Implementing MPLS Static Labeling

The MPLS static feature enables you to statically assign local labels to an IPv4 prefix per VRF. 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.

Static labels are more advantageous than dynamic labels because static labels:

- Improve 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.Gives users full control over defined LSPs.
- Utilize system resources optimally because dynamic labeling is not processed.

To perform static binding of MPLS labels, you need to:

- Enable MPLS Encapsulation on an Interface, on page 5
- Define a Range for Static MPLS Labels, on page 6
- Allocate static label:
  - Setup a Static LSP, on page 7

    or

    Allocate Static MPLS Label to an IP Prefix and Configure a LSP, on page 8
  - Allocate Static MPLS Label for a Specific VRF, on page 9
- Verify MPLS Static Bindings, on page 10
- Identify and Clear Label Discrepancy, on page 12

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.

Feature History for Implementing MPLS Static Labeling

Release	Modification
Release 5.1.1	This feature was introduced.
Release 5.2.2	Recursive Label Statistics feature was added.
Release 5.3.2	MPLS Top Label Hash for OAM Packets feature was added.

- Recursive Label Statistics, on page 2
- MPLS Top Label Hash for OAM Packets, on page 3
- Enable MPLS Encapsulation on an Interface, on page 5
- Define a Range for Static MPLS Labels, on page 6
- Setup a Static LSP, on page 7
- Allocate Static MPLS Label to an IP Prefix and Configure a LSP, on page 8
- Allocate Static MPLS Label for a Specific VRF, on page 9
- Verify MPLS Static Bindings, on page 10
- Configuring Top Label Hash, on page 11
- Identify and Clear Label Discrepancy, on page 12
- Configure Top Label Hash: Example, on page 13

Recursive Label Statistics

The MPLS static feature is enhanced to provide recursive Label Switched Path (LSP) statistics for labels created using MPLS static configuration. The recursive label statistics feature helps in identifying the unique source and destination port LSPs.

Restrictions

- LSP statistics works only for labels allocated through MPLS static configuration in a VRF, which means that it only works for recursive VPN labels.
- No packet rate support.
- During MPLS static configuration or de-configuration, label discrepancies can get generated.

Use the **clear mpls static local-label discrepancy** command to clear any discrepancy between statically allocated and dynamically allocated local labels. It is recommended to execute this command upon removal of a static configuration, so that the label prefix is reallocated to the dynamic label range which then will also free the allocated statistic point. Use the **all** keyword to clear all label discrepancies. The static label configuration takes precedence while clearing discrepancy.

MPLS Top Label Hash for OAM Packets

The MPLS top label hash feature lets label switching routers (LSRs) to be based on top label hashing for MPLS OAM packets. LSRs commonly generate a hash of the label stack or some elements of the label stack as a method of discriminating flows, and use this discriminator to distribute the flows over available equal cost multipaths (ECMPs) that exist in the network.

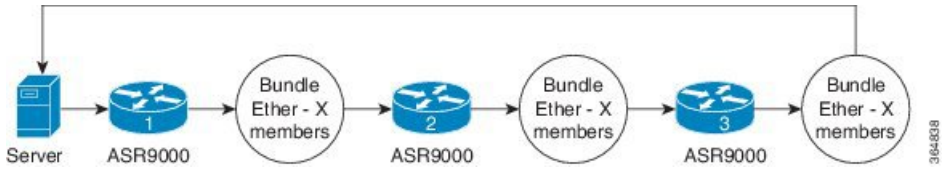
In order to determine which path (ECMP) or link aggregation group (LAG) member to choose, the system computes a hash. Certain bits out of this hash are used to identify member or path to be taken.

To configure top label hash, use the **top-label-hash** command under the MPLS static address family IPv4 unicast submode.

The benefit of the top label hash feature is that it can be used when a user wants to monitor all bundles along with members to ensure they are up and running.

Use Case Scenario

Consider the following example:



All devices have static LSPs to forward the traffic corresponding to monitoring. OAM server constructs the packets with number of labels equal to the number of hops between two ends (server to server). So, for the example shown, the packet has four labels.

Packet example:

- Packet 1: label A1, B1, C1, D1
- Packet 2: label A2, B2, C2, D2
- Packet 3: Label A3, B3, C3, D3

Top label hashing is required because you do not want to hash based on "Dx" label in every hop.

Top label hashing allows, ASR9000-1 to make decision based on "Ax" label, ASR9000-2 to make decision on "Bx" label, ASR9000-3 to make decision based on "Cx" label and so on. The user needs to define the sequence of labels to be used, such that each label uses different bundle member.

If server receives the packet back as expected, then that means end-to-end path is good and members are functioning correctly.

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

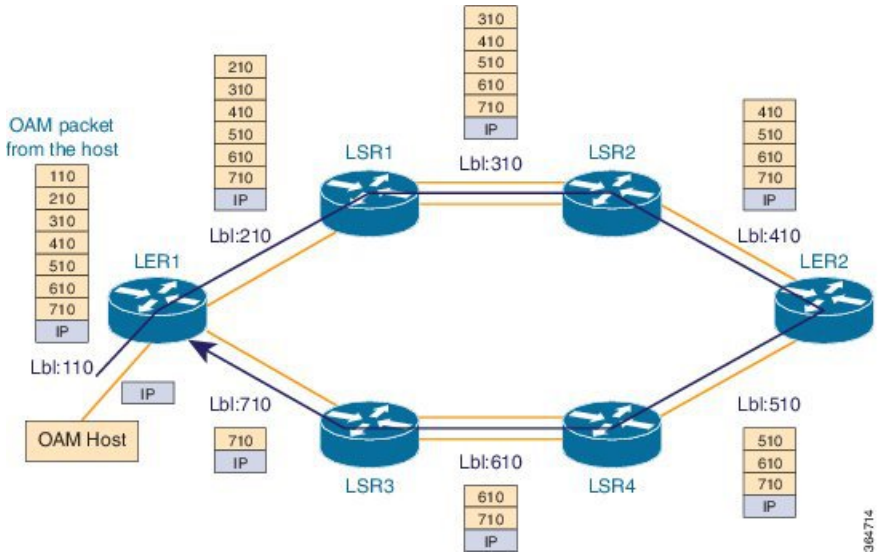
In regular operations, an LSR should receive only a labeled packet with a label at the top of the stack that is known to the LSR, because the LSR would have previously advertised that label. However, it is possible, in some cases, when something goes amiss in the MPLS network, the LSR starts receiving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

Functional Overview: Top Label Hash

The user configured top label hash value is pushed to the hardware abstraction layer (HAL). The FIB of the router computes the hash value based on the LSP paths and programs this hash value in the data plane. Based on the hash value, the OAM packet is then forwarded to the LAG member.

Figure 1: MPLS Top Label Hash for OAM Packets

This figure shows an OAM packet traversing different LAG members based on the top label hash value.



The OAM host sends the OAM packet with full label stack of the static LSP path. The packet is loop over bundle interface from LER1 > LSR1 > LSR2 > LER2 > LSR4 > LSR3 > LER1.

Each static LSP out-label is programmed as a 'pop' label.

Enable MPLS Encapsulation on an Interface

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.

SUMMARY STEPS

1. **configure**
2. **mpls static**
3. **interface** *interface*
4. **commit**

DETAILED STEPS

Procedure

Step 1 **configure**

Step 2 **mpls static**

Example:

```
RP/0/RSP0/CPU0:router(config)# mpls static
```

Enters MPLS-static configuration mode.

Step 3 **interface** *interface*

Example:

```
RP/0/RSP0/CPU0:router(config-mpls-static)# interface gigabitethernet 0/0/0/3
```

Enables MPLS encapsulation on the specified interface.

Step 4 **commit**

What to do next

To verify the interfaces on which MPLS is enabled, use the **show mpls interfaces** command from the EXEC mode. For example:

```
RP/0/RSP0/CPU0:router# show  mpls interfaces
Mon May 12 06:21:30.937 DST
Interface                  LDP      Tunnel   Static   Enabled
-------------------------- -------- -------- -------- --------
GigabitEthernet0/0/0/3     No       No       Yes      Yes
```

For the interface on which MPLS static is enabled, the "Static" column displays "Yes".

Define a Range for Static MPLS Labels

The MPLS label range configuration defines the dynamic label range. 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.

Note The allocable range for MPLS labels is from 16 to 1048575. Label values from 0 to15 are reserved according to RFC-3032.

SUMMARY STEPS

1. **configure**
2. **mpls label range** *minimum_value maximum_value*
3. **commit**

DETAILED STEPS

Procedure

Step 1 **configure**

Step 2 **mpls label range** *minimum_value maximum_value*

Example:

```
RP/0/RSP0/CPU0:router(config)# mpls label range 20000 30000
```

Specifies the range through which dynamic MPLS labels are allocated. All labels falling outside this range (16 to 19999 and 30001 to 1048575) can be manually allocated as static labels.

Step 3 **commit**

Setup a Static LSP

In this task, a static MPLS LSP is setup for a specific ingress label.

SUMMARY STEPS

1. **configure**
2. **mpls static**
3. **address-family ipv4 unicast**
4. **local-label** *label-value* **allocate**
5. **forward path** *path_id* **nexthop** *nexthop_address interface_type interface_id* **out-label** *outgoing_label*
6. **commit**

DETAILED STEPS

Procedure

Step 1 **configure**

Step 2 **mpls static**

Example:

```
RP/0/RSP0/CPU0:router(config)# mpls static
```

Enters MPLS-static configuration mode.

Step 3 **address-family ipv4 unicast**

Example:

```
RP/0/RSP0/CPU0:router(config-mpls-static)# address-family ipv4 unicast
```

Applies the static configuration to an IPv4 address family in the default VRF.

Step 4 **local-label** *label-value* **allocate**

Example:

```
RP/0/RSP0/CPU0:router(config-mpls-static-af)# local-label 30500 allocate
```

Specifies the incoming label value as 30500.

Step 5 **forward path** *path_id* **nexthop** *nexthop_address interface_type interface_id* **out-label** *outgoing_label*

Example:

```
RP/0/RSP0/CPU0:router(config-mpls-static-af-lbl)# forward path 1 nexthop 10.2.2.2 gigabitEthernet
0/0/0/1 out-label 30501
```

For packets that are received with the label, 30500, the MPLS protocol swaps labels and applies the label, 30501. After applying the new label, it forwards the packets to the next hop, 10.2.2.2, through the GigabitEthernet interface, 0/0/0/1.

Step 6 **commit**

Allocate Static MPLS Label to an IP Prefix and Configure a LSP

Static MPLS label bindings for IP prefixes are used by MPLS applications such as Label Distribution Protocol (LDP) or Border Gateway Protocol (BGP) for MPLS switching. It is possible to define a static LSP for the static label.

SUMMARY STEPS

1. **configure**
2. **mpls static**
3. **address-family ipv4 unicast**
4. **local-label** *label-value* **allocate per-prefix** *IPv4_prefix_entry*
5. **forward path** *path_id* **nexthop** *nexthop_address* **out-label** *outgoing_label*
6. **commit**

DETAILED STEPS

Procedure

Step 1 **configure**

Step 2 **mpls static**

Example:

```
RP/0/RSP0/CPU0:router(config)# mpls static
```

Enters MPLS-static configuration mode.

Step 3 **address-family ipv4 unicast**

Example:

```
RP/0/RSP0/CPU0:router(config-mpls-static)# address-family ipv4 unicast
```

Applies the static configuration to an IPv4 address family in the default VRF.

Step 4 **local-label** *label-value* **allocate per-prefix** *IPv4_prefix_entry*

Example:

```
RP/0/RSP0/CPU0:router(config-mpls-static-af)# local-label 30500 allocate per-prefix 100.1.1.0/24
```

The MPLS protocol requests label 30500 to be statically bound as a local label for the prefix 100.1.1.0/24.

Step 5 **forward path** *path_id* **nexthop** *nexthop_address* **out-label** *outgoing_label*

Example:

```
RP/0/RSP0/CPU0:router(config-mpls-static-af-lbl)# forward path 1 nexthop 10.2.2.2 out-label 30501
```

For packets that are received with the label, 30500, the MPLS protocol swaps labels and applies the label, 30501. After applying the new label, it forwards the packets to the next hop, 10.2.2.2.

Example:

```
RP/0/RSP0/CPU0:router(config-mpls-static-af-lbl)# forward path 1 nexthop gigabitEthernet 0/0/0/4
out-label pop
```

For packets that are received with the label, 30500, the MPLS protocol removes the existing label. After removing the label, it forwards the packets to the next hop through the egress interface, GigabitEthernet 0/0/0/4.

Step 6 **commit**

Allocate Static MPLS Label for a Specific VRF

In this task, a static MPLS label is allocated to an IP prefix for a specific VRF.

Note When a static MPLS label is allocated to an IP prefix for a specific VRF, it is not possible to define a static LSP for that static label.

SUMMARY STEPS

1. **configure**
2. **mpls static**
3. **vrf** *vrf_name* **address-family ipv4 unicast**
4. • **local-label** *label-value* **allocate per-prefix** *IPv4_prefix_entry*
   • **local-label** *label-value* **allocate per-vrf** forward **path** *path-id* **pop-and-lookup**
5. **commit**

DETAILED STEPS

Procedure

Step 1 **configure**

Step 2 **mpls static**

Example:

```
RP/0/RSP0/CPU0:router(config)# mpls static
```

Enters MPLS-static configuration mode.

Step 3 **vrf** *vrf_name* **address-family ipv4 unicast**

Example:

```
RP/0/RSP0/CPU0:router(config-mpls-static)# vrf vrf1 address-family ipv4 unicast
```

Applies the static configuration to an IPv4 address family in the VRF named *vrf1*.

Step 4
- **local-label** *label-value* **allocate per-prefix** *IPv4_prefix_entry*
- **local-label** *label-value* **allocate per-vrf** forward **path** *path-id* **pop-and-lookup**

Example:

```
RP/0/RSP0/CPU0:router(config-mpls-static-vrf-af)# local-label 30500 allocate per-prefix 100.1.1.0/24
```

The MPLS protocol requests label 30500 to be statically bound as a local label for the prefix 100.1.1.0/24 in the VRF named *vrf1*.

Example:

```
RP/0/RSP0/CPU0:router(config-mpls-static-vrf-af)# local-label 30500 allocate per-vrf forward path 1
 pop-and-lookup
```

The MPLS protocol requests single label 30500 to be statically bound as a local label for all the prefixes in the VRF named *vrf1*. When the router receives packets with VRF label 30500, it removes the label and then performs IP-based lookup to forward the packets.

Step 5 **commit**

Verify MPLS Static Bindings

These are the show commands that can be used to verify MPLS static bindings and LSPs.

SUMMARY STEPS

1. **show mpls static local-label** *label_value*
2. **show mpls label range**
3. **show mpls lsd forwarding**

DETAILED STEPS

Procedure

Step 1 **show mpls static local-label** *label_value*

Example:

```
RP/0/RSP0/CPU0:router#show mpls static local-label 200
Tue Apr 22 18:21:55.764 UTC
Label   VRF             Type         Prefix           RW Configured   Status
------- --------------- ------------ ---------------- --------------- --------
200     default         Per-Prefix   10.10.10.10/32   Yes             Created
```

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

Step 2 **show mpls label range**

Example:

```
RP/0/RSP0/CPU0:router#show mpls label range
Mon Apr 28 19:56:00.596 IST
Range for dynamic labels: Min/Max: 16000/1048575
```

Checks the dynamic range and ensures that the specified local-label value is outside this range.

Step 3 **show mpls lsd forwarding**

Example:

```
RP/0/RSP0/CPU0:router#show mpls lsd forwarding
Tue Apr 29 15:59:52.011 UTC
In_Label, (ID), Path_Info: <Type>
89, (IPv4, 'default':4U, 10.6.2.55/32), 1 Paths
   1/1: IPv4, 'default':4U, Gi0/0/0/21, nh=0.0.0.0, lbl=89, tun_id=0, flags=0x0 ()
110, (IPv4, 'default':4U, 172.16.0.1/32), 1 Paths
   1/1: IPv4, 'default':4U, Gi0/1/0/0, nh=10.12.1.2, lbl=Pop, tun_id=0, flags=0x0 ()
120, (IPv4, 'default':4U, 192.168.0.1/32), 1 Paths
   1/1: IPv4, 'default':4U, Gi0/1/0/0, nh=10.12.1.2, lbl=0, tun_id=0, flags=0x0 ()
130, (IPv4, 'default':4U, 209.165.201.1/32), 1 Paths
   1/1: IPv4, 'default':4U, Gi0/1/0/0, nh=10.12.1.2, lbl=200, tun_id=0, flags=0x0 ()
```

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

Configuring Top Label Hash

Perform this task to configure MPLS top label hash entries.

SUMMARY STEPS

1. **configure**
2. **mpls static**
3. **address-family ipv4 unicast**
4. **top-label-hash**
5. **local-label** *label-value* **allocate**
6. **forward path** *path-count* **nexthop** *interface-type interface-id* **out-label pop**
7. **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters Global Configuration mode.

	Command or Action	Purpose
Step 2	**mpls static** **Example:** `RP/0/RSP0/CPU0:router(config)# mpls static`	Enters MPLS-static configuration mode.
Step 3	**address-family ipv4 unicast** **Example:** `RP/0/RSP0/CPU0:router(config-mpls-static)# address-family ipv4 unicast`	Applies the static configuration to an IPv4 address family.
Step 4	**top-label-hash** **Example:** `RP/0/RSP0/CPU0:router(config-mpls-static-af)# top-label-hash`	Enables top label hash.
Step 5	**local-label** *label-value* **allocate** **Example:** `RP/0/RSP0/CPU0:router(config-mpls-static-af-tlhash)# local-label 25000 allocate`	Specifies the incoming label value. In this example, the label value is 25000.
Step 6	**forward path** *path-count* **nexthop** *interface-type interface-id* **out-label pop** **Example:** `RP/0/RSP0/CPU0:router(config-mpls-static-af-tlhash-lbl)# forward path 1 nexthop bundle-ether 1 out-label pop`	The received label is incremented by one and the label is swapped for the incremented label by the MPLS protocol. For example: For packets that are received with the label 25000, the MPLS protocol swaps labels and applies the label, 25001. After applying the new label, it forwards the packets to the next hop, through the specified interface (Bundle-Ether interface in this case). Sets the output label to 'pop' off the top of the label stack.
Step 7	**commit**	

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 (per VRF) 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.

Complete these steps to identify and clear the discrepancies.

Procedure

Step 1 To identify a label discrepancy, execute one of these:

- **show mpls static local-label discrepancy**
- **show log**

Example:

```
RP/0/RSP0/CPU0:router#show mpls static local-label discrepancy
Tue Apr 22 18:36:31.614 UTC
Label   VRF             Type         Prefix           RW Configured   Status
------- --------------- ------------ ---------------- --------------- --------
16003   default         Per-Prefix   10.0.0.1/32       No              Discrepancy
```

Example:

```
RP/0/RSP0/CPU0:router#show log
Thu Apr 24 14:18:57.655 UTC
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.
```

Step 2 **clear mpls static local-label discrepancy all**

Example:

```
RP/0/RSP0/CPU0:router# clear mpls static local-label discrepancy all
```

Clears label discrepancy 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.

Configure Top Label Hash: Example

This example shows how to configure MPLS top label hash entries:

```
configure
 mpls static
   crossconnect 25000 bundle-ether 1 pop
   address-family ipv4 unicast
    top-label-hash
      local-label 25000 allocate forward path 1 nexthop bundle-ether 1 out-label pop
    !
   !
```