MPLS Traffic Engineering Path Link and Node Protection Configuration Guide, Cisco IOS XE 17 (NCS 4200 Series)

First Published: 2020-07-30

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

CONTENTS

CHAPTER 1

Feature History

The following table lists the new and modified features supported in the MPLS Traffic Engineering Path Link and Node Protection Configuration Guide in Cisco IOS XE 17 releases.

Feature	Description
Cisco IOS XE Amsterdam 17.3.1	
Static PW over P2MP	The Static Pseudowires over Point-to-Multipoint Traffic Engineering (P2MP TE) feature emulates the essential attributes of a unidirectional P2MP service. It can be used to transport layer 2 multicast services from a single source to one or more destinations. This feature is supported on the Cisco RSP2 module. This feature is supported on the Cisco RSP3 module.

The following table lists the new and modified features supported in the MPLS Traffic Engineering Path Link and Node Protection Configuration Guide in Cisco IOS XE 17 releases, on Cisco NCS 4206 and Cisco NCS 4216 routers.

Feature	Description
Cisco IOS XE Amsterdam 17.3.1	
Static PW over P2MP	The Static Pseudowires over Point-to-Multipoint Traffic Engineering (P2MP TE) feature emulates the essential attributes of a unidirectional P2MP service. It can be used to transport layer 2 multicast services from a single source to one or more destinations. This feature is supported on the Cisco RSP2 module.

CHAPTER 2

MPLS Traffic Engineering Nonstop Routing Support

Note This technology is not applicable for the Cisco ASR 900 RSP3 Module.

The MPLS Traffic Engineering Nonstop Routing Support feature assists the Multiprotocol Label Switching (MPLS) Traffic Engineering (TE) routing devices to recover from an interruption in service. This feature also defines the checkpoint and recovery scheme for the devices.

Prerequisites for MPLS Traffic Engineering Nonstop Routing Support

Your network must support the following Cisco features before you enable Multiprotocol Label Switching (MPLS) Traffic Engineering (TE):

- MPLS
- Cisco Express Forwarding
- Intermediate System-to-Intermediate System (IS-IS) or Open Shortest Path First (OSPF)

Before enabling MPLS TE Nonstop Routing (NSR), a full-mode check needs to be done by the system to verify if the **mpls traffic-eng nsr** command is permitted or is restricted due to conflicts or user privileges.

Restrictions for MPLS Traffic Engineering Nonstop Routing Support

Multiprotocol Label Switching (MPLS) Traffic Engineering (TE) Nonstop Routing (NSR) and Resource Reservation Protocol (RSVP) Graceful Restart (GR) are both mutually exclusive recovery mechanisms. Hence, MPLS TE NSR cannot be enabled when RSVP GR is enabled.

Information About MPLS Traffic Engineering Nonstop Routing Support

MPLS Traffic Engineering Nonstop Routing Support Overview

Multiprotocol Label Switching (MPLS) Traffic Engineering (TE) Nonstop Routing (NSR) enables routing devices to recover from an interruption in service. The MPLS TE NSR functionality defines a checkpoint for the control plane of the routing devices. Resource Reservation Protocol (RSVP) Graceful Restart (GR) is another method for recovering and restarting interrupted services.

To avoid conflict and guarantee deterministic behavior, only one of the above mentioned recovery methods can be configured at a given time.

The MPLS TE NSR feature differs from the RSVP GR feature in the following ways:

- MPLS TE NSR devices are fully independent and do not rely on neighbor nodes for a stateful switchover (SSO) recovery.
- MPLS TE NSR supports the SSO recovery of Fast Reroute (FRR) active tunnels.
- MPLS TE NSR has an active standby mode. This helps with most of the recovery states being created before the SSO recovery actually happens, ensuring a faster recovery after SSO.
- MPLS TE NSR **show** commands display recovery information in standby mode as well.
- Label switched paths (LSPs) which are not fully signaled, do not resume signaling after an interruption and will go down on SSO.

How to Configure MPLS Traffic Engineering Nonstop Routing Support

Configuring MPLS Traffic Engineering Nonstop Routing Support

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip cef** **Example:** `Device(config)# ip cef`	Enables standard Cisco Express Forwarding operations.
Step 4	**mpls traffic-eng nsr** **Example:** `Device(config)# mpls traffic-eng nsr`	Enables the MPLS Traffic Engineering (TE) Non-Stop Routing (NSR) functionality on a device. **Note** Enabling the MPLS TE NSR functionality automatically enables the Resource Reservation Protocol (RSVP) NSR functionality as well.
Step 5	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Verifying MPLS Traffic Engineering Nonstop Routing Support

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** Device> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show mpls traffic-eng nsr** **Example:** Device# show mpls traffic-eng nsr counters TE NSR counters database TE NSR check pointed data oos TE NSR out of sync database summary TE NSR summary | Output modifiers <cr>	Displays options to obtain Multiprotocol Label Switching (MPLS) Traffic Engineering (TE) Nonstop Routing (NSR) configuration information such as the database status, counter numbers, devices which are out of sync, and the summary of all the devices.
Step 3	**show mpls traffic-eng nsr counters** **Example:** Device# show mpls traffic-eng nsr counters	Displays information about the data structures or states that are successfully created or removed, along with errors counts.
Step 4	**show mpls traffic-eng nsr database** **Example:** Device# show mpls traffic-eng nsr database	Displays information pertaining to the write and read databases supporting MPLS TE NSR. The write and read databases store the data that is used for recovering TE state on a standby device after stateful switchover (SSO).
Step 5	**show mpls traffic-eng nsr oos** **Example:** Device# show mpls traffic-eng nsr oos	Displays information pertaining to the out of sync databases supporting MPLS TE NSR. The out of sync databases indicate the devices whose states are not in sync with each other.
Step 6	**show mpls traffic-eng nsr summary** **Example:** Device# show mpls traffic-eng nsr summary	Displays a summary of MPLS TE NSR information such as the current TE NSR state (standby-hot / recovering / staling / active), recovery time, and the recovery result (success / failure).
Step 7	**end** **Example:** Device(config)# end	Exits privileged EXEC mode.

Configuration Examples for MPLS Traffic Engineering Nonstop Routing Support

Example: Configuring MPLS Traffic Engineering Nonstop Routing Support

The following example shows how to configure Multiprotocol (MPLS) Traffic Engineering (TE) Nonstop Routing (NSR) support on a device:

```
enable
 configure terminal
  ip cef
  mpls traffic-eng nsr
  end
```

Example: Verifying MPLS Traffic Engineering Nonstop Routing Support

Displaying MPLS Traffic Engineering Nonstop Routing Support Verification Options

The following example shows how to display the options that help you verify Multiprotocol Label Switching (MPLS) Traffic Engineering (TE) Nonstop Routing (NSR) information:

```
enable
 show mpls traffic-eng nsr ?
  counters  TE NSR counters
  database  TE NSR check pointed data
  oos       TE NSR out of sync database
  summary   TE NSR summary
  |         Output modifiers
  <cr>
```

Verifying MPLS Traffic Engineering Nonstop Routing Support Counters

The following example shows how to verify information about the data structures or states that are successfully created or removed, along with errors counts:

```
enable
 show mpls traffic-eng nsr counters

State: Active

Bulk sync
  Last bulk sync was successful (entries sent: 24)
  initiated: 1

Send timer
  started: 7

Checkpoint Messages (Items) Sent
```

```
  Succeeded:      13  (101)
    Acks accepted:13  (101)
    Acks ignored:     (0)
    Nacks:        0  (0)
  Failed:         0  (0)
  Buffer alloc:   13
  Buffer freed:   13

ISSU:
  Checkpoint Messages Transformed:
    On Send:
      Succeeded:         13
      Failed:            0
      Transformations:   0
    On Recv:
      Succeeded:         0
      Failed:            0
      Transformations:   0

  Negotiation:
    Started:              1
    Finished:             1
    Failed to Start:      0
    Messages:
      Sent:
        Send succeeded:   5
        Send failed:      0
        Buffer allocated:        5
        Buffer freed:            0
        Buffer alloc failed:     0
      Received:
        Succeeded:        7
        Failed:           0
        Buffer freed:     7
  Init:
    Succeeded:            1
    Failed:               0

  Session Registration:
    Succeeded:            0
    Failed:               0

  Session Unregistration:
    Succeeded:            0
    Failed:               0

Errors:
  None
```

Verifying MPLS Traffic Engineering Nonstop Routing Support Databases

The following example shows how to verify information pertaining to the write and read databases supporting MPLS TE NSR. The write and read databases store the data that is used for recovering TE state on a standby device after Stateful Switchover (SSO):

```
Device# show mpls traffic-eng nsr database if-autotun
IF_AUTOTUN WRITE DB

  Header:
    State: Checkpointed     Action: Add
```

```
    Seq #: 14               Flags: 0x0
  Data:
    te_nsr_seq_num: 28
    Tunnel ID: 100 (if_handle: 25), prot_if_handle: 3
    template_unit: n/a, dest: 10.2.0.1, flags=0x0

IF_AUTOTUN READ DB

Device# show mpls traffic-eng nsr database lsp-ac ?
  |  Output modifiers
  <cr>

Device# show mpls traffic-eng nsr database lsp-ac
 LM Tunnel WRITE DB:

Tun ID: 1   LSP ID: 11   (P2MP)
  SubGrp ID:   1
  SubGrp Orig: 10.1.0.1
  Dest:   10.2.0.1
  Sender: 10.1.0.1     Ext. Tun ID: 10.1.0.1
  Header:
    State: Checkpointed     Action: Add
    Seq #: 7                Flags: 0x0
    TE NSR Seq #: 14

LM Tunnel READ DB:

Device# show mpls traffic-eng nsr database internal

Write DB:
                    Checkpointed
   Entry Type       or Ack-Pending  Send-Pending
     PCALC Node                  0             0
     PCALC Link                  0             0
     PCALC Auto-Mes              0             0
     PCALC SRLG                  0             0
     lm_tunnel_t                 0             0
     NSR LSP FRR                 0             0
     nsr_if_autotun              0             0
     nsr_tspvif_set              0             0
     nsr_slsp_head               0             0

Read DB:
   Entry Type        Checkpointed
     PCALC Node                 5
     PCALC Link                12
     PCALC Auto-Mesh            0
     PCALC SRLG                 0
     lm_tunnel_t                5
     NSR LSP FRR                0
     nsr_if_autotun             0
     nsr_tspvif_setup           3
     nsr_slsp_head              5

TE NSR Sequence Bulk Sync List:
Entries: 0; next avail seq num: 132

TE NSR Sequence State Creation List:
Entries: 30; next expected seq num: 132
   Seq Num: 7  EntryPtr: 0x5A03B208
     Type: PCALC Node  Action: Add  Bundle Seq #: 1
   Seq Num: 8  EntryPtr: 0x5A0B8B38
     Type: PCALC Link  Action: Add  Bundle Seq #: 2
   Seq Num: 9  EntryPtr: 0x5A0B8DA0
```

```
      Type: PCALC Link  Action: Add  Bundle Seq #: 2
    Seq Num: 10  EntryPtr: 0x59FF1BB0
      Type: PCALC Node  Action: Add  Bundle Seq #: 1
    Seq Num: 11  EntryPtr: 0x5A0B9008
      Type: PCALC Link  Action: Add  Bundle Seq #: 2
    Seq Num: 32  EntryPtr: 0x586F2A50
      Type: PCALC Node  Action: Add  Bundle Seq #: 4
    Seq Num: 33  EntryPtr: 0x5949FC58
      Type: PCALC Link  Action: Add  Bundle Seq #: 5
    Seq Num: 34  EntryPtr: 0x5949FEC0
      Type: PCALC Link  Action: Add  Bundle Seq #: 5
    Seq Num: 60  EntryPtr: 0x5725BC30
      Type: lm_tunnel_t  Action: Add  Bundle Seq #: 12
    Seq Num: 61  EntryPtr: 0x5725BE00
      Type: nsr_tspvif_setup  Action: Add  Bundle Seq #: 12
    Seq Num: 62  EntryPtr: 0x59FC9E80
      Type: nsr_slsp_head  Action: Add  Bundle Seq #: 12
    Seq Num: 79  EntryPtr: 0x59296190
      Type: lm_tunnel_t  Action: Add  Bundle Seq #: 16
    Seq Num: 80  EntryPtr: 0x59296360
      Type: nsr_tspvif_setup  Action: Add  Bundle Seq #: 16
    Seq Num: 81  EntryPtr: 0x571EB7F8
      Type: nsr_slsp_head  Action: Add  Bundle Seq #: 16
    Seq Num: 98  EntryPtr: 0x5A04B770
      Type: lm_tunnel_t  Action: Add  Bundle Seq #: 20
    Seq Num: 99  EntryPtr: 0x59296108
      Type: nsr_tspvif_setup  Action: Add  Bundle Seq #: 20
    Seq Num: 100  EntryPtr: 0x57258670
      Type: nsr_slsp_head  Action: Add  Bundle Seq #: 20
    Seq Num: 101  EntryPtr: 0x5A060348
      Type: lm_tunnel_t  Action: Add  Bundle Seq #: 20
    Seq Num: 102  EntryPtr: 0x5A03B2B0
      Type: nsr_slsp_head  Action: Add  Bundle Seq #: 20
    Seq Num: 103  EntryPtr: 0x5B1F12B0
      Type: lm_tunnel_t  Action: Add  Bundle Seq #: 20
    Seq Num: 104  EntryPtr: 0x5A03B400
      Type: nsr_slsp_head  Action: Add  Bundle Seq #: 20
    Seq Num: 121  EntryPtr: 0x57258358
      Type: PCALC Node  Action: Add  Bundle Seq #: 21
    Seq Num: 122  EntryPtr: 0x59FAF080
      Type: PCALC Link  Action: Add  Bundle Seq #: 22
    Seq Num: 123  EntryPtr: 0x59502AC0
      Type: PCALC Link  Action: Add  Bundle Seq #: 23
    Seq Num: 124  EntryPtr: 0x594AE918
      Type: PCALC Link  Action: Add  Bundle Seq #: 21
    Seq Num: 125  EntryPtr: 0x59502120
      Type: PCALC Link  Action: Add  Bundle Seq #: 23
    Seq Num: 126  EntryPtr: 0x59FAFA20
      Type: PCALC Link  Action: Add  Bundle Seq #: 22
    Seq Num: 129  EntryPtr: 0x59FC9CC0
      Type: PCALC Node  Action: Add  Bundle Seq #: 24
    Seq Num: 130  EntryPtr: 0x5A060518
      Type: PCALC Link  Action: Add  Bundle Seq #: 24
    Seq Num: 131  EntryPtr: 0x59FAFC88
      Type: PCALC Link  Action: Add  Bundle Seq #: 24

Device# show mpls traffic-eng nsr database lsp-frr
LSP-FRR WRITE DB

Tun ID: 1   LSP ID: 10   (P2MP)
  SubGrp ID:   1
  SubGrp Orig: 10.1.0.1
  Dest:   10.2.0.1
  Sender: 10.1.0.1     Ext. Tun ID: 10.1.0.1
```

```
  Header:
    State: Checkpointed     Action: Add
    Seq #: 45               Flags: 0x0
  Data:
    te_nsr_seq_num: 164
    LSP Protected if_num: 3 (Ethernet0/0)
    LSP Next-Hop Info: rrr_id 10.2.0.1, address 10.2.0.1, label 17
    LSP Next-Next-Hop Info: rrr_id 0.0.0.0, address 0.0.0.0, label 16777216
    LSP Hold Priority: 7
    LSP bw_type: any pool
    LSP desired_bit_type: 0x0n    LSP Backup ERO address 10.1.2.2
    LSP advertise_bw: NO

LSP-FRR READ DB

Device# show mpls traffic-eng nsr database lsp-frr filter destination ?
  Hostname or A.B.C.D  IP addr or name of destination (tunnel tail)

Device# show mpls traffic-eng nsr database lsp-frr filter lsp-id ?
  <0-65535>  LSP ID

Device# show mpls traffic-eng nsr database lsp-frr filter source ?
  Hostname or A.B.C.D  IP addr or name of sender (tunnel head)

Device# show mpls traffic-eng nsr database lsp-frr filter tunnel-id ?
  <0-65535>  tunnel ID

Device# show mpls traffic-eng nsr database lsp-head
SLSP_HEAD WRITE DB

  Tun ID: 0  (P2P), lsp_id: 7
  Header:
    State: Checkpointed     Action: Add
    Seq #: 6                Flags: 0x0
  Data:
    te_nsr_seq_num: 18
    bandwidth: 5, thead_flags: 0x1, popt: 1
    feature flags: none
    output_if_num: 11, output_nhop: 10.1.3.2
    backup_output_if_num: 0
    output_tag: 19
    backup_output_tag: 16777218
    RRR path setup info
      Destination: 10.3.0.1, Id: 10.3.0.1 Router Node (ospf) flag:0x0
      IGP: ospf, IGP area: 0, Number of hops: 3, metric: 128
      Hop 0: 10.1.3.2, Id: 10.2.0.1 Router Node (ospf), flag:0x0
      Hop 1: 10.2.3.3, Id: 10.3.0.1 Router Node (ospf), flag:0x0
      Hop 2: 10.3.0.1, Id: 10.3.0.1 Router Node (ospf), flag:0x0

SLSP_HEAD READ DB

Device# show mpls traffic-eng nsr database lsp-head filter destination ?
  Hostname or A.B.C.D  IP addr or name of destination (tunnel tail)

Device# show mpls traffic-eng nsr database lsp-head filter lsp-id ?
  <0-65535>  LSP ID

Device# show mpls traffic-eng nsr database lsp-head filter source ?
  Hostname or A.B.C.D  IP addr or name of sender (tunnel head)

Device# show mpls traffic-eng nsr database lsp-head filter tunnel-id ?
  <0-65535>  tunnel ID

Device# show mpls traffic-eng nsr database pcalc auto-mesh
```

```
  PCALC Auto-Mesh WRITE DB:

  PCALC Auto-Mesh READ DB:

Device# show mpls traffic-eng nsr database pcalc nbr
 PCALC Link WRITE DB:
  Header:
    State: Checkpointed     Action: Add
    Seq #: 4                Flags: 0x0
    TE NSR Seq #: 26
    IGP Id:10.1.2.2     Area:0   Nbr IGP Id:10.1.2.2
    IP:10.1.2.1         Nbr IP:0.0.0.0  Framgment ID:1
    Intf ID    Local:0     Remote:0


 PCALC Link READ DB:

Device# show mpls traffic-eng nsr database pcalc node
 PCALC Node WRITE DB:
  Header:
    State: Checkpointed     Action: Add
    Seq #: 4                Flags: 0x0
    TE NSR Seq #: 25
    Router Id 10.1.0.1
    node_id 1
    num_links 2
    tlvs_len 0
    flags 0x6
    rid_frag_id 0
    bcid_mismatch 0
    incarnation  0

Device# show mpls traffic-eng nsr database pcalc srlg
 PCALC SRLGs WRITE DB:

 PCALC SRLGs READ DB:

Device# show mpls traffic-eng nsr database summary
MPLS-TE Non-Stop-Routing is ENABLED

Write DB Coalescing: INACTIVE
Write DB:
  Send-Pending:     0
  Ack-Pending :     0
  Checkpointed:    35
  Total       :    35

Read DB:
  Total       :     0

Device# show mpls traffic-eng nsr database tun-setup
TSPVIF_SETUP WRITE DB

  Tun ID: 0, lsp_id: 7
  Header:
    State: Checkpointed     Action: Add
    Seq #: 6                Flags: 0x0
  Data:
    te_nsr_seq_num: 17
    Setup Evt: allocating current tspsetup, chkpt_flags: 0x0

TSPVIF_SETUP READ DB
```

Verifying MPLS Traffic Engineering Nonstop Routing Support Out-of-Sync Databases

The following example shows how to verify information pertaining to the out-of-sync databases supporting MPLS TE NSR. The out-of-sync databases indicate the **active and standby RSP** whose states are not in sync with each other:

```
enable
 show mpls traffic-eng nsr oos
  Tunnel: 4000
  Time created: 02/20/13-12:03:13
  Time synced: 02/20/13-12:03:14
  Key:
    Source:                    10.1.0.1
    Destination:               10.2.0.1
    ID:                        4000
    Ext Tun ID:                10.1.0.1
    Instance:                  4
    Slsp p2mp ID:              0
    Slsp p2mp subgroup ID:     0
    Slsp p2mp subgroup origin: 0

  RSVP States:
    Signal:       Unknown
    Fast-Reroute: Disabled
    Delete State: True

  TE States:
    Signal:       Unknown
    Fast-Reroute: Disabled
    Delete State: True

  Update History:
    Total number of updates: 2

      Update Time: 02/20/13-12:03:13
        Client Updating: RSVP
        Update State:
          Signal:       Unknown
          Fast-Reroute: Unknown
          Delete State: True

      Update Time: 02/20/13-12:03:14
        Client Updating: TE
        Update State:
          Signal:       Unknown
          Fast-Reroute: Unknown
          Delete State: True
```

Verifying MPLS Traffic Engineering Nonstop Routing Support Information Summary

The following example shows how to view a summary of MPLS TE NSR information such as the current TE NSR state (standby-hot / recovering / staling / active), recovery time, and the recovery result (success / failure):

```
enable
 show mpls traffic-eng nsr summary
  State:
```

```
Graceful-Restart: Disabled
HA state: Active
Checkpointing: Allowed
Messages:
 Send timer: not running (Interval: 1000 msec)
 Items sent per Interval: 200
 CF buffer size used: 3968
```

Additional References for MPLS Traffic Engineering Nonstop Routing Support

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Multiprotocol Label Switching High Availability Configuration Guide	Cisco IOS XE Multiprotocol Label Switching High Availability Configuration Guide
MPLS TE commands	Cisco IOS Multiprotocol Label Switching Command Reference

Standards and RFCs

Standard/RFC	Title
RFC 2205	*Resource Reservation Protocol (RSVP)*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for MPLS Traffic Engineering Nonstop Routing Support

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 1: Feature Information for MPLS Traffic Engineering Nonstop Routing Support

Feature Name	Releases	Feature Information
MPLS Traffic Engineering Nonstop Routing Support	Cisco IOS XE Release 3.10S, 3.13S	The MPLS Traffic Engineering Non-Stop Routing Support feature assists the Multiprotocol Label Switching (MPLS) Traffic Engineering (TE) routing devices to recover from an interruption in service. The MPLS TE Nonstop Routing (NSR) support functionality also defines the checkpoint and recovery scheme for the devices. From Cisco IOS XE 3.13S, support was provided for ASR 903. The following commands were introduced: **mpls traffic-eng nsr** and **show mpls traffic-eng nsr**.

CHAPTER 3

MPLS Traffic Engineering over Bridge Domain Interfaces

The MPLS Traffic Engineering(TE) over Bridge Domain Interfaces(BDI) feature enables MPLS traffic engineering over Bridge Domain Interfaces.

Prerequisites for Configuring MPLS TE over BDI

You must have:

- Enabled MPLS TE on all relevant routers and interfaces
- Configured MPLS TE tunnels

Your network must support the following Cisco IOS features:

- IP Cisco Express Forwarding
- Multiprotocol Label Switching (MPLS)

Your network must support at least one of the following protocols:

- Intermediate SystemtoIntermediate System (ISIS)
- Open Shortest Path First (OSPF)

Restrictions for MPLS TE over BDI

- MPLS TE - Verbatim Path Support
- Explicit Path Node exclusion
- P2MP TE Tunnels
- Auto-tunnel one-hops and backups

- Auto bandwidth
- Inter area or AS TE
- Auto route destinations
- FRR link ornode protection

Information About MPLS Traffic Engineering over BDI

Features of MPLS Traffic Engineering over BDI

The MPLS Traffic Engineering over BDI feature enables MPLS TE tunnels over BDI.

Supported Features

Your network must support the following:

- MPLS TE tunnels
- Policy Routing onto MPLS TE Tunnels
- MPLS TE - Forwarding Adjacency
- MPLS TE – RSVP Hello State Timer
- MPLS TE - LSP Attributes
- MPLS TE - IP Explicit Address Exclusion
- MPLS TE - Configurable Path Calculation Metric for Tunnels
- MPLS TE - Verbatim Path Support
- Pseudo-wire mapping onto TE tunnels.

How to Configure MPLS Traffic Engineering over BDI

This section assumes that you want to configure MPLS TE over BDI.

Configuring MPLS TE over BDI

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Router> enable`	
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters interface configuration mode.
Step 3	**interface bdi30** **Example:** `Router(config)# interface bdi30`	Specifies the bridge domain interface and enters interface configuration mode.
Step 4	**mpls traffic-eng tunnels** **Example:** `Router(config-if)# mpls traffic-eng tunnels`	Enables an MPLS TE tunnel to use an established tunnel for the bridge domain interface.
Step 5	**end** **Example:** `Router(config-if)# end`	Returns to privileged EXEC mode.

Configuring the RSVP Bandwidth

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type slot* / *subslot* / *port* **Example:** `Router(config)# interface gigabitEthernet 0/0/0`	Configures the interface type and enters interface configuration mode.

	Command or Action	Purpose
Step 4	**ip rsvp bandwidth** [*interface-kbps* [*single-flow-kbps*[**bc1** *kbps* | **sub-pool** *kbps*]*]*| **percent** *percent-bandwidth* [*single-flow-kbps*]] **Example:** `Router(config-if)# ip rsvp bandwidth 7500 7500`	Enables RSVP on an interface. • The optional *interface-kbps* and *single-flow-kbps* arguments specify the amount of bandwidth that can be allocated by RSVP flows or to a single flow, respectively. Values are from 1 to 10000000. • The optional **sub-pool**and *kbps*keyword and argument specify subpool traffic and the amount of bandwidth that can be allocated by RSVP flows. Values are from 1 to 10000000. **Note** Repeat this command for each interface on which you want to enable RSVP.
Step 5	**end** **Example:** `Router(config-if)# end`	(Optional) Returns to privileged EXEC mode.

Verifying That MPLS TE over BDI Is Operational

To verify that MPLS TE over BDI can function, perform the following task.

Procedure

Step 1 **enable**

Enables privileged EXEC mode.

Step 2 **show mpls traffic-eng tunnels brief**

Use this command to monitor and verify the state of the tunnels.

Step 3 **show mpls traffic-eng tunnels summary**

Use this command to monitor and verify the state of the tunnels.

Step 4 **show mpls traffic-eng tunnels tunnel1**

Use this command to verify that tunnels are up and using BDI.

Troubleshooting Tips

This section describes how you can use the `show mpls traffic-eng tunnels tunnel5` to check for issues.

```
Router# show mpls traffic-eng tunnels tunnel5

Name: router_t5                            (Tunnel5) Destination: 10.0.0.3
  Status:
    Admin: up         Oper: up     Path: valid       Signalling: connected
    path option 1, type dynamic (Basis for Setup, path weight 2)

  Config Parameters:
    Bandwidth: 0        kbps (Global)  Priority: 5  5   Affinity: 0x0/0xFFFF
    Metric Type: TE (default)
    AutoRoute: disabled LockDown: disabled Loadshare: 0 [0] bw-based
    auto-bw: disabled
  Active Path Option Parameters:
    State: dynamic path option 1 is active
    BandwidthOverride: disabled  LockDown: disabled  Verbatim: disabled


  InLabel  :  -
  OutLabel : BDI31, 21
  Next Hop : 12.0.0.2
  RSVP Signalling Info:
       Src 10.0.0.1, Dst 10.0.0.3, Tun_Id 5, Tun_Instance 1
    RSVP Path Info:
      My Address: 12.0.0.1
      Explicit Route: 12.0.0.2 14.0.0.2 14.0.0.1 10.0.0.3
      Record   Route:   NONE
      Tspec: ave rate=0 kbits, burst=1000 bytes, peak rate=0 kbits
    RSVP Resv Info:
      Record   Route:   NONE
      Fspec: ave rate=0 kbits, burst=1000 bytes, peak rate=0 kbits
  Shortest Unconstrained Path Info:
    Path Weight: 2 (TE)
    Explicit Route: 12.0.0.1 12.0.0.2 14.0.0.2 14.0.0.1
                    10.0.0.3
  History:
    Tunnel:
      Time since created: 1 minutes, 38 seconds
      Time since path change: 1 minutes, 36 seconds
      Number of LSP IDs (Tun_Instances) used: 1
    Current LSP: [ID: 1]
      Uptime: 1 minutes, 36 seconds
```

Configuration Example for MPLS Traffic Engineering over BDI

The following example enables the BDI on the router:

```
Router(config)#interface bdi30
Router(config-if)#mpls traffic-eng tunnels
```

Configuring Interface Tunnel Example

The following example configures an interface tunnel

```
interface Tunnel1
ip unnumbered Loopback0
```

```
tunnel source Loopback0
tunnel mode mpls traffic-eng
tunnel destination 10.0.0.4
tunnel mpls traffic-eng path-option 1 dynamic
```

Configuring RSVP Bandwidth Example

The following example configures RSVP bandwidth

ip rsvp bandwidth [*interface-kbps*] [*single-flow-kbps*]

```
Router(config-if)# ip rsvp bandwidth 500 500
```

CHAPTER 4

MPLS Point-to-Multipoint Traffic Engineering Support for Static Pseudowires

Table 2: Feature History

Feature Name	Release	Description
Static PW over P2MP	Cisco IOS XE Amsterdam 17.3.1	The Static Pseudowires over Point-to-Multipoint Traffic Engineering (P2MP TE) feature emulates the essential attributes of a unidirectional P2MP service. It can be used to transport layer 2 multicast services from a single source to one or more destinations. This feature is supported on the Cisco RSP2 module. This feature is supported on the Cisco RSP3 module.

The MPLS Point-to-Multipoint Traffic Engineering: Support for Static Pseudowires feature allows you to configure a point-to-multipoint pseudowire (PW) to transport Layer 2 traffic from a single source to one or more destinations. This feature provides traffic segmentation for Multiprotocol Label Switching (MPLS) Point-to-Multipoint Traffic Engineering (P2MP TE) tunnels.

The MPLS Point-to-Multipoint Traffic Engineering: Support for Static Pseudowires feature uses Layer 2 Virtual Private Network (L2VPN) static PWs to provide point-to-multipoint Layer 2 connectivity over an MPLS network to transport Layer 2 traffic. The static PW does not need Label Distribution Protocol (LDP).

Prerequisites for MPLS Point-to-Multipoint Traffic Engineering Support for Static Pseudowires

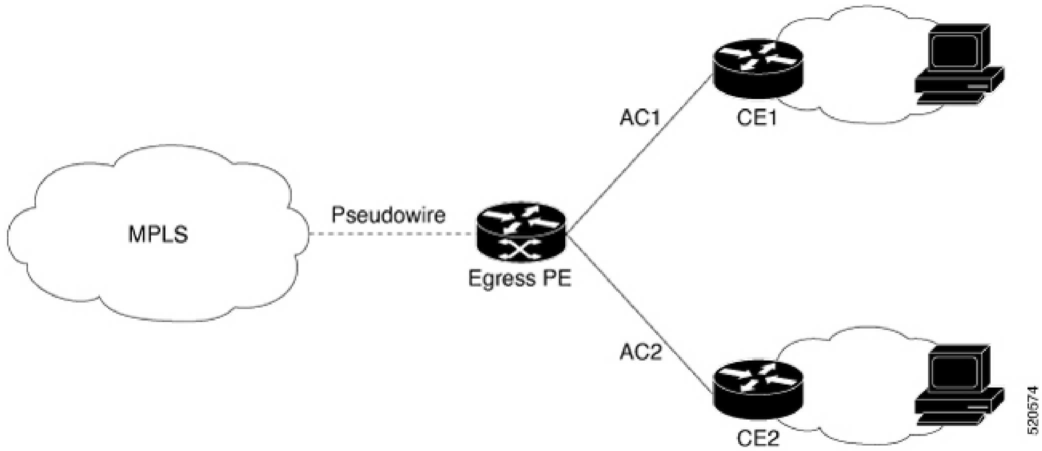
Before configuring the MPLS Point-to-Multipoint Traffic Engineering: Support for Static Pseudowires feature, ensure that the following prerequisite is met:

- If a Cisco RSP3 module acts as a P2MP TE midpoint, it should be running the Cisco IOS XE Release 17.3.1 or later releases.

Restrictions for MPLS Point-to-Multipoint Traffic Engineering Support for Static Pseudowires

- Only EVC-based Ethernet over MPLS is supported. TDM MPLS is **not** supported.
- Multiple Xconnects cannot be configured with same P2MP Tunnel as it leads to traffic drop for one of the Connects.
- If the preferred-paths under pseudowire-class of the Xconnects are swapped, Xconnect interface should be flapped to resume traffic.
- P2MP Tunnel cannot be used to forward Static PW traffic and Global IPv4 multicast traffic (MVPN profile 8) simultaneously.
- Static PW over P2MP is standardized as unidirectional. But the current configuration model does not block packet forwarding from the receiver to the source.
- Local bindings must be unique. Otherwise traffic will accidentally merge.
- Replication of egress is not supported. Only a single CE connects to a PE which is part of the Tunnel destination list of one P2MP Pseudowire.
- Effective Cisco IOS XE 17.3.1, the Static PW over Point-to-Multipoint tunnel can be scaled up to 400 tunnels and 400 Static PWs.

Figure 1: Egress Packet Replication



Note You must use the **no show ip rsvp** command to check tunnel bandwidth. If the total tunnel bandwidth exceeds beyond 750 MB (megabits per second), then the sub-LSPs go down when toggling the **traffic-eng** command with a maximum tunnel bandwidth of 749.9 MB.

For example, there are 82 P2MP tunnels and you configure 9146 kbps for each tunnel. Then the total bandwidth allocated is (9146 kbps * 82 tunnels) = 749.9 MB.

Information About MPLS Point-to-Multipoint Traffic Engineering Support for Static Pseudowires

Overview of MPLS Point-to-Multipoint Traffic Engineering Support for Static Pseudowires

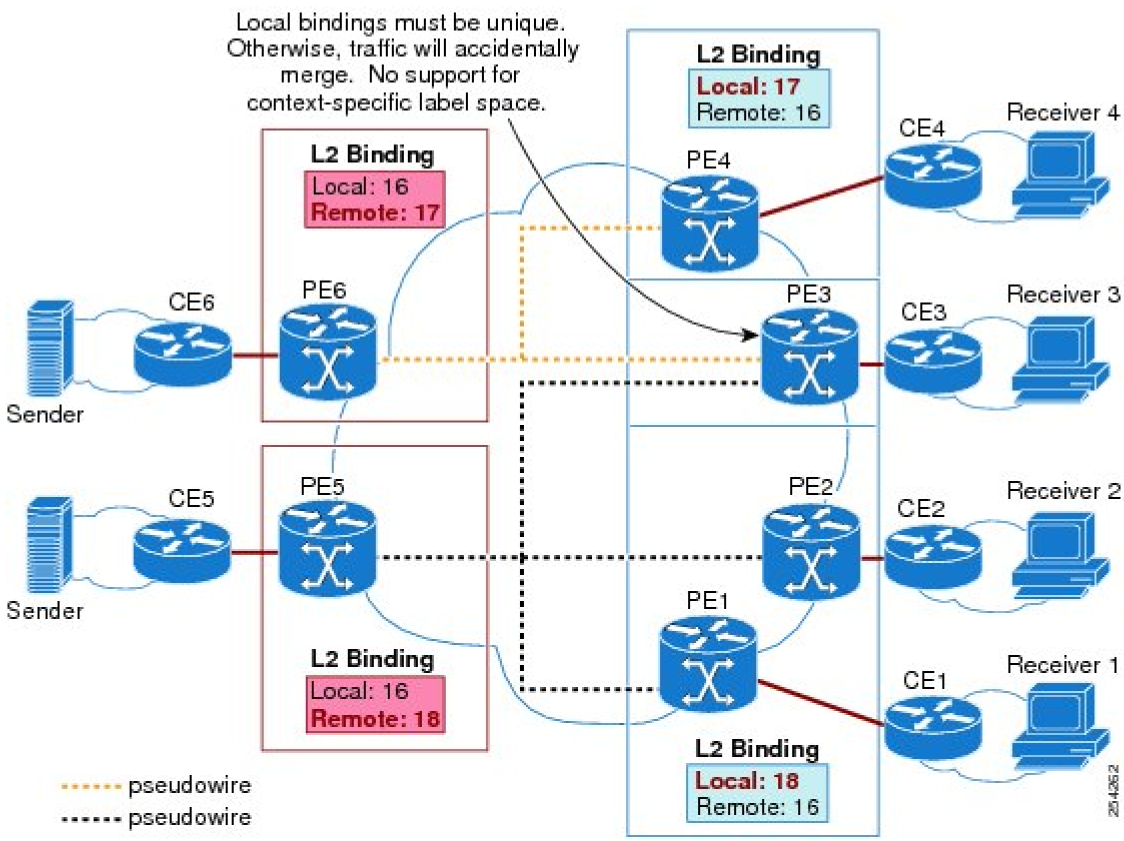
The MPLS Point-to-Multipoint Traffic Engineering: Support for Static Pseudowires feature transports Layer 2 traffic from a single source to one or more destinations. This feature has the following characteristics:

- It uses L2VPN static PWs to provide point-to-multipoint Layer 2 connectivity over an MPLS network to transport Layer 2 traffic.
- The segmentation for MPLS P2MP TE tunnels provided by this feature allows for applications such as video distribution and clock distribution (mobile backhaul).
- This feature is compatible with Cisco nonstop forwarding (NSF), stateful switchover (SSO). See NSF/SSO—MPLS TE and RSVP Graceful Restart and MPLS Point-to-Multipoint Traffic Engineering for information on configuring NSF/SSO with this feature.
- In this implementation, the PW is bidirectional, in accordance with the Framework and Requirements for Virtual Private Multicast Service .

VC Label Collisions

This feature does not support context-specific label spaces. When configuring the MPLS Point-to-Multipoint Traffic Engineering: Support for Static Pseudowires feature, ensure that local bindings are unique. Otherwise, traffic unintentionally merges. In the figure below, both PWs share router PE 3 as an endpoint. The local label on each PW is 16, which causes a collision.

Figure 2: Avoiding VC Label Collisions



Label Spoofing

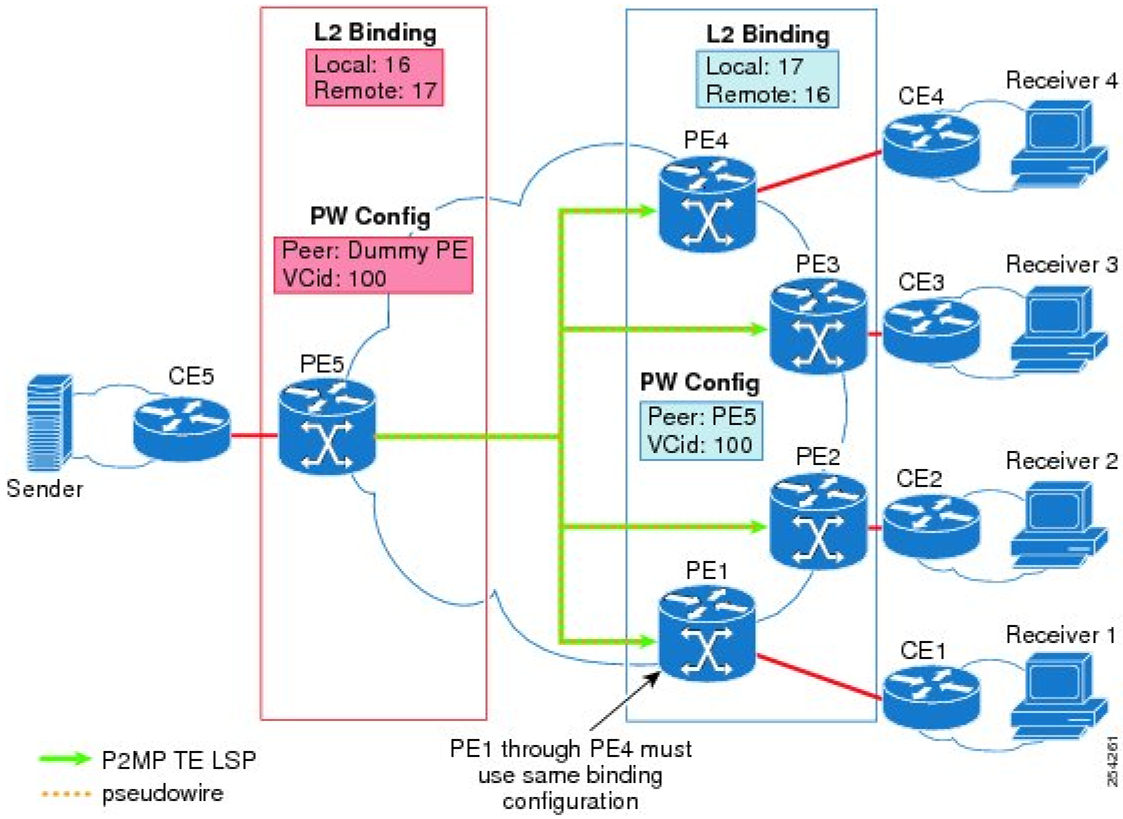
For P2MP static PWs, there is no signaling protocol to verify that the labels are configured correctly on either end. If the labels are not configured correctly, traffic might go to the wrong destinations. Because the traffic going into wrong destinations is a multicast confutation, scalability might be impacted.

The P2MP static PW does not have a context-specific label in the upstream direction and does not use a signaling protocol. Therefore, it is possible to spoof a PW label and route the traffic to the wrong destination. If a PW label is spoofed at the headend, it cannot be validated at the tailend, because the MPLS lookup at the tailend is performed on the global table. So if a spoofed label exists in the global table, traffic is routed to the wrong destination: customer equipment (CE).

The same situation can happen if the user incorrectly configures the static PW label. If the wrong PW label is configured, traffic goes to the wrong destination (CE).

The figure below shows PW label allocation with no context-specific label space.

Figure 3: PW Label Allocation with No Context-Specific Label Space



How to Configure MPLS Point-to-Multipoint Traffic Engineering Support for Static Pseudowires

Configuring the MPLS Label Range

You must specify a static range of MPLS labels using the **mpls label range** command with the **static** keyword.

Procedure

	Command or Action	Purpose
Step 1	enable **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	mpls label range *minimum-value maximum-value{* **static***minimum-static-value maximum-static-value}* **Example:** `Router(config)# mpls label range 1001 1003 static 10000 25000`	Specifies a static range of MPLS labels

Configuring the Headend Routers

Perform this task to configure the headend routers:

- MPLS Static Label range must be configured to configure the Static PW Label under Xconnect.
- Under the Pseudowire class, the P2MPTE tunnel interface should be specified as the preferred path.
- 172.10.255.255 is a fake peer IP address. It is very important that this IP address be reserved by the network domain administrator so that it is not used by any other routers in the network.
- Instead of a fake peer IP address, if peer IP address is used that is present in the routing table, then the traffic will flow through the LSP path (formed by the LDP) towards the peer. This happens only if the fallback option under pseudowire class is not disabled (default) and the preferred path is down.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**pseudowire-class** *class-name* **Example:** `Router(config)# pseudowire-class static-pw`	**S** pecifies a static AToM PW class and enters PW class configuration mode.
Step 4	**encapsulation mpls** **Example:** `Router(config-pw)# encapsulation mpls`	Specifies MPLS as the data encapsulation method for tunneling Layer 2 traffic over the PW.
Step 5	**protocol none** **Example:**	**S** pecifies that no signaling will be used in L2TPv3 sessions created from the static PW.

	Command or Action	Purpose
	`Router(config-pw)# protocol none`	
Step 6	**preferred-path** [**interfacetunnel***tunnel-number*][**disable-fallback**] **Example:** `Router(config-pw)# preferred-path interface tunnel 1 disable-fallback`	Specifies the P2MP tunnel as the traffic path and disables the router from using the default path when the preferred path is unreachable.
Step 7	**exit** **Example:** `Router(config-pw)# exit`	ExitsPW class configuration mode and returns to global configuration mode.
Step 8	**interface tunnel** *number* **Example:** `Router(config)# interface tunnel 1`	Configures a tunnel and enters interface configuration mode.
Step 9	**ip unnumbered loopback** *number* **Example:** `Router(config-if)# ip unnumbered loopback 0`	Enables IP processing on a loopback interface without assigning an explicit IP address to the interface. • Specifying loopback 0 gives the tunnel interface an IP address that is the same as that of loopback interface 0. • This command is not effective until loopback interface 0 has been configured with an IP address.
Step 10	**tunnel mode mpls traffic-eng point-to-multipoint** **Example:** `Router(config-if)# tunnel mode mpls traffic-eng point-to-multipoint`	Enables MPLS P2MP TE on the tunnel.
Step 11	**tunnel destination list mpls traffic-eng** {**identifier***dest-list-id* | **name***dest-list-name*} **Example:** `Router(config-if)# tunnel destination list mpls traffic-eng name in-list-01`	Specifies a destination list to specify the IP addresses of point-to-multipoint destinations.
Step 12	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.

	Command or Action	Purpose
Step 13	**interface loopback** *number* **Example:** `Router(config)# interface loopback 0`	Configures a loopback interface and enters interface configuration mode.
Step 14	**ip address** [*ip-addressmask* [**secondary**]] **Example:** `Router(config-if)# ip address 172.16.255.5 255.255.255.255`	Specifies a primary IP address for the loopback interface.
Step 15	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 16	**interface ethernet** *number* **Example:** `Router(config)# interface ethernet 0/0/0`	Configures an Ethernet interface and enters interface configuration mode.
Step 17	**no ip address** [*ip-addressmask* [**secondary**]] **Example:** `Router(config-if)# no ip address service instance 100 ethernet`	Disables IP processing on the interface.
Step 18	**no keepalive** [*period* [*retries*]] **Example:** `Router(config-if)# no keepalive`	Disables the keepalive packets on the interface. • When the interface goes down, the session continues without shutting down because the keepalive packets are disabled.
Step 19	**xconnect** *peer-ip-address vcid* **encapsulation mpls manual pw-class** *class-name* **Example:** `Router(config-if)# xconnect 172.16.255.255 100 encapsulation mpls manual pw-class static-pw`	Configures a static AToM PW and enters xconnect configuration mode where the static PW labels are set.
Step 20	**mpls label** *local-pseudowire-label remote-pseudowire-label* **Example:**	Configures the AToM static PW connection by defining local and remote circuit labels. • The label must be an unused static label within the static label range configured using the **mplslabelrange** command.

	Command or Action	Purpose
	`Router(config-if-xconn)# mpls label 16 17`	• The **mplslabel**command checks the validity of the label entered and displays an error message if it is not valid. The value supplied for the *remote-pseudowire-label*argument must be the value of the peer PE's local PW label.
Step 21	**mpls control-word** **Example:** `Router(config-if-xconn)# mpls control-word`	Checks whether the MPLS control word is sent. • This command must be set for Frame Relay data-link connection identifier (DLCI) and ATM adaptation layer 5 (AAL5) attachment circuits. For other attachment circuits, the control word is included by default. • If you enable the inclusion of the control word, it must be enabled on both ends of the connection for the circuit to work properly. • Inclusion of the control word can be explicitly disabled using the **nomplscontrol-word**command.
Step 22	**end** **Example:** `Router(config-if-xconn)# end`	Exits xconnect configuration mode.

Configuring the MPLS Label Range

You must specify a static range of MPLS labels using the **mpls label range** command with the **static** keyword.

Procedure

	Command or Action	Purpose
Step 1	enable **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	mpls label range *minimum-value maximum-value{* **static***minimum-static-value maximum-static-value}* **Example:** `Router(config)# mpls label range 1001 1003 static 10000 25000`	Specifies a static range of MPLS labels

Configuring the Tailend Routers

Perform this task to configure the tailend routers:

- MPLS Static Label range should be configured to configure Static PW label under Xconnect.
- The loopback address of the headend router (source of the tree) should be configured under the Xconnect of all tailend routers. The loopback address of the headend router in this example is 10.0.0.1
- All tailend routers should be configured with same remote Virtual Circuit (VC) label of 200, the local VC label of the headend router.

Procedure

Step 1 **enable**

Example:

```
Router> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **configure terminal**

Example:

```
Router# configure terminal
```

Enters global configuration mode.

Step 3 **pseudowire-class** *class-name*

Example:

```
Router(config)# pseudowire-class static-pw
```

Specifies a static AToM PW class and enters PW class configuration mode.

Step 4 **encapsulation mpls**

Example:

```
Router(config-pw)# encapsulation mpls
```

Specifies MPLS as the data encapsulation method for tunneling Layer 2 traffic over the PW.

Step 5 **protocol none**

Example:

```
Router(config-pw)# protocol none
```

Specifies that no signaling will be used in L2TPv3 sessions created from the static PW.

Step 6 **exit**

Example:

```
Router(config-pw)# exit
```

ExitsPW class configuration mode and returns to global configuration mode.

Step 7 **interface loopback** *number*

Example:

```
Router(config)# interface loopback 0
```

Configures a loopback interface and enters interface configuration mode.

Step 8 **ip address** [*ip-addressmask* [**secondary**]]

Example:

```
Router(config-if)# ip address 172.16.255.1 255.255.255.255
```

Specifies a primary IP address for the loopback interface.

Step 9 **exit**

Example:

```
Router(config-if)# exit
```

Exits interface configuration mode and returns to global configuration mode.

Step 10 **interface ethernet** *number*

Example:

```
Router(config)# interface ethernet 0/0
```

Configures an Ethernet interface and enters interface configuration mode.

Step 11 **no ip address** [*ip-addressmask* [**secondary**]]

Example:

```
Router(config-if)# no ip address
```

Disables IP processing on the interface.

Step 12 **no keepalive** [*period* [*retries*]]

Example:

```
Router(config-if)# no keepalive
```

Disables the keepalive packets on the interface.

- When the interface goes down, the session continues without shutting down because the keepalive packets are disabled.

Step 13 **xconnect** *peer-ip-address vcid* **encapsulation mpls manual pw-class** *class-name*

Example:

```
Router(config-if)# xconnect 172.16.255.5 100 encapsulation mpls manual pw-class static-pw
```

Configures a static AToM PW and enters xconnect configuration mode where the static PW labels are set.

Step 14 **mpls label** *local-pseudowire-label remote-pseudowire-label*

Example:

```
Router(config-if-xconn)# mpls label 17 16
```

Configures the AToM static PW connection by defining local and remote circuit labels.

- The label must be an unused static label within the static label range configured using the **mplslabelrange** command.
- The**mplslabel**command checks the validity of the label entered and displays an error message if it is not valid. The value supplied for the *remote-pseudowire-label*argument must be the value of the peer PE's local PW label.

Step 15 **mpls control-word**

Example:

```
Router(config-if-xconn)# mpls control-word
```

Checks whether the MPLS control word is sent.

- This command must be set for Frame Relay data-link connection identifier (DLCI) and ATM adaptation layer 5 (AAL5) attachment circuits. For other attachment circuits, the control word is included by default.
- If you enable inclusion of the control word, it must be enabled on both ends of the connection for the circuit to work properly.
- Inclusion of the control word can be explicitly disabled using the **nomplscontrol-word** command.

Step 16 **end**

Example:

```
Router(config-if-xconn)# end
```

Exits xconnect configuration mode.

Verifying the Static PW Configuration

To verify the L2VPN static PW configuration, use the **showrunning-config** EXEC command. To verify that the L2VPN static PW was provisioned correctly, use the **showmplsl2transportvcdetail**and **pingmplspseudowire**EXEC commands as described in the following steps.

Procedure

---

show mpls l2transport vc detail

For nonstatic PW configurations, this command lists the type of protocol used to send the MPLS labels (such as LDP). For static PW configuration, the value of the signaling protocol field should be Manual.

The following is sample output from the **showmplsl2transportvcdetail**command:

Example:

```
PE21_RSP2#sh mpls l2transport vc 3750 detail
Local interface: Gi0/2/5 up, line protocol up, Eth VLAN 3750 up
  Destination address: 172.10.255.255, VC ID: 3750, VC status: up
    Output interface: Tu3750, imposed label stack {}
    Preferred path: Tunnel3750,  active
    Default path: ready
    No adjacency
  Create time: 3d20h, last status change time: 3d20h
    Last label FSM state change time: 3d20h
  Signaling protocol: Manual
    Status TLV support (local/remote)   : disabled/N/A
      LDP route watch                   : enabled
      Label/status state machine        : established, LruRru
      Last local dataplane   status rcvd: No fault
      Last BFD dataplane     status rcvd: Not sent
      Last BFD peer monitor  status rcvd: No fault
      Last local AC  circuit status rcvd: No fault
      Last local AC  circuit status sent: No fault
      Last local PW i/f circ status rcvd: No fault
      Last local LDP TLV     status sent: No status
      Last remote LDP TLV    status rcvd: Not sent
      Last remote LDP ADJ    status rcvd: No fault
    MPLS VC labels: local 10750, remote 11750
    Group ID: local 21, remote 21
    MTU: local 1500, remote 1500
  Sequencing: receive disabled, send disabled
  Control Word: On (configured: autosense)
  SSO Descriptor: 172.10.255.255/3750, local label: 10750
  Dataplane:
    SSM segment/switch IDs: 1008461/4594 (used), PWID: 112
VC statistics:
    transit packet totals: receive 0, send 105053403
    transit byte totals:   receive 0, send 53787342336
    transit packet drops:  receive 0, seq error 0, send 0
```

Configuration Examples for MPLS Point-to-Multipoint Traffic Engineering Support for Static Pseudowires

Example Configuring the Headend Router (PE5)

In the following sample configuration of the headend router, note the following:

- The **preferred-pathinterfacetunnel1**command specifies the P2MP tunnel as the preferred path.
- The **tunnelmodemplstraffic-engpoint-to-multipoint** command enables the P2MP tunnel.
- The **mplslabel**command defines the static binding.
- The **xconnect**command creates a dummy peer.

```
Router(config)# pseudowire-class STATIC-PW
Router(config-pw-class)# encapsulation mpls
Router(config-pw-class)# protocol none
Router(config-pw-class)# preferred-path interface Tunnel1

!
Router(config)# interface Tunnel1
Router(config-if)# description PE5->PE1,PE2,PE3,PE4-EXCIT
Router(config-if)# ip unnumbered loopback 0
Router(config-if)# tunnel mode mpls traffic-eng point-to-multipoint
Router(config-if)# tunnel destination list mpls traffic-eng name P2MP-EXCIT-DST-LIST
Router(config-if)# tunnel mpls traffic-eng priority 7 7
Router(config-if)# tunnel mpls traffic-eng bandwidth 10000
!
Router(config)# interface loopback 0
Router(config-if)# ip address 172.16.255.5 255.255.255.255
!
Router(config)# interface ethernet 0/0
Router(config-if)# description CONNECTS to CE5
Router(config-if)# no ip address
Router(config-if)# no keepalive
Router(config-if)# xconnect 172.16.255.255 100 encapsulation mpls manual pw-class static-pw
Router(config-if-xconn)# mpls label 16 17
Router(config-if-xconn)# mpls control-word
!
```

Example Configuring the Tailend Router (PE1)

In the following sample configuration of the tailend router, note the following:

- All the tailend routers must use the same binding configuration.
- The **xconnect** command must always be configured on tailend routers.

```
Router(config)# pseudowire-class static-pw
Router(config-pw-class)# encapsulation mpls
Router(config-pw-class)# protocol none
!
Router(config)# interface loopback 0
Router(config-if)# ip address 172.16.255.1 255.255.255.255
!
```

```
Router(config)# interface ethernet 0/0
Router(config-if)# description CONNECTS TO CE1
Router(config-if)# no ip address
Router(config-if)# no keepalive
Router(config-if)# xconnect 172.16.255.5 100 encapsulation mpls manual pw-class static-pw
Router(config-if-xconn)# mpls label 17 16
Router(config-if-xconn)# mpls control-word
!
```

CHAPTER 5

MPLS Traffic Engineering – Bundled Interface Support

Note This technology is not applicable for the Cisco ASR 900 RSP3 Module.

The MPLS Traffic Engineering - Bundled Interface Support feature enables Multiprotocol Label Switching (MPLS) traffic engineering (TE) tunnels over the bundled interfaces—EtherChannel and Gigabit EtherChannel (GEC).

The Resource Reservation Protocol (RSVP) notifies TE about bandwidth changes that occur when member links are added or deleted, or when links become active or inactive. TE notifies other nodes in the network via Interior Gateway Protocol (IGP) flooding. By default, the bandwidth available to TE Label-Switched Paths (LSPs) is 75 percent of the interface bandwidth. You can change the percentage of the global bandwidth available for TE LSPs by using an RSVP command on the bundled interface. Bandwidth reservation and preemption are supported.

The Fast Reroute (FRR) feature is supported on bundled interfaces. FRR is activated when a bundled interface goes down; for example, if you enter the **shutdown** command to shut down the interface or fewer than the required minimum number of links are operational.

Prerequisites for MPLS TE – Bundled Interface Support

- Configure Multiprotocol Label Switching (MPLS) traffic engineering (TE) tunnels.
- Enable Cisco Express Forwarding in global configuration mode.
- Enable Resource Reservation Protocol (RSVP) feature.
- Configure EtherChannel.
- Configure Gigabit EtherChannel.

Restrictions for MPLS TE – Bundled Interface Support

- Traffic engineering over switch virtual interfaces (SVIs) is not supported unless the SVI consists of a bundle of links that represent a single point-to-point interface.
- There must be a valid IP address configuration on the bundled interface and there must not be an IP address configuration on the member links.

Information About MPLS TE – Bundled Interface Support

Cisco EtherChannel Overview

Cisco EtherChannel technology builds upon standards-based 802.3 full-duplex Fast Ethernet to provide network managers with a reliable, high-speed solution for the campus network backbone. EtherChannel technology provides bandwidth scalability within the campus by providing up to 800 Mbps, 8 Gbps, or 80 Gbps of aggregate bandwidth for a Fast EtherChannel, Gigabit EtherChannel, or 10 Gigabit EtherChannel connection, respectively. Each of these connection speeds can vary in amounts equal to the speed of the links used (100 Mbps, 1 Gbps, or 10 Gbps). Even in the most bandwidth-demanding situations, EtherChannel technology helps to aggregate traffic, keeps oversubscription to a minimum, and provides effective link-resiliency mechanisms.

Cisco EtherChannel Benefits

Cisco EtherChannel technology allows network managers to provide higher bandwidth among servers, routers, and switches than a single-link Ethernet technology can provide.

Cisco EtherChannel technology provides incremental scalable bandwidth and the following benefits:

- Standards-based—Cisco EtherChannel technology builds upon IEEE 802.3-compliant Ethernet by grouping multiple, full-duplex point-to-point links. EtherChannel technology uses IEEE 802.3 mechanisms for full-duplex autonegotiation and autosensing, when applicable.
- Flexible incremental bandwidth—Cisco EtherChannel technology provides bandwidth aggregation in multiples of 100 Mbps, 1 Gbps, or 10 Gbps, depending on the speed of the aggregated links. For example, network managers can deploy EtherChannel technology that consists of pairs of full-duplex Fast Ethernet links to provide more than 400 Mbps between the wiring closet and the data center. In the data center, bandwidths of up to 800 Mbps can be provided between servers and the network backbone to provide large amounts of scalable incremental bandwidth.
- Load balancing—Cisco EtherChannel technology comprises several Fast Ethernet links and is capable of load balancing traffic across those links. Unicast, broadcast, and multicast traffic is evenly distributed across the links, providing improved performance and redundant parallel paths. When a link fails, traffic is redirected to the remaining links within the channel without user intervention and with minimal packet loss.
- Resiliency and fast convergence—When a link fails, Cisco EtherChannel technology provides automatic recovery by redistributing the load across the remaining links. When a link fails, Cisco EtherChannel technology redirects traffic from the failed link to the remaining links in less than one second. This convergence is transparent to the end user—no host protocol timers expire and no sessions are dropped.

Cisco Gigabit EtherChannel Overview

Cisco Gigabit EtherChannel (GEC) is a high-performance Ethernet technology that provides transmission rates in Gigabit per second (Gbps). A Gigabit EtherChannel bundles individual ethernet links (Gigabit Ethernet and 10 Gigabit Ethernet) into a single logical link that provides the aggregate bandwidth up to four physical links. All LAN ports in each EtherChannel must be of the same speed and must be configured as either Layer 2 or Layer 3 LAN ports. Inbound broadcast and multicast packets on one link in an EtherChannel are blocked from returning on any other link in the EtherChannel.

Load Balancing and Min-Links in EtherChannel

Load balancing affects the actual and practical bandwidth that can be used for TE. Multilink load balancing uses a per-packet load balancing method. All of the bundle interface bandwidth is available. EtherChannel load balancing has various load balancing methods, depending on the traffic pattern and the load balancing configuration. The total bandwidth available for TE may be limited to the bandwidth of a single member link.

On EtherChannel, min-links is supported only in the Link Aggregation Control Protocol (LACP). For other EtherChannel protocols, the minimum is one link, by default, and it is not configurable. To configure min-links for EtherChannel, use the **port-channel min-links** command.

How to Configure MPLS TE – Bundled Interface Support

Configuring MPLS TE on an EtherChannel Interface

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* [*name-tag*] **Example:** `Device(config)# interface port-channel 1`	Creates an EtherChannel bundle, assigns a group number to the bundle, and enters interface configuration mode.
Step 4	**ip address** *ip-address mask* [**secondary**] **Example:**	Specifies an IP address for the EtherChannel group.

	Command or Action	Purpose
	`Device(config-if)# ip address 10.0.0.4 255.255.255.0`	
Step 5	**mpls traffic-eng tunnels** **Example:** `Device(config-if)# mpls traffic-eng tunnels`	Enables MPLS TE tunnel signaling on an interface. • MPLS TE tunnel should be enabled on the device before enabling the signaling.
Step 6	**mpls traffic-eng backup-path** *tunnel* **Example:** `Device(config-if)# mpls traffic-eng backup-path Tunnel120`	(Optional) Configures the physical interface to use a backup tunnel in the event of a detected failure on that interface.
Step 7	**port-channel min-links** *min-num* **Example:** `Device(config-if)# port-channel min-links 2`	Specifies that a minimum number of bundled ports in an EtherChannel is required before the channel can be active.
Step 8	**ip rsvp bandwidth** [*interface-kbps*] [*single-flow-kbps*] **Example:** `Device(config-if)# ip rsvp bandwidth 100`	Enables RSVP for IP on an interface and specifies a percentage of the total interface bandwidth as available in the RSVP bandwidth pool.
Step 9	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuration Examples for MPLS TE Bundled Interface Support

Example: Configuring MPLS TE on an EtherChannel Interface

```
Device> enable
Device# configure terminal
Device(config)# interface port-channel 1
Device(config-if)# ip address 10.0.0.4 255.255.255.0
Device(config-if)# mpls traffic-eng tunnels
Device(config-if)# mpls traffic-eng backup-path Tunnel 120
Device(config-if)# port-channel min-links 2
Device(config-if)# ip rsvp bandwidth 100
Device(config-if)# end
```

Example: Configuring MPLS TE - Bundled Interface Support over Gigabit Etherchannel

The following example shows how to enable MPLS TE – bundled interface support over GEC on Cisco devices:

```
Device> enable
Device# configure terminal

! Enable global MPLS TE on routers
Device(config)# router ospf 100
Device(config-router)# network 10.0.0.1 0.0.0.255 area 0
Device(config-router)# mpls traffic-eng area 0
Device(config-router)# mpls traffic-eng router-id Loopback 0
Device(config-router)# exit

! Configure  GEC  interface and enable MPLS TE and RSVP on interface
Device(config)# interface Port-channel 1
Device(config-if)# ip address 10.0.0.1 255.255.255.0
Device(config-if)# mpls traffic-eng tunnels
Device(config-if)# ip rsvp bandwidth
Device(config-if)# exit

! Define explicit path
Device(config)# ip explicit-path name primary enable
Device(cfg-ip-expl-path)# next-address 172.12.1.2
Device(cfg-ip-expl-path)# next-address 172.23.1.2
Device(cfg-ip-expl-path)# next-address 172.34.1.2
Device(cfg-ip-expl-path)# next-address 10.4.4.4
Device(cfg-ip-expl-path)# exit

! Configure primary tunnel on head-end device
Device(config)# interface Tunnel 14
Device(config-if)# ip unnumbered Loopback 0
Device(config-if)# tunnel mode mpls traffic-eng
Device(config-if)# tunnel destination 10.10.10.0
Device(config-if)# tunnel mpls traffic-eng autoroute announce
Device(config-if)# tunnel mpls traffic-eng path-option 10 explicit name primary
Device(config-if)# tunnel mpls traffic-eng fast-reroute
Device(config-if)# exit

! Configure backup tunnel on head-end or mid-point device
Device(config)# interface Tunnel 23
Device(config-if)# ip unnumbered Looback 0
Device(config-if)# tunnel mode mpls traffic-eng
Device(config-if)# tunnel destination 10.20.10.0
Device(config-if)# tunnel mpls traffic-eng path-option 10 explicit name backup
Device(config-if)# exit

! Configure backup tunnel on protected GEC interface
Device(config)# interface Port-channel 1
Device(config-if)# ip address 10.0.0.1 255.255.255.0
Device(config-if)# mpls traffic-eng tunnels
Device(config-if)# mpls traffic-eng backup-path Tunnel 23
Device(config-if)# ip rsvp bandwidth percent 20
Device(config-if)# lacp min-bundle 2
Device(config-if)# exit

! Configure GEC interface
```

```
Device(config)# interface GigabitEthernet 0/0/1
Device(config-if)# no ip address
Device(config-if)# channel-group 1 mode active
Device(config-if)# exit

! Configure GEC interface
Device(config)# interface GigabitEthernet 0/0/2
Device(config-if)# no ip address
Device(config-if)# channel-group 1 mode active
Device(config-if)# exit
```

The **show mpls traffic-eng tunnels** command output displays information about a tunnel or one–line information about all tunnels configured on the device:

```
Device# show mpls traffic-eng tunnels tunnel 14

Name: ASR1013_t14                         (Tunnel10) Destination: 10.4.4.4
  Status:
    Admin: up         Oper: up     Path: valid       Signalling: connected
    path option 1, type explicit toR4overR3R3 (Basis for Setup, path weight 3)

  Config Parameters:
    Bandwidth: 0        kbps (Global)  Priority: 7  7   Affinity: 0x0/0xFFFF
    Metric Type: TE (default)
    AutoRoute: enabled  LockDown: disabled Loadshare: 0 [0] bw-based
    auto-bw: disabled
  Active Path Option Parameters:
    State: explicit path option 1 is active
    BandwidthOverride: disabled  LockDown: disabled  Verbatim: disabled


      InLabel  :  -
  OutLabel : Port-channel1, 1608
  Next Hop : 172.16.1.2
  FRR OutLabel : Tunnel23, 4868
  RSVP Signalling Info:
       Src 10.1.1.1, Dst 10.4.4.4, Tun_Id 14, Tun_Instance 35
    RSVP Path Info:
      My Address: 172.12.1.1
      Explicit Route: 172.12.1.2 172.23.1.1 172.23.1.2 172.34.1.1
                      172.34.1.2 10.4.4.4

  History:
    Tunnel:
      Time since created: 17 hours
      Time since path change: 18 minutes, 22 seconds
      Number of LSP IDs (Tun_Instances) used: 35
    Current LSP: [ID: 35]
      Uptime: 18 minutes, 22 seconds
      Selection: reoptimization
    Prior LSP: [ID: 32]
      ID: path option unknown
      Removal Trigger: signalling shutdown


Device# show mpls traffic-eng tunnels brief

show mpls traffic-eng tunnels brief
Signalling Summary:
    LSP Tunnels Process:           running
    Passive LSP Listener:          running
    RSVP Process:                  running
    Forwarding:                    enabled
```

```
    Periodic reoptimization:        every 3600 seconds, next in 3299 seconds
    Periodic FRR Promotion:         Not Running
    Periodic auto-bw collection:    every 300 seconds, next in 299 seconds

P2P TUNNELS/LSPs:
TUNNEL NAME                       DESTINATION      UP IF     DOWN IF   STATE/PROT^M
ASR1013_t14                       10.4.1.1                      -         Po12      up/up
On Mid Router:
P2P TUNNELS/LSPs:
TUNNEL NAME                       DESTINATION      UP IF     DOWN IF   STATE/PROT
ASR1013_t14                        10.4.1.1                   Po12       Po23      up/up
ASR1002F_t23                       10.2.1.1                   Po25        -        up/up
```

The **show mpls traffic-eng fast-reroute** command output displays information about FRR-protected MPLS TE tunnels originating, transmitting, or terminating on this device.

```
Device# show mpls traffic-eng fast-reroute database

P2P Headend FRR information:
Protected tunnel               In-label Out intf/label   FRR intf/label   Status
---------------------------    -------- --------------   --------------   ------

P2P LSP midpoint frr information:
LSP identifier                 In-label Out intf/label   FRR intf/label   Status
---------------------------    -------- --------------   --------------   ------
10.1.1.1 1 [2]                 16        Po23:16          Tu23:16          active
```