Enhanced Policy Based Routing

Table 1: Feature History

Feature Name	Release Information	Description
Enhanced Policy Based Routing for Cisco Catalyst SD-WAN	Cisco IOS XE Catalyst SD-WAN Release 17.4.1a Cisco vManage Release 20.4.1	This release extends Enhanced Policy Based Routing (ePBR) to Cisco Catalyst SD-WAN. ePBR is a protocol-independent traffic-steering mechanism that routes traffic based on flexible policies for traffic flows. You can create ePBR policies using CLI add-on templates in Cisco SD-WAN Manager.

Overview of ePBR

Enhanced Policy Based Routing (ePBR) is an advanced version of Policy Based Routing (PBR). With this feature, traffic forwarding is based on policies rather than routing tables, and gives you more control over routing. ePBR extends and complements the existing mechanisms provided by routing protocols. ePBR is an advanced local data policy that routes traffic based on flexible match criteria such as IPv4 and IPv6 addresses, port numbers, protocols, or packet size.

ePBR matches traffic using flexible Cisco Common Classification Policy Language (C3PL language). It supports matching prefixes, applications, Differentiated Services Code Point (DSCP), Security Group Tags (SGT), and so on. With ePBR, based on match conditions, you can configure a single or multiple next hops for traffic forwarding. You also have the option to configure Internet Protocol Service Level Agreement (IP SLA) tracking. If a configured next hop is unavailable, traffic is routed to the next available hop through dynamic probing enabled by the IP SLA tracker.

Features and Benefits

- Supports both IPv4 and IPv6.
- Supports multiple next hops; and if the next hop isn't reachable, ePBR automatically switches to the next available hop.

- You have the option to configure IP SLA tracking. If this is configured, the next hop is selected only when the IP SLA probe is successful.

  SLA probes can be configured in the same or a different VRF.

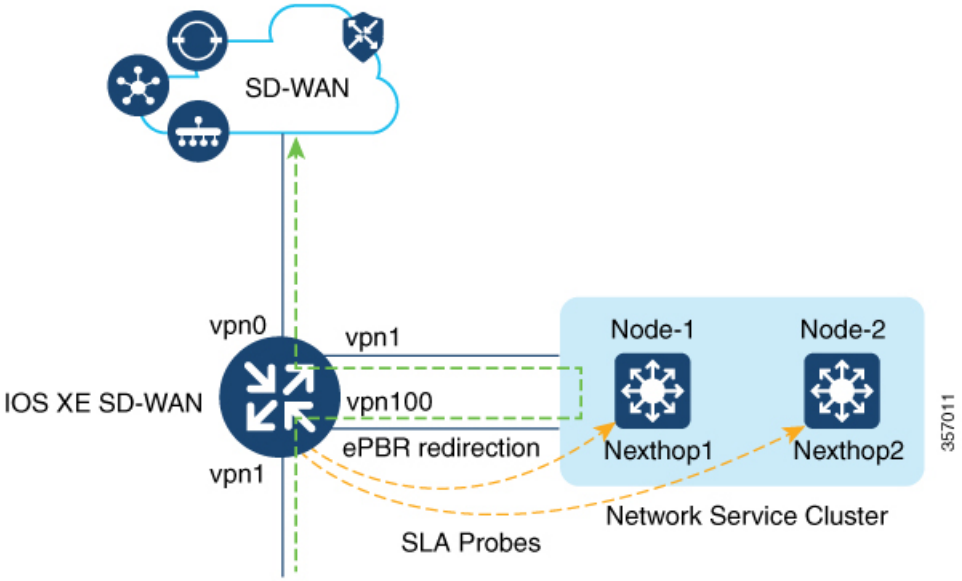
- If the current hop isn't reachable, syslog messages are generated and the user is notified of the same.

How ePBR Works

- ePBR is applicable to unicast routing only and is based on traffic matching using C3PL.
- All packets received on an ePBR-enabled interface are passed through policy maps. The policy maps used by ePBR dictate the policy, determining where to forward packets.
- ePBR policies are based on a classification criteria (match) and an action criteria (set) that are applied to traffic flow.
- To enable ePBR, you must create a policy map that specifies the packet match criteria and desired policy-route action. Then you associate the policy map on the required interface.
- The match criteria is specified in a class. The policy map then calls the class and takes action based on the set statement.
- The set statements in ePBR policies define the route in terms of next hops, DSCP, VRFs, and so on.

Usage Example

Figure 1: Traffic Redirection with ePBR



This example shows that traffic is coming into VPN 1 interface. Based on the classification configured on VPN 1, the traffic overrides the regular route forwarding and is redirected to a next-hop in VPN 100, where additional network services are applied to the incoming traffic. Network services, such as WAN optimization,

are then applied on the redirected traffic before it is forwarded to the Cisco Catalyst SD-WAN overlay network through VPN 0.

Configure ePBR

To configure ePBR using Cisco SD-WAN Manager, create a CLI add-on feature template and attach it to the device template.

This section provides examples of ePBR configurations that you can add to the CLI add-on template.

Configure ePBR for IPv4

In the following example:

- The extended ACLs define the network or the host.
- Class maps match the parameters in the ACLs.
- Policy maps with ePBR then take detailed actions based on the set statements configured.
- Multiple next-hops are configured. ePBR chooses the first available next-hop.

```
ip access-list extended test300
 100 permit ip any 192.0.2.1 0.0.0.255
ip access-list extended test100
 100 permit ip any 192.0.2.20 0.0.0.255
!
class-map match-any test300
 match access-group name test300
class-map match-any test100
 match access-group name test1
!
policy-map type epbr test300
 class test300
  set ipv4 vrf 300 next-hop 10.0.0.2 10.0.40.1 10.0.50.1 ...
policy-map type epbr test100
 class test100
  set ipv4 vrf 100 next-hop 10.10.0.2 10.20.20.2 10.30.30.2 ...
!
interface GigabitEthernet0/0/1
 service-policy type epbr input test300
interface GigabitEthernet0/0/2
 service-policy type epbr input test100
```

Configure IPv4 Tracking

This example shows how to configure ePBR along with tracking. In the example:

- IP SLA operations of type ICMP Echo are configured and ACLs are defined.
- Class maps are then used to match parameters in the ACLs and the policy map takes action based on the set statements configured.
- The number 10 in `set ipv4 vrf 300 next-hop verify-availability 10.10.0.2 10 track 2` represents the sequence number.

```
ip sla 1
```

```
  icmp-echo 10.0.0.2
  vrf 100
ip sla schedule 1 life forever start-time now
track 1 ip sla 1 state
ip sla 2
  icmp-echo 10.10.0.2
  vrf 300
ip sla schedule 2 life forever start-time now
track 2 ip sla 2 state
ip access-list extended test300
 100 permit ip any 10.10.0.2 0.0.0.255
ip access-list extended test100
 100 permit ip any 10.10.0.3 0.0.0.255
class-map match-any test300
 match access-group name test300
class-map match-any test100
 match access-group name test100
policy-map type epbr test300
 class test300
  set ipv4 vrf 300 next-hop verify-availability 10.10.0.2 10 track 2
policy-map type epbr test100
 class test100
  set ipv4 vrf 100 next-hop verify-availability 10.0.0.2 10 track 1
!
interface GigabitEthernet0/0/1
 service-policy type epbr input test300
interface GigabitEthernet0/0/2
 service-policy type epbr input test100
```

Configure ePBR for IPv6

In the following example:

- The extended ACLs define the network or the host.
- Class maps are used to match the parameters in the ACLs.
- Policy maps with ePBR then take detailed actions based on the set statements configured. .
- Single or multiple next-hop addresses can be configured. ePBR selects the first available next-hop address

```
ipv6 access-list test300_v6
  sequence 100 permit ipv6 any 2001:DB81::/32
ipv6 access-list test100_v6
 sequence 100 permit ipv6 any 2001:DB82::/32
!
class-map match-any test300_v6
 match access-group name test300_v6
class-map match-any test100_v6
 match access-group name test100_v6
policy-map type epbr test300_v6
 class test300_v6
  set ipv6 vrf 300 next-hop 2001:DB8::1
policy-map type epbr test100_v6
 class test100_v6
  set ipv6 vrf 100 next-hop 2001:DB8::2 2001:DB8:FFFF:2 ...
!
interface GigabitEthernet0/0/1
 service-policy type epbr input test300_v6
interface GigabitEthernet0/0/2
 service-policy type epbr input test100_v6
```

Configure IPv6 Tracking

This example shows how to configure ePBR for IPv6 along with tracking enabled. In this example:

- IP SLA operations of type ICMP Echo are configured and ACLs are defined.
- Class maps are then used to match parameters in the ACLs and the policy map takes action based on the set statements configured.
- Tracking is configured such that if the result of the IP SLA is unavailable, the packets aren't sent to the next-hop configured on the class.

```
ip sla 3
  icmp-echo 2001:DB8::1
  vrf 100
ip sla schedule 3 life forever start-time now
track 3 ip sla 3 state
ip sla 4
  icmp-echo 2001:DB8::2
  vrf 300
ip sla schedule 4 life forever start-time now
track 4 ip sla 4 state
ipv6 access-list test300_v6
  sequence 100 permit ipv6 any 2001:DB8::/32
ipv6 access-list test100_v6
 sequence 100 permit ipv6 any  2001:DB8::1/32
class-map match-any test300_v6
 match access-group name test300_v6
class-map match-any test100_v6
 match access-group name test100_v6
policy-map type epbr test300_v6
 class test300_v6
  set ipv6 vrf 300 next-hop verify-availability 2001:DB8::2 10 track 4
policy-map type epbr test100_v6
 class test100_v6
  set ipv6 vrf 100 next-hop verify-availability 2001:DB8::1 10 track 3
interface GigabitEthernet0/0/1
 service-policy type epbr input test300_v6
interface GigabitEthernet0/0/2
 service-policy type epbr input test100_v6
```

Configure ePBR for IPv4 with Multiple Next Hops and SLA Tracking

In the following example:

- IP SLA operations of type ICMP Echo are configured and ACLs are defined.
- Class maps are then used to match parameters in the ACLs and the policy map takes action based on the set statements configured.
- Tracking is configured for next hops such that if the previous IP address isn’t reachable, and the IP SLA confirms the next hop as reachable, packets flow to the next hop address.

```
ip sla 1
  icmp-echo 10.0.0.2
  vrf 100
ip sla schedule 1 life forever start-time now
track 1 ip sla 1 state
ip sla 2
  icmp-echo 10.10.0.2
  vrf 300
ip sla schedule 2 life forever start-time now
```

```
track 2 ip sla 2 state
ip sla 3
  icmp-echo 10.20.0.2
  vrf 400
ip sla schedule 3 life forever start-time now
track 3 ip sla 3 state
ip access-list extended test300
 100 permit ip any 192.0.2.1  255.255.255.0
ip access-list extended test100
 100 permit ip any 192.0.2.10 255.255.255.0
!
class-map match-any test300
 match access-group name test300
class-map match-any test100
 match access-group name test100
!
policy-map type epbr test300
 class test300
  set ipv4 vrf 300 next-hop verify-availability 10.10.0.2 10 track 2
  set ipv4 vrf 400 next-hop verify-availability 10.20.0.2 11 track 3
policy-map type epbr test100
 class test100
  set ipv4 vrf 100 next-hop verify-availability 10.0.0.2 10 track 1
!
interface GigabitEthernet0/0/1
 service-policy type epbr input test300
interface GigabitEthernet0/0/2
 service-policy type epbr input test100
!
```

Note When next hops are configured along with the tracker, if the next hop is unreachable or if the IP SLA fails, the next available hop is selected. This means that when the tracker is configured, both next hop availability and IP SLA results are checked.

Monitor ePBR

ePBR can't be monitored through Cisco SD-WAN Manager. To verify your configuration or monitor ePBR statistics, use the show commands described below.

Verify Availability of Next Hop

The following is sample output from the **show platform software epbr track** command.

```
Device# show platform software epbr track
Track Object:
obj num:2:
  track:0x7F94B4376760
   seq:10, nhop:123.0.0.2, nhop_reachable:1, track_handle:0x7F94AFDAE240,
   global:0, vrf_name:300, track_reachable:1
   parent:0x7F94B4383778, oce:0x7F94B81193A8
obj num:1:
  track:0x7F94B8187810
   seq:10, nhop:100.0.0.2, nhop_reachable:1, track_handle:0x7F94AFDAE1D0,
   global:0, vrf_name:100, track_reachable:1
   parent:0x7F94B8187778, oce:0x7F94B81188B8
```

In this example, `nhop_reachable` has the value 1, which indicates that the next hop is reachable. `track_reachable` represents the result of SLA probe and has the value 1, which indicates that the next hop is reachable. If the next hop isn't reachable, the value would be 0 for these parameters.

View Next Hop Configuration

Use the **show platform software epbr R0 feature-object redirect** to view the next hop configuration.

Note To be able to view this output, you must have tracker configured.

```
Device# show platform software epbr r0 feature-object redirect
FMAN EPBR Redirect Feature Objectep

Feature Object ID: 9876543211
  Flags: 0x3
  Table ID: 0x4
  Next-hop: 10.10.10.2
  P2P ADJ-ID: 0

Feature Object ID: 1234567890
  Flags: 0x3
  Table ID: 0x2
  Next-hop: 172.16.0.0
  P2P ADJ-ID: 0
```