or XML documents that contain the API methods or Managed Object (MO) descriptions.

The REST API is the interface into the management information tree (MIT) and allows manipulation of the object model state. The same REST interface is used by the APIC CLI, GUI, and SDK, so that whenever information is displayed, it is read through the REST API, and when configuration changes are made, they are written through the REST API. The REST API also provides an interface through which other information can be retrieved, including statistics, faults, and audit events. It even provides a means of subscribing to push-based event notification, so that when a change occurs in the MIT, an event can be sent through a web socket.

Standard REST methods are supported on the API, which includes POST, GET, and DELETE operations through HTTP. The POST and DELETE methods are idempotent, meaning that there is no additional effect if they are called more than once with the same input parameters. The GET method is nullipotent, meaning that it can be called zero or more times without making any changes (or that it is a read-only operation).

Payloads to and from the REST interface can be encapsulated through either XML or JSON encoding. In the case of XML, the encoding operation is simple: the element tag is the name of the package and class, and any properties of that object are specified as attributes of that element. Containment is defined by creating child elements.

For JSON, encoding requires definition of certain entities to reflect the tree-based hierarchy; however, the definition is repeated at all levels of the tree, so it is fairly simple to implement after it is initially understood.

- All objects are described as JSON dictionaries, in which the key is the name of the package and class. The value is another nested dictionary with two keys: attribute and children.
- The attribute key contains a further nested dictionary describing key-value pairs that define attributes on the object.
- The children key contains a list that defines all the child objects. The children in this list are dictionaries containing any nested objects, which are defined as described here.

Authentication

REST API username- and password-based authentication uses a special subset of request Universal Resource Identifiers (URIs), including **aaaLogin**, **aaaLogout**, and **aaaRefresh** as the DN targets of a POST operation. Their payloads contain a simple XML or JSON payload containing the MO representation of an **aaaUser** object with the attribute name and **pwd** defining the username and password: for example, **<aaaUser name='admin' pwd='password'/>**. The response to the POST operation will contain an authentication token as both a Set-Cookie header and an attribute to the **aaaLogin** object in the response named token, for which the XPath is **/imdata/aaaLogin/@token** if the encoding is XML. Subsequent operations on the REST API can use this token value as a cookie named **APIC-cookie** to authenticate future requests.

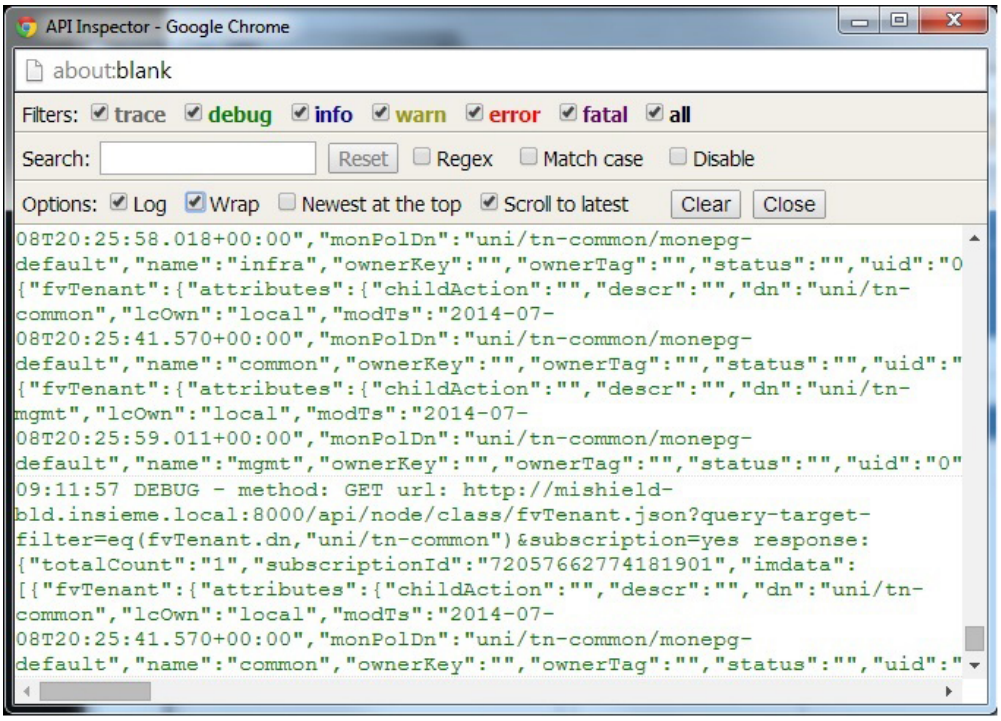
Subscription

The REST API supports the subscription to one or more MOs during your active API session. When any MO is created, changed, or deleted because of a user- or system-initiated action, an event is generated. If the event changes the data on any of the active subscribed queries, the APIC will send out a notification to the API client that created the subscription.

API Inspector

The API Inspector provides a real-time display of REST API commands that the APIC processes to perform GUI interactions. The figure below shows REST API commands that the API Inspector displays upon navigating to the main tenant section of the GUI.

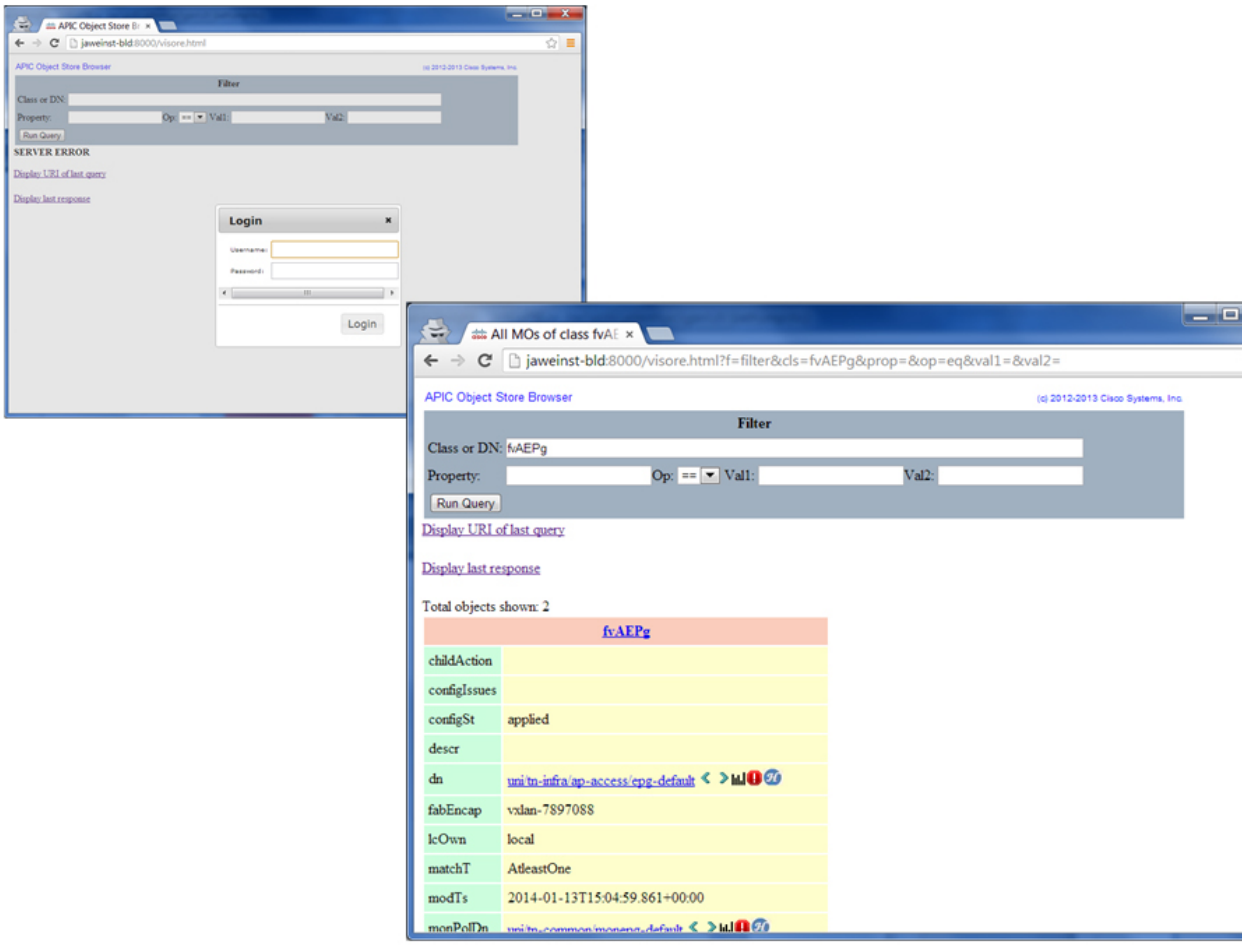
Figure 113: API Inspector



Visore Managed Object Viewer

Visore is a read-only management information tree (MIT) browser as shown in the figure below. It enables distinguished name (DN) and class queries with optional filters.

Figure 114: Visore MO Viewer



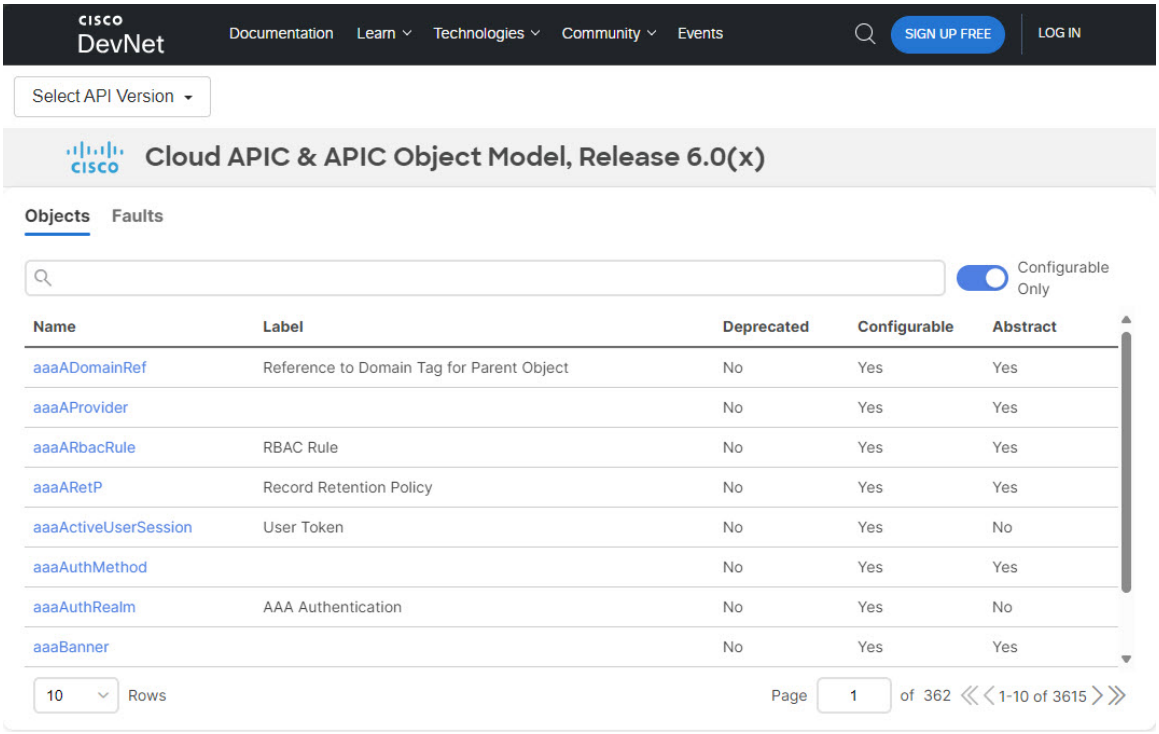
The Visore managed object viewer is at this location: `http(s)://host[:port]/visore.html`

Management Information Model Reference

The Management Information Model (MIM) contains all of the managed objects in the system and their properties. For details, see the *Cisco APIC Management Information Model Reference Guide*.

See the following figure for an example of how an administrator can use the MIM to research an object in the MIT in the Cisco APIC 6.0 releases.

Figure 115: MIM Reference for the Cisco APIC 6.0 Releases

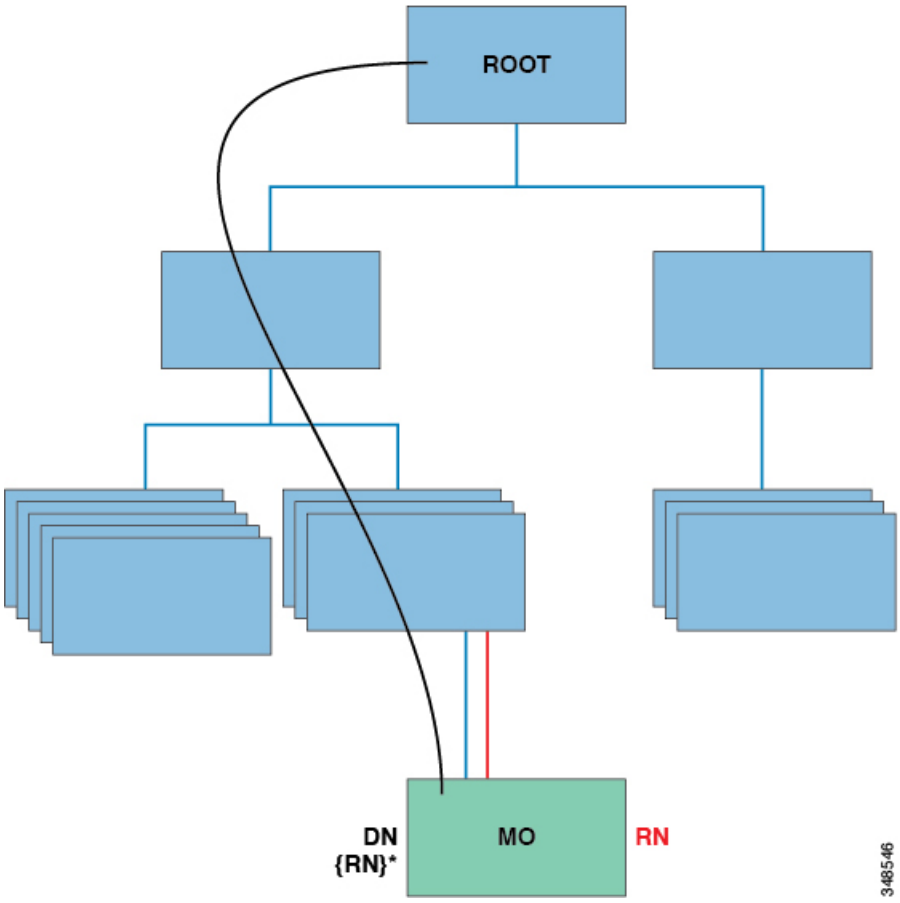


Locating Objects in the MIT

The Cisco ACI uses an information-model-based architecture (management information tree [MIT]) in which the model describes all the information that can be controlled by a management process. Object instances are referred to as managed objects (MOs).

The following figure shows the distinguished name, which uniquely represents any given MO instance, and the relative name, which represents the MO locally underneath its parent MO. All objects in the MIT exist under the root object.

Figure 116: MO Distinguished and Relative Names



Every MO in the system can be identified by a unique distinguished name (DN). This approach allows the object to be referred to globally. In addition to its distinguished name, each object can be referred to by its relative name (RN). The relative name identifies an object relative to its parent object. Any given object's distinguished name is derived from its own relative name that is appended to its parent object's distinguished name.

A DN is a sequence of relative names that uniquely identifies an object:

```
dn = {rn}/{rn}/{rn}/{rn}
```

```
dn =”sys/ch/lcslot-1/lc/leafport-1”
```

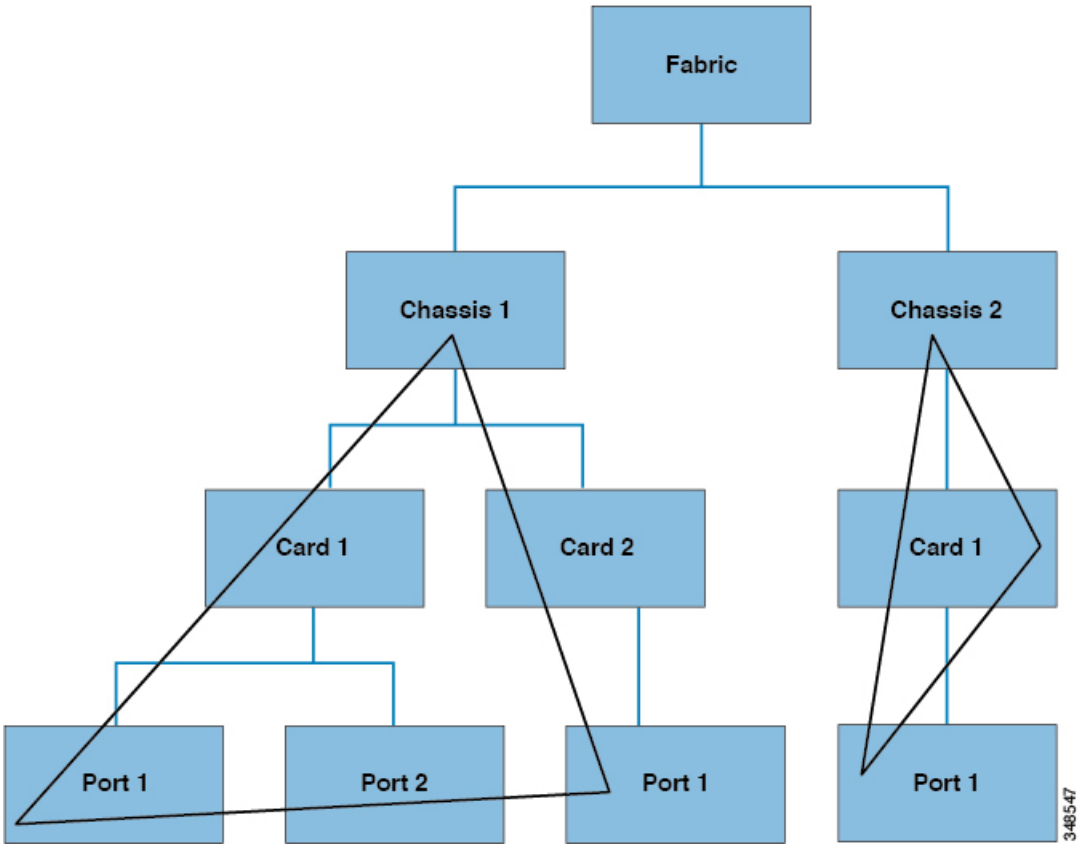
Distinguished names are directly mapped to URLs. Either the relative name or the distinguished name can be used to access an object, depending on the current location in the MIT.

Because of the hierarchical nature of the tree and the attribute system used to identify object classes, the tree can be queried in several ways for obtaining managed object information. Queries can be performed on an object itself through its distinguished name, on a class of objects such as a switch chassis, or on a tree-level to discover all members of an object.

Tree-Level Queries

The following figure shows two chassis that are queried at the tree level.

Figure 117: Tree-Level Queries

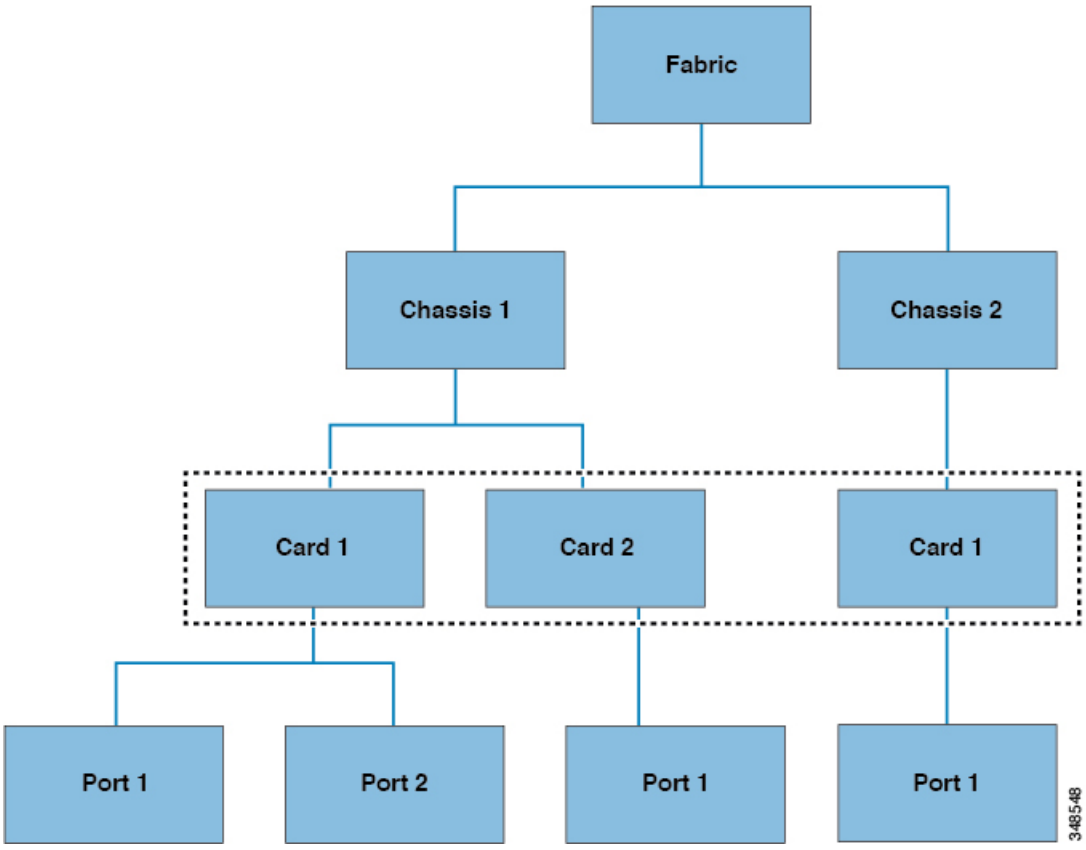


Both queries return the referenced object and its child objects. This approach is useful for discovering the components of a larger system. In this example, the query discovers the cards and ports of a given switch chassis.

Class-Level Queries

The following figure shows the second query type: the class-level query.

Figure 118: Class-Level Queries

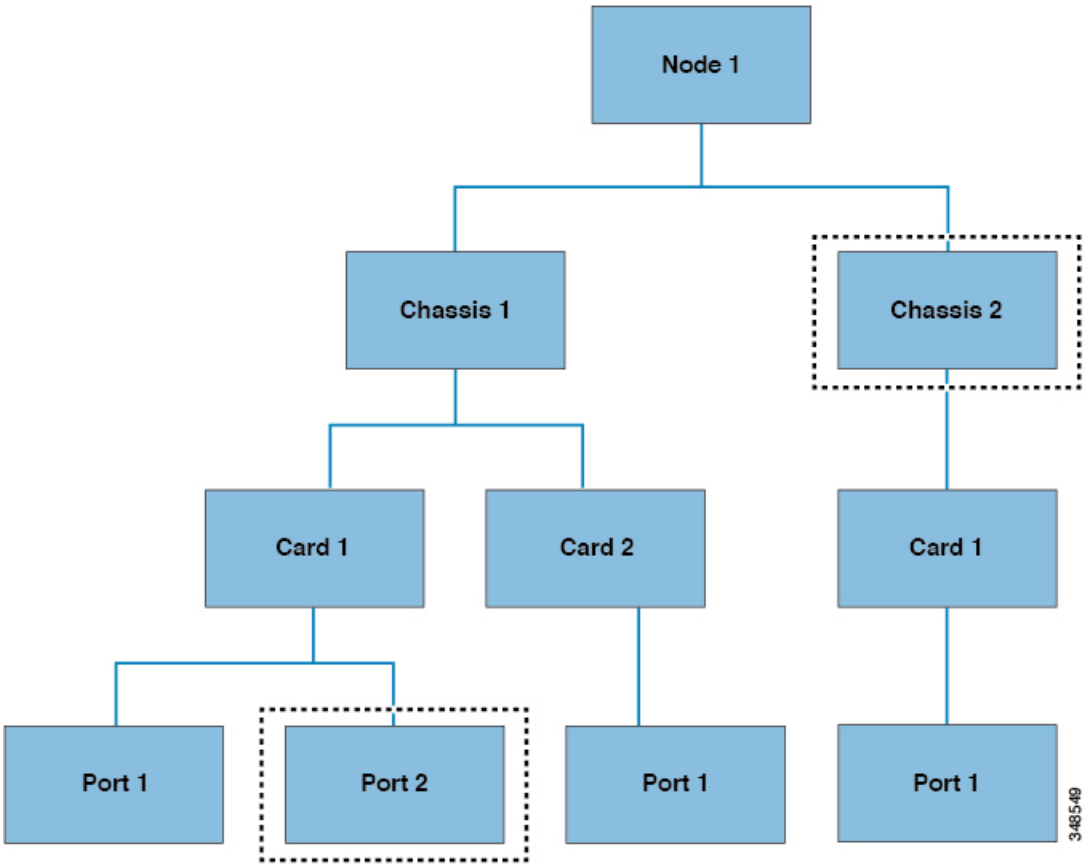


Class-level queries return all the objects of a given class. This approach is useful for discovering all the objects of a certain type that are available in the MIT. In this example, the class used is Cards, which returns all the objects of type Cards.

Object-Level Queries

The third query type is an object-level query. In an object-level query a distinguished name is used to return a specific object. The figure below shows two object-level queries: for Node 1 in Chassis 2, and one for Node 1 in Chassis 1 in Card 1 in Port 2.

Figure 119: Object-Level Queries

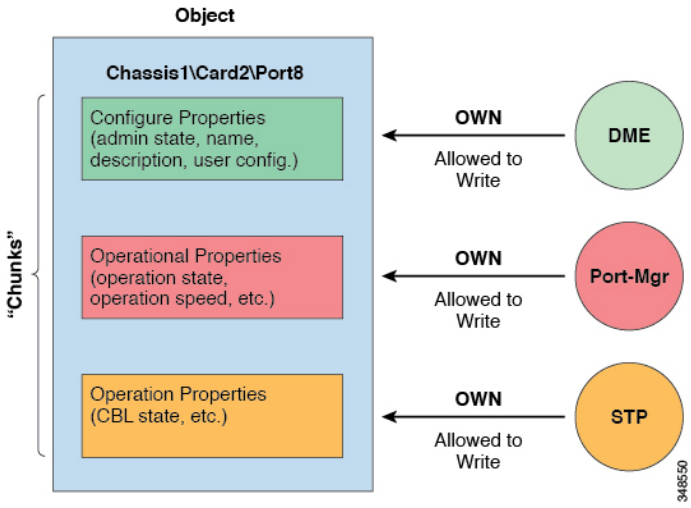


For all MIT queries, an administrator can optionally return the entire subtree or a partial subtree. Additionally, the role-based access control (RBAC) mechanism in the system dictates which objects are returned; only the objects that the user has rights to view will ever be returned.

Managed-Object Properties

Managed objects in the Cisco ACI contain properties that define the managed object. Properties in a managed object are divided into chunks that are managed by processes in the operating system. Any object can have several processes that access it. All these properties together are compiled at runtime and are presented to the user as a single object. The following figure shows an example of this relationship.

Figure 120: Managed Object Properties



The example object has three processes that write to property chunks that are in the object. The data management engine (DME), which is the interface between the Cisco APIC (the user) and the object, the port manager, which handles port configuration, and the spanning tree protocol (STP) all interact with chunks of this object. The APIC presents the object to the user as a single entity compiled at runtime.

Accessing the Object Data Through REST Interfaces

REST is a software architecture style for distributed systems such as the World Wide Web. REST has increasingly displaced other design models such as Simple Object Access Protocol (SOAP) and Web Services Description Language (WSDL) due to its simpler style. The Cisco APIC supports REST interfaces for programmatic access to the entire Cisco ACI solution.

The object-based information model of Cisco ACI makes it a very good fit for REST interfaces: URLs and URIs map directly to distinguished names that identify objects on the MIT, and any data on the MIT can be described as a self-contained structured text tree document that is encoded in XML or JSON. The objects have parent-child relationships that are identified using distinguished names and properties, which are read and modified by a set of create, read, update, and delete (CRUD) operations.

Objects can be accessed at their well-defined address, their REST URLs, using standard HTTP commands for retrieval and manipulation of Cisco APIC object data. The URL format used can be represented as follows:

```
<system>/api/[mo|class]/[dn|class][:method].[xml|json]?{options}
```

The various building blocks of the preceding URL are as follows:

- `system`: System identifier; an IP address or DNS-resolvable hostname
- `mo | class`: Indication of whether this is a MO in the MIT, or class-level query
- `class`: MO class (as specified in the information model) of the objects queried; the class name is represented as `<pkgName><ManagedObjectClassName>`
- `dn`: Distinguished name (unique hierarchical name of the object in the MIT) of the object queried
- `method`: Optional indication of the method being invoked on the object; applies only to HTTP POST requests
- `xml | json`: Encoding format

- `options`: Query options, filters, and arguments

With the capability to address and access an individual object or a class of objects with the REST URL, an administrator can achieve complete programmatic access to the entire object tree and to the entire system.

The following are REST query examples:

- Find all EPGs and their faults under tenant solar.

  ```
  http://192.168.10.1:7580/api/mo/uni/tn-solar.xml?query-target=subtree&target-subtree-class=fvAEPg&rsp-subtree-include=faults
  ```

- Filtered EPG query

  ```
  http://192.168.10.1:7580/api/class/fvAEPg.xml?query-target-filter=eq(fvAEPg.fabEncap,%20"vxlan-12780288")
  ```

Configuration Export/Import

All APIC policies and configuration data can be exported to create backups. This is configurable via an export policy that allows either scheduled or immediate backups to a remote server. Scheduled backups can be configured to execute periodic or recurring backup jobs. By default, all policies and tenants are backed up, but the administrator can optionally specify only a specific subtree of the management information tree. Backups can be imported into the APIC through an import policy, which allows the system to be restored to a previous configuration.

Configuration Database Sharding

The APIC cluster uses a large database technology called sharding. This technology provides scalability and reliability to the data sets generated and processed by the APIC. The data for APIC configurations is partitioned into logically bounded subsets called shards which are analogous to database shards. A shard is a unit of data management, and the APIC manages shards in the following ways:

- Each shard has three replicas.
- Shards are evenly distributed across the appliances that comprise the APIC cluster.

One or more shards are located on each APIC appliance. The shard data assignments are based on a predetermined hash function, and a static shard layout determines the assignment of shards to appliances.

Configuration File Encryption

As of release 1.1(2), the secure properties of APIC configuration files can be encrypted by enabling AES-256 encryption. AES encryption is a global configuration option; all secure properties conform to the AES configuration setting. It is not possible to export a subset of the ACI fabric configuration such as a tenant configuration with AES encryption while not encrypting the remainder of the fabric configuration. See the *Cisco Application Centric Infrastructure Fundamentals*, "Secure Properties" chapter for the list of secure properties.

The APIC uses a 16 to 32 character passphrase to generate the AES-256 keys. The APIC GUI displays a hash of the AES passphrase. This hash can be used to see if the same passphrases was used on two ACI fabrics. This hash can be copied to a client computer where it can be compared to the passphrase hash of another ACI fabric to see if they were generated with the same passphrase. The hash cannot be used to reconstruct the original passphrase or the AES-256 keys.

Observe the following guidelines when working with encrypted configuration files:

- Backward compatibility is supported for importing old ACI configurations into ACI fabrics that use the AES encryption configuration option.

Note Reverse compatibility is not supported; configurations exported from ACI fabrics that have enabled AES encryption cannot be imported into older versions of the APIC software.

- Always enable AES encryption when performing fabric backup configuration exports. Doing so will assure that all the secure properties of the configuration will be successfully imported when restoring the fabric.

Note If a fabric backup configuration is exported without AES encryption enabled, none of the secure properties will be included in the export. Since such an unencrypted backup would not include any of the secure properties, it is possible that importing such a file to restore a system could result in the administrator along with all users of the fabric being locked out of the system.

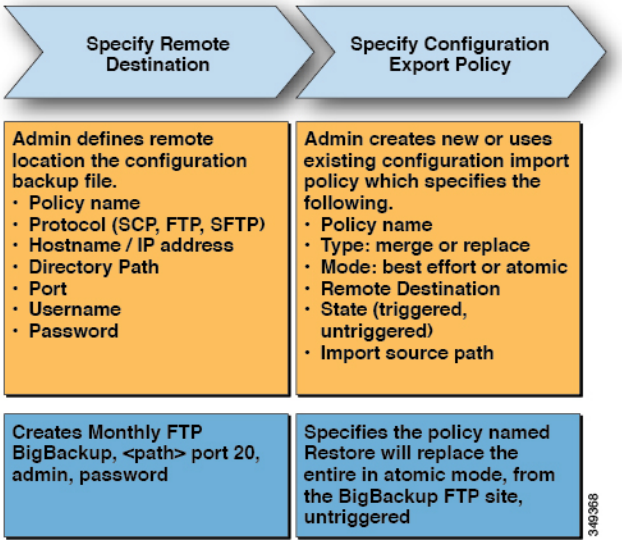
- The AES passphrase that generates the encryption keys cannot be recovered or read by an ACI administrator or any other user. The AES passphrase is not stored. The APIC uses the AES passphrase to generate the AES keys, then discards the passphrase. The AES keys are not exported. The AES keys cannot be recovered since they are not exported and cannot be retrieved via the REST API.
- The same AES-256 passphrase always generates the same AES-256 keys. Configuration export files can be imported into other ACI fabrics that use the same AES passphrase.
- For troubleshooting purposes, export a configuration file that does not contain the encrypted data of the secure properties. Temporarily turning off encryption before performing the configuration export removes the values of all secure properties from the exported configuration. To import such a configuration file that has all secure properties removed, use the import merge mode; do not use the import replace mode. Using the import merge mode will preserve the existing secure properties in the ACI fabric.
- By default, the APIC rejects configuration imports of files that contain fields that cannot be decrypted. Use caution when turning off this setting. Performing a configuration import inappropriately when this default setting is turned off could result in all the passwords of the ACI fabric to be removed upon the import of a configuration file that does not match the AES encryption settings of the fabric.

Note Failure to observe this guideline could result in all users, including fabric administrations, being locked out of the system.

Configuration Export

The following figure shows how the process works for configuring an export policy.

Figure 121: Workflow for Configuring an Export Policy



The APIC applies this policy in the following way:

- A complete system configuration backup is performed once a month.
- The backup is stored in XML format on the BigBackup FTP site.
- The policy is triggered (it is active).

Configuration Import

An administrator can create an import policy that performs the import in one of the following two modes:

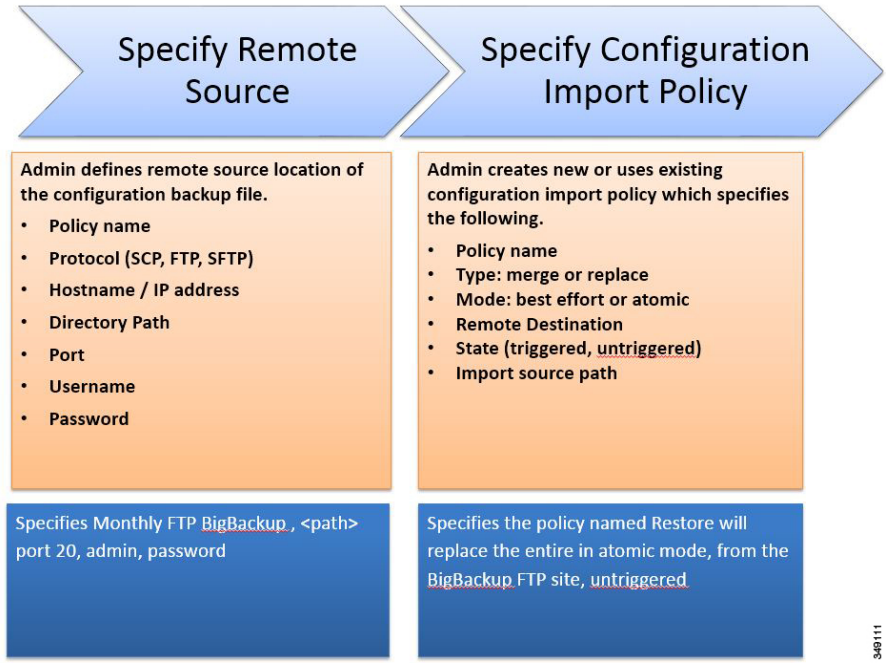
- Best-effort—ignores objects within a shard that cannot be imported. If the version of the incoming configuration is incompatible with the existing system, shards that are incompatible are not be imported while the import proceeds with those that can be imported.
- Atomic—ignores shards that contain objects that cannot be imported while proceeding with shards that can be imported. If the version of the incoming configuration is incompatible with the existing system, the import terminates.

An import policy supports the following combinations of mode and type:

- Best-effort Merge—imported configuration is merged with existing configuration but ignores objects that cannot be imported.
- Atomic Merge—imported configuration is merged with the existing configuration, but ignores shards that contain objects that cannot be imported.
- Atomic Replace—overwrites existing configuration with imported configuration data. Any objects in the existing configuration that do not exist in the imported configuration are deleted. Objects are deleted from the existing configuration that have children in the existing configuration but do not have children in the incoming imported configuration. For example, if an existing configuration has two tenants, solar and wind, but the imported backed up configuration was saved before the tenant wind was created, tenant soar is restored from the backup but tenant wind is deleted.

The following figure shows how the process works for configuring an import policy.

Figure 122: Workflow for Configuring an Import Policy



The APIC applies this policy in the following way:

- A policy is created to perform a complete system configuration restore from monthly backup.
- The atomic replace mode does the following:
  - Overwrites the existing configuration.
  - Deletes any existing configuration objects that are not present in the imported file.
  - Deletes non-present children objects.
- The policy is untriggered (it is available but has not been activated).

Tech Support, Statistics, Core

An administrator can configure export policies in the APIC to export statistics, technical support collections, faults and events, to process core files and debug data from the fabric (APIC as well as switch) to any external host. The exports can be in a variety of formats, including XML, JSON, web sockets, SCP, or HTTP. Exports are subscribable, and can be streaming, periodic, or on-demand.

Note The maximum number of statistics export policies is approximately equal to the number of tenants. Each tenant can have multiple statistics export policies and multiple tenants can share the same export policy, but the total number of policies is limited to approximately the number of tenants.

An administrator can configure policy details such as the transfer protocol, compression algorithm, and frequency of transfer. Policies can be configured by users who are authenticated using AAA. A security mechanism for the actual transfer is based on a username and password. Internally, a policy element handles the triggering of data.

Programmability Using Puppet

About Puppet

Puppet is a configuration management tool from Puppet Labs, Inc. Although Puppet was originally designed for large scale server management, many datacenter operators would like to consolidate server and network device provisioning using the same tool.

The following items are the primary components of a Puppet implementation:

- **Manifest** – A Puppet manifest is a collection of property definitions for setting the state of a managed device (node). The details for checking and setting these property states are abstracted so that a manifest can be used for more than one operating system or platform.
- **Master** – A Puppet master (server) typically runs on a separate dedicated server and serves multiple nodes. The Puppet master compiles configuration manifests and provides them to the nodes on request.
- **Agent or Device** – A Puppet agent runs on the node, where it periodically connects to the Puppet master to request a configuration manifest. The agent reconciles the received manifest with the current state of the node, updating the node state as necessary to resolve any differences. For nodes that cannot run an embedded Puppet agent or prefer not to, Puppet supports a construct called a Puppet device. A Puppet device is essentially a proxy mechanism, external to the node, that requests the manifest from the Puppet master on behalf of the node. The Puppet device then applies any updates required by the received manifest to the node. To leverage this capability, a vendor must provide a vendor-specific implementation of the device class along with a Puppet module that makes use of the device. The vendor-specific device class uses a proprietary protocol or API to configure the remote node.

For further information and documentation about Puppet, see the Puppet website at the following URL: https://puppet.com/.

Cisco ciscoacipuppet Puppet Module

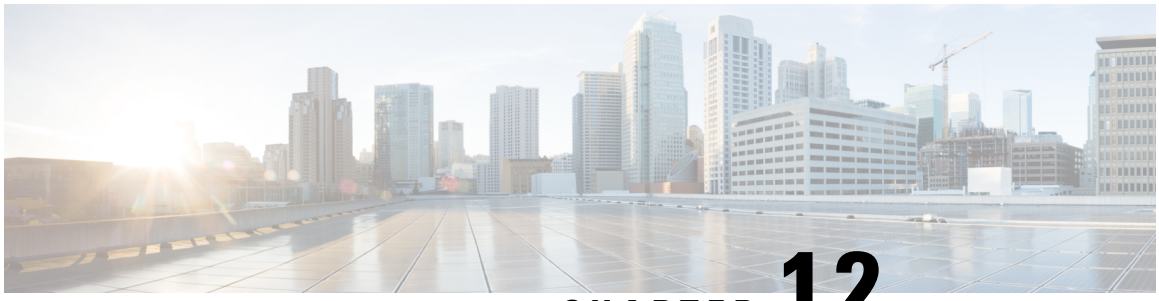
An APIC controller does not run an embedded Puppet agent. Instead, Cisco provides a Puppet module ("ciscoacipuppet"), which uses a Cisco ACI-specific Puppet device to relay configuration management requests to the APIC controller. The ciscoacipuppet module interprets change information in the received Puppet manifest and translates the change requests into APIC REST API messages to implement configuration changes in the ACI fabric.

For details on the installation, setup, and usage of the ciscoacipuppet module, refer to the documentation on GitHub and Puppet Forge at the following URLs:

- **GitHub** – https://github.com/cisco/cisco-network-puppet-module
- **Puppet Forge** – https://forge.puppet.com/puppetlabs/ciscoacipuppet

Puppet Guidelines and Limitations for ACI

- Only a subset of APIC managed objects can be provisioned using the ciscoacipuppet Puppet module. To understand the level of support and the limitations, refer to the ciscoacipuppet module documentation on GitHub and Puppet Forge.



CHAPTER 12

Monitoring

This chapter contains the following sections:

Faults, Errors, Events, Audit Logs

Note For information about faults, events, errors, and system messages, see the *Cisco APIC Faults, Events, and System Messages Management Guide* and the *Cisco APIC Management Information Model Reference*, a Web-based application.

The APIC maintains a comprehensive, current run-time representation of the administrative and operational state of the ACI Fabric system in the form of a collection of MOs. The system generates faults, errors, events, and audit log data according to the run-time state of the system and the policies that the system and user create to manage these processes.

The APIC GUI enables you to create customized "historical record groups" of fabric switches, to which you can then assign customized switch policies that specify customized size and retention periods for the audit logs, event logs, health logs, and fault logs maintained for the switches in those groups.

The APIC GUI also enables you to customize a global controller policy that specifies size and retention periods for the audit logs, event logs, health logs, and fault logs maintained for the controllers on this fabric.

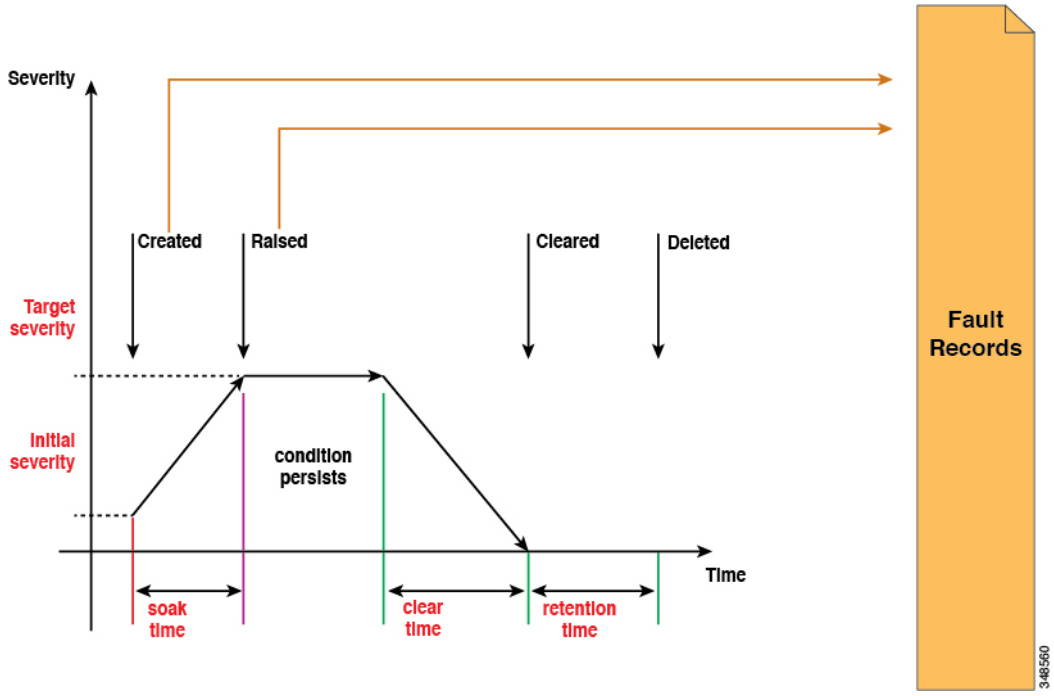
Faults

Based on the run-time state of the system, the APIC automatically detects anomalies and creates fault objects to represent them. Fault objects contain various properties that are meant to help users diagnose the issue, assess its impact and provide a remedy.

For example, if the system detects a problem associated with a port, such as a high parity-error rate, a fault object is automatically created and placed in the management information tree (MIT) as a child of the port

object. If the same condition is detected multiple times, no additional instances of the fault object are created. After the condition that triggered the fault is remedied, the fault object is preserved for a period of time specified in a fault life-cycle policy and is finally deleted. See the following figure.

Figure 123: Fault Life Cycle



A life cycle represents the current state of the issue. It starts in the soak time when the issue is first detected, and it changes to raised and remains in that state if the issue is still present. When the condition is cleared, it moves to a state called "raised-clearing" in which the condition is still considered as potentially present. Then it moves to a "clearing time" and finally to "retaining". At this point, the issue is considered to be resolved and the fault object is retained only to provide the user visibility into recently resolved issues.

Each time that a life-cycle transition occurs, the system automatically creates a fault record object to log it. Fault records are never modified after they are created and they are deleted only when their number exceeds the maximum value specified in the fault retention policy.

The severity is an estimate of the impact of the condition on the capability of the system to provide service. Possible values are warning, minor, major and critical. A fault with a severity equal to warning indicates a potential issue (including, for example, an incomplete or inconsistent configuration) that is not currently affecting any deployed service. Minor and major faults indicate that there is potential degradation in the service being provided. Critical means that a major outage is severely degrading a service or impairing it altogether. Description contains a human-readable description of the issue that is meant to provide additional information and help in troubleshooting.

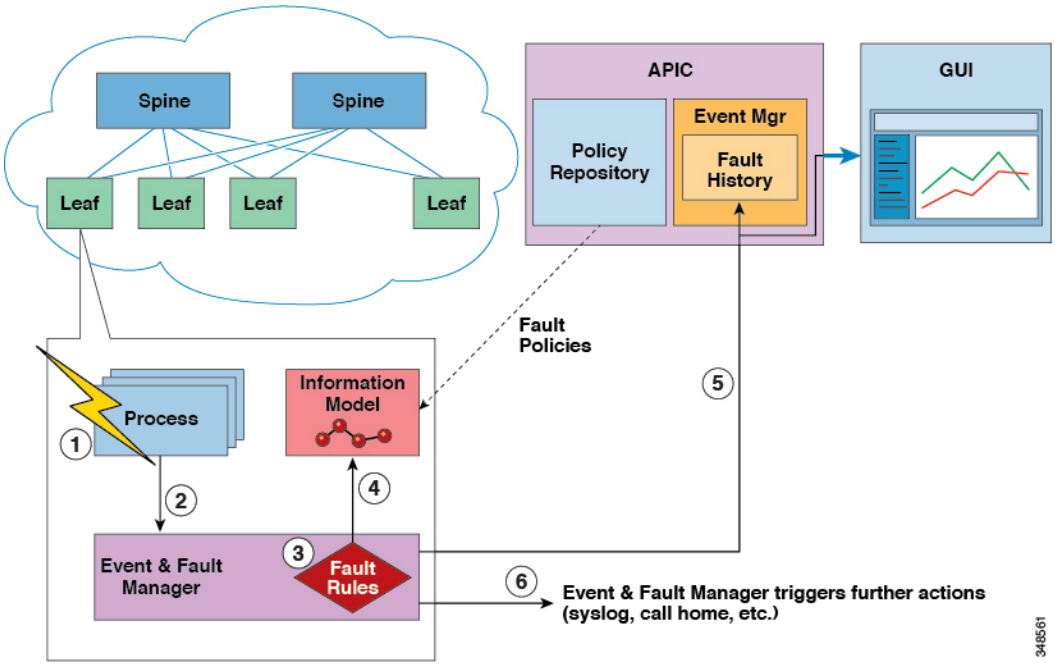
Events

Event records are objects that are created by the system to log the occurrence of a specific condition that might be of interest to the user. They contain the fully qualified domain name (FQDN) of the affected object, a timestamp and a description of the condition. Examples include link-state transitions, starting and stopping

of protocols, and detection of new hardware components. Event records are never modified after creation and are deleted only when their number exceeds the maximum value specified in the event retention policy.

The following figure shows the process for fault and events reporting.

Figure 124: Faults and Events Reporting/Export



1. Process detects a faulty condition.
2. Process notifies Event and Fault Manager.
3. Event and Fault Manager processes the notification according to the fault rules.
4. Event and Fault Manager creates a fault Instance in the MIM and manages its life cycle according to the fault policy.
5. Event and Fault Manager notifies the APIC and connected clients of the state transitions.
6. Event and Fault Manager triggers further actions (such as syslog or call home).

Errors

APIC error messages typically display in the APIC GUI and the APIC CLI. These error messages are specific to the action that a user is performing or the object that a user is configuring or administering. These messages can be the following:

- Informational messages that provide assistance and tips about the action being performed
- Warning messages that provide information about system errors related to an object, such as a user account or service profile, that the user is configuring or administering
- Finite state machine (FSM) status messages that provide information about the status of an FSM stage

Many error messages contain one or more variables. The information that the APIC uses to replace these variables depends upon the context of the message. Some messages can be generated by more than one type of error.

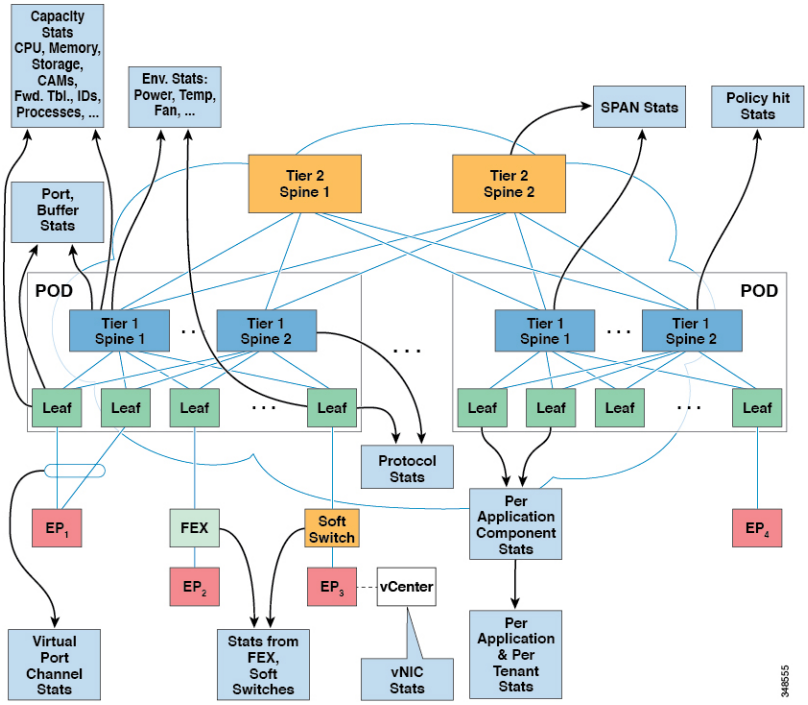
Audit Logs

Audit records are objects that are created by the system to log user-initiated actions, such as login/logout and configuration changes. They contain the name of the user who is performing the action, a timestamp, a description of the action and, if applicable, the FQDN of the affected object. Audit records are never modified after creation and are deleted only when their number exceeds the maximum value specified in the audit retention policy.

Statistics Properties, Tiers, Thresholds, and Monitoring

Statistics enable trend analysis and troubleshooting. Statistics gathering can be configured for ongoing or on-demand collection. Statistics provide real-time measures of observed objects. Statistics can be collected in cumulative counters and gauges.

Figure 125: Various Sources of Statistics



Policies define what statistics are gathered, at what intervals, and what actions to take. For example, a policy could raise a fault on an EPG if a threshold of dropped packets on an ingress VLAN is greater than 1000 per second.

Statistics data are gathered from a variety of sources, including interfaces, VLANs, EPGs, application profiles, ACL rules, tenants, or internal Cisco Application Policy Infrastructure Controller (APIC) processes. Statistics accumulate data in 5-minute, 15-minute, 1-hour, 1-day, 1-week, 1-month, 1-quarter, or 1-year sampling intervals. Shorter duration intervals feed longer intervals.

A variety of statistics properties are available, including average, minimum, maximum, trend, and rate of change. Collection and retention times are configurable. Policies can specify if the statistics are to be gathered from the current state of the system or to be accumulated historically or both. For example, a policy could specify that historical statistics be gathered for 5-minute intervals over a period of 1 hour. The 1 hour is a moving window. Once an hour has elapsed, the incoming 5 minutes of statistics are added, and the earliest 5 minutes of data are abandoned.

Note The maximum number of 5-minute granularity sample records is limited to 3 samples (15 minutes of statistics). All other sample intervals are limited to 1,000 sample records. For example, hourly granularity statistics can be maintained for up to 41 days. Statistics will not be maintained for longer than these limits. To gather statistics for longer durations, create an export policy.

About Statistics Data

The following types of managed objects (MOs) are associated with statistics data that is collected by the observer module:

- History data
- Current data

The MO names corresponding to these objects start with a two-letter prefix: HD or CD. HD indicates history data while CD indicates current data. For example, "CDl2IngrBytesAg15min." The MO name is also an indicator of the time interval for which the data is collected. For example, "CDl2IngrBytesAg15min" indicates that the MO corresponds to 15-minute intervals.

A CD object holds currently running data, and the values that the object holds change as time passes. However, at the end of a given time interval, the data collected in a CD object is copied to an HD object and the CD object attributes are reset to 0. For example, at the end of a given 15-minute interval, the data in the CDl2IngrBytesAg15min object is moved to the HDl2IngrBytesAg15min object and the CDl2IngrBytesAg15min object is reset.

If a CD...15min object data is closely observed for more than 15 minutes, you can notice that the value goes to 0, then gets incremented twice and goes to 0 again. This is because the values are getting updated every 5 minutes. The third update (at the end of 15 minutes) goes unnoticed, as the data was rolled up to the HD object and the CD object was reset as soon as that update occurred.

CD...15min objects are updated every 5 minutes and CD...5min objects are updated every 10 seconds. CD...15min objects are rolled up as HD...15min objects and CD...5min are rolled up as HD...5min objects.

The data that any CD object holds is dynamic and for all practical purposes it must be considered to be internal data. HD data objects can be used for any further analytical purposes and can be considered to be published or static data.

The HD objects are also rolled up as time passes. For example, three consecutive HD...5min data objects contribute to one HD...15min object. The length of time that one HD...5min object resides in the system is decided by the statistic collection policies.

Configuring Monitoring Policies

Administrators can create monitoring policies with the following four broad scopes:

- Fabric Wide: includes both fabric and access objects
- Access (also known as infrastructure): access ports, FEX, VM controllers, and so on
- Fabric: fabric ports, cards, chassis, fans, and so on
- Tenant: EPGs, application profiles, services, and so on

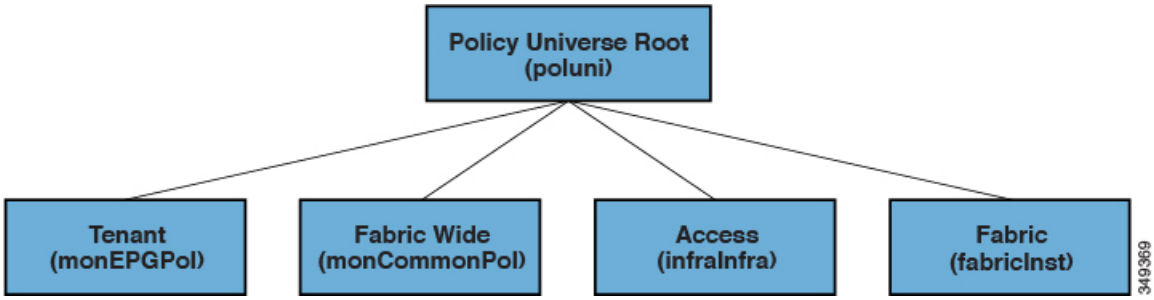
The Cisco Application Policy Infrastructure Controller (APIC) includes the following four classes of default monitoring policies:

- `monCommonPol` (uni/fabric/moncommon): applies to all fabric, access, and tenant hierarchies
- `monFabricPol` (uni/fabric/monfab-default): applies to fabric hierarchies
- `monInfraPol` (uni/infra/monifra-default): applies to the access infrastructure hierarchy
- `monEPGPol` (uni/tn-common/monepg-default): applies to tenant hierarchies

In each of the four classes of monitoring policies, the default policy can be overridden by a specific policy. For example, a monitoring policy applied to the Solar tenant (*tn-solar*) would override the default one for the Solar tenant while other tenants would still be monitored by the default policy.

Each of the four objects in the figure below contains monitoring targets.

Figure 126: Four Classes of Default Monitoring Policies



The Infra monitoring policy contains `monInfra` targets, the fabric monitoring policy contains `monFab` targets, and the tenant monitoring policy contains `monEPG` targets. Each of the targets represent the corresponding class of objects in this hierarchy. For example, under the `monInfra-default` monitoring policy, there is a target representing FEX fabric-facing ports. The policy details regarding how to monitor these FEX fabric-facing ports are contained in this target. Only policies applicable to a target are allowed under that target. Note that not all possible targets are auto-created by default. The administrator can add more targets under a policy if the target is not there.

The common monitoring policy (`monCommonPol`) has global fabric-wide scope and is automatically deployed on all nodes in the fabric, including the Cisco APICs. Any source (such as syslog, callhome, or SNMP) located under the common monitoring policy captures all faults, events, audits and health occurrences. The single common monitoring policy monitors the whole fabric. The threshold of the severity for syslog and snmp or urgency for callhome can be configured according to the level of detail that a fabric administrator determines is appropriate.

Multiple monitoring policies can be used to monitor individual parts of the fabric independently. For example, a source under the global monitoring policy reflects a global view. Another source under a custom monitoring policy deployed only to some nodes could closely monitor their power supplies. Or, specific fault or event occurrences for different tenants could be redirected to n.jpgy specific operators.

Sources located under other monitoring policies capture faults, events and audits within a smaller scope. A source located directly under a monitoring policy, captures all occurrences within the scope (for example fabric or infra). A source located under a target, captures all occurrences related to that target (for example, `eqpt:Psu` for power supply). A source located under a fault/event severity assignment policy captures only the occurrences that match that particular fault or event as ide.jpgied by the fault/event code.

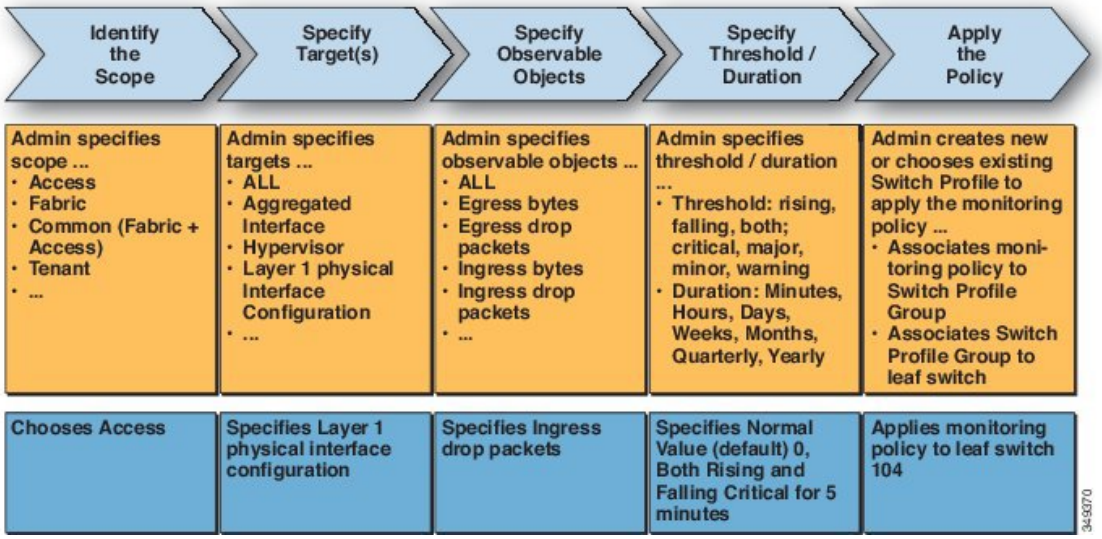
When a fault/event/audit is generated, all applicable sources are used. For example, consider the following configuration:

- Syslog source 4, pointing to syslog group 4 is defined for fault F0123.
- Syslog source 3, pointing to syslog group 3 is defined for target power supply (`eqpt:Psu`).
- Syslog source 2, pointing to syslog group 2 is defined for scope infra.
- Syslog source 1, pointing to syslog group 1 is defined for the common monitoring policy.

If fault F0123 occurs on an MO of class `eqpt:Psu` in scope infra, a syslog message is sent to all the destinations in syslog groups 1-4, assuming the severity of the message is at or above the minimum defined for each source and destination. While this example illustrates a syslog configuration, callhome and SNMP configurations would operate in the same way.

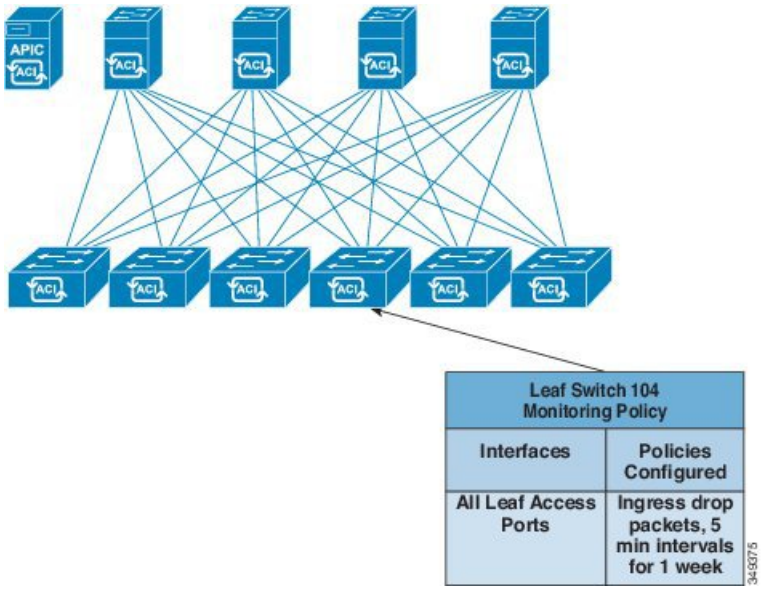
The following figure shows how the process works for configuring a fabric monitoring policy for statistics.

Figure 127: Workflow for Configuring an Access Monitoring Policy



The Cisco APIC applies this monitoring policy as shown in the following figure:

Figure 128: Result of Sample Access Monitoring Policy



Monitoring policies can also be configured for other system operations, such as faults or health scores. The structure of monitoring policies map to this hierarchy:

Monitoring Policy

- **Statistics Export**
- **Collection Rules**
- **Monitoring Targets**
  - **Statistics Export**
  - **Collection Rules**
  - **Statistics**
    - **Collection Rules**
    - **Thresholds Rules**
    - **Statistics Export**

Statistics Export policies option in the following figure define the format and destination for statistics to be exported. The output can be exported using the FTP, HTTP, or SCP protocols. The format can be JSON or XML. The user or administrator can also choose to compress the output. Export can be defined under Statistics, Monitoring Targets, or under the top-level monitoring policy. The higher-level definition of Statistics Export takes precedence unless there is a defined lower-level policy.

Monitoring policies are applied to specific observable objects (such as ports, cards, EPGs, and tenants) or groups of observable objects by using selectors or relations. Monitoring policies define the following things:

- Statistics are collected and retained in the history.
- Threshold crossing faults are triggered.

- Statistics are exported.

Collection rules are defined per sampling interval, as specified by the granularity. The rules configure whether the collection of statistics should be turned on or off, and when turned on, what the history retention period should be. Monitoring Targets correspond to observable objects (such as ports and EPGs). Collection Rules can be defined under Statistics, Monitoring Targets, or under the top-level Monitoring Policy. The higher-level definition of Collection Rules takes precedence unless there is a defined lower-level policy.

Statistics correspond to groups of statistical counters (such as ingress-counters, egress-counters, or drop-counters).

Threshold rules are defined under collection rules and are applied to the corresponding sampling-interval that is defined in the parent collection rule.

Tetration Analytics

About Cisco Tetration Analytics Agent Installation

The Cisco Tetration agent installation is accomplished by downloading the RPM Package Manager (RPM) file from the Cisco Tetration cluster and upload it to APIC. The Cisco Tetration cluster send a notification to the switch whenever a later version of the Cisco Tetration agent is uploaded.

There are two possible scenarios regarding the installation of the image on the switch:

- The Cisco Tetration image is not installed on the switch: the switch receives a notification from APIC, downloads and installs the Cisco Tetration agent image on the container on the switch.
- The Cisco Tetration image is installed on the switch and the switch receives a notification from the APIC. The switch checks if the APIC version is higher than that of the agent image already installed. If the version is higher, the switch downloads and installs the latest Cisco Tetration image on the container on the switch.

The image is installed in persistent memory. On reboot, after receiving controller notification from APIC, the switch starts the Cisco Tetration agent irrespective of the image that is available on APIC.

NetFlow

About NetFlow

The NetFlow technology provides the metering base for a key set of applications, including network traffic accounting, usage-based network billing, network planning, as well as denial of services monitoring, network monitoring, outbound marketing, and data mining for both service providers and enterprise customers. Cisco provides a set of NetFlow applications to collect NetFlow export data, perform data volume reduction, perform post-processing, and provide end-user applications with easy access to NetFlow data. If you have enabled NetFlow monitoring of the traffic flowing through your datacenters, this feature enables you to perform the same level of monitoring of the traffic flowing through the Cisco Application Centric Infrastructure (Cisco ACI) fabric.

Instead of hardware directly exporting the records to a collector, the records are processed in the supervisor engine and are exported to standard NetFlow collectors in the required format.

For detailed information about configuring and using NetFlow, see *Cisco APIC and NetFlow*.

For information about configuring NetFlow with virtual machine networking, see the *Cisco ACI Virtualization Guide*.

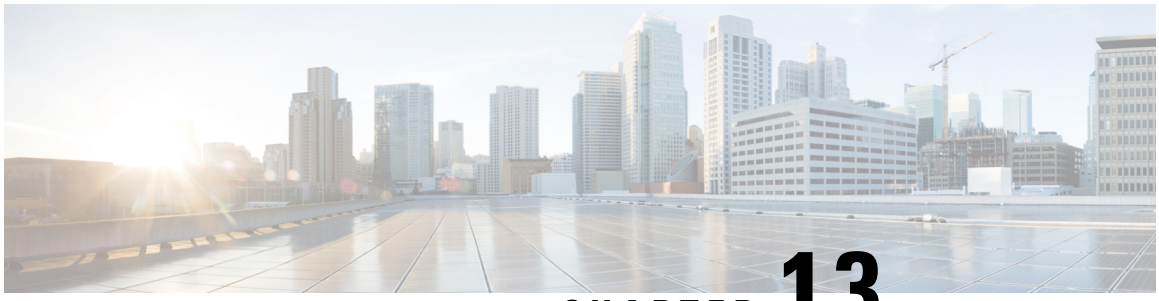
NetFlow Support and Limitations

NetFlow is supported on EX, FX, and FX2 and newer switches. For a full list of switch models supported on a specific release, see the *Cisco Nexus 9000 ACI-Mode Switches Release Notes* for that release.

NetFlow on remote leaf switches is supported starting with Cisco Application Policy Infrastructure Controller (APIC) release 4.0(1).

The following list provides information about the available support for NetFlow and the limitations of that support:

- Cisco Application Centric Infrastructure (ACI) supports only ingress and not egress NetFlow. Packets entering from a spine switch cannot be captured reliably with NetFlow on a bridge domain.
- NetFlow on spine switches is not supported, and tenant level information cannot be derived locally from the packet on the spine switch.
- The hardware does not support any active/inactive timers. The flow table records get aggregated as the table gets flushed, and the records get exported every minute.
- At every export interval, software cache gets flushed and the records that are exported in the next interval will have a reset packet/byte count and other statistics, even if the flow was long-lived.
- The filter TCAM has no labels for bridge domain or interfaces. If a NetFlow monitor is added to 2 bridge domains, the NetFlow monitor uses 2 rules for IPv4, or 8 rules for IPv6. As such, the scale is very limited with the 1K filter TCAM.
- ARP/ND are handled as IP packets and their target protocol addresses are put in the IP fields with some special protocol numbers from 249 to 255 as protocol ranges. NetFlow collectors might not understand this handling.
- The ICMP checksum is part of the Layer 4 src port in the flow record, so for ICMP records, many flow entries will be created if this is not masked, as is similar for other non-TCP/UDP packets.
- Cisco ACI-mode switches support only two active exporters.



CHAPTER 13

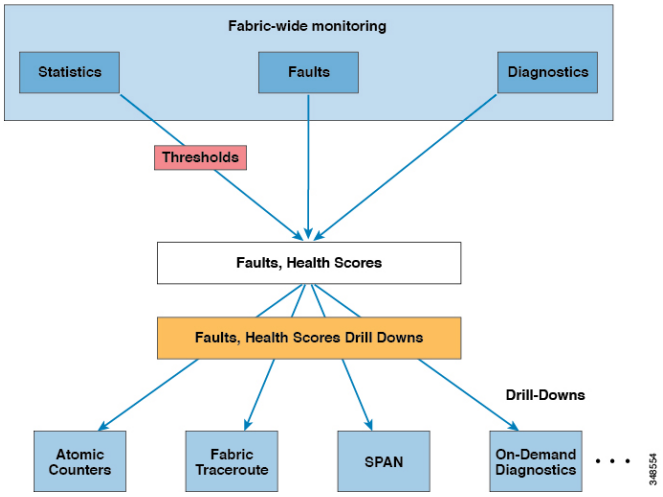
Troubleshooting

This chapter contains the following sections:

Troubleshooting

The ACI fabric provides extensive troubleshooting and monitoring tools as shown in the following figure.

Figure 129: Troubleshooting



About ACL Contract Permit and Deny Logs

To log and/or monitor the traffic flow for a contract rule, you can enable and view the logging of packets or flows that were allowed to be sent because of contract permit rules or the logging of packets or flows that were dropped because of:

- Taboo contract deny rules
- Deny actions in contract subjects
- Contract or subject exceptions
- ACL contract permit in the ACI fabric is only supported on Nexus 9000 Series switches with names that end in EX or FX, and all later models. For example, N9K-C93180LC-EX or N9K-C9336C-FX.
- Deny logging in the ACI fabric is supported on all platforms.
- Using log directive on filters in management contracts is not supported. Setting the log directive will cause zoning-rule deployment failure.

For information on standard and taboo contracts and subjects, see *Cisco Application Centric Infrastructure Fundamentals* and *Cisco APIC Basic Configuration Guide*.

EPG Data Included in ACL Permit and Deny Log Output

Up to Cisco APIC, Release 3.2(1), the ACL permit and deny logs did not identify the EPGs associated with the contracts being logged. In release 3.2(1) the source EPG and destination EPG are added to the output of ACI permit and deny logs. ACL permit and deny logs include the relevant EPGs with the following limitations:

- Depending on the position of the EPG in the network, EPG data may not be available for the logs.
- When configuration changes occur, log data may be out of date. In steady state, log data is accurate.

The most accurate EPG data in the permit and deny logs results when the logs are focussed on:

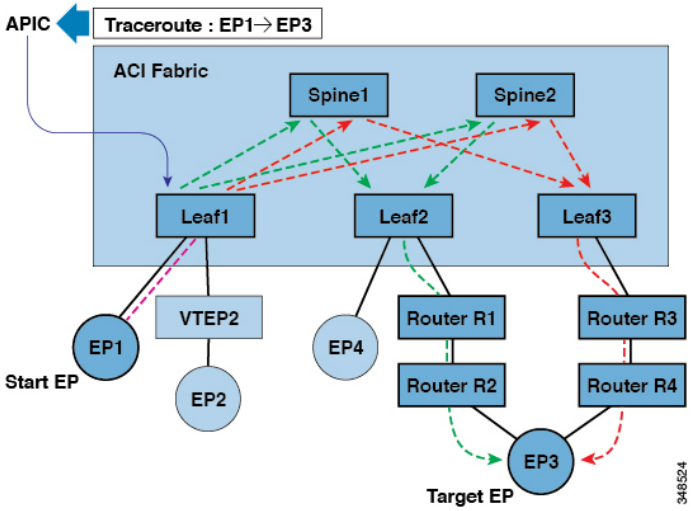
- Flows from EPG to EPG, where the ingress policy is installed at the ingress TOR and the egress policy is installed at the egress TOR.
- Flows from EPG to L3Out, where one policy is applied on the border leaf TOR and the other policy is applied on a non-BL TOR.

EPGs in the log output are not supported for uSeg EPGs or for EPGs used in shared services (including shared L3Outs).

ARPs, ICMP Pings, and Traceroute

ARPs for the default gateway IP address are trapped at the ingress leaf switch. The ingress leaf switch unicasts the ARP request to the destination and the destination sends the ARP response.

Figure 130: APIC Endpoint to Endpoint Traceroute



A traceroute that is initiated from the tenant endpoints shows the default gateway as an intermediate hop appears at the ingress leaf switch.

Traceroute modes include from endpoint to endpoint, and from leaf to leaf (TEP to TEP). Traceroute discovers all paths across the fabric, points of exit for external endpoints, and helps to detect if any path is blocked.

Traceroute works with IPv6 source and destination addresses but configuring source and destination addresses across IPv4 and IPv6 addresses is not allowed. Source (`RsTrEpIpSrc`) and destination (`RsTrEpIpDst`) relations support source and destination of type `fvIp`. At times, multiple IP addresses are learned from the same endpoint. The administrator chooses the desired source and destination addresses.

Atomic Counters

Atomic counters detect drops and misrouting in the fabric. The resulting statistics enable quick debugging and isolation of application connectivity issues. Atomic counters require an active fabric Network Time Protocol (NTP) policy. Atomic counters work for either IPv6 or IPv4 source and destination addresses but not across different address families.

For example, an administrator can enable atomic counters on all leaf switches to trace packets from endpoint 1 to endpoint 2. If any leaf switches have nonzero counters, other than the source and destination leaf switches, an administrator can drill down to those leaf switches.

In conventional settings, it is nearly impossible to monitor the amount of traffic from a baremetal NIC to a specific IP address (an endpoint) or to any IP address. Atomic counters allow an administrator to count the number of packets that are received from a baremetal endpoint without any interference to its data path. In addition, atomic counters can monitor per-protocol traffic that is sent to and from an endpoint or an application group.

Leaf-to-leaf (TEP to TEP) atomic counters can provide the following:

- Counts of drops, admits, and excess packets.
- Short-term data collection such as the last 30 seconds, and long-term data collection such as 5 minutes, 15 minutes, or more.

- A breakdown of per-spine traffic is available only when the number of TEPs, leaf or VPC, is less than 64.
- Ongoing monitoring.

Note Leaf-to-leaf (TEP to TEP) atomic counters are cumulative and cannot be cleared. However, because 30-second atomic counters reset at 30-second intervals, they can be used to isolate intermittent or recurring problems.

Tenant atomic counters can provide the following:

- Application-specific counters for traffic across the fabric, including drops, admits, and excess packets
- Modes include the following:
  - Endpoint-to-endpoint MAC address, or endpoint-to-endpoint IP address. Note that a single target endpoint could have multiple IP addresses associated with it.
  - EPG to EPG
  - EPG to endpoint
  - EPG to * (any)
  - Endpoint to external IP address

Note Use of atomic counters is not supported when the endpoints are in different tenants or in different Virtual Routing and Forwarding (VRF) instances (also known as contexts or private networks) within the same tenant. Atomic counters work for IPv6 source and destinations but configuring source and destination IP addresses across IPv4 and IPv6 addresses is not allowed.

Endpoint-to-endpoint atomic counter statistics are not reported for Layer 2 bridged traffic with IPv6 headers when the endpoints belong to the same EPG.

For atomic counters to work for traffic flowing from an EPG or ESG to an L3Out EPG, configure the L3Out EPG with 0/1 and 128/1 to match all prefixes instead of 0/0.

About Digital Optical Monitoring

Real-time digital optical monitoring (DOM) data is collected from SFPs, SFP+, and XFPs periodically and compared with warning and alarm threshold table values. The DOM data collected are transceiver transmit bias current, transceiver transmit power, transceiver receive power, and transceiver power supply voltage.

Health Scores

The ACI fabric uses a policy model to combine data into a health score. Health scores can be aggregated for a variety of areas such as for the system, infrastructure, tenants, applications, or services.

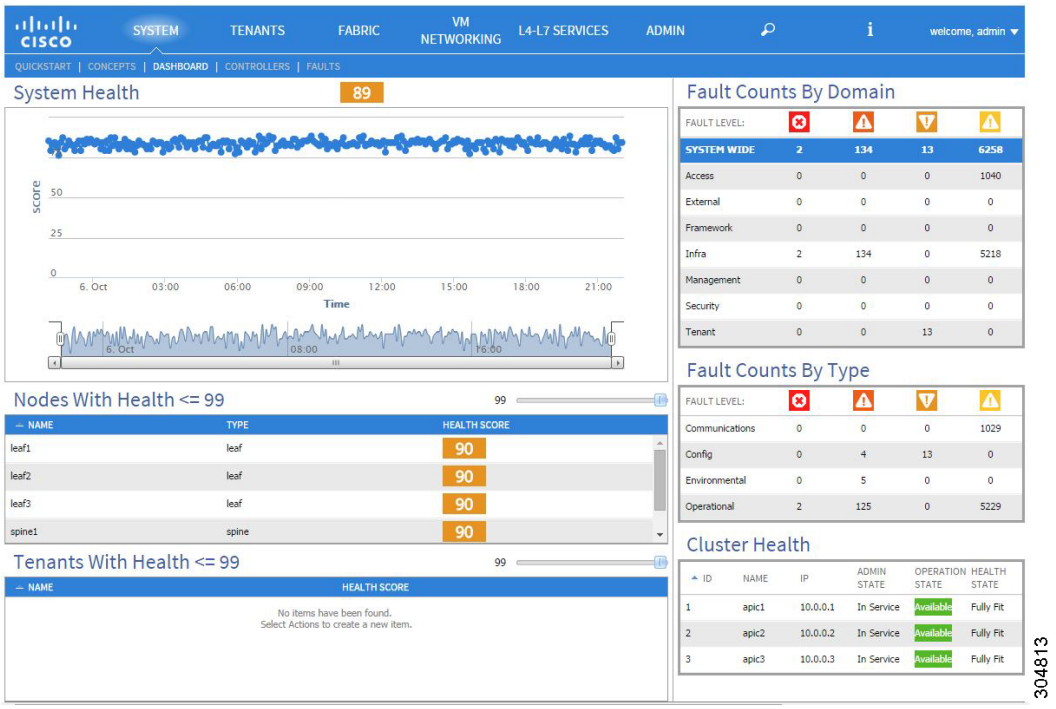
ACI fabric health information is available for the following views of the system:

- System — aggregation of system-wide health, including pod health scores, tenant health scores, system fault counts by domain and type, and the APIC cluster health state.
- Pod — aggregation of health scores for a pod (a group of spine and leaf switches), and pod-wide fault counts by domain and type.
- Tenant — aggregation of health scores for a tenant, including performance data for objects such as applications and EPGs that are specific to a tenant, and tenant-wide fault counts by domain and type.
- Managed Object — health score policies for managed objects (MOs), which includes their dependent and related MOs. These policies can be customized by an administrator.

System and Pod Health Scores

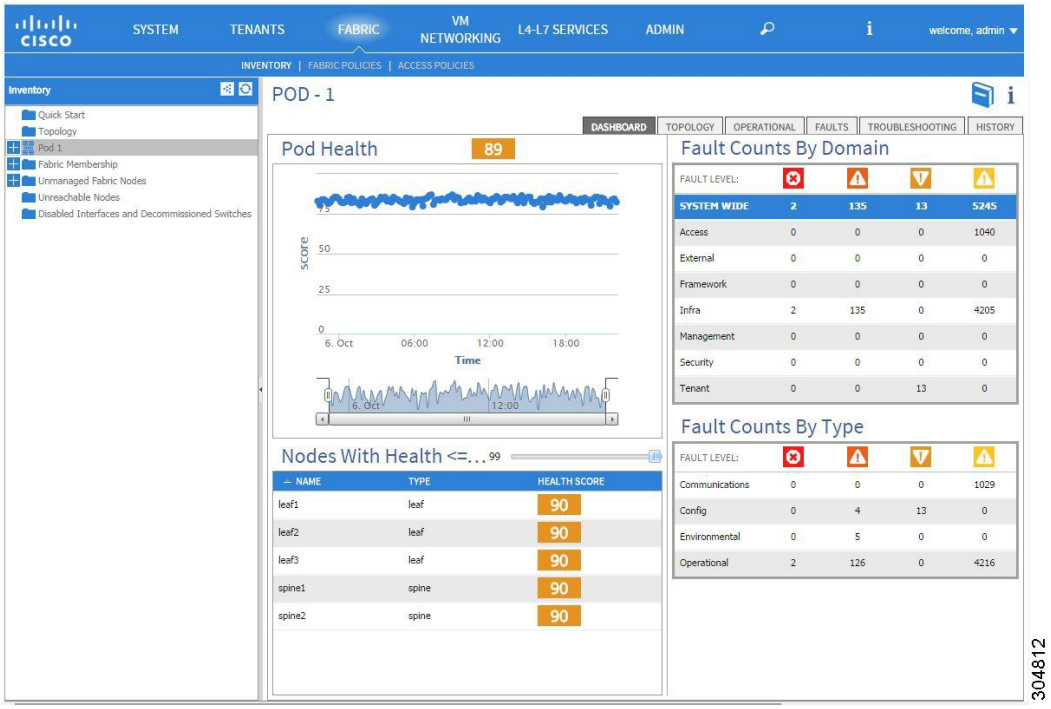
The system and pod health scores are based on the leaf and spine switches health scores as well as the number of end-points learned on the leaf switches. The GUI System Dashboard also displays system-wide fault counts by domain type, along with the APIC cluster per-node admin state, operational state, and health state.

Figure 131: System Health Scores



The pod health scores are based on the leaf and spine switches health scores as well as the number of end-points learnt on the leaf switches. The GUI fabric pod dashboard screen also displays pod-wide fault counts by domain and type.

Figure 132: Pod Health Scores



The system and pod health scores are calculated the same way. The calculation is based on the weighted average of the leaf health scores divided by the total number of learned end points of the leaf switches times the spine coefficient which is derived from the number of spines and their health scores.

The following equation shows how this calculation is done.

Figure 133: System and Pod Health Score Calculation

$$Health_{Fabric} = \frac{\sum_{i=1}^{N_{Leaf}} Health_{Leaf_i} \times Weight_{Leaf_i}}{\sum_{i=1}^{N_{Leaf}} Weight_{Leaf_i}} \times \left(1 - \left(1 - \frac{\sum_{i=1}^{N_{Spine}} Health_{Spine_i}}{N_{Spine} \times 100}\right)^{N_{Spine}}\right)$$

304814

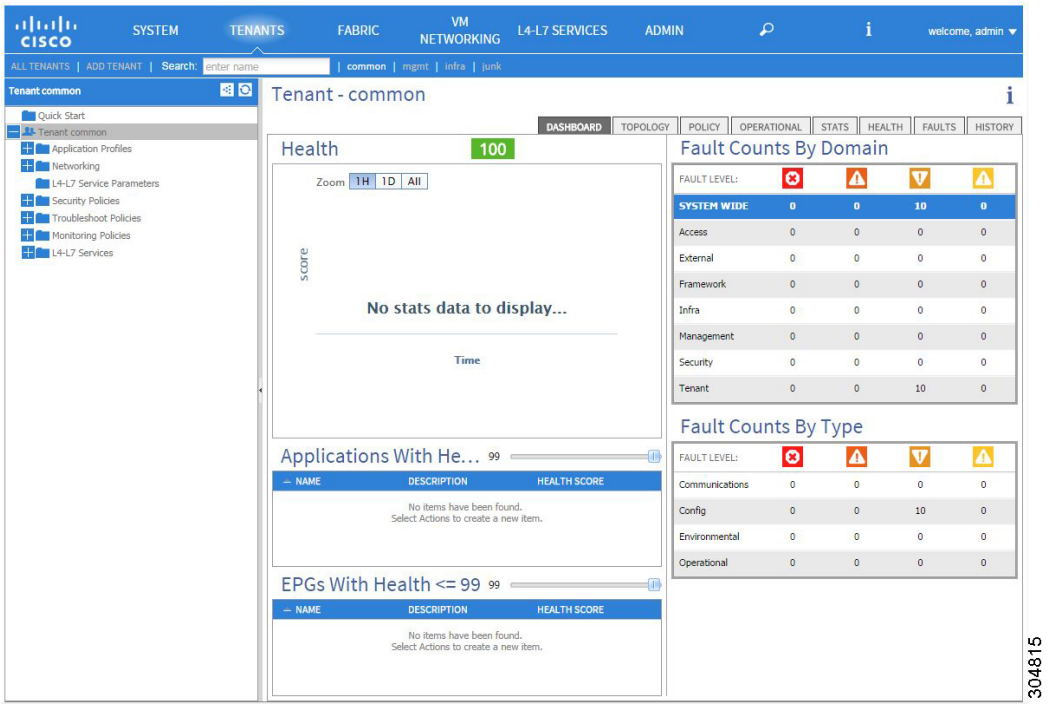
The following legend defines the equation components.

- $Health_{Leafi}$ is the health score of the leaf switch.
- $Weight_{Leafi}$ is the number of end-points on the leaf switch.
- N_{Leaf} is the number of leaf switches in the fabric.
- $Health_{Spinei}$ is the health score of the spine switch.
- N_{Spine} is the number of spine switches in the fabric.

Tenant Health Scores

Tenant health scores aggregate the tenant-wide logical objects health scores across the infrastructure they happen to use. The GUI tenant dashboard screen also displays tenant-wide-fault counts by domain and type.

Figure 134: Tenant Health Scores

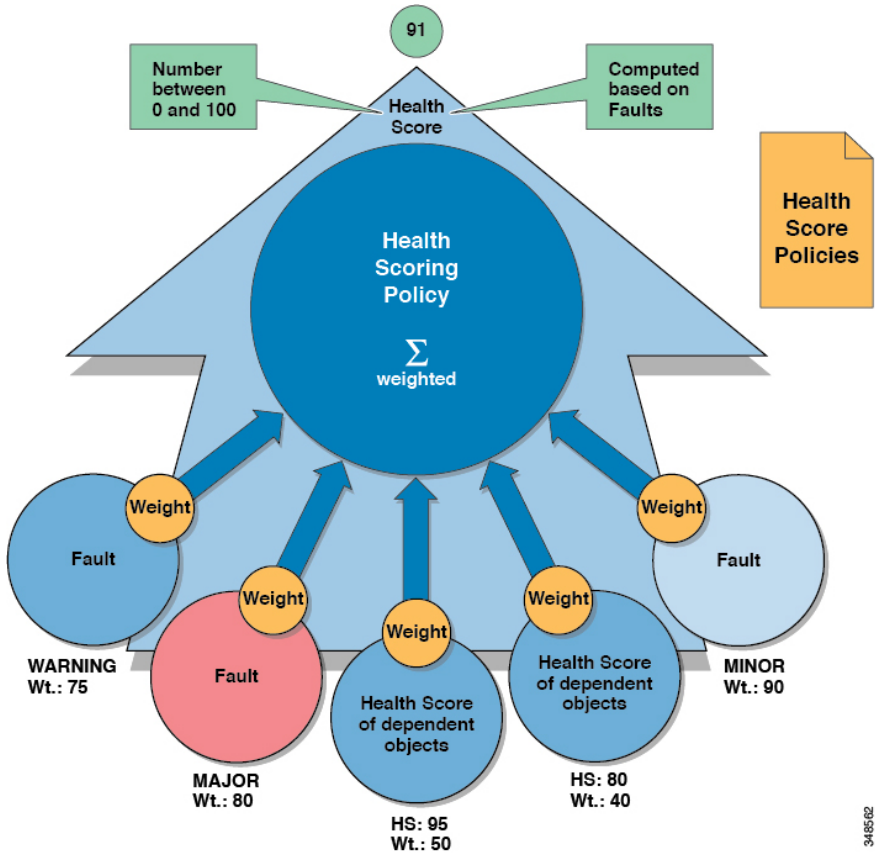


For example, an EPG could be using ports of two leaf switches. Each leaf switch would contain a deployed EPG component. The number of learned endpoints is a weighting factor. Each port could have a different number of learned endpoints. So the EPG health score would be derived by summing the health score of each EPG component times its number of learned endpoints on that leaf, divided by the total number of learned endpoints across the leaf switches the EPG uses.

MO Health Scores

Each managed object (MO) belongs to a health score category. By default, the health score category of an MO is the same as its MO class name.

Figure 135: MO Health Score



Each health score category is assigned an impact level. The five health score impact levels are Maximum, High, Medium, Low, and None. For example, the default impact level of fabric ports is Maximum and the default impact level of leaf ports is High. Certain categories of children MOs can be excluded from health score calculations of its parent MO by assigning a health score impact of None. These impact levels between objects are user configurable. However, if the default impact level is None, the administrator cannot override it.

The following factors are the various impact levels:

Maximum: 100% High: 80% Medium: 50% Low: 20% None: 0%

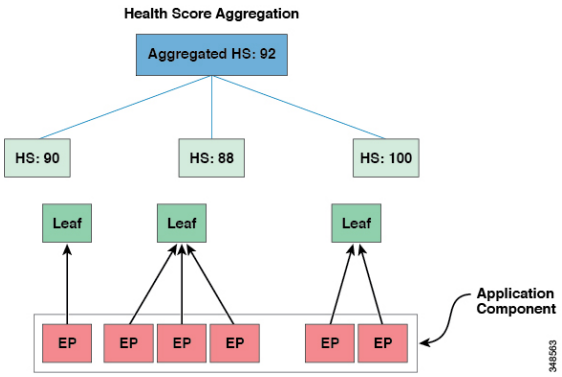
The category health score is calculated using an Lp -Norm formula. The health score penalty equals 100 minus the health score. The health score penalty represents the overall health score penalties of a set of MOs that belong to a given category and are children or direct relatives of the MO for which a health score is being calculated.

The health score category of an MO class can be changed by using a policy. For example, the default health score category of a leaf port is `eqpt:LeafP` and the default health score category of fabric ports is `eqpt:FabP`. However, a policy that includes both leaf ports and fabric ports can be made to be part of the same category called ports.

Health Score Aggregation and Impact

The health score of an application component can be distributed across multiple leaf switches as shown in the following figure.

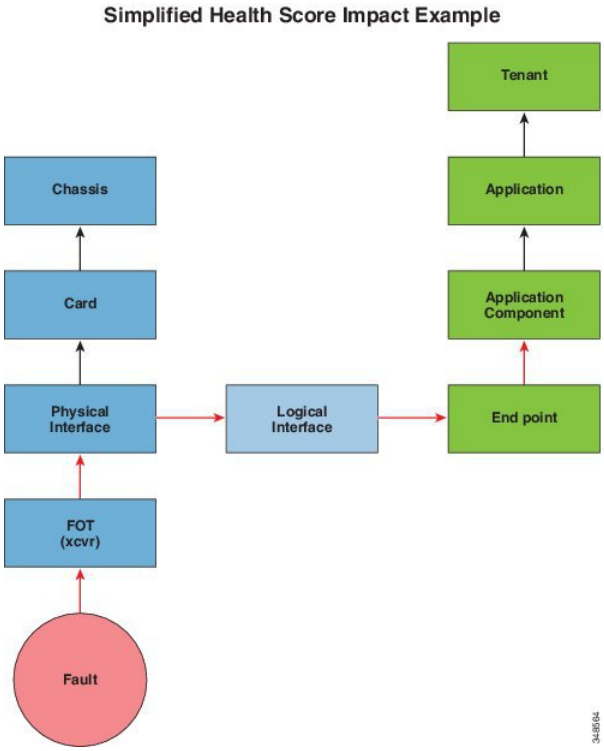
Figure 136: Health Score Aggregation



The aggregated health score is computed at the APIC.

In the following figure, a hardware fault impacts the health score of an application component.

Figure 137: Simplified Health Score Impact Example



About SPAN

You can use the Switched Port Analyzer (SPAN) utility to perform detailed troubleshooting or to take a sample of traffic from a particular application host for proactive monitoring and analysis.

SPAN copies traffic from one or more ports, VLANs, or endpoint groups (EPGs) and sends the copied traffic to one or more destinations for analysis by a network analyzer. The process is nondisruptive to any connected devices and is facilitated in the hardware, which prevents any unnecessary CPU load.

You can configure SPAN sessions to monitor traffic received by the source (ingress traffic), traffic transmitted from the source (egress traffic), or both. By default, SPAN monitors all traffic, but you can configure filters to monitor only selected traffic.

You can configure SPAN on a tenant or on a switch. When configured on a switch, you can configure SPAN as a fabric policy or an access policy.

APIC supports the encapsulated remote extension of SPAN (ERSPAN).

Multinode SPAN

APIC traffic monitoring policies can SPAN policies at the appropriate places to track members of each application group and where they are connected. If any member moves, APIC automatically pushes the policy to the new leaf switch. For example, when an endpoint VMotions to a new leaf switch, the SPAN configuration automatically adjusts.

Additional Information

Refer to the *Cisco APIC Troubleshooting Guide* for detailed information about the configuration, use, and limitations of SPAN.

About SNMP

The Cisco Application Centric Infrastructure (ACI) provides extensive SNMPv1, v2, and v3 support, including Management Information Bases (MIBs) and notifications (traps). The SNMP standard allows any third-party applications that support the different MIBs to manage and monitor the Cisco ACI fabric.

SNMPv3 provides extended security. Each SNMPv3 device can be selectively enabled or disabled for SNMP service. In addition, each device can be configured with a method of handling SNMPv1 and v2 requests.

Beginning in the 5.1(1) release, SNMPv3 supports the Secure Hash Algorithm-2 (SHA-2) authentication type.

For more information about using SNMP, see the *Cisco ACI MIB Quick Reference*.

About Syslog

During operation, a fault or event in the Cisco Application Centric Infrastructure (ACI) system can trigger the sending of a system log (syslog) message to the console, to a local file, and to a logging server on another system. A system log message typically contains a subset of information about the fault or event. A system log message can also contain audit log and session log entries.

Note For a list of syslog messages that the APIC and the fabric nodes can generate, see http://www.cisco.com/c/en/us/td/docs/switches/datacenter/aci/apic/sw/1-x/syslog/guide/aci_syslog/ACI_SysMsg.html.

Many system log messages are specific to the action that a user is performing or the object that a user is configuring or administering. These messages can be the following:

- Informational messages, providing assistance and tips about the action being performed
- Warning messages, providing information about system errors related to an object, such as a user account or service profile, that the user is configuring or administering

In order to receive and monitor system log messages, you must specify a syslog destination, which can be the console, a local file, or one or more remote hosts running a syslog server. In addition, you can specify the minimum severity level of messages to be displayed on the console or captured by the file or host. The local file for receiving syslog messages is `/var/log/external/messages`.

A syslog source can be any object for which an object monitoring policy can be applied. You can specify the minimum severity level of messages to be sent, the items to be included in the syslog messages, and the syslog destination.

You can change the display format for the Syslogs to NX-OS style format.

Additional details about the faults or events that generate these system messages are described in the *Cisco APIC Faults, Events, and System Messages Management Guide*, and system log messages are listed in the *Cisco ACI System Messages Reference Guide*.

Note Not all system log messages indicate problems with your system. Some messages are purely informational, while others may help diagnose problems with communications lines, internal hardware, or the system software.

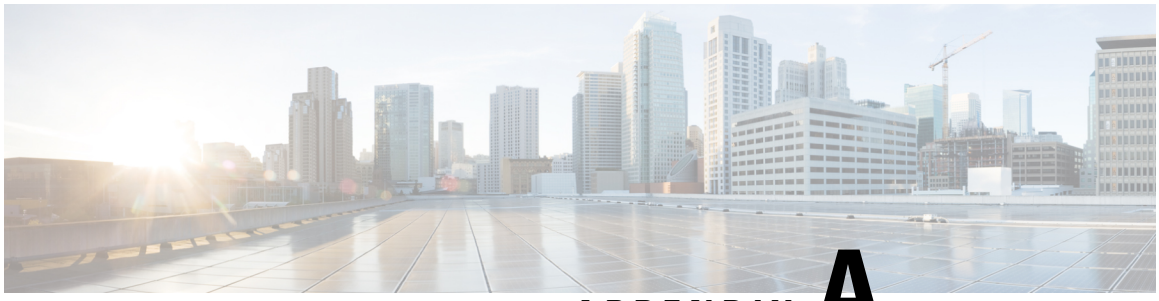
About the Troubleshooting Wizard

The Troubleshooting Wizard allows you to understand and visualize how your network is behaving, which can ease your networking concerns should issues arise. For example, you might have two endpoints that are having intermittent packet loss, but you do not understand why. Using the Troubleshooting Wizard, you can evaluate the issue so that you can effectively resolve the issue rather than logging onto each machine that you suspect to be causing this faulty behavior.

This wizard allows you (the administrative user) to troubleshoot issues that occur during specific time frames for the chosen source and destination. You can define a time window in which you want to perform the debug, and you can generate a troubleshooting report that you can send to TAC.

Related Topics

Getting Started with the Troubleshooting Wizard
Topology in the Troubleshooting Wizard



APPENDIX A

Label Matching

This chapter contains the following sections:

Label Matching

Label matching is used to determine which consumer and provider EPGs can communicate. Contract subjects of a given producer or consumer of that contract determine that consumers and providers can communicate.

The match type algorithm is determined by the matchT attribute that can have one of the following values:

- `All`
- `AtLeastOne` (default)
- `None`
- `AtmostOne`

When both EPG and contract subject labels exist, label matching is done first for EPGs, then for contract subjects.

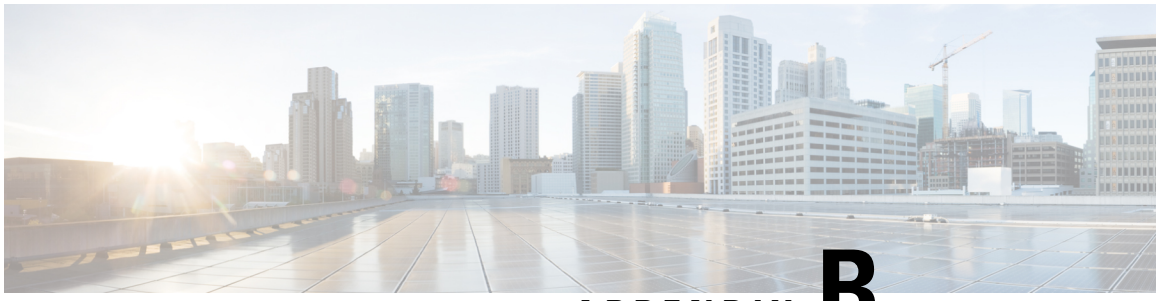
When checking for a match of provider labels, `vzProvLbl`, and consumer labels, `vzConsLbl`, the `matchT` is determined by the provider EPG.

When checking for a match of provider or consumer subject labels, `vzProvSubjLbl`, `vzConsSubjLbl`, in EPGs that have a subject, the `matchT` is determined by the subject.

The same matchT logic is the same for EPG and contract subject labels. The following table shows simple examples of all the EPG and contract subject provider and consumer match types and their results. In this table, a [ ] entry indicates no labels (NULL).

matchT	**vzProvLbl** **vzProvSubLbl**	**vzConsLbl** **vzConsSubLbl**	**Result should be**
`All`	[ ]	[ ]	match
`All`	LabelX, LabelY	LabelX, LabelY	match
`All`	LabelX, LabelY	LabelX, LabelZ	no match

matchT	vzProvLbl vzProvSubLbl	vzConsLbl vzConsSubLbl	Result should be
All	LabelX, LabelY	LabelX	no match
All	LabelX	LabelX, LabelY	match
All	[ ]	LabelX	no match
All	LabelX	[ ]	no match
AtLeastOne	LabelX, LabelY	LabelX	match
AtLeastOne	LabelX, LabelY	LabelZ	no match
AtLeastOne	LabelX	[ ]	no match
AtLeastOne	[ ]	LabelX	no match
AtLeastOne	[ ]	[ ]	match
None	LabelX	LabelY	match
None	LabelX	LabelX	no match
None	LabelX, LabelY	LabelY	no match
None	LabelX	LabelX, LabelY	no match
None	[ ]	LabelX	no match
None	LabelX	[ ]	match
None	[ ]	[ ]	match
AtmostOne	LabelX	LabelX	match
AtmostOne	LabelX, LabelY	LabelX, LabelY	no match
AtmostOne	LabelX, LabelZ	LabelX, LabelY	match
AtmostOne	LabelX	LabelY	no match
AtmostOne	[ ]	LabelX	no match
AtmostOne	LabelX	[ ]	no match
AtmostOne	[ ]	[ ]	match



APPENDIX B

Contract Scope Examples

This chapter contains the following sections:

Contract Scope Examples

Let's say we have EPG1 and EPG2 in VRF1 and EPG3 and EPG4 in VRF2 using a contract called C1, and the `scope = context`.

- EPG1 provides and EPG2 consumes contract C1
- EPG3 provides and EPG4 consumes contract C1

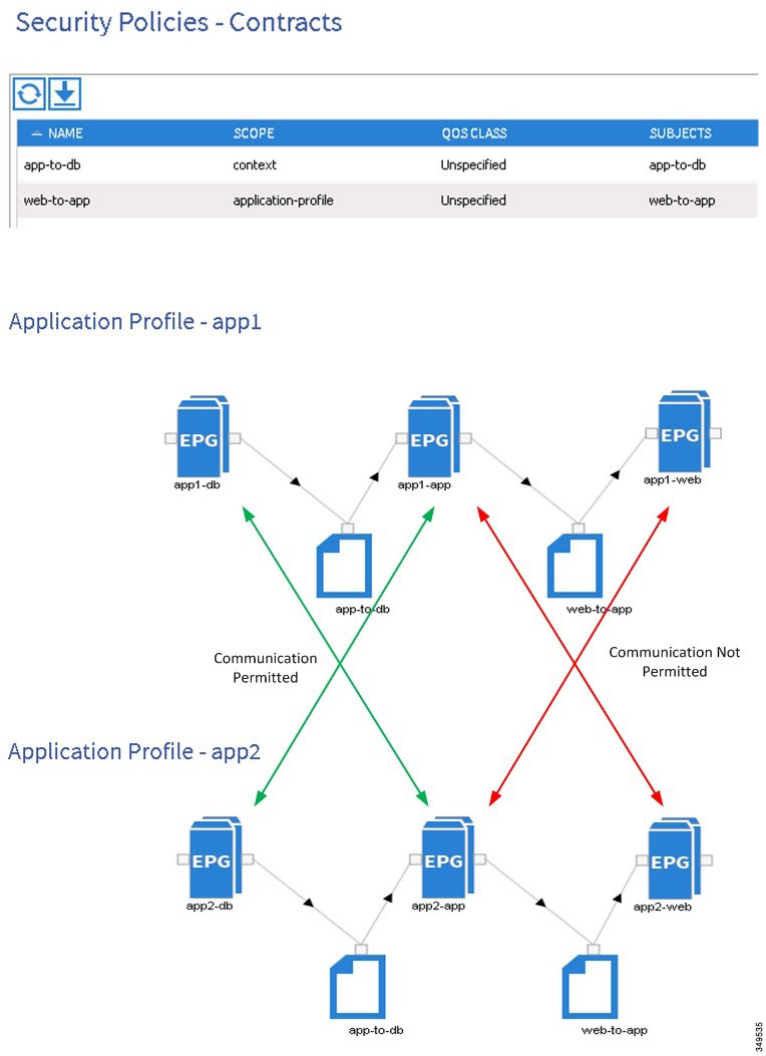
In this example all four EPGs share the same contract, but two of them are in one Virtual Routing and Forwarding (VRF) instance (also known as a context or private network) and two of them in the other VRF. The contract is applied only between EPG1 and EPG2, and then separately between EPG3 and EPG4. The contract is limited to whatever the scope is, which in this case is the VRF.

The same thing applies if the `scope = application profile`. If two application profiles have EPGs and if the `scope = application profile`, then the contract is enforced on EPGs in their application profiles.

Below you see an APIC GUI screenshot of two contracts.

Contract Scope Examples

Figure 138: Security Policy Contract Example



One contract is for web-to-app communication, which has a scope of application profile. The app-to-db contract has a scope of VRF. The app1 and app2 applications profiles are in the same VRF. Each application profile contains EPGs.

Because the scope of the contract app-to-db is enforced at the VRF level, and both application profiles belong to the same VRF, all consumers of the app-to-db contract are allowed to communicate with its provider EPGs.

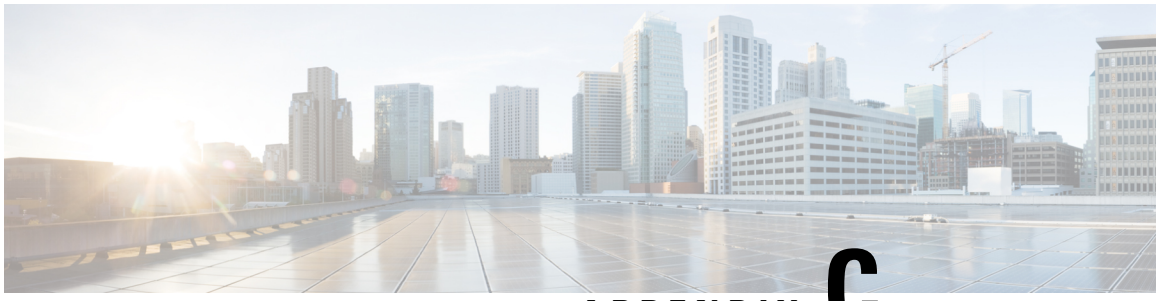
- EPG-app1-db can communicate bi-directionally with EPG-app1-app
- EPG-app2-db can communicate bi-directionally with EPG-app2-app
- EPG-app1-db can communicate bi-directionally with EPG-app2-app
- EPG-app2-db can communicate bi-directionally with EPG-app1-app

The next pairs of endpoints using the web-to-app contracts with a scope of application-profile allow only the provider and consumers of the contract to communicate within that application profile.

- EPG-app1-app can communicate with EPG-app1-web
- EPG-app2-app can communicate with EPG-app2-web

Unlike those above, the app and db EPGs cannot communicate outside of their application profiles.

Contract Scope Examples



APPENDIX C

Secure Properties

This chapter contains the following sections:

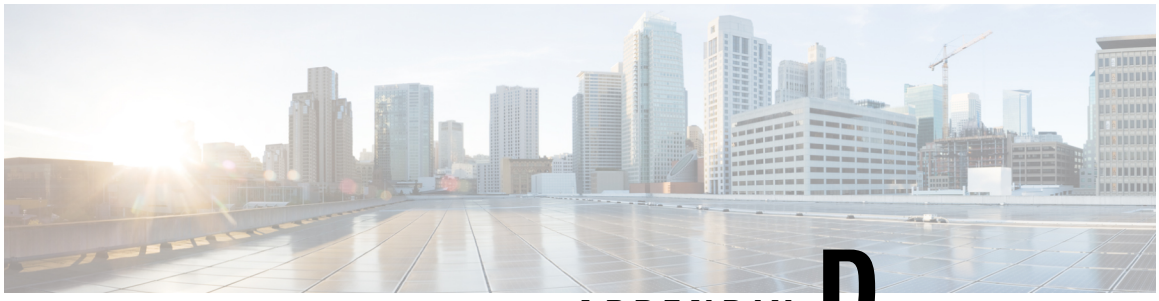
Secure Properties

The table below lists the secure properties of managed objects that include a password field property type.

Property Type	Managed Object Class	Property
Password Field	*pki:KeyRing*	*key*
	pki:WebTokenData	*hashSecret*
	pki:WebTokenData	*initializationVector*
	pki:WebTokenData	*key*
	pki:CsyncSharedKey	*key*
	pki:CertReq	*pwd*
	mcp:Inst	*key*
	mcp:InstPol	*key*
	sysdebug:BackupBehavior	*pwd*
	stats:Dest	*userPasswd*
	firmware:CcoSource	*password*
	firmware:InternalSource	*password*
	f firmware:OSource	*password*
	firmware:Source	*password*
	bgp:PeerDef	*password*
	bgp:Peer	*password*
	bgp:APeerP	*password*
	bgp:PeerP	*password*
	bfd:AuthP	*key*
	comp:UsrAccP	*pwd*
	comp:Ctrlr	*pwd*
	aaa:LdapProvider	*key*
	aaa:LdapProvider	*monitoringPassword*
	aaa:UserData	*pwdHistory*
	aaa:TacacsPlusProvider	*key*
	aaa:TacacsPlusProvidermonitoring	*password*
	aaa:AProvider	*key*

Property Type	Managed Object Class	Property
	aaa:AProvider	*monitoringPassword*
	aaa:RadiusProvider	*key*
	aaa:RadiusProvider	*monitoringPassword*
	aaa:User	*pwd*
	aaa:ChangePassword	*newPassword*
	aaa:ChangePassword	*oldPassword*
	ospf:AuthP	*key*
	ospf:IfPauth	*Key*
	ospf:AIfPauth	*Key*
	ospf:IfDef	*authKey*
	file:RemotePath	*userPasswd*
	file:ARemotePath	*userPasswd*
	vmm:UsrAccP	*pwd*
	snmp:UserSecP	*authKey*
	snmp:UserSecP	*privKey*
	snmp:UserP	*authKey*
	snmp:UserP	*privKey*
	snmp:AUserP	*authKey*
	snmp:AUserP	*privKey*
	vns:VOspfVEncapAsc	*authKey*
	vns:SvcPkgSource	*password*
	vns:SvcPkgSource	*webtoken*
	vns:CCredSecret	*value*

Secure Properties



APPENDIX D

Configuration Zone Supported Policies

This chapter contains the following sections:

Configuration Zone Supported Policies

The following policies are supported for configuration zones:

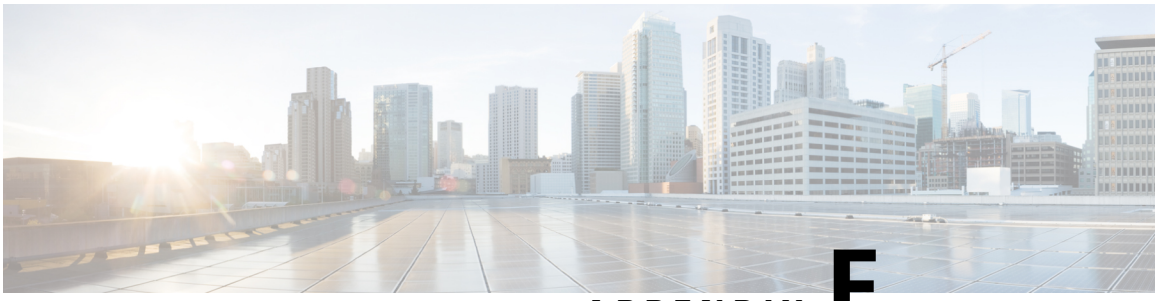
```
analytics:CfgSrv
bgp:InstPol
callhome:Group
callhome:InvP
callhome:QueryGroup
cdp:IfPol
cdp:InstPol
comm:Pol
comp:DomP
coop:Pol
datetime:Pol
dbgexp:CoreP
dbgexp:TechSupP
dhcp:NodeGrp
dhcp:PodGrp
edr:ErrDisRecoverPol
ep:ControlP
ep:LoopProtectP
eqptdiagp:TsOdFabP
eqptdiagp:TsOdLeafP
fabric:AutoGEp
fabric:ExplicitGEp
fabric:FuncP
fabric:HIfPol
fabric:L1IfPol
fabric:L2IfPol
fabric:L2InstPol
fabric:L2PortSecurityPol
fabric:LeCardP
fabric:LeCardPGrp
fabric:LeCardS
fabric:LeNodePGrp
fabric:LePortP
fabric:LePortPGrp
fabric:LFPortS
fabric:NodeControl
fabric:OLeafS
```

Configuration Zone Supported Policies

```
fabric:OSpineS
fabric:PodPGrp
fabric:PortBlk
fabric:ProtGEp
fabric:ProtPol
fabric:SFPortS
fabric:SpCardP
fabric:SpCardPGrp
fabric:SpCardS
fabric:SpNodePGrp
fabric:SpPortP
fabric:SpPortPGrp
fc:DomP
fc:FabricPol
fc:IfPol
fc:InstPol
file:RemotePath
fvns:McastAddrInstP
fvns:VlanInstP
fvns:VsanInstP
fvns:VxlanInstP
infra:AccBaseGrp
infra:AccBndlGrp
infra:AccBndlPolGrp
infra:AccBndlSubgrp
infra:AccCardP
infra:AccCardPGrp
infra:AccNodePGrp
infra:AccPortGrp
infra:AccPortP
infra:AttEntityP
infra:CardS
infra:ConnFexBlk
infra:ConnFexS
infra:ConnNodeS
infra:DomP
infra:FexBlk
infra:FexBndlGrp
infra:FexGrp
infra:FexP
infra:FuncP
infra:HConnPortS
infra:HPathS
infra:HPortS
infra:LeafS
infra:NodeBlk
infra:NodeGrp
infra:NodeP
infra:OLeafS
infra:OSpineS
infra:PodBlk
infra:PodGrp
infra:PodP
infra:PodS
infra:PolGrp
infra:PortBlk
infra:PortP
infra:PortS
infra:PortTrackPol
infra:Profile
infra:SHPathS
infra:SHPortS
infra:SpAccGrp
infra:SpAccPortGrp
```

```
infra:SpAccPortP
infra:SpineP
infra:SpineS
isis:DomPol
l2ext:DomP
l2:IfPol
l2:InstPol
l2:PortSecurityPol
l3ext:DomP
lacp:IfPol
lacp:LagPol
lldp:IfPol
lldp:InstPol
mcp:IfPol
mcp:InstPol
mgmt:NodeGrp
mgmt:PodGrp
mon:FabricPol
mon:InfraPol
phys:DomP
psu:InstPol
qos:DppPol
snmp:Pol
span:Dest
span:DestGrp
span:SpanProv
span:SrcGrp
span:SrcTargetShadow
span:SrcTargetShadowBD
span:SrcTargetShadowCtx
span:TaskParam
span:VDest
span:VDestGrp
span:VSpanProv
span:VSrcGrp
stormctrl:IfPol
stp:IfPol
stp:InstPol
stp:MstDomPol
stp:MstRegionPol
trig:SchedP
vmm:DomP
vpc:InstPol
vpc:KAPol
```

Configuration Zone Supported Policies



APPENDIX E

ACI Terminology

This chapter contains the following sections:

ACI Terminology

Cisco ACI Term	Industry Standard Term (Approximation)	Description
Alias	Alias	A changeable name for a given object. While the name of an object, once created, cannot be changed, the Alias is a field that can be changed. For more details, refer to "Using Tags and Alias" section under "Using the REST API":
API Inspector	—	The API Inspector in the Cisco APIC GUI provides a real-time display of the REST API commands that the Cisco APIC processes to perform GUI interactions.
App Center	—	The Cisco ACI App Center allows you to fully enable the capabilities of the Cisco APIC by writing applications running on the controller. Using the Cisco ACI App Center, customers, developers, and partners are able to build applications to simplify, enhance, and visualize their use cases. These applications are hosted and shared at the Cisco ACI App Center and installed in the Cisco APIC.

Cisco ACI Term	Industry Standard Term (Approximation)	Description
Application Policy Infrastructure Controller (APIC)	Approximation of cluster controller	The Cisco APIC, which is implemented as a replicated synchronized clustered controller, provides a unified point of automation and management, policy programming, application deployment, and health monitoring for the Cisco ACI multitenant fabric. The minimum recommended size for a Cisco APIC cluster is three controllers.
Application Profile	—	An application profile (fvAp) defines the policies, services, and relationships between endpoint groups (EPGs).
Atomic Counters	Atomic Counters	Atomic counters allow you to gather statistics about traffic between leafs. Using atomic counters, you can detect drops and misrouting in the fabric, enabling quick debugging and isolation of application connectivity issues. For example, an administrator can enable atomic counters on all leaf switches to trace packets from endpoint 1 to endpoint 2. If any leaf switches have nonzero counters, other than the source and destination leaf switches, an administrator can drill down to those leaf switches.
Attachable Entity Profile	—	An Attachable Access Entity Profile (AEP) is used to group domains with similar requirements. By grouping domains into AEPs and associating them, the fabric knows where the various devices in the domain live and the Application Policy Infrastructure Controller (APIC) can push the VLANs and policy where it needs to be.
Border Leaf Switches	Border Leaf Switches	Border leaf switches refers to a leaf that is connected to a layer 3 device like external network devices or services such as firewalls and router ports. Other devices like servers can also connect to it.
Bridge Domain	Bridge Domain	A bridge domain is a set of logical ports that share the same flooding or broadcast characteristics. Like a virtual LAN (VLAN), bridge domains span multiple devices.

Cisco ACI Term	Industry Standard Term (Approximation)	Description
Cisco ACI Optimizer	—	The Cisco ACI Optimizer feature in the Cisco APIC GUI is a Cisco APIC tool that enables you to determine how many leaf switches you will need for your network and suggests how to deploy each application and external EPG on each leaf switch without violating any constraints. It can also help you determine if your current setup has what you need, if you are exceeding any limitations, and suggests how to deploy each application and external EPG on each leaf switch.
Cisco Application Virtual Switch (AVS)	—	Cisco AVS is a distributed virtual switch that is integrated with the Cisco ACI architecture as a virtual leaf and managed by the Cisco APIC. It offers different forwarding and encapsulation options and extends across many virtualized hosts and data centers defined by the VMware vCenter server.
Configuration Zones	—	Configuration zones divide the Cisco ACI fabric into different zones that can be updated with configuration changes at different times. This limits the risk of deploying a faulty fabric-wide configuration that may disrupt traffic or even bring the fabric down. An administrator can deploy a configuration to a non-critical zone, and then deploy it to critical zones when satisfied that it is suitable. For more details, refer to: Configuration Zones
Consumer	—	An EPG that consumes a service.
Context or VRF Instance	Virtual Routing and Forwarding (VRF) or Private Network	A virtual routing and forwarding instance defines a Layer 3 address domain that allows multiple instances of a routing table to exist and work simultaneously. This increases functionality by allowing network paths to be segmented without using multiple devices. Cisco ACI tenants can contain multiple VRFs.
Contract	Approximation of Access Control List (ACL)	The rules that specify what and how communication in a network is allowed. In Cisco ACI, contracts specify how communications between EPGs take place. Contract scope can be limited to the EPGs in an application profile, a tenant, a VRF, or the entire fabric.

Cisco ACI Term	Industry Standard Term (Approximation)	Description
Distinguished Name (DN)	Approximation of Fully Qualified Domain Name (FQDN)	A unique name that describes a MO and locates its place in the MIT.
Endpoint Group (EPG)	Endpoint Group	A logical entity that contains a collection of physical or virtual network endpoints. In Cisco ACI, endpoints are devices connected to the network directly or indirectly. They have an address (identity), a location, attributes (e.g., version, patch level), and can be physical or virtual. Endpoint examples include servers, virtual machines, storage, or clients on the Internet.
Fabric	—	The Cisco ACI fabric includes Cisco Nexus 9000 Series switches with the Cisco APIC controller to run in the leaf/spine Cisco ACI fabric mode. These switches form a “fat-tree” network by connecting each leaf node to each spine node; all other devices connect to the leaf nodes. The Cisco APIC manages the Cisco ACI fabric.
Filter	Approximation of Access Control List and approximation of Firewall	Cisco ACI uses a whitelist model: all communication is blocked by default; communication must be given explicit permission. A Cisco ACI filter is a TCP/IP header field, such as a Layer 3 protocol type or Layer 4 ports, that are used to allow inbound or outbound communications between EPGs.
GOLF	—	The Cisco ACI GOLF feature (also known as Layer 3 EVPN Services for Fabric WAN) enables much more efficient and scalable Cisco ACI fabric WAN connectivity. It uses the BGP EVPN protocol over OSPF for WAN routers that are connected to spine switches.
L2 Out	Bridged Connection	A bridged connection connects two or more segments of the same network so that they can communicate. In Cisco ACI, an L2 Out is a bridged (Layer 2) connection between a Cisco ACI fabric and an outside Layer 2 network, which is usually a switch.

Cisco ACI Term	Industry Standard Term (Approximation)	Description
L3 Out	Routed Connection	A routed Layer 3 connection uses a set of protocols that determine the path that data follows in order to travel across multiple networks from its source to its destination. Cisco ACI routed connections perform IP forwarding according to the protocol selected, such as BGP, OSPF, or EIGRP.
Label	—	Label matching is used to determine which consumer and provider EPGs can communicate. Contract subjects of a given producer or consumer of that contract determine that consumers and providers can communicate. A label matching algorithm is used determine this communication. For more details, refer to: ACI Fundamentals Guide
Managed Object (MO)	MO	An abstract representation of network resources that are managed. In Cisco ACI, an abstraction of a Cisco ACI fabric resource.
Management Information Tree (MIT)	MIT	A hierarchical management information tree containing all the managed objects (MOs) of a system. In Cisco ACI, the MIT contains all the MOs of the Cisco ACI fabric. The Cisco ACI MIT is also called the Management Information Model (MIM).
Microsegmentation with Cisco ACI	Microsegmentation, micro-segmentation	Microsegmentation with the Cisco Application Centric Infrastructure (ACI) provides the ability to automatically assign endpoints to logical security zones called endpoint groups (EPGs) based on various network-based or virtual machine (VM)-based attributes.

Cisco ACI Term	Industry Standard Term (Approximation)	Description
Multipod	—	Multipod enables provisioning a more fault-tolerant fabric comprised of multiple pods with isolated control plane protocols. Also, multipod 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, multipod enables provisioning multiple pods per floor or building and providing connectivity between pods through spine switches. Multipod uses MP-BGP EVPN as the control-plane communication protocol between the Cisco ACI spine switches in different pods. For more details, refer to the Multipod White Paper:

Cisco ACI Term	Industry Standard Term (Approximation)	Description
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 they can be shared. Once a fabric administrator configures domains in the Cisco ACI fabric, tenant administrators can associate tenant endpoint groups (EPGs) to domains. A domain is configured to be associated with a VLAN pool. EPGs are then configured to use the VLANs associated with a domain. You can configure the following domain types: • 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 Cisco ACI fabric. • Routed outside network domain profiles (l3extDomP) are used to connect a router to a leaf switch in the Cisco ACI fabric. • Fibre Channel domain profiles (fcDomP) are used to connect Fibre Channel VLANs and VSANs.
Policy	—	Named entity that contains generic specifications for controlling some aspect of system behavior. For example, a Layer 3 Outside Network Policy would contain the BGP protocol to enable BGP routing functions when connecting the fabric to an outside Layer 3 network.
Profile	—	Named entity that contains the necessary configuration details for implementing one or more instances of a policy. For example, a switch node profile for a routing policy would contain all the switch-specific configuration details required to implement the BGP routing protocol.

Cisco ACI Term	Industry Standard Term (Approximation)	Description
Provider	—	An EPG that provides a service.
Quota Management	Quota Management	The Quota management feature enables an admin to limit what managed objects can be added under a given tenant or globally across tenants. Using Quota Management, you can limit any tenant or group of tenants from exceeding Cisco ACI maximums per leaf switch or per fabric or unfairly consuming most available resources, potentially affecting other tenants on the same fabric. For example, a user has configured a bridge domain quota of maximum 6 across the entire ACI policy model with a fault action. The code would be: `apic1(config)# quota fvBD max 6 scope uni exceed-action fault`
REST API	REST API	The Cisco Application Policy Infrastructure Controller (APIC) REST API is a programmatic interface that uses REST architecture. The API accepts and returns HTTP (not enabled by default) or HTTPS messages that contain JavaScript Object Notation (JSON) or Extensible Markup Language (XML) documents. The REST API is the interface into the management information tree (MIT) and allows manipulation of the object model state. The same REST interface is used by the Cisco APIC CLI, GUI, and SDK, so that whenever information is displayed, it is read through the REST API, and when configuration changes are made, they are written through the REST API. The REST API also provides an interface through which other information can be retrieved, including statistics, faults, and audit events. It even provides a means of subscribing to push-based event notification, so that when a change occurs in the MIT, an event can be sent through a web socket.
Schema	—	In a Cisco ACI Multi-Site configuration, the Schema is a container for single or multiple templates that are used for defining policies.

Cisco ACI Term	Industry Standard Term (Approximation)	Description
Site	Site	The Cisco APIC cluster domain or single fabric, treated as a Cisco ACI region and availability zone. It can be located in the same metro-area as other sites, or spaced world-wide.
Stretched ACI	—	Stretched Cisco ACI fabric is a partially meshed design that connects Cisco ACI leaf and spine switches distributed in multiple locations. The stretched fabric is a single Cisco ACI fabric. The sites are one administration domain and one availability zone. Administrators are able to manage the sites as one entity; configuration changes made on any Cisco APIC controller node are applied to devices across the sites. The stretched Cisco ACI fabric preserves live VM migration capability across the sites. Objects (tenants, VRFs, EPGs, bridge-domains, subnets, or contracts) can be stretched when they are deployed to multiple sites.
Subject	Approximation of Access Control List	In Cisco ACI, subjects in a contract specify what information can be communicated and how.
Tags	—	Object tags simplify API operations. In an API operation, an object or group of objects is referenced by the tag name instead of by the distinguished name (DN). Tags are child objects of the item they tag; besides the name, they have no other properties. For more details, refer to "Using Tags and Alias" section under "Using the REST API".
Template	Template	In a Cisco ACI Multi-Site configuration, templates are framework to hold policies and configuration objects that are pushed to the different sites. These templates reside within schemas that are defined for each site.

Cisco ACI Term	Industry Standard Term (Approximation)	Description
Tenant	Tenant	A secure and exclusive virtual computing environment. In Cisco ACI, a tenant is a unit of isolation from a policy perspective, but it does not represent a private network. Tenants can represent a customer in a service provider setting, an organization or domain in an enterprise setting, or just a convenient grouping of policies. Cisco ACI tenants can contain multiple private networks (VRF instances).
vzAny	—	The vzAny managed object provides a convenient way of associating all endpoint groups (EPGs) in a Virtual Routing and Forwarding (VRF) instance to one or more contracts, instead of creating a separate contract relation for each EPG. For more details, see Use vzAny to Automatically Apply Communication Rules to all EPGs in a VRF.