Inspur CN6000 Series INOS-CN Label Switching Configuration Guide

First Published: December 2017

CONTENTS

CHAPTER 3

CHAPTER 4

Preface

This preface includes the following sections:

Audience

This publication is for network administrators who install, configure, and maintain Inspur CN6048TP, CN6132Q-V, CN61108PC-V, and CN61108TC-V switches.

Document Conventions

Command descriptions use the following conventions:

Convention	Description
bold	Bold text indicates the commands and keywords that you enter literally as shown.
Italic	Italic text indicates arguments for which the user supplies the values.
[x]	Square brackets enclose an optional element (keyword or argument).
[x | y]	Square brackets enclosing keywords or arguments separated by a vertical bar indicate an optional choice.
{x | y}	Braces enclosing keywords or arguments separated by a vertical bar indicate a required choice.

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

Examples use the following conventions:

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

CHAPTER 1

New and Changed Information

This chapter provides release-specific information for each new and changed feature in the *Inspur CN6000 Series INOS-CN Label Switching Configuration Guide*.

New and Changed Information

This table summarizes the new and changed features for the *Inspur CN6000 Series INOS-CN Label Switching Configuration Guide* and tells you where they are documented.

Table 1: New and Changed Features

Feature	Description	Changed in Release	Where Documented
Label Switching	Initial release	7.0(3)I7(1)	

New and Changed Information

CHAPTER 2

Configuring Static MPLS

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

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.

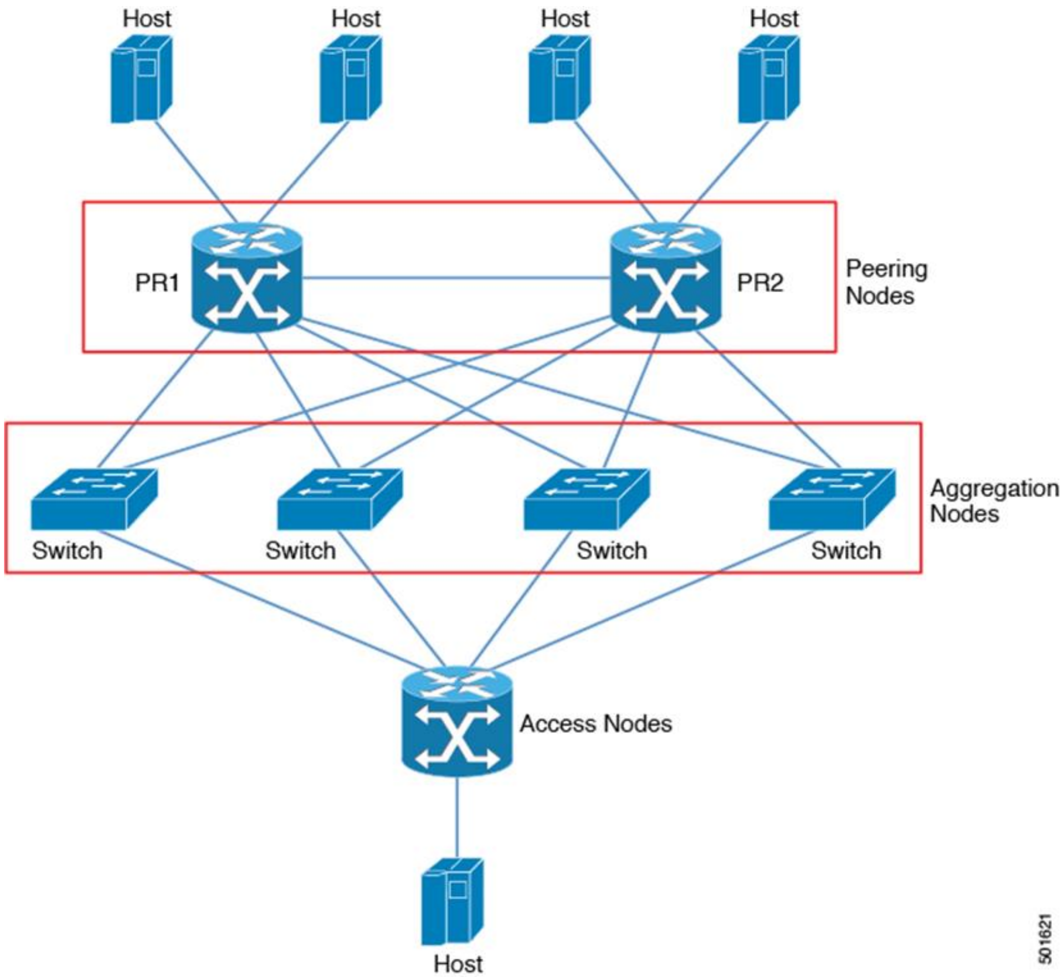
In a swap operation, the label is swapped with a new label, and the packet is forwarded to the next hop that is determined by the incoming label.

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.

Licensing Requirements for Static MPLS

The following table shows the licensing requirements for this feature:

Product	License Requirement
Inspur INOS-CN	Static MPLS requires no license. Any feature not included in a license package is bundled with the software image and is provided at no extra charge to you.

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 (only for swap operation case) for Inspur CN61108PC-V, CN61108TC-V, and CN6132Q-V switches, but not for Inspur CN6048TP 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 generally 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.
- 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 11 for an example.
- VRFs, vPCs, 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:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

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

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

Step 6	**show feature | inc mpls_static**
	
	Example:
	`switch(config)# show feature | inc mpls static`
	`mpls_static           1         enabled`
		(Optional)
		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** *min-value max-value* [**static** *min-static-value max-static-value*] **Example:** `switch(config)# mpls label range 17 99 static 100 10000`	Reserves a range of labels for static label assignment. The range for the minimum and maximum values is from 16 to 471804.
Step 3	**show mpls label range** **Example:** `switch(config)# show mpls label range`	(Optional) Displays the label range that is configured for static MPLS.
Step 4	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) 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** *type slot/port*
	
	Example:
	`switch(config)# interface ethernet 2/2`
	`switch(config-if)#`
		Enters the interface configuration mode for the specified interface.

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

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

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

	Command or Action	Purpose

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

Step 7	**next-hop** {**auto-resolve** | *destination-ip-next-hop* **out-label implicit-null** | **backup** *local-egress-interface destination-ip-next-hop* **out-label** *output-label-value*} **Example:** `switch(config-mpls-static-af-lbl)# next-hop auto-resolve`	Specifies the next hop. These options are available: • **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	**copy running-config startup-config** **Example:** `switch(config-mpls-static-af-lbl)# copy running-config startup-config`	(Optional) 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.

Command	Purpose
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:
      2000:1111:2121:1111:1111:1111:1111:1 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>
 <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
```

CHAPTER 3

Configuring MPLS Label Imposition

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

About MPLS Label Imposition

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

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

The 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.
- Multicast is not supported for the MPLS label imposition.
- For the MPLS label imposition, up to 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.

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:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **install feature-set mpls** **Example:** `switch(config)# install feature-set mpls`	Installs the MPLS feature set. The **no** form of this command uninstalls the MPLS feature set.
Step 3	[**no**] **feature-set mpls** **Example:** `switch(config)# feature-set mpls`	Enables the MPLS feature set. The **no** form of this command disables the MPLS feature set.
Step 4	[**no**] **feature mpls 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	**show feature-set**
	
	Example:
	`switch(config)# show feature-set`
	`Feature Set Name      ID        State`
	`--------------------  --------  --------`
	`mpls                  4         enabled`
		(Optional)
		Displays the status of the MPLS feature set.

Step 6	**show feature | grep segment-routing**
	
	Example:
	`switch(config)# show feature | grep segment-routing`
	`segment-routing          1        enabled`
		(Optional)
		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:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

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

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

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

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

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

	Command or Action	Purpose

Step 6	**lsp** *name*
	
	Example:
	`switch(config-mpls-static-af)# lsp lsp1`
	`switch(config-mpls-static-lsp)#`
		Specifies a name for LSP.

Step 7	**in-label** *value* **allocate policy** *prefix*
	
	Example:
	`switch(config-mpls-static-lsp)# in-label 8100 allocate policy 15.15.1.0/24`
	`switch(config-mpls-static-lsp-inlabel)#`
		Configures an in-label value and a prefix value (optional).

Step 8	**forward**
	
	Example:
	`switch(config-mpls-static-lsp-inlabel)# forward`
	`switch(config-mpls-static-lsp-inlabel-forw)#`
		Enters the forward mode.

Step 9	**path** *number* **next-hop** *ip-address* **out-label-stack** *label-id* *label-id* **Example:** `switch(config-mpls-static-lsp-inlabel-forw)# path 1 next-hop 13.13.13.13 out-label-stack 16 3000`	Specifies the path. The maximum number of supported paths is 32.
Step 10	**copy running-config startup-config** **Example:** `switch(config-mpls-static-lsp-inlabel-forw)# copy running-config startup-config`	(Optional) 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.

Command	Purpose
show mpls label range	Displays the label range that is configured for MPLS label imposition.
show mpls static binding {**all** | **ipv4**}	Displays the configured static prefix or label bindings.
show mpls switching [**detail**]	Displays MPLS label switching information.
show running-config mpls static	Displays the running static MPLS configuration.

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

```
slot  1
=======
 Virtual Object 16 :
    LFIB-ECMP-idx1:0x514ca(333002), LFIB-ECMP-idx2:0x0(0) ADJ-idx 0
    Hw vobj-index (0): unit-0:200022 unit-1:0 unit-2:0, cmn-index: 99004
    Hw NVE vobj-index (0): unit-0:0 unit-1:0 unit-2:0, cmn-index: 99004
    Hw vobj-index (1): unit-0:0 unit-1:0 unit-2:0, cmn-index: 0
    Hw NVE vobj-index (1): unit-0:0 unit-1:0 unit-2:0, cmn-index: 0
    Num prefixes : 0
Partial Install: No
    Active paths:
        Recursive NH 12.12.3.2/32 ,Label stack : 3132 16, table 0x1
        Recursive NH 12.12.4.2/32 ,Label stack : 3132 16, table 0x1
        Recursive NH 12.12.1.2/32 ,Label stack : 3132 16, table 0x1
        Recursive NH 12.12.2.2/32 ,Label stack : 3132 16, table 0x1
    CNHs:
        12.12.1.2, port-channel121
        Hw adj: unit-0:100006 unit-1:0 unit-2:0, cmn-index: 6, LIF:4155
        Hw NVE adj: unit-0:0 unit-1:0 unit-2:0, cmn-index: 6, LIF:4155
        12.12.2.2, Ethernet1/51
        Hw adj: unit-0:100009 unit-1:0 unit-2:0, cmn-index: 7, LIF:4150
        Hw NVE adj: unit-0:0 unit-1:0 unit-2:0, cmn-index: 7, LIF:4150
        12.12.3.2, Vlan122
        Hw adj: unit-0:100012 unit-1:0 unit-2:0, cmn-index: 8, LIF:122
        Hw NVE adj: unit-0:0 unit-1:0 unit-2:0, cmn-index: 8, LIF:122
        12.12.4.2, Vlan123
        Hw adj: unit-0:100017 unit-1:0 unit-2:0, cmn-index: 9, LIF:123
        Hw NVE adj: unit-0:0 unit-1:0 unit-2:0, cmn-index: 9,
    LIF:123 Hw instance new : (0x182bc, 99004) ls count new 4
    FEC:
        FEC-ECMP-idx1:0x514cb(333003), FEC-ECMP-idx2:0x0(0) ADJ-idx 0
        Hw instance new: (0x182bd, 99005) ls count new 4
        label list count: (1)
        VOBJ Refcount :
1 Virtual Object 12 :
    LFIB-ECMP-idx1:0x514c8(333000), LFIB-ECMP-idx2:0x0(0) ADJ-idx 0
    Hw vobj-index (0): unit-0:200016 unit-1:0 unit-2:0, cmn-index: 99002
    Hw NVE vobj-index (0): unit-0:0 unit-1:0 unit-2:0, cmn-index: 99002
    Hw vobj-index (1): unit-0:0 unit-1:0 unit-2:0, cmn-index: 0
    Hw NVE vobj-index (1): unit-0:0 unit-1:0 unit-2:0, cmn-index: 0
    Num prefixes : 1
Partial Install: No
    Active paths:
        Recursive NH 12.12.1.2/32 ,Label stack : 3131 17, table 0x1
        Recursive NH 12.12.2.2/32 ,Label stack : 3131 17, table 0x1
        Recursive NH 12.12.3.2/32 ,Label stack : 3131 17, table 0x1
        Recursive NH 12.12.4.2/32 ,Label stack : 3131 17, table 0x1
    CNHs:
        12.12.1.2, port-channel121
        Hw adj: unit-0:100006 unit-1:0 unit-2:0, cmn-index: 6, LIF:4155
        Hw NVE adj: unit-0:0 unit-1:0 unit-2:0, cmn-index: 6, LIF:4155
        12.12.2.2, Ethernet1/51
        Hw adj: unit-0:100009 unit-1:0 unit-2:0, cmn-index: 7, LIF:4150
```

```
        Hw NVE adj: unit-0:0 unit-1:0 unit-2:0, cmn-index: 7, LIF:4150
        12.12.3.2, Vlan122
        Hw adj: unit-0:100012 unit-1:0 unit-2:0, cmn-index: 8, LIF:122
        Hw NVE adj: unit-0:0 unit-1:0 unit-2:0, cmn-index: 8, LIF:122
        12.12.4.2, Vlan123
        Hw adj: unit-0:100017 unit-1:0 unit-2:0, cmn-index: 9, LIF:123
        Hw NVE adj: unit-0:0 unit-1:0 unit-2:0, cmn-index: 9,
    LIF:123 Hw instance new : (0x182ba, 99002) ls count new 4
    FEC:
        FEC-ECMP-idx1:0x514c9(333001), FEC-ECMP-idx2:0x0(0) ADJ-idx 0
        Hw instance new: (0x182bb, 99003) ls count new 4
        label list count: (1)
        VOBJ Refcount : 2
```

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

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

 17
```

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

```
LI_TEST1 25.25.0.0/16: (vrf: default) Incoming label: 8100
LSP Type: POLICY
  Outgoing labels:
     (path 1) 12.12.1.2 3131,17
     (path 2) 12.12.2.2 3131,17
     (path 3) 12.12.3.2 3131,17
     (path 4) 12.12.4.2 3131,17
LI_TEST2 (vrf: default) Incoming label: 8200
LSP Type: XC
  Outgoing labels:
     (path 1) 12.12.3.2 3132,16
     (path 2) 12.12.4.2 3132,16
     (path 3) 12.12.1.2 3132,16
     (path 4) 12.12.2.2 3132,16
```

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

```
Legend:
(P)=Protected, (F)=FRR active, (*)=more labels in stack.
Local      Out-Label   FEC                                         Out-Interface
Next-Hop
8200       3132        Label 8200
12.12.3.2                                  *
8200       3132        Label 8200
12.12.4.2                                  *
8200       3132        Label 8200
12.12.1.2                                  *
8200       3132        Label 8200

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

8100       3131        Pol 25.25.0.0/16
```

```
12.12.3.2                                        *
8100         3131        Pol 25.25.0.0/16
12.12.4.2                                        *
```

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

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

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

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

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

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

slot 1
=======

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

Displaying MPLS Label Imposition Statistics

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

Command	Purpose
show forwarding [**ipv4**] **adjacency mpls stats**	Displays MPLS IPv4 adjacency statistics.

Command	Purpose
show forwarding mpls drop-stats	Displays MPLS forwarding packet drop statistics.
show forwarding mpls label *label* **stats** [**platform**]	Displays MPLS label forwarding statistics.
show mpls forwarding statistics [**interface** *type slot/port*]	Displays MPLS forwarding statistics.
show mpls switching labels *low-label-value* [*high-label-value*] [**detail**]	Displays MPLS label switching statistics. The range for the label value is from 0 to 524286.

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

```
slot   1

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

       12.12.2.2   Eth1/51       0            0        SWAP 3132 16
```

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

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

  SWAP  |           |                    |                 |            | 17
    "   |0x1        |25.25.0.0/16        |12.12.2.2        |Eth1/51      |3131

  SWAP  |           |                    |                 |            | 17
    "   |0x1        |25.25.0.0/16        |12.12.3.2        |Vlan122      |3131

  SWAP  |           |                    |                 |            | 17
    "   |0x1        |25.25.0.0/16        |12.12.4.2        |Vlan123      |3131

  SWAP
        |           |                    |                 |            | 17
Input Pkts : 126906012            Input Bytes : 64975876096
SWAP Output Pkts: 126959183       SWAP Output Bytes: 65764550340

TUNNEL Output Pkts: 126959053     TUNNEL Output Bytes: 66272319384
```

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

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

    Packets/Bytes input dropped  : 0/0
    Packets/Bytes output dropped : 0/0
```

Clearing MPLS Label Imposition Statistics

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

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

Configuration Examples for MPLS Label Imposition

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

```
switch(config-if)# mpls static configuration
switch(config-mpls-static)# address-family ipv4
unicast switch(config-mpls-static-af)# lsp LI_TEST1
switch(config-mpls-static-lsp)# in-label 8100 allocate policy 25.25.0.0/16 switch(config-
mpls-static-lsp-inlabel)# forward switch(config-mpls-static-lsp-inlabel-forw)# path 1
next-hop 12.12.1.2 out-label-stack 3131
 17
switch(config-mpls-static-lsp-inlabel-forw)# path 2 next-hop 12.12.2.2 out-label-stack
 3131 17
switch(config-mpls-static-lsp-inlabel-forw)# path 3 next-hop 12.12.3.2 out-label-stack
 3131 17
switch(config-mpls-static-lsp-inlabel-forw)# path 4 next-hop 12.12.4.2 out-label-stack
 3131 17
```

To remove a next-hop, you can use
no path 1

To remove the named lsp, you can use
no lsp LI_TEST1

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

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

```
  16
switch(config-mpls-static-lsp-inlabel-forw)# path 4 next-hop 12.12.2.2 out-label-stack 3132
  16
```

CHAPTER 4

Configuring Segment Routing

This chapter contains information on how to configure segment routing.

About Segment Routing

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

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

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

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

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.

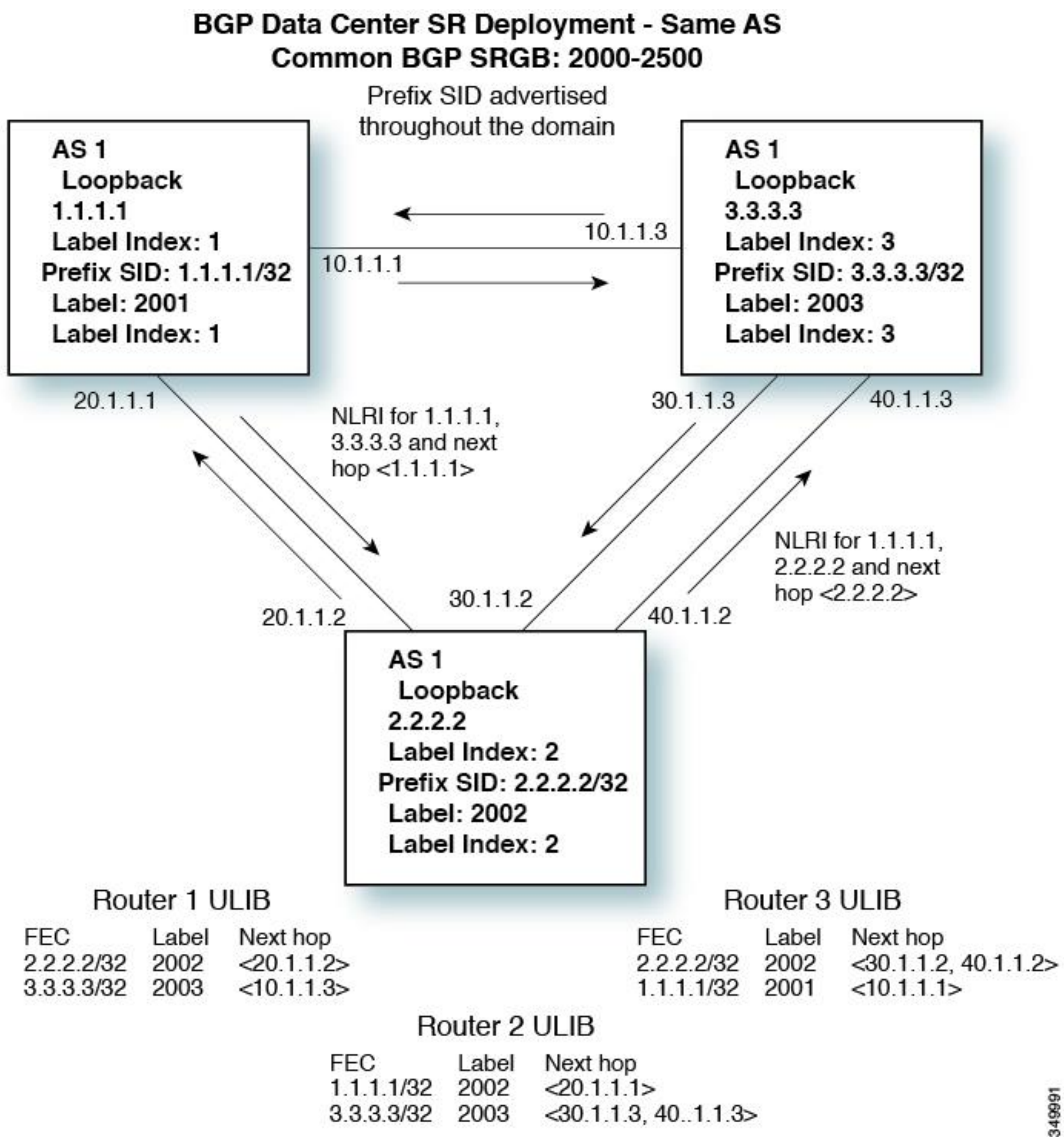
High Availability for Segment Routing

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



Licensing Requirements for Segment Routing

The following table shows the licensing requirements for this feature:

Product	License Requirement
Inspur INOS-CN	Segment routing does not require a license. Any feature not included in a license package is bundled with the software image and is provided at no extra charge to you. Border Gateway Protocol (BGP) requires an INOS-CN LAN Enterprise License.

Guidelines and Limitations for Segment Routing

Segment routing has the following guidelines and limitations:

- 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, Inspur 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.
- Segment routing and segment routing EVPN are supported on Inspur CN61108PC-V, CN61108TC-V, and CN6132Q-V switches.
- Inspur CN61108PC-V, CN61108TC-V, and CN6132Q-V switches support Link-State distribution and Egress Peer Engineering (EPE) using BGP.

Overview of BGP Egress Peer Engineering With Segment Routing

Inspur CN6000 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).

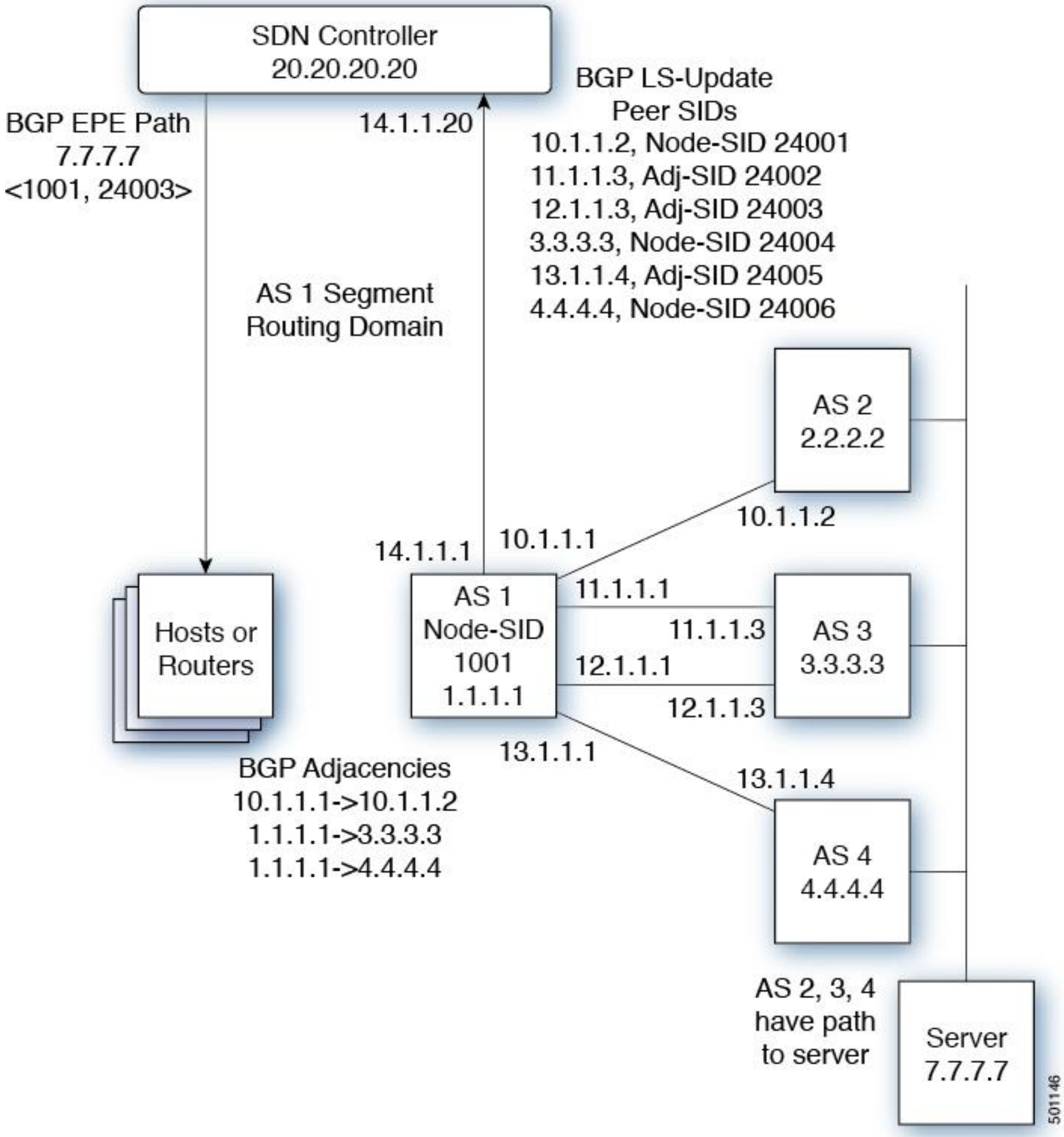
Segment Routing (SR) leverages source routing. A node steers a packet through a controlled set of instructions, known as segments, by prepending the packet with an SR header. A segment can represent any topological or service-based instruction. SR allows steering a flow through any topological path or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

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:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**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 3	**show running-config | inc 'feature mpls segment-routing'** **Example:** `switch(config)# show running-config | inc 'feature mpls segment-routing'`	(Optional) Displays the status of the MPLS segment routing feature.
Step 4	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) 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:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

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

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

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 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. See Enabling MPLS Segment Routing, on page 35.

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:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

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

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

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

	Command or Action	Purpose

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

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

Step 7	**address-family ipv4 labeled-unicast**
	
	Example:
	`switch(config-router-neighbor)# address-family ipv4 labeled-unicast`
	`switch(config-router-neighbor-af)#`
		Advertises the labeled IPv4 unicast routes as specified in RFC 3107.

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

Configuring BGP EVPN and Label Allocation Mode

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

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.

	Command or Action	Purpose

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

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

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

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

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

Step 7	**encapsulation mpls**
	
	Example:
	`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`
	BGP segment routing configuration:
	`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`
		Enables BGP EVPN address family and sends EVPN type-5 route update to the neighbors.
		
		Note The default tunnel encapsulation in EVPN for the IP route type in INOS-CN is VXLAN. To override that, a CLI is introduced to indicate MPLS tunnel encapsulation.

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

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 35.

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

Step 3	[**no**] **global-block** *beginning-label ending-label* **Example:** `switch(config-segment-routing-mpls)# global-block 10000 25000`	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. The permissive values for the beginning MPLS label and the ending MPLS label are limited to the values you configured for the minimum and maximum dynamic label range (using the **mpls label range** command). The beginning label must be less than or equal to the ending label.
Step 4	**show mpls label range** **Example:** `switch(config-segment-routing-mpls)# show mpls label range`	(Optional) Displays the configured SRGB range of labels.
Step 5	**copy running-config startup-config** **Example:** `switch(config-segment-routing-mpls)# copy running-config startup-config`	(Optional) 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 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:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

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

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

	Command or Action	Purpose

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

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

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

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

Configuring BGP Link State Address Family

With the introduction of RFC 7752 in Inspur INOS-CN, you can configure an link-state in address family configuration mode. The link-state address family is supported in global configuration mode and results in an entry into the config-router-af submode for address family link-state.

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

Step 2	[**no**] **address-family link-state**
	
	Example:
	`switch(config)# router bgp 64497`
	`switch(config-router)# addressfamily link-state`
		Enters link-state address family configuration mode.

Configuring Neighbor Egress Peer Engineering Using BGP

With the introduction of RFC 7752 in Inspur IBOS, 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.

SUMMARY STEPS

1. **configure terminal**
2. [**no**|**default**] **egress-engineering** [**peer-set** *peer-set-name*]

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	[**no**|**default**] **egress-engineering** [**peer-set** *peer-set-name*]
	
	Example:
	`switch(config)# router bgp 64497`
	`switch(config-router)# neighbor 30.1.1.1`
	`switch(config-router)#`
	`egress-engineering peer-set NewPeer`
		Specifies whether a Peer-Node-SID is allocated for the neighbor and advertised in an instance of a BGP Link-State (BGP-LS) address family Link NLRI. If the neighbor is a multi-hop neighbor, a BGP-LS Link NLRI instance will also be advertised for each Equal-Cost-MultiPath (ECMP) path to the neighbor and will include a unique Peer-Adj-SID.
		
		Optionally, you can add the neighbor to a peer-set. The Peer-Set-SID will also be advertised in the BGP-LS Link NLRI in the same instance as the Peer-Node-SID. BGP Link-State NLRI will be advertised to all neighbors with the link-state address family configured.
		
		See RFC 7752 and draft-ietf-idr-bgpls-segment-routing-epe-05 for more information on EPE.

Configuration Example for Egress Peer Engineering

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

```
 hostname epe-as-1
install feature-set
mpls feature-set mpls
feature telnet
feature bash-shell
feature scp-server
feature bgp
feature mpls segment-routing
segment-routing
mpls vlan 1
vrf context management
  ip route 0.0.0.0/0 10.30.97.1
  ip route 0.0.0.0/0 10.30.108.1
interface Ethernet1/1
  no switchport
  ip address 10.1.1.1/24
  no shutdown
interface Ethernet1/2
  no switchport
  ip address 11.1.1.1/24
  no shutdown
interface Ethernet1/3
  no switchport
```

```
  ip address 12.1.1.1/24
  no shutdown
interface Ethernet1/4
  no switchport
  ip address 13.1.1.1/24
  no shutdown
interface Ethernet1/5
  no switchport
  ip address 14.1.1.1/24
  no shutdown
interface mgmt0 ip
  address dhcp
  vrf member management

interface loopback1
  ip address 1.1.1.1/32
line console
line vty
ip route 2.2.2.2/32 10.1.1.2
ip route 3.3.3.3/32 11.1.1.3
ip route 3.3.3.3/32 12.1.1.3
ip route 4.4.4.4/32 13.1.1.4
ip route 20.20.20.20/32 14.1.1.20
router bgp 1
  address-family ipv4 unicast
  address-family link-state
 neighbor 10.1.1.2
    remote-as 2 address-
    family ipv4 egress-
    engineering
 neighbor 3.3.3.3 remote-
   as 3 address-family
   ipv4 update-source
   loopback1 ebgp-multihop
   2 egress-engineering

 neighbor 4.4.4.4 remote-
   as 4 address-family
   ipv4 update-source
   loopback1 ebgp-multihop
   2 egress-engineering

neighbor 20.20.20.20
   remote-as 1 address-
   family link-state
   update-source loopback1
   ebgp-multihop 2
```

Verifying the Segment Routing Configuration

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

Command	Purpose
show bgp ipv4 labeled-unicast *prefix*	Displays the advertised label index and the selected local label for the specified IPv4 prefix.
show bgp paths	Displays the BGP path information, including the advertised label index.

Command	Purpose
show mpls label range	Displays the configured SRGB range of labels.
show route-map [*map-name*]	Displays information about a route map, including the label index.
show running-config | inc 'feature mpls segment-routing'	Displays the status of the MPLS 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

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

```
vrf context management
  ip route 0.0.0.0/0 10.30.108.1
interface Ethernet1/1
  no switchport
  ip address 10.1.1.1/24
  no shutdown
interface mgmt0 ip
  address dhcp
  vrf member management
interface loopback1
  ip address 10.10.10.10/32
interface loopback2
  ip address 20.20.20.20/32
line console
line vty
router bgp 1
  address-family ipv4 unicast
    network 10.10.10.10/32 route-map label-index-10
    network 20.20.20.20/32 route-map label-index-20
    allocate-label all
  neighbor 10.1.1.2 remote-as 2 address-
    family ipv4 labeled-unicast
```

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

```
hostname s2
install feature-set
mpls feature-set mpls
feature telnet
feature bash-shell
feature scp-server
feature bgp
feature mpls segment-routing
segment-routing
mpls vlan 1
vrf context management
  ip route 0.0.0.0/0 10.30.97.1
  ip route 0.0.0.0/0 10.30.108.1
interface Ethernet1/1
  no switchport
  ip address 10.1.1.2/24
  ipv6 address
  10:1:1::2/64 no shutdown
interface mgmt0 ip
  address dhcp
  vrf member management
interface loopback1
  ip address 2.2.2.2/32
line console
line vty
router bgp 2
  address-family ipv4 unicast
    allocate-label all
  neighbor 10.1.1.1 remote-as 1 address-
    family ipv4 labeled-unicast
```

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

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

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

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

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

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

```
hostname epe-as-1
install feature-set
mpls feature-set mpls
feature telnet
feature bash-shell
feature scp-server
feature bgp
feature mpls segment-routing
segment-routing
mpls vlan 1
vrf context management
  ip route 0.0.0.0/0 10.30.97.1
  ip route 0.0.0.0/0 10.30.108.1
interface Ethernet1/1
  no switchport
  ip address 10.1.1.1/24
  no shutdown
interface Ethernet1/2
  no switchport
  ip address 11.1.1.1/24
  no shutdown
interface Ethernet1/3
  no switchport
  ip address 12.1.1.1/24
  no shutdown
interface Ethernet1/4
  no switchport
  ip address 13.1.1.1/24
  no shutdown
interface Ethernet1/5
  no switchport
  ip address 14.1.1.1/24
  no shutdown
```

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

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

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

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

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

IPv4 routes for table 2/base
                                      +----------------------    +----------------
Prefix            | Next-hop                                 | Interface              | Labels
          | Partial_______________________________Install
------------------+                   +----------------------    +----------------
0.0.0.0/32            Drop                                       Null0
127.0.0.0/8           Drop                                       Null0
255.255.255.255/32    Receive                                    sup-eth1
*41.11.2.0/24         27.1.31.4                                  Ethernet1/3            PUSH
  30002 492529
                      27.1.32.4                                  Ethernet1/21           PUSH
  30002 492529
                      27.1.33.4                                  port-channel23         PUSH
  30002 492529
                      27.11.31.4                                 Ethernet1/3.11         PUSH
  30002 492529
                      27.11.33.4                                 port-channel23.11      PUSH
  30002 492529
                      37.1.53.4                                  Ethernet1/53/1         PUSH
  29002 492529
                      37.1.54.4                                  Ethernet1/54/1         PUSH
  29002 492529
                      37.2.53.4                                  Ethernet1/53/2         PUSH
  29002 492529
                      37.2.54.4                                  Ethernet1/54/2         PUSH
  29002 492529
                      80.211.11.1                                Vlan801                PUSH
  30002 492529
```

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

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

476 identical, 0 modified, 0 filtered received paths using 0 bytes

Neighbor        V    AS MsgRcvd MsgSent    TblVer  InQ OutQ Up/Down  State/PfxRcd
1.1.1.1         4    11       0       0         0    0    0 23:01:53 Shut (Admin)

1.1.1.9         4    11    4637    1836  17370542    0    0 23:01:40 476
```

```
1.1.1.10        4    11       0       0        0    0    0 23:01:53 Shut (Admin)
1.1.1.11        4    11       0       0        0    0    0 23:01:52 Shut (Admin)
```

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

```
show bgp l2vpn evpn 41.11.2.0
BGP routing table information for VRF default, address family L2VPN
EVPN Route Distinguisher: 14.1.4.1:115
BGP routing table entry for [5]:[0]:[0]:[24]:[41.11.2.0]:[0.0.0.0]/224, version
17369591 Paths: (1 available, best #1)
Flags: (0x000002) on xmit-list, is not in l2rib/evpn, is not in HW
  Advertised path-id 1
  Path type: external, path is valid, received and used, is best path
             Imported to 2 destination(s)
  AS-Path: 11 , path sourced external to AS
    1.1.1.9 (metric 0) from 1.1.1.9 (14.1.4.1)
      Origin incomplete, MED 0, localpref 100, weight 0
      Received label 492529
      Extcommunity: RT:2:20
  Path-id 1 not advertised to any peer

Route Distinguisher: 2.2.2.3:113
BGP routing table entry for [5]:[0]:[0]:[24]:[41.11.2.0]:[0.0.0.0]/224, version
17369595 Paths: (1 available, best #1)
Flags: (0x000002) on xmit-list, is not in l2rib/evpn, is not in HW
  Advertised path-id 1
  Path type: external, path is valid, is best path
             Imported from 14.1.4.1:115:[5]:[0]:[0]:[24]:[41.11.2.0]:[0.0.0.0]/224 AS-
  Path: 11 , path sourced external to AS
    1.1.1.9 (metric 0) from 1.1.1.9 (14.1.4.1)
```

CHAPTER 5

Configuring MPLS Segment Routing OAM

This chapter contains information about configuring MPLS Segment Routing OAM

Overview of MPLS Segment Routing OAM

BGP MPLS segment Routing has been deployed on the Inspur CN6000 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 Inspur INOS-CN MPLS OAM is supported on Inspur CN6000 Series switches. 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 Inspur CN6000 Series switches, use the **feature mpls oam** CLI command. Use the **no feature mpls oam** CLI command to disable MPLS OAM on Inspur CN6000 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:

- MPLS segment routing OAM is supported on Inspur CN61108PC-V, CN61108TC-V, and CN6132Q-V switches. It is not supported on Inspur CN6048TP switches.
- 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 Inspur CN6100-V 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.
  (**Note**: Inspur CN61108PC-V, CN61108TC-V, and CN6132Q-V switches support **ping mpls**. Inspur CN6048TP switches do not support **ping mpls.**)
- 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 Inspur CN6100-V switches, 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 traceroute 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

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

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

IETF RFCs Supported for Label Switching