Cisco Nexus 3000 Series NX-OS Label Switching Configuration Guide, Release 10.1(x)

First Published: 2019-07-20

Last Modified: 2021-09-14

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 the user supplies the values.
[x]	Square brackets enclose an optional element (keyword or argument).
[x | y]	Square brackets enclosing keywords or arguments separated by a vertical bar indicate an optional choice.
{x | y}	Braces enclosing keywords or arguments 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 will include 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 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 3000 Series Switches

The entire Cisco Nexus 3000 Series switch documentation set is available at the following URL:

https://www.cisco.com/c/en/us/support/switches/nexus-3000-series-switches/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 nexus3k-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 Marketplace.
- 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 web-based tool that acts as a gateway to the Cisco bug tracking system that maintains a comprehensive list of defects and vulnerabilities in Cisco products and software. BST provides you with detailed defect information about your products and software.

CHAPTER 1

New and Changed Information

This chapter provides release-specific information for each new and changed feature in the *Cisco Nexus 3000 Series NX-OS Label Switching Configuration Guide, Release 10.1(x)*.

New and Changed Information

This table summarizes the new and changed features for the *Cisco Nexus 3000 Series NX-OS Label Switching Configuration Guide, Release 10.1(x)* and tells you where they are documented.

Table 1: New and Changed Features for Cisco NX-OS Release 10.1(x)

Feature	Description	Changed in Release	Where Documented
No updates since Cisco NX-OS Release 9.3(1)	First 10.1(x) release	Not applicable	Not applicable

New and Changed Information

CHAPTER 2

Platform Support for Label Switching Features

This chapter defines platform support for features that are not supported across the entire suite of Cisco Platforms.

Platform Support for Label Switching Features

The following tables list the supported platforms for each feature and the release in which they were first introduced. See the Release Notes for details about the platforms supported in the initial product release.

Static MPLS

Return to Configuring Static MPLS, on page 5.

Feature	Supported Platform(s) or Line Cards	First Supported Release	Platform Exceptions
Adjacency statistics	Cisco Nexus 3100-V platform switches	7.0(3)F3(1)	Cisco Nexus 3000 platform switches
Static MPLS	Cisco Nexus 3000 platform switches	7.0(3)I2(1)	None

MPLS Label Impostion

Return to Configuring MPLS Label Imposition, on page 19 .

Feature	Supported Platform(s) or Line Cards	First Supported Release	Platform Exceptions
MPLS Label Imposition	Cisco Nexus 3000-XL switches	7.0(3)I5(2)	None

Segment Routing

Return to Configuring Segment Routing, on page 31.

Feature	Supported Platform(s) or Line Cards	First Supported Release	Platform Exceptions
Adjacency SIDs on OSPFv2	Cisco Nexus 3100 Series switches	9.3(1)	None
BGP EVPN and Label Allocation Mode	Cisco Nexus 3100 Series switches	7.0(3)I6(1)	None
BGP Egress Peer Engineering	Cisco Nexus 3100 Series switches	7.0(3)I5(1)	None
BGP Link State Distribution	Cisco Nexus 3100 Series switches	7.0(3)I5(1)	None
IS-IS Segment Routing	Cisco Nexus 3172 Series switches Cisco Nexus 3132 Series switches Cisco Nexus C31128PQ-10GE Series switches Cisco Nexus N3K-C3164Q-40GE switches	7.0(3)I7(3)	None
Segment Routing	Cisco Nexus 3100 Series switches	7.0(3)I4(1)	None
Segment routing and SR-EVPN	Cisco Nexus C31108PC-V switches Cisco Nexus C31108TC-V switches Cisco Nexus C3132Q-V switches	7.0(3)I7(1)	None
Segment Routing Application (SR-APP)	Cisco Nexus 3100 Series switches	7.0(3)I7(3)	None
Segment Routing (command change from segment-routing mpls to segment-routing)	Cisco Nexus 3100 Series switches	9.3(1)	None

MPLS Segment Routing OAM

Return to Configuring MPLS Segment Routing OAM, on page 65.

Feature	Supported Platform(s) or Line Cards	First Supported Release	Platform Exceptions
MPLS OAM Nil FEC	Cisco Nexus 3000 platform switches	7.0(3)I6(1)	None
MPLS OAM BGP and IGP Prefix SID	Cisco Nexus 3000 platform switches	9.3(1)	None

CHAPTER 3

Configuring Static MPLS

This chapter contains information about configuring static MPLS.

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*.

About Static MPLS

Generally, label switching routers (LSRs) use a label distribution protocol to dynamically learn the labels that they should use to label-switch packets. Examples of such protocols include:

- Label Distribution Protocol (LDP), the Internet Engineering Task Force (IETF) standard that is used to bind labels to network addresses
- Resource Reservation Protocol (RSVP), which is used to distribute labels for traffic engineering (TE)
- Border Gateway Protocol (BGP), which is used to distribute labels for MPLS virtual private networks (VPNs)

To use a learned label to label-switch packets, an LSR installs the label into its Label Forwarding Information Base (LFIB).

The static MPLS feature enables you to statically configure the following:

- The binding between a label and an IPv4 or IPv6 prefix
- The action corresponding to the binding between a label and an IPv4 or IPv6 prefix (label swap or pop)
- The contents of an LFIB cross-connect entry

Label Swap and Pop

As a labeled packet traverses the MPLS domain, the outermost label of the label stack is examined at each hop. Depending on the contents of the label, a swap or pop (dispose) operation is performed on the label stack. Forwarding decisions are made by performing an MPLS table lookup for the label carried in the packet header. The packet header does not need to be reevaluated during packet transit through the network. Because the label has a fixed length and is unstructured, the MPLS forwarding table lookup process is both straightforward and fast.

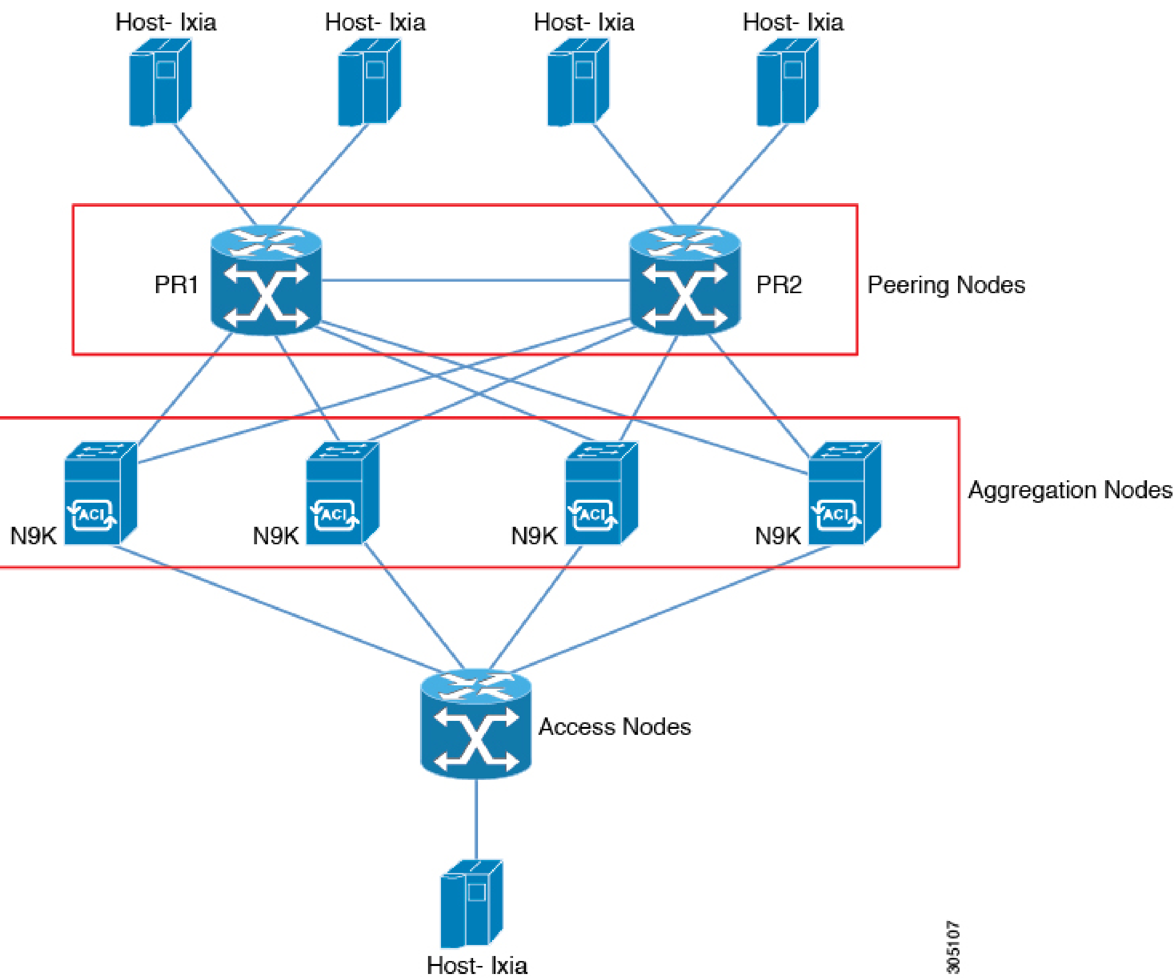
In a swap operation, the label is swapped with a new label, and the packet is forwarded to the next hop that is determined by the incoming label.

In a pop operation, the label is removed from the packet, which may reveal an inner label below. If the popped label was the last label on the label stack, the packet exits the MPLS domain. Typically, this process occurs at the egress LSR. A failure of the primary link in the aggregator reroutes the MPLS traffic to the backup link and results in a swap operation.

Static MPLS Topology

This diagram illustrates the static MPLS source routing topology. The access nodes perform the swap operation, and the aggregation nodes perform the pop operation for the primary path and the swap operation for the backup path.

Figure 1: Static MPLS Topology



Benefits of Static MPLS

- Static bindings between labels and IPv4 or IPv6 prefixes can be configured to support MPLS hop-by-hop forwarding through neighbor routers that do not implement LDP label distribution.
- Static cross-connects can be configured to support MPLS label switched path (LSP) midpoints when neighbor routers do not implement either LDP or RSVP label distribution but do implement an MPLS forwarding path.

Prerequisites for Static MPLS

Static MPLS has the following prerequisites:

- You must configure the ACL TCAM region size for MPLS, save the configuration, and reload the switch.

> **Note** By default the MPLS region size is zero. You need to configure this region to 256 in order to support static MPLS.

Guidelines and Limitations for Static MPLS

Static MPLS has the following guidelines and limitations:

- Static MPLS and MPLS stripping cannot be enabled at the same time.
- Adjacency statistics are supported for Cisco Nexus 3100 Series switches but not for Cisco Nexus 3000 Series switches.
- Equal-cost multipath (ECMP) is not supported with label pop.
- Label pop and swap operations are supported, but label push operations are not.
- MPLS packets will be forwarded as long as the ingress label matches the configured label and the configured FEC (prefix) is in the routing table.
- The device performs as a label switching router (LSR). It performs as a label edge router (LER) only for penultimate hop popping (PHP), when the outermost label of an MPLS tagged packet is removed by an LSR before the packet is passed to an adjacent LER. This means that label switching router (LSR) functions with 2 (or more) incoming labels.

  The device generally performs as a label switching router (LSR). It performs as a label edge router (LER) for penultimate hop popping, by installing the explicit null label as the out-label in the label FIB (LFIB) by an LSR before the packet is passed to an adjacent LER. This means that label switching router (LSR) unctions with 1 or more labels.

  > **Note** If user intentionally uses implicit-null CLI on LSR, the output packet going to LER, will still have an explicit-null and the inner label.

- Static MPLS supports up to 128 labels.
- The backup path is supported only for a single adjacency and not for ECMP.
- The output for most of the MPLS commands can be generated in XML or JSON. See Verifying the Static MPLS Configuration, on page 12 for an example.
- VRFs, vPCs, FEX, and VXLAN are not supported with static MPLS.
- Subinterfaces are not supported for static MPLS.
- The Forwarding Equivalence Class (FEC) should exactly match routes in the routing table.
- When you configure fast reroute (backup), you can specify only the connected next hop (and not the recursive next hop) as the next-hop prefix in the backup configuration.
- When multiple FECs are sharing the backup (the same next-hop and interface), any change to the backup configuration requires a reconfiguration of all the other FECs that are sharing the backup configuration.

- When the backup path is active, the **show mpls switching labels** command will not show the out label/out interface/next hop and related statistics. You can use the **show forwarding mpls label** *label* **stats platform** command to check the statistics.
- If traffic ingresses or egresses on a non-default unit (where the default unit is unit0), the corresponding ULIB statistics will not be displayed in the output of the **show mpls switching labels** *low-label-value* [*high-label-value*] **detail** command. You can use the **show forwarding mpls label** *label* **stats platform** command to check the statistics.
- If the backup and primary paths are pointing to the same interface, the backup action swap takes precedence.
- Physical (Ethernet) and port channels are supported only for backup.

Configuring Static MPLS

Enabling Static MPLS

You must install and enable the MPLS feature set and then enable the MPLS static feature before you can configure MPLS static labels.

Procedure

	Command or Action	Purpose

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

Step 2	[**no**] **install feature-set mpls** **Example:** `switch(config)# install feature-set mpls`	Installs the MPLS feature set. The **no** form of this command uninstalls the MPLS feature set.
Step 3	[**no**] **feature-set mpls** **Example:** `switch(config)# feature-set mpls`	Enables the MPLS feature set. The **no** form of this command disables the MPLS feature set.
Step 4	[**no**] **feature mpls static** **Example:** `switch(config)# feature mpls static`	Enables the static MPLS feature. The **no** form of this command disables the static MPLS feature.

Step 5	(Optional) **show feature-set**
	Example:
	`switch(config)# show feature-set`
	`Feature Set Name      ID        State`
		Displays the status of the MPLS feature set.

	Command or Action	Purpose
	`--------------------  --------  --------` `mpls                  4         enabled`	

Step 6	(Optional) **show feature | inc mpls_static**
	Example:
	`switch(config)# show feature | inc mpls_static`
	`mpls_static           1         enabled`
		Displays the status of static MPLS.

Reserving Labels for Static Assignment

You can reserve the labels that are to be statically assigned so that they are not dynamically assigned.

Before you begin

Ensure that the static MPLS feature is enabled.

Procedure

	Command or Action	Purpose

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

Step 2	[**no**] **mpls label range** *min-value max-value* [**static** *min-static-value max-static-value*] **Example:** `switch(config)# mpls label range 17 99 static 100 10000`	Reserves a range of labels for static label assignment. The range for the minimum and maximum values is from 16 to 471804.
Step 3	(Optional) **show mpls label range** **Example:** `switch(config)# show mpls label range`	Displays the label range that is configured for static MPLS.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring Static Label and Prefix Binding Using the Swap and Pop Operations

In a top-of-rack configuration, the outer label is swapped to the specified new label. The packet is forwarded to the next-hop address, which is auto-resolved by the new label.

In an aggregator configuration, the outer label is popped, and the packet with the remaining label is forwarded to the next-hop address. Pop operations are performed in the primary path, and swap operations are performed in the backup path.

Before you begin

Ensure that the static MPLS feature is enabled.

Procedure

	Command or Action	Purpose

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

Step 2	**interface** *type slot/port*
	Example:
	`switch(config)# interface ethernet 2/2`
	`switch(config-if)#`
		Enters the interface configuration mode for the specified interface.

Step 3	[**no**] **mpls ip forwarding** **Example:** `switch(config-if)# mpls ip forwarding`	Enables MPLS on the specified interface. The **no** form of this command disables MPLS on the specified interface.

Step 4	**mpls static configuration**
	Example:
	`switch(config-if)# mpls static configuration`
	`switch(config-mpls-static)#`
		Enters MPLS static global configuration mode.

Step 5	**address-family** {**ipv4** | **ipv6**} **unicast**
	Example:
	`switch(config-mpls-static)# address-family ipv4 unicast`
	`switch(config-mpls-static-af)#`
		Enters global address family configuration mode for the specified IPv4 or IPv6 address family.

Step 6	**local-label** *local-label-value* **prefix** *destination-prefix destination-prefix-mask*
	Example:
	`switch(config-mpls-static-af)# local-label 2000 prefix 1.255.200.0 255.255.255.25`
	`switch(config-mpls-static-af-lbl)#`
		Specifies static binding of incoming labels to IPv4 or IPv6 prefixes. The *local-label-value* is the range of the static MPLS label defined in the **mpls label range** command.

Step 7	**next-hop** {**auto-resolve** | *destination-ip-next-hop* **out-label implicit-null** | **backup** *local-egress-interface destination-ip-next-hop* **out-label** *output-label-value*}	Specifies the next hop. These options are available: • **next-hop auto-resolve**—Use this option for label swap operations.

	Command or Action	Purpose
	Example: `switch(config-mpls-static-af-lbl)# next-hop auto-resolve`	• **next-hop** *destination-ip-next-hop* **out-label implicit-null**—Use this option for the primary path in label pop operations. • **next-hop backup** *local-egress-interface destination-ip-next-hop* **out-label** *output-label-value*—Use this option for the backup path in label pop operations.
Step 8	(Optional) **copy running-config startup-config** **Example:** `switch(config-mpls-static-af-lbl)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Verifying the Static MPLS Configuration

To display the static MPLS configuration, perform one of the following tasks:

Command	Purpose
show feature | inc mpls_static	Displays the status of static MPLS.
show feature-set	Displays the status of the MPLS feature set.
show ip route	Displays routes from the unicast Routing Information Base (RIB).
show mpls label range	Displays the label range that is configured for static MPLS.
show mpls static binding {**all** | **ipv4** | **ipv6**}	Displays the configured static prefix or label bindings.
show mpls switching [**detail**]	Displays MPLS switching information.
show mpls switching label [**detail**]	Displays the MPLS switching label information.
show forwarding mpls [**label** *label*] **stats**	Displays the adjacency statistics based on the label enabled.
show forwarding adjacency mpls stats	Displays the adjacency statistics

This example shows sample output for the **show mpls static binding all** command:

```
1.255.200.0/32: (vrf: default) Incoming label: 2000
  Outgoing labels:
      1.21.1.1 implicit-null
      backup 1.24.1.1 2001

2000:1:255:201::1/128: (vrf: default) Incoming label: 3000
  Outgoing labels:
      2000:1111:2121:1111:1111:1111:1111:1 implicit-null
```

```
backup 2000:1:24:1::1 3001
```

This example shows sample output for the **show mpls switching detail** command:

```
VRF default

IPv4 FEC
 In-Label                       : 2000
 Out-Label stack                : Pop Label
 FEC                            : 1.255.200.0/32
 Out interface                  : Po21
 Next hop                       : 1.21.1.1
 Input traffic statistics       : 0 packets, 0 bytes
 Output statistics per label    : 0 packets, 0 bytes
IPv6 FEC
 In-Label                       : 3000
 Out-Label stack                : Pop Label
 FEC                            : 2000:1:255:201::1/128
 Out interface                  : port-channel21
 Next hop                       : 2000:1111:2121:1111:1111:1111:1111:1
 Input traffic statistics       : 0 packets, 0 bytes
 Output statistics per label    : 0 packets, 0 bytes
```

This example shows normal, XML, and JSON sample output for the **show mpls switching** command when the switch is configured with a static IPv4 prefix:

```
switch# show run mpls static | sec 'ipv4 unicast'
address-family ipv4 unicast
local-label 100 prefix 192.168.0.1 255.255.255.255 next-hop auto-resolve out-label 200

switch# show mpls switching
Legend:
(P)=Protected, (F)=FRR active, (*)=more labels in stack.
IPV4:
In-Label   Out-Label  FEC name           Out-Interface      Next-Hop

VRF default
100        200        192.168.0.1/32     Eth1/23            1.12.23.2


switch# show mpls switching | xml
<?xml version="1.0" encoding="ISO-8859-1"?> <nf:rpc-reply
xmlns:nf="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns="http://w
ww.cisco.com/nxos:1.0:ulib">
 <nf:data>
  <show>
   <mpls>
    <switching>
     <__XML__OPT_Cmd_ulib_show_switching_cmd_labels>
      <__XML__OPT_Cmd_ulib_show_switching_cmd_detail>
       <__XML__OPT_Cmd_ulib_show_switching_cmd_vrf>
        <__XML__OPT_Cmd_ulib_show_switching_cmd___readonly__>
         <__readonly__>
          <TABLE_vrf>
           <ROW_vrf>
            <vrf_name>default</vrf_name>
            <TABLE_inlabel>
             <ROW_inlabel>
              <in_label>100</in_label>
              <out_label_stack>200</out_label_stack>
              <ipv4_prefix>192.168.0.1/32</ipv4_prefix>
              <out_interface>Eth1/23</out_interface>
```

```
              <ipv4_next_hop>1.12.23.2</ipv4_next_hop>
              <nhlfe_p2p_flag> </nhlfe_p2p_flag>
             </ROW_inlabel>
            </TABLE_inlabel>
           </ROW_vrf>
          </TABLE_vrf>
         </__readonly__>
        </__XML__OPT_Cmd_ulib_show_switching_cmd___readonly__>
       </__XML__OPT_Cmd_ulib_show_switching_cmd_vrf>
      </__XML__OPT_Cmd_ulib_show_switching_cmd_detail>
     </__XML__OPT_Cmd_ulib_show_switching_cmd_labels>
    </switching>
   </mpls>
  </show>
 </nf:data>
</nf:rpc-reply>
]]>]]>
```


```
switch# show mpls switching | json
{"TABLE_vrf": {"ROW_vrf": {"vrf_name": "default", "TABLE_inlabel":
{"ROW_inlabel
": {"in_label": "100", "out_label_stack": "200", "ipv4_prefix":
"192.168.0.1/32"
, "out_interface": "Eth1/23", "ipv4_next_hop": "1.12.23.2",
"nhlfe_p2p_flag": nu
ll}}}}}
```

Displaying Static MPLS Statistics

To monitor static MPLS statistics, perform one of the following tasks:

Command	Purpose
show forwarding [**ipv6**] **adjacency mpls stats**	Displays MPLS IPv4 or IPv6 adjacency statistics.
show forwarding mpls drop-stats	Displays the MPLS forwarding packet drop statistics.
show forwarding mpls ecmp [**module** *slot* | **platform**]	Displays the MPLS forwarding statistics for equal-cost multipath (ECMP).
show forwarding mpls label *label* **stats** [**platform**]	Displays MPLS label forwarding statistics.
show mpls forwarding statistics [**interface** *type slot/port*]	Displays MPLS forwarding statistics.
show mpls switching labels *low-label-value* [*high-label-value*] [**detail**]	Displays the MPLS label switching statistics. The range for the label value is from 0 to 524286.

This example shows sample output for the **show forwarding adjacency mpls stats** command:

```
FEC             next-hop   interface  tx packets  tx bytes  Label info
--------------- ---------- ---------- ----------- --------- ----------
1.255.200.0/32  1.21.1.1   Po21       87388       10836236  POP 3
1.255.200.0/32  1.24.1.1   Po24       0           0         SWAP 2001
switch(config)#
switch(config)# show forwarding mpls drop-stats
```

```
Dropped packets : 73454
Dropped bytes : 9399304
```

This example shows sample output for the **show forwarding ipv6 adjacency mpls stats** command:

```
FEC                     next-hop        interface  tx packets  tx bytes  Label info
----------------------  --------------  ---------- ----------- --------- -----------
2000:1:255:201::1/128  2000:1.21.1.1    Po21       46604       5778896   POP 3
2000:1:255:201::1/128  2000:1:24:1::1  Po24       0           0         SWAP 3001
```

This example shows sample output for the **show forwarding mpls label 2000 stats** command:

```
--------+-----------+-------------------+----------------+-------------+-------
Local   |Prefix     |FEC                |Next-Hop        |Interface    |Out
Label   |Table Id   |(Prefix/Tunnel id) |                |             |Label
--------+-----------+-------------------+----------------+-------------+-------
2000    |0x1        |1.255.200.0/32     |1.21.1.1        |Po21         |Pop Label
 HH: 100008, Refcount: 1
Input Pkts : 77129                Input Bytes : 9872512
Output Pkts: 77223                Output Bytes: 9575652
```

This example shows sample output for the **show mpls forwarding statistics** command:

```
MPLS software forwarding stats summary:
    Packets/Bytes sent           : 0/0
    Packets/Bytes received       : 0/0
    Packets/Bytes forwarded      : 0/0
    Packets/Bytes originated     : 0/0
    Packets/Bytes consumed       : 0/0
    Packets/Bytes input dropped  : 0/0
    Packets/Bytes output dropped : 0/0
```

Clearing Static MPLS Statistics

To clear the static MPLS statistics, perform these tasks:

Command	Purpose
clear forwarding [**ipv6**] **adjacency mpls stats**	Clears the MPLS IPv4 or IPv6 adjacency statistics.
clear forwarding mpls drop-stats	Clears the MPLS forwarding packet drop statistics.
clear forwarding mpls stats	Clears the ingress MPLS forwarding statistics.
clear mpls forwarding statistics	Clears the MPLS forwarding statistics.
clear mpls switching label statistics [**interface** *type slot/port*]	Clears the MPLS switching label statistics.

Configuration Examples for Static MPLS

This example shows how to reserve labels for static assignment:

```
switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)# mpls label range 17 99 static 100 10000
switch(config)# show mpls label range
Downstream Generic label region: Min/Max label: 17/99
Range for static labels: Min/Max Number: 100/10000
```

This example shows how to configure MPLS static label and IPv4 prefix binding in a top-of-rack configuration (swap configuration):

```
switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)# interface ethernet 1/1
switch(config-if)# mpls ip forwarding
switch(config-if)# mpls static configuration
switch(config-mpls-static)# address-family ipv4 unicast
switch(config-mpls-static-af)# local-label 2000 prefix 1.255.200.0/32
switch(config-mpls-static-af-lbl)# next-hop auto-resolve out-label 2000
```

This example shows how to configure MPLS static label and IPv6 prefix binding in a top-of-rack configuration (swap configuration):

```
switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)# interface ethernet 1/1
switch(config-if)# mpls ip forwarding
switch(config-if)# mpls static configuration
switch(config-mpls-static)# address-family ipv6 unicast
switch(config-mpls-static-af)# local-label 3001 prefix 2000:1:255:201::1/128
switch(config-mpls-static-af-lbl)# next-hop auto-resolve out-label 3001
```

This example shows how to configure MPLS static label and IPv4 prefix binding in an aggregator configuration (pop configuration):

```
switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)# interface ethernet 1/1
switch(config-if)# mpls ip forwarding
switch(config-if)# mpls static configuration
switch(config-mpls-static)# address-family ipv4 unicast
switch(config-mpls-static-af)# local-label 2000 prefix 1.255.200.0/32
switch(config-mpls-static-af-lbl)# next-hop 1.31.1.1 out-label implicit-null
switch(config-mpls-static-af-lbl)# next-hop backup Po34 1.34.1.1 out-label 2000
```

This example shows how to configure MPLS static label and IPv6 prefix binding in an aggregator configuration (pop configuration):

```
switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)# interface ethernet 1/1
switch(config-if)# mpls ip forwarding
switch(config-if)# mpls static configuration
switch(config-mpls-static)# address-family ipv6 unicast
switch(config-mpls-static-af)# local-label 3001 prefix 2000:1:255:201::1/128
switch(config-mpls-static-af-lbl)# next-hop 2000:1:31:1::1 out-label implicit-null
switch(config-mpls-static-af-lbl)# next-hop backup Po34 2000:1:34:1::1 out-label 3001
```

Additional References

Related Documents

Related Topic	Document Title
MPLS TCAM regions	Configuring IP ACLs section in the *Cisco Nexus 3000 Series NX-OS Security Configuration Guide*.

CHAPTER 4

Configuring MPLS Label Imposition

This chapter contains information on how to configure multiprotocol label switching (MPLS) label imposition.

About MPLS Label Imposition

An outgoing label stack having one or more labels can be statically provisioned using the MPLS Label Stack Imposition feature. The outgoing label stack is used in the following two types of statically configured MPLS bindings:

- Prefix and Label to Label Stack - Here an IP prefix or an incoming label is mapped to an outgoing stack, similar to static MPLS. An incoming prefix is mapped to out-label-stack for IP-only ingress traffic.
- Label to Label Stack - Here only an incoming label is mapped to an outgoing stack without any prefix.

The new MPLS binding types are implemented in the static MPLS component and are available only when the **feature mpls segment-routing** command is enabled.

If configured next-hops of MPLS label imposition are SR recursive next-hops (RNH), then they are resolved to actual next-hops using RIB. The outer label of the out-label stack is imposed automatically from the SR allocated labels.

ECMP is also supported by adding a number of path configurations.

Note The static MPLS process is started when either the **feature mpls segment-routing** command or the **feature mpls static** command is run. Certain standard static MPLS commands will not be available when static MPLS is run using the **feature mpls segment-routing** command, and the commands for MPLS bindings will not be available when the **feature mpls static** command is run.

Guidelines and Limitations for MPLS Label Imposition

The MPLS label imposition has the following guidelines and limitations:

- The MPLS label imposition supports only IPv4.
- The maximum number of labels in an out-label stack is three for Nexus 3000-XL switches.
- The MPLS label imposition is supported on Nexus 3000-XL running in Nexus 9000 mode.
- Multicast is not supported for the MPLS label imposition.
- For the MPLS label imposition, upto 128 Label Switched Paths (LSPs) can be configured and each LSP can have a maximum of 32 next-hops.
- In the multi-label stack configuration, changing an outgoing path is not allowed, instead delete it.
- Sub-interfaces are not supported for multi-label imposition.
- Contention between MPLS label imposition and Segment Routing or any other routing protocol including static routes is not supported.
- The maximum number of labels in an out-label stack is three for Nexus 3000-XL switches. If more than three labels are tried to impose, then the trailing label is truncated automatically and a syslog error message appears signaling to correct the configurations.

Configuring MPLS Label Imposition

Enabling MPLS Label Imposition

You must install and enable the MPLS feature set and then enable the MPLS segment routing feature before you can configure MPLS label imposition.

Procedure

	Command or Action	Purpose

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

Step 2	[**no**] **install feature-set mpls** **Example:** `switch(config)# install feature-set mpls`	Installs the MPLS feature set. The **no** form of this command uninstalls the MPLS feature set.
Step 3	[**no**] **feature-set mpls** **Example:** `switch(config)# feature-set mpls`	Enables the MPLS feature set. The **no** form of this command disables the MPLS feature set.

	Command or Action	Purpose
Step 4	[**no**] **feature mpls segment-routing** **Example:** `switch(config)# feature mpls segment-routing`	Enables the MPLS segment routing feature. The **no** form of this command disables the MPLS segment routing feature.

Step 5	(Optional) **show feature-set**
	Example:
	`switch(config)# show feature-set`
	`Feature Set Name      ID        State`
	`--------------------  --------  --------`
	`mpls                  4         enabled`
		Displays the status of the MPLS feature set.

Step 6	(Optional) **show feature | grep segment-routing**
	Example:
	`switch(config)# show feature | grep segment-routing`
	`segment-routing        1        enabled`
		Displays the status of MPLS segment routing.

Reserving Labels for MPLS Label Imposition

You can reserve the labels that are to be statically assigned. Dynamic label allocation is not supported.

Before you begin

Ensure that the MPLS segment routing feature is enabled.

Procedure

	Command or Action	Purpose

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

Step 2	[**no**] **mpls label range** *min-value max-value* [**static** *min-static-value max-static-value*] **Example:** `switch(config)# mpls label range 17 99 static 100 10000`	Reserves a range of labels for static label assignment. The range for the minimum and maximum values is from 16 to 471804.
Step 3	(Optional) **show mpls label range** **Example:** `switch(config)# show mpls label range`	Displays the label range that is configured for static MPLS.

	Command or Action	Purpose
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring MPLS Label Imposition

You can configure MPLS label imposition on the device.

Note The **feature mpls segment-routing** command cannot be enabled when the following commands are in use: **feature nv overlay**, **nv overlay evpn**, **feature vpc**, and **feature vn-segment-vlan-based**.

Before you begin

Ensure that the MPLS segment routing feature is enabled.

Set a static label range as follows: **mpls label range 16 16 static 17 50000**.

Procedure

	Command or Action	Purpose

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

Step 2	**interface** *type slot/port*
	Example:
	`switch(config)# interface ethernet 2/2`
	`switch(config-if)#`
		Enters the interface configuration mode for the specified interface.

Step 3	[**no**] **mpls ip forwarding** **Example:** `switch(config-if)# mpls ip forwarding`	Enables MPLS on the specified interface. The **no** form of this command disables MPLS on the specified interface.

Step 4	**mpls static configuration**
	Example:
	`switch(config-if)# mpls static configuration`
	`switch(config-mpls-static)#`
		Enters MPLS static global configuration mode.

Step 5	**address-family ipv4 unicast** **Example:**	Enters global address family configuration mode for the specified IPv4 address family.

	Command or Action	Purpose
	`switch(config-mpls-static)# address-family ipv4 unicast switch(config-mpls-static-af)#`	
Step 6	**lsp** *name* **Example:** `switch(config-mpls-static-af)# lsp lsp1 switch(config-mpls-static-lsp)#`	Specifies a name for LSP.
Step 7	**in-label** *value* **allocate policy** *prefix* **Example:** `switch(config-mpls-static-lsp)# in-label 8100 allocate policy 15.15.1.0/24 switch(config-mpls-static-lsp-inlabel)#`	Configures an in-label value and a prefix value (optional).
Step 8	**forward** **Example:** `switch(config-mpls-static-lsp-inlabel)# forward switch(config-mpls-static-lsp-inlabel-forw)#`	Enters the forward mode.
Step 9	**path** *number* **next-hop** *ip-address* **out-label-stack** *label-id label-id* **Example:** `switch(config-mpls-static-lsp-inlabel-forw)# path 1 next-hop 13.13.13.13 out-label-stack 16 3000`	Specifies the path. The maximum number of supported paths is 32.
Step 10	(Optional) **copy running-config startup-config** **Example:** `switch(config-mpls-static-lsp-inlabel-forw)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Verifying the MPLS Label Imposition Configuration

To display the MPLS label imposition configuration, perform one of the following tasks:

Command	Purpose
show feature | grep segment-routing	Displays the status of MPLS label imposition.
show feature-set	Displays the status of the MPLS feature set.
show forwarding ecmp recursive	Displays VOBJ and the label stack.

Command	Purpose
show forwarding mpls ecmp [**module** *slot* | **platform**]	Displays the MPLS forwarding statistics for equal-cost multipath (ECMP).
show forwarding mpls label *label*	Displays MPLS label forwarding statistics for a particular label.
show mpls label range	Displays the label range that is configured for MPLS label imposition.
show mpls static binding {**all** | **ipv4**}	Displays the configured static prefix or label bindings.
show mpls switching [**detail**]	Displays MPLS label switching information.
show running-config mpls static	Displays the running static MPLS configuration.

This example shows sample output for the **show forwarding ecmp recursive** command:

```
slot  1
=======
 Virtual Object 16 :
    LFIB-ECMP-idx1:0x514ca(333002), LFIB-ECMP-idx2:0x0(0) ADJ-idx 0
    Hw vobj-index (0): unit-0:200022 unit-1:0 unit-2:0, cmn-index: 99004
    Hw NVE vobj-index (0): unit-0:0 unit-1:0 unit-2:0, cmn-index: 99004
    Hw vobj-index (1): unit-0:0 unit-1:0 unit-2:0, cmn-index: 0
    Hw NVE vobj-index (1): unit-0:0 unit-1:0 unit-2:0, cmn-index: 0
    Num prefixes : 0
Partial Install: No
    Active paths:
        Recursive NH 12.12.3.2/32 ,Label stack : 3132 16, table 0x1
        Recursive NH 12.12.4.2/32 ,Label stack : 3132 16, table 0x1
        Recursive NH 12.12.1.2/32 ,Label stack : 3132 16, table 0x1
        Recursive NH 12.12.2.2/32 ,Label stack : 3132 16, table 0x1
    CNHs:
        12.12.1.2, port-channel121
        Hw adj: unit-0:100006 unit-1:0 unit-2:0, cmn-index: 6, LIF:4155
        Hw NVE adj: unit-0:0 unit-1:0 unit-2:0, cmn-index: 6, LIF:4155
        12.12.2.2, Ethernet1/51
        Hw adj: unit-0:100009 unit-1:0 unit-2:0, cmn-index: 7, LIF:4150
        Hw NVE adj: unit-0:0 unit-1:0 unit-2:0, cmn-index: 7, LIF:4150
        12.12.3.2, Vlan122
        Hw adj: unit-0:100012 unit-1:0 unit-2:0, cmn-index: 8, LIF:122
        Hw NVE adj: unit-0:0 unit-1:0 unit-2:0, cmn-index: 8, LIF:122
        12.12.4.2, Vlan123
        Hw adj: unit-0:100017 unit-1:0 unit-2:0, cmn-index: 9, LIF:123
        Hw NVE adj: unit-0:0 unit-1:0 unit-2:0, cmn-index: 9, LIF:123
    Hw instance new : (0x182bc, 99004) ls count new 4
    FEC:
        FEC-ECMP-idx1:0x514cb(333003), FEC-ECMP-idx2:0x0(0) ADJ-idx 0
        Hw instance new: (0x182bd, 99005) ls count new 4
        label list count: (1)
        VOBJ Refcount : 1
Virtual Object 12 :
    LFIB-ECMP-idx1:0x514c8(333000), LFIB-ECMP-idx2:0x0(0) ADJ-idx 0
    Hw vobj-index (0): unit-0:200016 unit-1:0 unit-2:0, cmn-index: 99002
    Hw NVE vobj-index (0): unit-0:0 unit-1:0 unit-2:0, cmn-index: 99002
    Hw vobj-index (1): unit-0:0 unit-1:0 unit-2:0, cmn-index: 0
    Hw NVE vobj-index (1): unit-0:0 unit-1:0 unit-2:0, cmn-index: 0
    Num prefixes : 1
Partial Install: No
    Active paths:
```

```
        Recursive NH 12.12.1.2/32 ,Label stack : 3131 17, table 0x1
        Recursive NH 12.12.2.2/32 ,Label stack : 3131 17, table 0x1
        Recursive NH 12.12.3.2/32 ,Label stack : 3131 17, table 0x1
        Recursive NH 12.12.4.2/32 ,Label stack : 3131 17, table 0x1
    CNHs:
        12.12.1.2, port-channel121
        Hw adj: unit-0:100006 unit-1:0 unit-2:0, cmn-index: 6, LIF:4155
        Hw NVE adj: unit-0:0 unit-1:0 unit-2:0, cmn-index: 6, LIF:4155
        12.12.2.2, Ethernet1/51
        Hw adj: unit-0:100009 unit-1:0 unit-2:0, cmn-index: 7, LIF:4150
        Hw NVE adj: unit-0:0 unit-1:0 unit-2:0, cmn-index: 7, LIF:4150
        12.12.3.2, Vlan122
        Hw adj: unit-0:100012 unit-1:0 unit-2:0, cmn-index: 8, LIF:122
        Hw NVE adj: unit-0:0 unit-1:0 unit-2:0, cmn-index: 8, LIF:122
        12.12.4.2, Vlan123
        Hw adj: unit-0:100017 unit-1:0 unit-2:0, cmn-index: 9, LIF:123
        Hw NVE adj: unit-0:0 unit-1:0 unit-2:0, cmn-index: 9, LIF:123
    Hw instance new : (0x182ba, 99002) ls count new 4
    FEC:
        FEC-ECMP-idx1:0x514c9(333001), FEC-ECMP-idx2:0x0(0) ADJ-idx 0
        Hw instance new: (0x182bb, 99003) ls count new 4
        label list count: (1)
        VOBJ Refcount : 2
```

This example shows sample output for the **show forwarding mpls label 8100** command:

```
slot 1
=======
--------+-----------+-------------------+----------------+-------------+----------------------
Local|Prefix|FEC          |Next-Hop  |Interface | Out Label |Table Id |(Prefix/Tunnel
id)|Label
--------+-----------+-------------------+----------------+------------+----------------------
8100 |0x1   |25.25.0.0/16 |12.12.1.2 |Po121     |3131 SWAP  |         |                 |
 17
"    |0x1   |25.25.0.0/16 |12.12.2.2 |Eth1/51   |3131 SWAP  |         |                 |
 17
"    |0x1   |25.25.0.0/16 |12.12.3.2 |Vlan122   |3131 SWAP  |         |                 |
 17
"    |0x1   |25.25.0.0/16 |12.12.4.2 |Vlan123   |3131 SWAP  |         |                 |
 17
```

This example shows sample output for the **show mpls static binding all** command:

```
LI_TEST1 25.25.0.0/16: (vrf: default) Incoming label: 8100
LSP Type: POLICY
  Outgoing labels:
     (path 1)  12.12.1.2 3131,17
     (path 2)  12.12.2.2 3131,17
     (path 3)  12.12.3.2 3131,17
     (path 4)  12.12.4.2 3131,17

LI_TEST2 (vrf: default) Incoming label: 8200
LSP Type: XC
  Outgoing labels:
     (path 1)  12.12.3.2 3132,16
     (path 2)  12.12.4.2 3132,16
     (path 3)  12.12.1.2 3132,16
     (path 4)  12.12.2.2 3132,16
```

This example shows sample output for the **show mpls switching** command:

```
Legend:
```

```
(P)=Protected, (F)=FRR active, (*)=more labels in stack.

Local      Out-Label  FEC                                          Out-Interface
Next-Hop
8200       3132       Label 8200
12.12.3.2                                  *
8200       3132       Label 8200
12.12.4.2                                  *
8200       3132       Label 8200
12.12.1.2                                  *
8200       3132       Label 8200
12.12.2.2                                  *

Local      Out-Label  FEC                                          Out-Interface
Next-Hop
8100       3131       Pol 25.25.0.0/16
12.12.1.2                                  *
8100       3131       Pol 25.25.0.0/16
12.12.2.2                                  *
8100       3131       Pol 25.25.0.0/16
12.12.3.2                                  *
8100       3131       Pol 25.25.0.0/16
12.12.4.2                                  *
```

This example shows sample output for the **show running-config mpls static** command:

```
mpls static configuration
  address-family ipv4 unicast
    lsp LI_TEST2
      in-label 8100 allocate policy 25.25.0.0 255.255.0.0
        forward
          path 1 next-hop 12.12.1.2 out-label-stack 3131 17
          path 2 next-hop 12.12.2.2 out-label-stack 3131 17
          path 3 next-hop 12.12.3.2 out-label-stack 3131 17
          path 4 next-hop 12.12.4.2 out-label-stack 3131 17
```

This example shows sample output for the **show running-config mpls static all** command.

```
switch# show running-config mpls static all

!Command: show running-config mpls static all
!Time: Mon Aug 21 14:59:46 2017

version 7.0(3)I7(1)
logging level mpls static 5
mpls static configuration
address-family ipv4 unicast
lsp 9_label_stack_LPM
in-label 72000 allocate policy 71.200.11.0 255.255.255.0
forward
path 1 next-hop 27.1.32.4 out-label-stack 21901 29701 27401 24501 25801
lsp 9_label_stack_LPM_01
in-label 72001 allocate policy 72.201.1.1 255.255.255.255
lsp DRV-01
in-label 71011 allocate policy 71.111.21.0 255.255.255.0
forward
path 1 next-hop 27.1.31.4 out-label-stack implicit-null
lsp DRV-02
in-label 71012 allocate policy 71.111.22.0 255.255.255.0
forward
path 1 next-hop 8.8.8.8 out-label-stack 28901
lsp DRV-03
```

```
switch# show forwarding mpls label 72000

slot 1
=======

--------+-----------+-------------------+----------------+-------------+-------
Local |Prefix |FEC |Next-Hop |Interface |Out
Label |Table Id |(Prefix/Tunnel id) | | |Label
--------+-----------+-------------------+----------------+-------------+-------
72000 |0x1 |71.200.11.0/24 |27.1.32.4 |Eth1/21 |21901 SWAP
| | | | | 29701
| | | | | 27401
| | | | | 24501
| | | | | 25801
```

Displaying MPLS Label Imposition Statistics

To monitor MPLS label imposition statistics, perform one of the following tasks:

Command	Purpose
show forwarding [**ipv4**] **adjacency mpls stats**	Displays MPLS IPv4 adjacency statistics (both, packets and bytes).
show forwarding mpls drop-stats	Displays MPLS forwarding packet drop statistics.
show forwarding mpls label *label* **stats** [**platform**]	Displays MPLS label forwarding statistics.
show mpls forwarding statistics [**interface** *type slot/port*]	Displays MPLS forwarding statistics.
show mpls switching labels *low-label-value* [*high-label-value*] [**detail**]	Displays MPLS label switching statistics. The range for the label value is from 0 to 524286.

This example shows sample output for the **show forwarding adjacency mpls stats** command:

```
slot  1
=======

 FEC   next-hop    interface  tx packets   tx bytes  Label info
------ ----------- ---------- ------------ --------- --------------
       12.12.3.2   Vlan122      0           0         SWAP 3131 17
       12.12.3.2   Vlan122      0           0         SWAP 3132 16
       12.12.4.2   Vlan123      0           0         SWAP 3131 17
       12.12.4.2   Vlan123      0           0         SWAP 3132 16
       12.12.1.2   Po121        0           0         SWAP 3131 17
       12.12.1.2   Po121        0           0         SWAP 3132 16
       12.12.2.2   Eth1/51      0           0         SWAP 3131 17
       12.12.2.2   Eth1/51      0           0         SWAP 3132 16
```

This example shows sample output for the **show forwarding mpls label 8100 stats** command:

```
slot  1
=======
--------+-----------+-------------------+----------------+-------------+-------
Local   |Prefix     |FEC                |Next-Hop        |Interface    |Out
Label   |Table Id   |(Prefix/Tunnel id) |                |             |Label
--------+-----------+-------------------+----------------+-------------+-------
8100    |0x1        |25.25.0.0/16       |12.12.1.2       |Po121        |3131
```

```
  SWAP
        |           |                   |                |            | 17
    "   |0x1        |25.25.0.0/16       |12.12.2.2       |Eth1/51      |3131
  SWAP
        |           |                   |                |            | 17
    "   |0x1        |25.25.0.0/16       |12.12.3.2       |Vlan122      |3131
  SWAP
        |           |                   |                |            | 17
    "   |0x1        |25.25.0.0/16       |12.12.4.2       |Vlan123      |3131
  SWAP
        |           |                   |                |            | 17

Input Pkts : 126906012            Input Bytes : 64975876096
SWAP Output Pkts: 126959183       SWAP Output Bytes: 65764550340
TUNNEL Output Pkts: 126959053     TUNNEL Output Bytes: 66272319384
```

This example shows sample output for the **show mpls forwarding statistics** command:

```
MPLS software forwarding stats summary:
    Packets/Bytes sent           : 0/0
    Packets/Bytes received       : 0/0
    Packets/Bytes forwarded      : 0/0
    Packets/Bytes originated     : 0/0
    Packets/Bytes consumed       : 0/0
    Packets/Bytes input dropped  : 0/0
    Packets/Bytes output dropped : 0/0
```

Clearing MPLS Label Imposition Statistics

To clear the MPLS label imposition statistics, perform these tasks:

Command	Purpose
clear forwarding [**ipv4**] **adjacency mpls stats**	Clears the MPLS IPv4 adjacency statistics.
clear forwarding mpls drop-stats	Clears the MPLS forwarding packet drop statistics.
clear forwarding mpls stats	Clears the ingress MPLS forwarding statistics.
clear mpls forwarding statistics	Clears the MPLS forwarding statistics.
clear mpls switching label statistics [**interface** *type slot/port*]	Clears the MPLS switching label statistics.

Configuration Examples for MPLS Label Imposition

This example shows how to configure MPLS label imposition by allocating a prefix and an incoming-label to out-label-stack binding:

```
switch(config-if)# mpls static configuration
switch(config-mpls-static)# address-family ipv4 unicast
switch(config-mpls-static-af)# lsp LI_TEST1
switch(config-mpls-static-lsp)# in-label 8100 allocate policy 25.25.0.0/16
switch(config-mpls-static-lsp-inlabel)# forward
```

```
switch(config-mpls-static-lsp-inlabel-forw)# path 1 next-hop 12.12.1.2 out-label-stack 3131
 17
switch(config-mpls-static-lsp-inlabel-forw)# path 2 next-hop 12.12.2.2 out-label-stack 3131
 17
switch(config-mpls-static-lsp-inlabel-forw)# path 3 next-hop 12.12.3.2 out-label-stack 3131
 17
switch(config-mpls-static-lsp-inlabel-forw)# path 4 next-hop 12.12.4.2 out-label-stack 3131
 17
```

To remove a next-hop, you can use

no path 1

To remove the named lsp, you can use

no lsp LI_TEST1

This example shows how to configure MPLS label imposition by allocating an incoming-label to out-label-stack binding (no prefix):

```
switch(config-if)# mpls static configuration
switch(config-mpls-static)# address-family ipv4 unicast
switch(config-mpls-static-af)# lsp LI_TEST1
switch(config-mpls-static-lsp)# in-label 8200 allocate
switch(config-mpls-static-lsp-inlabel)# forward
switch(config-mpls-static-lsp-inlabel-forw)# path 1 next-hop 12.12.3.2 out-label-stack 3132
 16
switch(config-mpls-static-lsp-inlabel-forw)# path 2 next-hop 12.12.4.2 out-label-stack 3132
 16
switch(config-mpls-static-lsp-inlabel-forw)# path 3 next-hop 12.12.1.2 out-label-stack 3132
 16
switch(config-mpls-static-lsp-inlabel-forw)# path 4 next-hop 12.12.2.2 out-label-stack 3132
 16
```

CHAPTER 5

Configuring Segment Routing

This chapter contains information on how to configure segment routing.

About Segment Routing

Segment routing is a technique by which the path followed by a packet is encoded in the packet itself, similar to source routing. A node steers a packet through a controlled set of instructions, called segments, by prepending the packet with a segment routing header. Each segment is identified by a segment ID (SID) consisting of a flat unsigned 32-bit integer.

Border Gateway Protocol (BGP) segments, a subclass of segments, identify a BGP forwarding instruction. Prefix segments steer packets along the shortest path to the destination, using all available equal-cost multi-path (ECMP) paths.

Border Gateway Protocol - Link State (BGP-LS) is an extension to BGP for distributing the network's Link-State (LS) topology model to external entities. BGP-LS advertise routing updates only when they occur which uses bandwidth more effectively. They advertise only the incremental change to all routers as a multicast update. They use variable length subnet masks, which are scalable and use addressing more efficiently.

The segment routing architecture is applied directly to the MPLS data plane.

Segment Routing Application Module

Segment Routing Application (SR-APP) module is used to configure the segment routing functionality. Segment Routing Application (SR-APP) is a separate internal process that handles all the CLIs related to segment routing. It is responsible for reserving the SRGB range and for notifying the clients about it. It is also responsible for maintaining the prefix to SID mappings. The SR-APP support is also available for the BGP, IS-IS, and OSPF protocols.

The SR-APP module maintains the following information:

- Segment routing operation state
- Segment routing global block label ranges
- Prefix SID mappings

For more information, see Configuring Segment Routing, on page 33.

Guidelines and Limitations for Segment Routing

Segment routing has the following guidelines and limitations:

- Beginning with Cisco NX-OS Release 9.3(1), the segment-routing MPLS command has been changed to segment-routing.
- You can configure segment routing using Segment Routing Application (SR-APP) module. SR-APP is a separate internal process that handles all the CLIs related to segment routing. SR-APP reserves the SRGB range for notifying the clients and it maintains SID mappings prefixes.
- BGP allocates an SRGB label for Internal Border Gateway Protocol route-reflector clients only when next-hop-self is in effect (for example, the prefix is advertised with the next hop being one of the local IP/IPv6 addresses on RR). When you have configured next-hop-self on a RR, the next hop is changed for the routes that are being affected (subject to route-map filtering).
- Static MPLS, MPLS segment routing, and MPLS stripping cannot be enabled at the same time.
- Because static MPLS, MPLS segment routing, and MPLS stripping are mutually exclusive, the only segment routing underlay for multihop BGP is single-hop BGP. Internal Border Gateway Protocol multi-hop topologies with eBGP running as an overlay are not supported.
- MPLS pop followed by a forward to a specific interface is not supported. The penultimate hop pop (PHP) is avoided by installing the Explicit NULL label as the outlabel in the label FIB (LFIB) even when the control plane installs an IPv4 Implicit NULL label.
- BGP labeled unicast and BGP segment routing are not supported for IPv6 prefixes.
- BGP labeled unicast and BGP segment routing are not supported over tunnel interfaces (including GRE and VXLAN) or with vPC access interfaces.
- MTU path discovery (RFC 2923) is not supported over MPLS label switched paths (LSPs) or segment routed paths.
- The BGP configuration commands **neighbor-down fib-accelerate** and **suppress-fib-pending** are not supported for MPLS prefixes.
- The uniform model as defined in RFC 2973 and RFC 3270 is not supported. Hence, the IP DSCP bits are not copied into the imposed MPLS header.
- Reconfiguration of the segment routing global block (SRGB) results in an automatic restart of the BGP process to update the existing URIB and ULIB entries. Traffic loss occurs for a few seconds, so you should not reconfigure the SRGB in production.
- If the segment routing global block (SRGB) is set to a range but the route-map label-index delta value is outside of the configured range, the allocated label is dynamically generated. For example, if the SRGB

is set to a range of 16000-23999 but a route-map label-index is set to 9000, the label is dynamically allocated.

- For network scalability, Cisco recommends using a hierarchical routing design with multi-hop BGP for advertising the attached prefixes from a top-of-rack (ToR) or border leaf switch.
- BGP sessions are not supported over MPLS LSPs or segment routed paths.
- The Layer 3 forwarding consistency checker is not supported for MPLS routes.
- Cisco Nexus 3000 switches support Link-State distribution and Egress Peer Engineering (EPE) using BGP.
- Segment routing and SR-EVPN are supported on Cisco Nexus C31108PC-V, Cisco Nexus C31108TC-V and Cisco Nexus C3132Q-V switches.
- Deleting the segment routing configuration removes all the related segment routing configurations.
- If you downgrade the Cisco Nexus device from Cisco NX-OS Release 9.3(1) to the previous NX-OS releases by setting the boot variables and reloading the switch, all earlier configurations of the segment-routing MPLS will be lost.
- Before performing an ISSD from Cisco NX-OS Release 9.3(1), you must disable the segment routing configuration. Failure to do so results in the loss of the existing segment routing configurations.

Configuring Segment Routing

Configuring Segment Routing

Before you begin

Confirm that the following conditions are met before configuring segment routing.

- The **install feature-set mpls**, **feature-set mpls** and **feature mpls segment-routing** commands should be present before configuring the **segment-routing** command.
- If the global block is configured, the specified range is used. Otherwise, the default 16000 – 23999 range is used.
- BGP now uses both **set label-index** <*value*> configuration and the new **connected-prefix-sid-map** CLI. In case of a conflict, the configuration in SR-APP is preferred.

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	**segment-routing**
	Example:
	`switch(config)# segment-routing`
	`switch(config-sr)# mpls`
	`switch(config-sr-mpls)#`
		Enables the MPLS segment routing functionality. The **no** form of this command disables the MPLS segment routing feature.

Step 3	**connected-prefix-sid-map**
	Example:
	`switch(config-sr-mpls)#`
	`connected-prefix-sid-map`
	`switch(config-sr-mpls)#`
		Configures the connected prefix segment identifier mappings.

Step 4	**global-block** *<min>* *<max>*
	Example:
	`switch(config-sr-mpls)# global-block`
	`<min> <max>`
	`switch(config-sr-mpls)#`
		Specifies the global block range for the segment routing bindings.

Step 5	**connected-prefix-sid-map**
	Example:
	`switch(config-sr-mpls)#`
	`connected-prefix-sid-map`
	`switch(config-sr-mpls-conn-pfsid)#`
		Configures the connected prefix segment identifier mappings.

Step 6	**address-family ipv4**
	Example:
	`switch(config-sr-mpls-conn-pfsid)#address-family`
	`ipv4`
		Configures the IPv4 address family.

Step 7	*<prefix>*/*<masklen>* [**index**|**absolute**] *<label>*
	Example:
	`switch(config-sr-mpls)# 2.1.1.5/32`
	`absolute 201101`
		The optional keywords **index** or **absolute** indicate whether the label value entered should be interpreted as an index into the SRGB or as an absolute value.

Example

See the following configuration examples of the show commands:

```
switch# show segment-routing mpls
Segment-Routing Global info

Service Name: segment-routing

State: Enabled

Process Id: 29123

Configured SRGB: 17000 - 24999

SRGB Allocation status: Alloc-Successful
```

```
Current SRGB: 17000 - 24999

Cleanup Interval: 60

Retry Interval: 180
```

The following CLI displays the clients that are registered with SR-APP. It lists the VRFs, for which the clients have registered interest.

```
switch# show segment-routing mpls clients
            Segment-Routing Mpls Client Info

Client: isis-1
    PIB index: 1    UUID: 0x41000118    PID: 29463    MTS SAP: 412
    TIBs registered:
        VRF: default Table: base

Client: bgp-1
    PIB index: 2    UUID: 0x11b    PID: 18546    MTS SAP: 62252
    TIBs registered:
        VRF: default Table: base

Total Clients: 2
```

In the **show segment-routing mpls ipv4 connected-prefix-sid-map** CLI command example, SRGB indicates whether the prefix SID is within the configured SRGB. The **Indx** field indicates that the configured label is an index into the global block. The **Abs** field indicates that the configured label is an absolute value.

If the SRGB field displays N, it means that the configured prefix SID is not within the SRGB range and it is not provided to the SR-APP clients. Only the prefix SIDs that fall into the SRGB range are given to the SR-APP clients.

```
switch# show segment-routing mpls ipv4 connected-prefix-sid-map
            Segment-Routing Prefix-SID Mappings
Prefix-SID mappings for VRF default Table base
Prefix             SID   Type Range SRGB
13.11.2.0/24       713   Indx 1     Y
30.7.7.7/32        730   Indx 1     Y
59.3.24.0/30       759   Indx 1     Y
150.101.1.0/24     801   Indx 1     Y
150.101.1.1/32     802   Indx 1     Y
150.101.2.0/24     803   Indx 1     Y
1.1.1.1/32         16013 Abs  1     Y
```

The following CLI displays the **show running-config segment-routing** output.

```
switch# show running-config segment-routing ?

> Redirect it to a file
>> Redirect it to a file in append mode
all Show running config with defaults
| Pipe command output to filter

switch# show running-config segment-routing
switch# show running-config segment-routing

!Command: show running-config segment-routing
```

```
!Running configuration last done at: Thu Dec 12 19:39:52 2019
!Time: Thu Dec 12 20:06:07 2019

version 9.3(3) Bios:version 05.39
segment-routing
    mpls
        connected-prefix-sid-map
            address-family ipv4
                2.1.1.1/32 absolute 100100

switch#
```

Enabling MPLS on an Interface

You can enable MPLS on an interface for use with segment routing.

Before you begin

You must install and enable the MPLS feature set using the **install feature-set mpls** and **feature-set mpls** commands.

Procedure

	Command or Action	Purpose

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

Step 2	**interface** *type slot/port*
	Example:
	`switch(config)# interface ethernet 2/2`
	`switch(config-if)#`
		Enters the interface configuration mode for the specified interface.

Step 3	[**no**] **mpls ip forwarding** **Example:** `switch(config-if)# mpls ip forwarding`	Enables MPLS on the specified interface. The **no** form of this command disables MPLS on the specified interface.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring the Segment Routing Global Block

You can configure the beginning and ending MPLS labels in the segment routing global block (SRGB).

Before you begin

You must install and enable the MPLS feature set using the **install feature-set mpls** and **feature-set mpls** commands.

You must enable the MPLS segment routing feature.

Procedure

	Command or Action	Purpose

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

Step 2	[**no**] **segment-routing**
	Example:
	`switch(config)# segment-routing`
	`switch(config-sr)# mpls`
		Enters the segment routing configuration mode and enables the default SRGB of 16000 to 23999. The **no** form of this command unallocates that block of labels.
		If the configured dynamic range cannot hold the default SRGB, an error message appears, and the default SRGB will not be allocated. If desired, you can configure a different SRGB in the next step.

Step 3	[**no**] **global-block** *beginning-label ending-label* **Example:** `switch(config-sr-mpls)# global-block 16000 471804`	Specifies the MPLS label range for the SRGB. Use this command if you want to change the default SRGB label range that is configured with the **segment-routing** command. The permissive values for the beginning MPLS label and the ending MPLS label are from 16000 to 471804. The **mpls label range** command permits 16 as the minimum label, but the SRGB can start only from 16000. **Note** The minimum value for the **global-block** command starts from 16000. If you upgrading from previous releases, you should modify the SRGB so that it falls within the supported range before triggering an upgrade.
Step 4	(Optional) **show mpls label range** **Example:** `switch(config-sr-mpls)# show mpls label range`	Displays the SRGB, only if the SRGB allocation is successful.
Step 5	**show segment-routing**	Displays the configured SRGB.

	Command or Action	Purpose
Step 6	**show segment-routing mpls** **Example:** `switch(config-sr-mpls)# show segment-routing mpls`	Displays the configured SRGB.
Step 7	(Optional) **copy running-config startup-config** **Example:** `switch(config-sr-mpls)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring the Label Index

You can set the label index for routes that match the **network** command. Doing so causes the BGP prefix SID to be advertised for local prefixes that are configured with a route map that includes the **set label-index** command, provided the route map is specified in the **network** command that specifies the local prefix.

Note Segment Routing Application (SR-APP) module is used to configure the segment routing functionality. BGP now uses both **set label-index** <*value*> configuration under route-map and the new **connected-prefix-sid-map** CLI for prefix SID configuration. In case of a conflict, the configuration in SR-APP is preferred.

Note Route-map label indexes are ignored when the route map is specified in a context other than the **network** command. Also, labels are allocated for prefixes with a route-map label index independent of whether the prefix has been configured by the **allocate-label route-map** *route-map-name* command.

Procedure

	Command or Action	Purpose

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

Step 2	**route-map** *map-name*
	Example:
	`switch(config)# route-map SRmap`
	`switch(config-route-map)#`
		Creates a route map or enters route-map configuration mode for an existing route map.

Step 3	[**no**] **set label-index** *index* **Example:** `switch(config-route-map)# set label-index 10`	Sets the label index for routes that match the **network** command. The range is from 0 to 471788. By default, a label index is not added to the route.

	Command or Action	Purpose

Step 4	**exit**
	Example:
	`switch(config-route-map)# exit`
	`switch(config)#`
		Exits route-map configuration mode.

Step 5	**router bgp** *autonomous-system-number*
	Example:
	`switch(config)# router bgp 64496`
	`switch(config-router)#`
		Enables BGP and assigns the AS number to the local BGP speaker. The AS number can be a 16-bit integer or a 32-bit integer in the form of a higher 16-bit decimal number and a lower 16-bit decimal number in xx.xx format.

Step 6	Required: **address-family ipv4 unicast**
	Example:
	`switch(config-router)# address-family ipv4 unicast`
	`switch(config-router-af)#`
		Enters global address family configuration mode for the IPv4 address family.

Step 7	**network** *ip-prefix* [**route-map** *map-name*] **Example:** `switch(config-router-af)# network 10.10.10.10/32 route-map SRmap`	Specifies a network as local to this autonomous system and adds it to the BGP routing table.
Step 8	(Optional) **show route-map** [*map-name*] **Example:** `switch(config-router-af)# show route-map`	Displays information about route maps, including the label index.
Step 9	(Optional) **copy running-config startup-config** **Example:** `switch(config-router-af)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuration Examples for Segment Routing

The examples in this section show a common BGP prefix SID configuration between two routers.

This example shows how to advertise a BGP speaker configuration of 10.10.10.10/32 and 20.20.20.20/32 with a label index of 10 and 20, respectively. It uses the default segment routing global block (SRGB) range of 16000 to 23999.

```
hostname s1
install feature-set mpls
feature-set mpls

feature telnet
feature bash-shell
feature scp-server
feature bgp
feature mpls segment-routing

segment-routing
```

```
  mpls
  vlan 1
segment-routing
  mpls
    connected-prefix-sid-map
    address-family ipv4
    2.1.1.1/32 absolute 100100

route-map label-index-10 permit 10
  set label-index 10
route-map label-index-20 permit 10
  set label-index 20

vrf context management
  ip route 0.0.0.0/0 10.30.108.1

interface Ethernet1/1
  no switchport
  ip address 10.1.1.1/24
  no shutdown

interface mgmt0
  ip address dhcp
  vrf member management

interface loopback1
  ip address 10.10.10.10/32

interface loopback2
  ip address 20.20.20.20/32

line console
line vty

router bgp 1
  address-family ipv4 unicast
    network 10.10.10.10/32 route-map label-index-10
    network 20.20.20.20/32 route-map label-index-20
    allocate-label all
  neighbor 10.1.1.2 remote-as 2
    address-family ipv4 labeled-unicast
```

This example shows how to receive the configuration from a BGP speaker.

```
hostname s2
install feature-set mpls
feature-set mpls

feature telnet
feature bash-shell
feature scp-server
feature bgp
feature mpls segment-routing

segment-routing mpls
vlan 1

vrf context management
  ip route 0.0.0.0/0 10.30.97.1
  ip route 0.0.0.0/0 10.30.108.1

interface Ethernet1/1
  no switchport
  ip address 10.1.1.2/24
```

```
  ipv6 address 10:1:1::2/64
  no shutdown

interface mgmt0
  ip address dhcp
  vrf member management

interface loopback1
  ip address 2.2.2.2/32
line console

line vty

router bgp 2
  address-family ipv4 unicast
    allocate-label all
  neighbor 10.1.1.1 remote-as 1
    address-family ipv4 labeled-unicast
```

This example shows how to display the configuration from a BGP speaker. The **show** command in this example displays the prefix 10.10.10.10 with label index 10 mapping to label 16010 in the SRGB range of 16000 to 23999.

```
switch# show bgp ipv4 labeled-unicast 10.10.10.10/32

BGP routing table information for VRF default, address family IPv4 Label Unicast
BGP routing table entry for 10.10.10.10/32, version 7
Paths: (1 available, best #1)
Flags: (0x20c001a) on xmit-list, is in urib, is best urib route, is in HW, , has label
  label af: version 8, (0x100002) on xmit-list
  local label: 16010

  Advertised path-id 1, Label AF advertised path-id 1
  Path type: external, path is valid, is best path, no labeled nexthop, in rib
  AS-Path: 1 , path sourced external to AS
    10.1.1.1 (metric 0) from 10.1.1.1 (10.10.10.10)
      Origin IGP, MED not set, localpref 100, weight 0
      Received label 0
      Prefix-SID Attribute: Length: 10
        Label Index TLV: Length 7, Flags 0x0 Label Index 10

  Path-id 1 not advertised to any peer
  Label AF advertisement
  Path-id 1 not advertised to any peer
```

This example shows how to configure egress peer engineering on a BGP speaker.

```
hostname epe-as-1
install feature-set mpls
feature-set mpls

feature telnet
feature bash-shell
feature scp-server
feature bgp
feature mpls segment-routing

segment-routing mpls
vlan 1

vrf context management
  ip route 0.0.0.0/0 10.30.97.1
  ip route 0.0.0.0/0 10.30.108.1
```

```
interface Ethernet1/1
  no switchport
  ip address 10.1.1.1/24
  no shutdown

interface Ethernet1/2
  no switchport
  ip address 11.1.1.1/24
  no shutdown

interface Ethernet1/3
  no switchport
  ip address 12.1.1.1/24
  no shutdown

interface Ethernet1/4
  no switchport
  ip address 13.1.1.1/24
  no shutdown

interface Ethernet1/5
  no switchport
  ip address 14.1.1.1/24
  no shutdown
```

The following is an example of show ip route vrf 2 command.

```
show ip route vrf 2
IP Route Table for VRF "2"
'*' denotes best ucast next-hop
'**' denotes best mcast next-hop
'[x/y]' denotes [preference/metric]
'%<string>' in via output denotes VRF <string>

41.11.2.0/24, ubest/mbest: 1/0
    *via 1.1.1.9%default, [20/0], 13:26:48, bgp-2, external, tag 11 (mpls-vpn)
42.11.2.0/24, ubest/mbest: 1/0, attached
    *via 42.11.2.1, Vlan2, [0/0], 13:40:52, direct
42.11.2.1/32, ubest/mbest: 1/0, attached
    *via 42.11.2.1, Vlan2, [0/0], 13:40:52, local
```

The following is an example of **show forwarding route vrf 2** command.

```
slot  1
=======

IPv4 routes for table 2/base

------------------+---------------------------------------+---------------------+----------------+----------------
Prefix            | Next-hop                              | Interface           | Labels
          | Partial Install
------------------+---------------------------------------+---------------------+----------------+----------------
0.0.0.0/32           Drop                                    Null0
127.0.0.0/8          Drop                                    Null0
255.255.255.255/32   Receive                                 sup-eth1
*41.11.2.0/24        27.1.31.4                               Ethernet1/3            PUSH
  30002 492529
                     27.1.32.4                               Ethernet1/21           PUSH
```

```
  30002 492529
                   27.1.33.4                                  port-channel23         PUSH
  30002 492529
                   27.11.31.4                                 Ethernet1/3.11         PUSH
  30002 492529
                   27.11.33.4                                 port-channel23.11      PUSH
  30002 492529
                   37.1.53.4                                  Ethernet1/53/1         PUSH
  29002 492529
                   37.1.54.4                                  Ethernet1/54/1         PUSH
  29002 492529
                   37.2.53.4                                  Ethernet1/53/2         PUSH
  29002 492529
                   37.2.54.4                                  Ethernet1/54/2         PUSH
  29002 492529
                   80.211.11.1                                Vlan801                PUSH
  30002 492529
```

The following is an example of **show bgp l2vpn evpn summary** command.

```
show bgp l2vpn evpn summary
BGP summary information for VRF default, address family L2VPN EVPN
BGP router identifier 2.2.2.3, local AS number 2
BGP table version is 17370542, L2VPN EVPN config peers 4, capable peers 1
1428 network entries and 1428 paths using 268464 bytes of memory
BGP attribute entries [476/76160], BGP AS path entries [1/6]
BGP community entries [0/0], BGP clusterlist entries [0/0]
476 received paths for inbound soft reconfiguration
476 identical, 0 modified, 0 filtered received paths using 0 bytes

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
1.1.1.1         4    11       0       0        0    0    0 23:01:53 Shut (Admin)
1.1.1.9         4    11    4637    1836 17370542    0    0 23:01:40 476
1.1.1.10        4    11       0       0        0    0    0 23:01:53 Shut (Admin)
1.1.1.11        4    11       0       0        0    0    0 23:01:52 Shut (Admin)
```

The following is an example of **show bgp l2vpn evpn** command.

```
show bgp l2vpn evpn 41.11.2.0
BGP routing table information for VRF default, address family L2VPN EVPN
Route Distinguisher: 14.1.4.1:115
BGP routing table entry for [5]:[0]:[0]:[24]:[41.11.2.0]:[0.0.0.0]/224, version 17369591
Paths: (1 available, best #1)
Flags: (0x000002) on xmit-list, is not in l2rib/evpn, is not in HW

  Advertised path-id 1
  Path type: external, path is valid, received and used, is best path
             Imported to 2 destination(s)
  AS-Path: 11 , path sourced external to AS
    1.1.1.9 (metric 0) from 1.1.1.9 (14.1.4.1)
      Origin incomplete, MED 0, localpref 100, weight 0
      Received label 492529
      Extcommunity: RT:2:20

  Path-id 1 not advertised to any peer

Route Distinguisher: 2.2.2.3:113
```

```
BGP routing table entry for [5]:[0]:[0]:[24]:[41.11.2.0]:[0.0.0.0]/224, version 17369595
Paths: (1 available, best #1)
Flags: (0x000002) on xmit-list, is not in l2rib/evpn, is not in HW

  Advertised path-id 1
  Path type: external, path is valid, is best path
             Imported from 14.1.4.1:115:[5]:[0]:[0]:[24]:[41.11.2.0]:[0.0.0.0]/224
  AS-Path: 11 , path sourced external to AS
    1.1.1.9 (metric 0) from 1.1.1.9 (14.1.4.1)
```

Segment Routing with IS-IS Protocol

About IS-IS

IS-IS is an Interior Gateway Protocol (IGP) based on Standardization (ISO)/International Engineering Consortium (IEC) 10589 and RFC 1995. Cisco NX-OS supports Internet Protocol version 4 (IPv4) and IPv6. IS-IS is a dynamic link-state routing protocol that can detect changes in the network topology and calculate loop-free routes to other nodes in the network. Each router maintains a link-state database that describes the state of the network and sends packets on every configured link to discover neighbors. IS-IS floods the link-state information across the network to each neighbor. The router also sends advertisements and updates on the link-state database through all the existing neighbors

Segment routing on the IS-IS protocol supports the following:

- IPv4
- Level 1, level 2, and multi-level routing
- Prefix SIDs
- Multiple IS-IS instances on the same loopback interface for domain border nodes
- Adjacency SIDs for adjacencies

Configuring Segment Routing with IS-IS Protocol

You can configure segment routing with IS-IS protocol.

Before you begin

Ensure the following conditions are met to enable IS-IS Segment Routing:

- The **mpls segment-routing** feature is enabled.
- The IS-IS feature is enabled.
- Segment routing is enabled for at least one address family under IS-IS.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal**	Enters global configuration mode.

	Command or Action	Purpose
Step 2	**router isis** *instance-tag*	Creates a new IS-IS instance with the configured instance tag.
Step 3	**net** *network-entity-title*	Configures the NET for this IS-IS instance.
Step 4	**address-family** *ipv4* **unicast**	Enters address family configuration mode.
Step 5	**segment-routing**	Configures segment routing with IS-IS protocol. **Note** • The IS-IS command is supported only on the IPv4 address family. It is not supported on the IPv6 address family. • Redistribution is not supported from any other protocol to ISIS for the SR prefixes. You need to enable **ip router isis** command on all the prefix SID interfaces.
Step 6	(Optional) **show running-config segment-routing**	Displays the status of the segment routing.

Segment Routing with OSPFv2 Protocol

About OSPF

Open Shortest Path First (OSPF) is an Interior Gateway Protocol (IGP) developed by the OSPF working group of the Internet Engineering Task Force (IETF). Designed expressly for IP networks, OSPF supports IP subnetting and tagging of externally derived routing information. OSPF also allows packet authentication and uses IP multicast when sending and receiving packets.

Segment routing configuration on the OSPF protocol can be applied at the process or the area level. If you configure segment routing at the process level, it is enabled for all the areas. However, you can enable ore disable it per area level.

Segment routing on the OSPF protocol supports the following:

- OSPFv2 control plane
- Multi-area
- IPv4 prefix SIDs for host prefixes on loopback interfaces
- Adjacency SIDs for adjacencies

Adjacency SID Advertisement

OSPF supports the advertisement of segment routing adjacency SID. An Adjacency Segment Identifier (Adj-SID) represents a router adjacency in Segment Routing.

A segment routing-capable router may allocate an Adj-SID for each of its adjacencies and an Adj-SID sub-TLV is defined to carry this SID in the Extended Opaque Link LSA.

OSPF allocates the adjacency SID for each OSPF neighbor if the OSPF adjacency which are in two way or in FULL state. OSPF allocates the adjacency SID only if the segment routing is enabled. The label for adjacency SID is dynamically allocated by the system. This eliminates the chances of misconfiguration, as this has got only the local significance.

Connected Prefix-SID

OSPFv2 supports the advertisement of prefix SID for address associated with the loopback interfaces. In order to achieve this, OSPF uses Extended Prefix Sub TLV in its opaque Extended prefix LSA. When OSPF receives this LSA from its neighbor, SR label is added to the RIB corresponding to received prefix based upon the information present in extended prefix sub TLV.

For configuration, segment-routing has to be enabled under OSPF and corresponding to loopback interface that is configured with OSPF, prefix-sid mapping is required under the segment routing module.

Note SID will only be advertised for loopback addresses and only for intra-area and inter-area prefix types. No SID value will be advertised for external or NSSA prefixes.

Prefix Propagation Between Areas

To provide segment routing support across the area boundary, OSPF is required to propagate SID values between areas. When OSPF advertises the prefix reachability between areas, it checks if the SID has been advertised for the prefix. In a typical case, the SID value come from the router, which contributes to the best path to the prefix in the source area. In this case, OSPF uses such SID and advertises it between the areas. If the SID value is not advertised by the router which contributes to the best path inside the area, OSPF will use the SID value coming from any other router inside the source area.

Segment Routing Global Range Changes

OSPF advertises it's segment routing capability in terms of advertising the SID/Label Range TLV. In OSPFv2, SID/Label Range TLV is a carried in Router Information LSA.

The segment routing global range configuration will be under the “segment-routing mpls” configuration. When the OSPF process comes, it will get the global range values from segment-routing and subsequent changes should be propagated to it.

When OSPF segment routing is configured, OSPF must request an interaction with the segment routing module before OSPF segment routing operational state can be enabled. If the SRGB range is not created, OSPF will not be enabled. When an SRGB change event occurs, OSPF makes the corresponding changes in it's sub-block entries.

Conflict Handling of SID Entries

In an ideal situation, each prefix should have unique SID entries assigned.

When there is a conflict between the SID entries and the associated prefix entries use any of the following methods to resolve the conflict:

- Multiple SIDs for a single prefix - If the same prefix is advertised by multiple sources with different SIDs, OSPF will install the unlabeled path for the prefix. The OSPF takes into consideration only those SIDs that are from reachable routers and ignores those from unreachable routers. When multiple SIDs are advertised for a prefix, which is considered as a conflict, no SID will be advertised to the attached-areas for the prefix. Similar logic will be used when propagating the inter-area prefixes between the backbone and the non-backbone areas.
- Out of Range SID - For SIDs that do not fit in our SID range, labels are not used while updating the RIB.

MPLS Forwarding on an Interface

MPLS forwarding must be enabled before segment routing can use an interface. OSPF is responsible for enabling MPLS forwarding on an interface.

When segment routing is enabled for a OSPF topology, or OSPF segment routing operational state is enabled, it enables MPLS for any interface on which the OSPF topology is active. Similarly, when segment routing is disabled for a OSPF topology, it disables the MPLS forwarding on all interfaces for that topology.

Configuring Segment Routing with OSPFv2

Configure segment routing with OSPFv2 protocol.

Before you begin

Confirm that the following conditions are met before configuring segment routing with OSPFv2:

- The OSPFv2 feature is enabled.
- The segment-routing feature is enabled.
- Segment routing is enabled under OSPF.

Procedure

	Command or Action	Purpose

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

Step 2	[**no**]**router ospf** *process* **Example:** `switch(config)# router ospf test`	Enables the OSPF mode.

	Command or Action	Purpose
Step 3	**segment-routing** **Example:** `switch(config-router)# segment-routing mpls`	Configures the segment routing functionality under OSPF.

Configuring Segment Routing on OSPF Network- Area Level

Before you begin

Before you configure segment routing on OSPF network, OSPF must be enabled on your network.

Procedure

	Command or Action	Purpose
Step 1	**router ospf process** **Example:** `switch(config)# router ospf test`	Enables the OSPF mode.
Step 2	**area** <*area id*> **segment-routing** [**mpls** | **disable**] **Example:** `switch(config-router)# area 1 segment-routing mpls`	Configures segment routing mpls mode in a specific area.
Step 3	[**no**]**area** <*area id*> **segment-routing** [**mpls** | **disable**] **Example:** `switch(config-router)#area 1 segment-routing disable`	Disables segment routing mpls mode for the specified area.
Step 4	**show ip ospf** *process* **segment-routing** **Example:** `switch(config-router)# show ip ospf test segment-routing`	Shows the output for configuring segment routing under OSPF.

Configuring Prefix-SID for OSPF

This task explains how to configure prefix segment identifier (SID) index under each interface.

Before you begin

Segment routing must be enabled on the corresponding address family.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 2	[**no**]**router ospf** *process* **Example:** `switch(config)# router ospf test`	Configures OSPF.

Step 3	**segment-routing**
	Example:
	`switch(config-router)# segment-routing`
	`switch(config-sr)#mpls`
	`switch(config-sr-mpls)#`
		Configures the segment routing functionality under OSPF.

Step 4	**interface loopback** *interface_number* **Example:** `switch(config-sr-mpls)# Interface loopback 0`	Specifies the interface where OSPF is enabled.
Step 5	**ip address 1.1.1.1/32** **Example:** `switch(config-sr-mpls)# ip address 1.1.1.1/32`	Specifies the IP address configured on the ospf interface.
Step 6	**ip router ospf 1 area 0** **Example:** `switch(config-sr-mpls)# ip router ospf 1 area 0`	Specifies the OSPF enabled on the interface in area.

Step 7	**segment-routing**
	Example:
	`switch(config-router)#segment-routing`
	`(config-sr)#mpls`
		Configures prefix-sid mapping under SR module.

Step 8	**connected-prefix-sid-map**
	Example:
	`switch(config-sr-mpls)# connected-prefix-sid-map`
	`switch(config-sr-mpls-conn-pfxsid)#`
		Configures the prefix SID mapping under the segment routing module.

Step 9	**address-family ipv4**
	Example:
	`switch(config-sr-mpls-conn-pfxsid)# address-family ipv4`
	`switch(config-sr-mpls-conn-pfxsid-af)#`
		Specifies the IPv4 address family configured on the OSPF interface.

	Command or Action	Purpose
Step 10	**1.1.1.1/32 index 10** **Example:** `switch(config-sr-mpls-conn-af)# 1.1.1.1/32 index 10`	Associates SID 10 with the address 1.1.1.1/32.
Step 11	**exit** **Example:** `switch(config-sr-mpls-conn-af)# exit`	Exits segment routing mode and returns to the configuration terminal mode.

Configuring Prefix Attribute N-flag-clear

OSPF advertises prefix SIDs via Extended Prefix TLV in its opaque LSAs. It carries flags for the prefix and one of them is N flag (Node) indicating that any traffic sent along to the prefix is destined to the router originating the LSA. This flag typically marks host routes of router's loopback.

Procedure

	Command or Action	Purpose

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

Step 2	**interface loopback3** **Example:** `switch(config)# interface loopback3`	Specifies the interface loopback.
Step 3	**ip ospf prefix-attributes n-flag-clear** **Example:** `switch#(config-if)# ip ospf prefix-attributes n-flag-clear`	Clears the prefix N-flag.

Configuration Examples for Prefix SID for OSPF

This example shows the configuration for prefix SID for OSPF.

```
Router ospf 10
     Segment-routing mpls
Interface loop 0
    Ip address 1.1.1.1/32
    Ip router ospf 10 area 0
Segment-routing
   Mpls
      connected-prefix-sid-m
          address-family ipv4
              1.1.1.1/32 index 10
```

Configuring Egress Peer Engineering With Segment Routing

BGP Prefix SID

In order to support segment routing, BGP requires the ability to advertise a segment identifier (SID) for a BGP prefix. A BGP prefix SID is always global within the segment routing BGP domain and identifies an instruction to forward the packet over the ECMP-aware best path computed by BGP to the related prefix. The BGP prefix SID identifies the BGP prefix segment.

Adjacency SID

The adjacency segment Identifier (SID) is a local label that points to a specific interface and a next hop out of that interface. No specific configuration is required to enable adjacency SIDs. Once segment routing is enabled over BGP for an address family, for any interface that BGP runs over, the address family automatically allocates an adjacency SID toward every neighbor out of that interface.

High Availability for Segment Routing

In-service software upgrades (ISSUs) are minimally supported with BGP graceful restart. All states (including the segment routing state) must be relearned from the BGP router's peers. During the graceful restart period, the previously learned route and label state are retained.

Overview of BGP Egress Peer Engineering With Segment Routing

Cisco Nexus 3000 Series switches are often deployed in massive scale data centers (MSDCs). In such environments, there is a requirement to support BGP Egress Peer Engineering (EPE) with Segment Routing (SR).

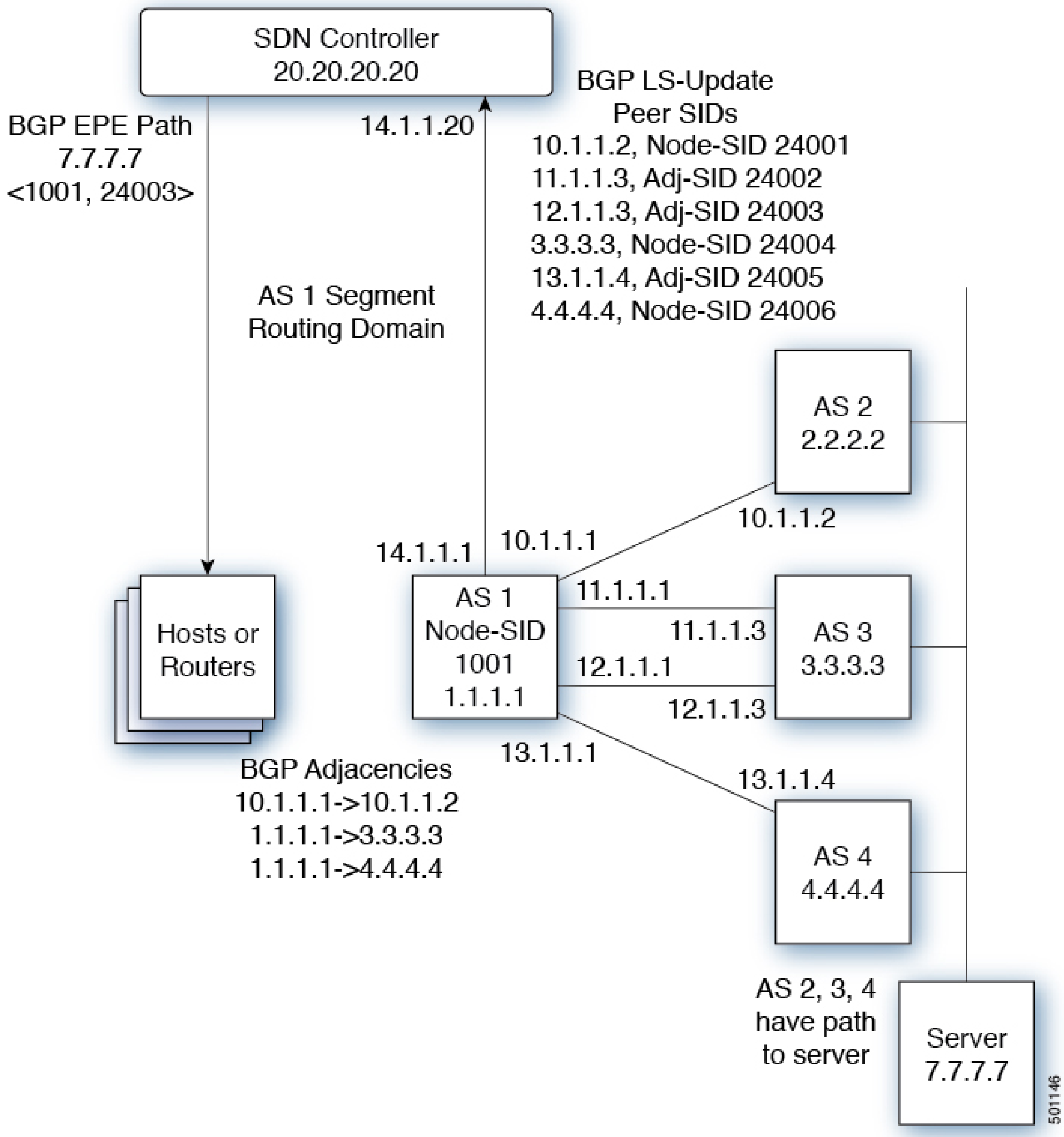
Segment Routing (SR) leverages source routing. A node steers a packet through a controlled set of instructions, known as segments, by prepending the packet with an SR header. A segment can represent any topological or service-based instruction. SR allows steering a flow through any topological path or any service chain while maintaining per-flow state only at the ingress node of the SR domain. For this feature, the Segment Routing architecture is applied directly to the MPLS data plane.

In order to support Segment Routing, BGP requires the ability to advertise a Segment Identifier (SID) for a BGP prefix. A BGP-Prefix is always global within the SR or BGP domain and it identifies an instruction to forward the packet over the ECMP-aware best-path that is computed by BGP to the related prefix. The BGP-Prefix-SID is the identifier of the BGP prefix segment.

The SR-based Egress Peer Engineering (EPE) solution allows a centralized (SDN) controller to program any egress peer policy at ingress border routers or at hosts within the domain.

In the following example, all three routers run iBGP and they advertise NRLI to one another. The routers also advertise their loopback as the next-hop and it is recursively resolved. This provides an ECMP between the routers as displayed in the illustration.

Figure 2: Example of Egress Peer Engineering



The SDN controller receives the Segment IDs from the egress router 1.1.1.1 for each of its peers and adjacencies. It can then intelligently advertise the exit points to other routers and hosts within the controller's routing domain. As displayed in the illustration, the BGP Network Layer Reachability Information (NLRI) contains both the Node-SID to the Router 1.1.1.1 and the Peer-Adjacency-SID 24003, indicating that the traffic to 7.7.7.7 should egress over 12.1.1.1->12.1.1.3.

Guidelines and Limitations for BGP Egress Peer Engineering

See the following guidelines and limitations for BGP Egress Peer Engineering:

- BGP Egress Peer Engineering is only supported for IPv4 BGP peers. IPv6 BGP peers are not supported.
- BGP Egress Peer Engineering is only supported in the default VPN Routing and Forwarding (VRF) instance.
- Any number of Egress Peer Engineering (EPE) peers may be added to an EPE peer set. However, the installed resilient per-CE FEC is limited to 32 peers.
- A given BGP neighbor can only be a member of a single peer-set. Peer-sets are configured. Multiple peer-sets are not supported. An optional **peer-set** name may be specified to add neighbor to a peer-set. The corresponding RPC FEC load-balances the traffic across all the peers in the peer-set. The peer-set name is a string that is a maximum length of 63 characters (64 NULL terminated). This length is consistent with the NX-OS policy name lengths. A peer can only be a member of a single peer-set.
- Adjacencies for a given peer are not separately assignable to different peer-sets.

Configuring Neighbor Egress Peer Engineering Using BGP

You can configure Egress Peer Engineering (EPE). The feature is valid only for external BGP neighbors and is not configured by default. EPE uses RFC 7752 encoding.

Before you begin

- You must enable BGP.
- After an upgrade, configure the TCAM region before configuring Egress Peer Engineering (EPE) on Cisco Nexus 3000 Series switches using the following commands:
  1. switch# **hardware access-list tcam region vpc-convergence 0**
  2. switch# **hardware access-list tcam region racl 0**
  3. switch# **hardware access-list tcam region mpls 256 double-wide**

For more information, see the Using Templates to Configure ACL TCAM Region Sizes and Configuring ACL TCAM Region Sizes sections in the *Cisco Nexus 3000 Series NX-OS Security Configuration Guide.*.

Procedure

	Command or Action	Purpose

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

Step 2	**router bgp** *<bgp autonomous number>*	Specifies the autonomous router BGP number.
Step 3	**neighbor** *<IP address>*	Configures the IP address for the neighbor.

	Command or Action	Purpose

Step 4	[**no**|**default**] **egress-engineering** [**peer-set** *peer-set-name*]
	Example:
	`switch(config)# router bgp 64497`
	`switch(config-router)# neighbor 30.1.1.1`
	`switch(config-router)# egress-engineering peer-set NewPeer`
		Specifies whether a Peer-Node-SID is allocated for the neighbor and advertised in an instance of a BGP Link-State (BGP-LS) address family Link NLRI. If the neighbor is a multi-hop neighbor, a BGP-LS Link NLRI instance will also be advertised for each Equal-Cost-MultiPath (ECMP) path to the neighbor and will include a unique Peer-Adj-SID.
		Optionally, you can add the neighbor to a peer-set. The Peer-Set-SID will also be advertised in the BGP-LS Link NLRI in the same instance as the Peer-Node-SID. BGP Link-State NLRI will be advertised to all neighbors with the link-state address family configured.
		See RFC 7752 and draft-ietf-idr-bgpls-segment-routing-epe-05 for more information on EPE.

Configuration Example for Egress Peer Engineering

See the Egress Peer Engineering sample configuration for the BGP speaker 1.1.1.1. Note that the neighbor 20.20.20.20 is the SDN controller.

```
 hostname epe-as-1
install feature-set mpls
feature-set mpls

feature telnet
feature bash-shell
feature scp-server
feature bgp
feature mpls segment-routing

segment-routing mpls
vlan 1

vrf context management
  ip route 0.0.0.0/0 10.30.97.1
  ip route 0.0.0.0/0 10.30.108.1

interface Ethernet1/1
  no switchport
  ip address 10.1.1.1/24
  no shutdown

interface Ethernet1/2
  no switchport
  ip address 11.1.1.1/24
  no shutdown

interface Ethernet1/3
  no switchport
```

```
  ip address 12.1.1.1/24
  no shutdown

interface Ethernet1/4
  no switchport
  ip address 13.1.1.1/24
  no shutdown

interface Ethernet1/5
  no switchport
  ip address 14.1.1.1/24
  no shutdown

interface mgmt0
  ip address dhcp
  vrf member management


interface loopback1
  ip address 1.1.1.1/32
line console

line vty
ip route 2.2.2.2/32 10.1.1.2
ip route 3.3.3.3/32 11.1.1.3
ip route 3.3.3.3/32  12.1.1.3
ip route 4.4.4.4/32  13.1.1.4
ip route 20.20.20.20/32 14.1.1.20

router bgp 1
  address-family ipv4 unicast
  address-family link-state
 neighbor 10.1.1.2
    remote-as 2
    address-family ipv4
    egress-engineering
 neighbor 3.3.3.3
   remote-as 3
   address-family ipv4
   update-source loopback1
   ebgp-multihop 2
   egress-engineering
 neighbor 4.4.4.4
   remote-as 4
   address-family ipv4
   update-source loopback1
   ebgp-multihop 2
   egress-engineering
neighbor 20.20.20.20
   remote-as 1
   address-family link-state
   update-source loopback1
   ebgp-multihop 2
```

Configuring EVPN Over Segment Routing or MPLS

You can configure EVPN over segment routing or MPLS.

Follow these steps to configure EVPN over segment routing or MPLS:

Before you begin

You must enable the command **feature interface-vlan** before configuring EVPN over segment routing or MPLS.

Procedure

	Command or Action	Purpose
Step 1	**feature bgp**	Enables BGP feature and configurations.
Step 2	**install feature-set mpls**	Enables MPLS configuration commands.
Step 3	**feature-set mpls**	Enables MPLS configuration commands.
Step 4	**feature mpls segment-routing**	Enables segment routing configuration commands.
Step 5	**feature mpls evpn**	Enables EVPN over MPLS configuration commands. This command is mutually exclusive with the **feature-nv** command.

Example

This example shows how to configure a VRF:

```
vrf context customer1
    rd auto
    address-family ipv4 unicast
        route-target import auto
        route-target export auto
        route-target import auto evpn
        route-target export auto evpn
```

This example shows how to configure a SRBGP over segment routing:

```
mpls label range 1000 25000
 segment-routing mpls
  global-block 11000 20000
!
int lo1
  ip address 200.0.0.1/32
!
interface e1/13
  description “MPLS interface towards Core”
  ip address 192.168.5.1/24
  mpls ip forwarding
  no shut
route-map label_index_pol_100 permit 10
  set label-index 100
route-map label_index_pol_101 permit 10
  set label-index 101
route-map label_index_pol_102 permit 10
  set label-index 102
route-map label_index_pol_103 permit 10
  set label-index 103

router bgp 65000
 address-family ipv4 unicast
   network 200.0.0.1/32 route-map label_index_pol_100
```

```
   network 192.168.5.1/32 route-map label_index_pol_101
   network 101.0.0.0/24 route-map label_index_pol_103
   allocate-label all
neighbor 192.168.5.6 remote-as 65000
    address-family ipv4 labeled-unicast
       send-community extended
```

Configuring the BGP Link State Address Family

You can configure the BGP link state address family for a neighbor session with a controller to advertise the corresponding SIDs. You can configure this feature in global configuration mode and neighbor address family configuration mode.

Before you begin

You must enable BGP.

Procedure

	Command or Action	Purpose

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

Step 2	**router bgp** <*bgp autonomous number*>	Specifies the autonomous router BGP number.

Step 3	[**no**] **address-family link-state**
	Example:
	`switch(config)# router bgp 64497`
	`switch (config-router af)# address-family link-state`
		Enters address-family interface configuration mode.
		Note This command can also be configured in neighbor address-family configuration mode.

Step 4	**neighbor** <*IP address*>	Configures the IP address for the neighbor.

Step 5	[**no**] **address-family link-state**
	Example:
	`switch(config)#router bgp 1`
	`switch(config-router)#address-family link-state`
	`switch(config-router)#neighbor 20.20.20.20`
	`switch(config-router)#address-family link-state`
		Enters address-family interface configuration mode.
		Note This command can also be configured in neighbor address-family configuration mode.

Configuring MPLS Label Allocation

You can configure MPLS label allocation for the IPv4 unicast address family.

Before you begin

You must install and enable the MPLS feature set using the **install feature-set mpls** and **feature-set mpls** commands.

You must enable the MPLS segment routing feature.

Procedure

	Command or Action	Purpose

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

Step 2	[**no**] **router bgp** *autonomous-system-number*
	Example:
	`switch(config)# router bgp 64496`
	`switch(config-router)#`
		Enables BGP and assigns the AS number to the local BGP speaker. The AS number can be a 16-bit integer or a 32-bit integer in the form of a higher 16-bit decimal number and a lower 16-bit decimal number in xx.xx format.
		Use the **no** option with this command to remove the BGP process and the associated configuration.

Step 3	Required: **address-family ipv4 unicast**
	Example:
	`switch(config-router)# address-family ipv4 unicast`
	`switch(config-router-af)#`
		Enters global address family configuration mode for the IPv4 address family.

Step 4	[**no**] **allocate-label** {**all** | **route-map** *route-map-name*} **Example:** `switch(config-router-af)# allocate-label route-map map1`	Configures local label allocation for routes matching the specified route map or for all routes advertised in this address family.

Step 5	Required: **exit**
	Example:
	`switch(config-router-af)# exit`
	`switch(config-router)#`
		Exits global address family configuration mode.

Step 6	**neighbor** *ipv4-address* **remote-as** *autonomous-system-number*
	Example:
	`switch(config-router)# neighbor 10.1.1.1 remote-as 64497`
	`switch(config-router-neighbor)#`
		Configures the IPv4 address and AS number for a remote BGP peer.

Step 7	**address-family ipv4 labeled-unicast** **Example:**	Advertises the labeled IPv4 unicast routes as specified in RFC 3107.

	Command or Action	Purpose
	`switch(config-router-neighbor)# address-family ipv4 labeled-unicast switch(config-router-neighbor-af)#`	
Step 8	(Optional) **show bgp ipv4 labeled-unicast** *prefix* **Example:** `switch(config-router-neighbor-af)# show bgp ipv4 labeled-unicast 10.10.10.10/32`	Displays the advertised label index and the selected local label for the specified IPv4 prefix.
Step 9	(Optional) **copy running-config startup-config** **Example:** `switch(config-router-neighbor-af)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring BGP EVPN and Label Allocation Mode

You can use MPLS tunnel encapsulation using the **encapsulation mpls** command. You can configure the label allocation mode for the EVPN address family.

Advertisement of (IP or Label) bindings from a Cisco Nexus 9000 Series switch via BGP EVPN enables a remote switch to send the routed traffic to that IP using the label for that IP to the switch that advertised the IP over MPLS.

The IP prefix routes (Type-5) are:

- Type-5 route with VXLAN encapsulation

```
RT-5 Route – IP Prefix

RD:  L3 RD
IP Length:  prefix length
IP address:  IP (4 bytes)
Label1:  L3VNI
Route Target

RT for IP-VRF
Tunnel Type VxLAN
Router MAC
```

- Type-5 route with MPLS encapsulation

```
RT-5 Route – IP Prefix

RD:  L3 RD
IP Length:  prefix length
IP address:  IP (4 bytes)
Label1:  BGP MPLS Label
Route Target
RT for IP-VRF
```

VPN labels can be per-prefix or aggregate (per-VRF). It is recommended to have per-VRF label for Layer 3 EVPN over MPLS.

Complete the following steps to configure BGP EVPN and label allocation mode:

Before you begin

You must install and enable the MPLS feature set using the **install feature-set mpls** and **feature-set mpls** commands.

You must enable the MPLS segment routing feature. See About Segment Routing, on page 31.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal**	Enters global configuration mode.

Step 2	[**no**] **router bgp** *autonomous-system-number*
	Example:
	`switch(config)# router bgp 64496`
	`switch(config-router)#`
		Enables BGP and assigns the AS number to the local BGP speaker. The AS number can be a 16-bit integer or a 32-bit integer in the form of a higher 16-bit decimal number and a lower 16-bit decimal number in xx.xx format.
		Use the **no** option with this command to remove the BGP process and the associated configuration.

Step 3	Required: **address-family l2vpn evpn**
	Example:
	`switch(config-router)# address-family l2vpn evpn`
	`switch(config-router-af)#`
		Enters global address family configuration mode for the Layer 2 VPN EVPN.

Step 4	Required: **exit**
	Example:
	`switch(config-router-af)# exit`
	`switch(config-router)#`
		Exits global address family configuration mode.

Step 5	**neighbor** *ipv4-address* **remote-as** *autonomous-system-number*
	Example:
	`switch(config-router)# neighbor 10.1.1.1 remote-as 64497`
	`switch(config-router-neighbor)#`
		Configures the IPv4 address and AS number for a remote BGP peer.

Step 6	**address-family l2vpn evpn**
	Example:
	`switch(config-router-neighbor)# address-family l2vpn evpn`
	`switch(config-router-neighbor-af)#`
		Advertises the labeled Layer 2 VPN EVPN.

Step 7	**encapsulation mpls** **Example:**	Enables BGP EVPN address family and sends EVPN type-5 route update to the neighbors.

	Command or Action	Purpose

	<pre>router bgp 100
	  address-family l2vpn evpn
	neighbor NVE2 remote-as 100
	    address-family l2vpn evpn
	      send-community extended
	      encapsulation mpls
	  vrf foo
	    address-family ipv4 unicast
	      advertise l2vpn evpn</pre> BGP segment routing configuration: <pre>router bgp 100
	 address-family ipv4 unicast
	   network 200.0.0.1/32 route-map
	label_index_pol_100
	    network 192.168.5.1/32 route-map
	label_index_pol_101
	    network 101.0.0.0/24 route-map
	label_index_pol_103
	    allocate-label all
	  neighbor 192.168.5.6 remote-as 20
	      address-family ipv4
	labeled-unicast
	         send-community extended</pre>
		Note The default tunnel encapsulation in EVPN for the IP route type in NX-OS is VXLAN. To override that, a new CLI is introduced to indicate MPLS tunnel encapsulation.

Step 8	**vrf** <*customer_name*>	Configures the VRF.
Step 9	**send-community**	To send the Border Gateway Protocol (BGP) community attribute to a peer, use the send-community command. To revert to the defaults, use the no form of this command.
Step 10	**send-community** *extended*	Provides an extended range to the send-community command, ensuring that communities can be assigned for many purposes, without overlap.
Step 11	**address-family ipv4 unicast**	Enters global address family configuration mode for the IPv4 address family.
Step 12	**advertise l2vpn evpn**	Advertises Layer 2 VPN EVPN.
Step 13	**redistribute direct route-map DIRECT_TO_BGP**	Redistributes the directly connected routes into BGP-EVPN.
Step 14	**label-allocation-mode per-vrf**	Sets the label allocation mode. The default label allocation is per VRF. You have to configure per-vrf label allocation mode explicitly. For the EVPN address family, the default label allocation is per-vrf, compared to per-prefix mode for the other address-families where the

	Command or Action	Purpose
		label allocation CLI is supported. No form of CLI is displayed in the running configuration.

Example

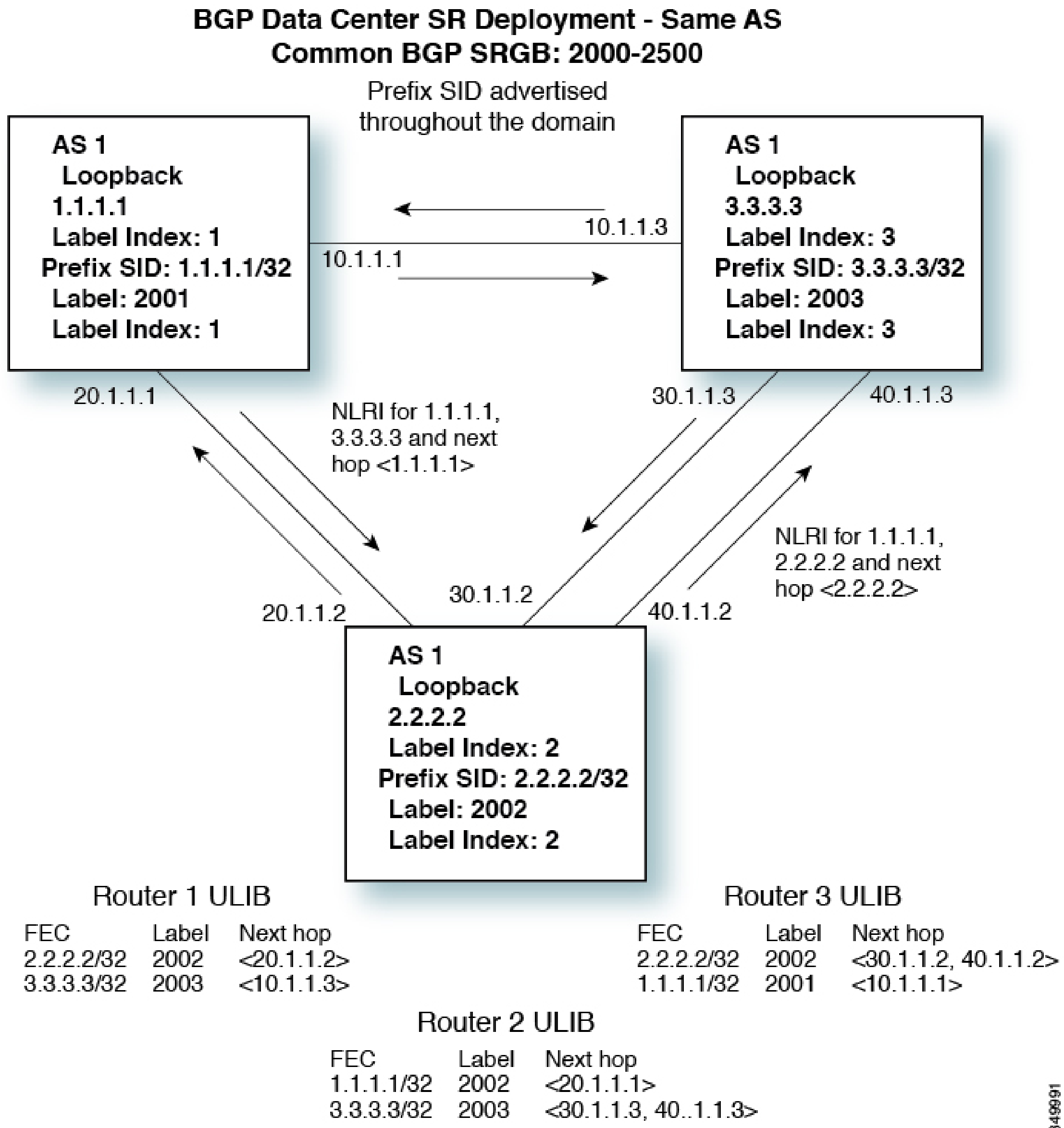
See the following example for configuring per-prefix label allocation:

```
router bgp 65000
    [address-family l2vpn evpn]
    neighbor 10.1.1.1
        remote-as 100
        address-family l2vpn evpn
    neighbor 20.1.1.1
        remote-as 65000
        address-family l2vpn evpn
 encapsulation mpls
    vrf customer1
        address-family ipv4 unicast
            advertise l2vpn evpn
            redistribute direct route-map DIRECT_TO_BGP
            no label-allocation-mode per-vrf
```

BGP Prefix SID Deployment Example

In the simple example below, all three routers are running iBGP and advertising Network Layer Reachability Information (NRLI) to one another. The routers are also advertising their loopback interface as the next hop, which provides the ECMP between routers 2.2.2.2 and 3.3.3.3.

Figure 3: BGP Prefix SID Simple Example



Verifying the Segment Routing Configuration

To display the segment routing configuration, perform one of the following tasks:

Command	Purpose
show bgp ipv4 labeled-unicast *prefix*	Displays the advertised label index and the selected local label for the specified IPv4 prefix.

Command	Purpose
show bgp paths	Displays the BGP path information, including the advertised label index.
show mpls label range	Displays the configured SRGB range of labels.
show route-map [*map-name*]	Displays information about a route map, including the label index.
show running-config | inc 'feature segment-routing'	Displays the status of the MPLS segment routing feature.
show ip ospf neighbors detail	Displays the list of OSPFv2 neighbors and the adjacency SID allocatte, laong with the corresponding flags.
show ip ospf database opaque-area	Displays the LSAs for the adjacency SID.
show ip ospf segment-routing adj-sid-database	Displays all locally allocated adjacency SIDs.
show running-config segment-routing	Displays the status of the segment routing feature.
show segment-routing mpls	Displays segment routing mpls information
show segment-routing mpls clients	Displays the clients registered with the SR-APP.
show segment-routing mpls details	Displays detailed information.
show segment-routing ipv4	Displays the information for the IPv4 address family.
show segment-routing ipv4 connected-prefix-sid	Displays the MPLS label range for the SRGB. **Note** This command is only available in Cisco NX-OS Release 9.3(1) .
show ip ospf *process*	Displays the OSPF mode.
show ip ospf *process* **segment-routing sid-database**	Displays the segment routing database details.
show ip ospf *process* **segment-routing global block**	Displays the segment routing global block information.

Additional References

Related Documents

Related Topic	Document Title
BGP	*Cisco Nexus 3000 Series Unicast Routing Configuration Guide*

CHAPTER 6

Configuring MPLS Segment Routing OAM

This chapter contains information about configuring MPLS Segment Routing OAM.

About MPLS Segment Routing OAM

MPLS segment routing (SR) has been deployed on the Cisco Nexus 3000 Series switches. As MPLS segment routing (SR) is deployed, a few diagnostic tools are required to help resolve the misconfigurations or failures in the segment routing network. Segment Routing Operations, Adminstration, and Maintenance (OAM) helps service providers monitor label-switched paths (LSPs) and quickly isolate forwarding problems to assist with fault detection and troubleshooting in the network.

MPLS SR OAM provides two main functions for diagnostics purposes:

1. MPLS ping
2. MPLS traceroute

The segment routing OAM feature provides support for the following FEC types:

- Ping and traceroute to SR-IGP IS-IS IPv4 prefixes. This allows validation of prefix SIDs distributed in an IS-IS SR underlay.
- Ping and traceroute to BGP IPv4 prefixes. This allows validation of prefix SIDs distributed in a BGP SR underlay.
- Ping and traceroute to Generic IPv4 prefixes. This allows validation of prefix SIDs distributed in an SR underlay agnostic to the protocol that performed the distribution. The validation is performed by checking the Unicast Routing Information Base (URIB) and Unicast Label Information Base (ULIB).

To enable MPLS OAM on Cisco Nexus 3000 Series switches, use the **feature mpls oam** CLI command. Use the **no feature mpls oam** CLI command to disable MPLS OAM on Cisco Nexus 3000 Series switches.

Segment Routing Ping

Similar to how an IP ping validates connectivity to an IP host, MPLS ping is used to validate unidirectional continuity along an MPLS Label-Switched Path (LSP). By providing a FEC representing the LSP to be validated, MPLS ping performs the following:

- Confirms that the echo requests for the FEC reach an endpoint for the LSP. Except for the Nil FEC, for all other FEC types it confirms that the endpoint is the correct egress for that FEC.
- Measures coarse round trip time.
- Measures coarse round trip delay.

The MPLS LSP ping feature is used to check the connectivity between ingress Label Switch Routers (LSRs) and egress LSRs along an LSP. MPLS LSP ping uses MPLS echo request and reply messages, similar to Internet Control Message Protocol (ICMP) echo request and reply messages, to validate an LSP. The destination IP address of the MPLS echo request packet is different from the address used to select the label stack. The destination IP address is defined as a 127.x.y.z/8 address and it prevents the IP packet from being IP switched to its destination, if the LSP is broken.

Segment Routing Traceroute

MPLS traceroute verifies forwarding and control plane at each hop of the LSP to isolate faults. Traceroute sends MPLS echo requests with monotonically increasing time-to-live (TTL), starting with TTL of 1. Upon TTL expiry, transit node processes the request in software and verifies if it has an LSP to the target FEC and intended transit node. The transit node sends echo reply containing return code specifying the result of above verification and label stack to reach the next-hop, as well as ID of the next-hop towards destination, if verification is successful. Originator processes echo reply to build the next echo request containing TTL+1. This process is repeated until the destination replies that it is the egress for the FEC.

The MPLS LSP traceroute feature is used to isolate the failure point of an LSP. It is used for hop-by-hop fault localization and path tracing. The MPLS LSP Traceroute feature relies on the expiration of the Time to Live (TTL) value of the packet that carries the echo request. When the MPLS echo request message hits a transit node, it checks the TTL value and if it is expired, the packet is passed to the control plane, else the message is forwarded. If the echo message is passed to the control plane, a reply message is generated based on the contents of the request message

Guidelines and Limitations for MPLS Segment Routing OAM

MPLS OAM Nil FEC has the following guidelines and limitations:

- A maximum of 4 labels can be specified in the **ping mpls nil-fec** and **traceroute mpls nil-fec** commands. This value is enforced by querying the platform and currently Cisco Nexus 3000 Series switches limit the label stack to 5. It means that for a Nil-FEC echo request, you can specify a maximum of 4 labels because internally an extra explicit-null is added.
- The nexthop specified in the ping and traceroute commands must be a connected nexthop on the originator and it should not be a recursive nexthop.
- There is no support for treetrace.

- Nil-FEC does not carry any information to identify the intended target. The packet may mis-forward at an incorrect node but the validation may return success if the packet ends up at a node after popping the non-null labels.
- Nil FEC operates on forwarding the information alone. It cannot detect the inconsistencies between the control plane and the forwarding plane by definition.
- Nil FEC ping and traceroute is not supported for deaggregator (per-VRF) labels. This includes the BGP EVPN-Layer 3 deaggregator labels.
- On Cisco Nexus 3000 Series switches that use Broadcom chipsets, there is no support to allow the software to send a query to determine which ECMP a packet takes. It means that for MPLS traceroutes that traverse one of these switches may display an error at the next hop if there is more than one ECMP as displayed in the following example:

  ```
  D 2 6.0.0.2 MRU 1496 [Labels: 2003/explicit-null Exp: 0/0] 4 ms
  ```

- When you use OAM to test a BGP EPE LSP (for example, the last label in the ping/traceroute label stack is an EPE label), the OAM will return a success if the final router has OAM enabled and MPLS is enabled on the incoming interface.

  For example, if you have a setup as A---B---C, A and B are in the SR network, and B acts like a PE and C acts like a CE, B is configured with C as a BGP EPE peer (using egress-engineering on B), then C must have OAM and MPLS forwarding enabled on the incoming interface.
- For all new FEC types supported in Cisco NX-OS Release 9.3(1), only a one-label stack is supported. FEC-Stack change TLV support and the associated validations are not supported. This limitation is not applicable to Nil-FEC.
- In Cisco NX-OS Release 9.3(1), the SR-IGP "any" prefix type and the adjacency SIDs described in RFC 8287 are not supported.
- OSPF ping and traceroute is not supported in Cisco NX-OS Release 9.3(1).

MPLS Ping and Traceroute for Nil FEC

The Nil FEC LSP ping and traceroute operations are extensions of regular MPLS ping and traceroute. The Nil FEC LSP ping and traceroute functionality supports segment routing and MPLS Static. It also acts as an additional diagnostic tool for all other LSP types.

Unlike the other FEC types, Nil FEC does not provide control plane validation. Nil FEC ping or traceroute probes can reach any switch on which the MPLS OAM functionality is enabled.

This feature allows operators to provide the ability to freely test any label stack by allowing them to specify the following:

- Label stack
- Outgoing interface
- Nexthop address

In case of segment routing, each segment nodal label and adjacent label along the routing path is put into the label stack of an echo request message from the initiator Label Switch Router (LSR); MPLS data plane forwards this packet to the label stack target, and the label stack target sends the echo message back.

Use the [**ping|traceroute**] **mpls nil-fec labels** *comma-separated-labels* [**output** {**interface** *tx-interface*} [**nexthop** *nexthop-ip-addr*]] CLI command to execute a ping or a traceroute.

MPLS Ping and Traceroute for BGP and IGP Prefix SID

MPLS ping and traceroute operations for Prefix SID are supported for the following BGP and IGP scenarios:

- Within an IS-IS level
- Across IS-IS levels
- BGP SR underlay

These FEC types perform an additional control plane check to ensure that the packets are not mis-routed. This validation ensures that the pinged FEC type is connected to the switch and is distributed to the other nodes. Nil FEC does not provide this validation.

MPLS echo request packets carry Target FEC Stack sub-TLVs. The Target FEC sub-TLVs are used by the responder for FEC validation. The IGP/BGP IPv4 prefix sub-TLV has been added to the Target FEC Stack sub-TLV. The IGP/BGP IPv4 prefix sub-TLV contains the prefix SID, the prefix length, and the protocol (IS-IS).

Use the **ping|traceroute sr-mpls** A.B.C.D/LEN fec-type [*bgp* | *igp {isis}* | *generic*] CLI command to execute a traceroute.

Verifying Segment Routing OAM

This section provides information on the CLI commands that can be used to verify the segment routing OAM features.

- Verifying Segment Routing OAM IS-IS, on page 68

Verifying Segment Routing OAM IS-IS

The following ping commands are used to display SR OAM when the underlying network is IS-IS:

```
switch# ping sr-mpls 11.1.1.3/32 fec-type igp isis

Sending 5, 100-byte MPLS Echos to IGP Prefix SID(IS-IS) FEC 11.1.1.3/32,
     timeout is 2 seconds, send interval is 0 msec:

Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no label entry,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'X' - unknown return code, 'x' - return code 0

Type Ctrl-C to abort.
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 2/2/3 ms
 Total Time Elapsed 18 ms
```

```
switch# traceroute sr-mpls 11.1.1.3/32 fec-type igp isis


Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no label entry,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'X' - unknown return code, 'x' - return code 0

Type Ctrl-C to abort.
  0 172.18.1.2 MRU 1500 [Labels: 16103 Exp: 0]
L 1 172.18.1.1 MRU 1504 [Labels: implicit-null Exp: 0] 4 ms
! 2 172.18.1.10 3 ms
```

```
switch# ping sr-mpls 11.1.1.3/32 fec-type igp isis verbose

Sending 5, 100-byte MPLS Echos to IGP Prefix SID(IS-IS) FEC 11.1.1.3/32,
     timeout is 2 seconds, send interval is 0 msec:

Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no label entry,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'X' - unknown return code, 'x' - return code 0

Type Ctrl-C to abort.
!    size 100, reply addr 172.18.1.10, return code 3
!    size 100, reply addr 172.18.1.10, return code 3
!    size 100, reply addr 172.18.1.10, return code 3
!    size 100, reply addr 172.18.1.10, return code 3
!    size 100, reply addr 172.18.1.10, return code 3

Success rate is 100 percent (5/5), round-trip min/avg/max = 2/2/3 ms
 Total Time Elapsed 17 ms
```

```
switch# ping sr-mpls 11.1.1.3/32 fec-type igp isis destination 127.0.0.1 127.0.0.2 repeat
1 verbose

Sending 1, 100-byte MPLS Echos to IGP Prefix SID(IS-IS) FEC 11.1.1.3/32,
     timeout is 2 seconds, send interval is 0 msec:

Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no label entry,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'X' - unknown return code, 'x' - return code 0

Type Ctrl-C to abort.
Destination address 127.0.0.1
!    size 100, reply addr 172.18.1.10, return code 3

Destination address 127.0.0.2
!    size 100, reply addr 172.18.1.22, return code 3

Success rate is 100 percent (2/2), round-trip min/avg/max = 3/3/3 ms
 Total Time Elapsed 8 ms
```

Examples for using Ping and Traceroute CLI commands

Examples for IGP or BGP SR Ping and Traceroute

Using CLI to Execute a Ping with Explicit Outgoing Information

Use the **ping sr-mpls** *fec* **fec-type igp isis** CLI command to execute an IS-IS SR ping and the **ping sr-mpls** *fec* **fec-type bgp** CLI command to execute a BGP ping.

```
switch# ping sr-mpls 11.1.1.3/32 fec-type igp isis

Sending 5, 100-byte MPLS Echos to IGP Prefix SID(IS-IS) FEC 11.1.1.3/32,
     timeout is 2 seconds, send interval is 0 msec:

Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no label entry,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'X' - unknown return code, 'x' - return code 0

Type Ctrl-C to abort.
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 2/2/3 ms
 Total Time Elapsed 18 ms
```

```
switch# ping sr-mpls 11.1.1.3/32 fec-type igp isis verbose

Sending 5, 100-byte MPLS Echos to IGP Prefix SID(IS-IS) FEC 11.1.1.3/32,
     timeout is 2 seconds, send interval is 0 msec:

Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no label entry,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'X' - unknown return code, 'x' - return code 0

Type Ctrl-C to abort.
!    size 100, reply addr 172.18.1.10, return code 3
!    size 100, reply addr 172.18.1.10, return code 3
!    size 100, reply addr 172.18.1.10, return code 3
!    size 100, reply addr 172.18.1.10, return code 3
!    size 100, reply addr 172.18.1.10, return code 3

Success rate is 100 percent (5/5), round-trip min/avg/max = 2/2/3 ms
 Total Time Elapsed 17 ms
```

Examples for Nil FEC Ping and Traceroute

Using CLI to Execute a Ping with Explicit Outgoing Information

Use the **ping sr-mpls nil-fec labels** *comma-separated-labels* [**output** {**interface** *tx-interface*} [**nexthop** *nexthop-ip-addr*]] CLI command to execute a ping.

For example, the following command sends an MPLS packet with the outermost two labels in the label stack being 2001 and 2000 out the interface Ethernet 1/1 with a nexthop IP address of 4.0.0.2:

```
switch# ping mpls nil-fec labels 2001,2000 output interface e1/1 nexthop 4.0.0.2
```

It is mandatory that the nexthop is a connected nexthop; it is not recursively resolved.

The above CLI format is a simplified version. The [**output** {**interface** *tx-interface*} [**nexthop** *nexthop-ip-addr*]] is mandatory to be present in the VSH server. For example:

```
switch# ping mpls nil-fec labels 1,2 ?
output Output options
switch# ping mpls nil-fec labels1,2
^
% Invalid command at '^' marker.
```

Using CLI to Execute a Traceroute with Explicit Outgoing Information

Use the following CLI command to execute a traceroute:

```
switch# ping mpls nil-fec labels 2001,2000 output interface e1/1 nexthop 4.0.0.2
```

Displaying Show Statistics

Use the following command to display the statistics about the echo requests sent by the local MPLS OAM service:

```
show mpls oam echo statistics
```

CHAPTER 7

IETF RFCs Supported for Label Switching

This appendix lists the IETF RFCs supported for label switching on the device.

IETF RFCs Supported for Label Switching

This table lists the IETF RFCs supported for label switching on the device.

RFCs	Title
RFC 3107	*Carrying Label Information in BGP-4*
RFC 7752	*North-Bound Distribution of Link-State and Traffic Engineering (TE) Information Using BGP*
RFC 8029	*Detecting Multiprotocol Label Switched (MPLS) Data-Plane failures.*
RFC 8287	*Label Switched Path (LSP) Ping/Traceroute for Segment Routing (SR) IGP-Prefix and IGP-Adjacency Segment Identifiers (SIDs) with MPLS Data Planes.*
Draft-ietf-idr-bgpls-segment-routing-epe-05	*Segment Routing BGP Egress Peer Engineering BGP-LS Extensions draft-ietf-idr-bgpls-segment-routing-epe-05*

INDEX