MPLS Static Labeling

Table 1: Feature History Table

Feature Name	Release Information	Feature Description
MPLS static labeling	Release 25.4.1	Introduced in this release on: Fixed Systems (8700 [ASIC: K100])(select variants only*) *This feature is supported on Cisco 8711-48Z-M routers.
MPLS static labeling	Release 25.1.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100])(select variants only*) *This feature is supported on Cisco 8011-4G24Y4H-I routers.

MPLS static labeling	Release 24.4.1	Introduced in this release on: Fixed Systems (8200 [ASIC: P100], 8700 [ASIC: P100, K100])(select variants only*); Modular Systems (8800 [LC ASIC: Q100, P100])(select variants only*) MPLS Static Labeling allows for manual assignment of labels to specific routes, providing fine-grained control over label distribution and traffic engineering. This feature is ideal for network scenarios where predictable and consistent routing paths are required. By configuring static labels, network operators can optimize resource allocation and ensure precise traffic management without relying on dynamic label distribution protocols. *Previously this feature was supported on Q200 and Q100. • 8712-MOD-M • 8212-48FH-M • 8711-32FH-M • 88-LC1-52Y8H-EM • 88-LC1-12TH24FH-E • 88-LC1-36EH

The MPLS static feature enables you to statically assign local labels to an IPv4/Ipv6 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

- 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

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 with in 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 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 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	• P1-F: Up P2-F: Up • P1-B: Up P2-B: Up	• P1-F: Flow P2-F: Backup • P1-B: N/A P2-B: N/A
P1-F Down	P1-F FRR to P2-F	• P1-F: Up P2-F: Down • P1-B: Up P2-B: Up	• P1-F: Down P2-F: Flow • P1-B: Backup P2-B: N/A
P2-F Down	P2-F FRR to P1-B	• P1-F: Down P2-F: Down • P1-B: Up P2-B: Up	• P1-F: Down P2-F: Down • P1-B: Flow P2-B: Backup
P1-B Down	P1-B FRR to P2-B	• P1-F: Down P2-F: Down • P1-B: Down P2-B: Up	• 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	• P1-F: Up P2-F: Up • P1-B: Up P2-B: Up	• P1-F: Flow P2-F: Flow • P1-B: N/A P2-B: N/A
P2-F Down	P2-F FRR to P1-F	• P1-F: Down P2-F: Up • P1-B: Up P2-B: Up	• P1-F: Flow P2-F: Down • P1-B: Backup P2-B: N/A
P1-F Down	P1-F FRR to P1-B	• P1-F: Down P2-F: Down • P1-B: Up P2-B: Up	• P1-F: Down P2-F: Down • P1-B: Flow P2-B: Flow
P2-B Down	P2-B FRR to P1-B	• P1-F: Down P2-F: Down • P1-B: Up P2-B: Down	• 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	• P1-F: Up P2-F: Up • P1-B: Up P2-B: Up	• P1-F: Flow P2-F: Flow • P1-B: N/A P2-B: N/A
P2-F Down	P2-F NO-FRR to P1-F	• P1-F: Down P2-F: Up • P1-B: Up P2-B: Up	• P1-F: Flow P2-F: Down • P1-B: Backup P2-B: N/A
P1-F Down	P1-F FRR to P1-B	• P1-F: Down P2-F: Down • P1-B: Up P2-B: Up	• P1-F: Down P2-F: Down • P1-B: Flow P2-B: Flow
P1-B Down	P1-B FRR to P2-B	• P1-F: Down P2-F: Down • P1-B: Down P2-B: Up	• 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 look up 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             Label   VRF             AFI  Type         Prefix          RW Configured
   Status
-------------------- ------- --------------- ---- ------------ ----------------
--------------- --------
anycast_5001         5001    default         N/A  X-Connect    N/A              Yes
      Created
  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

Table 2: Feature History Table

Feature Name	Release Information	Feature Description
MPLS static labeling	Release 25.4.1	Introduced in this release on: Fixed Systems (8700 [ASIC: K100])(select variants only*) *This feature is supported on Cisco 8711-48Z-M routers.
MPLS static labeling	Release 25.1.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100])(select variants only*) *This feature is supported on Cisco 8011-4G24Y4H-I routers.
MPLS static labeling	Release 24.4.1	Introduced in this release on: Fixed Systems (8200 [ASIC: P100], 8700 [ASIC: P100, K100])(select variants only*); Modular Systems (8800 [LC ASIC: Q100, P100])(select variants only*) MPLS Static Labeling allows for manual assignment of labels to specific routes, providing fine-grained control over label distribution and traffic engineering. This feature is ideal for network scenarios where predictable and consistent routing paths are required. By configuring static labels, network operators can optimize resource allocation and ensure precise traffic management without relying on dynamic label distribution protocols. *Previously this feature was supported on Q200 and Q100. • 8712-MOD-M • 8212-48FH-M • 8711-32FH-M • 88-LC1-52Y8H-EM • 88-LC1-12TH24FH-E • 88-LC1-36EH

The MPLS static feature enables you to statically assign local labels to an IPv4/Ipv6 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

- 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.

MPLS Static Forwarding Over A BVI

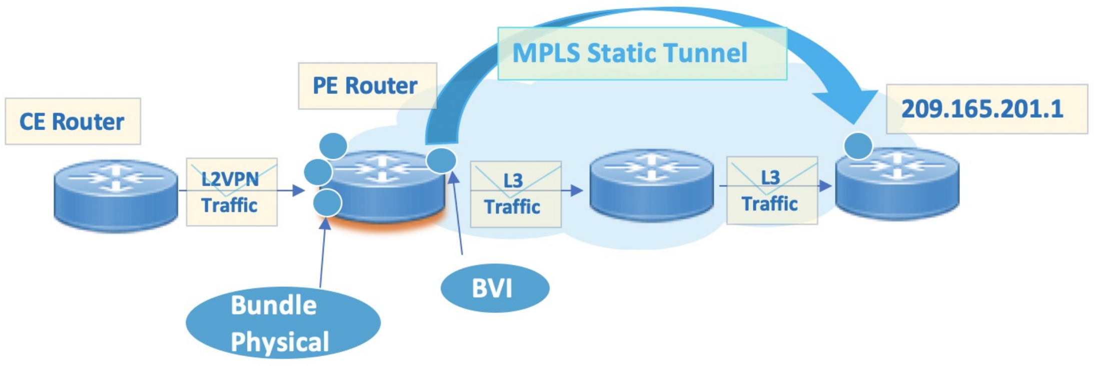
Table 3: Feature History Table

Feature Name	Release Information	Feature Description
MPLS Static Forwarding Over A BVI	Release 25.1.1	Introduced in this release on: Fixed Systems (8700 [ASIC: K100])(select variants only*) *This feature is supported on Cisco 8712-MOD-M routers.
MPLS Static Forwarding Over A BVI	Release 24.4.1	Introduced in this release on: Fixed Systems (8200 [ASIC: P100], 8700 [ASIC: P100])(select variants only*); Modular Systems (8800 [LC ASIC: P100])(select variants only*) *This feature is supported on: • 8212-48FH-M • 8711-32FH-M • 88-LC1-36EH • 88-LC1-12TH24FH-E • 88-LC1-52Y8H-EM

Feature Name	Release Information	Feature Description
MPLS Static Forwarding Over A BVI	Release 7.3.1	The router can receive MPLS L2VPN traffic from an L2 bridge domain, and forward the L3 (customer) traffic over an egress BVI, using an MPLS static LSP. For the incoming L2VPN traffic, the BVI serves as an L3 gateway. Since the router can perform switching for L2 traffic and routing for incoming L3 MPLS traffic, it enhances flexibility for transporting MPLS traffic.

Consider this sample topology, connecting a CE router to a PE router.

Figure 1: MPLS Static Forwarding Over A BVI



Pointers

- L2VPN packets are attached to specific bridge domains, and correspond to a VLAN (or 802.1Q tag). In turn, the VLANs are associated with specific bundle or physical interfaces for sending traffic between the CE and PE routers. These associations have to be configured on the CE and PE routers for transporting L2VPN traffic.
- The L2VPN traffic encapsulates (IPv4 or IPv6) customer payload, and is sent from the CE router to the PE router.
- The PE router does an MPLS label lookup on the incoming MPLS traffic, and removes the VLAN (or 802.1Q) header. In general, the router can perform a label operation like swap, PHP, or pop. After removing the VLAN header, the (previously) encapsulated IP traffic is sent towards the bridge-group virtual interface (BVI).
- With BVI support for the MPLS static function, the incoming labelled traffic can be resolved using a static LSP. The BVI resolves the nexthop to an L3 interface.

BVI pointers

- A BVI next hop can be a static route, a directly connected route, or a route resolved through BGP or an IGP.
- Only an MPLS static LSP can use a BVI as a next hop.

Configuration

The configurations explain how to enable forwarding of (incoming) L2VPN traffic over a (outgoing) BVI, through an MPLS static LSP.

Interfaces Configuration

- The **l2transport** keyword indicates that the interface is an L2 interface, and the L2 traffic belongs to the VLANs specified in the dot1q tags.
- The **rewrite ingress tag pop** command form instructs the router to remove the 802.1Q (or VLAN) tag and forward the payload.
- Corresponding configurations should also be enabled on the CE router.

```
Router# configure terminal
Router(config)# interface Bundle-Ether101.101 l2transport
Router(config-if)# encapsulation dot1q 2001
Router(config-if)# rewrite ingress tag pop 1 symmetric
Router(config-if)# mtu 2000

Router# configure terminal
Router(config)# interface Bundle-Ether102.101 l2transport
Router(config-if)# encapsulation dot1q 3001
Router(config-if)# rewrite ingress tag pop 1 symmetric
Router(config-if)# mtu 2000

Router# configure terminal
Router(config)# interface HundredGigE0/11/0/25.101 l2transport
Router(config-if)# encapsulation dot1q 101
Router(config-if)# rewrite ingress tag pop 1 symmetric

Router# configure terminal
Router(config)# interface HundredGigE0/11/0/31.101 l2transport
Router(config-if)# encapsulation dot1q 1001
Router(config-if)# rewrite ingress tag pop 1 symmetric
Router(config-if)# exit
Router(config)# commit
```

BVI Configuration

The BVI acts as an L3 gateway for the ingress L2VPN traffic. The customer IP traffic is sent to the BVI IP address that is configured in this task.

```
Router# configure terminal
Router(config)# interface BVI101
Router(config-if)# ipv4 address 192.0.2.1 255.255.255.0
Router(config-if)# ipv6 address 2001:DB8:A:B::1/64
Router(config-if)# mac-address 0.10.5500
Router(config-if)# load-interval 30
Router(config-if)# commit
```

L2VPN Configuration

- L2VPN is configured, and a bridge domain (bd1) is associated with it.
- A bundle interface and BVI are associated with the BD. The instructions enable VLAN traffic received at the bundle interface (Bundle-Ether101.101) to be routed through the L3 gateway, BVI BVI101. VLAN-to-interface association was done in an earlier step.

```
Router# configure
Router(config)# l2vpn
Router(config-l2vpn)# bridge group bg1
Router(config-l2vpn-bg)# bridge-domain bd1
Router(config-l2vpn-bg-bd)# interface Bundle-Ether101.101
Router(config-l2vpn-bg-bd-ac)# routed interface BVI101
```

Similar configurations are done for other interfaces.

```
Router(config-l2vpn-bg-bd)# interface Bundle-Ether102.101
Router(config-l2vpn-bg-bd-ac)# routed interface BVI101
.
Router(config-l2vpn-bg-bd)# interface HundredGigE0/11/0/25.101
Router(config-l2vpn-bg-bd-ac)# routed interface BVI101
.
Router(config-l2vpn-bg-bd)# interface HundredGigE0/11/0/31.101
Router(config-l2vpn-bg-bd-ac)# routed interface BVI101
Router(config-l2vpn-bg-bd-ac)# commit
```

MPLS Static Configuration

- BVI BVI101 is associated with the MPLS static LSP with a destination IP address 209.165.201.1.
- MPLS configurations (such as attaching an incoming label [1000101 and removing the MPLS label at the LSP egress device) are added.

```
Router# configure terminal
Router(config)# mpls static
Router(config-mpls-static)# lsp bvi-101
Router(config-mpls-static-lsp)# in-label 1000101 allocate
Router(config-mpls-static-lsp)# forward
Router(config-mpls-static-lsp-fwd)# path 1 resolve-nexthop 209.165.201.1 out-label pop
Router(config-mpls-static-lsp-fwd)# commit
```

Traffic Engineering

- Segment Routing (SR)is used for Traffic Engineering (TE).
- For the specified SR policy, the end point (or destination) is 209.165.201.1, same as the MPLS LSP destination for transporting L2VPN traffic.

```
Router(config)# segment-routing traffic-eng
Router(config-sr-te)# policy sr-static-mpls-to-LER2
Router(config-sr-te-policy)# binding-sid mpls 250100
Router(config-sr-te-policy)# color 100 end-point ipv4 209.165.201.1
Router(config-sr-te-policy)# candidate-paths
Router(config-sr-te-policy-path)# preference 100
Router(config-sr-te-policy-path-pref)# explicit segment-list to-LER2
Router(config-sr-te-policy-path-pref)# commit
```

A segment list is created with MPLS labels.

```
Router(config)# segment-routing traffic-eng
Router(config-sr-te)# segment-list to-LER2
Router(config-sr-te-sl)# index 10 mpls label 1000101
Router(config-sr-te-sl)# index 20 mpls label 300501
Router(config-sr-te-sl)# commit
```

Verification

Use this command to view MPLS static LSP details specific to the BVI:

```
Router# show mpls static lsp bvi-101 detail

LSP Name            Label   VRF            AFI  Type         Prefix          RW Configured
```

```
   Status
-------------------- ------- --------------- ---- ------------ ---------------- ------
bvi-101              1000101 default         N/A  X-Connect    N/A              Yes
      Created
  PRIMARY SET:
    [resolve-mode: nexthop 209.165.201.1]
    Path 0 : nexthop Bundle-Ether1 198.51.100.1, out-label Pop, Role: primary, Path-id: 2,
 Status: valid
    Path 1 : nexthop Bundle-Ether2 203.0.113.100, out-label Pop, Role: primary, Path-id:
1, Status: valid
```

Use this command to view MPLS static LSP details specific to the router:

```
Router# show mpls static local-label 1000101 detail

Label   VRF             Type         Prefix           RW Configured   Status
------- --------------- ------------ ---------------- --------------- --------
1000101 default         X-Connect    N/A              Yes             Created

  PRIMARY SET:
    [resolve-mode: nexthop 209.165.201.1]
    Path 0 : nexthop Bundle-Ether1 198.51.100.1, out-label Pop, Role: primary, Path-id: 2,
 Status: valid
    Path 1 : nexthop Bundle-Ether2 203.0.113.100, out-label Pop, Role: primary, Path-id:
1, Status: valid
```

MPLS Over GRE Tunnels

Table 4: Feature History Table

Feature Name	Release Information	Feature Description
MPLS Over GRE Tunnels	Release 25.4.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100])(select variants only*) *This feature is supported on: • 8011-12G12X4Y-A • 8011-12G12X4Y-D

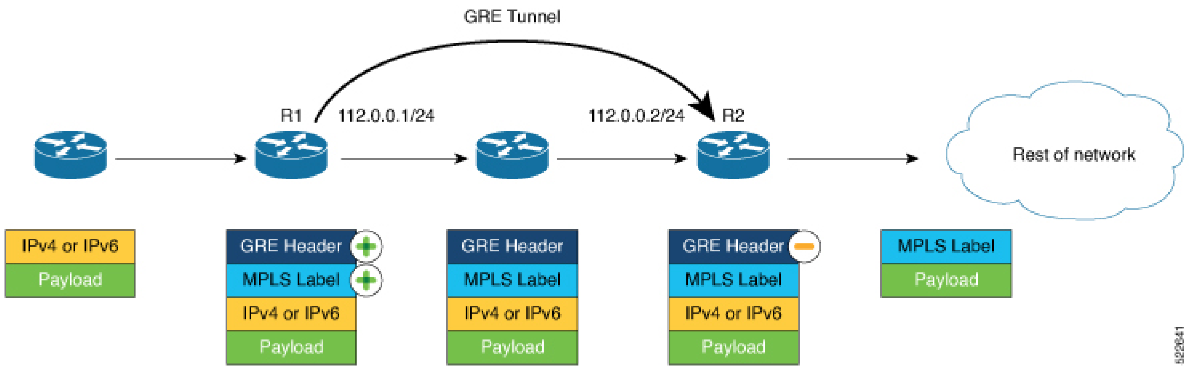
Feature Name	Release Information	Feature Description
MPLS Over GRE Tunnels	Release 7.5.2	The MPLS over generic routing encapsulation (MPLSoGRE) provides a mechanism for tunneling MPLS packets over a non-MPLS network. This feature uses MPLS over GRE to encapsulate MPLS packets inside GRE tunnels to create a virtual point-to-point link across non-MPLS networks. With this feature, the core network can be configured with IPv4 addresses to interconnect the MPLS networks through the IP network.

MPLS over GRE supports MPLS static forwarding over a GRE tunnel at line rate, which is the normal speed at which the traffic is sent through networks. For more information on line rate, see the Cisco 8000 Series Routers Data Sheet.

You can configure a provider router to send incoming customer traffic over the GRE tunnel, addressed to a set of load-balancing servers.

The GRE tunnel is configured with encapsulation that adds an MPLS packet and a GRE header to the incoming packet at the starting point of the tunnel. When the packet reaches the endpoint of the GRE tunnel, the GRE header is removed and the payload is forwarded to the destination based on the MPLS label.

Figure 2: MPLS Over GRE Tunnel



In the image, you can see that the GRE tunnel begins at router R1. R1 uses the policy based routing (PBR) process for GRE tunnel encapsulation, adds an MPLS label to the incoming packet, and then adds a GRE header. Then it sends the traffic towards router R2.

R2 uses the PBR process for GRE tunnel decapsulation, and based on the MPLS label, it forwards the traffic towards its destination.

Configuration Example

This example shows how to enable MPLS static forwarding over GRE tunnel.

Configuration on router R1.

1. Configure a tunnel interface.
2. Configure the mode of encapsulation as GRE for the tunnel interface.
3. Configure a policy map and redirect the traffic to next hop IP.
4. Assign the policy map to a VLAN subinterface.
5. Configure MPLS out labels to be applied to the incoming packets.

Configuration on router R2.

1. Configure a tunnel interface.
2. Configure the mode of decapsulation as GRE for the tunnel interface.
3. Configure a class map with match criteria for the source and destination IP addresses.
4. Configure a policy map and specify the class name configured in the class map.
5. Configure GRE decapsulation.

GRE Tunnel Configuration on R1

The GRE tunnel starts on R1.

The GRE tunnel destination must be a valid IPv4 address.

```
Router# configure
Router(config)# interface tunnel-ip1
Router(config-if)# ipv4 address 112.0.0.1 255.255.255.0
Router(config-if)# tunnel mode gre ipv4 encap
Router(config-if)# tunnel source Loopback 0
Router(config-if)# tunnel destination 50.0.0.1
Router(config-if)# commit
```

PBR Configuration for Encapsulation on R1

GRE encapsulation must be based on a policy map. Configure a policy map and redirect the traffic to next hop IP.

```
Router(config)# policy-map type pbr PBR_ENCAP_1
Router(config-pmap)# class type traffic class-default
Router(config-pmap-c)# redirect ipv4 nexthop 111.0.0.1
Router(config-pmap-c)# end-policy-map
```

Assign the policy map to one or more VLAN subinterfaces.

```
Router(config)# interface Bundle-Ether 111.1
Router(config-if)# service-policy type pbr input PBR_ENCAP_1
Router(config-if)# ipv4 address 50.0.2.1 255.255.255.0
Router(config-if)# encapsulation dot1q 1
```

MPLS Static Configuration on R1

Configure MPLS out labels. Ensure that the out label is the same for all paths of an MPLS static label switch path(LSP). You can configure up to a maximum of 16 paths. After applying the labels, the packets are forwarded to the specified next hop.

```
Router(config)# mpls static
Router(config-mpls-static)# lsp v4-encap-payload-1
Router(config-mpls-static-lsp)# in-label 10001 allocate per-prefix 111.0.0.1/32
Router(config-mpls-static-lsp)# forward
Router(config-mpls-static-lsp-fwd))# path 1 nexthop tunnel-ip1 out-label 20001
Router(config-mpls-static-lsp-fwd))# path 2 nexthop tunnel-ip2 out-label 20001
Router(config-mpls-static-lsp-fwd))# path 3 nexthop tunnel-ip3 out-label 20001
Router(config-mpls-static-lsp-fwd))# path 4 nexthop tunnel-ip4 out-label 20001
Router(config-mpls-static-lsp-fwd)# commit
```

GRE Tunnel Configuration on R2

The GRE tunnel stops on R2.

```
Router # configure
Router(config)# interface tunnel-ip1
Router(config-if)# ipv4 address 112.0.0.2 255.255.255.0
Router(config-if)# tunnel mode gre ipv4 decap
Router(config-if)# tunnel source Loopback 0
Router(config-if)# tunnel destination 10.0.0.1
Router(config-if)# commit
```

PBR Configuration for GRE Tunnel Decapsulation on R2

```
Router(config)# class-map type traffic match-all test_gre1
Router(config-cmap)# match protocol gre
Router(config-cmap)# match destination-address ipv4 50.0.0.1 255.255.255.255
Router(config-cmap)# match source-address ipv4 10.0.0.1 255.255.255.255
Router(config-cmap)# end-class-map
Router(config)# policy-map type pbr P1-test
Router(config-pmap)# class type traffic test_gre1
Router(config-pmap-c)#decapsulate gre
Router(config-pmap-c)# end-policy-map
Router(config)# vrf-policy vrf default address-family ipv4 policy type pbr input P1-test
```

Running Configuration

Use the following show commands to view the configuration.

Tunnel-IP configuration on R1

```
Router# show running-config interface tunnel-ip 1

interface tunnel-ip1
    ipv4 address 112.0.0.1 255.255.255.0
    tunnel mode gre ipv4 encap
    tunnel source Loopback 0
    tunnel destination 50.0.0.1
!
```

PBR Configuration for GRE Tunnel Encapsulation on R1

```
Router# show running-config policy-map type pbr *

policy-map type pbr PBR_ENCAP_1
 class type traffic class-default
  redirect ipv4 nexthop 111.0.0.1
 !
 end-policy-map
 !
```

MPLS Static Configuration on R1

```
Router# show running-config mpls static

mpls static
  lsp v4-encap-payload-1
     in-label 10001 allocate per-prefix 111.0.0.1/32
     forward
        path 1 nexthop tunnel-ip1 out-label 20001
        path 2 nexthop tunnel-ip2 out-label 20001
        path 3 nexthop tunnel-ip3 out-label 20001
        path 4 nexthop tunnel-ip4 out-label 20001
        !
```

Tunnel-IP Configuration on R2

```
Router# show running-config int tunnel-ip 1

interface tunnel-ip1
 ipv4 address 112.0.0.2 255.255.255.0
 tunnel mode gre ipv4 decap
 tunnel source Loopback 0
 tunnel destination 10.0.0.1
!
```

PBR Configuration for GRE Tunnel Decapsulation on R2

```
Router# show running-config class-map type traffic match-all

class-map type traffic match-all test_gre1
    match protocol gre
    match destination-address ipv4 50.0.0.1 255.255.255.255
    match source-address ipv4 10.0.0.1 255.255.255.255
end-class-map
!
policy-map type pbr P1-test
   class type traffic test_gre1
   decapsulate gre
!
   class type traffic class-default
!
end-policy-map
!

vrf-policy
vrf default address-family ipv4 policy type pbr input P1-test
!
```