Cisco Nexus 9000 Series NX-OS ePBR Configuration Guide, Release 10.5(x)

First Published: 2024-11-27

Last Modified: 2025-04-23

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

CONTENTS

Preface

This preface includes the following sections:

Audience

This publication is for network administrators who install, configure, and maintain Cisco Nexus switches.

Document Conventions

Command descriptions use the following conventions:

Convention	Description
bold	Bold text indicates the commands and keywords that you enter literally as shown.
Italic	Italic text indicates arguments for which you supply the values.
[x]	Square brackets enclose an optional element (keyword or argument).
[x | y]	Square brackets enclosing keywords or arguments that are separated by a vertical bar indicate an optional choice.
{x | y}	Braces enclosing keywords or arguments that are separated by a vertical bar indicate a required choice.
[x {y | z}]	Nested set of square brackets or braces indicate optional or required choices within optional or required elements. Braces and a vertical bar within square brackets indicate a required choice within an optional element.

Convention	Description
`variable`	Indicates a variable for which you supply values, in context where italics cannot be used.
string	A nonquoted set of characters. Do not use quotation marks around the string or the string includes the quotation marks.

Examples use the following conventions:

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

Related Documentation for Cisco Nexus 9000 Series Switches

The entire Cisco Nexus 9000 Series switch documentation set is available at the following URL:

https://www.cisco.com/en/US/products/ps13386/tsd_products_support_series_home.html

Documentation Feedback

To provide technical feedback on this document, or to report an error or omission, please send your comments to nexus9k-docfeedback@cisco.com. We appreciate your feedback.

Communications, services, and additional information

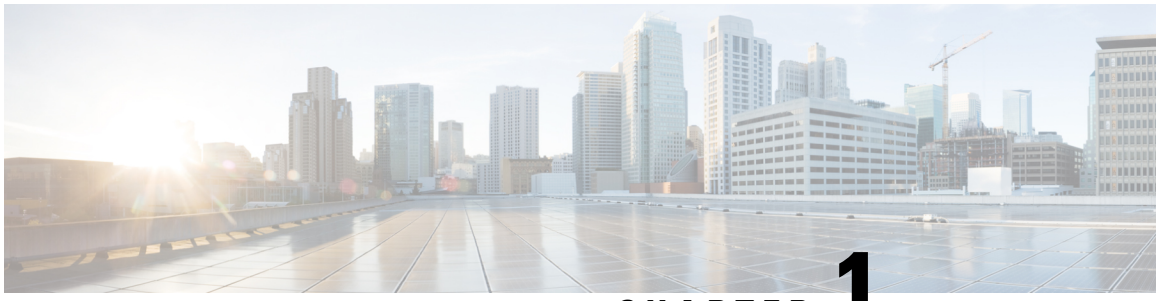
- To receive timely, relevant information from Cisco, sign up at Cisco Profile Manager.
- To get the business impact you're looking for with the technologies that matter, visit Cisco Services.
- To submit a service request, visit Cisco Support.
- To discover and browse secure, validated enterprise-class apps, products, solutions, and services, visit Cisco DevNet.
- To obtain general networking, training, and certification titles, visit Cisco Press.
- To find warranty information for a specific product or product family, access Cisco Warranty Finder.

Cisco Bug Search Tool

Cisco Bug Search Tool (BST) is a gateway to the Cisco bug-tracking system, which maintains a comprehensive list of defects and vulnerabilities in Cisco products and software. The BST provides you with detailed defect information about your products and software.

Documentation feedback

To provide feedback about Cisco technical documentation, use the feedback form available in the right pane of every online document.



CHAPTER 1

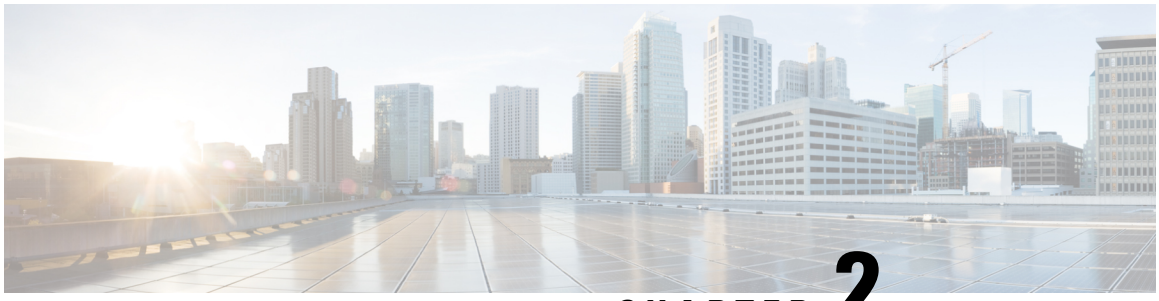
New and Changed Information

New and Changed Information

Table 1: New and Changed Features for Cisco NX-OS Release 10.5(x)

Feature	Description	Changed in Release	Where Documented
Service Chaining with service threshold	Added support for service chaining with service threshold.	10.5(3)F	Guidelines and Limitations, on page 67 Configuring ePBR Service, on page 69
Support for Layer 3 ePBR policies	Added the set-vrf option to redirect packets through a specified VRF instance for ePBR inter-VRF deployments.	10.5(2)F	Guidelines and Limitations for ePBR L3, on page 9 Configuring ePBR Set-VRF for an ePBR Policy, on page 22
Multi-Node and Multi-Site Service Chaining	Added support for multi node and multi site ePBR service chaining	10.5(2)F	ePBR Service and Service-chain, on page 55 ePBR and GPO Multi-Site , on page 61 Guidelines and Limitations, on page 67 Configuring ePBR Service-chain, on page 71 Configuring Failover Group, on page 72 Configuration Examples for SGACL service-chaining Configuration, on page 74

Feature	Description	Changed in Release	Where Documented
ePBR IPACL one-arm endpoints	Configuring a reverse IP address for one-arm service devices is not necessary.	10.5(1)F	Guidelines and Limitations for ePBR L3, on page 9 Configuring ePBR Service, Policy, and Associating to an Interface, on page 14



CHAPTER 2

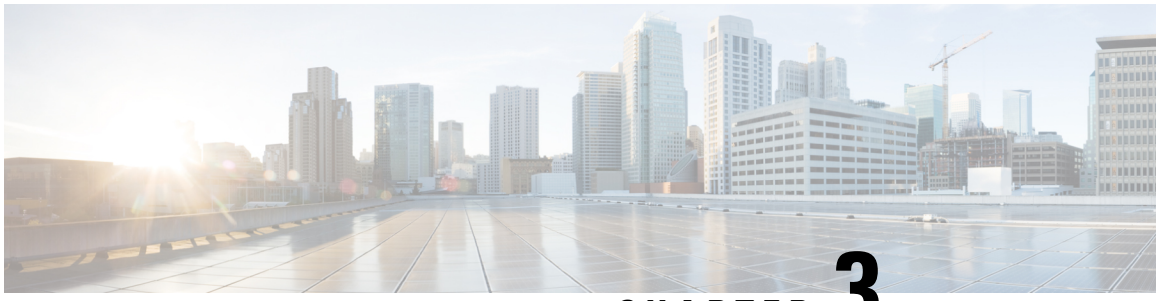
Overview

Licensing Requirements

For a complete explanation of Cisco NX-OS licensing recommendations and how to obtain and apply licenses, see the *Cisco NX-OS Licensing Guide* and the *Cisco NX-OS Licensing Options Guide* .

Supported Platforms

Starting with Cisco NX-OS release 7.0(3)I7(1), use the Nexus Switch Platform Support Matrix to know from which Cisco NX-OS releases various Cisco Nexus 9000 and 3000 switches support a selected feature.



CHAPTER 3

Configuring ePBR L3

This chapter describes how to configure Enhanced Policy-based Redirect (ePBR) on Cisco NX-OS devices.

Information About ePBR L3

Enhanced Policy-based Redirect (ePBR) in Elastic Services Re-direction (ESR) provides traffic redirection and service chaining across the NX-OS and fabric topologies by leveraging policy-based redirect solution and achieves service chaining without adding extra headers, and avoids latency in using extra headers.

ePBR enables application-based routing and provides a flexible, device-agnostic policy-based redirect solution without impacting application performance. The ePBR service flow includes the following tasks:

Licensing Requirements

For a complete explanation of Cisco NX-OS licensing recommendations and how to obtain and apply licenses, see the *Cisco NX-OS Licensing Guide* and the *Cisco NX-OS Licensing Options Guide* .

Configuring ePBR Service and Policy

You must first create an ePBR service which defines the attributes of service end points. Service end points are the service appliances such as firewall, IPS, etc., that can be associated with switches. You can also define probes to monitor the health of the service end points and can define the forward and reverse interfaces where the traffic policies are applied. ePBR also supports load balancing along with service chaining. ePBR allows you to configure multiple service end points as a part of the service configuration.

In events of service endpoint failures, traffic that was redirecting to the failed service endpoints will redirect to other reachable service endpoints, configured in the ePBR service. Resilient hashing is supported during endpoint failures for an ePBR service that has multiple service endpoints. Traffic that was always being redirected to a specific service endpoint continues to redirect to the same device in events of failures of other service endpoints that are part of the same service.

Beginning with Cisco NX-OS Release 10.2(1)F, the VRF of every service in a chain may either be unique or may be exactly identical. The service endpoints and interfaces defined for a service, should pertain to the VRF defined for the service.

Service end-point interfaces having an existing IPv4 PBR policy cannot be used inside an IPv4 ePBR service. Similarly service end-point interfaces having an existing ipv6 PBR policy cannot be used inside an IPv6 ePBR service.

After creating the ePBR service, you must create an ePBR policy. The ePBR policy allows you to define traffic selection, redirection of traffic to the service end point and various fail-action mechanisms on the end point health failure. You may use IP access-list end points with permit access control entries (ACE) to define the traffic of interest to match and take the appropriate action.

The ePBR policy supports multiple ACL match definitions. A match can have multiple services in a chain which can be sequenced by a sequence number. This allows flexibility to add, insert, and modify elements in a chain in a single service policy. In every service sequence, you can define the fail action method such as drop, forward, and bypass. The ePBR policy allows you to specify source or destination-based load balancing and bucket counts in order to have granular load balancing of traffic.

Applying ePBR to an Interface

After creating the ePBR policy you need to apply the policy on an interface. This allows you to define the interface at which the traffic ingresses into the NX-OS or Nexus fabric. You can also apply the policy in both the forward and reverse directions. There may only be two IPv4/IPv6 policies applied to the interface, one in the forward and one in the reverse direction.

Beginning with Cisco NX-OS Release 10.2(1)F, ePBR supports policy application on layer-3 port-channel sub-interfaces

Beginning with Cisco NX-OS Release 10.2(1)F, the interface on which the ePBR policy is applied may be on a different VRF than the VRF of the services in the chain.

ePBR IPv4 policies cannot be applied to an interface on which an IPv4 PBR policy is already applied. ePBR IPv6 policies cannot be applied to an interface on which an IPv6 PBR policy is already applied.

Creating Bucket and Load Balancing

ePBR computes the number of traffic buckets based on the service that has maximum number of service-end-points in the chain. If you configure the load balance buckets, your configuration will have the precedence. ePBR supports load balancing methods of source IP and destination IP but does not support L4-based source or destination load balancing methods.

ePBR Service Endpoint Out-of-Service

The ePBR service endpoint Out-of-Service feature provides the option to temporarily remove the endpoint from service. The following two methods can be used to move an endpoint Out-of-Service:

1. **Administrative Out-of-Service**: This method is used during maintenance or upgrades to temporarily move the service-endpoints to operationally down state and avoid sending traffic to the node, while still retaining the service-endpoints as a valid endpoint device in service.

The user would also require the ability to bring the service endpoint back in service on the Cisco NX-OS switch after the maintenance procedure has been completed. This is a standard feature provided used by load-balancers in the industry today.

2. **Auto Out-of-Service**: This method is used during recovery of endpoints after failures, ePBR detects the reachability of the endpoints getting re-established and attempts to redirect subsets of flows back to the node.

   Also, when certain networks may be tolerant to a rare endpoint failure and recovery but may require detecting endpoints that are losing and re-establishing connectivity constantly, each event disrupting end-to-end connections twice. It may be desirable to put such nodes Out-of-Service.

ePBR Object Tracking, Health Monitoring, and Fail-Action

ePBR creates SLA and Track objects based on the probe types configured in the service and supports various probes and timers such as ICMP, TCP, UDP, DNS, and HTTP. ePBR also supports user defined tracks, which allows you to create tracks with various parameters including milli second probes in associating with ePBR.

ePBR monitors the health of the end points by provisioning IP SLA probes and object tracks to track the IP SLA reachability when you apply the ePBR probe configuration.

You can configure the ePBR probe options for a service or for each of the forward or reverse end points. You can also configure frequency, timeout, retry up and down counts, and source loopback interface so that they can be used for source IP of an IP SLA session. The retry-up and down counts are used as multipliers for the frequency to determine **delay-up** and **delay-down** intervals. Once the service endpoint is initially detected as failed or recovered, the system will act on these events after the expiry of these intervals. You can define any type of tracks and associate them with the forward or the reverse end points. The same track objects is re-used for all policies using the same ePBR service.

You can define tracks separately and assign the track ID to each service-end point in ePBR. If you do not assign any user-defined track to an endpoint, ePBR will create a track using probe method for the end point. If no probe method is defined at the end point level, the probe method configured for the service level will be used.

ePBR supports the following fail-action mechanisms for its service chain sequences:

- Bypass
- Drop on Fail
- Forward

Bypass of a service sequence indicates that the traffic must be redirected to the next service sequence when there is a failure of the current sequence.

Drop on fail of a service sequence indicates that the traffic must be dropped when all the service-end-points of the service become unreachable.

Forward is the default option and indicates that upon failure of the current service, traffic should use the regular routing tables. This is the default fail-action mechanism.

Note Symmetry is maintained when fail-action bypass is configured for all the services in the service chain. In other fail-action scenarios, when there are one or more failed services, symmetry is not maintained in the forward and the reverse flow.

ePBR Session-based Configuration

ePBR sessions allow addition, deletion or modification of the following aspects of in-service services or policies. The in-service refers to a service that is associated with a policy that has been applied to an active interface or a policy that is being modified and currently configured on an active interface.

- Service endpoints with their interfaces and probes
- Reverse endpoints and probes
- Matches under policies
- Load-balance methods for matches
- Match sequences and fail-action

Note In ePBR Sessions, you cannot move interfaces from one service to another service in the same session. To move interfaces from one service to another service, perform the following steps:

1. Use a session operation to first remove it from the existing service.
2. Use a second session operation to add it to the existing service.

ePBR Multi-Site

Beginning with Cisco NX-OS Release 10.2(1)F, service-chaining in a VXLAN multisite fabric can be achieved by using the following configuration and topology guidelines.

- Endpoints in a service or services in the chain may be distributed across different leaf switches, in the same or different site.
- Every service should be in its unique VRF, which is different from the tenant VRF context in which the ePBR policy is applied.
- To segregate traffic for different tenant VRFs, the VLANs used for the services would be required to be segregated and new services and policies would need to be defined.
- Tenant VRF routes should be leaked to each of the service VRFs on every leaf switch hosting the services, to allow traffic to be routed back at the end of the service chain to its destination, in the tenant VRF.
- VNIs should be symmetrically allocated across different leaf switches and sites.
- The ePBR policy should be enabled on all layer-3 VNIs of the service VRFs being used, on all leaf switches hosting services and on the border leaf or border gateway switches, if it is acting as transit for multi-site.

- The service chain may be isolated to one site entirely, with traffic arriving from different sites. Although this scenario doesn't involve multi-site distribution of service devices, the layer-3 VNIs of the service VRFs on the border gateways or border leafs should only be treated as multi-site transit and the ePBR policy should be applied on them. The ePBR policy should be also applied on the host or tenant facing interfaces in the remote sites where the traffic is arriving from.

ACL Refresh

ePBR session ACL refresh allows you to update the policy generated ACLs, when the user-provided ACL gets modified or added or deleted with ACEs. On the refresh trigger, ePBR will identify the policies that are impacted by this change and create or delete or modify the buckets' generated ACLs for those policies.

For ePBR scale values, see Cisco Nexus 9000 Series NX-OS Verified Scalability Guide.

Guidelines and Limitations for ePBR L3

ePBR has the following guidelines and limitations:

- The L3 ePBR feature requires sufficient ing-racl TCAM to function properly. To verify the current TCAM carving, use the **show hardware access-list tcam region** command. If the appropriate TCAM size is not allocated, use the **hardware access-list tcam region ing-racl** *size multiple of 256* command to allocate the appropriate TCAM.
- Beginning with Cisco Nexus NX-OS Release 10.1(2), ePBR with IPv4 and IPv6 is supported on N9K-C93108TC-FX3P switch.
- Beginning with Cisco NX-OS Release 10.1(1) each match statement under ePBR policy can support three action types - redirect, drop, and exclude. There can be only one drop and/or exclude match statement per policy. The ACE rules for the traffic, which needs to be excluded or dropped in the forward as well as the reverse directions, should be manually added to the match access-list that is used with the action of exclude or drop. The statistics for the exclude and drop match access-list may display traffic hit counters for both directions.
- ePBR policies require at least one match with redirect action.
- Beginning with Cisco NX-OS Release 10.1(1), ePBR with IPv4, IPv6 and ePBR over VXLAN are supported on below platform switches: N9K-C9316D-GX, N9K-C93600CD-GX, N9K-C9364C-GX, N9K-C93180YC-FX3S, N9K-C93360YC-FX3, and N9K-C93108TC-FX3P.
- When fail-action is specified in any match statement, probe is mandatory in the configuration.
- Whenever there is OTM track changes ePBR statistics is reset due to RPM reprogramming.
- Do not share the same user defined ACL across multiple match statements in the ePBR configuration.
- Symmetry in traffic is maintained only when fail-action bypass is configured for ePBR Service. For the other fail-actions such as forward/drop in the service chain, symmetry is not maintained for the forward and reverse flow of traffic.
- Unique layer-4 source and destination port parameters should be specified for the match filters if traffic is required to match any source and any destination IP as per the match access-list definition, and is required to be redirected to devices distributed in a VXLAN environment in both forward and reverse directions or service-chained through one-arm devices.

- Feature ePBR and feature ITD cannot co-exist with the same ingress interface.
- With scaled ePBR configuration, it is recommended to remove the policies before you use the **no feature epbr** command.
- It is recommended that you classify probe traffic in a separate CoPP class. Otherwise, probe traffic will go in the default CoPP class and might be dropped causing IP SLA bouncing for probe traffic. For information on CoPP configuration for IP SLA, see Configuring CoPP for IP SLA Packets.
- ePBR is supported on the Cisco Nexus 9500 and Cisco Nexus 9300 platform switches with EX, FX, and FX2 line cards.
- Beginning with Cisco NX-OS Release 9.3(5) Catena feature is deprecated.
- If you want to remove the ePBR service endpoint which is configured to a port-channel that is removed from the system, perform the following steps:
  1. Delete the existing ePBR policy.
  2. Delete the existing ePBR service.
  3. Reconfigure the ePBR service endpoint to the required port-channel.
- Please do not modify the dynamically created access-list entries of ePBR that begin with the name “epbr_”. These access-lists are reserved for ePBR internal use.

Note Modifying these prefix strings can cause the ePBR to not function properly and would impact ISSU.

- Router ACLs may be enabled alongside layer-3 ePBR policies on supported layer-3 interfaces, only when statistics is not enabled for either ePBR policies or the router ACLs. See **Guidelines and Limitations for Policy-Based Routing** in the Policy-based routing chapter of *Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide* for details on this limitation.
- On Cisco Nexus N9K-C9316D-GX, N9K-C93600CD-GX, and N9K-C9364C-GX switches, before performing ISSD from Cisco NX-OS, Release 10.2 and later releases to Release 10.1 disable ePBR policies and proceed with downgrade.
- ePBR policy definition can be applied to a maximum of 32 interfaces of supported interface types across forward and reverse directions.
- Beginning with Cisco NX-OS Release 10.4(1)F, ePBR supports IPv4 and IPv6 policies on GRE and IP-IP tunnel interfaces for load-balancing and redirection on Cisco Nexus 9300-FX2/FX3/GX/GX2 platform switches:
- Beginning with Cisco NX-OS Release 10.4(1)F, ePBR supports redirection or load-balancing to Layer-3 endpoints reachable over IP-IP and GRE tunnel interfaces on Cisco Nexus 9300-FX2/FX3/GX/GX2 platform switches.

Note

- ePBR IPv6 policies are not supported on IP-IP tunnel interfaces.
- Currently ePBR does not support service-chaining to devices reachable over IP-IP and GRE tunnels.

- Configuration rollback and configuration replace are supported only when the ePBR policy is not associated with any interfaces and the ePBR service definitions are not used in any active ePBR policy in both the source and target configurations. However, configuration rollback and configuration replace do not support policy to interface association and disassociation.
- Disabling the atomic update may allow more TCAM resources to be made available for the ePBR policies, but it may cause possible disruption in traffic during configuration changes to the policies or during fail-over and recovery of service endpoints. For further details, see **Atomic ACL Updates** section of *Cisco Nexus 9000 Series NX-OS Security Configuration Guide*.
- Unique policies are generated for every interface that is configured with an ePBR policy. Additionally unique policies are also generated for every service interface that needs to steer the traffic to the next service function inside a service-chain configured for a match inside an ePBR policy. The scale of supported EPBR policies may vary with the available ACL labels in the system for PBR policies. For further details on ACL labels sizes, see **Maximum Label Sizes Supported for ACL Types** section of *Cisco Nexus 9000 Series NX-OS Security Configuration Guide*.
- The ePBR service or endpoint hold-down timers used should be compatible with the probe (track and IP SLA) frequencies and timeouts in use, so that the failures can be detected in time.
- Endpoint states of the forward and reverse arms of dual-arm devices are not synchronized automatically. If this is needed, identical probe track configuration on the forward and reverse arms should be used. Probe tracks configured for endpoints may be shared between the forward and reverse arms of the same endpoints, but not across endpoints in the same or different services.
- Beginning with Cisco NX-OS Release 10.5(1)F, it is no longer necessary to explicitly configure the reverse IP address for one-arm service devices. If a service endpoint is not assigned a reverse IP address, it will be treated as a one-arm device, and traffic will be redirected to the same IP address in both the forward and reverse directions.
- If the IP address of a loopback interface associated with a service probe is modified, you need to remove and reapply any policies and contracts that reference the service.
- Beginning with Cisco NX-OS Release 10.5(2)F, ePBR will support the **set-vrf** command to redirect packets through a specified VRF instance on Cisco Nexus 9300-FX2, FX3, GX, GX2, H2R and H1 Series switches with the following limitations:
  - The **source-vrf** and **destination-vrf** cannot be modified or deleted through an ePBR session.
  - The **set-vrf** is not supported for ePBR on VXLAN.
  - The **set-vrf** doesn't switch VRFs for drop and exclude traffic.

The following guidelines and limitations apply to ePBR over VXLAN feature:

- In VXLAN fabric, service chaining cannot be done to devices within same VLAN. All devices must be present in separate VLANs.

- When every service in the chain is in the same VRF, ePBR is only supported at a single site in a VXLAN multisite fabric.
- When every service in the chain is in the same VRF:
  - Active/Standby chain is supported with two service nodes with no restrictions.
  - Active/Standby chain with three or more service nodes in chain requires no two nodes of different type behind same service leaf.
  - In VXLAN fabric you cannot stitch traffic from one service in a leaf and come back later to the same leaf.

Note These restrictions are not applicable if every service in the chain is in a different VRF context.

- When service endpoints are distributed in a VXLAN environment or on VPC peers, the service endpoints must be configured in an identical order on all switches.
- For service-endpoints distributed in a VXLAN environment, you must configure source loopback interfaces for the probe, so that a unique source IP may be used for IP SLA sessions.
- The ePBR policy should always be originally applied on host or tenant facing interfaces. The ePBR policy should be applied on Layer-3 VNI interfaces pertaining to the tenant or service VRFs only as the transit interfaces.

  Only traffic arriving for the endpoints in the specific VRF will be redirected by the policies applied on the layer-3 VNI interfaces pertaining to that VRF. The statistics for the traffic matching the policy on the layer-3 VNI interface will not be visible via ePBR statistics command.
- Beginning with Cisco NX-OS Release 10.3(3)F, you can apply an ePBR Layer 3 policy on a new L3VNI interface on Cisco Nexus 9300-EX, 9300-FX, 9300-FX2, 9300-FX3, 9300-GX, and 9300-GX2 platform switches, and Cisco Nexus 9500 platform switches with 9700-X line cards.

The following guidelines and limitations apply to the match ACL feature:

- Only ACEs with the permit method are supported in the ACL. ACEs with any other method (such as deny or remark) are ignored.
- A maximum of 256 permit ACEs are supported in one ACL.
- ACEs with object-groups specified as address-groups or port-groups in either source or destination parameters are not supported.
- Beginning with Cisco NX-OS Release 10.4(1)F, the Layer-4 port ranges and other port operations (such as 'not equal to', 'greater than', 'lesser than') in the match access-list rules will be honored and used for filtering traffic in the bucket access-lists.
- The configuration **hardware access-list lou resource threshold** must be used for optimal utilization of TCAM ACEs, while using layer-4 port operators in access-lists. For more information on the command, see **Configuring IP ACLs** section of *Cisco Nexus 9000 Series NX-OS Security Configuration Guide*.

The following guidelines and limitations apply to inter-VRF service chaining:

- Beginning with Cisco NX-OS 10.2(1)F Release, every service in a chain should either exist in the same VRF or completely unique VRFs.
- In version 10.2(1)F, fail-action bypass mechanism is not supported when every service in the chain exists in a unique VRF.
- From Cisco NX-OS 10.2(2)F Release, fail-action bypass is supported when the services in the chain are in unique VRFs.
- If the services are in a different VRF than the VRF context of the interface on which the ePBR policy is applied, the user should ensure that the tenant routes are leaked to every service VRF, in order to ensure that the traffic is able to route back to the tenant VRF, at the end of the service chain.
- From Cisco NX-OS Release 10.2(2)F, PBR allows multiple backup next-hops related to different VRFs to be configured for a route-map sequence. This allows ePBR to enable fail-action bypass from service pertaining to one VRF to another effectively.
- Beginning with Cisco NX-OS Release 10.2(3)F, to minimize traffic disruptions during session operations of endpoint additions, service sequence additions, deletions, and modifications, it is recommended to have load-balance buckets configured ahead and avoid modification to the load-balance configuration. Ensure that the configured buckets for load-balance are greater than the number of endpoints configured in services for every sequence in the chain.

The following guidelines and limitations applies if you have configured ePBR using source IP-based load balancing:

- The prefix length in the source IPv4 of the ACE cannot be /32
- The prefix length in the source IPv6 address of the ACE cannot be /128
- The subnet for the source address must be compatible with the buckets configured.

The following guidelines and limitations applies if you have configured ePBR using destination IP-based load balancing:

- The prefix length in the destination IPv4 of the ACE cannot be /32
- The prefix length in the destination IPv6 address of the ACE cannot be /128
- The subnet for the destination address must be compatible with the buckets configured.

The following guidelines and limitations applies if you have configured ePBR service endpoint Out-of-Service feature:

- ePBR service endpoint Out-of-Service feature is supported for Layer-3 services on Cisco Nexus 9300-EX/FX/FX2/FX3/GX/GX2, C9364C, C9332C, and Cisco Nexus 9500 switches with 9700-EX/FX/GX line cards.
- ePBR out-of-service (shut or hold-down) requires endpoint to be configured with probes, either at a endpoint level or a service level.
- When the service is being used by a policy that is active, ePBR out-of-service (shut or hold-down) must be configured using **epbr sessions** only.

The following guidelines and limitations applies if you are using source IP-based load balancing and load-balancing traffic to more than 1 endpoint:

- The source IPv4 subnet mask of the ACE inside the match access-list cannot be /32, or the subnet mask of the source IPv6 address inside the match access-list cannot be /128.
- The destination IPv4 subnet mask of the ACE inside the match access-list cannot be /32, or the subnet mask of the source IPv6 address inside the match access-list cannot be /128.
- The subnet masks for the source address or destination address inside the match access-list, based on the load-balance method, must be compatible with the buckets configured for the match or must be compatible with the number of buckets required, based on the number of endpoints in the services being used for the match.

Configuring ePBR L3

Before you begin

Make sure you have configured IP SLA and PBR features before configuring the ePBR feature.

Configuring ePBR Service, Policy, and Associating to an Interface

The following section provides information about configuring the ePBR Service, ePBR Policy, and associating the policy on to an interface.

SUMMARY STEPS

1. **configure terminal**
2. **epbr service** *service-name*
3. [**no**] **probe** {**icmp** | *l4-proto port-number* [**control** *status*] | **http get** [*url-name* [**version** *ver*] | **dns host***host-name* **ctp**} [**frequency** *freq-num* | **timeout** *seconds* | **retry-down-count** *down-count* | **retry-up-count** *up-count* | **source-interface** *src-intf* | **reverse** *rev-src-intf*]
4. **vrf** *vrf-name*
5. **service-endpoint** {**ip** *ipv4 address* | **ipv6** *ipv6 address*} [**interface** *interface-name interface-number*]
6. **probe track** *track ID*
7. **reverse ip** *ip address* **interface** *interface-name interface-number*
8. **exit**
9. **epbr policy** *policy-name*
10. **match** { [**ip address** *ipv4 acl-name*] | [**ipv6 address** *ipv6 acl-name*] } [**redirect** | **drop** | **exclude**]
11. [**no**] **load-balance** [**method** { **src-ip** | **dst-ip**}] [**buckets** *sequence-number*] [**mask-position** *position-value*]
12. *sequence-number* **set service** *service-name* [**fail-action** { **bypass** | **drop** | **forward**}]
13. **interface** *interface-name interface-number*
14. **epbr** { **ip** | **ipv6**} **policy** *policy-name* [**reverse**]
15. **exit**

DETAILED STEPS

Procedure

	Command or Action	Purpose

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

Step 2	**epbr service** *service-name* **Example:** `switch(config)# epbr service firewall`	Creates a new ePBR service.
Step 3	[**no**] **probe** {**icmp** | *l4-proto port-number* [**control** *status*] | **http get** [*url-name* [**version** *ver*] | **dns host***host-name* **ctp**} [**frequency** *freq-num* | **timeout** *seconds* | **retry-down-count** *down-count* | **retry-up-count** *up-count* | **source-interface** *src-intf* | **reverse** *rev-src-intf*] **Example:** `switch(config)# probe icmp`	Configures the probe for the ePBR service. The probe types supported are ICMP, TCP, UDP, DNS, and HTTP, CTP. The options are as follows: • frequency—Specifies the frequency of the probe in seconds. The range is from 1 to 604800. • retry-down-count —Specifies the number of recounts undertaken by the probe when the node goes down. The range is from 1 to 5. • retry-up-count —Specifies the number of recounts undertaken by the probe when the node comes back up. The range is from 1 to 5. • timeout —Specifies the length of the timeout period in seconds. The range is from 1 to 604800.
Step 4	**vrf** *vrf-name* **Example:** `switch(config)# vrf tenant_A`	Specifies the VRF for the ePBR service.
Step 5	**service-endpoint** {**ip** *ipv4 address* | **ipv6** *ipv6 address*} [**interface** *interface-name interface-number*] **Example:** `switch(config-vrf)# service-endpoint ip 172.16.1.200 interface VLAN100`	Configures service endpoint for the ePBR service. You can repeat steps 2 to 5 to configure another ePBR service.
Step 6	**probe track** *track ID* **Example:** `switch(config-vrf)# probe track 30`	Defines a track separately and assign an existing track ID to each service-endpoint in ePBR. You can assign track ID to each endpoint.
Step 7	**reverse ip** *ip address* **interface** *interface-name interface-number*	Defines the reverse IP and interfaces where the traffic policies are applied.

	Command or Action	Purpose
	Example: `switch(config-vrf)# reverse ip 172.16.30.200 interface VLAN201`	**Note** Beginning with Cisco NX-OS Release 10.5(1)F, it is no longer necessary to explicitly configure the reverse IP address for one-arm service devices. If a service endpoint is not assigned a reverse IP address, it will be treated as a one-arm device, and traffic will be redirected to the same IP address in both the forward and reverse directions.
Step 8	**exit** **Example:** `switch(config-vrf)# exit`	Exits VRF configuration mode and enters global configuration mode.
Step 9	**epbr policy** *policy-name* **Example:** `switch(config)# epbr policy Tenant_A-Redirect`	Configures the ePBR policy.
Step 10	**match** { [**ip address** *ipv4 acl-name*] | [**ipv6 address** *ipv6 acl-name*] } [**redirect** | **drop** | **exclude**] **Example:** `switch(config)# match ip address WEB`	Matches an IPv4 or IPv6 address against an IP or IPv6 ACLs. Redirect is the default action for a match traffic. Drop is used when the traffic needs to be dropped on the incoming interface. Exclude option is used to exclude certain traffic from service-chaining on the incoming interface. You can repeat this step to match multiple ACLs based on the requirement.
Step 11	[**no**] **load-balance** [**method** { **src-ip** | **dst-ip**}] [**buckets** *sequence-number*] [**mask-position** *position-value*] **Example:** `switch(config)# load-balance method src-ip mask-position 3`	Computes the load balance method and the number of buckets to be used by the ePBR service. Beginning with Cisco NX-OS Release 10.3(3)F, the **mask-position** option is provided to choose the bits used for load-balancing in user-defined ACL. Default value is 0. If mask-position is configured, the load-balance bits start from configured mask-position. Based on number of buckets needed, more bits are taken to generate load-balancing buckets, toward the most-significant bit. **Note** For any ACE in user-defined ACLs, if bits used to generate load-balancing buckets overlap with the user-defined subnet, the mask position for the ACE will be reset internally to 0.
Step 12	*sequence-number* **set service** *service-name* [**fail-action** { **bypass** | **drop** | **forward**}] **Example:** `switch(config)# set service firewall fail-action drop`	Computes the fail-action mechanism.

	Command or Action	Purpose

Step 13	**interface** *interface-name interface-number*
	Example:
	`switch(config)# interface vlan 2010`
	`switch(config)# interface vni500001`
		Configures an interface and enters interface configuration mode.
		Note
		Beginning with Cisco NX-OS Release 10.3(3)F, you can apply an ePBR L3 policy on a new L3VNI interface.

Step 14	**epbr** { **ip** | **ipv6**} **policy** *policy-name* [**reverse**] **Example:** `switch(config-if)# epbr ip policy Tenant_A-Redirect`	An interface may be associated at any time with one or more of the following: • an IPV4 policy in the forward direction • an IPv4 policy in the reverse direction • an IPv6 policy in the forward direction • an IPv6 policy in the reverse direction
Step 15	**exit** **Example:** `switch(config-if)# end`	Exits interface configuration mode and returns to global configuration mode.

Modifying a Service Using ePBR Session

The following steps explain how to modify a service using ePBR session.

SUMMARY STEPS

1. **epbr session**
2. **epbr service** *service-name*
3. [**no**] **service-endpoint** {**ip** *ipv4 address* | **ipv6** *ipv6 address*} [**interface** *interface-name interface-number*]
4. **service-endpoint** {**ip** *ipv4 address* | **ipv6** *ipv6 address*} [**interface** *interface-name interface-number*]
5. **reverse ip** *ip address* **interface** *interface-name interface-number*
6. **commit**
7. **abort**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**epbr session** **Example:** `switch(config)# epbr session`	Enters ePBR session mode.

	Command or Action	Purpose
Step 2	**epbr service** *service-name* **Example:** `switch(config-epbr-sess)# epbr service TCP_OPTIMIZER`	Specifies the configured ePBR service in the ePBR session mode.
Step 3	[**no**] **service-endpoint** {**ip** *ipv4 address* | **ipv6** *ipv6 address*} [**interface** *interface-name interface-number*] **Example:** `switch(config-epbr-sess-svc)# no service-end-point ip 172.16.20.200 interface VLAN200`	Disables the configured service endpoint for the ePBR service.
Step 4	**service-endpoint** {**ip** *ipv4 address* | **ipv6** *ipv6 address*} [**interface** *interface-name interface-number*] **Example:** `switch(config-epbr-sess-svc)#service-end-point ip 172.16.25.200 interface VLAN200`	Modifies the service endpoint and replaces the IP for the ePBR service.
Step 5	**reverse ip** *ip address* **interface** *interface-name interface-number* **Example:** `switch(config-epbr-sess-svc-ep)# reverse ip 172.16.30.200 interface VLAN201`	Defines the reverse IP and interfaces where the traffic policies are applied.
Step 6	**commit** **Example:** `switch(config-epbr-sess)# commit`	Completes the modification of the ePBR service using the ePBR session. **Note** Restart the ePBR session after you complete this step.
Step 7	**abort** **Example:** `switch(config-epbr-sess)# abort`	Aborts the session and clears or resets the current configuration under the session. Use this command to abandon the current session configuration in case of errors or unsupported configuration identified during commits. **Note** Restart a new ePBR session after this with the rectified configuration.

Modifying a Policy Using ePBR Session

The following steps explain how to modify a policy using ePBR Session.

SUMMARY STEPS

1. **epbr session**
2. **epbr policy** *policy-name*
3. [**no**] **match** { [**ip address** *ipv4 acl-name*] | [**ipv6 address** *ipv6 acl-name*] [**l2 address** *ipv6 acl-name*]} **vlan** {**vlan** | **vlan range** | **all**} [**redirect** | **drop** | **exclude**] }

4. **match** { [**ip address** *ipv4 acl-name*] | [**ipv6 address** *ipv6 acl-name*] [**l2 address** *ipv6 acl-name*]} **vlan** {**vlan** | **vlan range** | **all**} [**redirect** | **drop** | **exclude**] }
5. *sequence-number* **set service** *service-name* [**fail-action** { **bypass** | **drop** | **forward**}]
6. [**no**] **load-balance** [**method** { **src-ip** | **dst-ip**}] [**buckets** *sequence-number*] [**mask-position** *position-value*]
7. **commit**
8. **end**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**epbr session** **Example:** `switch(config)# epbr session`	Enters ePBR session mode.
Step 2	**epbr policy** *policy-name* **Example:** `switch(config-epbr-sess)# epbr policy Tenant_A-Redirect`	Specifies the configured ePBR policy in the ePBR session mode.
Step 3	[**no**] **match** { [**ip address** *ipv4 acl-name*]	[**ipv6 address** *ipv6 acl-name*] [**l2 address** *ipv6 acl-name*]} **vlan** {**vlan**	**vlan range**	**all**} [**redirect**	**drop**	**exclude**] } **Example:** `switch(config-epbr-sess-pol)# no match ip address WEB`	Disables the IP address matching against the IP or IPv6 ACLs.
Step 4	**match** { [**ip address** *ipv4 acl-name*]	[**ipv6 address** *ipv6 acl-name*] [**l2 address** *ipv6 acl-name*]} **vlan** {**vlan**	**vlan range**	**all**} [**redirect**	**drop**	**exclude**] } **Example:** `switch(config-epbr-sess-pol)# match ip address HR`	Modifies the IP address matching against the IP or IPv6 ACLs.
Step 5	*sequence-number* **set service** *service-name* [**fail-action** { **bypass**	**drop**	**forward**}] **Example:** `switch(config-epbr-sess-pol-match)# set service firewall fail-action drop`	Adds, modifies, or deletes sequences for a match, or modifies the fail-action for an existing sequence.
Step 6	[**no**] **load-balance** [**method** { **src-ip**	**dst-ip**}] [**buckets** *sequence-number*] [**mask-position** *position-value*] **Example:** `switch(config-epbr-sess-pol-match)# load-balance method src-ip mask-position 3`	Computes the load balance method and the number of buckets to be used by the ePBR service. **Note** On omitting this configuration in the session context while modifying the service-chain for an existing match, the

	Command or Action	Purpose
		load-balance configuration for the match will be reset to default. Beginning with Cisco NX-OS Release 10.3(3)F, the **mask-position** option is provided to choose the bits used for load-balancing in user-defined ACL. Default value is 0. If mask-position is configured, the load-balance bits start from configured mask-position. Based on number of buckets needed, more bits are taken to generate load-balancing buckets, toward the most-significant bit. **Note** For any ACE in user-defined ACLs, if bits used to generate load-balancing buckets overlap with the user-defined subnet, the mask position for the ACE will be reset internally to 0.
Step 7	**commit** **Example:** `switch(config-epbr-sess)#commit`	Completes the modification of the ePBR policy using the ePBR session.
Step 8	**end** **Example:** `switch(config-epbr-sess)#end`	Exits the ePBR session mode.

Updating the Access-list Used by ePBR Policies

The following steps explain how to update the access-list used by ePBR policies:

SUMMARY STEPS

1. **epbr session access-list** *acl-name* **refresh**
2. **end**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**epbr session access-list** *acl-name* **refresh** **Example:** `switch(config)# epbr session access-list WEB refresh`	Updates or refreshes the policy generated ACLs.

	Command or Action	Purpose
Step 2	**end** **Example:** `switch(config)# end`	Exits the global configuration mode.

Configuring ePBR Service Endpoint Out-of-Service

The following section provides information about configuring the ePBR Service Endpoint Out-of-Service.

SUMMARY STEPS

1. **configure terminal**
2. **epbr service** *service-name*
3. [**no**] **shut**
4. **service-endpoint** [**interface** *interface-name interface-number*]
5. [**no**] **hold-down threshold count** *threshold count* **time** *threshold time*

DETAILED STEPS

Procedure

	Command or Action	Purpose

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

Step 2	**epbr service** *service-name* **Example:** `switch(config)# epbr service s1`	Enters to the configured service.
Step 3	[**no**] **shut** **Example:** `switch(config)# shut`	Shuts the endpoint to move it out of service The **no** form of the command shuts the node to bring an endpoint back into service.
Step 4	**service-endpoint** [**interface** *interface-name interface-number*] **Example:** `switch(config-epbr-svc)# service-end-point ip 1.1.1.1`	Configures service endpoint for the ePBR service. You can repeat steps 2 to 5 to configure another ePBR service.
Step 5	[**no**] **hold-down threshold count** *threshold count* **time** *threshold time* **Example:**	configures the threshold timers and failure counts at the endpoint level and/or the service level, with the endpoint level parameters overriding the service level parameters.

	Command or Action	Purpose
	`switch(config)# hold-down threshold count 2 time 5`	For threshold counts greater than 1, timer is mandatory. For threshold count of 1, timer is ignored or disallowed.

Configuring ePBR Set-VRF for an ePBR Policy

Beginning with Cisco NX-OS Release 10.5(2)F, ePBR will support the **set-vrf** command for ePBR L3 policies. This enhancement eliminates the need for route-leaking from host VRFs to service VRFs in ePBR inter-VRF deployments.

The **set-vrf** feature allows the traffic from last hop being routed in the host VRF context without route-leaking.

You can configure the **set-vrf** command either at the ePBR policy level or the match level. The match level takes the priority if both are configured.

To configure set-vrf, follow these steps:

Before you begin

- You must configure one dedicated port-channel interface, and one port-channel sub-interface per host VRF context before applying ePBR policy to an interface.

  The following example illustrates how to create port-channel and port-channel sub-interface for both source-vrf (vrf551) and destination vrf (vrf555):

```
int port-channel 1
   no shut
 int e1/1
   channel-group 1
   link loopback
   no shut
int port-channel 1.1
   encapsulation dot1q 10
   vrf member vrf551
   ip forward
   ipv6 address use-link-local-only
   ipv6 nd dad attempts 0
   ipv6 nd prefix default no-advertise
   ipv6 nd suppress-ra
   mtu 9216
   no shut
int port-channel 1.2
   encapsulation dot1q 11
   vrf member vrf555
   ip forward
   ipv6 address use-link-local-only
   ipv6 nd dad attempts 0
   ipv6 nd prefix default no-advertise
   ipv6 nd suppress-ra
   mtu 9216
   no shut
```

- You must also associate equivalent RPM configuration under the VRF context before applying an ePBR policy.

  The following example illustrates how to create VRF context configuration:

```
vrf context vrf551
    pbr set-vrf recirc interface port-channel1.1
```

```
vrf context vrf555
    pbr set-vrf recirc interface port-channel1.2
```

SUMMARY STEPS

1. **configure terminal**
2. **epbr policy** *policy-name-IPv4* | *policy-name-IPv6*
3. (Optional) **source-vrf** *source-vrf-name* **destination-vrf** *destination-vrf-name*
4. (Optional) **match** { [**ip address** *ipv4 acl-name*] | [**ipv6 address** *ipv6 acl-name*] } **source-vrf** *source-vrf-name* **destination-vrf** *destination-vrf-name*

DETAILED STEPS

Procedure

	Command or Action	Purpose

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

Step 2	**epbr policy** *policy-name-IPv4* | *policy-name-IPv6*
	Example:
	For IPV4:
	`switch(config)# epbr policy p_v4`
	`switch(config-epbr-policy)#`
	For IPV6:
	`switch(config-epbr-policy)# epbr policy p_v6`
		Configures an ePBR policy and enters ePBR policy configuration mode.

Step 3	(Optional) **source-vrf** *source-vrf-name* **destination-vrf** *destination-vrf-name* **Example:** `switch(config-epbr-policy)# source-vrf vrf551 destination-vrf vrf555`	Sets the `destination-vrf` for the forward direction and `source-vrf` for the reverse direction.

Step 4	(Optional) **match** { [**ip address** *ipv4 acl-name*] | [**ipv6 address** *ipv6 acl-name*] } **source-vrf** *source-vrf-name* **destination-vrf** *destination-vrf-name*
	Example:
	For IPv4:
	`switch(config-epbr-policy)# match ip address acl1 source-vrf vrf551 destination-vrf vrf555`
	For IPV6:
	`switch(config-epbr-policy)# match ipv6 address acl1 source-vrf vrf551 destination-vrf vrf555`
		Matches an IPv4 or IPv6 ACLs for the specified source and destination VRFs.

ePBR Show Commands

The following list provides the show commands associated with ePBR.

SUMMARY STEPS

1. **show epbr policy** *policy-name* [**reverse**]
2. **show epbr statistics** *policy-name* [**reverse**]
3. **show tech-support epbr**
4. **show running-config epbr**
5. **show startup-config epbr**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**show epbr policy** *policy-name* [**reverse**] **Example:** `switch# show epbr policy Tenant_A-Redirect`	Displays information on the ePBR policy applied in forward or reverse direction.
Step 2	**show epbr statistics** *policy-name* [**reverse**] **Example:** `switch# show ePBR statistics policy pol2`	Displays the ePBR policy statistics.
Step 3	**show tech-support epbr** **Example:** `switch# show tech-support epbr`	Displays the technical support information for ePBR.
Step 4	**show running-config epbr** **Example:** `switch# show running-config epbr`	Displays the running configuration for ePBR.
Step 5	**show startup-config epbr** **Example:** `switch# show startup-config epbr`	Displays the startup configuration for ePBR

Verifying ePBR Configuration

To verify the ePBR configuration, use the following commands:

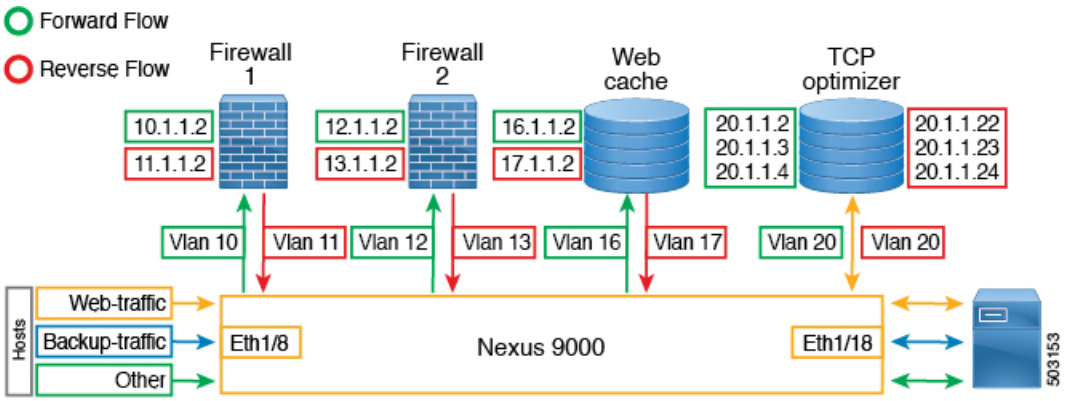
Command	Purpose
show ip/ipv6 policy vrf <context>	Displays the IPv4/IPv6 route-map policies created for the Layer-3 ePBR policy, at the interfaces at which service chain is applied and the relevant end-point interfaces of the service-chain.
show route-map dynamic <route-map name>	Displays the next-hops configured for traffic re-direction for specific bucket access-lists, used for forwarding traffic at every point in the service chain.
show ip/ipv6 access-list <access-list name> dynamic	Displays the traffic match criteria for a bucket access-list.
show ip sla configuration dynamic	Displays the IP SLA configuration generated by ePBR, for the service-end-points in the chain, when probes are enabled.
show track dynamic	Displays the tracks generated by ePBR, for the service-end-points in the chain, when probes are enabled.

Configuration Examples for ePBR L3

Example: ePBR NX-OS Configuration

The following topology illustrates ePBR NX-OS configuration.

Figure 1: ePBR NX-OS Configuration



Example: Use-Case: Create a Service Chain for Web Traffic in Forward Direction Only

The following configuration example shows how to create a service chain for web traffic in forward direction only.

```
IP access list web_traffic
        10 permit tcp any any eq www

ePBR service FW1
  service-end-point ip 10.1.1.2 interface Vlan10
```

```
    reverse interface Vlan11

ePBR service FW2
  service-end-point ip 12.1.1.2 interface Vlan12
    reverse interface Vlan13

ePBR service Web_cache
  service-end-point ip 16.1.1.2 interface Vlan16
    reverse interface Vlan17

ePBR policy tenant_1
  match ip address web-traffic
    10 set service FW1
    20 set service FW2
    30 set service Web_cache

interface Eth1/8
  ePBR ip policy tenant_1
```

The following example shows how to verify the configuration of service chain creation for web traffic in forward direction.

```
switch# show ePBR policy tenant_1

Policy-map : tenant_1
  Match clause:
    ip address (access-lists): web-traffic
  Service chain:
    service FW1, sequence 10, fail-action No fail-action
      IP 10.1.1.2
    service FW2, sequence 20, fail-action No fail-action
      IP 12.1.1.2
    service Web_cache, sequence 30, fail-action No fail-action
      IP 16.1.1.2
  Policy Interfaces:
    Eth1/8
```

Example: Use-Case : Load Balance TCP Traffic Using ePBR in Forward Direction Only

The following configuration example shows how to load balance TCP traffic using ePBR in forward direction only.

```
IP access list tcp_traffic
        10 permit tcp any any

ePBR service TCP_Optimizer
  service-interface Vlan20
  service-end-point ip 20.1.1.2
  service-end-point ip 20.1.1.3
  service-end-point ip 20.1.1.4

ePBR policy tenant_1
  match ip address tcp_traffic
    10 set service TCP_Optimizer

interface Eth1/8
  ePBR ip policy tenant_1
```

The following example shows how to verify the configuration of load balance TCP traffic using EPBR in forward direction.

```
switch# show ePBR policy tenant_1

Policy-map : tenant_1
  Match clause:
```

```
    ip address (access-lists): tcp_traffic
  Service chain:
    service TCP_Optimizer, sequence 10, fail-action No fail-action
      IP 20.1.1.2
      IP 20.1.1.3
      IP 20.1.1.4
  Policy Interfaces:
    Eth1/8
```

Example: Use-Case: Create a Service Chain for Web Traffic in Both Directions

The following configuration example shows how to create a service chain for web traffic in both forward and reverse directions.

```
IP access list web_traffic
        10 permit tcp any any eq www


ePBR service FW1
  service-end-point ip 10.1.1.2 interface Vlan10
    reverse ip 11.1.1.2 interface Vlan11

ePBR service FW2
  service-end-point ip 12.1.1.2 interface Vlan12
    reverse ip 13.1.1.2 interface Vlan13

ePBR service Web_cache
  service-end-point ip 16.1.1.2 interface Vlan16
    reverse ip 17.1.1.2 interface Vlan17

ePBR policy tenant_1
  match ip address web-traffic
    10 set service FW1
    20 set service FW2
    30 set service Web_cache

interface Eth1/8
  ePBR ip policy tenant_1

interface Eth1/18
  ePBR ip policy tenant_1 reverse
```

The following example shows how to verify the configuration of service chain creation for web traffic in both forward and reverse directions.

```
switch# show ePBR policy tenant_1

Policy-map : tenant_1
  Match clause:
    ip address (access-lists): web-traffic
  Service chain:
    service FW1, sequence 10, fail-action No fail-action
      IP 10.1.1.2
    service FW2, sequence 20, fail-action No fail-action
      IP 12.1.1.2
    service Web_cache, sequence 30, fail-action No fail-action
      IP 16.1.1.2
  Policy Interfaces:
    Eth1/8

switch# show ePBR policy tenant_1 reverse

Policy-map : tenant_1
  Match clause:
    ip address (access-lists): web-traffic
```

```
  Service chain:
    service Web_cache, sequence 30, fail-action No fail-action
      IP 17.1.1.2
    service FW2, sequence 20, fail-action No fail-action
      IP 13.1.1.2
    service FW1, sequence 10, fail-action No fail-action
      IP 11.1.1.2
  Policy Interfaces:
    Eth1/18
```

Example: Use-Case: Load Balance TCP Traffic Using ePBR in Both Directions

The following configuration example shows how to load balance TCP traffic using ePBR in both forward and reverse directions.

```
ePBR service TCP_Optimizer
  service-interface Vlan20
  service-end-point ip 20.1.1.2
    reverse ip 20.1.1.22
  service-end-point ip 20.1.1.3
    reverse ip 20.1.1.23
  service-end-point ip 20.1.1.4
    reverse ip 20.1.1.24

ePBR policy tenant_1
  match ip address tcp_traffic
    10 set service TCP_Optimizer

interface Eth1/8
  ePBR ip policy tenant_1

interface Eth1/18
  ePBR ip policy tenant_1 reverse
```

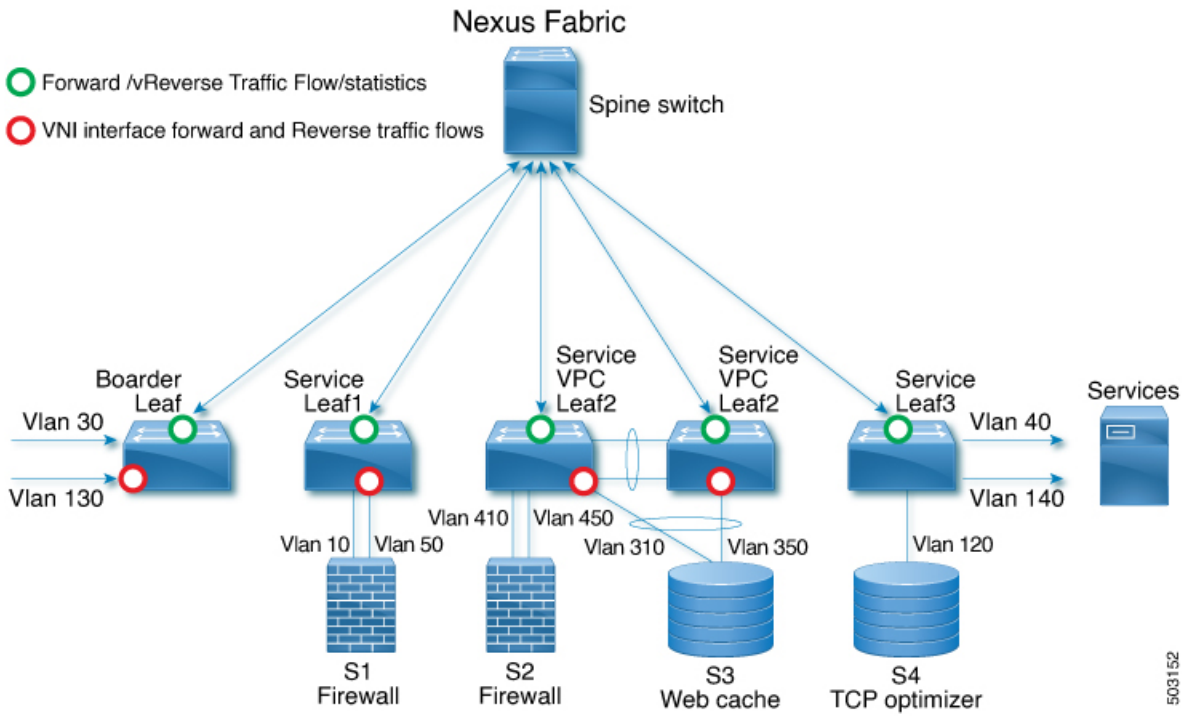
The following example shows how to verify the configuration of load balance TCP traffic using ePBR in both directions.

```
switch# show ePBR policy tenant_1

Policy-map : tenant_1
  Match clause:
    ip address (access-lists): tcp_traffic
  Service chain:
    service TCP_Optimizer, sequence 10, fail-action No fail-action
      IP 20.1.1.2
      IP 20.1.1.3
      IP 20.1.1.4
  Policy Interfaces:
    Eth1/8

switch# show ePBR policy tenant_1 reverse

Policy-map : tenant_1
  Match clause:
    ip address (access-lists): tcp_traffic
  Service chain:
    service TCP_Optimizer, sequence 10, fail-action No fail-action
      IP 20.1.1.22
      IP 20.1.1.23
      IP 20.1.1.24
  Policy Interfaces:
    Eth1/18
```

Example: ePBR Policy Creation with VXLAN Fabric

The following example/topology shows how to configure ePBR over VXLAN fabric.

Figure 2: Configuring ePBR over VXLAN Fabric




```
ip access-list acl1
        10 permit ip 30.1.1.0/25 40.1.1.0/25
        20 permit ip 30.1.1.128/25 40.1.1.128/25
ip access-list acl2
        10 permit ip 130.1.1.0/25 140.1.1.0/25
        20 permit ip 130.1.1.128/25 140.1.1.128/25

epbr service s1
  vrf vrf_s1
  service-end-point ip 10.1.1.2 interface Vlan10
     probe icmp frequency 4 retry-down-count 1 retry-up-count 1 timeout 2 source-interface
 loopback9
    reverse ip 50.1.1.2 interface Vlan50

      probe icmp frequency 4 retry-down-count 1 retry-up-count 1 timeout 2 source-interface
 loopback10

epbr service s2
  vrf vrf_s2
  service-end-point ip 41.1.1.2 interface Vlan410
     probe icmp source-interface loopback11
    reverse ip 45.1.1.2 interface Vlan450

       probe icmp source-interface loopback12

 epbr service s3
  vrf vrf_s3
  service-end-point ip 31.1.1.2 interface Vlan310
     probe http get index.html source-interface loopback13
    reverse ip 35.1.1.2 interface Vlan350
```

```
     probe http get index.html source-interface loopback14

epbr service s4
  service-interface Vlan120
  vrf vrf_s4
   probe udp 6900 control enable source-interface loopback15
  service-end-point ip 120.1.1.2

    reverse ip 120.1.1.2

epbr policy p1
 statistics
  match ip address acl1
    load-balance buckets 16 method src-ip
    10 set service s1 fail-action drop
    20 set service s2 fail-action drop
    30 set service s4 fail-action bypass
  match ip address acl2
    load-balance buckets 8 method dst-ip
    10 set service s1 fail-action drop
    20 set service s3 fail-action forward
    30 set service s4 fail-action bypass

! VXLAN L3 VNI interface for vrf_s1, vrf_s2, vrf_s3, vrf_s4 to which the policy is applied
 on all service leafs
interface vlan 100
epbr ip policy p1
epbr ip policy p1 reverse

interface vlan 101
epbr ip policy p1
epbr ip policy p1 reverse

interface vlan 102
epbr ip policy p1
epbr ip policy p1 reverse

interface vlan 103
epbr ip policy p1
epbr ip policy p1 reverse

Apply forward policy on ingress interface in border leaf where traffic coming in needs to
be service-chained:

interface Vlan 30 - Traffic matching acl1
  epbr ip policy p1
  int vlan 130  -  Traffic matching acl2
  epbr ip policy p1

Apply the reverse policy On leaf connected to server if reverse traffic flow needs to be
enabled:

int vlan 40 - Traffic matching reverse flow for acl1
epbr ip policy p1 rev
int vlan 140 - Traffic matching reverse flow for acl1
epbr ip policy p1 rev
```

Example: Configuring ePBR Service

The following example shows how to configure ePBR service.

```
epbr service FIREWALL
  probe icmp
  vrf TENANT_A
```

```
  service-endpoint ip 172.16.1.200 interface VLAN100
        reverse ip 172.16.2.200 interface VLAN101
service-endpoint ip 172.16.1.201 interface VLAN100
        reverse ip 172.16.2.201 interface VLAN101

epbr service TCP_Optimizer
   probe icmp
   vrf TENANT_A
service-endpoint ip 172.16.20.200 interface VLAN200
       reverse ip 172.16.30.200 interface VLAN201
```

Example: Configuring ePBR Policy

The following example shows how to configure ePBR Policy.

```
epbr service FIREWALL
  probe icmp
  service-end-point ip 1.1.1.1 interface Ethernet1/1
     reverse ip 1.1.1.2 interface Ethernet1/2
epbr service TCP_Optimizer
 probe icmp
 service-end-point ip 1.1.1.1 interface Ethernet1/3
     reverse ip 1.1.1.4 interface Ethernet1/4
epbr policy Tenant_A-Redirect
 match ip address WEB
  load-balance method src-ip
  10 set service FIREWALL fail-action drop
  20 set service TCP_Optimizer fail-action bypass
match ip address APP
  10 set service FIREWALL fail-action drop
match ip address exclude_acl exclude
match ip address drop_acl drop
```

The following example shows the output of show ePBR Policy command with fail-action drop information.

```
switch(config-if)# show epbr policy Tenant_A-Redirect

Policy-map : Tenant_A-Redirect
  Match clause:
    ip address (access-lists): WEB
action:Redirect
   service FIREWALL, sequence 10, fail-action Drop
     IP 1.1.1.1 track 1 [INACTIVE]
   service TCP_Optimizer, sequence 20, fail-action Bypass
     IP 1.1.1.1 track 2 [INACTIVE]
Match clause:
     ip address (access-lists): APP
action:Redirect
   service FIREWALL, sequence 10, fail-action Drop
     IP 1.1.1.1 track 1 [INACTIVE]
Match clause:
    ip address (access-lists): exclude_acl
action:Deny
Match clause:
   ip address (access-lists): drop_acl
action:Drop
 Policy Interfaces:
  Eth1/4
```

Example: Associating an Interface with ePBR Policy

The following example shows how to configure ePBR Policy.

```
interface vlan 2010
  epbr ip policy Tenant_A-Redirect

interface vlan 2011
  epbr ip policy Tenant_A-Redirect reverse
```

Example: ePBR Policy applied in forward direction

The following example shows the sample Output for policy applied in forward direction.

```
show epbr policy Tenant_A-Redirect
policy-map Tenant_A-Redirect
 Match clause:
  ip address (access-lists): WEB
 Service chain:
  service FIREWALL  , sequence 10 , fail-action drop
   ip 172.16.1.200 track 10  [ UP ]
   ip 172.16.1.201 track 11 [ DOWN ]
                    service TCP_Optimizer, sequence 20 , fail-action bypass
   ip 172.16.20.200 track 12  [ UP] ]

 Match clause:
  ip address (access-lists): APP
 Service chain:
  service FIREWALL  , sequence 10 , fail-action drop
   ip 172.16.1.200 track 10  [ UP ]
   ip 172.16.1.201 track 11 [ DOWN ]

 Policy Interfaces:
     Vlan 2010
```

Example: ePBR Policy applied in reverse direction

The following example shows the sample Output for policy applied in reverse direction.

```
show epbr policy Tenant_A-Redirect reverse
policy-map Tenant_A-Redirect
 Match clause:
  ip address (access-lists): WEB

 Service chain:
  service TCP_Optimizer, sequence 20 , fail-action bypass
   ip 172.16.30.200 track 15  [ UP] ]

  service FIREWALL  , sequence 10 , fail-action drop
   ip 172.16.2.200 track 13  [ UP ]
   ip 172.16.2.201 track 14 [ DOWN ]

 Match clause:
  ip address (access-lists): APP

 Service chain:

  service FIREWALL  , sequence 10 , fail-action drop
   ip 172.16.2.200 track 13  [ UP ]
   ip 172.16.2.201 track 14 [ DOWN ]

 Policy Interfaces:
     Vlan 2011
```

Example: User-defined Track

The following example shows to assign track ID to each end point.

```
epbr service FIREWALL
  probe icmp
```

```
  service-end-point ip 1.1.1.2 interface Ethernet1/21
probe track 30
reverse ip 1.1.1.3 interface Ethernet1/22
  probe track 40
 service-end-point ip 1.1.1.4 interface Ethernet1/23
    reverse ip 1.1.1.5 interface Ethernet1/24
```

Example: Modifying ePBR Service Using ePBR Session

The following example shows to replace the IP of ePBR service and add another service end point.

```
switch(config)#epbr session
switch(config-epbr-sess)#epbr service TCP_OPTIMIZER
switch(config-epbr-sess-svc)# no service-end-point ip 172.16.20.200 interface VLAN200
switch(config-epbr-sess-svc)#service-end-point ip 172.16.25.200 interface VLAN200
switch(config-epbr-sess-svc-ep)# reverse ip 172.16.30.200 interface VLAN201
switch(config-epbr-sess)#commit
```

Example: Modifying ePBR Policy Using EPBR Session

The following example shows to replace the IP of ePBR policy and add a service chain for the modified policy traffic.

```
switch(config)#epbr session
switch(config-epbr-sess)#epbr policy Tenant_A-Redirect
switch(config-epbr-sess-pol)# no match ip address WEB
switch(config-epbr-sess-pol)#match ip address WEB
switch(config-epbr-sess-pol-match)# 10  set service Web-FW fail-action drop load-balance
method src-ip
switch(config-epbr-sess-pol-match)# 20 set service TCP_Optimizer fail-action bypass
switch(config-epbr-sess-pol)#match ip address HR
switch(config-epbr-sess-pol-match)# 10  set service Web-FW
switch(config-epbr-sess-pol-match)# 20 set service TCP_Optimizer
switch(config-epbr-sess)#commit
```

Example: Displaying ePBR Statistics Policy

The following example shows the display of ePBR statistics policy.

```
switch# show epbr statistics policy pol2

Policy-map pol2, match testv6acl

    Bucket count: 2

      traffic match : epbr_pol2_1_fwd_bucket_1
        two : 0
      traffic match : epbr_pol2_1_fwd_bucket_2
        two : 0
```

Example: Displaying how mask-position is used

The following example shows the sample of how mask-position is used:

```
IP access list acl1
        10 permit tcp 10.0.0.0/24 any
epbr policy l3_Pol
  statistics match ip address acl1
  load-balance buckets 4 mask-position 5
10 set service s1_l3
switch# show ip access-list dynamic
IP access list epbr_l3_Pol_1_fwd_bucket_1
 10 permit tcp 10.0.0.0 0.0.0.159 any
IP access list epbr_l3_Pol_1_fwd_bucket_2
 10 permit tcp 10.0.0.32 0.0.0.159 any
IP access list epbr_l3_Pol_1_fwd_bucket_3
 10 permit tcp 10.0.0.64 0.0.0.159 any
```

```
IP access list epbr_l3_Pol_1_fwd_bucket_4
 10 permit tcp 10.0.0.96 0.0.0.159 any
```

Additional References

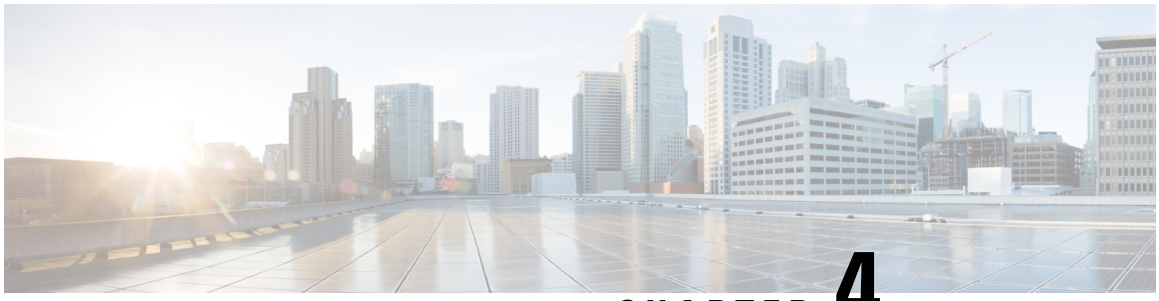
For additional information related to configuring ePBR, see the following sections:

Related Documents

Related Topic	Document Title
Configuring CoPP for IP SLA Packets	*Cisco Nexus 9000 Series NX-OS IP SLAs Configuration Guid 9.3(x)*
ePBR Licensing	*Cisco NX-OS Licensing Guide*
ePBR Scale Values	*Cisco Nexus 9000 Series NX-OS Verified Scalability Guide*

Standards

Standards
No new or modified standards are supported by this feature, and support for existing standards has not been modified by feature.



CHAPTER 4

Configuring ePBR L2

Information About ePBR L2

Enhanced Policy-based Redirect Layer2 (ePBR) in Elastic Services Re-direction (ESR) provides transparent service redirection and service chaining of Layer1/ Layer2 service appliances by leveraging Port ACL and VLAN translation. This action helps achieve service chaining and load-balancing capabilities without adding extra headers and avoids latency in using extra headers.

ePBR enables application-based routing and provides a flexible, device-agnostic policy-based redirect solution without impacting application performance. The ePBR service flow includes the following tasks:

Configuring ePBR Service and Policy

You must first create an ePBR service which defines the attributes of service end points. Service end points are the service appliances such as firewall, IPS, etc., that can be associated with switches. You can also define probes to monitor the health of the service end points and can define the forward and reverse interfaces where the traffic policies are applied. ePBR also supports load balancing along with service chaining. ePBR allows you to configure multiple service end points as a part of the service configuration.

After creating the ePBR service, you must create an ePBR policy. The ePBR policy allows you to define traffic selection, redirection of traffic to the service end point and various fail-action mechanisms on the end point health failure. You may use IP access-list end points with permit access control entries (ACE) to define the traffic of interest to match and take the appropriate action.

The ePBR policy supports multiple ACL match definitions. A match can have multiple services in a chain which can be sequenced by a sequence number. This allows flexibility to add, insert, and modify elements in

a chain in a single service policy. In every service sequence, you can define the fail action method such as drop, forward, and bypass. The ePBR policy allows you to specify source or destination-based load balancing and bucket counts in order to have granular load balancing of traffic.

Applying ePBR to an L2 Interface

After creating the ePBR policy you need to apply the policy on an interface. This allows you to define the interface at which the traffic ingresses the NX-OS switch and the interface through which traffic needs to exit the switch after redirection or service-chaining. You can also apply the policy in both the forward and reverse directions into the NX-OS switch.

Enabling Production Interfaces as Access Port

If the service-chaining switch is inserted in between the two L3 routers for traffic redirection, the production interfaces are enabled as access port with the following limitations:

- You must use the VLAN of the port as part of the match configuration.
- It is limited to mac-learn disable mode.

Enabling Production Interfaces as Trunk Ports

Production interfaces may be configured as trunk ports. The VLANs of the incoming traffic that needs to be service-chained that is trunked by the interfaces must be configured as part of the match configuration.

Alternatively, using 'vlan all' in the match configuration will allow any traffic pertaining to any incoming VLANs on the interface to be matched and service chained.

Creating Bucket and Load Balancing

ePBR computes the number of traffic buckets based on the service that has maximum number of service-end-points in the chain. If you configure the load balance buckets, your configuration will have the precedence. ePBR supports load balancing methods of source IP and destination IP but does not support L4-based source or destination load balancing methods.

ePBR Object Tracking, Health Monitoring, and Fail-Action

Layer-2 ePBR performs link state monitoring of the service end-points by default. The user may additionally enable CTP (Configuration Testing Protocol) if supported by the service.

You can configure the ePBR probe options for a service or for each of the forward or reverse end points. You can also configure frequency, timeout, and retry up and down counts. The same track objects is re-used for all policies using the same ePBR service.

If no probe method is defined at the end point level, the probe method configured for the service level will be used.

ePBR supports the following fail-action mechanisms for its service chain sequences:

- Bypass

- Drop on Fail
- Forward

Bypass of a service sequence indicates that the traffic must be redirected to the next service sequence when there is a failure of the current sequence.

Drop on fail of a service sequence indicates that the traffic must be dropped when all the service-end-points of the service become unreachable.

Forward is the default option and indicates that upon failure of the current service, traffic should forward to the egress interface. This is the default fail-action mechanism.

Note Symmetry is maintained when fail-action bypass is configured for all the services in the service chain. In other fail-action scenarios, when there are one or more failed services, symmetry is not maintained in the forward and the reverse flow.

Beginning with Cisco NX-OS Release 10.4(1)F, ePBR L2 fail-action feature is optimized to modify only the ACEs that are currently affected by the failure of the node. However, the fail-action optimization will be enabled only for those service-chains where the user configures **load-balance buckets** under the ePBR match statement.

The fail-action optimization is supported on Cisco Nexus 9300-EX/FX/FX2/FX3/GX/GX2, C9364C, C9332C, and Cisco Nexus 9500 switches with 9700-EX/FX/GX line cards.

ePBR Session-based Configuration

ePBR sessions allow addition, deletion or modification of the following aspects of in-service services or policies. The in-service refers to a service that is associated with a policy that has been applied to an active interface or a policy that is being modified and currently configured on an active interface.

- Service endpoints with their interfaces and probes
- Reverse endpoints and probes
- Matches under policies
- Load-balance methods for matches
- Match sequences and fail-action

Note In ePBR Sessions, you cannot move interfaces from one service to another service in the same session. To move interfaces from one service to another service, perform the following steps:

1. Use a session operation to first remove it from the existing service.
2. Use a second session operation to add it to the existing service.

ACL Refresh

ePBR session ACL refresh allows you to update the policy generated ACLs, when the user-provided ACL gets modified or added or deleted with ACEs. On the refresh trigger, ePBR will identify the policies that are impacted by this change and create or delete or modify the buckets' generated ACLs for those policies.

For ePBR scale values, see Cisco Nexus 9000 Series NX-OS Verified Scalability Guide.

Guidelines and Limitations for ePBR L2

ePBR has the following guidelines and limitations:

- When fail-action is specified in any match statement, probe is mandatory in the configuration.
- To disable MAC learning on the switch, use the command **mac-learn disable**.
- Do not share the same user defined ACL across multiple match statements in the ePBR configuration.
- Symmetry in traffic is maintained only when fail-action bypass is configured for ePBR Service. For the other fail-actions such as forward/drop in the service chain, symmetry is not maintained for the forward and reverse flow of traffic.
- Feature ePBR and feature ITD cannot co-exist with the same ingress interface.
- With scaled ePBR configuration, it is recommended to remove the policies before you use the **no feature epbr** command.
- ePBRv6 over VXLAN is not supported on Cisco Nexus 9500 series switches.
- If you want to remove the ePBR service endpoint which is configured to a port-channel that is removed from the system, perform the following steps:
  1. Delete the existing ePBR policy.
  2. Delete the existing ePBR service.
  3. Reconfigure the ePBR service endpoint to the required port-channel.
- Please do not modify the dynamically created access-list entries of ePBR that begin with the name "epbr_". These access-lists are reserved for ePBR internal use.

Note Modifying these prefix strings can cause the ePBR to not function properly and would impact ISSU.

- All redirection rules are programmed in ACL TCAM using ing-ifacl region. This region needs to be carved and allocated prior to the application of ePBR L2 policies.

Note For steps on how to carve TCAM region, refer to the **Configuring IP ACLs** section of *Cisco Nexus 9000 Series NX-OS Security Configuration Guide*.

- ePBR policies require at least one match with redirect action.
- ePBR L2 requires a VLAN range to be reserved for VLAN translation and Q-in-Q. It is recommended that this range does not overlap with the VLANs used for traffic match configuration.
- The ePBR 'infra' VLANs should be reserved prior to the application of the ePBR Layer-2 policies.
- For production interfaces configured as trunk ports, enable VLAN trunking only for the VLANs specified in the ePBR 'infra vlan' range.
- You must add the native VLAN to the list of trunk allowed VLANs, which aligns with available access features like selective QinQ and selective Q-in-VNI.
- ePBR L2 expects the service appliance to be configured to forward the packet as is without modifying or stripping the VLAN headers.
- Each match in an ePBR L2 policy needs to have a unique match VLAN or unique VLAN range when applied on trunk interfaces. Only a single match with 'vlan all' can exist in a policy that is applied on trunk interfaces.
- ePBR L2 policy definition can be applied to a maximum of 32 interfaces of supported interface types across forward and reverse directions.
- Beginning with Cisco NX-OS Release 10.3(1)F, multiple matches in the same EPBR L2 policy may share the same VLAN or VLAN range or may be configured with ‘vlan all’ in a policy that is applied on trunk interfaces.

Note Ensure that the ACL filters across the configured match ACLs are unique and do not overlap when multiple match ACLs of the same address family (IPv4, ipv6, or L2) share the same VLANs in a policy.

- For a production port pair, the policy that is applied on an interface in the forward direction and on its reverse interface in the reverse direction, should consist of matches, that are individually mapped to identical match-vlans or vlan ranges.
- In order to load-balance between multiple service devices and uniquely detect failure of these devices via CTP health-checks, each service device should be defined as a unique endpoint in the ePBR service.
- Bucket-based load-balance is not supported for layer-2 matches in the ePBR policy.
- In order to service-chain or redirect IPv6 traffic such as Neighbor discovery, ICMPv6 aces with protocol types of ND-NA and ND-NS should be explicitly defined in the user-defined match access-list.
- In order to service-chain or redirect Layer-2 traffic for protocols such as ARP (0x806), VN-tag (0x8926), FCOE (0x8906), MPLS Unicast (0x8847), MPLS Multicast (0x8848), the protocol information should be explicitly added to the ACEs inside the user-defined match access-list.
- Beginning with Cisco NX-OS Release 10.4(1)F, ePBR L2 supports redirection of all control traffic that matches the ePBR policy. For more information, see Enforcing Redirection and Drop for Control Traffic, on page 47 section.
- Defaulting ePBR production and/or service interfaces while they are in use should be avoided to prevent any unintended behavior.

- Beginning with Cisco NX-OS Release 10.3(1)F, ePBR L2 supports only redirection of L2 control packets on Cisco Nexus 9300-GX platform switches. Service-chaining is not supported on Cisco Nexus 9300-GX platform switches.
- Beginning with Cisco NX-OS Release 10.4(1)F, the ePBR provides **mask-position** option to choose the bits used for load-balancing in user-defined ACL for IPv4 or IPv6 matches on Cisco Nexus 9300-EX/FX/FX2/FX3/GX/GX2, and Cisco Nexus 9500 platform switches with Nexus 9700- EX/FX/GX line cards.
- Configuration rollback and configuration replace are supported only when the ePBR policy is not associated with any interfaces and the ePBR service definitions are not used in any active ePBR policy in both the source and target configurations. However, configuration rollback and configuration replace do not support policy to interface association and disassociation.

The following guidelines and limitations apply to the match ACL feature:

- Only ACEs with the permit method are supported in the ACL. ACEs with any other method (such as deny or remark) are ignored.
- A maximum of 256 permit ACEs are supported in one ACL.
- Beginning with Cisco NX-OS Release 10.4(1)F, the Layer-4 port ranges and other port operations (such as 'not equal to', 'greater than', 'lesser than') in the match access-list rules will be honored and used for filtering traffic in the bucket access-lists.
- The configuration **hardware access-list lou resource threshold** must be used for optimal utilization of TCAM ACEs, while using layer-4 port operators in access-lists. For more information on the command, see **Configuring IP ACLs** section of *Cisco Nexus 9000 Series NX-OS Security Configuration Guide*.

The following guidelines and limitations apply to inter-VRF service chaining:

- Beginning with Cisco NX-OS Release 10.2(3)F, to minimize traffic disruptions during session operations of endpoint additions, service sequence additions, deletions, and modifications, it is recommended to have load-balance buckets configured ahead and avoid modification to the load-balance configuration. Ensure that the configured buckets for load-balance are greater than the number of endpoints configured in services for every sequence in the chain.

The following guidelines and limitations applies if you have configured ePBR using source IP-based load balancing:

- The prefix length in the source IPv4 of the ACE cannot be /32
- The prefix length in the source IPv6 address of the ACE cannot be /128
- The subnet for the source address must be compatible with the buckets configured.

The following guidelines and limitations applies if you have configured ePBR using destination IP-based load balancing:

- The prefix length in the destination IPv4 of the ACE cannot be /32
- The prefix length in the destination IPv6 address of the ACE cannot be /128
- The subnet for the destination address must be compatible with the buckets configured.

Configuring ePBR Service, Policy, and Associating to an Interface

The following section provides information about configuring the ePBR Service, ePBR Policy, and associating the policy on to an interface.

SUMMARY STEPS

1. **configure terminal**
2. **[no] epbr infra vlans** *[vlan range]*
3. **epbr service** *service-name* **type** *l2*
4. **mode** [**full duplex** | **half duplex**]
5. **probe** {**ctp**} [**frequency** *seconds*] [**timeout** *seconds*] [**retry-down-count** *count*] **retry-up-count** *count*]
6. **service-endpoint** [**interface** *interface-name interface-number*]
7. **reverse interface** *interface-name interface-number*
8. **exit**
9. **epbr policy** *policy-name*
10. **match** { [**ip address** *ipv4 acl-name*] | [**ipv6 address** *ipv6 acl-name*] | [**l2 address** *l2 acl-name*]} {**drop** | **exclude** | **redirect** | **vlan**{**vlan** | **vlan range** | **all**}}
11. [**no**] **load-balance** [**method** { **src-ip** | **dst-ip**}] [**buckets** *count*] [**mask-position** *position-value*]
12. *sequence-number* **set service** *service-name* [**fail-action** { **bypass** | **drop** | **forward**}]
13. **interface** *interface-name interface-number*
14. **epbr** {**l2**} **policy** *policy-name egress-interface interface-name* [**reverse**]
15. **exit**

DETAILED STEPS

Procedure

	Command or Action	Purpose

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

Step 2	**[no] epbr infra vlans** *[vlan range]*	VLAN range is using to indicate the VLANs reserved for selective dot1q translation while redirecting to the service devices.
Step 3	**epbr service** *service-name* **type** *l2* **Example:** `switch(config)# epbr service firewall type l2`	Creates a new ePBR L2 service.
Step 4	**mode** [**full duplex**	**half duplex**]	Configures the service to be in half-duplex or full-duplex mode.

	Command or Action	Purpose
Step 5	**probe** {**ctp**} [**frequency** *seconds*] [**timeout** *seconds*] [**retry-down-count** *count*] **retry-up-count** *count*] **Example:** `switch(config)# probe icmp`	Configures the probe for the ePBR service. The options are as follows: • frequency—Specifies the frequency of the probe in seconds. The range is from 1 to 604800. • retry-down-count —Specifies the number of recounts undertaken by the probe when the node goes down. The range is from 1 to 5. • retry-up-count —Specifies the number of recounts undertaken by the probe when the node comes back up. The range is from 1 to 5. • timeout —Specifies the length of the timeout period in seconds. The range is from 1 to 604800.
Step 6	**service-endpoint** [**interface** *interface-name interface-number*] **Example:** `switch(config-epbr-svc)# service-end-point interface Ethernet1/3`	Configures service endpoint for the ePBR service. You can repeat steps 2 to 5 to configure another ePBR service.
Step 7	**reverse interface** *interface-name interface-number* **Example:** `switch(config-epbr-fwd-svc)# reverse interface Ethernet1/4`	Defines the reverse interface where the traffic policies are applied.

Step 8	**exit**
	Example:
	`switch(config-epbr-reverse-svc)# exit`
	`switch(config-epbr-fwd-svc)# exit`
	`switch(config-epbr-svc)# exit`
	`switch(config)#`
		Exits ePBR service configuration mode and enters global configuration mode.

Step 9	**epbr policy** *policy-name* **Example:** `switch(config)# epbr policy Tenant_A-Redirect`	Configures the ePBR policy.
Step 10	**match** { [**ip address** *ipv4 acl-name*] | [**ipv6 address** *ipv6 acl-name*] | [**l2 address** *l2 acl-name*]} {**drop** | **exclude** | **redirect** | **vlan**{**vlan** | **vlan range** | **all**}} **Example:** `switch (config) # match ip address WEB vlan 10`	Matches an IPv4, or IPv6 address, or a mac address against an IP, or IPv6, or MAC ACLs. Redirect is the default action for a match traffic. Drop is used when the traffic needs to be dropped on the incoming interface. Exclude option is used to exclude certain traffic from service-chaining on the incoming interface. You can repeat this step to match multiple ACLs based on the requirement.

	Command or Action	Purpose
Step 11	[**no**] **load-balance** [**method** { **src-ip** | **dst-ip**}] [**buckets** *count*] [**mask-position** *position-value*] **Example:** `switch(config)# load-balance method src-ip mask-position 3`	Computes the load balance method and the number of buckets to be used by the ePBR service. Beginning with Cisco NX-OS Release 10.4(1)F, the **mask-position** option is provided to choose the bits used for load-balancing in user-defined ACL for IPv4 or IPv6 matches. Default value is 0. If mask-position is configured, the load-balance bits start from configured mask-position. Based on number of buckets needed, more bits are taken to generate load-balancing buckets, toward the bits direction depending on whether the least significant bit or most significant bit is selected. **Note** For any ACE in user-defined ACLs, if bits used to generate load-balancing buckets overlap with the user-defined subnet, the mask position for the ACE will be reset internally to 0.
Step 12	*sequence-number* **set service** *service-name* [**fail-action** { **bypass** | **drop** | **forward**}] **Example:** `switch(config)# set service firewall fail-action drop`	Configures the fail-action mechanism.
Step 13	**interface** *interface-name interface-number* **Example:** `switch(config)# interface Ethernet1/1`	Enters into interface configuration mode.
Step 14	**epbr** {**l2**} **policy** *policy-name egress-interface interface-name* [**reverse**] **Example:** `epbr l2 policy Tenant_A_Redirect egress-interface Ethernet1/2`	An interface may be associated at any time with one forward policy and one reverse policy of the following: • an IPV4 policy in the forward direction • an IPv4 policy in the reverse direction • an IPv6 policy in the forward direction • an IPv6 policy in the reverse direction • a l2 policy in the forward direction • a l2 policy in the reverse direction
Step 15	**exit** **Example:** `switch(config-if)# end`	Exits policy configuration mode and returns to global mode.

Modifying a Service Using ePBR Session

The following steps explain how to modify a service using ePBR session.

SUMMARY STEPS

1. **epbr session**
2. **epbr service** *service-name type l2*
3. [**no**] **service-endpoint** [**interface** *interface-name*]
4. **service-endpoint** [**interface** *interface-name*]
5. **reverse** [**interface** *interface-name*]
6. **commit**
7. **abort**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**epbr session** **Example:** `switch(config)# epbr session`	Enters ePBR session mode.
Step 2	**epbr service** *service-name type l2* **Example:** `switch(config-epbr-sess)# epbr service TCP_OPTIMIZER`	Specifies the configured ePBR service in the ePBR session mode.
Step 3	[**no**] **service-endpoint** [**interface** *interface-name*] **Example:** `switch(config-epbr-sess-svc)# no service-end-point interface ethernet 1/3`	Disables the configured service endpoint for the ePBR service.
Step 4	**service-endpoint** [**interface** *interface-name*] **Example:** `switch(config-epbr-sess-svc)# service-end-point interface ethernet 1/15`	Add a service endpoint to the service.
Step 5	**reverse** [**interface** *interface-name*] **Example:** `switch(config-epbr-sess-fwd-svc)# reverse interface ethernet 1/4`	Defines the reverse interfaces where the traffic policies are applied.
Step 6	**commit** **Example:** `switch(config-epbr-sess)#commit`	Completes the modification of the ePBR service using the ePBR session. **Note**

	Command or Action	Purpose
		Restart the ePBR session after you complete this step.
Step 7	**abort** **Example:** `switch(config-epbr-sess)# abort`	Aborts the session and clears or resets the current configuration under the session. Use this command to abandon the current session configuration in case of errors or unsupported configuration identified during commits. **Note** Restart a new ePBR session after this with the rectified configuration.

Modifying a Policy Using ePBR Session

The following steps explain how to modify a policy using ePBR Session.

SUMMARY STEPS

1. **epbr session**
2. **epbr policy** *policy-name*
3. [**no**] **match** { [**ip address** *ipv4 acl-name*] | [**ipv6 address** *ipv6 acl-name*] | **l2 address** *mac acl-name*]} **vlan** {**all** | **vlan-id** | **vlan-id-range**
4. **match** { [**ip address** *ipv4 acl-name*] | [**ipv6 address** *ipv6 acl-name*] | **l2 address** *mac acl-name*]} **vlan** {**all** | **vlan-id** | **vlan-id-range**]
5. *sequence-number* **set service** *service-name* [**fail-action** { **bypass** | **drop** | **forward**}]
6. [**no**] **load-balance** [**method** { **src-ip** | **dst-ip**}] [**buckets** *count*] [**mask-position** *position-value*]
7. **commit**
8. **end**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**epbr session**	
Step 2	**epbr policy** *policy-name* **Example:** `switch(config-epbr-sess)# epbr policy Tenant_A-Redirect`	Specifies the configured ePBR policy in the ePBR session mode.
Step 3	[**no**] **match** { [**ip address** *ipv4 acl-name*]	[**ipv6 address** *ipv6 acl-name*]	**l2 address** *mac acl-name*]} **vlan** {**all**	**vlan-id**	**vlan-id-range** **Example:**	Disables the match against IP, IPv6, or L2 ACLs.

	Command or Action	Purpose
	`switch(config-epbr-sess-pol)# no match ip address WEB`	
Step 4	**match** { [**ip address** *ipv4 acl-name*] | [**ipv6 address** *ipv6 acl-name*] | **l2 address** *mac acl-name*]} **vlan** {**all** | **vlan-id** | **vlan-id-range**] **Example:** `switch(config-epbr-sess-pol)# match ip address HR`	Modifies the match against the IP, IPv6 or L2 ACLs.
Step 5	*sequence-number* **set service** *service-name* [**fail-action** { **bypass** | **drop** | **forward**}] **Example:** `switch(config-epbr-sess-pol-match)# set service firewall fail-action drop`	Configures the fail-action mechanism.
Step 6	[**no**] **load-balance** [**method** { **src-ip** | **dst-ip**}] [**buckets** *count*] [**mask-position** *position-value*] **Example:** `switch(config)# load-balance method src-ip mask-position 3`	Configures the load-balance method and buckets for the match. **Note** On omitting this configuration in the session context while modifying the service-chain for an existing match, the load-balance configuration for the match will be reset to default. Beginning with Cisco NX-OS Release 10.4(1)F, the **mask-position** option is provided to choose the bits used for load-balancing in user-defined ACL for IPv4 or IPv6 matches. Default value is 0. If mask-position is configured, the load-balance bits start from configured mask-position. Based on number of buckets needed, more bits are taken to generate load-balancing buckets, toward the bits direction depending on whether the least significant bit or most significant bit is selected. **Note** For any ACE in user-defined ACLs, if bits used to generate load-balancing buckets overlap with the user-defined subnet, the mask position for the ACE will be reset internally to 0.
Step 7	**commit** **Example:** `switch(config-epbr-sess)#commit`	Completes the modification of the ePBR policy using the ePBR session.
Step 8	**end** **Example:** `switch(config-epbr-sess)#end`	Exits the ePBR session mode.

Updating the Access-list Used by ePBR Policies

The following steps explain how to update the access-list used by ePBR policies:

SUMMARY STEPS

1. **epbr session access-list** *acl-name* **refresh**
2. **end**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**epbr session access-list** *acl-name* **refresh** **Example:** `switch(config)# epbr session access-list WEB refresh`	Updates or refreshes the policy generated ACLs.
Step 2	**end** **Example:** `switch(config)# end`	Exits the global configuration mode.

Enforcing Redirection and Drop for Control Traffic

Beginning with Cisco NX-OS Release 10.4(1)F, the following configuration options may be used to control redirection and drop behavior for control traffic through an ePBR L2 policy.

The **all** configuration option is used inside the ACEs in the user-defined match access-list for ePBR, in order to indicate that the highest priority is needed for an ACE . See **Applying an IP ACL Rule Prioritization over SUP Rule** or **Applying a MAC ACL Rule Prioritization over SUP Rule** section of *Cisco Nexus 9000 Series NX-OS Security Configuration Guide* for details on this configuration.

If the **all** option is used, the following behavior is observed:

- For matches with **redirection** or **exclude** action, ePBR generates corresponding redirection ACEs to enforce redirection of all matching traffic, including control traffic toward the specified service devices or the egress interface respectively.
- For matches with **drop** action, ePBR generates deny ACEs to enforce the drop of all matching traffic, including control traffic. If this option is not detected as configured, any control traffic that is typically copied or redirected to the supervisor on Cisco NX-OS 9000 series switches, may continue to do so, even if it matches the ePBR Layer-2 policy definition.

The **all** option has no effect if the match access-lists are used inside ePBR Layer-3 policies.

The **default-traffic-action redirect-all** configuration option is used inside an ePBR Layer-2 policy to specify that any traffic that does not match redirect, exclude, or drop matches, including control traffic must be

redirected toward the specified egress interface. If this option is not configured, any control traffic that does not match the access-lists inside the policy, and which is typically copied or redirected to the supervisor on Cisco NX-OS 9000 series switches, may continue to do so, instead of redirecting to the egress interface.

You can configure the default catch-all traffic behavior at a policy level using the following commands.

SUMMARY STEPS

1. **configure terminal**
2. **epbr policy** *policy-name*
3. **default-traffic-action** [**redirect** | **redirect-all**]

DETAILED STEPS

Procedure

	Command or Action	Purpose

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

Step 2	**epbr policy** *policy-name* **Example:** `switch(config)# epbr policy p3`	Configures the ePBR policy.
Step 3	**default-traffic-action** [**redirect**	**redirect-all**] **Example:** `switch(config-epbr-policy)# default-traffic-action redirect-all`	Sets the default catch-all behavior for an ePBR policy. • **redirect**: Redirects the data traffic. **redirect** is the default option. • **redirect-all**: Redirects all traffic. **Note** • This option is not supported inside Layer-3 ePBR policies. • This option cannot be modified inside ePBR sessions and requires the policy to be disabled, re-configured, and applied back.

ePBR Show Commands

The following list provides the show commands associated with ePBR.

SUMMARY STEPS

1. **show epbr policy** *policy-name* [**reverse**]

2. **show epbr statistics** *policy-name* [**reverse**]
3. **show tech-support epbr**
4. **show running-config epbr**
5. **show startup-config epbr**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**show epbr policy** *policy-name* [**reverse**] **Example:** `switch# show epbr policy Tenant_A-Redirect`	Displays information on the ePBR policy applied in forward or reverse direction.
Step 2	**show epbr statistics** *policy-name* [**reverse**] **Example:** `switch# show ePBR statistics policy pol2`	Displays the ePBR policy statistics.
Step 3	**show tech-support epbr** **Example:** `switch# show tech-support epbr`	Displays the technical support information for ePBR.
Step 4	**show running-config epbr** **Example:** `switch# show running-config epbr`	Displays the running configuration for ePBR.
Step 5	**show startup-config epbr** **Example:** `switch# show startup-config epbr`	Displays the startup configuration for ePBR

Verifying ePBR Configuration

To verify the ePBR configuration, use the following commands:

Command	Purpose
show ip access-list <access-list name> dynamic	Displays the traffic match criteria for a bucket access-list.
show ip sla configuration dynamic	Displays the IP SLA configuration generated by ePBR, for the service-end-points in the chain, when probes are enabled.
show track dynamic	Displays the tracks generated by ePBR, for the service-end-points in the chain, when probes are enabled.

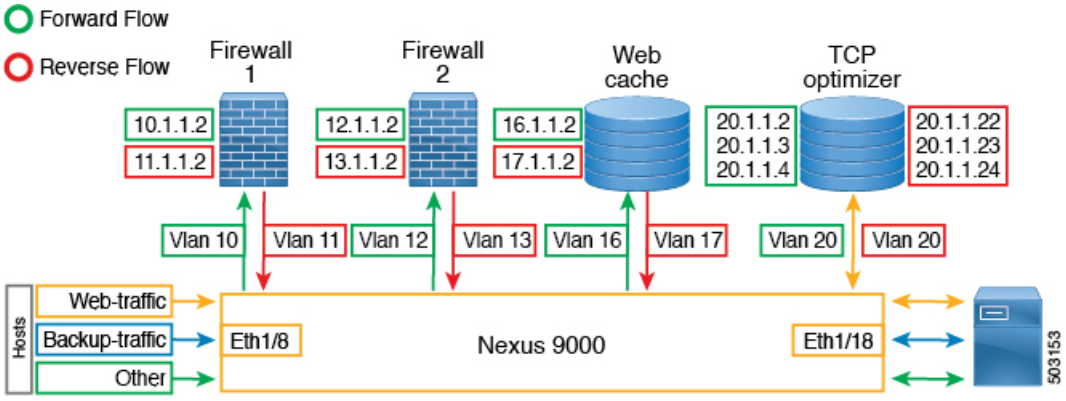
Command	Purpose
show ip access-list summary	Displays the summary of the traffic match criteria for a bucket access-list.
show [ip | ipv6 | mac] access-lists dynamic	Displays the dynamic entries of match criteria.

Configuration Examples for ePBR

Example: ePBR NX-OS Configuration

The following topology illustrates ePBR NX-OS configuration:

Figure 3: ePBR NX-OS Configuration



Example: Service Configuration for Access and Trunk Ports

The following configuration example shows how to perform service configuration for access and trunk ports:

```
epbr infra vlans 100-200

epbr service app_1 type l2
   service-end-point interface Ethernet1/3
    reverse  interface Ethernet1/4

epbr service app_2 type l2
   probe ctp frequency 2 retry-down-count 1 retry-up-count 1 timeout 1
   service-end-point interface port-channel10
    reverse  interface port-channel11

epbr service app_3 type l2
   probe ctp frequency 2 retry-down-count 1 retry-up-count 1 timeout 1
   service-end-point interface Ethernet1/9
    reverse  interface Ethernet1/10

epbr service app_4 type l2
   probe ctp frequency 2 retry-down-count 1 retry-up-count 1 timeout 1
   service-end-point interface port-channel12
    reverse  interface port-channel13
```

Example: Configuring Access Ports

The following example shows how to configure access ports:

```
epbr policy p1
  statistics
  match ipv6 address flow2 vlan 10
    load-balance buckets 2
    10 set service app_1
    20 set service app_3
    25 set service app_4
    30 set service app_2
  match l2 address flow3 vlan 10
    20 set service app_2
    25 set service app_4
    50 set service app_3
  match ip address flow1 vlan 10
    10 set service app_1
    15 set service app_3
    20 set service app_2

interface Ethernet1/1
  switchport
  switchport access vlan 10
  no shutdown
  epbr l2 policy p1 egress-interface Ethernet1/2

interface Ethernet1/2
  switchport
  switchport access vlan 10
  no shutdown
  epbr l2 policy p1 egress-interface Ethernet1/1 reverse
```

Example: Configuring Trunk Ports

The following configuration example shows how to configure trunk ports:

```
epbr policy p3
  statistics
  match ip address flow1 vlan 10
    load-balance buckets 2
    10 set service app_1
    20 set service app_2
  match ipv6 address flow2 vlan 20
    load-balance buckets 2
    10 set service app_3
    20 set service app_4
  match l2 address flow3 vlan 30
    10 set service app_1
    20 set service app_2

interface Ethernet1/27
  switchport
  switchport mode trunk
  no shutdown
  epbr l2 policy p3 egress-interface Ethernet1/28

interface Ethernet1/28
  switchport
  switchport mode trunk
  no shutdown
  epbr l2 policy p3 egress-interface Ethernet1/27 reverse

Collecting statistics
```

Collecting statistics:

```
itd-san-2# show epbr statistics policy p1

Policy-map p1, match flow2

    Bucket count: 2

      traffic match : bucket 1
        app_1 : 8986 (Redirect)
        app_3 : 8679 (Redirect)
        app_4 : 8710 (Redirect)
        app_2 : 8725 (Redirect)
      traffic match : bucket 2
        app_1 : 8696 (Redirect)
        app_3 : 8680 (Redirect)
        app_4 : 8711 (Redirect)
        app_2 : 8725 (Redirect)

Policy-map p1, match flow3

    Bucket count: 1

      traffic match : bucket 1
        app_2 : 17401 (Redirect)
        app_4 : 17489 (Redirect)
        app_3 : 17461 (Redirect)

Policy-map p1, match flow1

    Bucket count: 1

      traffic match : bucket 1
        app_1 : 17382 (Redirect)
        app_3 : 17348 (Redirect)
        app_2 : 17411 (Redirect)
```

Example: Viewing ePBR Policy

The following example shows how to view an ePBR policy:

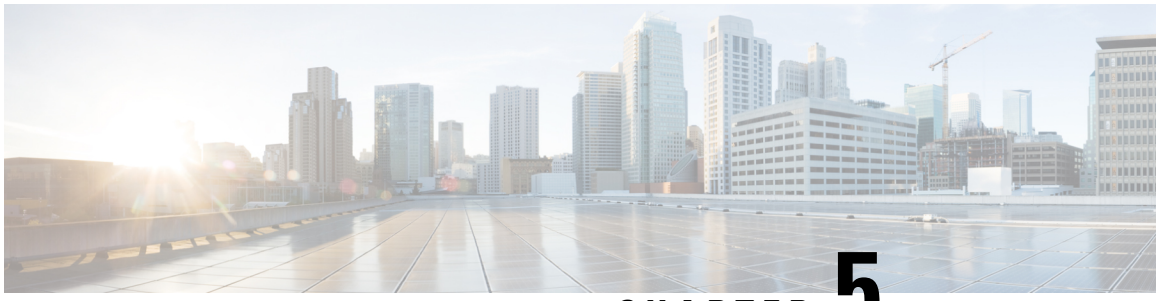
```
show epbr policy p3

Policy-map : p3
Match clause:
ip address (access-lists): flow1
action:Redirect
service app_1, sequence 10, fail-action No fail-action
Ethernet1/3 track 4 [UP]
service app_2, sequence 20, fail-action No fail-action
port-channel10 track 10 [UP]
Match clause:
ipv6 address (access-lists): flow2
action:Redirect
service app_3, sequence 10, fail-action No fail-action
Ethernet1/9 track 13 [UP]
service app_4, sequence 20, fail-action No fail-action
port-channel12 track 3 [UP]
Match clause:
layer-2 address (access-lists): flow3
action:Redirect
service app_1, sequence 10, fail-action No fail-action
Ethernet1/3 track 4 [UP]
service app_2, sequence 20, fail-action No fail-action
port-channel10 track 10 [UP]
```

```
Policy Interfaces:
egress-interface Eth1/28
```

Example: Displaying how mask-position is used

The following example shows the sample of how mask-position is used:

```
ip access-list acl1
   10 permit tcp 10.1.1.0/24 any
epbr service s1_l2 type l2
  service-end-point interface Ethernet1/2
    reverse interface Ethernet1/3
epbr policy l2_pol
   statistics
   match ip address acl1 vlan all
   load-balance buckets 4 mask-position 5
   10 set service s1_l2
interface Ethernet1/18
  epbr l2 policy l2_pol egress-interface Ethernet1/19
switch(config-if)# show access-lists epbr_Ethernet1_18_ip dyn

IP access list epbr_Ethernet1_18_ip
        statistics per-entry
        200001 permit tcp 10.1.1.0 0.0.0.159 any vlan 100 redirect Ethernet1/2 [
match=0]
        200002 permit tcp 10.1.1.32 0.0.0.159 any vlan 100 redirect Ethernet1/2
[match=0]
        200003 permit tcp 10.1.1.64 0.0.0.159 any vlan 100 redirect Ethernet1/2
[match=0]
        200004 permit tcp 10.1.1.96 0.0.0.159 any vlan 100 redirect Ethernet1/2
[match=0]
        4294967295 permit ip any any redirect Ethernet1/19 [match=0]
```



CHAPTER 5

Configuring Service Chaining with Security Groups

Information About ePBR and Group Policy Option

Beginning with Cisco NX-OS Release 10.5(1)F, users can redirect traffic flows between endpoints part of different Security-Groups. The redirection can happen through a single service function (as a firewall or a load-balancer) or through a chain of service functions. Beginning with Cisco NX-OS Release 10.5(2)F, users can include up to 5 service functions in a service chain. A given service function is built with one or more endpoints, representing the service devices performing such function. Traffic flows can be load-balanced across these service endpoints, while ensuring that both directions of traffic flow symmetrically use the same service endpoint. The onboarding of these service-devices, health monitoring mechanisms, and the user intent of chaining and load-balancing the traffic based on the properties of these service devices is captured and enforced through ePBR. To know more about micro-segmentation configuration, See Micro-segmentation for VXLAN Fabrics Using Group Policy Option (GPO).

ePBR Service and Service-chain

You must first create a service function, which is defined with one or more endpoints with their specific attributes. Service endpoints are the service appliances such as firewall, IPS, and so on, that are available in the network to which traffic needs to be redirected. You can also define probes to monitor the health of the service endpoints. ePBR also supports load balancing along with service-chaining. ePBR allows you to

configure multiple service endpoints as a part of a specific service function and would load-balance traffic among these endpoints, always ensuring that the two legs of the same traffic flow are using the same service endpoint. This is required when the different service endpoints defined for a given service function are not clustered and hence do not share connection states between them.

You must specify the VRF context for the service as the context in which the service endpoints are reachable.

After creating one (or more) ePBR services, you must create an ePBR service-chain. The ePBR service-chain allows you to define the service function (or the chain of service functions) through which traffic should be redirected along with the order in which this needs to be done.

Services used in a chain are identified by a sequence number. In NXOS 10.5(1)F, only a single service function may be specified inside a service-chain, thereby supporting only redirection and load-balancing capabilities to a single service functions before traffic is permitted to its destination.

In every service sequence, you can define the fail-action method such as drop, forward, and bypass indicating the action that needs to be taken in the event of failures of all endpoints in the service. If no fail-action is configured, the default behavior is to drop the traffic when the service is considered as failed.

The ePBR service-chain also allows you to specify the manner in which traffic needs to be load-balanced amongst the endpoints inside a service.

Beginning with Cisco NX-OS Release 10.5(2)F, ePBR multi-node service-chains are supported with Group Policy Options. A maximum of 5 service functions (nodes) can be configured in a service chain. Multi node service chain can contain firewall, load balancer, NAT, IPS and other devices.

Beginning with Cisco NX-OS Release 10.5(2)F, sources and destinations for the contracts using ePBR single node or multi node service-chains may be distributed across multiple sites using VXLAN Group Policy Options.

Beginning with Cisco NX-OS Release 10.5(2)F,ePBR multi-node service-chains and multi-site features are supported with sources and destinations in different VRF contexts. Service devices can belong to source VRF, destination VRF, or any other VRF.

Security Group for Service

You must configure security-group identifiers, as services can also be deployed in one arm mode of ePBR services in order to use the service for VxLAN GPO based redirection and chaining. This configuration is required to correctly steer the traffic to the service devices and through the chain.

These security-groups must be defined in the system as selector of type layer4-7. Each of the connected interfaces for the service endpoints inside the service must be mapped to the correct security-group as match interface selectors. For more details, see *Creating a Security Group on* Micro-segmentation for VXLAN Fabrics Using Group Policy Option (GPO).

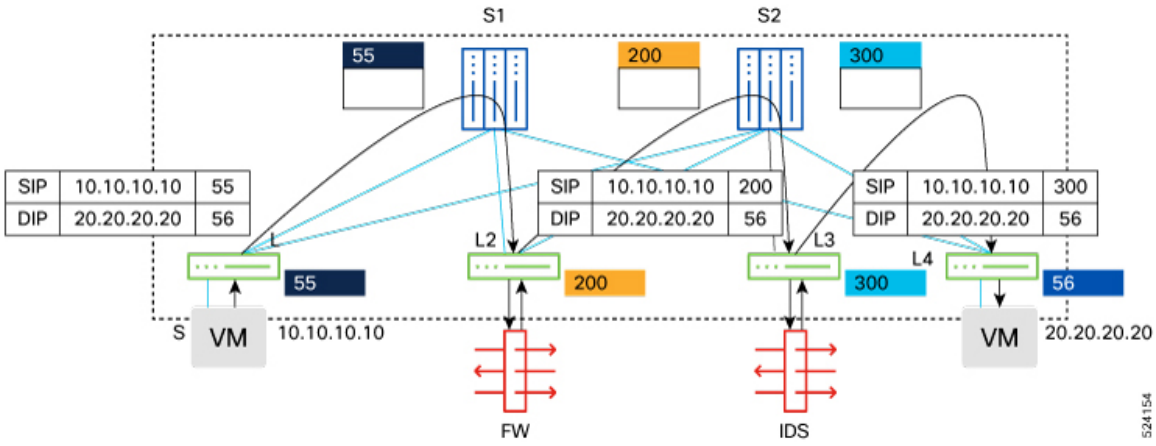
Connected interfaces for all the forward arms of the service endpoints must be mapped to the same identifier that is specified as the forward security-group for the ePBR service.

Connected interfaces for all the reverse arms of the service endpoints must be mapped to the same identifier that is specified as the reverse security-group for the ePBR service.

Only one security-group identifier should be configured for ePBR services with one-arm endpoints.

Two unique forward and reverse security-group identifiers should be configured for ePBR services with dual-arm endpoints. See figure 1 for a topology that explains the micro-segmentation based redirection and chaining.

Figure 4: Micro-segmentation with Service Chaining



Using ePBR Service-chains with SGACL Policies and Contracts

ePBR service-chain with GPO can provide traffic redirection using GPO policies and contracts. Service-chain can be enabled for security contracts by attaching it to match class-maps inside policies used by contracts. For more details about the configuration, see Micro-segmentation for VXLAN Fabrics Using Group Policy Option (GPO).

ePBR Health Monitoring and Fail-action

ePBR monitors the health of the endpoints by provisioning IP SLA probes and object tracks to track the IP SLA reachability when you apply the probe configuration.

ePBR supports various probes for protocols such as ICMP, TCP, UDP, DNS, and HTTP. ePBR also supports user defined tracks, which allows you to create tracks with various parameters including millisecond probes and associate them with ePBR endpoints.

You can configure ePBR probe options for a service if all the endpoints of the service require similar probing methods and protocols. If one or more endpoints require a different probing mechanism, you can configure probe options specific to those forward and reverse endpoints. You can also configure frequency, timeout, retry up and down counts. For service-endpoints distributed in a VXLAN environment, users must configure source loopback interfaces for the endpoints or service probes. The IP Addresses of these loopback interfaces are used as the unique source IP for the IP SLA sessions established with these endpoints.

When probes are configured for the service, forward and reverse arms do not need to have a unique loopback. They can share the same loopback or a different loopback can be provided.

You can define tracks separately and assign the track ID to the forward and reverse arm of each service-endpoint in ePBR. These track IDs should not be re-used across different endpoints in the same or different ePBR service but may be shared between the forward and reverse arms of the endpoint. If you do not assign any user-defined track to an endpoint, ePBR will create a track using the probe method for the endpoint. If no probe method is defined at the endpoint level, the probe method configured for the service level will be used.

In events of device failures, traffic that was redirecting to the failed devices will redirect to other reachable devices, until the service is detected as failed. Resilient hashing is supported during device failures

for a service function deployed with multiple service endpoints. Traffic that was always being redirected to a specific service endpoint continues to redirect to the same device in events of failures of other service endpoints part of the same service fucntion.

ePBR supports the following fail-action mechanisms for its service chain sequences:

- Drop
- Forward
- Bypass

Drop indicates that the traffic must be dropped when the service in the current sequence is considered as failed. This is the default behavior when no fail-action is configured.

Forward indicates that upon failure of the service in the current sequence, traffic should use the regular routing. This fail-action mechanism is only supported when a single service function is defined in the chain.

Bypass indicates that the traffic must be redirected to the next service function in the chain when the service in the current sequence is considered as failed. For a service-chain with a single sequence, when using bypass traffic would use regular routing like the fail-action option of forward.

Load-Balancing Methods for Service Functions

Beginning with Cisco NX-OS 10.5(1)F, ePBR with GPO supports load-balancing traffic between service endpoints that are part of the same service function. Load-balance method may be configured for a service-chain if the same load-balance mechanism is desired for every service function in the chain. If one or more service functions or sequences inside the chain require a different load-balancing mechanism, this may be configured for the specific sequence inside the chain. Traffic may be load-balanced using source IP parameters, destination-IP parameters or source IP, destination IP along with the protocol indications available in the IP headers. ePBR with micro-segmentation ensures traffic is symmetrically load-balanced to the same service device in both directions.

Weighted Load-balancing

Beginning with Cisco NX-OS 10.5(1)F, ePBR with GPO supports load-balancing traffic to service endpoints proportional to the configured weights of the endpoints.

Each service endpoint configured inside a service-function can have a weight configuration. The weight range is 1-10. The total number of weights per service function is up to 128. The service-endpoints can be optionally configured based on the bandwidth or capacity of the device. If a service function is not configured with weights, all the service-endpoints configured inside the service function are considered to have a weight of 1, and the traffic is load-balanced through equal-cost multipath mechanism.

During endpoint failures, endpoints with higher weights will be preferred over endpoints with lower weights to receive the traffic of the failed endpoints.

Note that the weighted traffic distribution to the service devices is still dependent on the choice of the load-balancing algorithm and the distribution of the source and/or destination IP addresses of the traffic flows being received for service-chaining by the Nexus 9000 switch. See figure 2 for a weighted load-balancing arrangement.

Figure 5: Weighted Load-balancing

N+M Redundancy

Beginning with Cisco NX-OS 10.5(1)F, ePBR with GPO supports the ability to define service endpoints in hot-standby mode. M hot-standby service endpoints may be defined for a service function, with N primary (active) endpoints. When all primary service endpoints are available, no traffic is redirected to the hot-standby service endpoints.

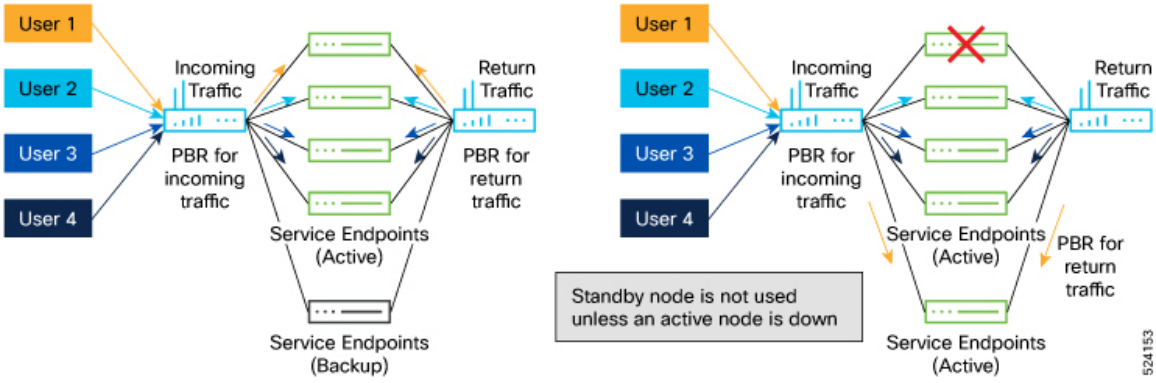
On failure of an active service endpoint inside an ePBR service function with hot-standby endpoints, traffic that was load-balanced to the failed service endpoint, is now redirected to an available hot-standby service endpoint.

On subsequent failures of more active endpoints and after all hot-standby endpoints have been utilized as backups for active endpoints, traffic that was handled by newly failed active endpoints may start being redirected to one or more available active and hot-standby endpoints.

When the active endpoint recovers, traffic that was being redirected to it, prior to its failure, will be restored to it. This behavior is unavoidable, and the traffic sessions may be required to get reestablished through the restored, stateful service endpoint.

Hot-standby endpoints may be configured with weights. On failure of a weighted active endpoint inside an ePBR service function with weighted hot-standby endpoints, traffic is first redirected to a weighted hot-standby endpoint with equal or higher weight than the failed active endpoint. See figure 3 for a N+M redundancy arrangement.

Figure 6: N+M Redundancy



Redirection to NAT Devices

Beginning with Cisco NX-OS 10.5(1)F, ePBR with GPO supports redirection of traffic to service devices that modify the destination and/or source IP addresses of the traffic. These devices may be external load-balancers, NATTing firewalls and CGNAT devices.

Service devices may perform only destination NAT (load-balancers with SNAT disabled), only source NAT (CGNAT devices for return traffic) or both (load-balancers with SNAT enabled).

Traffic to devices such as external load-balancers performing destination NAT in the forward direction do not need policy-based redirection but need to be permitted in order to reach the VIP address exposed by the load-balancer.

Similarly, traffic in the reverse direction returning to devices such as external load-balancers or CGNAT devices, that have performed Source NAT in the forward direction, do not need policy-based redirection, but need to be permitted.

Traffic to devices such as external load-balancers that do not have source NAT enabled, require policy-based redirection for traffic in the reverse direction.

As described above, traffic to these services needs to be handled in different ways based on their NAT capabilities. Additionally, due to the modification of the IP addresses of the traffic by these appliances, the destination and/or source security-group tags may be different before and after redirection to these services. Handling these variances may ordinarily require complex asymmetric, uni-directional contracts.

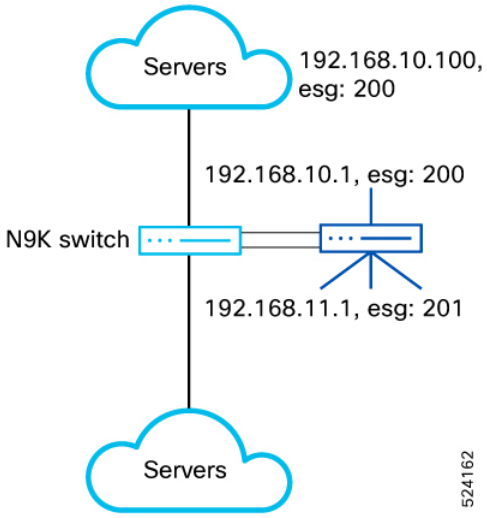
ePBR simplifies the contract creation for the user by allowing the user to indicate that the service function in the ePBR service-chain at a particular sequence, has destination and/or source NAT capabilities. This is done by configuring an action for the service inside the chain, for the forward and reverse directions of traffic.

- Services that only perform destination NAT on the traffic only are configured with action of route for the forward direction.
- Services that only perform both destination and source NAT on the traffic, are configured with action of route for both directions of traffic.
- Services that perform only source NAT on the traffic are configured with action of route only for the reverse direction of traffic.

The configuration options of route allow users to create a single contract from consumer to provider ESGs. This configuration reduces the burden of creating separate contracts between consumer to load- balancer then from load-balancer to the provider ESG due to the change in destination ESG.

When no action is configured for any direction, the service functions inside the chain are treated as requiring redirection in both directions. Fail-action and threshold features will not be supported for service functions in the service-chain that have action of route configured for either the forward or reverse directions.

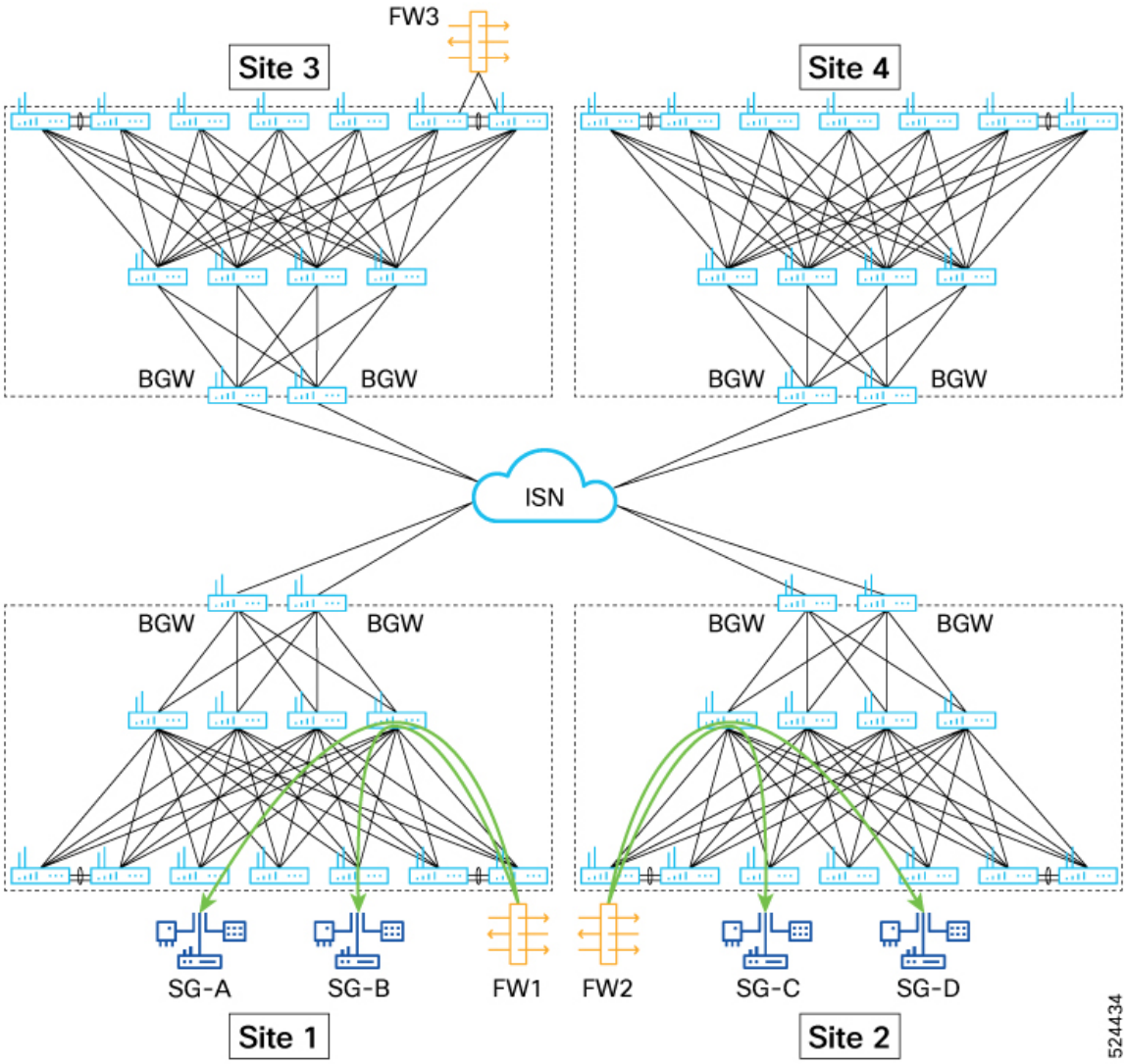
Figure 7: 2-arm Load-balancer (without SNAT) Service Device Insertion



ePBR and GPO Multi-Site

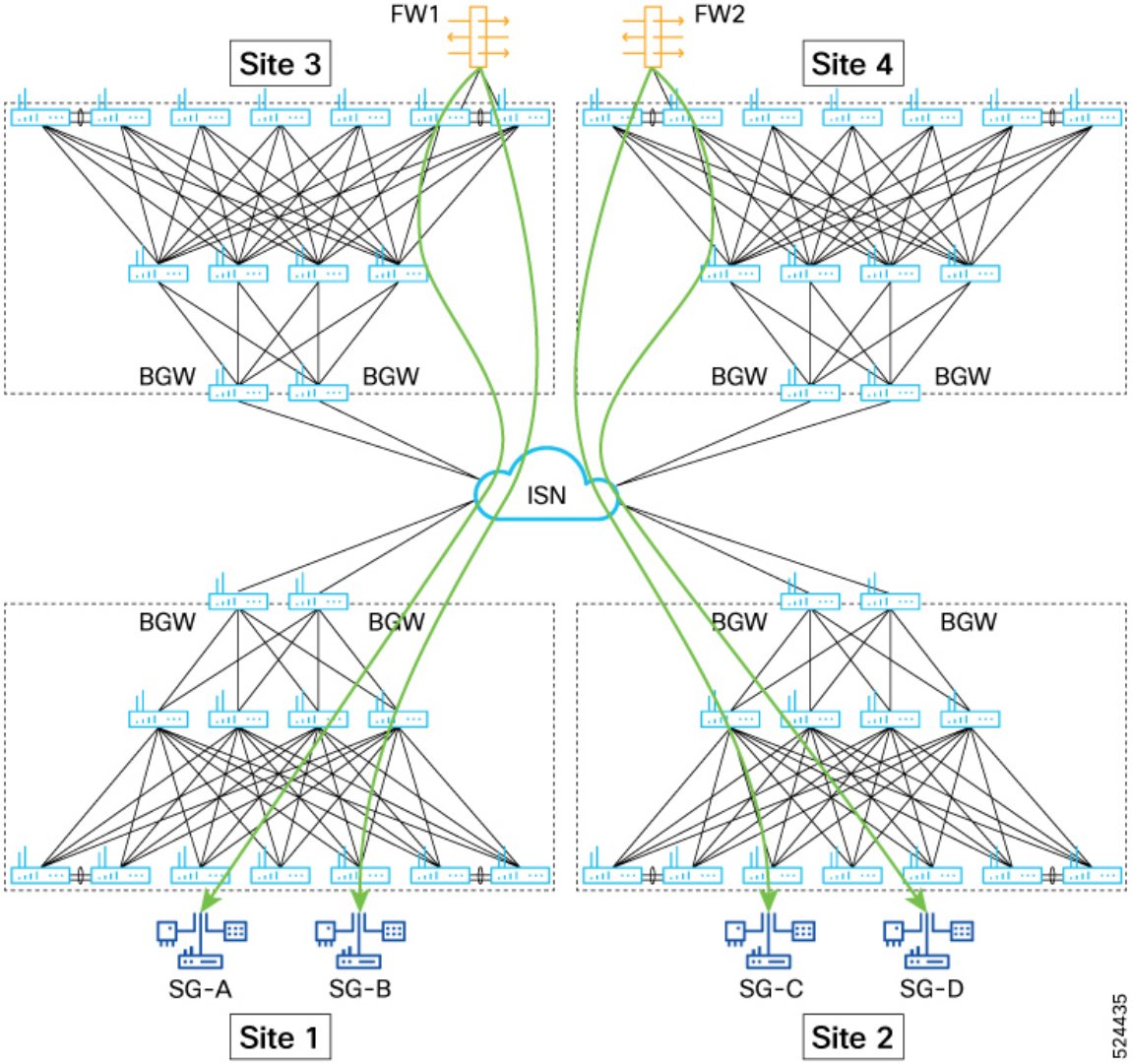
Beginning with Cisco NX-OS Release 10.5(2)F, traffic flows between endpoints of different security-groups belonging to multiple sites may be redirected to a service-chain by enabling multi-site mode for the service-chain. These single node or multi-node service-chains may consist of service functions such as firewall, load balancer, NAT, IPS, TCP optimizer and so on. Using this feature, users can interconnect and manage Security Group with service-chaining across different NX-OS VXLAN EVPN fabrics, whether those are physically collocated or geographically dispersed.

Figure 8: Local Site Local Site Security Groups with Local Service Chain



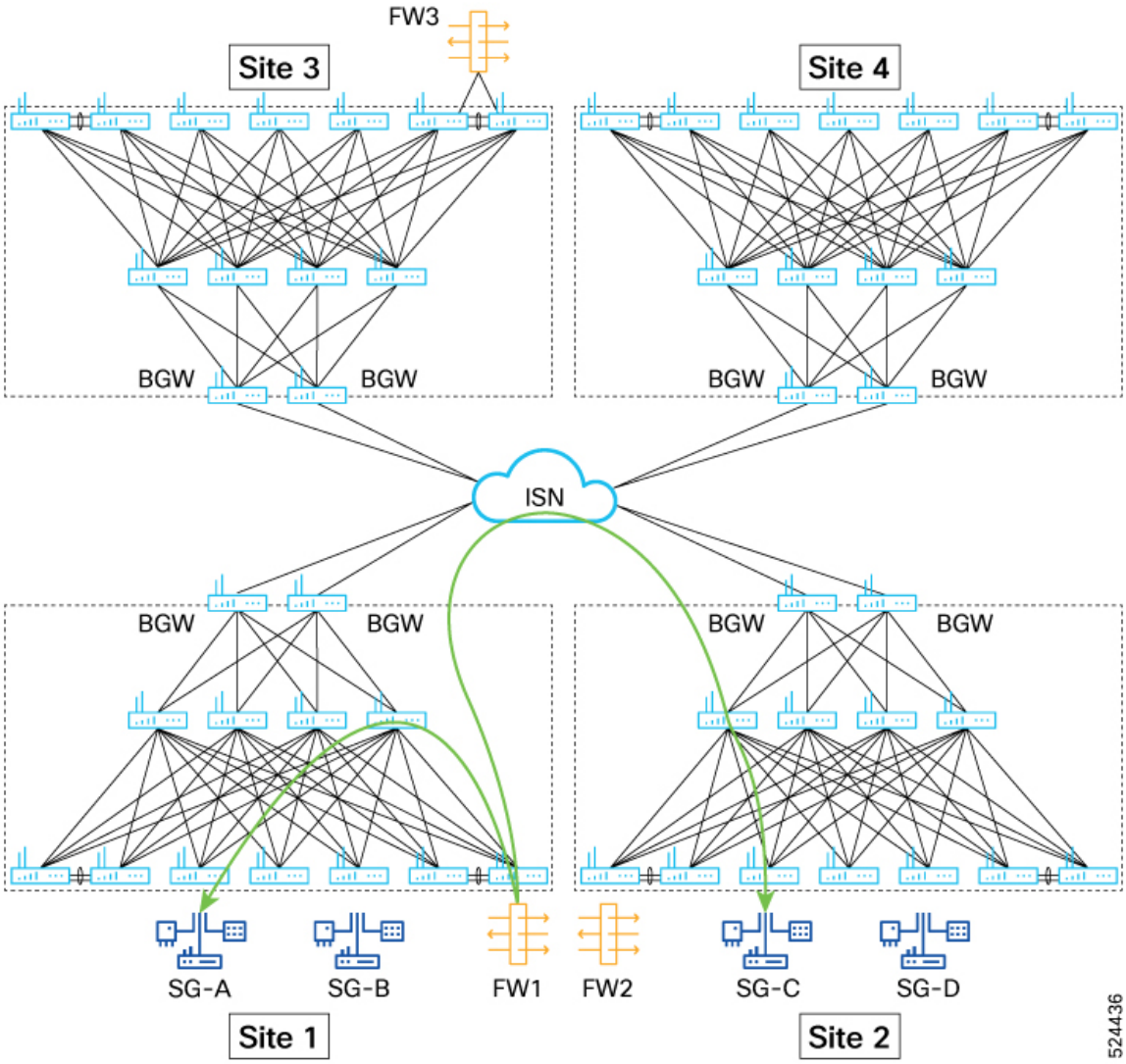
Sites 1 and 2 have their own service functions, and service chains are created between SG-A and SG-B, and SG-C and SG-D using the service functions FW 1 and FW 2 respectively within the same site. The failover service chain for site 1 can be created using FW2 from site 2, and for site 2 using FW1 from site 1.

Figure 9: Local Site Security Groups with No Local Service Chain



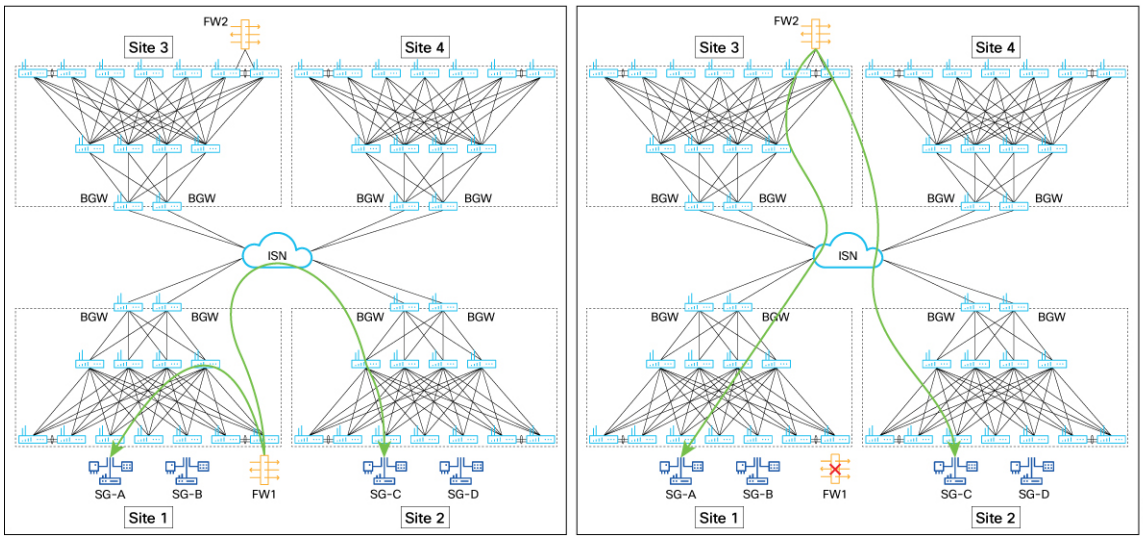
If there are no service functions available within the same site, users can use service functions available in the remote sites to create service chain using contract.

Figure 10: Service-Chain Inspection for Source and Destination Workload in Different Sites



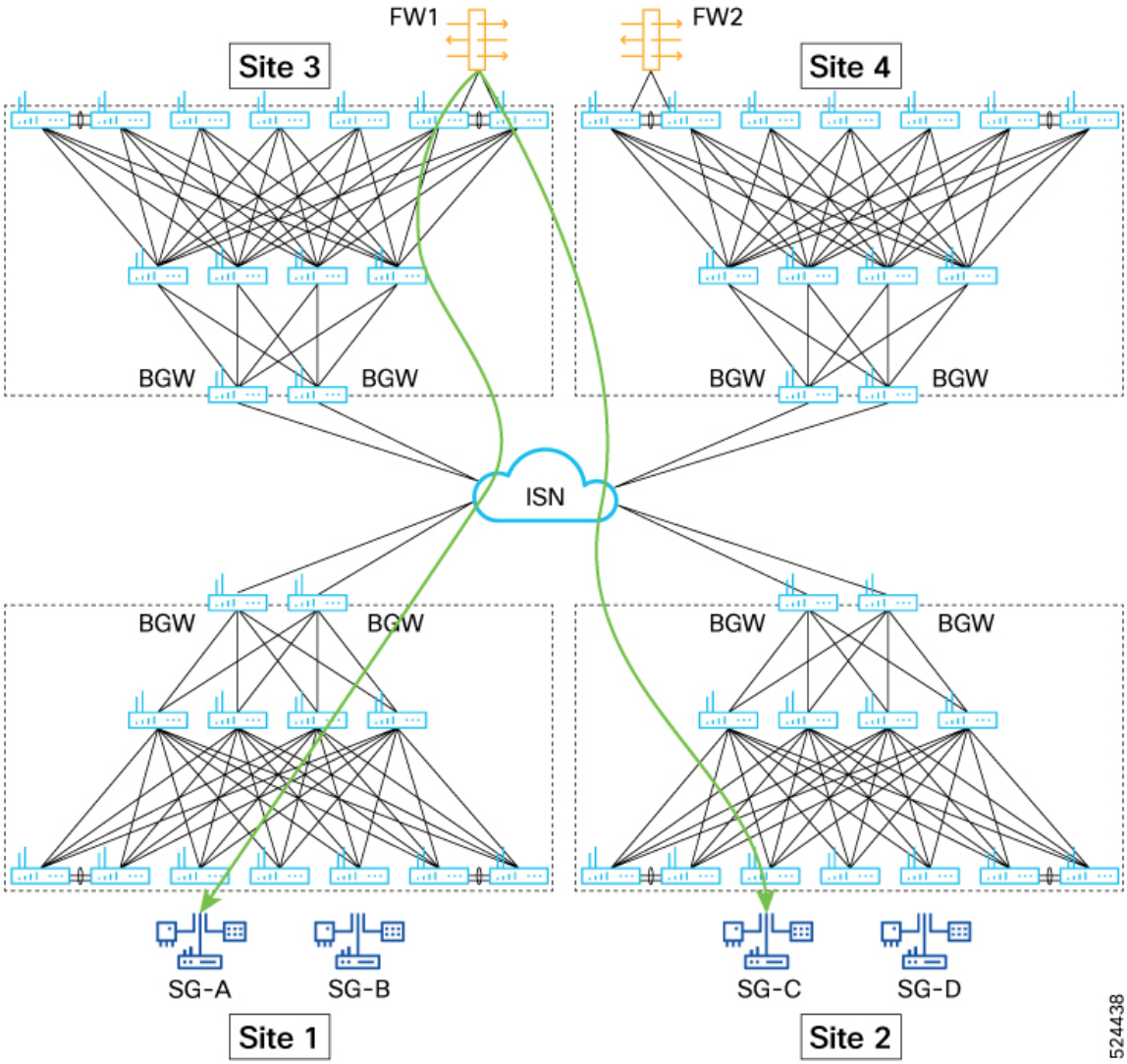
While configuring service chain for workloads that span over multiple sites, select a service chain that is either located at the source site or the destination site. The ePBR policy is enforced on the site that has smaller security group.

Figure 11: Service Chain only in One of the Two Site (Inter site Flow) and Failover



Inter-site workload with service chain inspection, where the service chain is only present in one site. Both forward and reverse flows should traverse the same chain. In the event of service chain failure, there should be a subsequent failover to the third-site service chain for both forward and reverse flows.

Figure 12: No Service Chain in source or destination sites



Inter-site workload with service chain inspection, where the service chain is not present in source or destination sites but only present in third site. Both forward and reverse flows should traverse the same chain.

ePBR Fail-over Group for Service-chain

Beginning with Cisco NX-OS Release 10.5(2)F, ePBR supports fail-over groups for service-chains to allow traffic to fail-over to service-chains located in remote sites of the fabric. Fail-over group is the collection of fallback service-chains that should be used when the primary service-chain fails. Users may configure fallback service-chains and assign preferences to them based on inter-site latency, geographical proximity or capacity. The fallback service-chains need to be defined in the system before they can be referenced as a member chain inside a fail-over group of the same number of service nodes as the primary chain. A maximum of 5 member chains may be configured inside a fail-over group. The following are some of the common deployment.

Guidelines and Limitations

ePBR with micro-segmentation has the following guidelines and limitations:

- Beginning with Cisco NX-OS Release 10.5(3)F, users can configure threshold value in percentage for the number of active endpoints in a service group. Once the percentage of active endpoints goes below the threshold, the service group is considered as down, and traffic is dropped, bypassed, or forwarded based on the fail action configured.
  - If no threshold value is configured or configured value is 0, this feature remains disabled.
  - After the service group is disabled, the service group is considered up again When the percentage of the active service endpoints in a service group equals or goes above the threshold value.
- ePBR with micro-segmentation is supported on all the platforms where micro-segmentation is supported. For more information, see **Guidelines and Limitations**.
- Beginning with Cisco NX-OS Release 10.5(3)F, support for ISSU is added for GPO based service redirection feature for multi-node and multisite fabrics.
- In NXOS 10.5(1)F SGACL based service redirection is only supported to a single service function in the chain. The service function may contain one or more layer-3 one-arm or dual-arm service endpoints.
- Load-balancer service functions inside an GPO based ePBR service-chain may only consist of a single load-balancer endpoint.
- Service functions with a mix of one-arm and dual-arm service endpoints are not supported.
- The sum total of weights across all active endpoints in the ePBR service cannot exceed 128.
- For the external load-balancer to monitor the health of the server cluster, contracts with permit action must be explicitly created between the layer4-7 security-tags of the load-balancer service and the servers.
- Service with one-arm devices must not be configured with a reverse security-group identifier.
- Services with dual-arm devices must be configured with a reverse security-group identifier that is different from the forward security-group.
- Services with dual-arm devices should use different service VLANs for the forward and reverse arms of the endpoints. The forward arms of one or more service endpoints in the service function can share one service VLAN and the reverse arms of one or more service endpoints can use a different service VLAN.
- Users must ensure that service VLANs are used exclusively for the service endpoints and are not used for any other host traffic. Hosts cannot be connected to such VLANs. This is required to avoid incorrect classification of such traffic.
- The forward and reverse security groups defined inside an ePBR service must be defined as layer4-7 security group selectors on the VXLAN leaf switches that have the connected interfaces (interface VLANs) configured.
- In NXOS 10.5(1)F service endpoint connected interfaces used in services must be interface VLANs only. Endpoint connected interfaces cannot be layer-3 physical interfaces, subinterfaces , layer-3 port-channels or port-channel subinterfaces, or any other interfaces that are supported for IPACL EPBR.
- While security-groups and service VLANs may be shared between ePBR services, users must ensure that the contracts that use these services in chains do not have conflicting match filters or actions.

- In NXOS 10.5(1)F, the service that the traffic is redirected to, must be configured in the same VRF context as the contract.
- Match class-maps for IPv4 traffic must be configured with service-chains containing IPv4 services and match class-maps for IPv6 traffic must be configured with service-chains containing IPv6 traffic.
- Traffic that is required to match any-any source and destination security-groups in the contract and requires redirection to service devices, may only redirect to one-arm service devices.

  Traffic that is required to match any-any source and destination security-groups in the contract cannot be configured to redirect to a multi-node service-chain.
- Users must ensure that multiple contracts using the same source and destination security-groups are not configured with policies and match class-maps having different service redirection results for the same traffic flows.
- When fail-action is configured for a sequence inside a service-chain , it is recommended that probing is consistently enabled for the service via service-level or endpoint-level probes.
- It is recommended that probe traffic is classified in a separate CoPP class. Otherwise, probe traffic may use the default CoPP class and might be dropped causing continuous IP SLA state changes during spikes in supervisor traffic. For information on CoPP configuration for IP SLA, see Configuring CoPP for IP SLA Packets.
- ePBR administrative and operational out-of-service features are not supported for services used in service redirection with micro-segmentation. For more information, see Configuring ePBR L3.
- Endpoint states of the forward and reverse arms of dual-arm devices are not synchronized automatically. If this is needed, identical probe track configuration on the forward and reverse arms should be used.
- Probe tracks configured for endpoints may be shared between the forward and reverse arms of the same endpoints, but not across endpoints in the same or different services.
- Probe tracks must be used for any automatic synchronization of endpoint states across the forward and reverse arms of dual-arm devices.
- The service nodes can be part of either the source VRF or the destination VRF or in a separate VRF. If some of the service nodes are part of the source VRF and some are a part of the destination VRF, all consecutive elements following the source, must uniformly pertain to the source VRF. Once the VRF for an element in the chain pertains to the destination VRF, all consecutive elements following this until the end of the service-chain must pertain to the destination VRF.
- Dual-arm service end-points cannot have each arm in a different VRF.

Multinode Service-Chaining Guidelines and Limitations:

- A maximum of 5 service functions (nodes) are supported in a service chain.
- In multinode configuration, only bypass and drop fail-action options are supported. Fail-action option of forward is not supported.
- Only a single service function that performs IP address translation (load-balancer or CGNAT devices) may be configured inside a multi-node service-chain.

Multi-site Service-Chaining Guidelines and Limitations:

- All service functions for a given service chain should belong to a single site.

- A maximum of 5 fail-over service-chains are supported inside a failover group.
- A maximum of 10 sites are supported in a multi-site fabric that is using EPBR service-chains with VXLAN Group Policy Options.
- Multi-Site failover options are not supported with service-chains that consist of load-balancer devices because load-balancer devices have unique VIPs and failover to a different load-balancer implies a VIP changed which is outside the scope of the VTEP making the failover decision.
- Load-balancer devices without Source NAT enabled, used in the service-chain, and the servers they are load-balancing to must co-exist in the same site.
- A service-chain and the fail-over service-chains it is configured to use must consist of the same number of service nodes. Each service node may, however, have a varying number of service-endpoints.
- Every service-node inside the service-chain and the failover service-chains it is configured to use must be configured with the same service security-groups.
- Before configuring multi-site, ensure that the TCAM programming scale is set to less than 80% of the limit specified for a single-site configuration. This is because enabling the mode multi-site knob increases TCAM programming requirements compared to the same configuration without the knob.

Configuring ePBR for Micro-segmentation

Configuring ePBR Service

Before you begin

The following section provides information about configuring ePBR services.

SUMMARY STEPS

1. **configure terminal**
2. **epbr service** *service-name*
3. **[no]threshold** *threshold-value*
4. **vrf** *vrf-name*
5. **[no] security-group <fwdGrp> [reverse**<*revGrp*>]
6. **[no] probe {icmp |** <*l4-proto*> **<port-num> [control**<*status*> **] | http get [** <*url-name*> **[version** <*ver*> **] | dns host** <*host-name*> **ctp} [frequency** <*freq-num*> **| timeout** <*timeout*> **| retry-down-count** <*down-count*> **| retry-up-count** <*up-count*> **| source-interface** <*src-intf*> **| reverse** <*rev-src-intf*> **]+**
7. **service-endpoint {ip** *ipv4-address* | **ipv6** *ipv6-address*}
8. **probe track** *track-ID*
9. **reverse {ip** *ipv4-address* | **ipv6** *ipv6-address*}
10. **mode hot-standby**
11. **weight** <*weight*>
12. **exit**

DETAILED STEPS

Procedure

	Command or Action	Purpose

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

Step 2	**epbr service** *service-name* **Example:** `switch(config)# epbr service firewall`	Creates a new ePBR service function.
Step 3	**[no]threshold** *threshold-value* **Example:** `switch(config)# threshold 26`	Configures threshold in percentage for the number of active endpoints in a service group. Default: 0 Range: 0-100
Step 4	**vrf** *vrf-name* **Example:** `switch(config-epbr-svc)# vrf tenant_A`	Specifies the VRF for the ePBR service function.

Step 5	**[no] security-group <fwdGrp> [reverse**<*revGrp*>]
	Example:
	`switch(config-epbr-svc)# security-group 10 reverse 20`
	`switch(config-epbr-svc)# security-group 30`
		Configures forward and reverse service security-group tags. For single arm devices, a single forward security-group must be specified. For dual arm devices the forward and reverse security-group must be unique.
		The **no** form of this command removes the configuration.

Step 6	**[no] probe {icmp** | <*l4-proto*> **<port-num> [control**<*status*> **] | http get [** <*url-name*> **[version** <*ver*> **] | dns host** <*host-name*> **ctp} [frequency** <*freq-num*> **| timeout** <*timeout*> **| retry-down-count** <*down-count*> **| retry-up-count** <*up-count*> **| source-interface** <*src-intf*> **| reverse** <*rev-src-intf*> **]+**	Configures the probe for the service function. The same configuration may also be applied for the forward and reverse arms of service endpoints. The no form of this command removes the configuration. For service-endpoints distributed in a VXLAN environment, you must configure source loopback interfaces for the probe, so that a unique source IP may be used for IP SLA sessions.
Step 7	**service-endpoint {ip** *ipv4-address* | **ipv6** *ipv6-address*} **Example:** `switch(config-vrf)# service-endpoint ip 172.16.1.200`	Configures service endpoint for the ePBR service. You can repeat steps 6 to 10 to configure another ePBR service endpoints.
Step 8	**probe track** *track-ID* **Example:** `switch(config-epbr-fwd-svc)# probe track 30`	Configures user-defined track for the forward or reverse arm of the service endpoint.

	Command or Action	Purpose
Step 9	**reverse {ip** *ipv4-address* | **ipv6** *ipv6-address*} **Example:** `switch(config-epbr-fwd-svc)# reverse ip 172.16.30.200`	Defines the reverse IP address for dual-arm service endpoints. Note that this is not needed for one-arm endpoints.
Step 10	**mode hot-standby** **Example:** `switch(config-epbr-fwd-svc)# mode hot-standby`	Configures the service-endpoint as a hot-standby endpoint.
Step 11	**weight** <*weight*> **Example:** `switch(config-epbr-fwd-svc)# weight 6`	Configures the weight for the active or hot-standby endpoint. Default value is 1.
Step 12	**exit** **Example:** `switch(config-vrf)# exit`	Exits the ePBR service configuration mode.

Configuring ePBR Service-chain

SUMMARY STEPS

1. **configure terminal**
2. **[no] epbr service-chain** <*chain-name*>
3. **[no] mode multisite [failover-group <group-name>]**
4. **load-balance method** <*lb-method*> **{ src-ip | dst-ip | src-dst-ipprotocol}**
5. *sequence-number* **set service** *service-name***[fail-action { bypass | drop | forward}]**
6. **action {route | redirect} [reverse-action {route| redirect}]**

DETAILED STEPS

Procedure

	Command or Action	Purpose

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

Step 2	**[no] epbr service-chain** <*chain-name*> **Example:** `Switch(config-epbr-svc-chain)# epbr service-chain web`	Configures ePBR service-chain. The no form of this command removes the configuration.

	Command or Action	Purpose
Step 3	**[no] mode multisite [failover-group <group-name>]** **Example:** `mode multisite failover-group fallback-web-chain3`	Beginning with NX-OS 10.5(2)F, you can configure mode multi-site and fail-over group for service-chains. • Fail-over groups may only be configured when mode multi-site is enabled for the service-chain. You can use the mode multi-site without using fail-over groups.
Step 4	**load-balance method** *<lb-method>* **{ src-ip | dst-ip | src-dst-ipprotocol}** **Example:** `switch(config-epbr-svc-chain)# load-balance method src-ip`	Configures the load-balance method for the ePBR service-chain. The same configuration may also be applied to the individual service functions inside the service-chain. Default option is **src-dst-ipprotocol**.

Step 5	*sequence-number* **set service** *service-name***[fail-action { bypass | drop | forward}]**
	Example:
	`switch(config-epbr-svc-chain)#`
	`10 set service fw2 fail-action drop`
	`20 set service tcp_optim2 fail-action bypass`
		Specifies the service function at the specific sequence in the chain and the fail-action mechanism for that sequence.
		Beginning with NX-OS 10.5(2)F, GPO with multi-node service-chain is supported.

Step 6	**action {route | redirect} [reverse-action {route| redirect}]** **Example:** `switch(config-epbr-svc-chain-seq)# action route reverse-action route`	Configure the forward and/or reverse action for the service in the chain to indicate destination and/or source NAT capabilities of the service. Default option is **redirect**.

Configuring Failover Group

Follow the steps to configure the fail-over group.

SUMMARY STEPS

1. **configure terminal**
2. **epbr service-chain** *service-chain-name*
3. **epbr failover-group** *failover-group-name*
4. **[no] service-chain <name> preference <preference>**

DETAILED STEPS

Procedure

	Command or Action	Purpose

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

	Command or Action	Purpose
Step 2	**epbr service-chain** *service-chain-name* **Example:** `Switch(config-epbr-svc-chain)# epbr service-chain web`	Configures service-chain.
Step 3	**epbr failover-group** *failover-group-name* **Example:** `switch(config-epbr-svc-chain)# epbr failover-group fallback-web-chain1`	Configures fail-over group.
Step 4	**[no] service-chain <name> preference <preference>** **Example:** `switch(config-epbr-fail-group)# service-chain site1-web-chain preference 20`	Configure fallback service-chain inside the fail-over group and assign preferences to the fallback service-chain.

Verifying ePBR Service-chain Configuration

Use the following commands to verify the ePBR service-chain configuration:

SUMMARY STEPS

1. **show epbr service [<svc-name>]**
2. **show epbr service-chain [<chain-name>] [reverse]**
3. **show tech-support epbr**
4. **show consistency-checker epbr service-chain { <svcChainName> | all }**
5. **show running-config epbr**
6. **show startup-config epbr**

DETAILED STEPS

Procedure

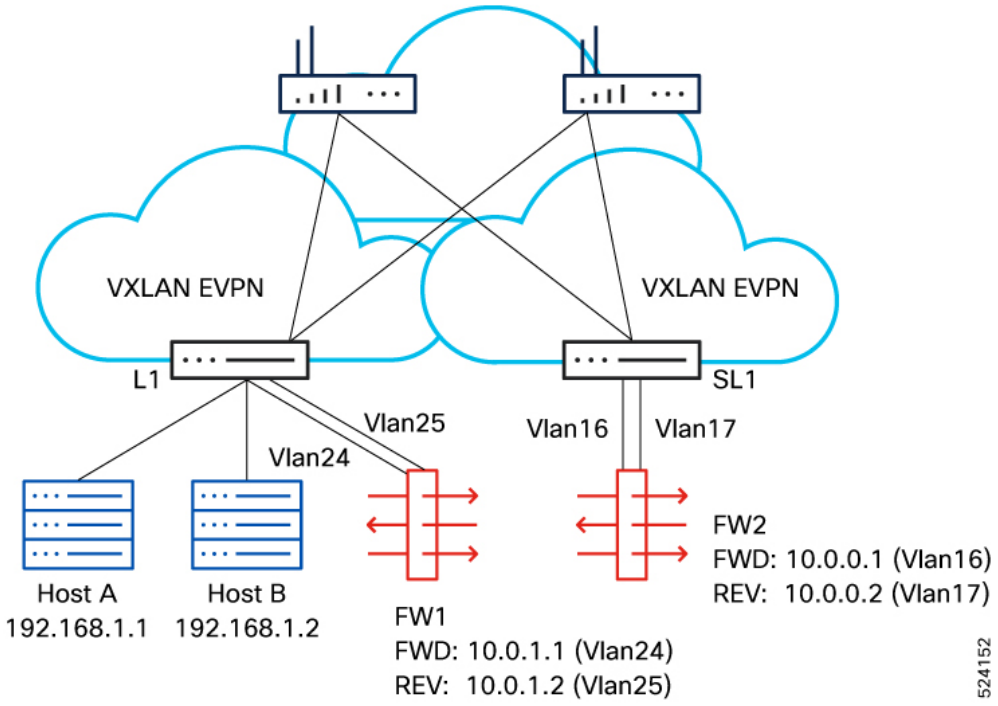
	Command or Action	Purpose
Step 1	**show epbr service [<svc-name>]** **Example:** `switch# show epbr service fw`	Displays information on the ePBR service function and endpoints.
Step 2	**show epbr service-chain [<chain-name>] [reverse]** **Example:** `switch# show epbr service-chain web`	Displays information on the ePBR service-chain in forward or reverse direction.
Step 3	**show tech-support epbr** **Example:** `switch# show tech-support epbr`	Displays the technical support information for ePBR.

	Command or Action	Purpose
Step 4	**show consistency-checker epbr service-chain { <svcChainName> | all }** **Example:** `show consistency-checker epbr service-chain web`	Performs consistency checks on ePBR configuration, redirection information for ePBR in the control plane and health monitoring mechanisms that are enabled.
Step 5	**show running-config epbr** **Example:** `switch# show running-config epbr`	Displays the running configuration for ePBR.
Step 6	**show startup-config epbr** **Example:** `switch# show startup-config epbr`	Displays the startup configuration for ePBR.

Configuration Examples for SGACL service-chaining Configuration

See figure 5 for the configuration example showing SGACL service-chaining configuration.

Figure 13: Configuration Example



1. Create layer4-7 selectors for the service.

```
security-group 2010 name FWD
  type layer4-7
```

```
  match interface vlan 24
  match interface vlan 16
security-group 2011 name REV
  type layer4-7
  match interface vlan 25
  match interface vlan 17
```

2. Creating ePBR service and endpoints.

```
epbr service fw
  vrf tenant
  security-group 2010 reverse 2011
  probe tcp 80 frequency 5 timeout 3 source-interface
   loopback10 reverse loopback11
  service-end-point ip 10.0.1.1
    reverse ip 10.0.1.2
  service-end-point ip 10.0.0.1
    reverse ip 10.0.0.2
```

3. Create security-group selectors for host traffic.

```
security-group 5051 name sec_5051
  match connected-endpoints vrf tenant ipv4 151.1.1.0/24

security-group 5050 name sec_5050
  match connected-endpoints vrf tenant ipv4 150.1.1.0/24
```

4. Create security class-maps to define the layer3, layer-4 match criteria.

```
class-map type security match-any class_ipv4_tcp
  match ipv4 tcp dport 80
  match ipv4 tcp dport 443
```

5. Configure the ePBR service-chain. Configuration of class-maps, policy-maps and contracts under vrf need to be consistent on all leafs.

```
epbr service-chain web
  load-balance method src-dst-ipprotocol
  10 set service fw fail-action drop
```

6. Configure the security policy-map and attach the service-chain to the required match class-map.

```
policy-map type security web_policy
  class type security class_ipv4_tcp
    service-chain web
```

7. Configure the contract.

```
vrf context tenant
  security contract source 5050 destination 5051 policy web_policy
```

For more details on moving the VRF context to enforced mode, see *Configuring Security contracts between Security Groups*.

Verifying Configuration

- The following example shows how to verify ePBR service and endpoint.

```
show epbr service fw

Legend:

Operational State (Op-STS):  UP:Reachable, DOWN:Unreachable,
```

```
                              SVC-ADMIN-DOWN:Service shut

                              ADMIN-DOWN:Admin shut, OPER-DOWN:Out-of-service

Probe:
Protocol/Frequency(sec)/Timeout(sec)/Retry-Up-Count/Retry-Down-Count

Hold-down Threshold:          Count/Time(sec)

Service mode:                 Full:Full-Duplex, Half:Half-Duplex

Type:                         L3:Layer-3, L2:Layer-2

Threshold:                    Threshold High/Low


Name                                Type      Service mode  VRF

=============================================================================================


fw                                  L3        Full
tenant


Security-group    Reverse security-group   Threshold

=======================================================

2010              2011


Endpoint IP/Intf         Track SLA       Op-ST          Probe               Hold-down
 Role Weight

Reverse IP/Intf          Track SLA       Op-ST          Probe

=============================================================================================



10.0.1.1/               1  20001     UP          TCP/5/3/0/0
       A     1

10.0.1.2/               3   20003     UP          TCP/5/3/0/0


10.0.0.1/               2   20002     UP             TCP/5/3/0/0
          A     1

10.0.0.2/               4   20004     UP             TCP/5/3/0/0
```

- The following example shows how to verify the ePBR service-chain in forward or reverse direction.

```
show epbr service-chain web

Service-chain : web

    service:fw, sequence:10, fail-action:Drop
```

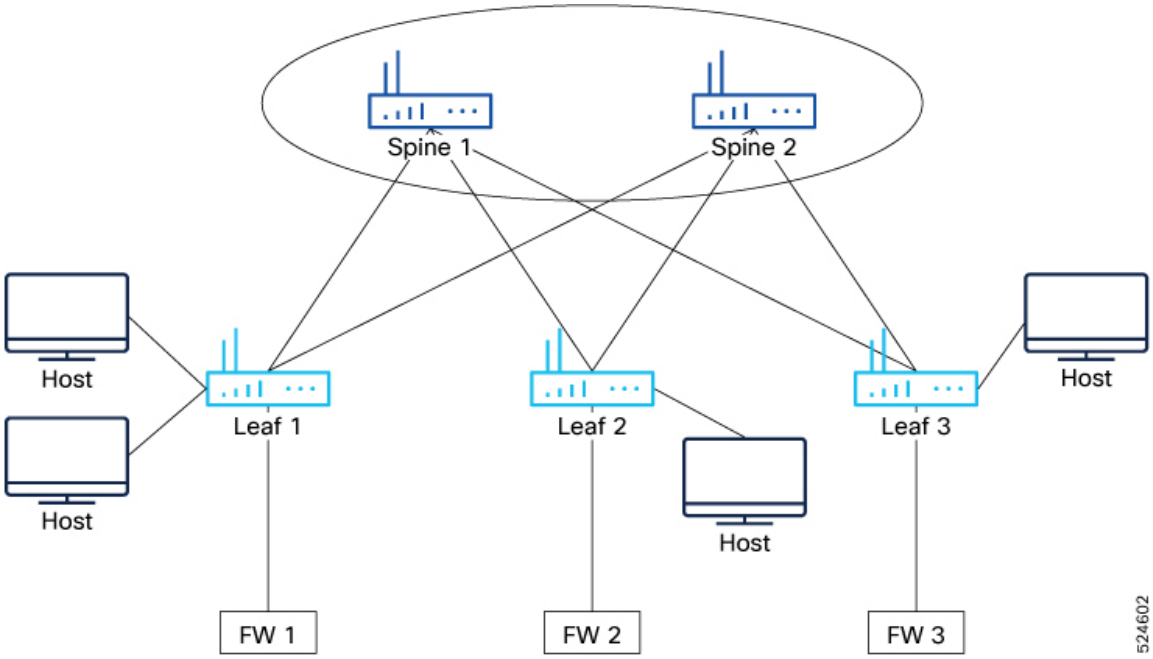
```
      load-balance: Source-Destination-ipprotocol, action:Redirect

      state:UP

      IP 10.0.1.1 track 1 [UP]

      IP 10.0.0.1 track 2 [UP]


show epbr service-chain web reverse


Service-chain : web

    service:fw, sequence:10, fail-action:Drop

      load-balance: Source-Destination-ipprotocol, action:Redirect

      state:UP

      IP 10.0.1.2 track 3 [UP]

      IP 10.0.0.2 track 4 [UP]
```

- The following example shows how to verify consistency checker for service-chain.

```
show consistency-checker epbr service-chain chain1
EPBR CC: Service Chain validation passed
show consistency-checker epbr service-chain all
EPBR CC: Service Chain validation passed
```

Configuration Example for Multi-node Single Site Service-chaining

Figure 14: Configuration example



Following is a configuration example that has three firewalls as services that are part of the service chain. Each firewall is implimented with multiple service endpoints.

1. Configuring ePBR service fw1

```
epbr service fw1
  vrf tenant
  security-group 2010 reverse 2011
  probe icmp frequency 2 retry-down-count 2 retry-up-count 1 timeout 1 source-interface
 loopback3 reverse loopback4
  service-end-point ip 10.1.1.2
    weight 10
    reverse ip 11.1.1.2
  service-end-point ip 18.1.1.2
    reverse ip 19.1.1.2
  service-end-point ip 20.1.1.2
    mode hot-standby
    reverse ip 21.1.1.2
  service-end-point ip 253.1.1.2
    mode hot-standby
    weight 10
    reverse ip 254.1.1.2
  service-end-point ip 26.1.1.2
    weight 5
    reverse ip 27.1.1.2
  service-end-point ip 34.1.1.2
    mode hot-standby
    weight 6
    reverse ip 35.1.1.2
```

2. Configuring ePBR service fw2

```
epbr service fw2
  vrf tenant
  security-group 2013
  probe icmp frequency 2 retry-down-count 2 retry-up-count 1 timeout 1 source-interface
 loopback3 reverse loopback4
  service-end-point ip 255.1.1.2
    mode hot-standby
  service-end-point ip 50.1.1.2
    weight 10
  service-end-point ip 54.1.1.2
    weight 5
  service-end-point ip 58.1.1.2
  service-end-point ip 59.1.1.2
    mode hot-standby
    weight 10
  service-end-point ip 62.1.1.2
    mode hot-standby
    weight 6
```

3. Configuring ePBR services fw3

```
epbr service fw3
  vrf tenant
  security-group 2014 reverse 2015
  probe icmp frequency 2 retry-down-count 2 retry-up-count 1 timeout 1 source-interface
 loopback3 reverse loopback4
  service-end-point ip 12.1.1.2
    weight 10
    reverse ip 13.1.1.2
  service-end-point ip 22.1.1.2
    weight 10
    reverse ip 23.1.1.2
  service-end-point ip 32.1.1.2
```

```
    weight 5
    reverse ip 33.1.1.2
  service-end-point ip 40.1.1.2
    reverse ip 41.1.1.2
```

4. Configure the ePBR multi-node service-chain

```
epbr service-chain FW-chain-v4
  load-balance method dst-ip
  10 set service service1-v4-2arm fail-action bypass
    load-balance method src-ip
  20 set service service3-v4-1arm fail-action drop
  30 set service service5-v4-2arm fail-action bypass
    load-balance method src-dst-ipprotocol
```

Verifying Multi-node Service Chain

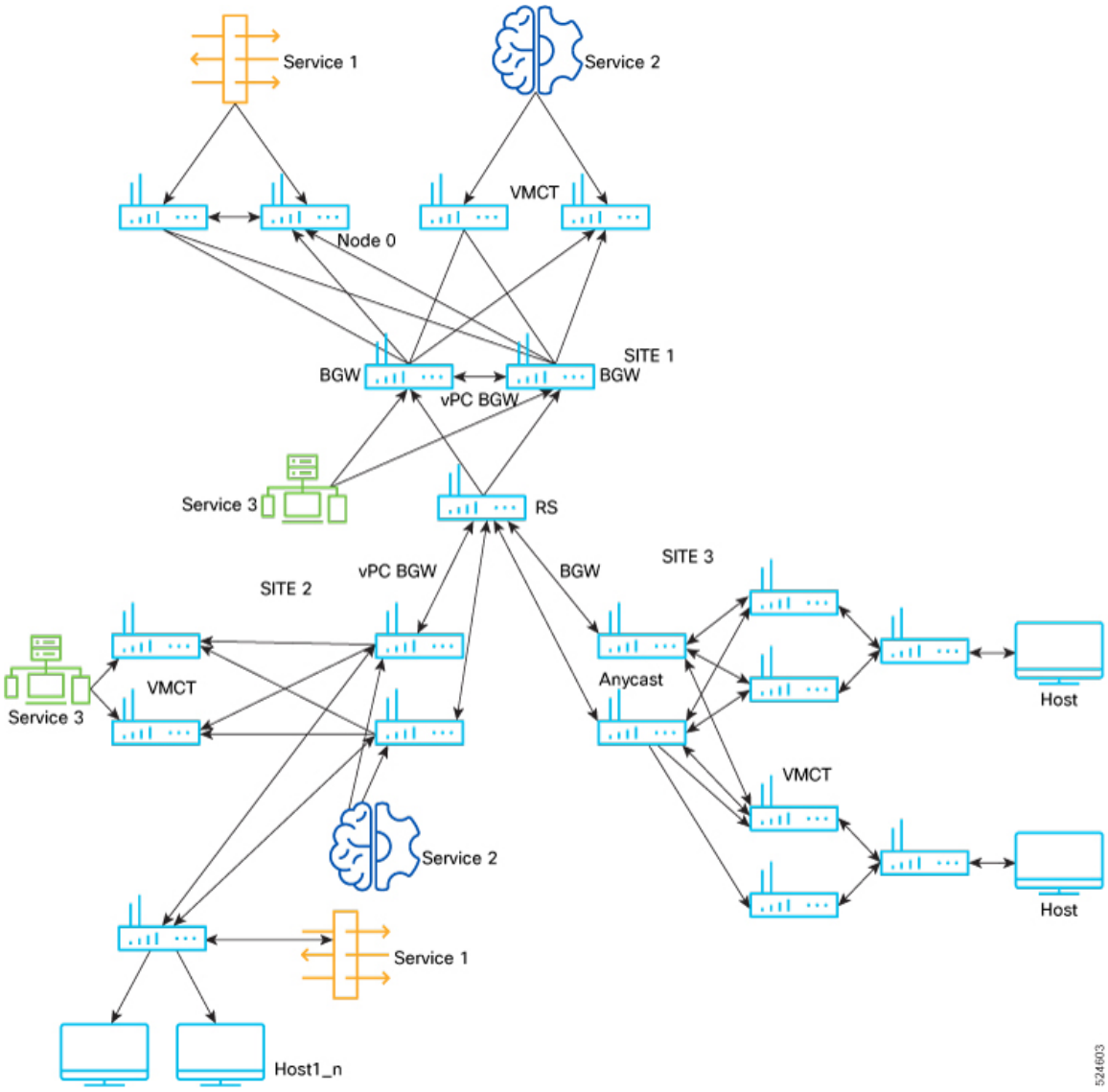
sh epbr service-chain FW-chain-v4

```
Service-chain : FW-chain-v4  state:UP

    service:fw1, sequence:10, fail-action:Bypass

      load-balance:Source-Destination-ipprotocol, action:Redirect

      state:UP

      IP 10.1.1.2 track 1 [UP]

      IP 18.1.1.2 track 2 [UP]

      IP 26.1.1.2 track 3 [UP]

      IP 20.1.1.2 track 4 [UP] [HOT-STANDBY]

      IP 34.1.1.2 track 5 [UP] [HOT-STANDBY]

      IP 253.1.1.2 track 6 [UP] [HOT-STANDBY]

    service:fw2, sequence:20, fail-action:Drop

      load-balance:Source-Destination-ipprotocol, action:Redirect

      state:UP

      IP 50.1.1.2 track 7 [UP]

      IP 54.1.1.2 track 8 [UP]

      IP 58.1.1.2 track 9 [UP]

      IP 59.1.1.2 track 10 [UP] [HOT-STANDBY]

      IP 62.1.1.2 track 11 [UP] [HOT-STANDBY]

      IP 255.1.1.2 track 12 [UP] [HOT-STANDBY]

    service:fw3, sequence:30, fail-action:Bypass

      load-balance:Source-Destination-ipprotocol, action:Redirect

      state:UP
```

```
IP 12.1.1.2 track 13 [UP]

IP 22.1.1.2 track 14 [UP]

IP 32.1.1.2 track 15 [UP]

IP 40.1.1.2 track 16 [UP]
```

Configuration example for multi-site ePBR with GPO

Figure 15: Configuration example



Site 1

1. Create security class-maps to define the layer3, layer-4 match criteria.

```
class-map type security match-any web_class
  match ipv4 tcp dport 80
```

2. Configure the security policy-map and attach the service-chain to the required match class-map

```
policy-map type security web
  class type security web_class
    service-chain site1-web-chain
```

3. Creating ePBR service and endpoints.

```
epbr service fw
  security-group 100
  probe icmp
  service-end-point ip 10.1.1.2
  service-end-point ip 20.1.1.2
   mode hot-standby

epbr service fw2
  security-group 100
  probe icmp
  service-end-point ip 11.1.1.2

epbr service fw3
  security-group 100
  probe icmp
  service-end-point ip 13.1.1.2
```

4. Configuring Multi-site mode and fail over group and chain

```
epbr service-chain site1-web-chain
  mode multisite failover-group fallback-web-chain1
  load-balance method dst-ip
  10 set service fw fail-action drop

epbr service-chain site2-web-chain
  load-balance method dst-ip
  10 set service fw2 fail-action drop

epbr service-chain site3-web-chain
  load-balance method dst-ip
  10 set service fw3 fail-action drop

epbr failover-group fallback-web-chain1
  service-chain site2-web-chain preference 5
  service-chain site3-web-chain preference 20
```

Site 2

1. Create security class-maps to define the layer3, layer-4 match criteria.

```
 class-map type security match-any web_class
  match ipv4 tcp dport 80
```

2. Configure the security policy-map and attach the service-chain to the required match class-map.

```
policy-map type security web
  class type security web_class
    service-chain site2-web-chain
```

3. Creating ePBR service and endpoints.

```
epbr service fw
  security-group 100
  probe icmp
  service-end-point ip 10.1.1.2
  service-end-point ip 20.1.1.2
   mode hot-standby
```

```
 epbr service fw2
  security-group 100
  probe icmp
  service-end-point ip 11.1.1.2

epbr service fw3
  security-group 100
  probe icmp
  service-end-point ip 13.1.1.2
```

4. Configuring Multi-site mode and fail over group and chain

```
epbr service-chain site1-web-chain
  load-balance method dst-ip
  10 set service fw fail-action drop

epbr service-chain site2-web-chain
  mode multisite failover-group fallback-web-chain2
  load-balance method dst-ip
  10 set service fw2 fail-action drop

epbr service-chain site3-web-chain
  load-balance method dst-ip
  10 set service fw3 fail-action drop

epbr failover-group fallback-web-chain2
  service-chain site1-web-chain preference 5
  service-chain site3-web-chain preference 25
```

Site 3

1. Create security class-maps to define the layer3, layer-4 match criteria.

```
class-map type security match-any web_class
  match ipv4 tcp dport 80
```

2. Configure the security policy-map and attach the service-chain to the required match class-map

```
policy-map type security web
  class type security web_class
    service-chain site3-web-chain
```

3. Creating ePBR service and endpoints

```
epbr service fw
  security-group 100
  probe icmp
  service-end-point ip 10.1.1.2
  service-end-point ip 20.1.1.2
   mode hot-standby


epbr service fw2
  security-group 100
  probe icmp
  service-end-point ip 11.1.1.2

epbr service fw3
  security-group 100
  probe icmp
  service-end-point ip 13.1.1.2
```

4. Configuring service-chain and multi-site

```
epbr service-chain site1-web-chain
  load-balance method dst-ip
```

```
  10 set service fw fail-action drop


epbr service-chain site2-web-chain
  load-balance method dst-ip
  10 set service fw2 fail-action drop

 epbr service-chain site3-web-chain
  mode multisite failover-group fallback-web-chain3
  load-balance method dst-ip
  10 set service fw3 fail-action drop
```

5. Configure failover group

```
epbr failover-group fallback-web-chain3
  service-chain site1-web-chain preference 20
  service-chain site2-web-chain preference 25
```

Verifying Multi-site Configuration

You can use the following show commands to verify multi-site configuration.

- The following example shows how to verify ePBR service-chain state.

```
show epbr service-chain site1-web-chain
Service-chain : site1-web-chain  state:DOWN


 mode: multisite, failover-group:fallback-web-chain [AVAILABLE][IN USE]  failover-chain:
site3-web-chain
   service:fw, sequence:10, fail-action:Drop
     load-balance:Destination-ip, action:Redirect
     state:DOWN
     IP 10.1.1.2 track 9 [DOWN]
     IP 20.1.1.2 track 10 [DOWN][HOT-STANDBY]
   service:tcp_optim, sequence:20, fail-action:Bypass
     load-balance:Destination-ip, action:Redirect
     state:UP
     IP 30.1.1.2 track 11 [UP]
```

- The following example shows how to get the details of the failover chains inside the failover group.

```
show epbr failover-group fallback-web-chain

Failover group : fallback-web-chain
    Failover Service-chain : site2-web-chain  Preference: 1  state: DOWN
      service:fw2, sequence:10, fail-action:Drop
        load-balance:Destination-ip, action:Redirect
        state:DOWN
        IP 11.1.1.2 track 12 [DOWN]
      service:tcp_optim2, sequence:20, fail-action:Bypass
        state: UP
        load-balance:Destination-ip, action:Redirect
        state:UP

        IP 12.1.1.2 track 13 [UP]

    Failover Service-chain : site3-web-chain  Preference: 2  state: UP
      service:fw3, sequence:10, fail-action:Drop
        load-balance:Destination-ip, action:Redirect
        state:UP
        IP 13.1.1.2 track 14 [UP]
      service:tcp_optim2, sequence:20, fail-action:Bypass
        load-balance:Destination-ip, action:Redirect
```

```
state:UP

IP 14.1.1.2 track 15 [UP]
```