

# CCIE Service Provider Lab Workbook v4.0

(<http://labs.ine.com/workbook/toc/service-provider-v4>) »

## CCIE SP v4 Advanced Technology Labs - MPLS TE

### MPLS TE Explicit Paths

« [MPLS Traffic Engineering with IS-IS \(/workbook/view/service-provider-v4/task/mpls-traffic-engineering-with-is-is-Mjg4MA%3D%3D\)](/workbook/view/service-provider-v4/task/mpls-traffic-engineering-with-is-is-Mjg4MA%3D%3D) | [MPLS TE Bandwidth Reservations \(/workbook/view/service-provider-v4/task/mpls-te-bandwidth-reservations-Mjg4Mg%3D%3D\)](/workbook/view/service-provider-v4/task/mpls-te-bandwidth-reservations-Mjg4Mg%3D%3D) »

Last updated: April 22, 2016

CONTENTS

#### Note:

**Initial Configuration & Diagrams:** [Load the initial configuration files for the section named MPLS TE with IS-IS, which can be found in CCIE SPv4 Topology Diagrams & Initial Configurations \(<http://labs.ine.com/workbook/view/service-provider-v4/task/ccie-spv4-topology-diagrams-initial-configs>\).](#) Refer to the **Base IPv4 Diagram** in order to complete this task.

### Task

- R2 and XR1 are preconfigured as PE routers for the MPLS L3VPN customer routers R1 and XR2, respectively, but the core of the Service Provider network is not running LDP.
- Configure the core of the Service Provider network to support MPLS TE tunnels as follows:
  - Enable MPLS TE support for the IS-IS Level 2 core.
  - Set the IS-IS MPLS TE Router-ID to be the Loopback0 interfaces.
  - Enable support for RSVP and MPLS TE on all transit interfaces running IS-IS in the core.
- Configure an MPLS TE tunnel from R2 to XR1 as follows:
  - Unnumber the tunnel to R2's Loopback0 interface.
  - Set the tunnel destination as XR1's Loopback0 interface.
  - Set the tunnel's path option to explicitly follow the path from R2 to R3 to R4 to R6 to R5 to XR1.
  - Configure Autoroute Announce on the tunnel so that the IS-IS core can use it for dynamic routing.
- Configure an MPLS TE tunnel from XR1 to R2 as follows:
  - Unnumber the tunnel to XR1's Loopback0 interface
  - Set the tunnel destination as R2's Loopback0 interface.
  - Set the tunnel's path option to explicitly follow the path from XR1 to R6 to R5 to R4 to R3 to R2.
  - Configure Autoroute Announce on the tunnel so that the IS-IS core can use it for dynamic routing.
- When complete, the following reachability should be achieved:
  - R1 and XR2 should have full IP reachability to each other, and a traceroute should indicate that their L3VPN tunnel is transiting over the MPLS TE tunnels in the core of the SP network.

### Configuration [Click to collapse](#)

```

R2:
mpls traffic-eng tunnels
!
ip explicit-path name TO_XR1 enable
next-address 20.2.3.3
next-address 20.3.4.4
next-address 20.4.6.6
next-address 20.5.6.5
next-address 20.5.19.19
next-address 19.19.19.19
!
interface Tunnel0
ip unnumbered Loopback0
tunnel mode mpls traffic-eng
tunnel destination 19.19.19.19
tunnel mpls traffic-eng autoroute announce
tunnel mpls traffic-eng path-option 1 explicit name TO_XR1
!
interface GigabitEthernet1.23
mpls traffic-eng tunnels
ip rsvp bandwidth
!
interface GigabitEthernet1.24
mpls traffic-eng tunnels
ip rsvp bandwidth
!
router isis
metric-style wide
mpls traffic-eng router-id Loopback0
mpls traffic-eng level-2

R3:
mpls traffic-eng tunnels
!
interface GigabitEthernet1.23
mpls traffic-eng tunnels
ip rsvp bandwidth
!
interface GigabitEthernet1.34
mpls traffic-eng tunnels
ip rsvp bandwidth
!
interface GigabitEthernet1.36
mpls traffic-eng tunnels
ip rsvp bandwidth
!
router isis
metric-style wide
mpls traffic-eng router-id Loopback0
mpls traffic-eng level-2

R4:
mpls traffic-eng tunnels

```

```
!  
interface GigabitEthernet1.24  
  mpls traffic-eng tunnels  
  ip rsvp bandwidth  
!  
interface GigabitEthernet1.34  
  mpls traffic-eng tunnels  
  ip rsvp bandwidth  
!
```

```
interface GigabitEthernet1.45  
  mpls traffic-eng tunnels  
  ip rsvp bandwidth  
!
```

```
interface GigabitEthernet1.46  
  mpls traffic-eng tunnels  
  ip rsvp bandwidth  
!
```

```
router isis  
  metric-style wide  
  mpls traffic-eng router-id Loopback0  
  mpls traffic-eng level-2
```

R5:

```
mpls traffic-eng tunnels  
!  
interface GigabitEthernet1.45  
  mpls traffic-eng tunnels  
  ip rsvp bandwidth  
!
```

```
interface GigabitEthernet1.56  
  mpls traffic-eng tunnels  
  ip rsvp bandwidth  
!
```

```
interface GigabitEthernet1.519  
  mpls traffic-eng tunnels  
  ip rsvp bandwidth  
!
```

```
router isis  
  metric-style wide  
  mpls traffic-eng router-id Loopback0  
  mpls traffic-eng level-2
```

R6:

```
mpls traffic-eng tunnels  
!  
interface GigabitEthernet1.36  
  mpls traffic-eng tunnels  
  ip rsvp bandwidth  
!
```

```
interface GigabitEthernet1.46  
  mpls traffic-eng tunnels  
  ip rsvp bandwidth  
!
```

```
interface GigabitEthernet1.56
```

```

mpls traffic-eng tunnels

ip rsvp bandwidth

!

interface GigabitEthernet1.619

mpls traffic-eng tunnels

ip rsvp bandwidth

!

router isis

metric-style wide

mpls traffic-eng router-id Loopback0

mpls traffic-eng level-2

XR1:

explicit-path name T0_R2

index 1 next-address strict ipv4 unicast 20.6.19.6

index 2 next-address strict ipv4 unicast 20.5.6.5

index 3 next-address strict ipv4 unicast 20.4.5.4

index 4 next-address strict ipv4 unicast 20.3.4.3

index 5 next-address strict ipv4 unicast 20.2.3.2

index 6 next-address strict ipv4 unicast 2.2.2.2

!

interface tunnel-te0

ip4 unnumbered Loopback0

autoroute announce

destination 2.2.2.2

path-option 1 explicit name T0_R2

!

router isis 1

address-family ipv4 unicast

metric-style wide

mpls traffic-eng level-2-only

mpls traffic-eng router-id Loopback0

!

rsvp

interface GigabitEthernet0/0/0/0.519

!

interface GigabitEthernet0/0/0/0.619

!

mpls traffic-eng

interface GigabitEthernet0/0/0/0.519

!

interface GigabitEthernet0/0/0/0.619

!

mpls ldp

```

## Verification

The previous examples that used dynamic path calculation for the MPLS TE tunnels didn't really show any advantage of using MPLS TE in the core over LDP. With LDP you simply enable it on all IGP interfaces, and all IGP learned routes then have MPLS labels allocated for them. With MPLS TE, RSVP will only allocate labels for destinations that you manually have tunnels configured for. For example, if you had 10 PE routers and wanted labels for all of their Loopbacks, with MPLS TE you would have to configure a full mesh of MPLS tunnels between them, i.e.  $n*(n-1)/2$  tunnels, or 45 tunnels. In a case like this, LDP becomes much more scalable. There are ways to overcome the manual configuration of a full

mesh of tunnels, such as using the Auto-Tunnel Mesh feature which will be discussed in a later section. Another alternative to using a full mesh of TE tunnels is configuring PE routers to terminate their tunnels on P routers in the core. All core P nodes would be meshed with either TE tunnels or LDP. This way, PE routers only have to establish TE tunnels to their adjacent core routers instead of to every other PE router in the network. The disadvantage of this design is that flexibility is lost, as the TE tunnel is not end to end between the PEs.

The real advantage of using MPLS TE, as shown in this particular example, is that you have much more granular control over how traffic routes through the core of the network. Instead of simply relying on the IGP shortest path to the destination, MPLS TE essentially allows you to do source routing, because the head-end of the tunnel can control the path that its traffic is going to use.

---

When configuring explicit paths, the most common problems that occur are that the path is entered incorrectly (such as a typo in the addresses) or that one of the requested links along the path can't satisfy the reservation. To make sure that the path is valid, use the `show mpls traffic-eng tunnels` command, as seen below.

```
R2#show mpls traffic-eng tunnels
```

```
P2P TUNNELS/LSPs:
```

```
Name: R2_t0 (Tunnel0) Destination: 19.19.19.19
```

```
Status:
```

```
Admin: up Oper: up Path: valid Signalling: connected
```

```
path option 1, type explicit TO_XR1 (Basis for Setup, path weight 50)
```

```
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 : GigabitEthernet1.23, 18
```

```
Next Hop : 20.2.3.3
```

```
RSVP Signalling Info:
```

```
Src 2.2.2.2, Dst 19.19.19.19, Tun_Id 0, Tun_Instance 47
```

```
RSVP Path Info:
```

```
My Address: 20.2.3.2
```

```
Explicit Route: 20.2.3.3 20.3.4.3 20.3.4.4 20.4.6.4  
                20.4.6.6 20.5.6.6 20.5.6.5 20.5.19.5  
                20.5.19.19 19.19.19.19
```

```
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
```

```
History:
```

```
Tunnel:
```

```
Time since created: 1 days, 1 hours, 21 minutes
```

```
Time since path change: 6 minutes, 45 seconds
```

```
Number of LSP IDs (Tun_Instances) used: 47
```

```
Current LSP: [ID: 47]
```

```
Uptime: 6 minutes, 45 seconds
```

```
Selection: reoptimization
```

```
Prior LSP: [ID: 46]
```

```
ID: path option unknown
```

```
Removal Trigger: configuration changed
```

```
LSP Tunnel XR1_t0 is signalled, connection is up
```

```
InLabel : GigabitEthernet1.23, implicit-null
```

```
Prev Hop : 20.2.3.3
```

```
OutLabel : -
```

```
RSVP Signalling Info:
```

```
Src 19.19.19.19, Dst 2.2.2.2, Tun_Id 0, Tun_Instance 4
```

```

RSVP Path Info:

My Address: 2.2.2.2

Explicit Route: NONE

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

```

P2MP TUNNELS:

P2MP SUB-LSPS:

From this output we need to verify that the path is valid and that the signaling is connected. In the above case both the path and the signaling are working, hence the tunnels are working. Final verification of a traceroute from the customer sites should show traffic following the requested explicit paths.

```

R1#traceroute 20.20.20.20

Type escape sequence to abort.

Tracing the route to 20.20.20.20

VRF info: (vrf in name/id, vrf out name/id)

 1 10.1.2.2 2 msec 2 msec 1 msec
 2 20.2.3.3 [MPLS: Labels 18/16015 Exp 0] 10 msec 8 msec 14 msec
 3 20.3.4.4 [MPLS: Labels 18/16015 Exp 0] 32 msec 31 msec 32 msec
 4 20.4.6.6 [MPLS: Labels 18/16015 Exp 0] 32 msec 32 msec 31 msec
 5 20.5.6.5 [MPLS: Labels 18/16015 Exp 0] 23 msec 21 msec 31 msec
 6 20.5.19.19 21 msec 15 msec 14 msec
 7 10.19.20.20 18 msec * 10 msec

RP/0/0/CPU0:XR2#traceroute 1.1.1.1

Thu May 28 23:34:53.687 UTC

Type escape sequence to abort.

Tracing the route to 1.1.1.1

 1 10.19.20.19 0 msec 0 msec 0 msec
 2 20.6.19.6 [MPLS: Labels 16/19 Exp 0] 0 msec 0 msec 0 msec
 3 20.5.6.5 [MPLS: Labels 16/19 Exp 0] 0 msec 0 msec 0 msec
 4 20.4.5.4 [MPLS: Labels 16/19 Exp 0] 0 msec 0 msec 0 msec
 5 20.3.4.3 [MPLS: Labels 16/19 Exp 0] 0 msec 0 msec 0 msec
 6 10.1.2.2 [MPLS: Label 19 Exp 0] 89 msec 19 msec 19 msec
 7 10.1.2.1 19 msec * 29 msec

```

The potential problem with this design, however, is that there is no fallback path option if one of the hops along the explicit path fails. For example, if the links of R5 go down, both of these tunnels will fail even though there are other potentially valid paths that could be used in the core of the topology. This is shown below.

```
R5#config t
Enter configuration commands, one per line. End with CNTL/Z.
R5(config)#int Gig1
R5(config-if)#shutdown
R5(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet1, changed state to administratively down
%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet1, changed state to down
```

```
R2#
%LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel0, changed state to down
```

```
R2#show mpls traffic-eng tunnels
```

P2P TUNNELS/LSPs:

```
Name: R2_t0 (Tunnel0) Destination: 19.19.19.19
```

Status:

```
Admin: up Oper: down Path: not valid Signalling: Down
path option 1, type explicit TO_XR1
```

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
```

History:

Tunnel:

```
Time since created: 1 days, 1 hours, 25 minutes
```

```
Time since path change: 8 seconds
```

```
Number of LSP IDs (Tun_Instances) used: 55
```

Prior LSP: [ID: 47]

```
ID: path option 1 [55]
```

```
Removal Trigger: path error
```

```
Last Error: CTRL:: Explicit path has unknown address, 20.5.19.19
```

P2MP TUNNELS:

When R5's link is disabled, R2's tunnel interface goes down. From the output above, we can see that there is a path calculation error, because R2 cannot reach the link between R5 and XR1. Even though R2 can still reach the Loopback of XR1, which is the tunnel destination, traffic cannot be routed over the tunnel because there is no valid path option. The result is that R2 and XR1 can still reach each other, but the end customers cannot.

```
R2#traceroute 19.19.19.19 source 2.2.2.2
```

Type escape sequence to abort.

Tracing the route to 19.19.19.19

VRF info: (vrf in name/id, vrf out name/id)

```
 1 20.2.3.3 4 msec
    20.2.4.4 1 msec
    20.2.3.3 1 msec
 2 20.4.6.6 1 msec
    20.3.6.6 2 msec
    20.4.6.6 2 msec
 3 20.6.19.19 77 msec * 2 msec
```

```
R1#traceroute 20.20.20.20
```

Type escape sequence to abort.

Tracing the route to 20.20.20.20

VRF info: (vrf in name/id, vrf out name/id)

```
 1 10.1.2.2 4 msec 1 msec 1 msec
 2 * * *
 3 * * *
 4 * * *
 5 * * *
 6 * * *
```

```
RP/0/0/CPU0:XR2#traceroute 1.1.1.1
```

Wed Apr 4 19:43:34.986 UTC

Type escape sequence to abort.

Tracing the route to 1.1.1.1

```
 1 10.19.20.19 !N * !N
```

To avoid this problem, either Fast Reroute (FRR) Link or Node Protection can be configured to automatically heal around the broken link or router, or a lower-priority dynamic path option can be configured. The second way is the easiest from a configuration point of view, and can be seen below.

```
R2#conf t
```

Enter configuration commands, one per line. End with CNTL/Z.

```
R2(config)#int tunnel0
```

```
R2(config-if)#tunnel mpls traffic-eng path-option 2 dynamic
```

```
R2(config-if)#end
```

```
R2#
```

```
%LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel0, changed state to up
```

```
RP/0/0/CPU0:XR1#config t
```

Wed Apr 4 19:46:01.789 UTC

```
RP/0/0/CPU0:XR1(config)#interface tunnel-te0
```

```
RP/0/0/CPU0:XR1(config-if)#path-option 2 dynamic
```

```
RP/0/0/CPU0:XR1(config-if)#commit
```

Thu May 28 23:40:25.624 UTC

```
RP/0/0/CPU0:May 28 23:40:25.854 : config[65687]: %MGBL-CONFIG-6-DB_COMMIT : Configuration committed by user 'cisco'. Use 'show configuration commit changes 1000000283' to view the changes.
```

Now, with alternate paths, the customer traffic falls back to follow the normal IGP shortest path tree.

```
R1#traceroute 20.20.20.20
Type escape sequence to abort.
Tracing the route to 20.20.20.20
VRF info: (vrf in name/id, vrf out name/id)
 1 10.1.2.2 4 msec 1 msec 1 msec
 2 20.2.3.3 [MPLS: Labels 17/16015 Exp 0] 9 msec 8 msec 6 msec
 3 20.3.6.6 [MPLS: Labels 17/16015 Exp 0] 29 msec 31 msec 32 msec
 4 20.6.19.19 26 msec 10 msec 14 msec
 5 10.19.20.20 16 msec * 11 msec
```

```
RP/0/0/CPU0:XR2#traceroute 1.1.1.1
```

```
Thu May 28 23:41:00.152 UTC
```

```
Type escape sequence to abort.
```

```
Tracing the route to 1.1.1.1
```

```
 1 10.19.20.19 0 msec 0 msec 0 msec
 2 20.6.19.6 [MPLS: Labels 18/19 Exp 0] 29 msec 9 msec 0 msec
 3 20.4.6.4 [MPLS: Labels 17/19 Exp 0] 0 msec 0 msec 0 msec
 4 10.1.2.2 [MPLS: Label 19 Exp 0] 0 msec 9 msec 0 msec
 5 10.1.2.1 0 msec * 9 msec
```

CONTENTS

« MPLS Traffic Engineering with IS-IS (/workbook/view/service-provider-v4/task/mpls-traffic-engineering-with-is-is-Mjg4MA%3D%3D) | MPLS TE Bandwidth Reservations (/workbook/view/service-provider-v4/task/mpls-te-bandwidth-reservations-Mjg4Mg%3D%3D) »