Implementing MPLS Static Labeling

- MPLS Static Labeling , on page 1

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:

- 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.
- Utilize system resources optimally because dynamic labeling is not processed.
- Static labeling on IPv6 packets is supported.

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.
- When paths of different technologies are resolved over ECMP, it results in *heterogeneous* ECMP, leading to severe network traffic issues. Don’t use ECMP for any combination of the following technologies:
  - LDP.
  - BGP-LU, including services over BGP-LU loopback peering or recursive services at Level-3.
  - VPNv4.

- 6PE and 6VPE.
- EVPN.
- Recursive static routing.

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.

```
RP/0/RP0/CPU0:router(config)#mpls label range table 0 17000 18000
RP/0/RP0/CPU0:router(config)#commit

RP/0/RP0/CPU0:router(config)#mpls static

RP/0/RP0/CPU0:router(config-mpls-static)# interface HundredGigE 0/0/0/3
RP/0/RP0/CPU0:router(config-mpls-static)#address-family ipv4 unicast
RP/0/RP0/CPU0:router(config-mpls-static-af)#local-label 24035 allocate

RP/0/RP0/CPU0:router(config-mpls-static-af-lbl)#forward
RP/0/RP0/CPU0:router(config-mpls-static-af-lbl-fwd)#
path 1 nexthop HundredGigE 0/0/0/1 10.2.2.2 out-label 24036
RP/0/RP0/CPU0:router(config-mpls-static-af-lbl-fwd)# commit
```

Verification

Verify the interfaces on which MPLS is enabled

```
RP/0/RP0/CPU0:router# show  mpls interfaces
Mon May 12 06:21:30.937 DST
Interface                  LDP      Tunnel   Static   Enabled
-------------------------- -------- -------- -------- --------
TenGigE0/0/0/5             No       No       Yes      Yes
```

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

```
RP/0/RP0/CPU0:router#show mpls static local-label all
Tue Apr 22 18:21:55.764 UTC
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.

```
RP/0/RP0/CPU0:router#show mpls label range
Mon Apr 28 19:56:00.596 IST
Range for dynamic labels: Min/Max: 17000/18000
```

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

```
RP/0/RP0/CPU0:router#show mpls lsd forwarding
Wed Nov 25 21:40:57.918 UTC
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


RP/0/RP0/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.
```

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. Clearing label discrepancy may result in traffic loss for the dynamic label which got cleared.

```
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 MPLS Static Labels for IPv6 Prefixes

MPLS static LSPs are created by assigning static labels to incoming IPv4/IPv6 prefixes and then specifying the next hop address for the packets with statically assigned labels. Packets are forwarded in the LSP based on this information. Effective with Cisco IOS-XR 6.2.2, you can assign static labels to IPv6 prefixes and set the next hop address in the static LSP as an IPv6 address. To configure static labels for IPv6 prefixes, you first need to configure named static LSPs and perform the remaining steps under LSP configuration mode.

Configuration Example

This example shows how to assign a static label to an IPv6 prefix and also configures an IPv6 address as the next hop.

```
RP/0/0/CPU0:Router# configure terminal
RP/0/0/CPU0:Router(config)# mpls static
RP/0/0/CPU0:Router(config-mpls-static)# lsp ipv6-1
RP/0/0/CPU0:Router(config-mpls-static-lsp)# in-label 25000 allocate per-prefix 2001:DB8:0:1::
 /64
RP/0/0/CPU0:Router(config-mpls-static-lsp)# forward
RP/0/0/CPU0:Router(config-mpls-static-lsp-fwd)# path 1 nexthop TenGigabitEthernet0/0/0/0
2001:DB8:0:5::64  out-label pop
```

Verification

The following examples show how to verify MPLS static labels for IPv6 prefixes.

```
RP/0/0/CPU0:Router# show mpls forwarding

Wed March 29 12:42:41.290 IST
Local  Outgoing    Prefix              Outgoing          Next Hop            Bytes
Label  Label       or ID               Interface                             Switched
------ ----------- ------------------  ------------     ---------------      ------------
25000  Pop         2001:DB8:0:1:: /64  TenGigE0/0/0/0     2001:DB8:0:5::64    0

RP/0/0/CPU0:Router# show mpls static lsp ipv6-1 detail

Wed March 29 12:45:55.574 IST

LSP Name  Label   VRF      AFI  Type       Prefix            RW Configured   Status
--------  ------- -------  ---- ---------- ----------------  --------------- --------
ipv6-1    25000   default  IPv6 Per-Prefix 2001:DB8:0:1:: /64      Yes       Created
    PRIMARY-PATHS:
        PATH 1 :  next-hop TenGigE0/0/0/0  2001:DB8:0:5::64  out-label POP
            STATUS: VALID
```

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 GigabitEthernet0/0/0/0
10.1.0.1 out-label 25000 backup-id 2
RP/0/0/CPU0:Router(config-mpls-static-lsp-fwd)# path 2 nexthop GigabitEthernet0/0/0/1
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 GigabitEthernet0/0/0/5
10.5.0.1 out-label pop backup-id 2
RP/0/0/CPU0:Router(config-mpls-static-lsp-backup)# path 2 nexthop GigabitEthernet0/0/0/6
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 GigabitEthernet0/0/0/0
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 GigabitEthernet0/0/0/1
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 GigabitEthernet0/0/0/5
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 GigabitEthernet0/0/0/6
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 GigabitEthernet0/0/0/0
10.1.0.1 out-label 25000 backup-id 2
RP/0/0/CPU0:Router(config-mpls-static-lsp-fwd)# path 2 nexthop GigabitEthernet0/0/0/1
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 GigabitEthernet0/0/0/5
10.5.0.1 out-label pop backup-id 2
RP/0/0/CPU0:Router(config-mpls-static-lsp-backup)# path 2 nexthop GigabitEthernet0/0/0/6
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.
- In Cisco IOS-XR 6.2.2, only host routes can be resolved. Host routes are routes with /32 mask for IPv4 addresses and /128 mask for IPv6 addresses.

Configuration Example

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

```
RP/0/0/CPU0:Router# configure terminal
RP/0/0/CPU0:Router(config)# mpls static
RP/0/0/CPU0:Router(config-mpls-static)# lsp ipv6-2
RP/0/0/CPU0:Router(config-mpls-static-lsp)# in-label 25000 allocate per-prefix 2001:DB8:0:1::
```

```
 /64  or  24:24:1::/64
RP/0/0/CPU0:Router(config-mpls-static-lsp)# forward
RP/0/0/CPU0:Router(config-mpls-static-lsp-fwd)# path 1 resolve-nexthop 2001:DB8:0:2::64
out-label pop
RP/0/0/CPU0: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.

```
RP/0/0/CPU0:Router# configure terminal
RP/0/0/CPU0:Router(config)# mpls static
RP/0/0/CPU0:Router(config-mpls-static)# lsp anycast_5001
RP/0/0/CPU0:Router(config-mpls-static-lsp)# in-label 5001 allocate
RP/0/0/CPU0:Router(config-mpls-static-lsp)# forward
RP/0/0/CPU0:Router(config-mpls-static-lsp-fwd)# path 1 resolve-nexthop 192.168.2.1 out-label
 pop
RP/0/0/CPU0:Router(config-mpls-static-lsp-fwd)# exit
```

Verification

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

```
RP/0/0/CPU0:Router# show  mpls static lsp anycast_5001 detail
Tue Sep 12 20:00:09.248 UTC
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
```

Configuring MPLS Static over BVI

A Bridge-group virtual interface (BVI) is a routed interface that represents a set of interfaces that gets bridged. By using BVI, you can convert multiple interfaces as members of a common broadcast domain. MPLS static over BVI feature allows you to specify a BVI interface as next hop while setting up a static LSP.

Only static MPLS tunnels can use BVI as a next hop. Also, a BVI next hop can be a static route, a directly connected route (IP address, not a subnet prefix), or a route resolved through BGP or IGP. The router will do an MPLS label lookup on incoming MPLS traffic, perform a label operation such as SWAP/PHP/POP, and forward the MPLS/IP traffic through the BVI next hop. The router can perform switching for Layer 2 traffic and routing for incoming Layer 3 MPLS traffic.

Restrictions

- If a BVI has multiple peers within a subnet, then the subnet prefix cannot be specified as the next hop IP address (though IP addresses within the subnet are BVI peers). You have to specify one of the peers (with a specific IP address) as the next hop.
- Back up paths over BVI (IPv4 or IPv6) are not supported.
- Fast Reroute (FRR) is not supported.
- Dynamic MPLS configuration is not supported. For example, label distribution using LDP is not supported.

Configuration Example

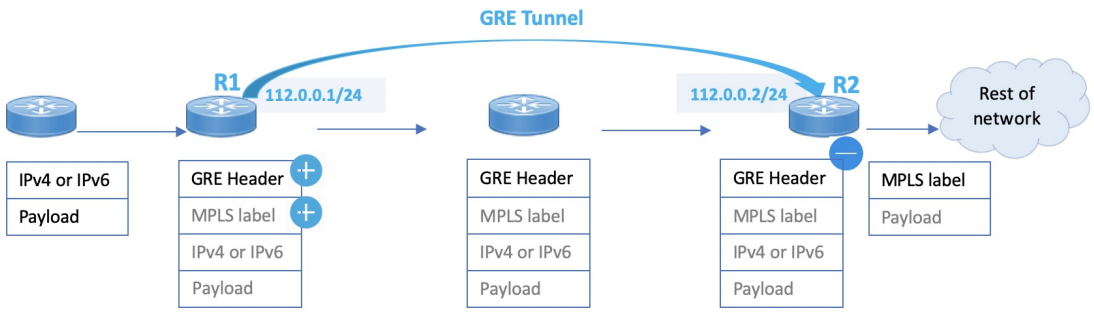
This example shows how to configure a BVI interface as next hop for a static LSP.

```
RP/0/0/CPU0:Router# configure terminal
RP/0/0/CPU0:Router(config)# mpls static
RP/0/0/CPU0:Router(config-mpls-static)# interface TenGig 0/0/0/0
RP/0/0/CPU0:Router(config-mpls-static)# lsp bvi
RP/0/0/CPU0:Router(config-mpls-static-lsp)# in-label 5001 allocate
RP/0/0/CPU0:Router(config-mpls-static-lsp)# forward
RP/0/0/CPU0:Router(config-mpls-static-lsp-fwd)# path 1 nexthop BVI1 192.168.2.1 out-label
pop
RP/0/0/CPU0:Router(config-mpls-static-lsp-fwd)# path 1 nexthop BVI1 192.168.2.1 out-label
4444
RP/0/0/CPU0:Router(config-mpls-static-lsp-fwd)# exit
```

MPLS Over Single-Pass GRE Tunnels

This feature supports MPLS static forwarding over a single-pass GRE tunnel at line rate. One use case is of a Provider router sending incoming customer traffic over the GRE tunnel, addressed to an anycast virtual IP address (VIP) destination shared by a set of load balancing servers.

Figure 1: MPLS Over a Single-Pass GRE Tunnel



In the image, you can see that the GRE tunnel begins at R1. R1 checks ACL configurations, adds an MPLS label to the incoming packet, and then adds a GRE header. Then it sends the traffic towards 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 a single-pass GRE tunnel at line rate.

GRE Tunnel Configuration on R1

The single-pass GRE tunnel starts on R1.

```
R1# configure
R1(config)# interface tunnel-ip1
R1(config-if)# ipv4 address 112.0.0.1 255.255.255.0
R1(config-if)# tunnel mode gre ipv4 encap
R1(config-if)# tunnel source TenGigE0/0/0/2
R1(config-if)# tunnel destination 50.0.0.1
R1(config-if)# commit
```

GRE tunnel destination address is an anycast address. GRE encapsulation must be based on an ACL or a policy-map, or both. A destination can be an individual address or a /28 prefix.

MPLS Static Configuration on R1

```
R1# configure
R1(config)# router static
R1(config-static)# address-family ipv4 unicast
R1(config-static-afi)# 111.0.0.1/32 tunnel-ip1
R1(config-static-afi)# commit

R1(config)# mpls static
R1(config-mpls-static)# lsp test
R1(config-mpls-static-lsp)# in-label 10000 allocate per-prefix 111.0.0.1/32
R1(config-mpls-static-lsp)# forward
R1(config-mpls-static-lsp-fwd)# path 1 nexthop tunnel-ip1 out-label 12000
R1(config-mpls-static-lsp-fwd)# commit
```

GRE Tunnel Configuration on R2

The single-pass GRE tunnel terminates on R2.

```
R2 # configure
R2(config)# interface tunnel-ip1
R2(config-if)# ipv4 address 112.0.0.2 255.255.255.0
R2(config-if)# tunnel mode gre ipv4 decap
R2(config-if)# tunnel source TenGigE0/0/0/2
R2(config-if)# tunnel destination 10.0.0.1
R2(config-if)# commit
```

Verification

Tunnel-IP configuration on R1

```
R1# 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 TenGigE0/0/0/2
    tunnel destination 50.0.0.1
!
```

MPLS Static Configuration on R1

```
R1# show running-config router static

router static
 address-family ipv4 unicast
  111.0.0.1/32 tunnel-ip1
 !
!
```

```
R1# show running-config mpls static

mpls static
  lsp test
     in-label 1000 allocate per-prefix 111.0.0.1/32
     forward
        path 1 nexthop tunnel-ip1 out-label 12000
        !
```

Tunnel-IP Configuration on R2

```
R2# show running-config int tunnel-ip 1

interface tunnel-ip1
 ipv4 address 112.0.0.2 255.255.255.0
 tunnel mode gre ipv4
 tunnel source TenGigE0/0/0/2
 tunnel destination 10.0.0.1
!
```

PBR Configuration for GRE Tunnel Decapsulation on R2

```
R2# 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
!
```