



Cisco Nexus 3000 Series NX-OS Label Switching Configuration Guide, Release 9.2(x)

First Published: 2018-07-17

Last Modified: 2018-11-05

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

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: <http://www.cisco.com/go/trademarks>. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

© 2018 Cisco Systems, Inc. All rights reserved.



CONTENTS

PREFACE

| | |
|--|------------|
| Preface | vii |
| Audience | vii |
| Document Conventions | vii |
| Related Documentation for Cisco Nexus 3000 Series Switches | viii |
| Documentation Feedback | viii |
| Communications, Services, and Additional Information | viii |

CHAPTER 1

| | |
|------------------------------------|----------|
| New and Changed Information | 1 |
| New and Changed Information | 1 |

CHAPTER 2

| | |
|---|----------|
| Configuring Static MPLS | 3 |
| Licensing Requirements | 3 |
| About Static MPLS | 3 |
| Label Swap and Pop | 4 |
| Static MPLS Topology | 4 |
| Benefits of Static MPLS | 5 |
| Prerequisites for Static MPLS | 5 |
| Guidelines and Limitations for Static MPLS | 6 |
| Configuring Static MPLS | 7 |
| Enabling Static MPLS | 7 |
| Reserving Labels for Static Assignment | 8 |
| Configuring Static Label and Prefix Binding Using the Swap and Pop Operations | 9 |
| Verifying the Static MPLS Configuration | 10 |
| Displaying Static MPLS Statistics | 12 |
| Clearing Static MPLS Statistics | 14 |
| Configuration Examples for Static MPLS | 14 |

| | |
|-----------------------|----|
| Additional References | 15 |
| Related Documents | 15 |

CHAPTER 3

| | |
|--|-----------|
| Configuring MPLS Label Imposition | 17 |
| About MPLS Label Imposition | 17 |
| Guidelines and Limitations for MPLS Label Imposition | 18 |
| Configuring MPLS Label Imposition | 18 |
| Enabling MPLS Label Imposition | 18 |
| Reserving Labels for MPLS Label Imposition | 19 |
| Configuring MPLS Label Imposition | 20 |
| Verifying the MPLS Label Imposition Configuration | 22 |
| Displaying MPLS Label Imposition Statistics | 25 |
| Clearing MPLS Label Imposition Statistics | 26 |
| Configuration Examples for MPLS Label Imposition | 27 |

CHAPTER 4

| | |
|---|-----------|
| Configuring Segment Routing | 29 |
| About Segment Routing | 29 |
| BGP Prefix SID | 29 |
| Segment Routing Global Block | 30 |
| High Availability for Segment Routing | 30 |
| BGP Prefix SID Deployment Example | 30 |
| Guidelines and Limitations for Segment Routing | 31 |
| Overview of BGP Egress Peer Engineering With Segment Routing | 33 |
| Guidelines and Limitations for BGP Egress Peer Engineering | 35 |
| Configuring Segment Routing | 35 |
| Configuring Segment Routing Using Segment Routing Application | 35 |
| Enabling MPLS Segment Routing | 37 |
| Enabling MPLS on an Interface | 37 |
| Configuring EVPN Over Segment Routing or MPLS | 38 |
| Configuring Segment Routing with IS-IS Protocol | 40 |
| Configuring MPLS Label Allocation | 40 |
| Configuring BGP EVPN and Label Allocation Mode | 42 |
| Configuring the Segment Routing Global Block | 45 |
| Configuring the Label Index | 47 |

| | |
|--|----|
| Configuring BGP Link State Neighbor Address Family | 48 |
| Configuring Neighbor Egress Peer Engineering Using BGP | 49 |
| Configuration Example for Egress Peer Engineering | 50 |
| Verifying the Segment Routing Configuration | 52 |
| Configuration Examples for Segment Routing | 52 |
| Additional References | 57 |
| Related Documents | 57 |

CHAPTER 5**Configuring MPLS Segment Routing OAM 59**

| | |
|---|----|
| Overview of MPLS Segment Routing OAM | 59 |
| Segment Routing OAM Support for LSP Ping and Traceroute | 59 |
| Guidelines and Limitations | 60 |
| Examples for Using Ping and Traceroute CLI Commands | 61 |

CHAPTER 6**IETF RFCs Supported for Label Switching 63**

| | |
|---|----|
| IETF RFCs Supported for Label Switching | 63 |
|---|----|



Preface

This preface includes the following sections:

- [Audience, on page vii](#)
- [Document Conventions, on page vii](#)
- [Related Documentation for Cisco Nexus 3000 Series Switches, on page viii](#)
- [Documentation Feedback, on page viii](#)
- [Communications, Services, and Additional Information, on page viii](#)

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. |
| <i>Italic</i> | 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 |
|-----------------|---|
| <i>variable</i> | 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 |
|-----------------------------|---|
| <code>screen font</code> | Terminal sessions and information the switch displays are in screen font. |
| boldface screen font | Information you must enter is in boldface screen font. |
| <i>italic screen font</i> | 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 9.2(x)*.

- [New and Changed Information, on page 1](#)

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 9.2(x)* and tells you where they are documented.

Table 1: New and Changed Features for Cisco NX-OS Release 9.2(x)

| Feature | Description | Changed in Release | Where Documented |
|--|----------------------|--------------------|------------------|
| No updates since Cisco NX-OS Release 7.x | First 9.2(x) release | Not applicable | Not applicable |



CHAPTER 2

Configuring Static MPLS

This chapter contains information about configuring static MPLS.

- [Licensing Requirements, on page 3](#)
- [About Static MPLS, on page 3](#)
- [Prerequisites for Static MPLS, on page 5](#)
- [Guidelines and Limitations for Static MPLS, on page 6](#)
- [Configuring Static MPLS, on page 7](#)
- [Verifying the Static MPLS Configuration, on page 10](#)
- [Displaying Static MPLS Statistics, on page 12](#)
- [Clearing Static MPLS Statistics, on page 14](#)
- [Configuration Examples for Static MPLS, on page 14](#)
- [Additional References, on page 15](#)

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.

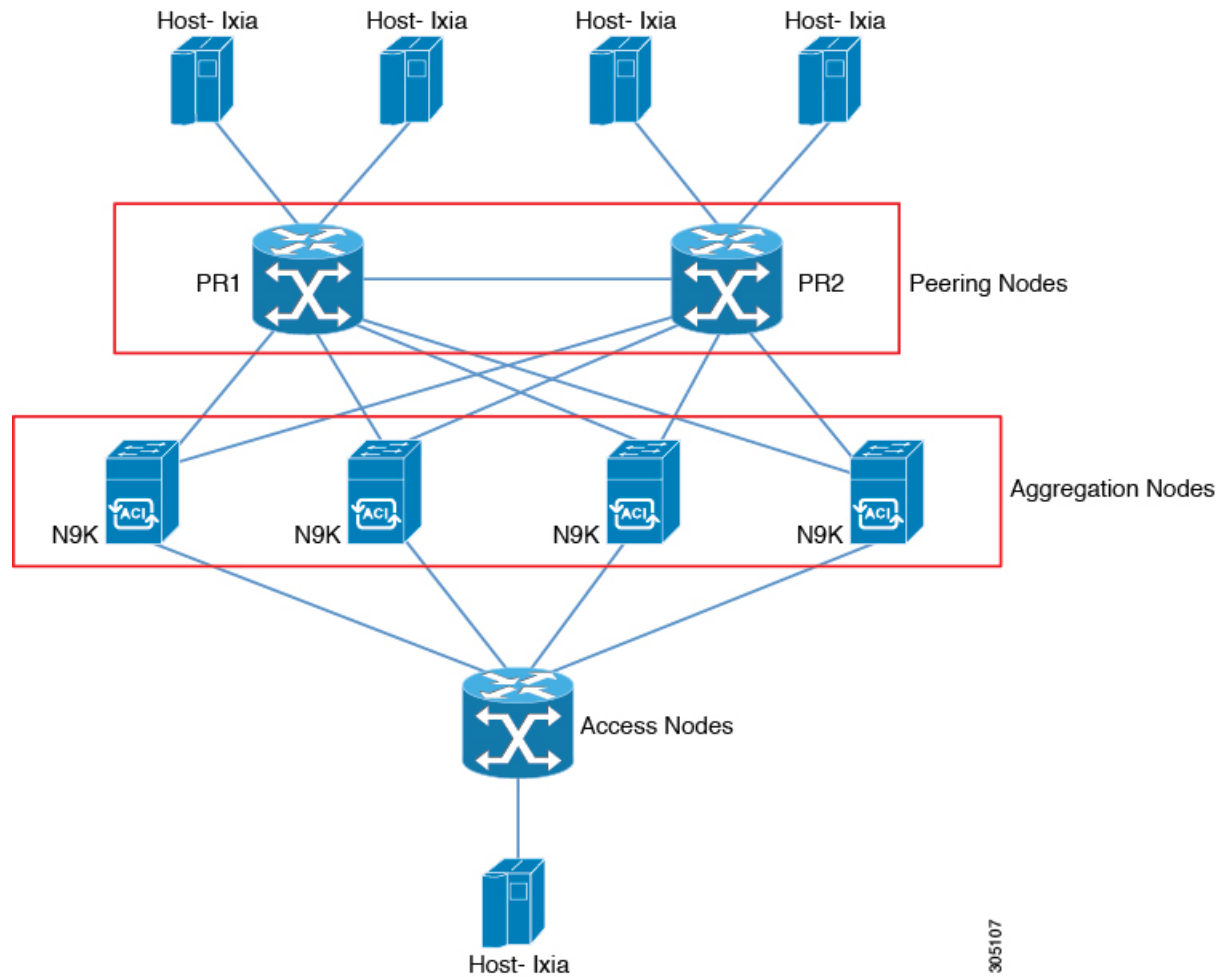
In a swap operation, the label is swapped with a new label, and the packet is forwarded to the next hop that is determined by the incoming label.

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) functions 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 10](#) 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.

SUMMARY STEPS

1. **configure terminal**
2. **[no] install feature-set mpls**
3. **[no] feature-set mpls**
4. **[no] feature mpls static**
5. (Optional) **show feature-set**
6. (Optional) **show feature | inc mpls_static**

DETAILED STEPS

| | Command or Action | Purpose |
|--------|--|--|
| Step 1 | configure terminal Example: <pre>switch# configure terminal switch(config)#</pre> | Enters global configuration mode. |
| Step 2 | [no] install feature-set mpls Example: <pre>switch(config)# install feature-set mpls</pre> | Installs the MPLS feature set. The no form of this command uninstalls the MPLS feature set. |
| Step 3 | [no] feature-set mpls Example: <pre>switch(config)# feature-set mpls</pre> | 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 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 ----- mpls 4 enabled | Displays the status of the MPLS feature set. |
| 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.

SUMMARY STEPS

1. **configure terminal**
2. **[no] mpls label range *min-value max-value* [static *min-static-value max-static-value*]**
3. **(Optional) show mpls label range**
4. **(Optional) copy running-config startup-config**

DETAILED STEPS

| | Command or Action | Purpose |
|---------------|--|---|
| Step 1 | configure terminal Example: switch# configure terminal switch(config)# | Enters global configuration mode. |
| Step 2 | [no] mpls label range <i>min-value max-value</i> [static <i>min-static-value max-static-value</i>] 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. |

| | Command or Action | Purpose |
|---------------|---|--|
| 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.

SUMMARY STEPS

1. **configure terminal**
2. **interface** *type slot/port*
3. **[no] mpls ip forwarding**
4. **mpls static configuration**
5. **address-family** {**ipv4** | **ipv6**} **unicast**
6. **local-label** *local-label-value* **prefix** *destination-prefix destination-prefix-mask*
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*}
8. (Optional) **copy running-config startup-config**

DETAILED STEPS

| | Command or Action | Purpose |
|---------------|---|--|
| Step 1 | configure terminal Example: switch# configure terminal switch(config)# | Enters global configuration mode. |
| Step 2 | interface <i>type slot/port</i> Example: switch(config)# interface ethernet 2/2 switch(config-if)# | Enters the interface configuration mode for the specified interface. |

| | Command or Action | Purpose |
|---------------|---|--|
| 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 <i>local-label-value</i> 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} Example: switch(config-mpls-static-af-lbl)# next-hop auto-resolve | Specifies the next hop. These options are available: <ul style="list-style-type: none"> • next-hop auto-resolve—Use this option for label swap operations. • 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. |

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:
    2000:1111:2121:1111:1111:1111:1111:1111:1 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://www.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 <i>Cisco Nexus 3000 Series NX-OS Security Configuration Guide</i> . |



CHAPTER 3

Configuring MPLS Label Imposition

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

- [About MPLS Label Imposition, on page 17](#)
- [Guidelines and Limitations for MPLS Label Imposition, on page 18](#)
- [Configuring MPLS Label Imposition, on page 18](#)
- [Verifying the MPLS Label Imposition Configuration, on page 22](#)
- [Displaying MPLS Label Imposition Statistics, on page 25](#)
- [Clearing MPLS Label Imposition Statistics, on page 26](#)
- [Configuration Examples for MPLS Label Imposition, on page 27](#)

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.

SUMMARY STEPS

1. **configure terminal**
2. **[no] install feature-set mpls**
3. **[no] feature-set mpls**
4. **[no] feature mpls segment-routing**
5. (Optional) **show feature-set**
6. (Optional) **show feature | grep segment-routing**

DETAILED STEPS

| | Command or Action | Purpose |
|---------------|---|-----------------------------------|
| Step 1 | configure terminal Example: <pre>switch# configure terminal switch(config)#</pre> | Enters global configuration mode. |

| | Command or Action | Purpose |
|--------|---|---|
| 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 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.

SUMMARY STEPS

1. **configure terminal**
2. **[no] mpls label range min-value max-value [static min-static-value max-static-value]**
3. (Optional) **show mpls label range**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

| | Command or Action | Purpose |
|--------|--|-----------------------------------|
| Step 1 | configure terminal Example: | Enters global configuration mode. |

| | Command or Action | Purpose |
|---------------|--|---|
| | switch# configure terminal switch(config)# | |
| Step 2 | [no] mpls label range <i>min-value max-value</i> [static <i>min-static-value max-static-value</i>] 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 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**.

SUMMARY STEPS

1. **configure terminal**
2. **interface** *type slot/port*
3. [no] **mpls ip forwarding**
4. **mpls static configuration**
5. **address-family ipv4 unicast**
6. **lsp** *name*
7. **in-label** *value* **allocate policy** *prefix*
8. **forward**
9. **path** *number* **next-hop** *ip-address* **out-label-stack** *label-id* *label-id*
10. (Optional) **copy running-config startup-config**

DETAILED STEPS

| | Command or Action | Purpose |
|--------|--|---|
| Step 1 | configure terminal Example: <pre>switch# configure terminal switch(config)#</pre> | Enters global configuration mode. |
| Step 2 | interface <i>type slot/port</i> Example: <pre>switch(config)# interface ethernet 2/2 switch(config-if)#</pre> | Enters the interface configuration mode for the specified interface. |
| Step 3 | [no] mpls ip forwarding Example: <pre>switch(config-if)# mpls ip forwarding</pre> | Enables MPLS on the specified interface. The no form of this command disables MPLS on the specified interface. |
| Step 4 | mpls static configuration Example: <pre>switch(config-if)# mpls static configuration switch(config-mpls-static)#</pre> | Enters MPLS static global configuration mode. |
| Step 5 | address-family ipv4 unicast Example: <pre>switch(config-mpls-static)# address-family ipv4 unicast switch(config-mpls-static-af)#</pre> | Enters global address family configuration mode for the specified IPv4 address family. |
| Step 6 | lsp <i>name</i> Example: <pre>switch(config-mpls-static-af)# lsp lsp1 switch(config-mpls-static-lsp)#</pre> | Specifies a name for LSP. |
| Step 7 | in-label <i>value</i> allocate policy <i>prefix</i> Example: <pre>switch(config-mpls-static-lsp)# in-label 8100 allocate policy 15.15.1.0/24 switch(config-mpls-static-lsp-inlabel)#</pre> | Configures an in-label value and a prefix value (optional). |
| Step 8 | forward Example: <pre>switch(config-mpls-static-lsp-inlabel)# forward switch(config-mpls-static-lsp-inlabel-forw)#</pre> | Enters the forward mode. |
| Step 9 | path <i>number</i> next-hop <i>ip-address</i> out-label-stack <i>label-id</i> <i>label-id</i> Example: <pre>switch(config-mpls-static-lsp-inlabel-forw)# path 1 next-hop 13.13.13.13 out-label-stack 16 3000</pre> | Specifies the path. The maximum number of supported paths is 32. |

| | Command or Action | Purpose |
|----------------|---|--|
| 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. |
| 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 :
  LFI-ECMP-idx1:0x514ca(333002), LFI-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-channell121
  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
```



```

FEC      next-hop      interface  tx packets  tx bytes  Label info
-----
12.12.3.2  Vlan122         0          0          0          SWAP 3131 17
12.12.3.2  Vlan122         0          0          0          SWAP 3132 16
12.12.4.2  Vlan123         0          0          0          SWAP 3131 17
12.12.4.2  Vlan123         0          0          0          SWAP 3132 16
12.12.1.2  Po121           0          0          0          SWAP 3131 17
12.12.1.2  Po121           0          0          0          SWAP 3132 16
12.12.2.2  Eth1/51         0          0          0          SWAP 3131 17
12.12.2.2  Eth1/51         0          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   |        |              |          |          |
"      |0x1     |25.25.0.0/16 |12.12.2.2 |Eth1/51    |17
SWAP   |        |              |          |          |3131
"      |0x1     |25.25.0.0/16 |12.12.3.2 |Vlan122    |17
SWAP   |        |              |          |          |3131
"      |0x1     |25.25.0.0/16 |12.12.4.2 |Vlan123    |17
SWAP   |        |              |          |          |3131
"      |        |              |          |          |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. |

| Command | Purpose |
|--|--|
| clear forwarding mpls stats | Clears the ingress MPLS forwarding statistics. |
| clear mpls forwarding statistics | Clears the MPLS forwarding statistics. |
| clear mpls switching label statistics [<i>interface type slot/port</i>] | 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 4

Configuring Segment Routing

This chapter contains information on how to configure segment routing.

- [About Segment Routing, on page 29](#)
- [Guidelines and Limitations for Segment Routing, on page 31](#)
- [Overview of BGP Egress Peer Engineering With Segment Routing, on page 33](#)
- [Guidelines and Limitations for BGP Egress Peer Engineering, on page 35](#)
- [Configuring Segment Routing, on page 35](#)
- [Verifying the Segment Routing Configuration, on page 52](#)
- [Configuration Examples for Segment Routing, on page 52](#)
- [Additional References, on page 57](#)

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 advertises 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.

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.

Segment Routing Global Block

The segment routing global block (SRGB) is the range of local labels reserved for MPLS segment routing. The default label range is from 16000 to 23999.

SRGB is the local property of a segment routing node. Each node can be configured with a different SRGB value, and hence the absolute SID value associated to a BGP prefix segment can change from node to node.

The SRGB must be a proper subset of the dynamic label range and must not overlap the optional MPLS static label range. If dynamic labels in the configured or defaulted SRGB range already have been allocated, the configuration is accepted, and the existing dynamic labels that fall in the SRGB range will remain allocated to the original client. If the BGP router attempts to allocate one of these labels, the SRGB mapping fails, and the BGP router reverts to dynamic label allocation. A change to the SRGB range results in the clients deallocating their labels independent of whether the new range can be allocated.

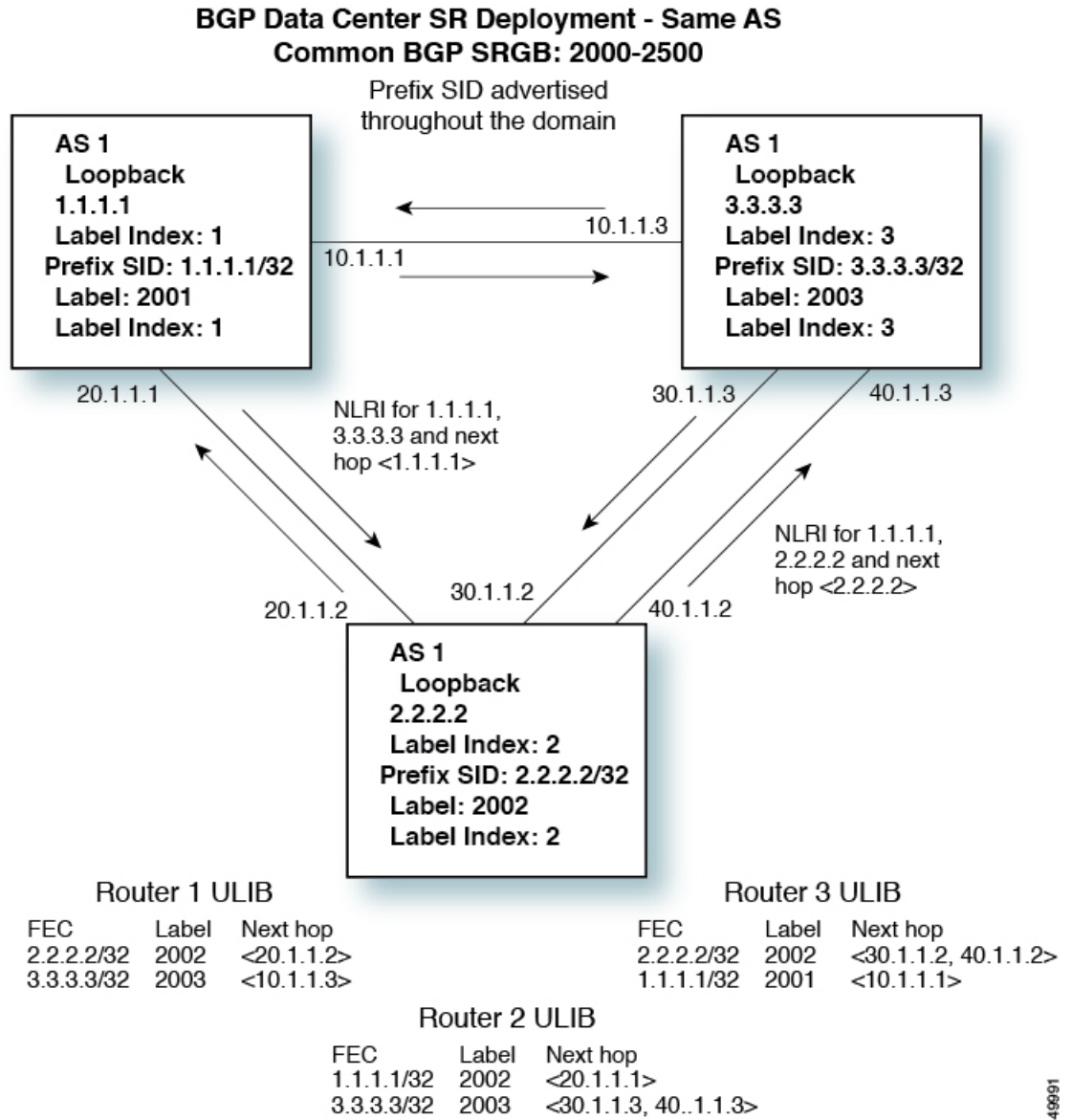
High Availability for Segment Routing

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.

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 2: BGP Prefix SID Simple Example



349691

Guidelines and Limitations for Segment Routing

Segment routing has the following guidelines and limitations:

- 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 and for notifying the clients and it maintains SID mappings prefixes.
- ISIS-SR is supported on Cisco Nexus 3172, Cisco Nexus 3132, Cisco Nexus C31128PQ-10GE and Cisco Nexus N3K-C3164Q-40GE switches.

- Segment routing is supported only on Cisco Nexus 3100 Series switches that are running in N9K mode.
- BGP allocates a SRGB label for iBGP 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 multi-hop BGP is single-hop BGP. iBGP 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 out-label 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. Consequently, 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 will occur 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 is supported on Cisco Nexus C31108PC-V, Cisco Nexus C31108TC-V and Cisco Nexus C3132Q-V switches.

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).

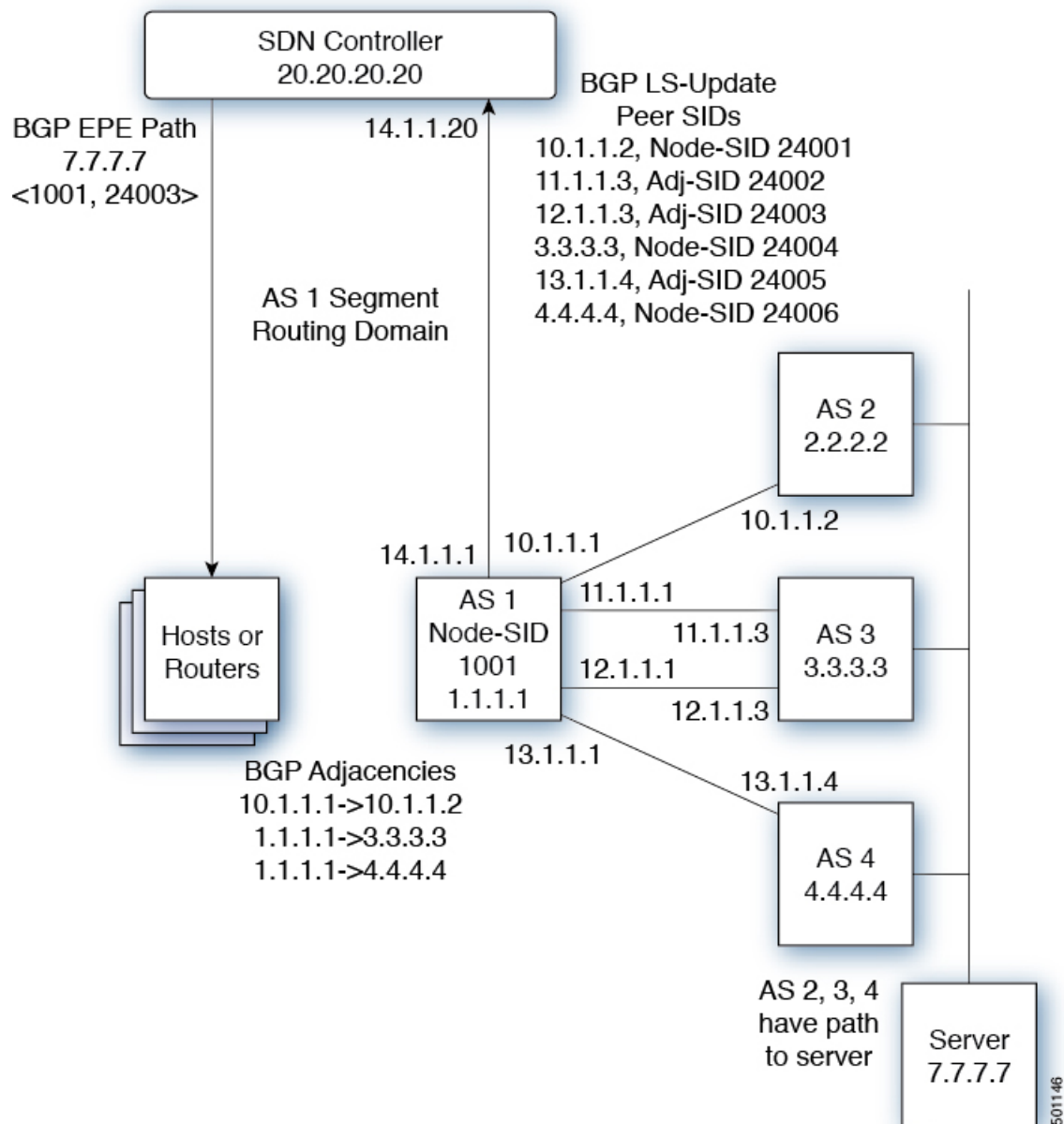
Segment Routing (SR) leverages source routing. A node steers a packet through a controlled set of instructions, known as segments, by prepending the packet with an SR header. A segment can represent any topological or service-based instruction. SR allows steering a flow through any topological path or 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 3: 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 Segment Routing

Configuring Segment Routing Using Segment Routing Application

You can configure segment routing using the 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 and for notifying the clients and it maintains SID mappings prefixes. SR-APP is supported for the BGP and IS-IS protocols.

Complete the following steps to configure segment routing:

Before you begin

Ensure the following conditions are met before configuring Segment Routing using the Segment Routing Application (SR-APP).

- The **feature-set mpls** and **feature mpls segment-routing** commands should be present for configuring the **segment-routing mpls** command.
- The **feature mpls segment-routing** command starts the SR-APP process.
- When **segment-routing mpls** is configured, SR state moves to SR_DISABLED and notifies the same to the clients. In this state, clients can allow protocol specific SR configurations but clients cannot do SID label bindings.
- If the global block is configured, the specified range is used. Otherwise, the default 16000 – 23999 range is used. If the value is less than 16000, then that value will be automatically converted to 16000.
- With the introduction of SR-APP, all configuration is done under **segment-routing mpls** and the prefix SID configuration is handled by SR-APP.

- 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.

SUMMARY STEPS

1. **configure terminal**
2. **segment-routing mpls**
3. **global-block <min> <max>**
4. **connected-prefix-sid-map**
5. **address-family ipv4**
6. **<prefix>/<masklen>[index|absolute] <label>**

DETAILED STEPS

| | Command or Action | Purpose |
|---------------|--|---|
| Step 1 | configure terminal | Enters global configuration mode. |
| Step 2 | segment-routing mpls | Activates the Segment Routing functionality |
| Step 3 | global-block <min> <max> Example: global-block 201000 280000 | Reserves the non-default SRGB range. |
| Step 4 | connected-prefix-sid-map | Provides the SID label for the interface IP covered by the prefix-SID map. |
| Step 5 | address-family ipv4 | Enters global address family configuration mode for the IPv4 address family. |
| Step 6 | <prefix>/<masklen>[index absolute] <label> Example: 2.1.1.5/32 absolute 201101 2.10.1.5/32 index 10001 | 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

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

!Command: show running-config segment-routing
!Time: Fri Dec 22 12:51:59 2017

version 9.2(1)
segment-routing mpls
  global-block 201000 280000
  connected-prefix-sid-map
    address-family ipv4
      2.1.1.5/32 absolute 201101
      2.10.1.5/32 index 10001
```

Enabling MPLS Segment Routing

You can enable MPLS segment routing as long as mutually-exclusive MPLS features such as static MPLS are not enabled.

Before you begin

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

SUMMARY STEPS

1. **configure terminal**
2. **[no] feature mpls segment-routing**
3. (Optional) **show running-config | inc 'feature mpls segment-routing'**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

| | Command or Action | Purpose |
|--------|--|---|
| Step 1 | configure terminal Example: <pre>switch# configure terminal switch(config)#</pre> | Enters global configuration mode. |
| Step 2 | [no] feature mpls segment-routing Example: <pre>switch(config)# feature mpls segment-routing</pre> | Enables the MPLS segment routing feature. The no form of this command disables the MPLS segment routing feature. |
| Step 3 | (Optional) show running-config inc 'feature mpls segment-routing' Example: <pre>switch(config)# show running-config inc 'feature mpls segment-routing'</pre> | Displays the status of the MPLS segment routing feature. |
| Step 4 | (Optional) copy running-config startup-config Example: <pre>switch(config)# copy running-config startup-config</pre> | Copies the running configuration to the startup configuration. |

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.

SUMMARY STEPS

1. **configure terminal**
2. **interface** *type slot/port*
3. **[no] mpls ip forwarding**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

| | Command or Action | Purpose |
|---------------|---|---|
| Step 1 | configure terminal Example: <pre>switch# configure terminal switch(config)#</pre> | Enters global configuration mode. |
| Step 2 | interface <i>type slot/port</i> Example: <pre>switch(config)# interface ethernet 2/2 switch(config-if)#</pre> | Enters the interface configuration mode for the specified interface. |
| Step 3 | [no] mpls ip forwarding Example: <pre>switch(config-if)# mpls ip forwarding</pre> | 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: <pre>switch(config-if)# copy running-config startup-config</pre> | Copies the running configuration to the startup configuration. |

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.

SUMMARY STEPS

1. **feature bgp**
2. **install feature-set mpls**
3. **feature-set mpls**
4. **feature mpls segment-routing**
5. **feature mpls evpn**

DETAILED STEPS

| | 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 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.

SUMMARY STEPS

1. **configure terminal**
2. **router isis instance-tag**
3. **net network-entity-title**
4. **address-family ipv4 unicast**
5. **segment-routing**
6. (Optional) **show running-config segment-routing**

DETAILED STEPS

| | Command or Action | Purpose |
|---------------|---|---|
| Step 1 | configure terminal | Enters global configuration mode. |
| 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 <ul style="list-style-type: none"> • 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. |

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.

SUMMARY STEPS

1. **configure terminal**
2. **[no] router bgp** *autonomous-system-number*
3. **address-family ipv4 unicast**
4. **[no] allocate-label** {**all** | **route-map** *route-map-name*}
5. **exit**
6. **neighbor** *ipv4-address* **remote-as** *autonomous-system-number*
7. **address-family ipv4 labeled-unicast**
8. (Optional) **show bgp ipv4 labeled-unicast** *prefix*
9. (Optional) **copy running-config startup-config**

DETAILED STEPS

| | Command or Action | Purpose |
|---------------|---|--|
| Step 1 | configure terminal Example: <pre>switch# configure terminal switch(config)#</pre> | Enters global configuration mode. |
| Step 2 | [no] router bgp <i>autonomous-system-number</i> Example: <pre>switch(config)# router bgp 64496 switch(config-router)#</pre> | 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: <pre>switch(config-router)# address-family ipv4 unicast switch(config-router-af)#</pre> | Enters global address family configuration mode for the IPv4 address family. |
| Step 4 | [no] allocate-label { all route-map <i>route-map-name</i> } Example: <pre>switch(config-router-af)# allocate-label route-map map1</pre> | 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: <pre>switch(config-router-af)# exit switch(config-router)#</pre> | Exits global address family configuration mode. |

| | Command or Action | Purpose |
|---------------|---|---|
| Step 6 | neighbor ipv4-address remote-as autonomous-system-number Example: <pre>switch(config-router)# neighbor 10.1.1.1 remote-as 64497 switch(config-router-neighbor)#</pre> | Configures the IPv4 address and AS number for a remote BGP peer. |
| Step 7 | address-family ipv4 labeled-unicast Example: <pre>switch(config-router-neighbor)# address-family ipv4 labeled-unicast switch(config-router-neighbor-af)#</pre> | Advertises the labeled IPv4 unicast routes as specified in RFC 3107. |
| Step 8 | (Optional) show bgp ipv4 labeled-unicast prefix Example: <pre>switch(config-router-neighbor-af)# show bgp ipv4 labeled-unicast 10.10.10.10/32</pre> | Displays the advertised label index and the selected local label for the specified IPv4 prefix. |
| Step 9 | (Optional) copy running-config startup-config Example: <pre>switch(config-router-neighbor-af)# copy running-config startup-config</pre> | 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 29](#).

SUMMARY STEPS

1. **configure terminal**
2. **[no] router bgp** *autonomous-system-number*
3. **address-family l2vpn evpn**
4. **exit**
5. **neighbor** *ipv4-address* **remote-as** *autonomous-system-number*
6. **address-family l2vpn evpn**
7. **encapsulation mpls**
8. **vrf** *<customer_name>*
9. **send-community**
10. **send-community** *extended*
11. **address-family ipv4 unicast**
12. **advertise l2vpn evpn**
13. **redistribute direct route-map** **DIRECT_TO_BGP**
14. **label-allocation-mode per-vrf**

DETAILED STEPS

| | Command or Action | Purpose |
|--------|---|--|
| Step 1 | configure terminal | Enters global configuration mode. |
| Step 2 | [no] router bgp <i>autonomous-system-number</i> Example: <pre>switch(config)# router bgp 64496 switch(config-router)#</pre> | 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: | Enters global address family configuration mode for the Layer 2 VPN EVPN. |

| | Command or Action | Purpose |
|---------------|---|---|
| | <pre>switch(config-router)# address-family l2vpn evpn switch(config-router-af)#</pre> | |
| Step 4 | <p>Required: exit</p> <p>Example:</p> <pre>switch(config-router-af)# exit switch(config-router)#</pre> | Exits global address family configuration mode. |
| Step 5 | <p>neighbor <i>ipv4-address</i> remote-as <i>autonomous-system-number</i></p> <p>Example:</p> <pre>switch(config-router)# neighbor 10.1.1.1 remote-as 64497 switch(config-router-neighbor)#</pre> | Configures the IPv4 address and AS number for a remote BGP peer. |
| Step 6 | <p>address-family <i>l2vpn evpn</i></p> <p>Example:</p> <pre>switch(config-router-neighbor)# address-family l2vpn evpn switch(config-router-neighbor-af)#</pre> | Advertises the labeled Layer 2 VPN EVPN. |
| Step 7 | <p>encapsulation mpls</p> <p>Example:</p> <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> <p>BGP segment routing configuration:</p> <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> | <p>Enables BGP EVPN address family and sends EVPN type-5 route update to the neighbors.</p> <p>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.</p> |
| Step 8 | vrf <customer_name> | Configures the VRF. |

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

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
```

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. See [Enabling MPLS Segment Routing, on page 37](#).

SUMMARY STEPS

1. **configure terminal**
2. **[no] segment-routing mpls**
3. **[no] global-block beginning-label ending-label**
4. (Optional) **show mpls label range**
5. **show segment-routing**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

| | Command or Action | Purpose |
|---------------|---|---|
| Step 1 | configure terminal Example: <pre>switch# configure terminal switch(config)#</pre> | Enters global configuration mode. |
| Step 2 | [no] segment-routing mpls Example: <pre>switch(config)# segment-routing mpls switch(config-segment-routing-mpls)#</pre> | <p>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.</p> <p>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.</p> |
| Step 3 | [no] global-block beginning-label ending-label Example: <pre>switch(config-segment-routing-mpls)# global-block 16000 471804</pre> | <p>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 mpls command.</p> <p>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.</p> <p>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.</p> |
| Step 4 | (Optional) show mpls label range Example: | Displays the SRGB, only if the SRGB allocation is successful. |

| | Command or Action | Purpose |
|---------------|---|--|
| | <code>switch(config-segment-routing-mpls)# show mpls label range</code> | |
| Step 5 | show segment-routing | Displays the configured SRGB. |
| Step 6 | (Optional) copy running-config startup-config Example: <code>switch(config-segment-routing-mpls)# copy running-config startup-config</code> | 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.

SUMMARY STEPS

1. **configure terminal**
2. **route-map map-name**
3. **[no] set label-index index**
4. **exit**
5. **router bgp autonomous-system-number**
6. **address-family ipv4 unicast**
7. **network ip-prefix [route-map map-name]**
8. (Optional) **show route-map [map-name]**
9. (Optional) **copy running-config startup-config**

DETAILED STEPS

| | Command or Action | Purpose |
|---------------|--|-----------------------------------|
| Step 1 | configure terminal Example: | Enters global configuration mode. |

| | Command or Action | Purpose |
|---------------|--|--|
| | switch# configure terminal switch(config)# | |
| Step 2 | route-map <i>map-name</i> 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 <i>index</i> 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. |
| Step 4 | exit Example: switch(config-route-map)# exit switch(config)# | Exits route-map configuration mode. |
| Step 5 | router bgp <i>autonomous-system-number</i> 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 <i>ip-prefix</i> [route-map <i>map-name</i>] 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 [<i>map-name</i>] 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. |

Configuring BGP Link State Neighbor Address Family

You can configure an **link-state** address family for a neighbor. The link-state address family is supported in global configuration mode and results in an entry into the **config-router-af** sub-mode for address family **link-state**.

You must configure the link-state unicast address family to exchange Link State NRLI with the neighbor. The valid commands in this sub-mode are **allowas-in**, **capability**, **filter-list**, **inherit**, **route-reflector-client**, and **send-community**.

Before you begin

Prerequisites - waiting for inputs from the reviewer.

SUMMARY STEPS

1. **configure terminal**
2. **[no] address-family link-state**

DETAILED STEPS

| | Command or Action | Purpose |
|--------|---|---|
| Step 1 | configure terminal Example: <pre>switch# configure terminal switch(config)#</pre> | Enters global configuration mode. |
| Step 2 | [no] address-family link-state Example: <pre>switch(config-router# neighbor 192.0.2.1 switch(config-router-neighbor)# address-family link-state</pre> | Enters link-state address neighbor family configuration mode. |

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*.

SUMMARY STEPS

1. **configure terminal**

2. **router bgp** <bgp autonomous number>
3. **neighbor** <IP address>
4. **[no|default] egress-engineering [peer-set peer-set-name]**

DETAILED STEPS

| | Command or Action | Purpose |
|---------------|---|---|
| Step 1 | configure terminal Example: <pre>switch# configure terminal switch(config)#</pre> | 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. |
| Step 4 | [no default] egress-engineering [peer-set peer-set-name] Example: <pre>switch(config)# router bgp 64497 switch(config-router)# neighbor 30.1.1.1 switch(config-router)# egress-engineering peer-set NewPeer</pre> | <p>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.</p> <p>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.</p> <p>See RFC 7752 and draft-ietf-idr-bgpls-segment-routing-epe-05 for more information on EPE.</p> |

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

```

Verifying the Segment Routing Configuration

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

| Command | Purpose |
|---|---|
| <code>show bgp ipv4 labeled-unicast <i>prefix</i></code> | Displays the advertised label index and the selected local label for the specified IPv4 prefix. |
| <code>show bgp paths</code> | Displays the BGP path information, including the advertised label index. |
| <code>show mpls label range</code> | Displays the configured SRGB range of labels. |
| <code>show route-map [<i>map-name</i>]</code> | Displays information about a route map, including the label index. |
| <code>show running-config inc 'feature mpls segment-routing'</code> | Displays the status of the MPLS segment routing feature. |
| <code>show running-config segment-routing</code> | Displays the status of the segment routing feature. |

This example shows how the `show bgp ipv4 labeled-unicast` command can be used with a prefix specification to display the advertised label index and the selected local label:

```

switch# show bgp ipv4 labeled-unicast 19.19.19.19/32
BGP routing table information for VRF default, address family IPv4 Label Unicast
BGP routing table entry for 19.19.19.19/32, version 2
Paths: (1 available, best #1)
Flags: (0x20c0012) on xmit-list, is in urib, is backup urib route, has label
label af: version 2, (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
AS-Path: 19 , path sourced external to AS
60.1.1.19 (metric 0) from 60.1.1.19 (100.100.100.100)
Origin IGP, MED not set, localpref 100, weight 0
Received label 3
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

```

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 Partial Install | Interface | Labels |
|--------------------|-------------------------------|-------------------|--------|
| 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)

```

Additional References

Related Documents

| Related Topic | Document Title |
|---------------|--|
| BGP | <i>Cisco Nexus 3000 Series Unicast Routing Configuration Guide</i> |



CHAPTER 5

Configuring MPLS Segment Routing OAM

This chapter contains information about configuring MPLS Segment Routing OAM.

- [Overview of MPLS Segment Routing OAM, on page 59](#)
- [Segment Routing OAM Support for LSP Ping and Traceroute , on page 59](#)
- [Guidelines and Limitations, on page 60](#)
- [Examples for Using Ping and Traceroute CLI Commands, on page 61](#)

Overview of MPLS Segment Routing OAM

BGP MPLS segment Routing has been deployed on the Cisco Nexus 3000 Series switches. As MPLS segment routing is deployed, few diagnostic tools are required to help resolve the misconfigurations or failures in the segment routing network. The Segment Routing OAM feature provides support for only Nil-FEC (Forwarding Equivalence Classes). No other FEC types are supported.. The Nil-FEC is the basic OAM FEC that is described in RFC-4379.

MPLS OAM provides two main functions for diagnostics purposes:

1. MPLS ping
2. MPLS traceroute

OAM draws the information from different information sources of each FEC type to help diagnose the issues. The Nil-FEC forwarding equivalence class type is not associated with a protocol like the other FEC types or an actual forwarding equivalence class, for example, LDP. Nil-FEC does not query BGP or other routing protocols in the control plane, but it validates the data plane programming.

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 OAM Support for LSP Ping and Traceroute

The Nil-FEC LSP ping and traceroute operations are extensions of regular MPLS ping and traceroute. Nil-FEC LSP Ping/Traceroute functionality supports segment routing and MPLS Static. It also acts as an additional diagnostic tool for all other LSP types. 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 mpls nil-fec labels** *comma-separated-labels* [**output** {**interface** *tx-interface*} [**nexthop** *nexthop-ip-addr*]] CLI command to execute a ping. Use the **traceroute mpls nil-fec labels** *comma-separated-labels* [**output** {**interface** *tx-interface*} [**nexthop** *nexthop-ip-addr*]] CLI command to execute a traceroute.

Guidelines and Limitations

See the following guidelines and limitations for configuring MPLS OAM Nil-FEC:

- 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 an OAM to test a BGP EPE LSP, for example, the last label in the ping or trace-route label stack is an EPE label, the OAM will return a success if the final router has OAM enabled and MPLS enabled on the incoming interface.

For example, in a network set up of A---B---C, A and B are in the segment routing network, B is configured as PE, C is configured as CE, and B is configured with C as a BGP EPE peer using egress-engineering on B, then C must have OAM enabled and MPLS forwarding enabled on the incoming interface.

Examples for Using Ping and Traceroute CLI Commands

Using CLI to Execute a Ping

Use the **ping 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.

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

Using CLI to Execute a Traceroute

Use the following CLI command to execute a traceroute:

```
traceroute mpls nil-fec labels <comma-separated-labels> output interface <tx-interface>
nexthop <nexthop-ip-addr>
```

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 6

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, on page 63](#)

IETF RFCs Supported for Label Switching

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

| RFCs | Title |
|---|--|
| RFC 3107 | <i>Carrying Label Information in BGP-4</i> |
| RFC 7752 | <i>North-Bound Distribution of Link-State and Traffic Engineering (TE) Information Using BGP</i> |
| Draft-ietf-idr-bgpls-segment-routing-epe-05 | <i>Segment Routing BGP Egress Peer Engineering BGP-LS Extensions draft-ietf-idr-bgpls-segment-routing-epe-05</i> |



INDEX

A

address-family {ipv4 | ipv6} unicast [10](#)
address-family ipv4 labeled-unicast [42](#)
address-family ipv4 unicast [21](#), [41](#), [48](#)
allocate-label {all | route-map} [41](#)

C

clear forwarding adjacency mpls stats [14](#), [26](#)
clear forwarding ipv4 adjacency mpls stats [26](#)
clear forwarding ipv6 adjacency mpls stats [14](#)
clear forwarding mpls drop-stats [14](#), [26](#)
clear forwarding mpls stats [14](#), [27](#)
clear mpls forwarding statistics [14](#), [27](#)
clear mpls switching label statistics [14](#), [27](#)

F

feature mpls segment-routing [19](#)
feature mpls static [8](#), [37](#)
feature-set mpls [7](#), [19](#)
forward [21](#)

G

global-block [46](#)

I

in-label [21](#)
install feature-set mpls [7](#), [19](#)

L

local-label [10](#)
lsp [21](#)

M

mpls ip forwarding [10](#), [21](#), [38](#)

mpls label range [8](#), [20](#)
mpls static configuration [10](#), [21](#)

N

neighbor [42](#), [44](#)
network [48](#)
next-hop [10](#)
next-hop auto-resolve [10](#)
next-hop backup [10](#)

R

route-map [48](#)

S

segment-routing mpls [46](#)
set label-index [48](#)
show bgp ipv4 labeled-unicast [42](#), [52](#)
show bgp paths [52](#)
show feature | grep segment-routing [19](#), [22](#)
show feature | inc mpls_static [8](#), [11](#)
show feature-set [8](#), [11](#), [19](#), [22](#)
show forwarding adjacency mpls stats [13](#), [25](#)
show forwarding ipv4 adjacency mpls stats [25](#)
show forwarding ipv6 adjacency mpls stats [13](#)
show forwarding mpls drop-stats [13](#), [25](#)
show forwarding mpls ecmp [13](#), [22](#)
show forwarding mpls ecmp module [13](#), [22](#)
show forwarding mpls ecmp platform [13](#), [22](#)
show forwarding mpls label [13](#), [22](#), [25](#)
show ip route [11](#)
show mpls forwarding statistics [13](#), [25](#)
show mpls label range [9](#), [11](#), [20](#), [22](#), [47](#), [52](#)
show mpls static binding {all | ipv4 | ipv6} [11](#)
show mpls static binding {all | ipv4} [22](#)
show mpls switching [11](#), [22](#)
show mpls switching detail [11](#), [22](#)
show mpls switching labels [13](#), [25](#)
show route-map [48](#), [52](#)
show running-config | inc 'feature mpls segment-routing' [37](#), [52](#)

