SR-TE Manual Preference Selection

Configuring SR-TE Manual Preference Selection

This section describes the configuration and execution commands introduced to support manual preference selection feature.

Guidelines and Limitations for SR-TE Manual Preference Selection

The following guidelines and limitations apply to the SR-TE manual preference selection feature:

- Beginning with Cisco NX-OS Release 10.2(2)F, the SR-TE manual preference selection feature allows you to lockdown, shutdown, or perform both on an SRTE policy or an on-demand color template; shutdown preference(s) of an SR-TE policy or an on-demand color template. Furthermore, this feature also allows you to force a specific preference to be active for the SR-TE policy and force path re-optimization for all or a specific SR-TE policy.

  This feature is supported on Cisco Nexus 9300-EX, 9300-FX, 9300-FX2, 9300-GX, and N9K-C9332D-GX2B platform switches.

- Beginning with Cisco NX-OS Release 10.5(3)F, route-maps can include both PBR and NON-PBR set commands. This change allows for more flexible routing configurations by enabling the integration of policy-based routing commands with traditional routing commands within the same route-map. Users should ensure that the appropriate set commands are configured based on their specific use-case requirements.

About SR-TE Manual Preference – Lockdown and Shutdown

Beginning with Cisco NX-OS Release 10.2(2)F, you can perform the following actions as appropriate:

- Lockdown an SRTE policy – You can enable lockdown under on-demand color templates or explicit policies. Lockdown disables auto re-optimization of path preferences for a policy. In case a new higher preferred path comes up for a policy which is locked down, then it does not automatically switch to use the new path and continues to use the current active path option until it is valid.

Note If an explicit policy configuration exists for the same color as the on-demand template, then the policy configuration takes precedence over the template configuration for the lockdown.

Example

Consider a scenario where there are multiple preferences on a policy. Assume that the higher preference path goes down due to some fault in the network. The fault could be an impending failure of a node in the higher preference path. When investigating and rectifying the fault, the operations team may need to reload or disable the problematic node and prevent any disruptions while this occurs. Then, locking down the lower preference path and preventing switching back to the higher preference path is a good option to use.

- Shutdown an SRTE policy – You can enable shutdown under on-demand color templates or explicit policies. The policy state changes to admin down, and a policy down notification is sent to all the clients interested in the policy. Disabling shutdown under on-demand color configuration changes the policy state to up or down based on the path validity of the policy.

Note If an explicit policy configuration exists for the same color as the on-demand template, then the policy configuration takes precedence over the template configuration for the shutdown.

- Shutdown preference[s] of an SRTE policy – You can shut down a path preference under an on-demand color template configuration or under a path preference of explicit policy configuration. This disables that path preference and stops it from entering any future path re-optimization until the preference is unshut. The path preference is shown as admin down or up in the output of `show srte policy` based on whether it is shut or unshut in the configuration.

Configuring SR-TE Manual Preference – Lockdown/Shutdown

You can configure lockdown, shutdown, or both on an SR-TE policy or an on-demand color template. You can also shutdown a preference under an SR-TE policy or an on-demand color template.

Before you begin

You must ensure that the mpls segment routing feature is enabled.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**segment-routing**	Enters the segment-routing mode.
Step 3	**traffic-engineering**	Enters the traffic engineering mode.

	Command or Action	Purpose
Step 4	**on-demand color***color_num* or **policy** *name*	Enters the on-demand mode to configure the color or configures the SR-TE policy respectively.
Step 5	(optional) [**no**] **lockdown**	Enables lockdown under the on-demand color template or explicit policy configuration. **Note** When an explicit policy configuration exists for the same color as the on-demand template, then the policy configuration takes precedence over the template configuration, and the policy is locked down.
Step 6	(optional) [**no**] **shutdown**	Shuts down any policy created from the on-demand color template or the configured SR-TE policy, as appropriate. **Note** When an explicit policy configuration exists for the same color as the on- demand template, then the policy configuration takes precedence over the template configuration, and the policy is shut down.
Step 7	**candidate-paths**	Specifies the candidate paths of the policy.
Step 8	**preference** *preference_number*	Specifies the preference of the candidate path.
Step 9	(optional) [**no**] **shutdown**	Shuts down a path preference under an SR-TE policy configuration or an on-demand color template configuration.

Force a Specific Path Preference for an SRTE Policy

To force a specific preference to be the active path option for an SRTE policy, use the `segment-routing traffic-engineering switch name <policy_name> pref <preference_number>` execution command. This command uses the preference until it is valid.

A sample output is as follows:

```
NX2# show srte policy Green_White
Policy: 8.8.8.0|801
Name: Green_White
Source: 2.2.2.0
End-point: 8.8.8.0
State: UP
Color: 801
Authorized: Y
Binding-sid Label: 22
```

```
Policy-Id: 3
Path type = MPLS Active path option
Path-option Preference:180 ECMP path count: 1
1. PCE Weighted: No
Delegated PCE: 11.11.11.11
Index: 1 Label: 16005
Index: 2 Label: 16008
NX2# segment-routing traffic-engineering switch name Green_White preference 170
NX2(cfg-pref)# show srte policy Green_white detail
Policy: 8.8.8.0|801
Name: Green_White
…..
Path type = MPLS Path options count: 4
Path-option Preference:180 ECMP path count: 1 Admin: UP Forced: No
1. PCE Weighted: No
Delegated PCE: 11.11.11.11
Index: 1 Label: 16005
Index: 2 Label: 16008
Path-option Preference:170 ECMP path count: 1 Admin: UP Forced: Yes Active path option
1. Explicit Weighted: No
Name: Yellow
Index: 1 Label: 16006
Index: 2 Label: 16008
```

To undo this manually selected preference, you can perform any one of the following options:

- Use the `segment-routing traffic-engineering reoptimize name <policy_name>` command. For more information, see the Force path re-optimization for an SRTE Policy or All SRTE Policies, on page 4 section.
- Switch to another preference
- Shut this policy
- Shut the selected preference

Force path re-optimization for an SRTE Policy or All SRTE Policies

When there are multiple preferences for an SRTE policy, you can re-optimize a policy, that is, pick the best preferred available path.

To force path re-optimization for a specific SRTE policy, use the `segment-routing traffic-engineering reoptimize name <policy_name>` command. The `<policy_name>` can be the name or alias name of the policy. This command undoes the preference switch command explained in the previous section and overrides lockdown if configured.

A sample output is as follows:

```
NX2# show srte policy Green_White
Policy: 8.8.8.0|801
Name: Green_White
Source: 2.2.2.0
End-point: 8.8.8.0
State: UP
Color: 801
Authorized: Y
Binding-sid Label: 22
Policy-Id: 3
Path type = MPLS Active path option
Path-option Preference:170 ECMP path count: 1
1. Explicit Weighted: Yes Weight: 1
```

```
Name: Yellow
Index: 1 Label: 16006
Index: 2 Label: 16008
NX2# segment-routing traffic-engineering reoptimize name Green_White
NX2# show srte policy Green_White
Policy: 8.8.8.0|801
Name: Green_White
Source: 2.2.2.0
End-point: 8.8.8.0
State: UP
Color: 801
Authorized: Y
Binding-sid Label: 22
Policy-Id: 3
Path type = MPLS Active path option
Path-option Preference:180 ECMP path count: 1
1. PCE Weighted: No
Delegated PCE: 11.11.11.11
Index: 1 Label: 16005
Index: 2 Label: 16008
```

To force path re-optimization for all SRTE policies, use the `segment-routing traffic-engineering reoptimize all` command to force path re-optimization for all SRTE policies present on the system. This command undoes the preference switch command explained in the previous point and overrides lockdown if configured.

Configuring SRTE Flow-based Traffic Steering

This chapter describes how to configure SRTE flow-based traffic steering on Cisco Nexus 9000-FX, 9000-FX2, 9000-FX3, 9000-GX, and 9300 platform switches.

About SRTE Flow-based Traffic Steering

The Flow-based Traffic Steering feature for Cisco NX-OS release 10.1(2) provides an alternate method of choosing the traffic to be steered, which is direct and flexible. This method allows configuring the source routing on the headend node directly, rather than on the egress node. Flow-based traffic steering allows the user to select which packets will be steered into an SRTE policy by matching fields in the incoming packet such as destination address, UDP or TCP port, DSCP bits and other properties. The matching is done by programming an ACL to steer packets into the policy.

To match and steer traffic, the Policy-Based Routing (PBR) feature is enhanced to support SRTE policies. The current PBR feature involves the RPM, ACL Manager, and AclQoS components. Beginning from Cisco NX-OS release 10.1(2), to add SRTE support, the RPM component also communicates with the SRTE and ULIB, and the communication with URIB is enhanced.

Thus, the flow-based traffic steering feature for SRTE includes the following:

- MPLS SR dataplane
- Steering IPv4 traffic is supported in default VRF, and steering IPv4 as well as IPv6 traffic is supported in non-default VRF
- Matching traffic by ACL based on a combination of the 5 tuple fields (source addr, destination addr, protocol, tcp/udp source port, tcp/udp destination port)
- Steering matched traffic into an SRTE policy

- Matching on the DSCP/TOS bits in the packet for IPv4 packets. Beginning with Cisco NX-OS Release 10.3(1)F, matching on the DSCP/TOS bits for the outer header for the VXLAN packets is also supported.
- Matching on the Traffic Class field of the packet for IPv6 packets
- Automatic enabling and disabling of ACLs based on time-period definitions
- When steering VRF cases, support steering into an SRTE policy without specifying a next hop
- Overlay ECMP using anycast endpoints
- Packets matched by ACL take precedence over regular routes
- Flow selection based on ToS/DSCP and timer-based ACL
- The next-hop-ip is used in steering traffic to SRTE policy from one endpoint to another

Guidelines and Limitations for Flow-based Traffic Steering for SRTE

The following guidelines and limitations apply to the Flow-based Traffic Steering for SRTE feature:

- Beginning with Cisco NX-OS release 10.1(2), the flow-based traffic steering features for SRTE are supported on the Cisco Nexus 9000-FX, 9000-FX2, 9000-FX3, 9000-GX, and 9300 platform switches.
- When the SRTE policy is applied to a route-map assigned to an interface in a VRF (to steer L3VPN/L3EVPN traffic), if the next hop in the set statement resolves to a BGP prefix, and that BGP prefix is already using an SRTE policy to steer traffic, then the route-map does not steer traffic.
- Underlay ECMP is only supported if label stack is the same for each active SRTE path (ECMP member) in the policy. The 9000-GX platforms do not have this limitation.
- The route-map tracking feature is not supported.
- When steering into SRTE policies, having multiple **set next-hop** in a single route-map sequence entry is not supported.
- When the SRTE policy is applied to a route-map assigned to an interface in a VRF (to steer L3VPN/L3EVPN traffic), if the next hop in the set statement resolves to a BGP route (overlay route) that has multiple next hops in RIB, the traffic is only steered to the first next hop in the route and will not ECMP over all next hops.
- When the SRTE policy name is used in the route-map set statement, rather than color and endpoint, it can only be used for default VRF steering. If not, you must select an SRTE path that is defined explicitly. Specifically, this cannot be used to select SRTE policies defined to use a segment-list containing the policy-endpoint keyword in place of a label.
- The following keywords, which are applicable for the next hop-ip specified in the **set ip next-hop** <>, are not supported in the route-map when steering into SRTE policies:
  - verify-availability
  - drop-on-fail
  - force-order
  - load-share

- Route-map with srte-policy can be applied on the interface even if the required features (segmenting-routing, l3 evpn or l3vpn) are not enabled on the device. But the set-actions with srte-policy are kept down, that is, default-routing will be done for those flows.
- A route-map can have set commands with srte-policy and without srte-policy.
- For set-commands without srte-policy information, steering is done only if the reachability to the next-hop-ip does not require MPLS label.
- When a route-map is associated with an interface in a non-default VRF, and that route-map contains a sequence that specifies a next hop IP address **N** and an SRTE policy, then all other sequences on that route-map and all other route-maps associated with the same VRF that also use the same next hop IP address must also have an SRTE policy. Associating another route-map or route-map sequence using the same next hop IP and a different SRTE policy to the same VRF is not allowed.
- Similarly, when a route-map is associated with an interface in a non-default VRF, and that route-map does not specify an SRTE policy but specifies a next hop IP address **N**, then another sequence in that route-map or a separate route-map is not applied that uses the same next hop IP address N and specifies an SRTE policy.
- The SRTE flow-based traffic steering cannot be used at the same time as VXLAN or EoMPLS PBR.
- The SR label stats are not supported for policy based routed traffic on the SRTE ingress node. However, ACL redirect stats are supported.
- The IPv6 traffic in the default VRF cannot be steered into an SRTE policy. The MPLS SR underlay is only supported for IPv4. However, if an IPv6 SR underlay is required, use SRv6 instead.
- The 9000-FX, 9000-FX2, 9000-FX3, and 9300 platform hardware are unable to push unique underlay label stack per ECMP member, which impacts underlay ECMP on those platforms. In other words, if there are multiple active segment-lists on an SRTE policy (a single preference is configured with multiple segment-lists) where the first hop of the segment lists is different, then such a configuration is not supported. In such cases, as a workaround, configure anycast SID to make the label stack same across all ECMP members.
- Modular platforms are not supported in Cisco NX-OS release 10.1(2).
- Beginning with Cisco NX-OS release 10.2(2)F, the flow-based traffic steering features for SRTE are supported on the Cisco N9K-C9332D-GX2B platform switches.
- Beginning with Cisco NX-OS Release 10.3(1)F, the DSCP Based SR-TE Flow Steering feature allows source routing of VXLAN packets that are matched using the DSCP fields in the IP header and steered into an SRTE path. Following are the guidelines and limitations for this feature:
  - This feature is supported only on the Cisco Nexus 9300-FX2, 9300-FX3, 9300-GX, 9300-GX2 TOR switches.
  - In cases where the VXLAN packets are not terminated, the ACL filters are applied on the VXLAN Packet Outer IP Header fields (IPv4).
- Beginning with Cisco NX-OS release 10.3(2)F, the flow-based traffic steering features for SRTE are supported on Cisco Nexus 9700-FX and 9700-GX line cards. Following are the guidelines and limitations for this feature:
  - When Cisco Nexus 9508 platform switch is in a VXLAN EVPN to MPLS SR L3VPN hand-off mode and the MPLS encapsulated packets are forwarded on an L2 port, the dot1q header does not get added.

- SVI/Sub-interfaces are not supported for core facing uplinks (MPLS or VXLAN) when Cisco Nexus 9500 platform switch is configured as EVPN to MPLS SR L3VPN hand off mode.
- The DSCP to MPLS EXP promotion does not work on the FX TOR/line cards in DCI Mode. The copying of Inner DSCP value to MPLS EXP does not work on the FX TOR/line cards in this hand off mode. The MPLS EXP will be set to 0x7.

- Beginning with Cisco NX-OS Release 10.3(2)F, the DSCP based SR-TE flow steering feature is supported on Cisco Nexus 9300-FX platform switches and Cisco Nexus 9700-FX and 9700-GX line cards. Following are the guidelines and limitations for this feature:
  - When Cisco Nexus 9500 platform switch is in a VXLAN EVPN to MPLS SR L3VPN hand-off mode and the MPLS encapsulated packets are forwarded on an L2 port, the dot1q header does not get added.
  - SVI/Sub-interfaces are not supported for core facing uplinks (MPLS or VXLAN) when Cisco Nexus 9500 platform switch is configured as EVPN to MPLS SR L3VPN hand off mode.
  - The DSCP to MPLS EXP promotion does not work on the FX TOR/line cards in DCI Mode. The copying of Inner DSCP value to MPLS EXP does not work on the FX TOR/line cards in this hand off mode. The MPLS EXP will be set to 0x7.

Configuration Process: SRTE Flow-based Traffic Steering

The configuration process for the SRTE Flow-based Traffic Steering feature is as follows:

1. Configure the IP access lists, especially matching the criteria on the IP access list.

   For more information, see the *Configuring IP ACLs* chapter in the *Cisco Nexus Series NX-OS Security Configuration Guide*.

2. Define the SRTE policy.

   For more information about configuring SRTE, see the *Configuring Segment Routing for Traffic Engineering* chapter in the *Cisco Nexus 9000 series NX-OS Label Switching Configuration Guide*.

3. Configure the route map that binds the match (IP access list configured in step 1) and action. The match refers to the fields to match on the packet, and the action refers to what SRTE policy to steer into and the VPN label to use, if any.

Configuring Flow Selection Based on ToS/DSCP and Timer-based ACL

In the SRTE flow-based traffic steering feature, the flow selection is based on ToS/DSCP and Timer based ACL.

Perform the following configuration procedure for the route map configuration in default and non-default VRF into a policy selected by different criteria to work properly.

Before you begin

You must ensure that the MPLS segment routing traffic engineering and PBR features are enabled.

Procedure

	Command or Action	Purpose

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

Step 2	[**ip** | **ipv6**] **access-list** *acl_name*
	Example:
	`switch(config)# ip access-list L4_PORT`
	`switch(config)#`
		Use a name to define an IP or IPv6 access list and enter the IP or IPv6 access-list configuration mode.

Step 3	**10 permit ip** *ip_address* **any**
	Example:
	`switch(config)# 10 permit ip any 5.5.0.0/16`
	`switch(config)#`
		Shows the IP or IPv6 access lists configured on the switch.

Step 4	**20 permit tcp** *tcp_address* [**any**]
	Example:
	`switch(config)# 20 permit tcp any 5.5.0.0/16`
	`switch(config)#`
		Sets TCP permit conditions for an IP or IPv6 access list.
		Note
		The **any** keyword is used for IPv6 only.

Step 5	[**ip** | **ipv6**] **access-list** *dscp_name*
	Example:
	`switch(config)# ip access-list dscp`
	`switch(config)#`
		Use a name to define DSCP for an IP or IPv6 access list and enter the IP or IPv6 access-list configuration mode.

Step 6	**10 permit tcp any** *tcp_address* **dscp** <*dscp value*>
	Example:
	`switch(config)# 10 permit tcp any 5.5.0.0/16 dscp af11`
	`switch(config)#`
		Set the DSCP value for an IP or IPv6 access list.
		Note
		The **any** keyword is used for IPv6 only.

Step 7	[**ip** | **ipv6**] **access-list** *acl_name*
	Example:
	`switch(config)# ip access-list acll`
	`switch(config)#`
		Use a name to define an IP or IPv6 access list and enter the IP or IPv6 access-list configuration mode.

Step 8	**10 permit tcp any** *tcp_address* **acl** *acl_name*
	Example:
	`switch(config)# 10 permit tcp any 5.5.0.0/16 eq www dscp af11`
	`switch(config)#`
		Sets TCP permit conditions for an IP or IPv6 access list.
		Note
		The **any** keyword is used for IPv6 only.

	Command or Action	Purpose

Step 9	[**ip** | **ipv6**] **access-list** *acl_name*
	Example:
	`switch(config)# ip access-list acl1`
	`switch(config)#`
		Use a name to define an IP or IPv6 access list and enter the IP or IPv6 access-list configuration mode.

Step 10	**10 permit tcp any** *any time - range tl*
	Example:
	`switch(config-acl)# 10 permit tcp any 5.5.0.0/16 eq www dscp af11`
	`switch(config)#`
		Sets time range value to define the time range for TCP for an IP or IPv6 access list.
		Note
		The **any** keyword is used for IPv6 only.

Step 11	**time-range** *name*
	Example:
	`switch(config-acl)# time-range t1`
	`switch(config)#`
		Use a name to define the time range for an IP or IPv6 access list.

Step 12	**F2(config-time-range)#**
	WOLF2(config-time-range)#
	Example:
	`switch(config-time-range)# 10 absolute start 20:06:56 8 february 2021`
	`end 20:10:56 8 february 2021`
		Define a time range for the configuration.

Configuring Route Map in Default and Non-default VRF for Flow-based Traffic Steering

The following sections show how to configure the route map in default and non-default VRF for the SRTE flow-based traffic steering feature:

Configuring Route Map in Default VRF into a Policy Selected by Color and Endpoint

Perform the following steps to configure a route-map that steers traffic in the default VRF into a policy that is selected by color and endpoint.

Before you begin

You must ensure that the MPLS segment routing traffic engineering and PBR features are enabled.

Procedure

	Command or Action	Purpose

Step 1	**route-map FLOW1** *seq_num*
	Example:
	`switch(config)# route-map FLOW1 seq 10`
	`switch(config-route-map)#`
		Names the route map FLOW1.

Step 2	**match** [**ip** | **ipv6**] **address** *acl_name*
	Example:
	`switch(config-route-map)# match ip address L4_PORT`
	`switch(config-route-map)#`
		Specifies the fields the route-map should match by attaching an ACL describing the fields.

Step 3	**set srte-policy color** *num* **endpoint** *ip address*
	Example:
	`switch(config-route-map)# set srte-policy color 121 endpoint 10.0.0.1`
	`switch(config-route-map)#`
		Configures the SRTE policy color and the end point of the policy.
		Note
		Only IPv4 address can be the endpoint.

Step 4	**interface** *interface-type/slot/port*
	Example:
	`switch(config-route-map)# interface ethernet 1/1`
	`switch(config-route-map-if)#`
		Enters interface configuration mode.

Step 5	[**ip** | **ipv6**] **policy route-map FLOW1**
	Example:
	`switch(config-route-map-if)# ip policy route-map FLOW1`
	`switch(config-route-map-if)#`
		Assigns a route map for IP or IPv6 policy-based routing to the interface. This applies the route-map for all the traffic ingressing into the interface.

Configuring Route Map in Default VRF into a Policy Selected by Name

Perform the following steps to configure a route-map that steers traffic in the default VRF into a policy that is selected by name.

Before you begin

You must ensure that the MPLS segment routing traffic engineering and PBR features are enabled.

Procedure

	Command or Action	Purpose
Step 1	**route-map FLOW1** *seq_num* **Example:**	Names the route map FLOW1.

	Command or Action	Purpose
	`switch(config)# route-map FLOW1 seq 10` `switch(config-route-map)#`	

Step 2	**match** [**ip** | **ipv6**] **address** *acl_name*
	Example:
	`switch(config-route-map)# match ip address L4_PORT`
	`switch(config-route-map)#`
		Specifies the fields the route-map should match by attaching an ACL describing the fields.

Step 3	**set srte-policy name** *policy-name*
	Example:
	`switch(config-route-map)# set srte-policy name policy1`
	`switch(config-route-map)#`
		Configures the SRTE policy name.

Step 4	**interface** *interface-type/slot/port*
	Example:
	`switch(config-route-map)# interface ethernet 1/1`
	`switch(config-route-map-if)#`
		Enters the interface configuration mode.

Step 5	[**ip** | **ipv6**] **policy route-map FLOW1**
	Example:
	`switch(config-route-map-if)# ip policy route-map FLOW1`
	`switch(config-route-map-if)#`
		Assigns a route map for IP or IPv6 policy-based routing to the interface. This applies the route-map for all the traffic ingressing into the interface.

Configuring Route Map in Non-default VRF into a Policy Selected by Nexthop, Color, and Endpoint

Perform the following steps to configure a route-map that steers traffic in a non-default VRF into a policy that is selected by color and endpoint. In this procedure, a nexthop is specified so that the correct MPLS VPN label is imposed on the traffic.

Before you begin

You must ensure that the MPLS segment routing traffic engineering and PBR features are enabled.

Procedure

	Command or Action	Purpose

Step 1	**route-map FLOW1** *seq_num*
	Example:
	`switch(config)# route-map FLOW1 seq 10`
	`switch(config-route-map)#`
		Names the route map FLOW1.

	Command or Action	Purpose

Step 2	**match** [**ip** | **ipv6**] **address** *acl_name*
	Example:
	`switch(config-route-map)# match ip address L4_PORT`
	`switch(config-route-map)#`
		Specifies the fields the route-map should match by attaching an ACL describing the fields.

Step 3	**set** [**ip** | **ipv6**] **next-hop** *destination-ip-next-hop* **srte-policy color** *num* **endpoint** *ip address*
	Example:
	`switch(config-route-map)# set ip next-hop 5.5.5.5 srte-policy color 121 endpoint 10.0.0.1`
	`switch(config-route-map)#`
		Redirects packet to the configured next-hop through the srte-policy (color and endpoint).

Step 4	**exit**
	Example:
	`switch(config-route-map)# exit`
	`switch(config)#`
		Exits the route-map configuration mode and returns to the global configuration mode.

Step 5	**interface** *interface-type/slot/port*
	Example:
	`switch(config)# interface ethernet 1/1`
	`switch(config-if)#`
		Enters the interface configuration mode.

Step 6	**vrf member** *vrf-name*
	Example:
	`switch(config-if)# vrf member vrf1`
	`switch(config-if)#`
		Adds this interface to a VRF.

Step 7	[**ip** | **ipv6**] **policy route-map FLOW1**
	Example:
	`switch(config-if)# ip policy route-map FLOW1`
	`switch(config-if)#`
		Assigns a route map for IP or IPv6 policy-based routing to the interface. This applies the route-map for all the traffic ingressing into the interface.

Step 8	[**no**] **shutdown**
	Example:
	`switch(config-if)# no shutdown`
	`switch(config-if)#`
		Disables the interface.

Configuring Route Map in Non-default VRF into a Policy Selected by Nexthop and Color

Perform the following steps to configure a route-map that steers traffic in the default VRF into a policy that is selected by color and endpoint, but the endpoint is not explicitly configured. The nexthop is specified so that the correct MPLS VPN label is imposed on the traffic and so the correct SRTE endpoint is derived from the route matching the nexthop.

Before you begin

You must ensure that the MPLS segment routing traffic engineering and PBR features are enabled.

Procedure

	Command or Action	Purpose

Step 1	**route-map FLOW1** *seq_num*
	Example:
	`switch(config)# route-map FLOW1 seq 10`
	`switch(config-route-map)#`
		Names the route map FLOW1.

Step 2	**match** [**ip** | **ipv6**] **address** *acl_name*
	Example:
	`switch(config-route-map)# match ip address L4_PORT`
	`switch(config-route-map)#`
		Specifies the fields the route-map should match by attaching an ACL describing the fields.

Step 3	**set** [**ip** | **ipv6**] **next-hop** *destination-ip-next-hop* **srte-policy color** *num*
	Example:
	`switch(config-route-map)# set ip next-hop 5.5.5.5 srte-policy color 121`
	`switch(config-route-map)#`
		Redirects packet to the configured next-hop through the srte-policy (color).

Step 4	**exit**
	Example:
	`switch(config-route-map)# exit`
	`switch(config)#`
		Exits the route-map configuration mode and returns to the global configuration mode.

Step 5	**interface** *interface-type/slot/port*
	Example:
	`switch(config)# interface ethernet 1/1`
	`switch(config-if)#`
		Enters the interface configuration mode.

Step 6	**vrf member** *vrf-name*
	Example:
	`switch(config-if)# vrf member vrf1`
	`switch(config-if)#`
		Adds this interface to a VRF.

Step 7	[**ip** | **ipv6**] **policy route-map FLOW1**
	Example:
	`switch(config-if)# ip policy route-map FLOW1`
	`switch(config-if-route-map)#`
		Assigns a route map for IP or IPv6 policy-based routing to the interface. This applies the route-map for all the traffic ingressing into the interface.

Step 8	[**no**] **shutdown** **Example:**	Disables the interface.

	Command or Action	Purpose
	`switch(config-if-route-map)# no shutdown` `switch(config-if-route-map)#`	

Configuring Route Map in Non-default VRF into a Policy Selected by Nexthop and Name

Perform the following steps to configure a route-map that steers traffic in a non-default VRF into a policy that is selected by name. The nexthop is specified so that the correct MPLS VPN label is imposed on the traffic

Before you begin

You must ensure that the MPLS segment routing traffic engineering and PBR features are enabled.

Procedure

	Command or Action	Purpose

Step 1	**route-map FLOW1** *seq_num*
	Example:
	`switch(config)# route-map FLOW1 seq 10`
	`switch(config-route-map)#`
		Names the route map FLOW1.

Step 2	**match** [**ip** | **ipv6**] **address** *acl_name*
	Example:
	`switch(config-route-map)# match ip address L4_PORT`
	`switch(config-route-map)#`
		Specifies the fields the route-map should match by attaching an ACL describing the fields.

Step 3	**set** [**ip** | **ipv6**] **next-hop** *destination-ip-next-hop* **srte-policy** *name*
	Example:
	`switch(config-route-map)# set ip next-hop 5.5.5.5 srte-policy policy1`
	`switch(config-route-map)#`
		Redirects packet to the configured next-hop through the srte-policy (name).

Step 4	**exit**
	Example:
	`switch(config-route-map)# exit`
	`switch(config)#`
		Exits the route-map configuration mode and returns to the global configuration mode.

Step 5	**interface** *interface-type/slot/port*
	Example:
	`switch(config)# interface ethernet 1/1`
	`switch(config-if)#`
		Enters the interface configuration mode.

	Command or Action	Purpose

Step 6	**vrf member** *vrf-name*
	Example:
	`switch(config-if)# vrf member vrf1`
	`switch(config-if)#`
		Adds this interface to a VRF.

Step 7	[**ip** | **ipv6**] **policy route-map FLOW1**
	Example:
	`switch(config-if)# ip policy route-map FLOW1`
	`switch(config-if)#`
		Assigns a route map for IP or IPv6 policy-based routing to the interface. This applies the route-map for all the traffic ingressing into the interface.

Step 8	[**no**] **shutdown**
	Example:
	`switch(config-if)# no shutdown`
	`switch(config-if)#`
		Disables the interface.

Configuring Route Map in Non-default VRF into a Policy Selected by Color and Endpoint

Perform the following steps to configure a route-map that steers traffic in a non-default VRF into a policy that is selected by color and endpoint. This procedure does not require a nexthop to be specified. The VPN label is derived by looking up the label assigned to the VRF on the local switch. This is only allowed to be configured when the same label is assigned to the VRF on all switches by using the BGP allocate-index configuration for the VRF on all switches.

Before you begin

You must ensure that the MPLS segment routing traffic engineering and PBR features are enabled.

Procedure

	Command or Action	Purpose

Step 1	**route-map FLOW1** *seq_num*
	Example:
	`switch(config)# route-map FLOW1 seq 10`
	`switch(config-route-map)#`
		Names the route map FLOW1.

Step 2	**match** [**ip** | **ipv6**] **address** *acl_name*
	Example:
	`switch(config-route-map)# match ip address L4_PORT`
	`switch(config-route-map)#`
		Specifies the fields the route-map should match by attaching an ACL describing the fields.

Step 3	**set srte-policy color** *num* **endpoint** *ip address* **Example:**	Configures the SRTE policy color and the end point of the policy.

	Command or Action	Purpose
	`switch(config-route-map)# set srte-policy color 121 endpoint 10.0.0.1 switch(config-route-map)#`	**Note** Only IPv4 address can be the endpoint.
Step 4	**interface** *interface-type/slot/port* **Example:** `switch(config-route-map)# interface ethernet 1/1 switch(config-route-map-if)#`	Enters the interface configuration mode.
Step 5	**vrf member** *vrf-name* **Example:** `switch(config-route-map-if)# vrf member vrf1 switch(config-route-map-if)#`	Adds this interface to a VRF.
Step 6	[**ip** | **ipv6**] **policy route-map FLOW1** **Example:** `switch(config-route-map-if)# ip policy route-map FLOW1 switch(config-route-map-if)#`	Assigns a route map for IP or IPv6 policy-based routing to the interface. This applies the route-map for all the traffic ingressing into the interface.
Step 7	[**no**] **shutdown** **Example:** `switch(config-route-map-if)# no shutdown switch(config-route-map-if)#`	Disables the interface.
Step 8	**exit** **Example:** `switch(config-route-map)# exit switch(config)#`	Exits the route-map configuration mode and returns to the global configuration mode.
Step 9	**feature bgp** **Example:** `switch(config)# feature bgp switch(config)#`	Enters the BGP feature.
Step 10	**router bgp as-number** **Example:** `switch(config)# router bgp 1.1 switch(config-router)#`	Configures a BGP routing process and enters router configuration mode.
Step 11	**vrf** *vrf-name* **Example:** `switch(config-router)# vrf vrf1 switch(config-router-vrf)#`	Associates the BGP process with a VRF.
Step 12	**allocate-index** *index* **Example:**	Assigns an index to the VRF. This instructs BGP to allocate a static MPLS local VPN label

	Command or Action	Purpose
	`switch(config-router-vrf)# allocate-index 10`	for the VRF. The MPLS VPN label assigned to the VRF is derived from the value specified - the index is used as an offset into a special range of MPLS label values. For a given index value the same local label is always allocated.

Configuring Route Map in Non-default VRF into a Policy Selected by Name

Perform the following steps to configure a route-map that steers traffic in a non-default VRF into a policy that is selected by name. This procedure does not require a nexthop to be specified. The VPN label is derived by looking up the label assigned to the VRF on the local switch. This is only allowed to be configured when the same label is assigned to the VRF on all switches by using the BGP allocate-index configuration for the VRF on all switches.

Before you begin

You must ensure that the MPLS segment routing traffic engineering and PBR features are enabled.

Procedure

	Command or Action	Purpose

Step 1	**route-map FLOW1** *seq_num*
	Example:
	`switch(config)# route-map FLOW1 seq 10`
	`switch(config-route-map)#`
		Names the route map FLOW1.

Step 2	**match** [**ip** | **ipv6**] **address** *acl_name*
	Example:
	`switch(config-route-map)# match ip address L4_PORT`
	`switch(config-route-map)#`
		Specifies the fields the route-map should match by attaching an ACL describing the fields.

Step 3	**set srte-policy** *name*
	Example:
	`switch(config-route-map)# set srte-policy policy1`
	`switch(config-route-map)#`
		Configures the SRTE policy name.

Step 4	**interface** *interface-type/slot/port*
	Example:
	`switch(config-route-map)# interface ethernet 1/1`
	`switch(config-route-map-if)#`
		Enters interface configuration mode.

Step 5	**vrf member** *vrf-name* **Example:**	Adds this interface to a VRF.

	Command or Action	Purpose
	`switch(config-route-map-if)# vrf member vrf1` `switch(config-route-map-if)#`	

Step 6	**[ip | ipv6] policy route-map FLOW1**
	Example:
	`switch(config-route-map-if)# ip policy route-map FLOW1`
	`switch(config-route-map-if)#`
		Assigns a route map for IP or IPv6 policy-based routing to the interface. This applies the route-map for all the traffic ingressing into the interface.

Step 7	**[no] shutdown**
	Example:
	`switch(config-route-map-if)# no shutdown`
	`switch(config-route-map-if)#`
		Disables the interface.

Step 8	**exit**
	Example:
	`switch(config-route-map)# exit`
	`switch(config)#`
		Exits the route-map configuration mode and returns to the global configuration mode.

Step 9	**feature bgp**
	Example:
	`switch(config)# feature bgp`
	`switch(config)#`
		Enters the BGP feature.

Step 10	**router bgp as-number**
	Example:
	`switch(config)# router bgp 1.1`
	`switch(config-router)#`
		Configures a BGP routing process and enters router configuration mode.

Step 11	**vrf** *vrf-name*
	Example:
	`switch(config-router)# vrf vrf1`
	`switch(config-router-vrf)#`
		Associates the BGP process with a VRF.

Step 12	**allocate-index** *index* **Example:** `switch(config-router-vrf)# allocate-index 10`	Assigns an index to the VRF. This instructs BGP to allocate a static MPLS local VPN label for the VRF. The MPLS VPN label assigned to the VRF is derived from the value specified - the index is used as an offset into a special range of MPLS label values. For a given index value the same local label is always allocated.