



MPLS Static Labeling

The MPLS static feature enables you to statically assign local labels to an IPv4 prefix. Also, Label Switched Paths (LSPs) can be provisioned for these static labels by specifying the next-hop information that is required to forward the packets containing static label.

If there is any discrepancy between labels assigned statically and dynamically, the router issues a warning message in the console log. By means of this warning message, the discrepancy can be identified and resolved.

The advantages of static labels over dynamic labels are:

- Improves security because the risk of receiving unwanted labels from peers (running a compromised MPLS dynamic labeling protocol) is reduced.
- Gives users full control over defined LSPs.
- Utilize system resources optimally because dynamic labeling is not processed.

Restrictions

- Static labeling on IPv6 packets is not supported.
- The router does not prevent label discrepancy at the time of configuring static labels. Any generated discrepancy needs to be subsequently cleared.
- Equal-cost multi-path routing (ECMP) is not supported.
- Interfaces must be explicitly configured to handle traffic with static MPLS labels.
- The MPLS per-VRF labels cannot be shared between MPLS static and other applications.
- [Forwarding Labeled Packets, on page 2](#)
- [Define Label Range and Enable MPLS Encapsulation, on page 2](#)
- [Identify and Clear Label Discrepancy, on page 4](#)
- [Configuring Backup within a Forwarding Set, on page 5](#)
- [Configuring Static LSP Next Hop Resolve, on page 8](#)
- [Configuring Static LSP Next Hop Resolve with Recursive Prefix, on page 9](#)
- [MPLS Static Labeling , on page 9](#)

Forwarding Labeled Packets

This section describes how labeled packets are forwarded in MPLS networks, how forwarding labeled packets are different from forwarding IP packets, how labeled packets are load-balanced, and what a LSR does with a packet with an unknown label.

Top Label Value

When a labeled packet is received, the label value at the top of the stack is looked up. The LSR sees the 20-bit field in the top label, which carries the actual value of the label. As a result of a successful lookup, the LSR learns:

- the next hop to which the packet is to be forwarded.
- what label operation to be performed before forwarding - swap, push, or pop.

The processing is always based on the top label, without regard to the possibility that in the past some other number of another label may have been "above it", or at present that some other number of another label may be below it. An unlabeled packet can be thought of as a packet whose label stack is empty (that is, a packet whose label stack has depth zero).

IP Lookup Versus Label Lookup

When a router receives an IP packet, an IP lookup is done. This means that the packet is looked up in the Cisco Express Forwarding (CEF) table. When a router receives a labeled packet, the label forwarding information base (LFIB) of the router is looked up. The router knows by looking at the protocol field in the Layer 2 header what type of packet it receives: a labeled packet or an IP packet.

Load Balancing Labeled Packets

If multiple equal-cost paths exist for an IPv4 prefix, Cisco IOS XR Software can load-balance labeled packets. When labeled packets are load-balanced, they can have the same or different outgoing labels. The outgoing labels are the same if the two links are between a pair of routers and both links belong to the platform label space. If multiple next-hop LSRs exist, the outgoing label for each path is usually different, because the next-hop LSRs assign labels independently.

Unknown Label

In regular operations, an LSR should receive only a labeled packet with a label at the top of the stack that is known to the LSR, because the LSR would have previously advertised that label. However, it is possible, in some cases, when something goes amiss in the MPLS network, the LSR starts receiving labeled packets with a top label that the LSR does not find in its LFIB. In such cases, the LSR drops the packet.

Define Label Range and Enable MPLS Encapsulation

By default, MPLS encapsulation is disabled on all interfaces. MPLS encapsulation has to be explicitly enabled on all ingress and egress MPLS interfaces through which the static MPLS labeled traffic travels.

Also, the dynamic label range needs to be defined. Any label that falls outside this dynamic range is available for manually allocating as static labels. The router does not verify statically-configured labels against the specified label range. Therefore, to prevent label discrepancy, ensure that you do not configure static MPLS labels that fall within the dynamic label range.



Note For Cisco IOS XR software release 7.5.2 onwards, MPLS static supports 200G Ethernet.

Configuration Example

You have to accomplish the following to complete the MPLS static labeling configuration. Values are provided as an example.

1. Define a dynamic label range, which in this task is set between 17000 and 18000.
2. Enable MPLS encapsulation on the required interface.
3. Setup a static MPLS LSP for a specific ingress label 24035.
4. Specify the forwarding information so that for packets that are received with the label, 24035, the MPLS protocol swaps labels and applies the label, 24036. After applying the new label, it forwards the packets to the next hop, 10.2.2.2, through the specified interface.

```
Router(config)#mpls label range table 0 17000 18000
Router(config)#commit
Router(config)#mpls static
Router(config-mpls-static)#interface HundredGigE 0/0/0/25
Router(config-mpls-static)#address-family ipv4 unicast
Router(config-mpls-static-af)#local-label 24035 allocate
Router(config-mpls-static-af-lbl)#forward
Router(config-mpls-static-af-lbl-fwd)#path 1 nexthop HundredGigE 0/0/0/27 10.2.2.2 out-label
24036
Router(config-mpls-static-af-lbl-fwd)# commit
```

Verification

Verify the interfaces on which MPLS is enabled

```
Router# show mpls interfaces
```

Interface	LDP	Tunnel	Static	Enabled
HundredGigE 0/0/0/25	No	No	Yes	Yes

Verify that the status is "Created" for the specified label value.

```
Router#show mpls static local-label all
```

Label	VRF	Type	Prefix	RW Configured	Status
24035	default	X-Connect	NA	Yes	Created

Check the dynamic range and ensure that the specified local-label value is outside this range.

```
Router#show mpls label range
```

```
Range for dynamic labels: Min/Max: 17000/18000
```

Verify that the MPLS static configuration has taken effect, and the label forwarding is taking place.

```
Router#show mpls lsd forwarding
```

```
In_Label, (ID), Path_Info: <Type>
```

```
24035, (Static), 1 Paths
  1/1: IPv4, 'default':4U, BE1.2, nh=10.20.3.1, lbl=35001, flags=0x0, ext_flags=0x0
```

Associated Commands

- mpls static
- mpls label range
- show mpls interfaces

Identify and Clear Label Discrepancy

During configuring or de-configuring static labels or a label range, a label discrepancy can get generated when:

- A static label is configured for an IP prefix that already has a binding with a dynamic label.
- A static label is configured for an IP prefix, when the same label value is dynamically allocated to another IP prefix.

Verification

Identify label discrepancy by using these show commands.

```
Router#show mpls static local-label discrepancy
Tue Apr 22 18:36:31.614 UTC
Label  VRF                Type          Prefix          RW Configured  Status
-----
24000  default              X-Connect     NA              Yes            Discrepancy
```

```
Router#show mpls static local-label all
Tue Apr 22 18:36:31.614 UTC
Label  VRF                Type          Prefix          RW Configured  Status
-----
24000  default              X-Connect     N/A            Yes            Discrepancy
24035  default              X-Connect     N/A            Yes            Created
```

```
Router#show log
```

```
Syslog logging: enabled (0 messages dropped, 0 flushes, 0 overruns)
  Console logging: level warnings, 199 messages logged
  Monitor logging: level debugging, 0 messages logged
  Trap logging: level informational, 0 messages logged
  Buffer logging: level debugging, 2 messages logged
```

```
Log Buffer (307200 bytes):
```

```
RP/0/RSP0/CPU0:Apr 24 14:18:53.743 : mpls_static[1043]:
%ROUTING-MPLS_STATIC-7-ERR_STATIC_LABEL_DISCREPANCY :
The system detected 1 label discrepancies (static label could not be allocated due to
conflict with other applications).
Please use 'clear mpls static local-label discrepancy' to fix this issue.
RP/0/RSP0/CPU0:Apr 24 14:18:53.937 : config[65762]: %MGBL-CONFIG-6-DB_COMMIT : Configuration
committed by user 'cisco'.
Use 'show configuration commit changes 1000000020' to view the changes.
```

Rectification

Label discrepancy is cleared by allocating a new label to those IP prefixes that are allocated dynamic label. The static label configuration takes precedence while clearing discrepancy. Traffic can be affected while clearing discrepancy.

```
Router# clear mpls static local-label discrepancy all
```

Verify that the discrepancy is cleared.

```
Router# show mpls static local-label all
```

```
Wed Nov 25 21:45:50.368 UTC
Label   VRF          Type          Prefix          RW Configured  Status
-----
24000   default      X-Connect     N/A             Yes            Created
24035   default      X-Connect     N/A             Yes            Created
```

Associated Commands

- show mpls static local-label discrepancy
- clear mpls static local-label discrepancy all

Configuring Backup within a Forwarding Set

Various types of FRR backups can be configured between links within a forwarding path set. You can configure the following types of FRR backups:

- Pure FRR Backup
- Reciprocal FRR backup
- One-way FRR backup

In pure FRR backup, there will be separate primary paths and backup paths. In reciprocal FRR backup, each path can act as both primary and backup. In one-way FRR backup, some paths act as both primary and backup while other paths may be just primary paths or backup paths.

Configuration Example: Pure FRR Backup

This example shows how to configure pure FRR backup within a forwarding path set.

```
RP/0/0/CPU0:Router# configure terminal
RP/0/0/CPU0:Router(config)# mpls static
RP/0/0/CPU0:Router(config-mpls-static)# lsp lsp1
RP/0/0/CPU0:Router(config-mpls-static-lsp)# in-label 25000 allocate
RP/0/0/CPU0:Router(config-mpls-static-lsp)# forward
RP/0/0/CPU0:Router(config-mpls-static-lsp-fwd)# path 1 nexthop hundredGigE 0/0/0/25 10.1.0.1
  out-label 25000 backup-id 2
RP/0/0/CPU0:Router(config-mpls-static-lsp-fwd)# path 2 nexthop hundredGigE 0/0/0/26 10.1.0.3
  out-label 25001 backup
RP/0/0/CPU0:Router(config-mpls-static-lsp-fwd)# exit
RP/0/0/CPU0:Router(config-mpls-static-lsp)# backup
RP/0/0/CPU0:Router(config-mpls-static-lsp-backup)# path 1 nexthop hundredGigE 0/0/0/27
  10.5.0.1 out-label pop backup-id 2
RP/0/0/CPU0:Router(config-mpls-static-lsp-backup)# path 2 nexthop hundredGigE 0/0/0/28
  10.6.0.2 out-label pop backup
RP/0/0/CPU0:Router(config-mpls-static-lsp-backup)# exit
```

The following table describes the forwarding behavior for pure FRR backup. Here P1-F and P2-F are the forwarding paths and P1-B and P2-B are the backup paths.

Action	Transient State	Interface Steady State	Forward Steady State
N/A	N/A	<ul style="list-style-type: none"> • P1-F: Up P2-F: Up • P1-B: Up P2-B: Up 	<ul style="list-style-type: none"> • P1-F: Flow P2-F: Backup • P1-B: N/A P2-B: N/A
P1-F Down	P1-F FRR to P2-F	<ul style="list-style-type: none"> • P1-F: Up P2-F: Down • P1-B: Up P2-B: Up 	<ul style="list-style-type: none"> • P1-F: Down P2-F: Flow • P1-B: Backup P2-B: N/A
P2-F Down	P2-F FRR to P1-B	<ul style="list-style-type: none"> • P1-F: Down P2-F: Down • P1-B: Up P2-B: Up 	<ul style="list-style-type: none"> • P1-F: Down P2-F: Down • P1-B: Flow P2-B: Backup
P1-B Down	P1-B FRR to P2-B	<ul style="list-style-type: none"> • P1-F: Down P2-F: Down • P1-B: Down P2-B: Up 	<ul style="list-style-type: none"> • P1-F: Down P2-F: Down • P1-B: Down P2-B: Flow

Configuration Example: Reciprocal FRR Backup

This example shows how to configure reciprocal FRR backup with in a forwarding path set.

```
RP/0/0/CPU0:Router# configure terminal
RP/0/0/CPU0:Router(config)# mpls static
RP/0/0/CPU0:Router(config-mpls-static)# lsp lsp1
RP/0/0/CPU0:Router(config-mpls-static-lsp)# in-label 25000 allocate
RP/0/0/CPU0:Router(config-mpls-static-lsp)# forward
RP/0/0/CPU0:Router(config-mpls-static-lsp-fwd)# path 1 nexthop hundredGigE 0/0/0/25 10.1.0.1
  out-label 25000 primary-and-backup backup-id 2
RP/0/0/CPU0:Router(config-mpls-static-lsp-fwd)# path 2 nexthop hundredGigE 0/0/0/26 10.1.0.3
  out-label 25001 primary-and-backup backup-id 1
RP/0/0/CPU0:Router(config-mpls-static-lsp-fwd)# exit
RP/0/0/CPU0:Router(config-mpls-static-lsp)# backup
RP/0/0/CPU0:Router(config-mpls-static-lsp-backup)# path 1 nexthop hundredGigE 0/0/0/27
  10.5.0.1 out-label pop primary-and-backup backup-id 2
RP/0/0/CPU0:Router(config-mpls-static-lsp-backup)# path 2 nexthop hundredGigE 0/0/0/28
  10.6.0.2 out-label pop primary-and-backup backup-id 1
RP/0/0/CPU0:Router(config-mpls-static-lsp-backup)# exit
```

The following table describes the forwarding behavior for reciprocal FRR backup.

Action	Transient State	Interface Steady State	Forward Steady State
N/A	N/A	<ul style="list-style-type: none"> • P1-F: Up P2-F: Up • P1-B: Up P2-B: Up 	<ul style="list-style-type: none"> • P1-F: Flow P2-F: Flow • P1-B: N/A P2-B: N/A
P2-F Down	P2-F FRR to P1-F	<ul style="list-style-type: none"> • P1-F: Down P2-F: Up • P1-B: Up P2-B: Up 	<ul style="list-style-type: none"> • P1-F: Flow P2-F: Down • P1-B: Backup P2-B: N/A
P1-F Down	P1-F FRR to P1-B	<ul style="list-style-type: none"> • P1-F: Down P2-F: Down • P1-B: Up P2-B: Up 	<ul style="list-style-type: none"> • P1-F: Down P2-F: Down • P1-B: Flow P2-B: Flow
P2-B Down	P2-B FRR to P1-B	<ul style="list-style-type: none"> • P1-F: Down P2-F: Down • P1-B: Up P2-B: Down 	<ul style="list-style-type: none"> • P1-F: Down P2-F: Down • P1-B: Flow P2-B: Down

Configuration Example: One-way FRR Backup

This example shows how to configure one-way FRR backup with in a forwarding path set.

```
RP/0/0/CPU0:Router# configure terminal
RP/0/0/CPU0:Router(config)# mpls static
RP/0/0/CPU0:Router(config-mpls-static)# lsp lsp1
RP/0/0/CPU0:Router(config-mpls-static-lsp)# in-label 25000 allocate
RP/0/0/CPU0:Router(config-mpls-static-lsp)# forward
RP/0/0/CPU0:Router(config-mpls-static-lsp-fwd)# path 1 nexthop hundredGigE 0/0/0/25 10.1.0.1
out-label 25000 backup-id 2
RP/0/0/CPU0:Router(config-mpls-static-lsp-fwd)# path 2 nexthop hundredGigE 0/0/0/26 10.1.0.3
out-label 25001 primary-and-backup
RP/0/0/CPU0:Router(config-mpls-static-lsp-fwd)# exit
RP/0/0/CPU0:Router(config-mpls-static-lsp)# backup
RP/0/0/CPU0:Router(config-mpls-static-lsp-backup)# path 1 nexthop hundredGigE 0/0/0/27
10.5.0.1 out-label pop backup-id 2
RP/0/0/CPU0:Router(config-mpls-static-lsp-backup)# path 2 nexthop hundredGigE 0/0/0/28
10.6.0.2 out-label pop primary-and-backup
RP/0/0/CPU0:Router(config-mpls-static-lsp-backup)# exit
```

The following table describes the forwarding behavior for one-way FRR backup.

Action	Transient State	Interface Steady State	Forward Steady State
N/A	N/A	<ul style="list-style-type: none"> • P1-F: Up P2-F: Up • P1-B: Up P2-B: Up 	<ul style="list-style-type: none"> • P1-F: Flow P2-F: Flow • P1-B: N/A P2-B: N/A
P2-F Down	P2-F NO-FRR to P1-F	<ul style="list-style-type: none"> • P1-F: Down P2-F: Up • P1-B: Up P2-B: Up 	<ul style="list-style-type: none"> • P1-F: Flow P2-F: Down • P1-B: Backup P2-B: N/A
P1-F Down	P1-F FRR to P1-B	<ul style="list-style-type: none"> • P1-F: Down P2-F: Down • P1-B: Up P2-B: Up 	<ul style="list-style-type: none"> • P1-F: Down P2-F: Down • P1-B: Flow P2-B: Flow
P1-B Down	P1-B FRR to P2-B	<ul style="list-style-type: none"> • P1-F: Down P2-F: Down • P1-B: Down P2-B: Up 	<ul style="list-style-type: none"> • P1-F: Down P2-F: Down • P1-B: Down P2-B: Flow

Configuring Static LSP Next Hop Resolve

You can specify the outgoing next hop instead of explicitly specifying the outgoing path while configuring static LSPs. This next hop is resolved using the routing information base (RIB) which provides a list of paths to auto-configure. While specifying the next hop for the incoming label in a static LSP, you can specify the next hop address with out the interface using the **resolve-nexthop** command.

The following restrictions apply for this feature:

- Only supports a single next hop address which may resolve to multiple paths.
- Non-default VRFs are not supported.

Configuration Example

This example shows how to configure the static LSP next hop without specifying the interface using the **resolve-nexthop** command.

```
Router# configure terminal
Router(config)# mpls static
Router(config-mpls-static)# lsp ipv6-2
Router(config-mpls-static-lsp)# in-label 25000 allocate per-prefix 2001:DB8:0:1::/64 or
24:24:1::/64
Router(config-mpls-static-lsp)# forward
Router(config-mpls-static-lsp-fwd)# path 1 resolve-nexthop 2001:DB8:0:2::64 out-label pop
Router(config-mpls-static-lsp-fwd)# exit
```


Configuring Static LSP Next Hop Resolve with Recursive Prefix

When a routing table entry references to another IP address and not to a directly connected exit interface, the next-hop IP address is resolved using another route with an exit interface. This is known as a recursive lookup because multiple lookups are required to resolve the next-hop IP address. Static LSP next hop resolve with recursive prefix feature supports resolution of recursive routes for static LSPs. In this feature, you can specify a next hop which is not directly connected using the **resolve-nexthop** command for a static LSP.

Restrictions

The following restrictions apply for this feature:

- Only eBGP routes are supported.

Configuration Example

This example shows how to configure the static LSP next hop resolve with recursive prefix. Here 192.168.2.1 is a recursive route learnt through eBGP.

```
Router# configure terminal
Router(config)# mpls static
Router(config-mpls-static)# lsp anycast_5001
Router(config-mpls-static-lsp)# in-label 5001 allocate
Router(config-mpls-static-lsp)# forward
Router(config-mpls-static-lsp-fwd)# path 1 resolve-nexthop 192.168.2.1 out-label pop
Router(config-mpls-static-lsp-fwd)# exit
```

Verification

This example shows how to verify the static LSP next hop resolve with recursive prefix configuration.

```
Router# show mpls static lsp anycast_5001 detail
```

LSP Name Status	Label	VRF	AFI	Type	Prefix	RW Configured
anycast_5001 Created	5001	default	N/A	X-Connect	N/A	Yes
PRIMARY SET:						
[resolve-mode: nexthop 192.168.2.1]						
Path 0 : nexthop BVI1 10.1.1.3, out-label Pop, Role: primary, Path-id: 0, Status: valid						
Path 1 : nexthop BVI1 10.1.1.4, out-label Pop, Role: primary, Path-id: 0, Status: valid						
Path 2 : nexthop BVI1 10.1.1.5, out-label Pop, Role: primary, Path-id: 0, Status: valid						
Path 3 : nexthop BVI1 10.1.1.6, out-label Pop, Role: primary, Path-id: 0, Status: valid						

MPLS Static Labeling

The MPLS static feature enables you to statically assign local labels to an IPv4 prefix. Also, Label Switched Paths (LSPs) can be provisioned for these static labels by specifying the next-hop information that is required to forward the packets containing static label.

If there is any discrepancy between labels assigned statically and dynamically, the router issues a warning message in the console log. By means of this warning message, the discrepancy can be identified and resolved.

The advantages of static labels over dynamic labels are:

- Improves security because the risk of receiving unwanted labels from peers (running a compromised MPLS dynamic labeling protocol) is reduced.
- Gives users full control over defined LSPs.
- Utilize system resources optimally because dynamic labeling is not processed.

Restrictions

- Static labeling on IPv6 packets is not supported.
- The router does not prevent label discrepancy at the time of configuring static labels. Any generated discrepancy needs to be subsequently cleared.
- Equal-cost multi-path routing (ECMP) is not supported.
- Interfaces must be explicitly configured to handle traffic with static MPLS labels.
- The MPLS per-VRF labels cannot be shared between MPLS static and other applications.