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.

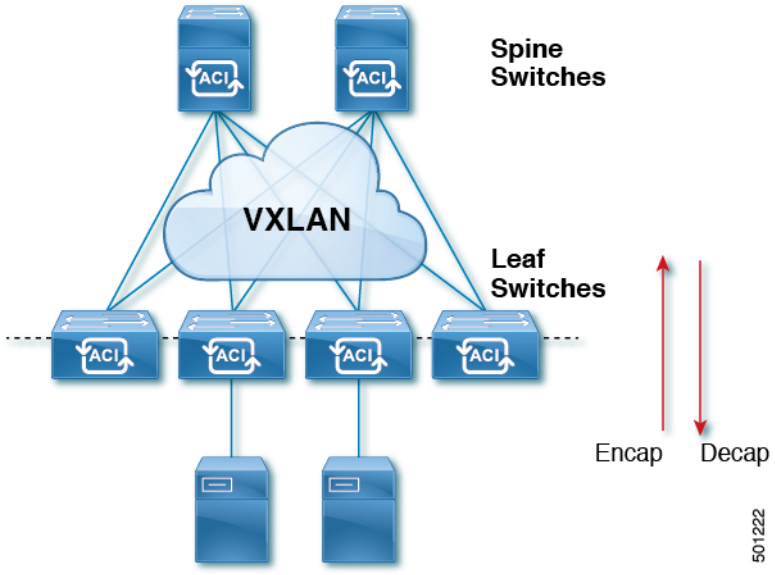
Today, application design drives east-west traffic from server to server through the data center access layer. Applications driving this shift include big data distributed processing designs like Hadoop, live virtual machine or workload migration as with VMware vMotion, server clustering, and multi-tier 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 1: ACI Fabric



As traffic enters the fabric, ACI encapsulates and applies policy to it, forwards it as needed across the fabric through a spine switch (maximum two-hops), and de-encapsulates it upon exiting the fabric. Within the fabric, ACI uses Intermediate System-to-Intermediate System Protocol (IS-IS) and Council of Oracle Protocol (COOP) for all forwarding of endpoint to endpoint communications. This enables all ACI links to be active, equal cost

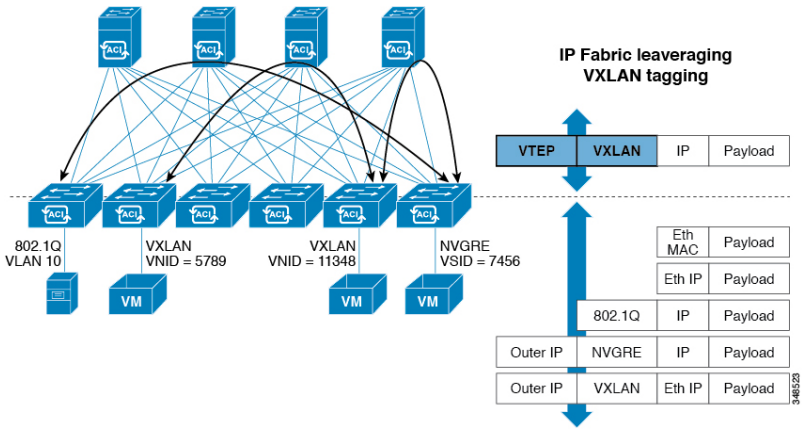
multipath (ECMP) forwarding in the fabric, and fast-reconverging. For propagating routing information between software defined networks within the fabric and routers external to the fabric, ACI uses the Multiprotocol Border Gateway Protocol (MP-BGP).

VXLAN in ACI

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

All traffic in the ACI fabric is normalized as VXLAN packets. At ingress, ACI encapsulates external VLAN, VXLAN, and NVGRE packets in a VXLAN packet. The following figure shows ACI encapsulation normalization.

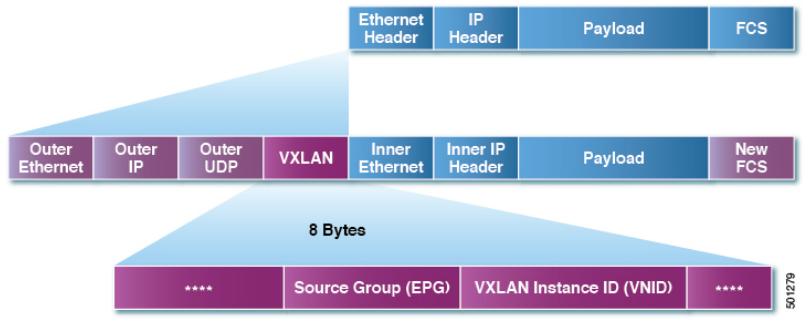
Figure 2: ACI Encapsulation Normalization



Forwarding in the ACI fabric is not limited to or constrained by the encapsulation type or encapsulation overlay network. An ACI bridge domain forwarding policy can be defined to provide standard VLAN behavior where required.

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

Figure 3: ACI VXLAN Packet Format



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

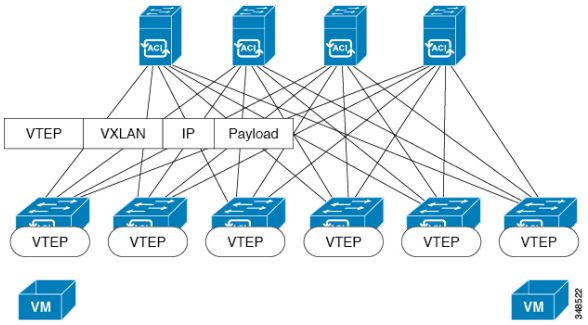
VXLAN enables ACI to deploy Layer 2 virtual networks at scale across the fabric underlay Layer 3 infrastructure. Application endpoint hosts can be flexibly placed in the data center network without concern for the Layer 3 boundary of the underlay infrastructure, while maintaining Layer 2 adjacency in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

Layer 3 VNIDs Facilitate Transporting Inter-subnet Tenant Traffic

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

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

Figure 4: ACI Decouples Identity and Location



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

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

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

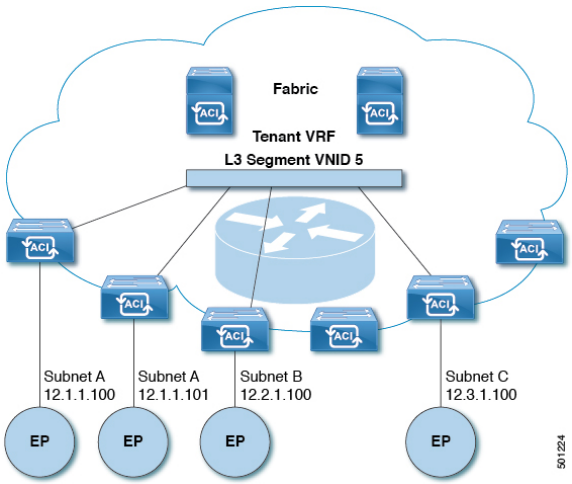
The VTEP in ACI maps the internal tenant MAC or IP address to a location using a distributed mapping database. After the VTEP completes a lookup, the VTEP sends the original data packet encapsulated in

VXLAN with the destination address of the VTEP on the destination leaf switch. The destination leaf switch de-encapsulates the packet and sends it to the receiving host. With this model, ACI uses a full mesh, single hop, loop-free topology without the need to use the spanning-tree protocol to prevent loops.

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

The following figure shows how routing within the tenant is done.

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



For each tenant VRF in the fabric, ACI assigns a single L3 VNID. ACI transports traffic across the fabric according to the L3 VNID. At the egress leaf switch, ACI routes the packet from the L3 VNID to the VNID of the egress subnet.

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

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

Note Cisco ACIdoes 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.

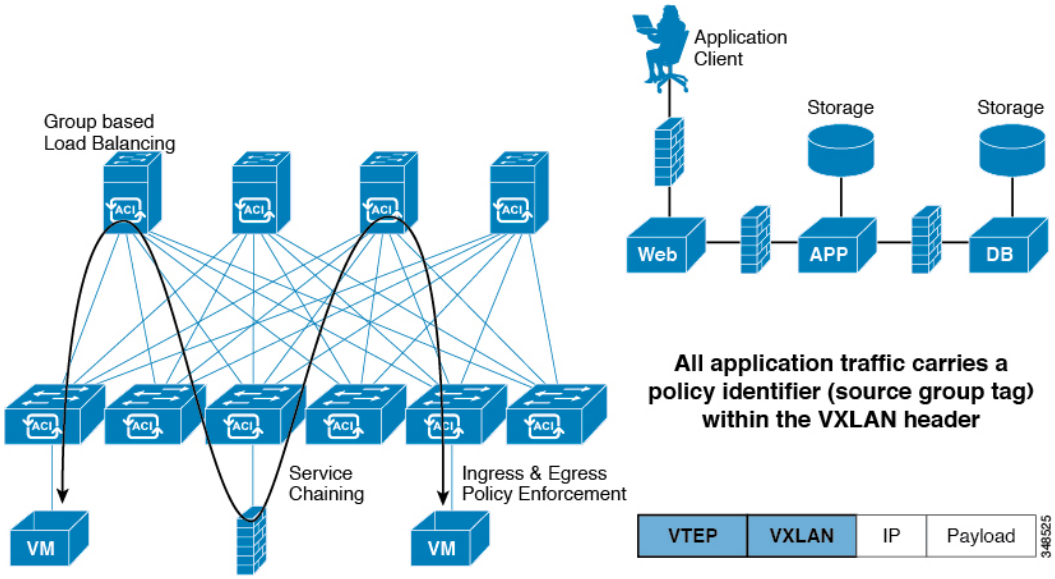
For the appropriate MTU values for each platform, see the relevant configuration guides.

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.

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 6: Policy Identification and Enforcement



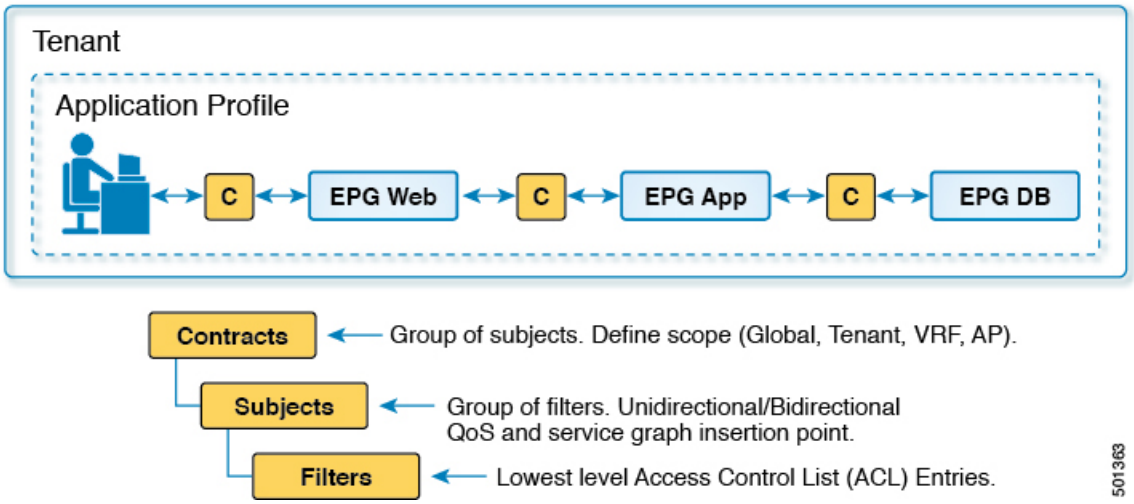
Fabric and access policies govern the operation of internal fabric and external access interfaces. The system automatically creates default fabric and access policies. Fabric administrators (who have access rights to the entire fabric) can modify the default policies or create new policies according 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 7: 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.

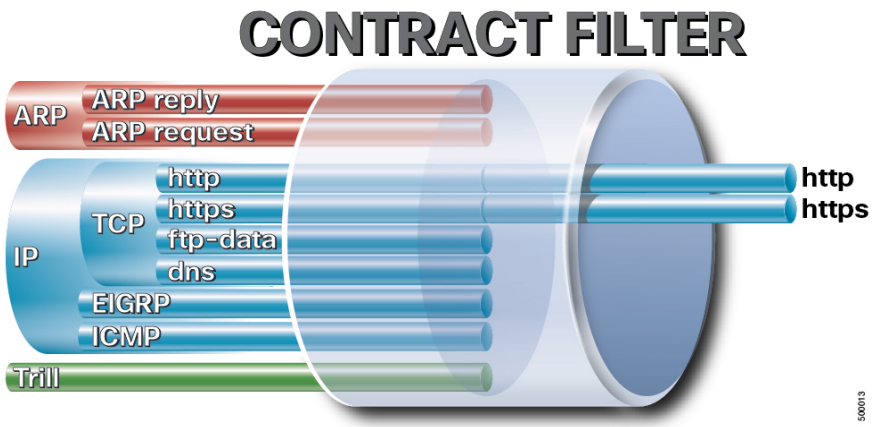
There is also directionality in the relationship between EPGs and contracts. EPGs can either provide or consume a contract. An EPG that provides a contract is typically a set of endpoints that provide a service to a set of client devices. The protocols used by that service are defined in the contract. 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

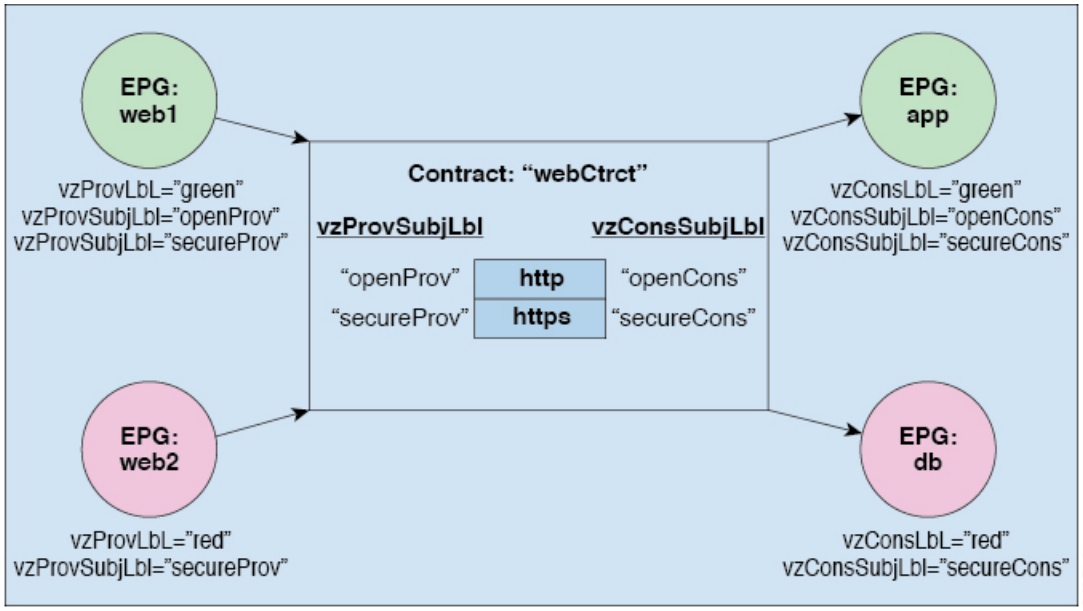
The contract is constructed in a hierarchical manner. It consists of one or more subjects, each subject contains one or more filters, and each filter can define one or more protocols.

Figure 8: Contract Filters



The following figure shows how contracts govern EPG communications.

Figure 9: 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.

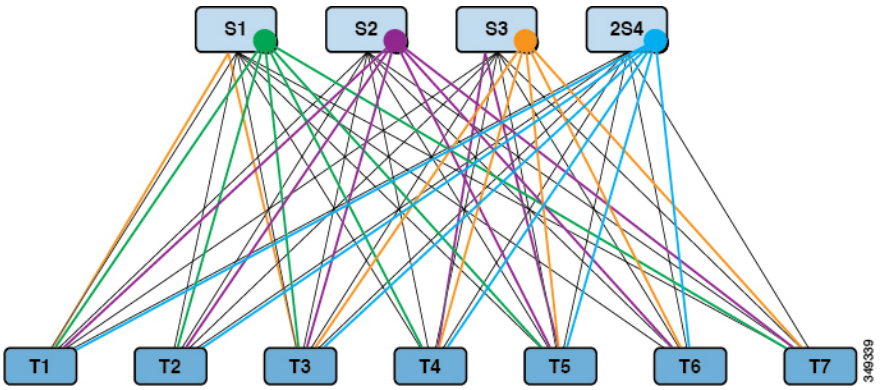
The ACI fabric consists of spine and leaf switches that are connected in a Clos topology (named after Charles Clos) where traffic that enters an ingress interface can be routed through any of the available middle stage spine switches, to the relevant egress switch. The leaf switches have two types of 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 10: 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 1: 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.

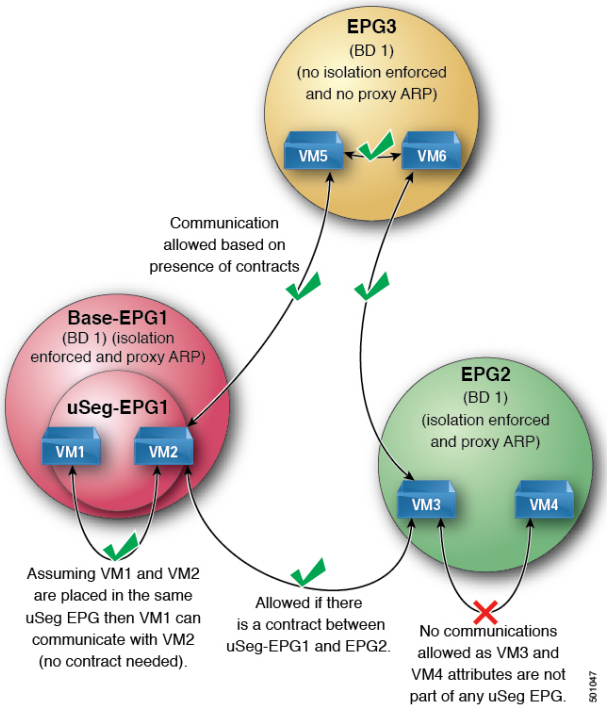
To address this type of behavior, we recommend configuring the NIC pair as the two legs of a VPC to achieve an Active/Active setup. If the server hardware does not support the Active/Active configuration (for example a blade chassis), then an active/standby type of NIC pair configuration will also prevent 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 11: 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 12: VM1 to VM2 Communication is Desired.

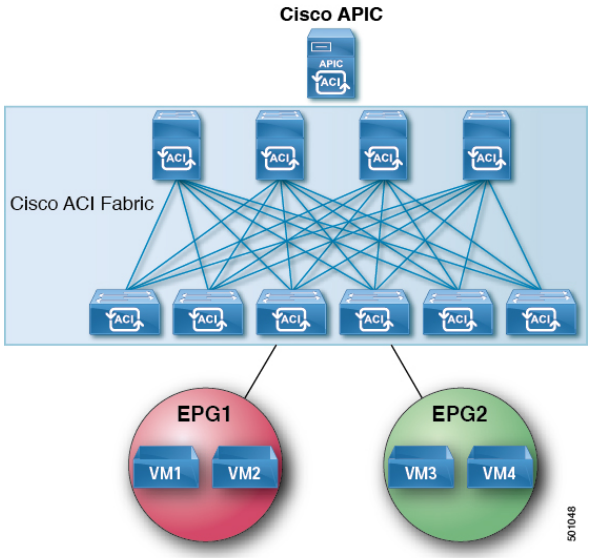


Table 2: ARP Table State

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

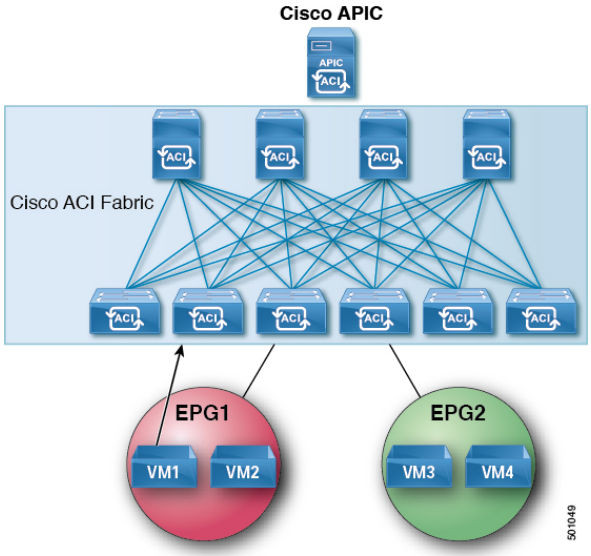


Table 3: ARP Table State

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 14: ACI Fabric Floods the Proxy ARP Request within the BD

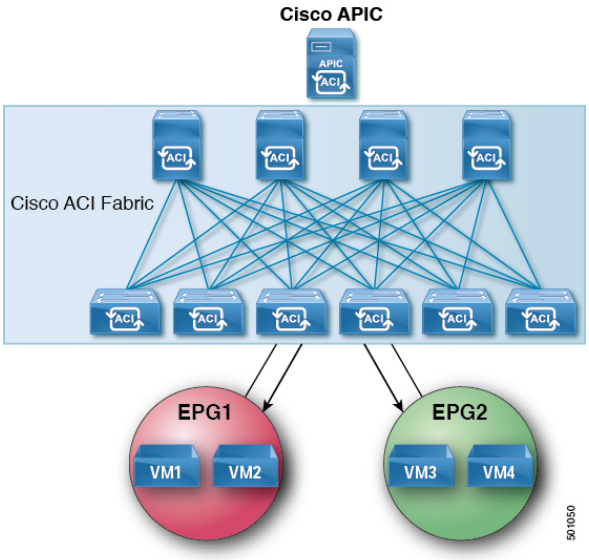


Table 4: ARP Table State

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 15: VM2 Sends an ARP Response to the ACI Fabric

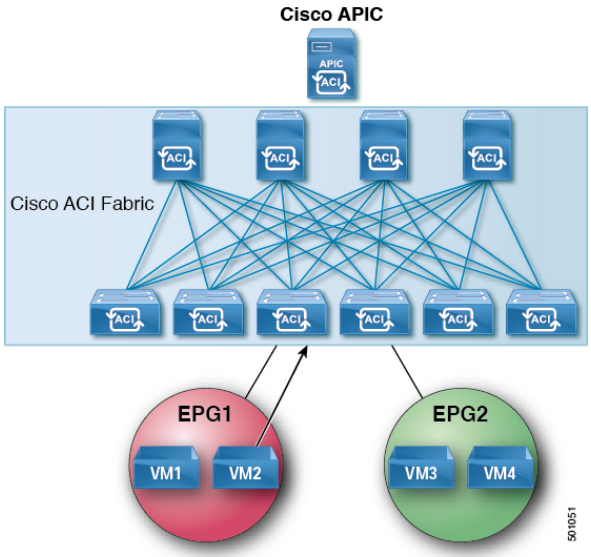


Table 5: ARP Table State

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 16: VM2 is Learned

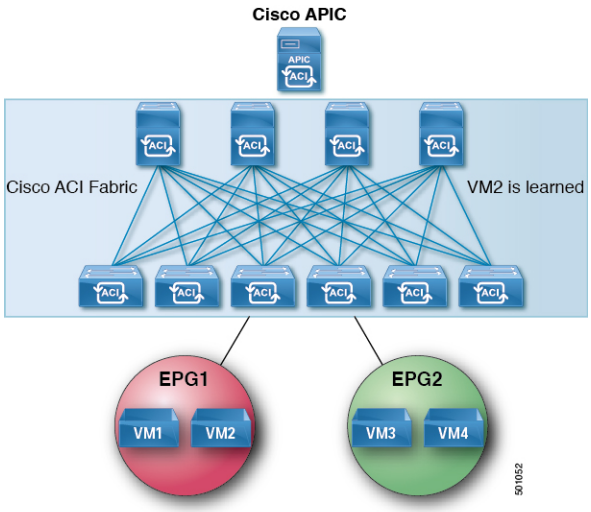


Table 6: ARP Table State

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

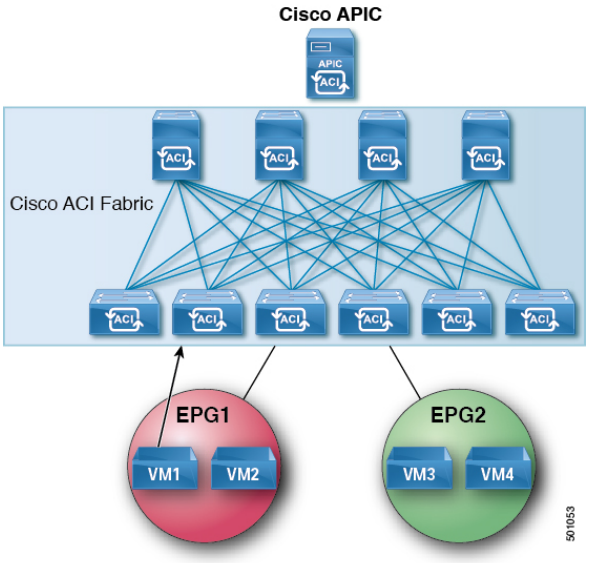


Table 7: ARP Table State

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 18: ACI Fabric Sends a Proxy ARP Response to VM1

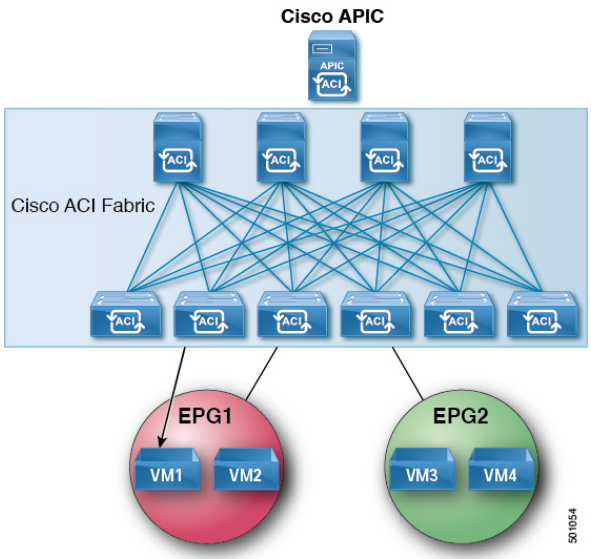


Table 8: ARP Table State

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.