

Implementing Cisco Service Provider Next-Generation Core Network Services

Version 1.01

Lab Guide

Text Part Number: 97-3155-02



Americas Headquarters

Cisco Systems, Inc.
San Jose, CA

Asia Pacific Headquarters

Cisco Systems (USA) Pte. Ltd.
Singapore

Europe Headquarters

Cisco Systems International BV Amsterdam,
The Netherlands

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco Website at www.cisco.com/go/offices.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: www.cisco.com/go/trademarks. Third party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

DISCLAIMER WARRANTY: THIS CONTENT IS BEING PROVIDED "AS IS" AND AS SUCH MAY INCLUDE TYPOGRAPHICAL, GRAPHICS, OR FORMATTING ERRORS. CISCO MAKES AND YOU RECEIVE NO WARRANTIES IN CONNECTION WITH THE CONTENT PROVIDED HEREUNDER, EXPRESS, IMPLIED, STATUTORY OR IN ANY OTHER PROVISION OF THIS CONTENT OR COMMUNICATION BETWEEN CISCO AND YOU. CISCO SPECIFICALLY DISCLAIMS ALL IMPLIED WARRANTIES, INCLUDING WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT AND FITNESS FOR A PARTICULAR PURPOSE, OR ARISING FROM A COURSE OF DEALING, USAGE OR TRADE PRACTICE. This learning product may contain early release content, and while Cisco believes it to be accurate, it falls subject to the disclaimer above.

Table of Contents

Lab Guide	1
Overview	1
Outline	1
Job Aids	2
Pod Access Information	2
Device Information	2
IP Addressing	5
Existing IP Routing	7
Lab 1-1: Implement MPLS in the Service Provider Core	8
Activity Objective	8
Visual Objective	8
Command List	9
Task 1: Configure and Verify OSPF	11
Task 2: Configure Basic MPLS	12
Task 3: Disable IP TTL Propagation	14
Task 4: Configure Conditional Label Distribution	16
Lab 2-1: Implement MPLS TE	18
Activity Objective	18
Visual Objective	18
Command List	19
Task 1: Log Existing Traffic Flow	22
Task 2: Configure Support for MPLS TE	22
Task 3: Build an MPLS TE Tunnel	23
Task 4: Configure the Backup Tunnel	27
Lab 3-1: Use QoS Mechanisms	29
Activity Objective	29
Visual Objective	29
Required Resources	29
Command List	30
Task 1: Explore Network Topology and Test Network Connectivity	31
Task 2: Monitor Interface Statistics Before and After Network Congestion	35
Lab 3-2: Implement MPLS Support for QoS	41
Activity Objective	41
Visual Objective	41
Required Resources	41
Command List	42
Task 1: Update Existing MPLS Tunnel Parameters	45
Task 2: Create a DS-TE Tunnel	46
Task 3: Assign Traffic to the Tunnels (Cisco IOS XE Only)	48
Lab 4-1: Implement Classification and Marking	53
Activity Objective	53
Visual Objective	53
Required Resources	53
Command List	54
Task 1: Configure MQC Classification	55
Task 2: Configure Class-Based Marking	55
Task 3: Configure a GRE Tunnel	57
Task 4: Configure QoS Preclassify	58
Task 5: Monitor QoS Preclassify	61
Lab 5-1: Implement Queuing	63
Activity Objective	63
Visual Objective	63
Command List	64
Task 1: Configure IP SLA	65
Task 2A: Configure CBWFQ on Cisco IOS XR Software	67
Task 2B: Configure CBWFQ on Cisco IOS XE Software	71
Task 3A: Configure LLQ on Cisco IOS XR Software	74
Task 3B: Configure LLQ on Cisco IOS XE Software	77

Lab 5-2: Implement Congestion Avoidance	80
Activity Objective	80
Visual Objective	80
Command List	81
Task 1: Configure IP SLA (Optional)	82
Task 2A: Enable WRED on Cisco IOS XR Software	83
Task 2B: Enable WRED on Cisco IOS XE Software	87
Task 3A: Customize WRED on Cisco IOS XR Software	90
Task 3B: Customize WRED on Cisco IOS XE Software	92
Lab 6-1: Implement Traffic Policing	95
Activity Objective	95
Visual Objective	95
Command List	96
Task 1A: Configure Single-Level Policing on Cisco IOS XR Software	98
Task 1B: Configure Single-Level Policing on Cisco IOS XE Software	100
Task 2: Configure Individual and Aggregate Policers on Cisco IOS Access Switches	103
Task 3: Configure Hierarchical Policing	106
Task 4: Configure LPTS Policing (IOS XR Software Only)	109
Lab 6-2: Implement Traffic Shaping	111
Activity Objective	111
Visual Objective	111
Command List	112
Task 1A: Configure Single-Level Traffic Shaping on IOS XR Software	112
Task 1B: Configure Single-Level Traffic Shaping on IOS XE Software	116
Task 2: Configure Hierarchical Traffic Shaping with CBWFQ on the Child Level	118
Task 3: Configure Hierarchical Traffic Shaping with Shaping on the Child Level	119
Answer Key	121
Lab 1-1 Answer Key: Implement MPLS in the Service Provider Core	121
Lab 2-1 Answer Key: Implement MPLS TE	123
Lab 3-1 Answer Key: Use QoS Mechanisms	130
Lab 3-2 Answer Key: Implement MPLS Support for QoS	132
Lab 4-1 Answer Key: Implement Classification and Marking	134
Lab 5-1 Answer Key: Implement Queuing	135
Lab 5-2 Answer Key: Implement Congestion Avoidance	138
Lab 6-1 Answer Key: Implement Traffic Policing	141
Lab 6-2 Answer Key: Implement Traffic Shaping	145
Appendix A: Lab Topology	149

Lab Guide

Overview

This guide presents the instructions and other information concerning the lab activities for this course. You can find the solutions in the lab activity Answer Key.

Outline

This guide includes these activities:

- Job Aids
- Lab 1-1: Implement MPLS in the Service Provider Core
- Lab 2-1: Implement MPLS TE
- Lab 3-1: Use QoS Mechanisms
- Lab 3-2: Implement MPLS Support for QoS
- Lab 4-1: Implement Classification and Marking
- Lab 5-1: Implement Queuing
- Lab 5-2: Implement Congestion Avoidance
- Lab 6-1: Implement Traffic Policing
- Lab 6-2: Implement Traffic Shaping
- Tear-Out Section

Job Aids

These job aids are available to help you complete lab activities.

Pod Access Information

The instructor will provide you with the team and pod numbers, as well as other team and pod access information. Write down the information in the table for future reference.

Parameter	Default Value	Value
Team number	$z = 1 - 4$	
Pod number	$x = 1, 3, 5, 7$ or $y = 2, 4, 6, 8$	
Remote lab SSH access IP address	128.107.245.9	
Remote lab SSH access username	instr	
Remote lab SSH access password	testMe	
Pod PE (Cisco IOS XR) router username	root	
Pod PE (Cisco IOS XR) router password	1ronMan	
Pod CE, SW, and PE privileged level password	cisco	

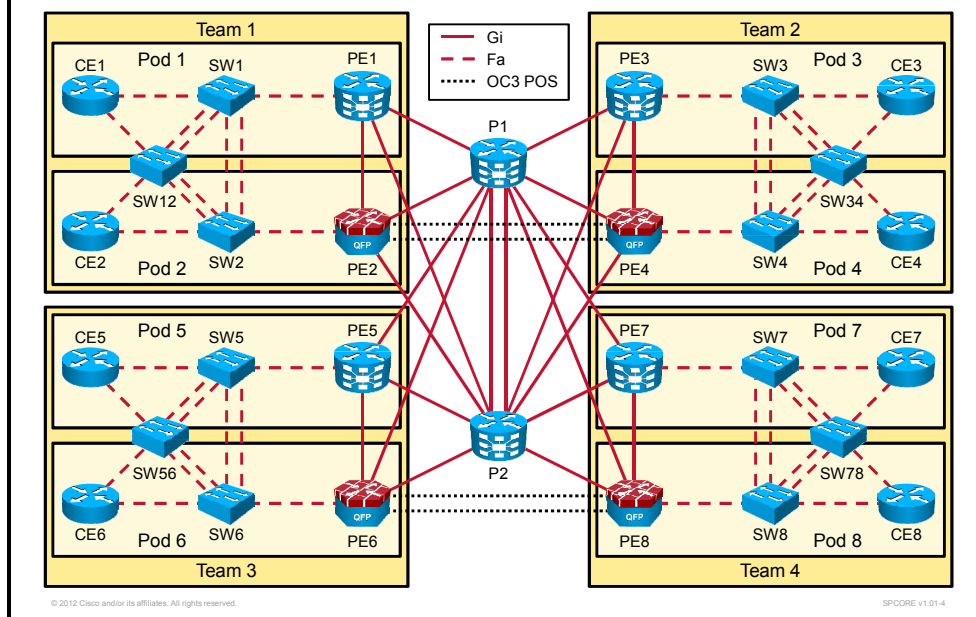
Device Information

This lab topology consists of four (4) teams and eight (8) pods. Two students will work in one pod and two pods will work in one team. Each pod has one switch and two routers. Two pods share one additional switch. All teams share the same core routers (P1 and P2).

The CE routers in both pods are running Cisco IOS Software. The first pod within a team (pods 1, 3, 5, or 7) will work on the PE router running Cisco IOS XR Software, and the second pod within the same team (pods 2, 4, 6, or 8) will work on the PE router running Cisco IOS XE Software.

Devices in the lab are connected with Fast Ethernet and Gigabit Ethernet connections, and two teams have redundant POS connections, as shown in the topology in the figure.

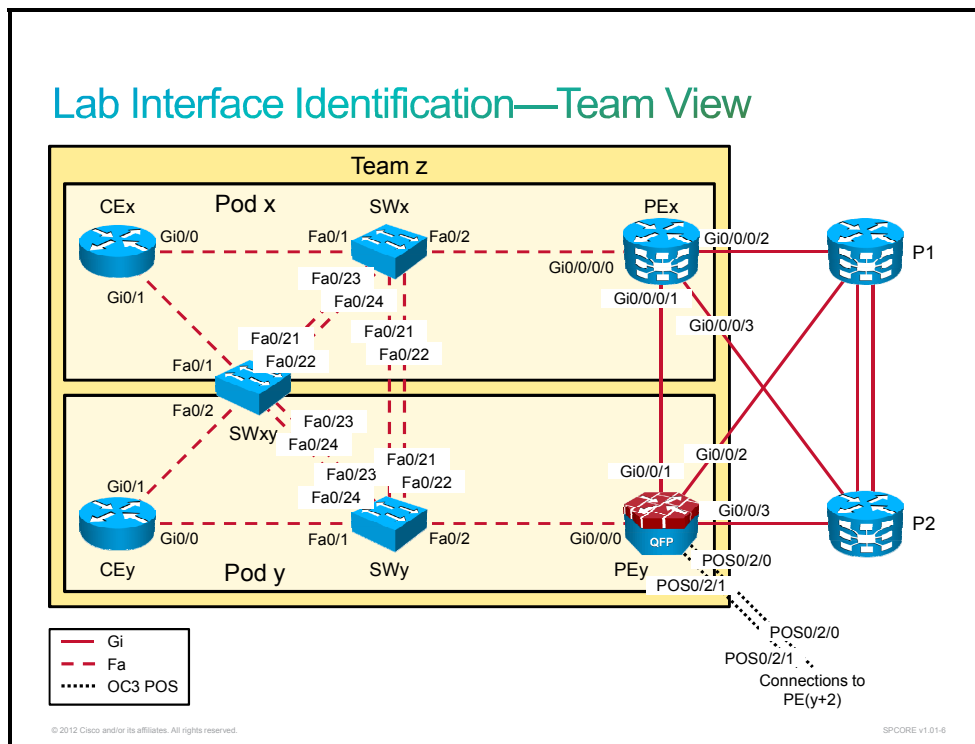
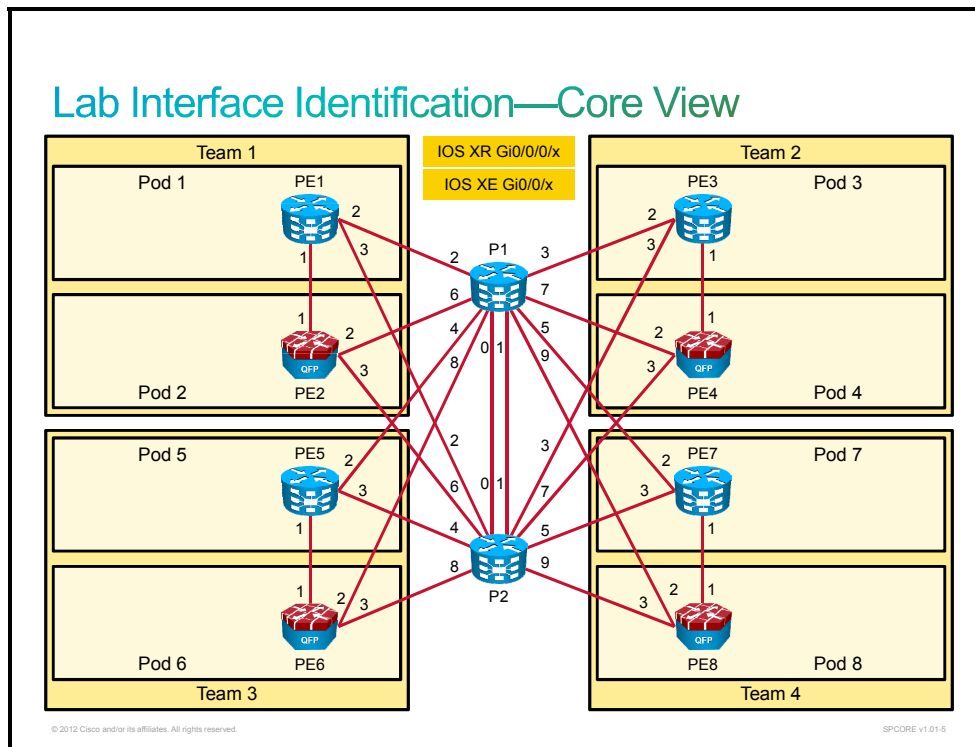
Lab Topology



Device Roles and Loopback IP Addresses

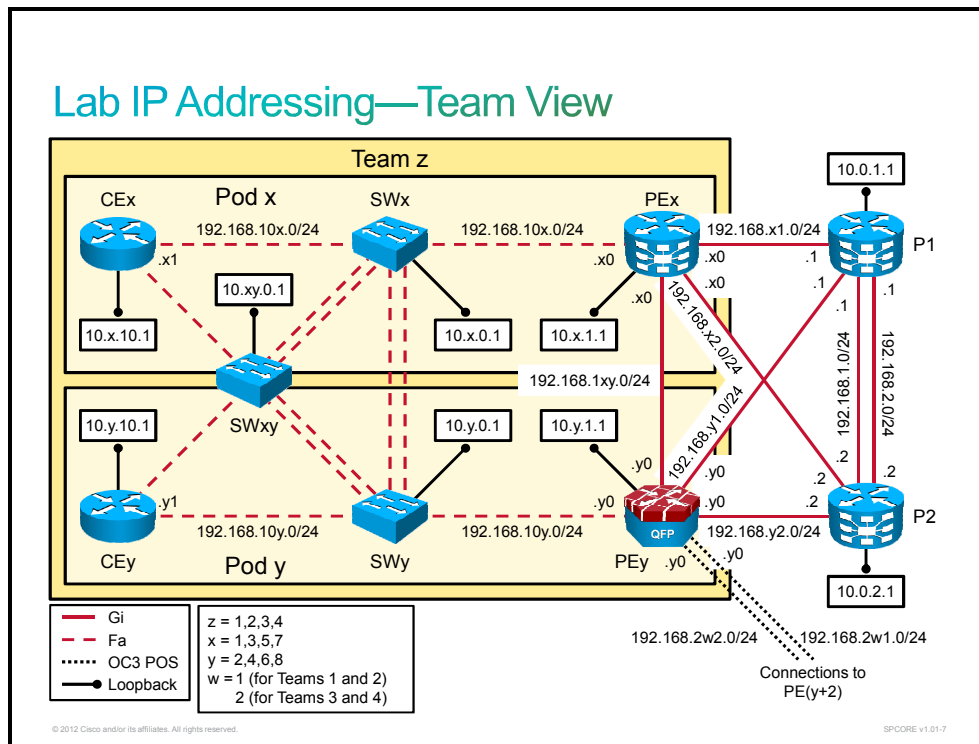
Device Name	Device Role	Lo0 IPv4 Address	Lo0 IPv6 Address
CEx CEy	Cisco 2900 pod router	10.x.10.1/32 10.y.10.1/32	2001:db8:10:x:10::1/128 2001:db8:10:y:10::1/128
PEx PEy	Cisco ASR 9000 or Cisco ASR 1000 pod router	10.x.1.1/32 10.y.1.1/32	2001:db8:10:x:1::1/128 2001:db8:10:y:1::1/128
SWx SWy	Cisco ME340x pod switch	10.x.0.1/32 10.y.0.1/32	2001:db8:10:x:0::1/128 2001:db8:10:y:0::1/128
SWxy	Cisco ME340x pod switch shared inside a team	10.xy.0.1/32	2001:db8:10:xy:0::1/128
P1	Cisco ASR 9000 core router	10.0.1.1/32	2001:db8:10:0:1::1/128
P2	Cisco ASR 9000 core router	10.0.2.1/32	2001:db8:10:0:2::1/128

The figures illustrate the interface identification that is used in this lab setup.

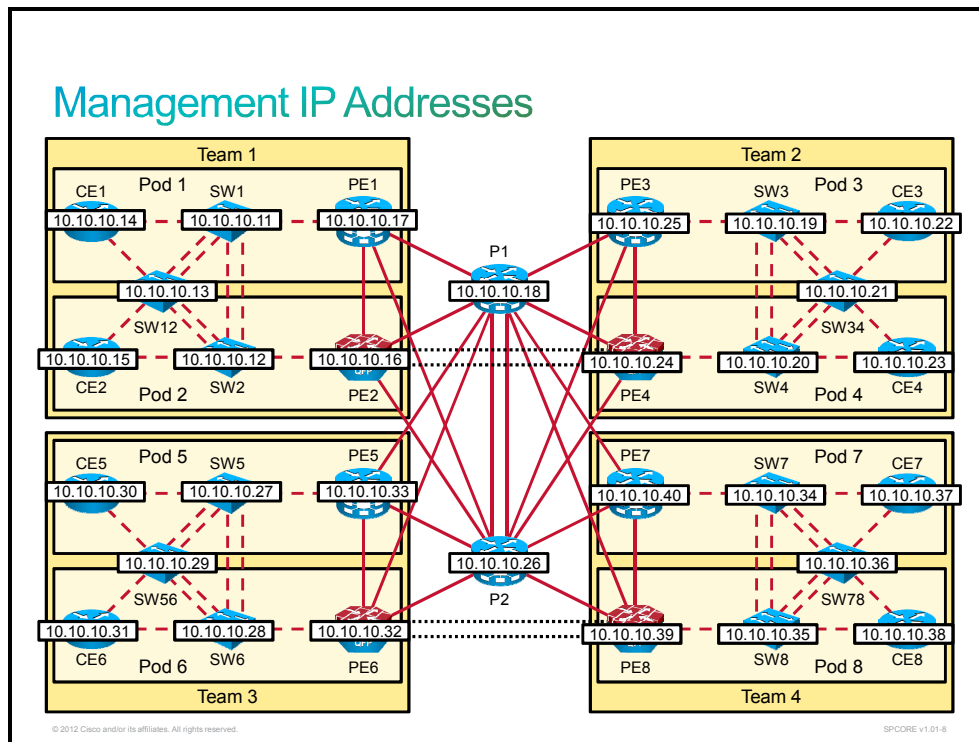


IP Addressing

The figure illustrates the IP addressing scheme that is used in this lab setup.



The figure illustrates the management IP addresses that are used in this lab setup.



Note Replace the x or y with your pod number to get the IP addresses within your pod (x is for odd-numbered pods 1, 3, 5, and 7; y is for even-numbered pods 2, 4, 6, and 8). Replace the xy (where x < y) with the numbers of the pods within the same team (for example, 12, 34, 56, or 78) to get IP addresses on the links between those pods.

Pod IP Addressing

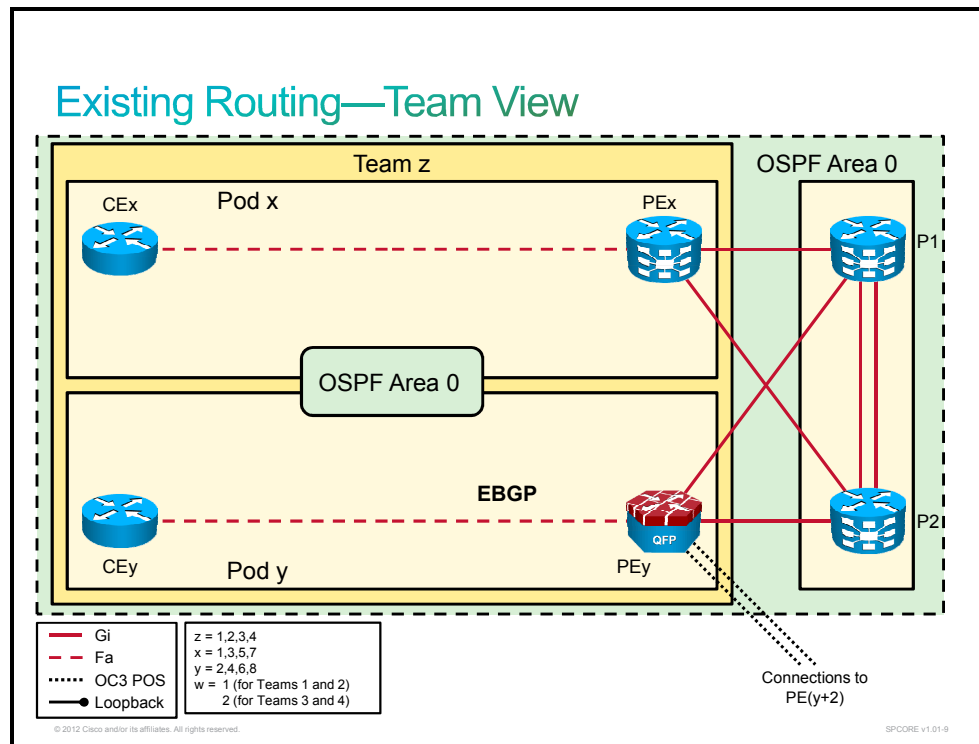
Device	Interface	IPv4 Address	IPv6 Address
CEx	Gi0/0	192.168.10x.x1/24	2001:db8:192:168:10x::x1/80
CEy	Gi0/0	192.168.10y.y1/24	2001:db8:192:168:10y::y1/80
P1		192.168.x1.1/24	2001:db8:192:168:x1::1/80
		192.168.y1.1/24	2001:db8:192:168:y1::1/80
P2		192.168.x2.2/24	2001:db8:192:168:x2::2/80
		192.168.y2.2/24	2001:db8:192:168:y2::2/80
PE2	POS0/2/0	192.168.211.20/24	2001:db8:192:168:211::20/80
	POS0/2/1	192.168.212.20/24	2001:db8:192:168:212::20/80
PE4	POS0/2/0	192.168.211.40/24	2001:db8:192:168:211::40/80
	POS0/2/1	192.168.212.40/24	2001:db8:192:168:212::40/80
PE6	POS0/2/0	192.168.221.60/24	2001:db8:192:168:221::60/80
	POS0/2/1	192.168.222.60/24	2001:db8:192:168:222::60/80
PE8	POS0/2/0	192.168.221.80/24	2001:db8:192:168:221::80/80
	POS0/2/1	192.168.222.80/24	2001:db8:192:168:222::80/80
PEx	Gi0/0/0/0	192.168.10x.x0/24	2001:db8:192:168:10x::x0/80
	Gi0/0/0/1	192.168.1xy.x0/24	2001:db8:192:168:1xy::x0/80
	Gi0/0/0/2	192.168.x1.x0/24	2001:db8:192:168:x1::x0/80
	Gi0/0/0/3	192.168.x2.x0/24	2001:db8:192:168:x2::x0/80
PEy	Gi0/0/0	192.168.10y.y0/24	2001:db8:192:168:10y::y0/80
	Gi0/0/1	192.168.1xy.y0/24	2001:db8:192:168:1xy::y0/80
	Gi0/0/2	192.168.y1.y0/24	2001:db8:192:168:y1::y0/80
	Gi0/0/3	192.168.y2.y0/24	2001:db8:192:168:y2::y0/80

Core IP Addressing

Device	Device IP Address	Peer	Peer IP Address
P1	192.168.1.1/24 2001:db8:192:168:1::1/80	P2	192.168.1.2/24 2001:db8:192:168:1::2/80
	192.168.2.1/24 2001:db8:192:168:2::1/80		192.168.2.2/24 2001:db8:192:168:2::2/80

Existing IP Routing

The figure illustrates the existing IP routing in the lab setup.



OSPF in a single area is preconfigured on the devices. All interlink subnets and the respective loopbacks are advertised in the OSPF configuration.

Lab 1-1: Implement MPLS in the Service Provider Core

Complete this lab activity to practice what you learned in the related module.

Activity Objective

In this lab activity, you will implement MPLS in the lab environment. You will enable LDP on core routers (PE and P routers), experiment with TTL propagation, and configure conditional label distribution.

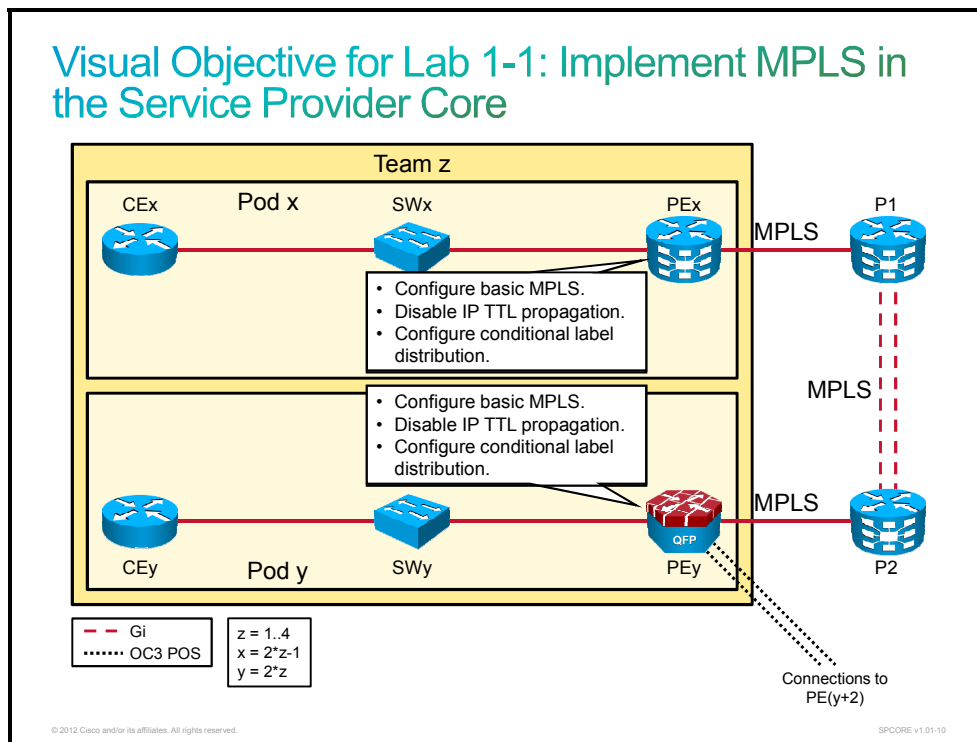
Note Students from two different pods will work in one team. The CE routers in both pods are running Cisco IOS Software. The first pod in the team will work on the PE router that is running Cisco IOS XR Software, and the second pod in the team will work on the PE router that is running Cisco IOS XE Software. Students in the same team should coordinate their lab activity.

In the lab activity, you will work on different Cisco routers running Cisco IOS (c2900), Cisco IOS XE (asr1001), and Cisco IOS XR (asr9k) Software. After completing this activity, you will be able to meet these objectives:

- Configure OSPF in service provider environment
- Enable LDP
- Experiment with TTL propagation
- Configure conditional label distribution

Visual Objective

The figure illustrates what you will accomplish in this activity.



Command List

The table describes the commands that are used in this lab activity.

Cisco IOS and IOS XE Commands

Command	Description
<code>[no] shutdown</code>	Enables or disables an interface on the router
<code>access-list access-list-number {deny permit} source</code>	Defines a standard IP access list
<code>configure terminal</code>	Enters configuration mode
<code>interface interface</code>	Enters interface configuration mode
<code>ip address ip_address subnet_mask</code>	Sets a primary or secondary IPv4 address for an interface and the subnet mask
<code>mpls ip</code>	Enables MPLS forwarding of packets along normally routed paths for a particular interface (interface configuration mode)
<code>mpls ldp advertise-labels [interface interface for prefix-access-list [to peer-access-list]]</code>	Controls the distribution of locally assigned (incoming) labels by means of LDP
<code>mpls ldp router-id interface</code>	Specifies the IP address of a preferred interface or a specific IP address as the LDP router ID
<code>no mpls ldp advertise-labels</code>	Disables label advertisement to all peers for all prefixes
<code>ping dest_IP source source_IP</code>	Verifies connectivity between source IP and destination IP
<code>show mpls forwarding-table</code>	Displays the contents of the MPLS LFIB
<code>show mpls interfaces</code>	Displays information about one or more interfaces that have been configured for MPLS
<code>show mpls ldp bindings</code>	Displays the contents of the LIB
<code>show mpls ldp neighbor</code>	Displays the status of LDP sessions
<code>show ospf neighbors</code>	Displays IS-IS neighbor information
<code>traceroute dest_IP</code>	Discovers the routes that packets actually take when traveling to their destination across an IP network

Cisco IOS XR Commands

Command	Description
<code>[no] shutdown</code>	Enables or disables an interface on the router
<code>commit</code>	Commits changes to the running configuration
<code>configure terminal</code>	Enters configuration mode
<code>disable</code>	Disables label advertisement to all peers for all prefixes in label advertise configuration mode

Command	Description
for <i>prefix-acl</i> [to <i>peer-acl</i>]	Specifies which destinations should have their labels advertised and which LSR neighbors should receive label advertisements
interface <i>interface</i>	Enters interface configuration mode (global mode)
interface <i>type interface-path-id</i>	Configures or enables MPLS LDP on an interface in MPLS LDP configuration mode
ipv4 access-list <i>name</i>	Defines an IPv4 access list by name in global configuration mode
label advertise disable label advertise for <i>x</i> to <i>y</i>	Controls the advertisement of local labels in MPLS LDP configuration mode
mpls ip-ttl-propagate disable forwarded	Configures the behavior controlling the propagation of the IP TTL field to and from the MPLS header
mpls ldp	Enters MPLS LDP configuration mode
ping <i>dest_IP</i> source <i>source_IP</i>	Verifies connectivity between source IP and destination IP (IPv4 and IPv6)
router-id <i>A.B.C.D</i>	Specifies the IP address of a preferred interface or a specific IP address as the LDP router ID (MPLS LDP configuration mode)
show ipv4 interface brief	Displays interface status and IPv4 addresses configured
show mpls forwarding	Displays the contents of the MPLS LFIB
show mpls interfaces	Displays information about one or more interfaces that have been configured for MPLS
show mpls ldp bindings	Displays the contents of the LIB
show mpls ldp neighbor	Displays the status of LDP sessions
show ospf neighbors	Displays OSPF neighbor information
traceroute <i>dest_IP</i>	Discovers the routes that packets actually take when traveling to their destination across an IP network

Task 1: Configure and Verify OSPF

In this task, you will configure and verify the OSPF routing protocol to establish basic connectivity inside the lab environment. An OSPF adjacency should be established from the PE router in your pod to the core router and to the CE router. IP addressing is preconfigured on the lab devices.

Activity Procedure

Complete these steps:

- Step 1** Enable these interfaces, if they are not already enabled:
- Enable interface GigabitEthernet0/0 on your pod CE router.
 - Enable interface GigabitEthernet0/0/0/0 on your pod router PEx and interface GigabitEthernet0/0/0 on your pod router PEy.
 - Enable interface GigabitEthernet0/0/0/2 on your pod router PEx and interface GigabitEthernet0/0/0/3 on your pod router PEy.
 - Enable interfaces FastEthernet0/1 and FastEthernet0/2 on your pod switch SW.
 - On core router P1, enable interfaces GigabitEthernet0/0/0/1, GigabitEthernet0/0/0/2, and the interface connected to your pod PEx router.
 - On core router P2, enable interfaces GigabitEthernet0/0/0/1, GigabitEthernet0/0/0/2, and the interface connected to your pod PEy router.

Note Use Telnet to connect to the core routers P1 and P2. Do not use the console connection to access P1 and P2 core routers.

- Step 2** Verify that OSPF is configured on the PE and CE routers in your pod. Verify that an OSPF adjacency is established between the PE and CE routers in your pod and between the PE and P routers.

Note OSPF is preconfigured on the CE, PE, and P routers. Router P1 has router ID 0.0.0.1 and router P2 has router ID 0.0.0.2.

- Step 3** Verify that an OSPF adjacency is established between the PE and CE routers in your pod and between the PE and P routers.

- Step 4** Verify connectivity from the CEx router to the CEy loopback interface IP address.

Activity Verification

You have completed this task when you attain these results:

- Verify that an OSPF adjacency is established between the PE and CE routers in your pod and between the PE and P routers. The PE router output should be similar to the following, taken from the Pod 1 PE1 router:

```
RP/0/RSP0/CPU0:PE1# show ospf neighbor
Mon Oct 24 12:18:18.696 UTC
```

```
* Indicates MADJ interface
```

```
Neighbors for OSPF 1
```

Neighbor ID	Pri	State	Dead Time	Address	Interface
-------------	-----	-------	-----------	---------	-----------

```

10.1.10.1      1      FULL/BDR      00:00:34      192.168.101.11  GigabitEthernet0/0/0/0
Neighbor is up for 00:02:35
0.0.0.1        1      FULL/BDR      00:00:31      192.168.11.1   GigabitEthernet0/0/0/2
Neighbor is up for 01:44:01

```

- Verify connectivity from the CEx router to the CEy loopback interface IP address.

```
CE1# ping 10.2.10.1
```

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.2.10.1, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms

Task 2: Configure Basic MPLS

In this task, you will configure basic MPLS. You will enable interfaces for MPLS and specify the LDP router ID.

Activity Procedure

Complete these steps:

- Step 1** Enable MPLS for interfaces on path PEx – P1 – P2 – PEy. MPLS should be enabled on both interfaces between the P1 and P2 routers. Verify with your instructor how to coordinate the MPLS configurations on the P1 and P2 routers between the different teams.
- Step 2** Set the LDP router ID for all routers on the PEx – P1 – P2 – PEy path to the IP address of the Loopback0 interface.

Activity Verification

You have completed this task when you attain these results:

- On the PEx router, you should see the P1 router as the LDP neighbor. The output should be similar to the following, taken from the PE1 router:

```
RP/0/RSP0/CPU0:PE1#show mpls ldp neighbor
```

```
Tue Oct 25 07:28:45.618 UTC
```

```
Peer LDP Identifier: 10.0.1.1:0
```

```
TCP connection: 10.0.1.1:646 - 10.1.1.1:46232
```

```
Graceful Restart: No
```

```
Session Holdtime: 180 sec
```

```
State: Oper; Msgs sent/rcvd: 1276/1295; Downstream-Unsolicited
```

```
Up time: 17:47:01
```

```
LDP Discovery Sources:
```

```
GigabitEthernet0/0/0/2
```

```
Addresses bound to this peer:
```

```
10.0.1.1      10.10.10.18      192.168.1.1      192.168.2.1
```

```
192.168.11.1
```

- On the PEy router, you should see the P2 router as the LDP neighbor. The output should be similar to the following, taken from the PE2 router:

```
PE2# show mpls ldp neighbor
```

```
Peer LDP Ident: 10.0.2.1:0; Local LDP Ident 10.2.1.1:0
```

```
TCP connection: 10.0.2.1:646 - 10.2.1.1:45205
```

```
State: Oper; Msgs sent/rcvd: 39/47; Downstream
```

```

Up time: 00:17:41
LDP discovery sources:
  GigabitEthernet0/0/3, Src IP addr: 192.168.22.2
Addresses bound to peer LDP Ident:
  10.10.10.26      10.0.2.1      209.165.200.225  209.165.201.1
  209.165.202.129 192.168.1.2      192.168.2.2      192.168.22.2

```

- On the PEx router, you should see interface GigabitEthernet0/0/0/2 enabled for MPLS. The output should be similar to the following, taken from the PE1 router:

```

RP/0/RSP0/CPU0:PE1# show mpls interfaces
Tue Oct 25 07:35:32.227 UTC
Interface                               LDP           Tunnel    Enabled
-----
GigabitEthernet0/0/0/2                 Yes           No         Yes

```

- On the PEy router, you should see interface GigabitEthernet0/0/3 enabled for MPLS. The output should be similar to the following, taken from the PE2 router:

```

PE2# show mpls interfaces
Interface          IP           Tunnel    BGP Static Operational
GigabitEthernet0/0/3  Yes (ldp)    No        No  No      Yes

```

- Verify the contents of the LFIB table on the PEx router. The output should be similar to the following, taken from the PE1 router:

```

RP/0/RSP0/CPU0:PE1# show mpls forwarding
Tue Oct 25 07:44:42.974 UTC
Local  Outgoing  Prefix      Outgoing    Next Hop      Bytes
Label  Label      or ID       Interface   Hop           Switched
-----
16000  Pop         10.0.1.1/32  Gi0/0/0/2   192.168.11.1  0
16001  16000       10.0.2.1/32  Gi0/0/0/2   192.168.11.1  177360
16003  Pop         192.168.1.0/24  Gi0/0/0/2   192.168.11.1  0
16005  Pop         192.168.2.0/24  Gi0/0/0/2   192.168.11.1  0
16012  16009       10.2.1.1/32  Gi0/0/0/2   192.168.11.1  0
16013  16013       10.2.10.1/32  Gi0/0/0/2   192.168.11.1  0
16014  16003       192.168.22.0/24  Gi0/0/0/2   192.168.11.1  0
16015  16017       192.168.102.0/24  Gi0/0/0/2   192.168.11.1  0
16016  Unlabelled  10.2.1.0/24   Gi0/0/0/1   point2point    0

```

- Verify the contents of the LFIB table on the PEy router. The output should be similar to the following, taken from the PE2 router:

```

PE2# show mpls forwarding-table
Local  Outgoing  Prefix      Bytes Label  Outgoing  Next Hop
Label  Label      or Tunnel Id  Switched     interface
19      16000      10.0.1.1/32    0            Gi0/0/3    192.168.22.2
20      Pop Label  10.0.2.1/32    0            Gi0/0/3    192.168.22.2
23      16007     192.168.71.0/24  0            Gi0/0/3    192.168.22.2
24      Pop Label  192.168.1.0/24  0            Gi0/0/3    192.168.22.2
25      Pop Label  192.168.2.0/24  0            Gi0/0/3    192.168.22.2
26      No Label   10.2.10.1/32    0            Gi0/0/0    192.168.102.81
27      16008     10.1.1.1/32     0            Gi0/0/3    192.168.22.2
31      No Label   10.1.1.0/24     0            Gi0/0/1    192.168.112.10
32      16013     192.168.107.0/24  0            Gi0/0/3    192.168.82.2

```

Task 3: Disable IP TTL Propagation

In this task, you will hide core routers P1 and P2 from customer routers PE when executing the **tracert** operation.

Activity Procedure

Complete these steps:

- Step 1** Perform the **tracert** operation from the CEx router to the loopback IP address of CEy. Observe that core routers P1 and P2 are shown in the output.
- Step 2** Perform the **tracert** operation from the CEy router to the loopback IP address of CEx. Observe that core routers P1 and P2 are shown in the output.

Note	Remember that, by default, IP TTL is copied into the MPLS label TTL field at label imposition.
-------------	--

- Step 3** Hide the P1 and P2 core routers from a **tracert** command only for forwarded packets by disabling IP TTL propagation.
- Step 4** Perform the **tracert** operation from the CEx router to the loopback IP address of CEy. Observe that core routers P1 and P2 are not shown in the output.
- Step 5** Perform the **tracert** operation from the CEy router to the loopback IP address of CEx. Observe that core routers P1 and P2 are not shown in the output.
- Step 6** Perform the **tracert** operation from the PEx router to the loopback IP address of CEy. Observe that core routers P1 and P2 are still shown in the output.
- Step 7** Perform the **tracert** operation from the PEy router to the loopback IP address of CEx. Observe that core routers P1 and P2 are still shown in the output.
- Step 8** Enable IP TTL propagation back on your PE routers.

Activity Verification

You have completed this task when you attain these results:

- Perform the **tracert** operation from the CEx router to the loopback IP address of CEy when IP TTL propagation is enabled:

```
CE1# traceroute 10.2.10.1
Type escape sequence to abort.
Tracing the route to 10.2.10.1
 0 192.168.101.10 0 msec 4 msec 0 msec
 1 192.168.11.1 0 msec 0 msec 4 msec
 2 192.168.1.2 0 msec 0 msec 4 msec
 3 192.168.22.20 0 msec 0 msec 0 msec
 4 192.168.102.21 0 msec * 0 msec
```

- Perform the **tracert** operation from the CEy router to the loopback IP address of CEx when IP TTL propagation is enabled:

```
CE2# traceroute 10.1.10.1
Type escape sequence to abort.
```

Tracing the route to 10.1.10.1

```
1 192.168.102.20 0 msec 0 msec 0 msec
2 192.168.22.2 4 msec 0 msec 0 msec
3 192.168.1.1 4 msec 0 msec 0 msec
4 192.168.11.10 0 msec 4 msec 0 msec
5 192.168.101.11 0 msec * 0 msec
```

- Perform the **tracert** operation from the CEx router to the loopback IP address of CEy when IP TTL propagation is disabled:

CE1# traceroute 10.2.10.1

Type escape sequence to abort.

Tracing the route to 10.2.10.1

```
1 192.168.101.10 0 msec 0 msec 4 msec
2 192.168.22.20 0 msec 0 msec 0 msec
3 192.168.102.21 0 msec * 0 msec
```

- Perform the **tracert** operation from the CEy router to the loopback IP address of CEx when IP TTL propagation is disabled:

CE2# traceroute 10.1.10.1

Type escape sequence to abort.

Tracing the route to 10.1.10.1

```
1 192.168.102.20 0 msec 0 msec 0 msec
2 192.168.101.10 4 msec 0 msec 0 msec
3 192.168.101.11 0 msec * 0 msec
```

- Perform the **tracert** operation from the PEx router to the loopback IP address of CEy:

RP/0/RSP0/CPU0:PE1# traceroute 10.2.10.1

Tue Oct 25 09:02:34.082 UTC

Type escape sequence to abort.

Tracing the route to 10.2.10.1

```
1 192.168.11.1 [MPLS: Label 16013 Exp 0] 2 msec 1 msec 1 msec
2 192.168.1.2 [MPLS: Label 16016 Exp 0] 0 msec 0 msec 1 msec
3 192.168.22.20 [MPLS: Label 26 Exp 0] 0 msec 0 msec 0 msec
4 192.168.102.21 0 msec * 0 msec
```

- Perform the **tracert** operation from the PEy router to the loopback IP address of CEx:

PE2#trace 10.1.10.1

Type escape sequence to abort.

Tracing the route to 10.1.10.1

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

```
1 192.168.22.2 [MPLS: Label 16001 Exp 0] 2 msec 1 msec 1 msec
```

```

2 192.168.1.1 [MPLS: Label 16001 Exp 0] 1 msec 1 msec 1 msec
3 192.168.11.10 [MPLS: Label 16002 Exp 0] 1 msec 1 msec 1 msec
4 192.168.101.11 0 msec * 0 msec

```

Task 4: Configure Conditional Label Distribution

In this task, you will configure conditional label distribution, which will advertise only the label for the PE router loopback address to the core routers.

Activity Procedure

Complete these steps:

- Step 1** Verify the contents of the P1 LIB table on the core router for the CEx loopback IP address network. Observe that the label for this network is received from two neighbors—P2 and PEx.
- Step 2** Verify the contents of the P2 LIB table on the core router for the CEy loopback IP address network. Observe that the label for this network is received from two neighbors—P1 and PEy.
- Step 3** Configure conditional label distribution on the PE1 router, which should advertise only the label for the PEx loopback prefix (10.x.1.1/32) and not the CEx loopback prefix (10.x.1.10/32).
- Step 4** In the same manner, the PEy router should advertise only the label for the PEy loopback prefix (10.y.1.1/32) and not the CEy loopback prefix (10.y.10.1/32).
- Step 5** Verify the contents of the P1 LIB table on the core router for the CEx loopback IP address network. Observe that the label for this network is received from one neighbor—P2. The label is not received from PEx.
- Step 6** Verify the contents of the P2 LIB table on the core router for the CEy loopback IP address network. Observe that the label for this network is received from one neighbor—P1. The label is not received from PEy.

Activity Verification

You have completed this task when you attain these results:

- Verify the contents of the P1 LIB table of the core router for the CEx loopback IP address network before conditional label distribution is configured:

```
RP/0/RSP0/CPU0:P1# show mpls ldp bindings 10.1.10.1/32
```

```
Tue Oct 25 12:11:29.200 UTC
```

```
10.1.10.1/32, rev 375
```

```
Local binding: label: 16001
```

```
Remote bindings: (2 peers)
```

Peer	Label
-----	-----
10.0.2.1:0	16001
10.1.1.1:0	16002

- Verify the contents of the P2 LIB table of the core router for the CEy loopback IP address network before conditional label distribution is configured:

```
RP/0/RSP0/CPU0:P2#sh mpls ldp bindings 10.2.10.1/32
```

```
Tue Oct 25 12:31:03.139 UTC
```

```
10.2.10.1/32, rev 354
```

```
Local binding: label: 16016
```

```
Remote bindings: (2 peers)
```

Peer	Label
-----	-----
10.0.1.1:0	16013
10.2.1.1:0	26

- Verify the contents of the P1 LIB table of the core router for the CEx loopback IP address network after conditional label distribution is configured:

```
RP/0/RSP0/CPU0:P1#show mpls ldp bindings 10.1.10.1/32
```

```
Tue Oct 25 12:12:16.816 UTC
```

```
10.1.10.1/32, rev 375
```

```
Local binding: label: 16001
```

```
Remote bindings: (1 peers)
```

Peer	Label
-----	-----
10.0.2.1:0	16001

- Verify the contents of the P2 LIB table of the core router for the CEy loopback IP address network after conditional label distribution is configured:

```
RP/0/RSP0/CPU0:P2#show mpls ldp bindings 10.2.10.1/32
```

```
Tue Oct 25 12:32:09.463 UTC
```

```
10.2.10.1/32, rev 354
```

```
Local binding: label: 16016
```

```
Remote bindings: (1 peers)
```

Peer	Label
-----	-----
10.0.1.1:0	16013

Lab 2-1: Implement MPLS TE

Complete this lab activity to practice what you learned in the related module.

Activity Objective

In this lab activity, you will implement MPLS TE tunnels in the lab environment. You will enable MPLS TE support and configure a primary MPLS TE tunnel between two PE routers. You will also configure a backup TE tunnel.

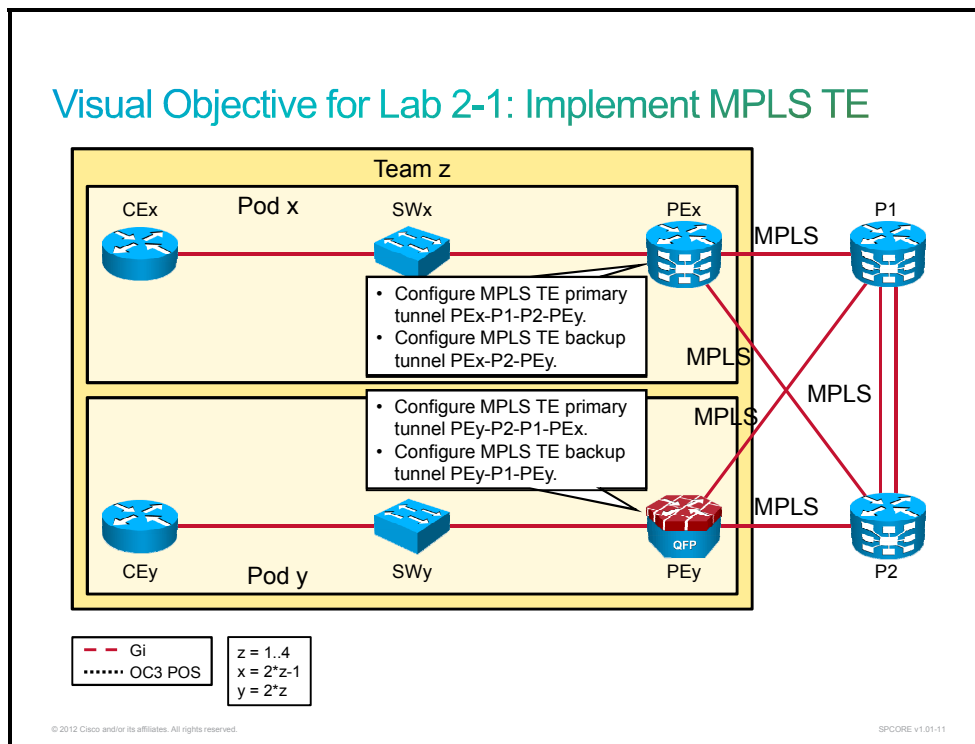
Note Students from two different pods will work in one team. The CE routers in both pods are running Cisco IOS Software. The first pod in the team will work on the PE router that is running Cisco IOS XR Software, and the second pod in the team will work on the PE router that is running Cisco IOS XE Software. Students in the same team should coordinate their lab activity.

In the lab activity, you will work on different Cisco routers running Cisco IOS (c2900), Cisco IOS XE (asr1001), and Cisco IOS XR (asr9k) Software. After completing this activity, you will be able to meet these objectives:

- Log the existing traffic flow
- Configure an MPLS TE primary tunnel between the PE routers in one team
- Configure a backup tunnel
- Monitor the configuration

Visual Objective

The figure illustrates what you will accomplish in this activity.



Command List

The table describes the commands that are used in this lab activity.

Cisco IOS and IOS XE Commands

Command	Description
<code>[no] shutdown</code>	Enables or disables an interface on the router
<code>configure terminal</code>	Enters configuration mode
<code>interface interface</code>	Enters interface configuration mode
<code>interface tunnel number</code>	Enters tunnel interface configuration mode
<code>ip explicit-path name word</code> <code>[enable disable]</code>	Enters the command mode for IP explicit paths and creates or modifies the specified path in global configuration mode
<code>ip rsvp bandwidth [interface-kbps [single-flow-kbps]]</code>	Enables RSVP for IP on an interface in interface configuration mode
<code>ip unnumbered interface</code>	Assigns a source address on the new tunnel
<code>mpls ip</code>	Enables MPLS forwarding of packets along normally routed paths for a particular interface (interface configuration mode)
<code>mpls traffic-eng area number</code>	Enables MPLS TE for indicated OSPF area
<code>mpls traffic-eng backup-path</code>	Configure the interface to use a backup tunnel in the event of a detected failure on the interface in interface configuration mode
<code>mpls traffic-eng router-id</code> { <code>router-id</code> <code>interface-type</code> <code>interface-instance</code> }	Specifies the traffic engineering routing identifier
<code>mpls traffic-eng tunnels</code>	Enables the MPLS TE tunnel feature on a device globally
<code>mpls traffic-eng tunnels</code>	Enables the MPLS TE tunnel feature on an interface in interface configuration mode
<code>next-address [loose strict]</code> <code>ip-address</code>	Specifies the next IP address in the explicit path; use in IP explicit path configuration mode
<code>ping dest_IP source source_IP</code>	Verifies connectivity between source IP and destination IP
<code>show ip route</code>	Displays the content of routing table
<code>show ip rsvp interface</code>	Displays information about all interface with RSVP enabled
<code>show ip rsvp neighbor</code>	Displays neighbors for established RSVP sessions
<code>show mpls traffic-eng tunnels</code>	Displays information about MPLS TE tunnels
<code>show mpls traffic-eng tunnels backup</code>	Displays fast reroute backup protection provided by tunnels
<code>traceroute dest_IP</code>	Discovers the routes that packets actually take when traveling to their destination across an IP network
<code>tunnel destination ip_address</code>	Assigns destination IP address of the MPLS TE tunnel in tunnel interface configuration mode

Command	Description
tunnel mpls traffic-eng autoroute announce	Specify that the IGP should use the tunnel (if the tunnel is up) in its enhanced SPF calculation
tunnel mpls traffic-eng bandwidth <i>bandwidth_in_kbps</i>	Sets the bandwidth required on tunnel in tunnel interface configuration mode
tunnel mpls traffic-eng fast-reroute	Enables fast reroute feature for MPLS TE tunnel interface
tunnel mpls traffic-eng path-option { <i>number</i> { dynamic explicit name <i>path-name</i> }	Configures a path option for an MPLS TE tunnel in interface configuration mode

Cisco IOS XR Commands

Command	Description
[no] shutdown	Enables or disables an interface on the router
autoroute announce	Specifies that the IGP should use the tunnel (if the tunnel is up) in its enhanced SPF calculation
backup-path tunnel-te <i>number</i>	Configure the interface to use a backup tunnel in the event of a detected failure on the interface in mpls traffic-engineering interface mode
bandwidth <i>total-bandwidth max-flow</i>	Sets the total reservable bandwidth and the maximum RSVP bandwidth available for a flow on this interface
commit	Commits changes to the running configuration
configure terminal	Enters configuration mode
destination <i>ip_address</i>	Assigns destination IP address of the MPLS TE tunnel in tunnel interface configuration mode
disable	Disables label advertisement to all peers for all prefixes in label advertise configuration mode
explicit-path { identifier <i>number</i> name <i>pathname</i> }	Configures an explicit path for an MPLS TE tunnel
fast-reroute	Enables the fast reroute feature for an MPLS TE tunnel interface
index <i>index-id</i> next-address [loose strict] ipv4 unicast <i>A.B.C.D</i>	Includes a path entry at a specific index in explicit path configuration mode
interface <i>interface</i>	Enters interface configuration mode (global configuration mode)
interface <i>interface</i>	Enables the MPLS TE tunnel feature on an interface (MPLS TE)
interface <i>interface</i>	Enables RSVP on an interface (RSVP)
interface tunnel-te <i>number</i>	Enters MPLS TE tunnel interface configuration mode
interface type <i>interface-path-id</i>	Configures or enables MPLS LDP on an interface in MPLS LDP configuration mode (MPLS LDP)
ipv4 access-list <i>name</i>	Defines an IPv4 access list by name in global configuration mode
ipv4 unnumbered interface	Assigns a source address on the new tunnel

Command	Description
mpls ldp	Enters MPLS LDP configuration mode
mpls traffic-eng	Enters MPLS TE configuration mode
mpls traffic-eng	Configures an OSPF area for MPLS TE (router OSPF area)
mpls traffic-eng router-id <i>interface</i>	Specifies the traffic engineering router identifier (router OSPF)
path-option <i>preference-priority</i> {dynamic explicit {name pathname}}	Configures a path option for an MPLS TE tunnel
ping <i>dest_IP</i> source <i>source_IP</i>	Verifies connectivity between source IP and destination IP (IPv4 and IPv6)
router-id <i>A.B.C.D</i>	Specifies the IP address of a preferred interface or a specific IP address as the LDP router ID (MPLS LDP)
rsvp	Enters RSVP configuration mode
show ipv4 interface brief	Displays interface status and IPv4 addresses configured
show mpls forwarding	Displays the contents of the MPLS LFIB
show mpls interfaces	Displays information about one or more interfaces that have been configured for MPLS
show mpls ldp bindings	Displays the contents of the LIB
show mpls ldp neighbor	Displays the status of LDP sessions
show mpls traffic-eng tunnels	Displays information about MPLS TE tunnels
show mpls traffic-eng tunnels backup	Displays fast reroute backup protection provided by tunnels
show ospf neighbors	Displays OSPF neighbor information
show route	Displays the content of routing table
show rsvp interface	Displays information about all interface with RSVP enabled
show rsvp session	Displays established RSVP sessions
signaled-bandwidth <i>bandwidth_in_kbps</i>	Sets the bandwidth required on a tunnel in tunnel interface configuration mode
traceroute <i>dest_IP</i>	Discovers the routes that packets actually take when traveling to their destination across an IP network
traceroute mpls ipv4 <i>dest_IP</i>	Discovers the routes that MPLS packets actually take when traveling to their destination across an IP network

Task 1: Log Existing Traffic Flow

In this task, you will verify the path of traffic flow before MPLS TE tunnels are implemented.

Activity Procedure

Complete these steps:

- Step 1** Enable interface GigabitEthernet0/0/0/3 on your pod router PEx and interface GigabitEthernet0/0/0/2 on your pod router PEy.
- Step 2** Enable MPLS on interface GigabitEthernet0/0/0/3 on your pod router PEx and interface GigabitEthernet0/0/0/2 on your pod router PEy.
- Step 3** If it is not already enabled, enable MPLS on all interfaces of core devices P1 and P2 connecting to your PE routers.
- Step 4** Verify the path that traffic takes from router PEx to CEy and from router PEy to CEx. Traffic should flow through only one core P router (P1 or P2).

Activity Verification

You have completed this task when you attain these results:

- Verify the path that traffic takes from router PEx to CEy and from router PEy to CEx. Traffic should flow through only one core P router:

```
RP/0/RSP0/CPU0:PE1# show route | include 10.2.10.1
Wed Nov 23 12:32:44.213 UTC
O    10.2.10.1/32 [110/4] via 192.168.12.2, 00:05:50, GigabitEthernet0/0/0/3
```

```
RP/0/RSP0/CPU0:PE1# traceroute 10.2.10.1
Tue Dec 13 13:31:13.289 UTC
```

```
Type escape sequence to abort.
Tracing the route to 10.2.10.1
```

```
 1  192.168.12.2 [MPLS: Label 16024 Exp 0] 2 msec  1 msec  1 msec
 2  192.168.22.60 0 msec  0 msec  0 msec
 3  192.168.102.21 0 msec  *  0 msec
```

Task 2: Configure Support for MPLS TE

In this task, you will configure all necessary steps to support MPLS TE tunnels between PEx and PEy routers.

Activity Procedure

Complete these steps:

- Step 1** Use Telnet to connect to the core devices P1 and P2 and enable MPLS TE functionality on all interfaces of core devices P1 and P2 connecting to your PE routers. Also enable MPLS functionality for the P1-P2 interconnection.
- Step 2** Enable MPLS TE functionality on all interfaces of the PEx and PEy routers connecting to the core devices P1 and P2.
- Step 3** Enable RSVP on all interfaces participating in MPLS TE and configure the bandwidth available for RSVP reservation. Interfaces participating in MPLS TE are:
 - P1-P2 interconnection link

- All interfaces between core router P1 and PE routers
 - All interfaces between core router P2 and PE routers
- Step 4** For all interfaces mentioned in the previous step:
- Set the maximum amount of bandwidth that may be allocated by RSVP flows to 1 Mb/s
 - Set the maximum amount of bandwidth, in kilobits per second, that may be allocated to a single flow to 200 kb/s
- Step 5** Enable MPLS TE support in the OSPF routing protocol on all routers participating in MPLS TE and specify that the traffic engineering router identifier for all nodes is the IP address associated with the Loopback0 interface.

Activity Verification

There are no verification steps in this task.

Task 3: Build an MPLS TE Tunnel

In this task, you will configure an MPLS TE tunnel between PEx and PEy and vice versa using the explicit path option.

Activity Procedure

Complete these steps:

- Step 1** Build the MPLS TE tunnel between routers PEx and PEy, using these attributes:
- Assign the Loopback0 interface as the source address of the tunnel.
 - Set the bandwidth required on this interface to 100 kb/s.
 - Assign the PEy Loopback0 interface as the destination IP address of the tunnel.
 - Set the path option to an explicit path following the PEx-P1-P2-PEy path.
 - Use the autoroute feature to automatically route traffic to prefixes behind router PEy.
- Step 2** Build the MPLS TE tunnel between router PEy and PEx, using these attributes:
- Assign the Loopback0 interface as the source address of the tunnel.
 - Set the bandwidth that is required on this interface to 100 kb/s.
 - Assign the PEx Loopback0 interface as the destination IP address of the tunnel.
 - Set the path option to an explicit path following the PEy-P2-P1-PEx path.
 - Use the autoroute feature to automatically route traffic to prefixes behind router PEy.
- Step 3** Verify that an RSVP session is established on your pod router PE.
- Step 4** Display information about all interfaces with RSVP enabled on your pod PE router.
- Step 5** Verify that the MPLS TE tunnel is operationally up and check the hop list of current LSPs on your pod router PE.
- Step 6** Verify that prefixes behind the tail end of the tunnel are routed to the MPLS TE tunnel interface.
- Step 7** Perform the **tracert** operation from the CEy router to the loopback IP address of CEx and vice versa.

Activity Verification

You have completed this task when you attain these results:

- Verify that an RSVP session is established on your pod router PE:

```
RP/0/RSP0/CPU0:PE1# show rsvp session
```

```
Thu Nov 24 08:42:20.377 UTC
```

Type	Destination	Add	DPort	Proto/ExtTunID	PSBs	RSBs	Reqs

LSP4	10.2.1.1		1	10.1.1.1	1	1	0

```
PE2# show ip rsvp neighbor
```

Neighbor	Encapsulation	Time since msg rcvd/sent
192.168.22.2	Raw IP	00:00:17 00:00:10

* Neighbors inactive for more than one hour are not shown.

Use the "inactive" keyword to display them.

- Display information about all interfaces with RSVP enabled on your pod PE router:

```
RP/0/RSP0/CPU0:PE1# show rsvp interface
```

```
Thu Nov 24 08:42:58.893 UTC
```

*: RDM: Default I/F B/W % : 75% [default] (max resv/bc0), 0% [default] (bc1)

Interface	MaxBW (bps)	MaxFlow (bps)	Allocated (bps)	MaxSub (bps)

Gi0/0/0/2	2M	200K	100K (5%)	0*
Gi0/0/0/3	2M	200K	0 (0%)	0*

```
PE2# show ip rsvp interface
```

interface	rsvp	allocated	i/f max	flow max	sub max	VRF
Gi0/0/2	ena	0	2M	200K	0	
Gi0/0/3	ena	100K	2M	200K	0	

- Verify that the MPLS TE tunnel is operationally up and check the hop list of current LSPs on your pod router PE:

```
RP/0/RSP0/CPU0:PE1#sh ipv4 interface brief | include tunnel
```

```
Thu Nov 24 08:39:54.793 UTC
```

tunnel-te1	10.1.1.1	Up	Up
------------	----------	----	----

```
PE2#sh ip int bri | inc Tunn
```

Tunnel1	10.2.1.1	YES TFTP	up	up
---------	----------	----------	----	----

```
RP/0/RSP0/CPU0:PE1# show mpls traffic-eng tunnels
```

```
Thu Nov 24 08:44:04.528 UTC
```

```
Signalling Summary:
```

```
      LSP Tunnels Process:  running
      RSVP Process:        running
      Forwarding:          enabled
      Periodic reoptimization: every 3600 seconds, next in 1870 seconds
      Periodic FRR Promotion: every 300 seconds, next in 245 seconds
      Auto-bw enabled tunnels: 0 (disabled)
```

```
Name: tunnel-te1 Destination: 10.2.1.1
```

```
Status:
```

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

```
<...output omitted...>
```

```
Tunnel has been up for: 00:21:40 (since Thu Nov 24 08:22:24 UTC 2011)
```

```
Current LSP:
```

```
Uptime: 00:21:40 (since Thu Nov 24 08:22:24 UTC 2011)
```

```
Path info (OSPF 1 area 0):
```

```
Hop0: 192.168.11.1
```

```
Hop1: 192.168.2.1
```

```
Hop2: 192.168.2.2
```

```
Hop3: 192.168.22.2
```

```
Hop4: 192.168.22.20
```

```
Hop5: 10.2.1.1
```

```
Displayed 1 (of 1) heads, 0 (of 0) midpoints, 0 (of 0) tails
```

```
Displayed 1 up, 0 down, 0 recovering, 0 recovered heads
```

```
PE2# show mpls traffic-eng tunnels
```

```
P2P TUNNELS/LSPs:
```

```
Name: PE2_t1 (Tunnel1) Destination: 10.1.1.1
```

```
Status:
```

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

```
      path option 1, type explicit Explicit (Basis for Setup, path weight 4)
```

```
Config Parameters:
```

```

Bandwidth: 100      kbps (Global)  Priority: 0  0  Affinity: 0x0/0xFFFF
Metric Type: TE (default)

AutoRoute announce: enabled  LockDown: disabled Loadshare: 100 [20000000]
bw-based

auto-bw: disabled

Active Path Option Parameters:

State: explicit path option 1 is active

BandwidthOverride: disabled  LockDown: disabled  Verbatim: disabled


InLabel   :  -
OutLabel  : GigabitEthernet0/0/3, 16005
Next Hop  : 192.168.22.2

RSVP Signalling Info:

    Src 10.2.1.1, Dst 10.1.1.1, Tun_Id 1, Tun_Instance 1

RSVP Path Info:

    My Address: 192.168.22.80

Explicit Route: 192.168.22.2 192.168.2.2 192.168.2.1 192.168.11.1
192.168.11.10 10.1.1.1

Record    Route:    NONE

Tspec: ave rate=100 kbits, burst=1000 bytes, peak rate=100 kbits

RSVP Resv Info:

Record    Route:    NONE

Fspec: ave rate=100 kbits, burst=1000 bytes, peak rate=100 kbits

<...output omitted...>

```

- Verify that the prefixes behind the tail end of the tunnel are routed to the MPLS TE tunnel interface:

```

RP/0/RSP0/CPU0:PE1# show ip route | include tunnel
Thu Nov 24 08:45:40.517 UTC

O   10.2.1.1/32 [110/3] via 10.2.1.1, 00:26:56, tunnel-te1
O   10.2.10.1/32 [110/4] via 10.2.1.1, 00:26:56, tunnel-te1
O   192.168.21.0/24 [110/3] via 10.2.1.1, 00:26:56, tunnel-te1
O   192.168.102.0/24 [110/3] via 10.2.1.1, 00:26:56, tunnel-te1


PE2# show ip route | include Tunnel
O       10.1.1.1/32 [110/3] via 10.1.1.1, 00:10:54, Tunnel1

```

```
O      10.1.10.1/32 [110/4] via 10.1.1.1, 00:10:54, Tunnel1
O      192.168.12.0/24 [110/3] via 10.1.1.1, 00:10:54, Tunnel1
O      192.168.101.0/24 [110/3] via 10.1.1.1, 00:10:54, Tunnel1
```

- Perform the **traceroute** operation from the CEy router to the loopback IP address of CEx and vice versa:

```
CE1# traceroute 10.2.10.1
```

Type escape sequence to abort.

Tracing the route to 10.2.10.1

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

```
 1 192.168.101.10 0 msec 4 msec 0 msec
 2 192.168.11.1 0 msec 16 msec 0 msec
 3 192.168.2.2 4 msec 0 msec 0 msec
 4 192.168.22.20 4 msec 0 msec 0 msec
 5 192.168.102.21 24 msec * 0 msec
```

```
CE2# traceroute 10.1.10.1
```

Type escape sequence to abort.

Tracing the route to 10.1.10.1

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

```
 1 192.168.102.20 0 msec 0 msec 0 msec
 2 192.168.22.2 4 msec 0 msec 0 msec
 3 192.168.2.1 8 msec 0 msec 8 msec
 4 192.168.11.10 0 msec 8 msec 0 msec
 5 192.168.101.11 8 msec * 0 msec
```

Task 4: Configure the Backup Tunnel

In this task, you will configure the fast reroute feature for the existing primary MPLS TE tunnel interface and configure the backup tunnel.

Activity Procedure

Complete these steps:

- Step 1** Enable the fast reroute feature for the existing primary MPLS TE tunnel on your PE router.
- Step 2** Configure the backup MPLS TE tunnel on your PE router for the existing primary MPLS TE tunnel:
 - The backup tunnel on PEx should take the path PEx-P2-PEy.
 - The backup tunnel on PEy should take the path PEy-P1-PEx.
 - Set the bandwidth that is required for the backup tunnel to 500 kb/s.

Step 3 Verify that the backup tunnel is up and operational.

Step 4 Verify basic fast reroute information for the backup tunnel.

Activity Verification

You have completed this task when you attain these results:

- Verify that the backup tunnel is up and operational:

```
RP/0/RSP0/CPU0:PE1# show ipv4 interface brief | include 1000
```

```
Wed Nov 30 08:17:29.714 UTC
```

```
tunnel-te1000          10.1.1.1          Up                Up
```

```
PE2# show ip interface brief | include Tunnel1000
```

```
Tunnel1000            10.2.1.1          YES TFTP         up                up
```

- Verify basic fast reroute information for the backup tunnel:

```
RP/0/RSP0/CPU0:PE1# show mpls traffic-eng tunnels backup
```

```
Wed Nov 30 08:18:15.598 UTC
```

```
tunnel-te1000
```

```
Admin: up, Oper: up
```

```
Src: 10.1.1.1, Dest: 10.2.1.1, Instance: 6
```

```
Fast Reroute Backup Provided:
```

```
Protected LSPs: 1 (0 active)
```

```
Protected S2L Sharing Families: 0
```

```
Protected S2Ls: 0
```

```
Backup BW: any-class (no limit specified), Inuse: 100 kbps
```

```
Protected i/fs: Gi0/0/0/2
```

```
PE2# show mpls traffic-eng tunnels backup
```

```
PE8_t1000
```

```
LSP Head, Admin: up, Oper: up
```

```
Tun ID: 1000, LSP ID: 19, Source: 10.8.1.1
```

```
Destination: 10.1.1.1
```

```
Fast Reroute Backup Provided:
```

```
Protected i/fs: Gi0/0/3
```

```
Protected LSPs/Sub-LSPs: 0, Active: 0
```

```
Backup BW: any pool unlimited; inuse: 0 kbps
```

```
Backup flags: 0x0
```

Lab 3-1: Use QoS Mechanisms

Complete this lab activity to practice what you learned in the related module.

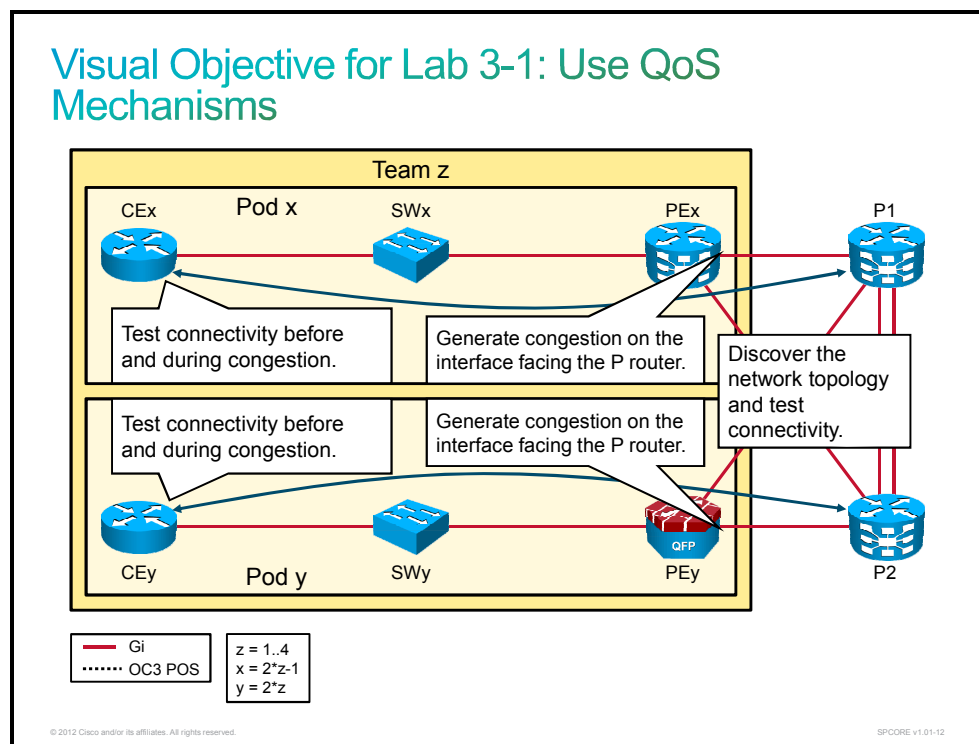
Activity Objective

In this activity, you will verify network connectivity and perform baseline QoS measurement. After completing this activity, you will be able to meet these objectives:

- Explore service provider NGN topology and test network connectivity between routers
- Identify interface statistics
- Test how the network responds to excessive loads
- Monitor interface statistics before and after network congestion and enable shaping and policing

Visual Objective

The figure illustrates what you will accomplish in this activity.



Required Resources

These are the resources and equipment that are required to complete this activity:

- A PC with access to the Internet
- An SSH client installed on the PC

Command List

The table describes the commands that are used in this activity.

Cisco IOS and IOS XE Commands

Command	Description
configure terminal	Enters configuration mode
class <i>name_of_class</i>	Creates a class map to be used for matching packets to a specified class
class-map <i>class_name</i>	Creates a class to be used with a class map and enters class-map configuration mode. The class map is used for matching packets to the specified class.
clear counters <i>interface</i>	Clears show interface counters and associated statistics on the specified interface
enable	Enters privilege EXEC mode
interface <i>interface</i>	Enters interface configuration mode
match protocol <i>protocol</i>	Configures the match criterion for a class map on the basis of a specified protocol
ping <i>destination_address</i> size <i>size_of_packet</i> repeat <i>packets</i> source <i>interface</i>	Pings the destination address and specifies packet size, number of packets sent to destination, and source interface
police rate <i>bps</i>	Configures basic traffic policing
policy-map <i>policy_name</i>	Enters policy-map configuration mode and creates or modifies a policy map that can be attached to one or more interfaces
service-policy { input output } <i>policy-map-name</i>	Attaches service traffic policy to an interface and specifies the direction in which the traffic policy should be applied
shape [average peak] <i>kbps</i>	Configures basic traffic shaping
show interfaces <i>interface</i>	Displays statistics of an interface
show ip interface brief	Displays summarized IP interface status
show running-config interface <i>interface</i>	Display the running configuration on an interface

Cisco IOS XR Commands

Command	Description
<code>configure terminal</code>	Enters configuration mode
<code>class name_of_class</code>	Creates a class map to be used for matching packets to a specified class
<code>class-map class_name</code>	Creates a class to be used with a class map and enters class-map configuration mode. The class map is used for matching packets to the specified class.
<code>clear counters interface</code>	Clears show interface counters and associated statistics on a specified interface
<code>enable</code>	Enters privilege EXEC mode
<code>interface interface</code>	Enters interface configuration mode
<code>match protocol protocol</code>	Configures the match criterion for a class map on the basis of a specified protocol
<code>ping destination_address size size_of_packet count packets source interface</code>	Pings the destination address and specifies packet size, number of packets sent to destination, and source interface
<code>police rate rate [bps kbps mbps]</code>	Configures traffic policing and enters policy-map police configuration mode
<code>policy-map policy_name</code>	Enters policy-map configuration mode and creates or modifies a policy map that can be attached to one or more interfaces
<code>service- policy {input output} po licy-map-name</code>	Attaches a service traffic policy to an interface and specifies the direction in which the traffic policy should be applied
<code>shape average rate [bps kbps mbps]</code>	Configures traffic shaping
<code>show interfaces interface</code>	Displays statistics of an interface
<code>show ip interface brief</code>	Displays summarized IP interface status
<code>show running-config interface interface</code>	Displays the running configuration on an interface

Task 1: Explore Network Topology and Test Network Connectivity

In this task, you will explore network topology, identify how P routers are connected with each other and with the CE router, and perform basic routing verification using the **ping** command.

Activity Procedure

Complete these steps:

- Step 1** Log into the CE router of your pod and display the status of IP interfaces. Also, show the interface configuration to view the subnet mask of the configured IP address. The output of the **show** command entered on the CE router should be similar to the following:

```
CE1#show ip interface brief
Interface          IP-Address      OK? Method Status
Protocol
```

```
GigabitEthernet0/0          192.168.101.11  YES NVRAM  up
up
GigabitEthernet0/1          unassigned      YES NVRAM
administratively down down
Loopback0                   10.1.10.1       YES manual up
up
```

Step 2 Repeat the previous step for router PE in your pod and both P routers. The output of the **show** command entered on the PE and P router should be similar to the following:

```
RP/0/RSP0/CPU0:PE1#show ip interface brief
```

```
Sat Oct 15 08:27:44.462 UTC
```

Interface Protocol	IP-Address	Status
Loopback0 Up	10.7.1.1	Up
MgmtEth0/RSP0/CPU0/0 Up	10.10.10.40	Up
MgmtEth0/RSP0/CPU0/1 Down	unassigned	Shutdown
GigabitEthernet0/0/0/0 Up	192.168.101.10	Up
GigabitEthernet0/0/0/1 Down	192.168.178.70	Shutdown
GigabitEthernet0/0/0/2 Up	192.168.11.10	Up
GigabitEthernet0/0/0/3 Down	192.168.72.70	Shutdown
GigabitEthernet0/0/0/4 Down	unassigned	Shutdown

< Output omitted >

```
RP/0/RSP0/CPU0:P1#show ip interface brief
```

```
Sat Oct 15 08:42:23.753 UTC
```

Interface Protocol	IP-Address	Status
Loopback0 Up	10.0.1.1	Up
MgmtEth0/RSP0/CPU0/0 Up	10.10.10.18	Up
MgmtEth0/RSP0/CPU0/1 Down	unassigned	Shutdown
GigabitEthernet0/0/0/0 Up	192.168.1.1	Up
GigabitEthernet0/0/0/1 Up	192.168.2.1	Up
GigabitEthernet0/0/0/2 Up	192.168.11.1	Up
GigabitEthernet0/0/0/3 Up	192.168.31.1	Up

GigabitEthernet0/0/0/4 Up	192.168.51.1	Up
GigabitEthernet0/0/0/5 Up	192.168.71.1	Up
GigabitEthernet0/0/0/6 Up	192.168.21.1	Up
GigabitEthernet0/0/0/7 Down	192.168.41.1	Down
GigabitEthernet0/0/0/8 Up	192.168.61.1	Up
GigabitEthernet0/0/0/9 Down	192.168.81.1	Down
GigabitEthernet0/0/0/10 Down	unassigned	Shutdown

< output omitted >

RP/0/RSP0/CPU0:P2#show ip interface brief

Sat Oct 15 08:42:35.851 UTC

Interface Protocol	IP-Address	Status
Loopback0 Up	10.0.2.1	Up
Loopback100 Up	209.165.200.225	Up
Loopback101 Up	209.165.201.1	Up
Loopback102 Up	209.165.202.129	Up
MgmtEth0/RSP0/CPU0/0 Up	10.10.10.26	Up
MgmtEth0/RSP0/CPU0/1 Down	unassigned	Shutdown
GigabitEthernet0/0/0/0 Up	192.168.1.2	Up
GigabitEthernet0/0/0/1 Up	192.168.2.2	Up
GigabitEthernet0/0/0/2 Up	192.168.12.2	Up
GigabitEthernet0/0/0/3 Down	192.168.32.2	Down
GigabitEthernet0/0/0/4 Up	192.168.52.2	Up
GigabitEthernet0/0/0/5 Up	192.168.72.2	Up
GigabitEthernet0/0/0/6 Up	192.168.22.2	Up
GigabitEthernet0/0/0/7 Up	192.168.42.2	Up
GigabitEthernet0/0/0/8 Up	192.168.62.2	Up
GigabitEthernet0/0/0/9 Up	192.168.82.2	Up

GigabitEthernet0/0/0/10 unassigned Shutdown
Down

< output omitted >

Step 3 Write down the IP subnets between these routers in the following table:

	CE	P1	P2
PE			

Step 4 Test ping connectivity between the CE and PE routers using Loopback0 as a source and destination of the test. The connectivity test should be successful.

Step 5 Test ping connectivity between the PE and P1 routers using Loopback0 as a source and destination of the test. The connectivity test should be successful.

Step 6 Test ping connectivity between the PE and P2 router using Loopback0 as a source and destination of the test. The connectivity test should be successful.

Step 7 Test ping connectivity between the CE router and both P routers using Loopback0 as a source and destination of the test. The connectivity test should be successful.

Step 8 If one or more of these tests failed, resolve the issue and try the test again. Otherwise, continue to the next task.

Activity Verification

You have completed this task when you attain these results:

Note	Outputs in the verification section are taken from team 1 and pod 1 and may differ from your outputs.
-------------	---

- Interfaces from IP subnets in the table should be up, and the line protocol should be up.
- The connectivity test between routers should be successful, as shown here:

```
RP/0/RSP0/CPU0:PE1# ping 10.1.10.1 source Loopback0
Sat Oct 15 08:32:30.082 UTC
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.7.10.1, timeout is 2
seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max =
1/1/1 ms
```

```
RP/0/RSP0/CPU0:PE1# ping 10.0.1.1 source Loopback0
Sat Oct 15 08:33:15.833 UTC
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.1.1, timeout is 2
seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max =
1/1/1 ms
```

```
RP/0/RSP0/CPU0:PE1# ping 10.0.2.1 source Loopback0
Sat Oct 15 08:33:54.730 UTC
Type escape sequence to abort.
```

```
Sending 5, 100-byte ICMP Echos to 10.0.2.1, timeout is 2
seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max =
1/1/2 ms
```

```
CE1# ping 10.0.1.1 source Loopback0
```

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.1.1, timeout is 2
seconds:
Packet sent with a source address of 10.7.10.1
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max =
1/1/4 ms
```

```
CE1# ping 10.0.2.1 source Loopback0
```

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.2.1, timeout is 2
seconds:
Packet sent with a source address of 10.7.10.1
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max =
1/1/4 ms
```

Task 2: Monitor Interface Statistics Before and After Network Congestion

In this task, you will test network connectivity before and after network congestion. You will configure a traffic-shaping mechanism on the interface to throttle traffic passing through it and learn how to recognize interface congestion using interface statistics.

Activity Procedure

Complete these steps:

- Step 1** From the CE router, ping the P router Loopback0 interface using a packet size of 1500 bytes and using Loopback0 as the source. Use 100 ping packets. Write down the statistics for the round-trip time:

round-trip min/avg/max = ____/____/____ ms

- Step 2** Paste the configuration found in this step to the PE router and commit the configuration changes. This configuration will shape all IPv4 traffic on the exiting interface to 64 kb/s, thus creating a bottleneck between the PE and P routers.

(Cisco IOS XR)

```
class-map IPv4traffic
  match protocol ipv4
!
policy-map shapeIPv4
  class IPv4traffic
    shape average 64 kbps
```

```

!
!
interface gigabitEthernet 0/0/0/2
 service-policy output shapeIPv4
!

```

(Cisco IOS XE)

```

class-map IPv4traffic
 match any
!
policy-map shapeIPv4
 class IPv4traffic
  shape average 64000
!
!
interface gigabitEthernet 0/0/3
 service-policy output shapeIPv4

```

Note	In the following steps, you will only use one P router. Pods 1, 3, 5, and 7 will use router P1, and Pods 2, 4, 6, and 8 will use router P2.
-------------	---

Step 3 Clear the interface statistics on the PE router interface that connects to the P router:

```
clear counters gigabitEthernet 0/0/0/2
```

Step 4 From the CE router, ping the P router Loopback0 interface using a packet size of 1500 bytes and using Loopback0 as the source. Use 100 ping packets. Write down the statistics for the round-trip time:

round-trip min/avg/max = ____/____/____ ms

How does the round-trip time differ before and after applying a traffic shaping policy?

Step 5 From the PE router, ping the P router Loopback0 interface using a count of 5000 packets, size of 1500 bytes, and Loopback0 as a source.

Step 6 While pinging the P router from the PE router, log into the CE router again and repeat Step 4. Write down the statistics for the round-trip time.

round-trip min/avg/max = ____/____/____ ms

Success rate: ____%

How do the two traffic flows affect the round-trip time and the success rate in the case of traffic shaping?

Step 7 Stop the ping sourced from the PE router to the P router and display the P router-facing interface statistics on the PE router. Write down the value for total output drops:

Total output drops: ____

Step 8 On the PE router, clear the interface statistics counters on the P router-facing interface.

```
clear counters gigabitEthernet 0/0/0/2
```

Step 9 Paste the configuration found in this step to the PE router and commit the configuration changes. Now you will use policing instead of shaping for the IPv4 traffic class. The policing rate is the same as the shaping rate.

(Cisco IOS XR)

```
policy-map shapeIPv4
class IPv4traffic
  no shape average 64 kbps
  police rate 64 kbps
!
```

(Cisco IOS XE)

```
policy-map shapeIPv4
class IPv4traffic
  no shape average 64000
  police 64000
```

- Step 10** From the CE router, ping the P router Loopback0 interface using a packet size of 1500 bytes and using Loopback0 as the source. Use 30 ping packets. Write down the statistics about round-trip time and success rate:

round-trip min/avg/max = ____/____/____ ms
Success rate: ____%

What is the effect of traffic policing on the round-trip time and success rate?

- Step 11** On PE router, show the interface statistics counters on the P router-facing interface. Write down the value for total output drops:

Total output drops: ____

How does traffic policing affect the packet drops on the output interface?

- Step 12** Paste the configuration found in this step to the routers PEx and PEy. This action removes the service and clears the policy map shapeIPv4 and the class map IPv4traffic.

(Cisco IOS XR)

```
configure terminal
interface gigabitethernet 0/0/0/2
  no service-policy output shapeIPv4
exit
no policy-map shapeIPv4
no class-map IPv4traffic
```

(Cisco IOS XE)

```
configure terminal
interface gigabitethernet 0/0/3
  no service-policy output shapeIPv4
exit
no policy-map shapeIPv4
no class-map IPv4traffic
```

Activity Verification

You have completed this task when you attain this result:

- The ping statistics before and after applying shaping to all IPv4 traffic should be different:

Before applying traffic shaping:

```
CE1# ping 10.0.1.1 repeat 100 size 1500 source Loopback0
Type escape sequence to abort.
Sending 100, 1500-byte ICMP Echos to 10.0.1.1, timeout is 2
seconds:
Packet sent with a source address of 10.7.10.1
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Success rate is 100 percent (100/100), round-trip min/avg/max
= 1/1/4 ms
```

After applying traffic shaping:

```
CE1# ping 10.0.1.1 repeat 100 size 1500 source Loopback0
Type escape sequence to abort.
Sending 100, 1500-byte ICMP Echos to 10.0.1.1, timeout is 2
seconds:
Packet sent with a source address of 10.7.10.1
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Success rate is 100 percent (100/100), round-trip min/avg/max
= 1/185/204 ms
```

- Round-trip time should increase when concurrent flows exist (continuous ping from PE to P):

```
CE1# ping 10.0.1.1 repeat 100 size 1500 source Loopback0

Type escape sequence to abort.
Sending 100, 1500-byte ICMP Echos to 10.0.1.1, timeout is 2
seconds:
Packet sent with a source address of 10.7.10.1
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Success rate is 100 percent (100/100), round-trip min/avg/max
= 332/380/396 ms
```

- The interface statistics before applying traffic policing should be as follows:

```
RP/0/RSP0/CPU0:PE1# show interfaces gigabitEthernet 0/0/0/2
Sat Oct 15 09:32:33.649 UTC
GigabitEthernet0/0/0/2 is up, line protocol is up
  Interface state transitions: 1
  Hardware is GigabitEthernet, address is 4055.392f.40aa (bia
4055.392f.40aa)
  Internet address is 192.168.71.70/24
  MTU 1514 bytes, BW 1000000 Kbit (Max: 1000000 Kbit)
    reliability 255/255, txload 0/255, rxload 0/255
  Encapsulation ARPA,
  Full-duplex, 1000Mb/s, THD, link type is force-up
```

- The ping statistics after applying traffic policing policy on the interface should be as follows:

- The interface statistics after the traffic policing service policy is applied should be as follows:

```
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0
abort
5365 packets output, 521494 bytes, 15 total output drops
Output 0 broadcast packets, 3202 multicast packets
0 output errors, 0 underruns, 0 applique, 0 resets
0 output buffer failures, 0 output buffers swapped out
0 carrier transitions
```

Lab 3-2: Implement MPLS Support for QoS

Complete this lab activity to practice what you learned in the related module.

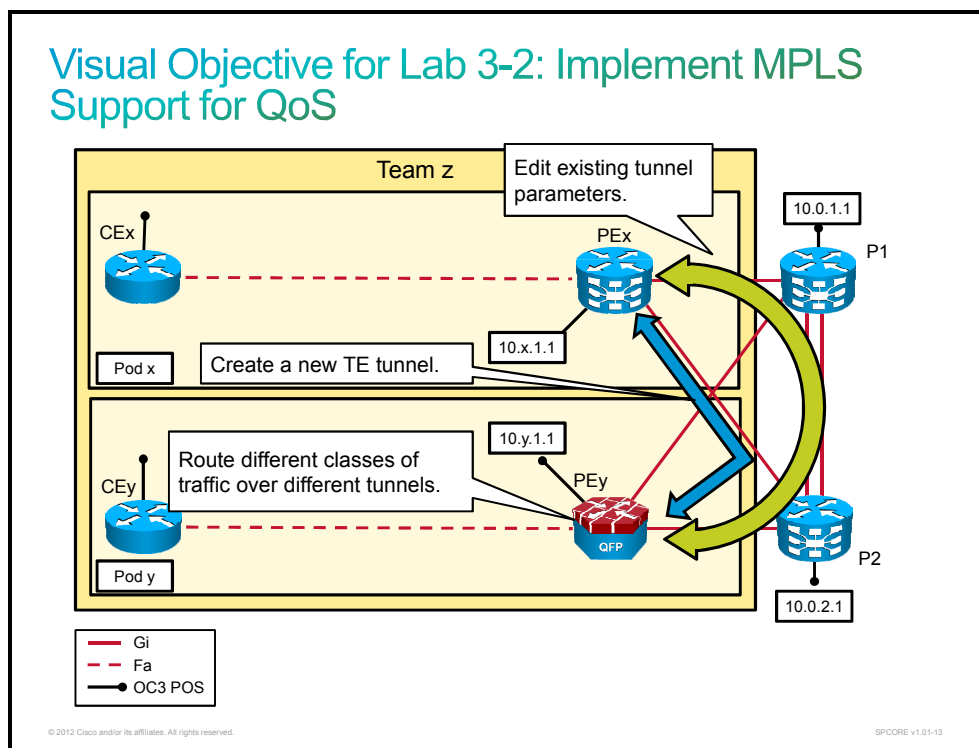
Activity Objective

In this activity, you will implement DS-TE and route different classes of traffic over different MPLS tunnels. After completing this activity, you will be able to meet these objectives:

- Update existing MPLS tunnel parameters
- Create a DS-TE tunnel
- Assign the traffic to the tunnels

Visual Objective

The figure illustrates what you will accomplish in this activity.



Required Resources

These are the resources and equipment that are required to complete this activity:

- A PC with access to the Internet
- An SSH client installed on the PC

Command List

Cisco IOS and IOS XE Commands

Command	Description
access-list <i>number</i> {deny permit} <i>protocol</i> source <i>source-wildcard</i> <i>destination destination-wildcard</i> [dscp <i>dscp-value</i>]	Defines an extended access list
configure terminal	Enters configuration mode
clear counters <i>interface</i>	Clears show interface counters and associated statistics on the specified interface
enable	Enters privileged EXEC mode
interface <i>interface</i>	Enters interface configuration mode
ip explicit-path { name <i>word</i> identifier <i>number</i> } [enable disable]	Enters the command mode for IP explicit paths and creates or modifies the specified path
ip rsvp bandwidth [<i>interface-bandwidth</i>] [<i>single-flow-bandwidth</i>] [sub-pool <i>bandwidth</i>]	Enables RSVP for IP on an interface
ip unnumbered <i>interface</i>	Specifies the MPLS TE tunnel IP address of an existing interface
ip policy route-map <i>route-map-name</i>	Applies route map policy to an interface (interface configuration mode)
match ip address <i>access-list</i>	Matches the condition in a route map in order to route packets based on set criteria
mpls traffic-eng tunnels	Enables MPLS TE tunnel signaling on an interface
next-address <i>ip-address</i>	Specifies the next IP address in the explicit path
tunnel destination <i>ip_address</i>	Sets the destination IP address of the tunnel
tunnel mode mpls traffic-eng	Sets the mode of a tunnel to MPLS for traffic engineering
tunnel mpls traffic-eng autoroute announce	Specifies that the IGP should use the tunnel (if the tunnel is up) in its enhanced SPF calculation
tunnel mpls traffic-eng bandwidth <i>kbps</i>	Configures the bandwidth required for an MPLS TE tunnel
tunnel mpls traffic-eng path-option { <i>number</i> { dynamic explicit { identifier <i>path-number</i> name <i>path-name</i> }}	Configures the explicit or dynamic path option for an MPLS TE tunnel
tunnel mpls traffic-eng priority <i>setup-priority</i> [<i>hold-priority</i>]	Configures the setup and reservation priority for MPLS TE tunnel

Command	Description
ping <i>destination_address</i> size <i>size_of_packet</i> count <i>packets</i> source <i>interface</i>	Pings destination address and specifies packet size, number of packets sent to destination, and source interface
route-map <i>map-tag</i> [permit deny] [<i>sequence-number</i>]	Defines conditions for packet routing using route maps
set interface <i>type number</i>	Sets route destination in route map configuration mode
show access-lists	Displays all access lists and their parameters
show interfaces <i>interface</i>	Displays statistics of an interface
show ip interface brief	Displays summarized IP interface status
show ip rsvp interface	Displays information related to RSVP
show route-map	Displays the contents of configured route maps
show running-config interface <i>interface</i>	Displays the running configuration on an interface

Cisco IOS XR Commands

Command	Description
autoroute announce	Specifies that the IGP should use the tunnel (if the tunnel is up) in its enhanced SPF calculation
signaled-bandwidth <i>bandwidth</i>	Configures required bandwidth reservation for the tunnel
commit	Commits the changes in the running configuration
configure terminal	Enters configuration mode
clear counters <i>interface</i>	Clears show interface counters and associated statistics on specified interface
destination <i>IP_address</i>	Configures the destination address of a tunnel interface (interface tunnel configuration mode)
enable	Enters privileged EXEC mode
explicit-path { identifier <i>path-number</i> name <i>path-name</i> }	Creates or modifies the specified path (explicit)
index <i>index</i> next-address <i>IP_address</i>	Inserts explicit path next addresses
interface <i>interface</i>	Enters interface configuration mode
ipv4 unnumbered <i>interface-name</i>	Specifies the MPLS TE tunnel IPv4 address
mpls traffic-eng interface	Enables MPLS TE on an interface and enters MPLS traffic engineering interface submenu
path-option <i>number</i> { dynamic explicit { <i>name-pathname</i> <i>path-number</i> }}	Configures the path option for an MPLS TE tunnel, explicitly defined or via a routing protocol

Command	Description
ping <i>destination_address</i> size <i>size_of_packet</i> count <i>packets</i> source <i>interface</i>	Pings a destination address and specifies packet size, number of packets sent to destination, and source interface
priority <i>setup-priority</i> <i>hold-priority</i>	Configures the setup and reservation priority for an MPLS TE tunnel
rsvp	Enters RSVP configuration submode
show interfaces <i>interface</i>	Displays statistics of an interface
show ip interface brief	Displays summarized IP interface status
show mpls traffic-eng tunnels <i>number</i>	Displays information about a specific tunnel
show running-config interface <i>interface</i>	Displays running configuration on an interface
show rsvp interface <i>type number</i>	Displays information about all interfaces with RSVP enabled

Task 1: Update Existing MPLS Tunnel Parameters

In this task, you will update existing tunnel parameters in order to prepare the tunnel for DS-TE transport.

Activity Procedure

Complete these steps:

- Step 1** Log into the PEx and PEy routers and enter global configuration mode.
- Step 2** Verify that the existing TE Tunnel1 has an explicit path defined (PEx – P1 – P2 – PEy or PEy – P2 – P1 – PEx) and that the tunnel is in the up state.
- Step 3** Update the setup and hold priority of the existing Tunnel1 to 0.
- Step 4** Change the bandwidth requirement for TE Tunnel1 to 40 kb/s.
- Step 5** Remove backup MPLS TE Tunnel 1000 from your PE router.

Activity Verification

You have completed this task when you attain these results:

Note	Outputs in the verification section are taken from team 1 and pod 1 and may differ from your outputs.
-------------	---

- When showing current status of the MPLS TE tunnel, your output should look similar to the following:

```
RP/0/RSP0/CPU0:PE1# show mpls traffic-eng tunnels 1
Thu Dec  1 11:54:23.054 UTC
Signalling Summary:
      LSP Tunnels Process:  running
      RSVP Process:       running
      Forwarding:         enabled
      Periodic reoptimization: every 3600 seconds, next
in 1778 seconds
      Periodic FRR Promotion: every 300 seconds, next in
115 seconds
      Auto-bw enabled tunnels: 0 (disabled)

Name: tunnel-te1  Destination: 10.2.1.1
Status:
  Admin:    up Oper:    up  Path:  valid  Signalling:
connected

      path option 1,  type explicit Explicit (Basis for Setup,
path weight 4)
      G-PID: 0x0800 (derived from egress interface properties)
      Bandwidth Requested: 40 kbps  CT0
      Creation Time: Wed Nov 30 14:37:34 2011 (21:16:49 ago)
Config Parameters:
  Bandwidth:      40 kbps (CT0) Priority:  0  0 Affinity:
0x0/0xffff
<...output omitted...>
```

```

Path info (OSPF 1 area 0):
Hop0: 192.168.11.1
Hop1: 192.168.2.1
Hop2: 192.168.2.2
Hop3: 192.168.22.2
Hop4: 192.168.22.20
Hop5: 10.2.1.1
<...output omitted...>

PE2# show mpls traffic-eng tunnels tunnel 1
Name: PE2_t1 (Tunnel1)
Destination: 10.1.1.1
Status:
  Admin: up      Oper: up      Path: valid
Signalling: connected
  path option 1, type explicit Explicit (Basis for Setup,
path weight 4)

Config Parameters:
  Bandwidth: 40      kbps (Global)  Priority: 0  0
Affinity: 0x0/0xFFFF
<...output omitted...>
  Explicit Route: 192.168.22.2 192.168.2.2 192.168.2.1
192.168.11.1 192.168.11.10 10.1.1.1
<...output omitted...>

```

Task 2: Create a DS-TE Tunnel

In this task, you will update the existing tunnel with another explicit path and create another tunnel. The two created tunnels will be used later for transport of two different classes of traffic.

Activity Procedure

Complete these steps:

- Step 1** Create a new explicit path named **Over_P2** on your PE router. The path will use the PE-x – P2 – PEy (PE-y – P1 – PEx) hops respectively.
- Step 2** On the PE router, create a new TE tunnel:
 - Specify Loopback0 as a source IP address of the tunnel.
 - Configure a tunnel destination to the Loopback0 address of the PE-y (PE-x) router.
 - Enable tunnel mode MPLS TE for the Cisco IOS XE router.
 - Set the setup priority value and hold priority value of the tunnel to 1.
 - Set the bandwidth of the tunnel to 50 kb/s.
 - Add the interface to the IGP routing table in order to enable traffic forwarding along the tunnel.
- Step 3** Verify the status of the configured tunnel using **show** commands.

Activity Verification

You have completed this task when you attain these results:

Note	Outputs in the verification section are taken from team 1 and pod 1 and may differ from your outputs.
-------------	---

- When showing the configured path, your output should be similar to the following:

```
RP/0/RSP0/CPU0:PE1# show explicit-paths
```

```
Thu Dec  1 13:56:06.483 UTC
```

```
Path Over_P2      status enabled
```

```
    0x1:  next-address strict 10.0.2.1
```

```
    0x2:  next-address strict 10.2.1.1
```

```
Path Explicit      status enabled
```

```
    0x1:  next-address strict 10.0.1.1
```

```
    0x2:  next-address strict 10.0.2.1
```

```
    0x3:  next-address strict 10.2.1.1
```

```
PE2#sh ip explicit-paths
```

```
PATH Explicit (strict source route, path complete, generation 5)
```

```
  1: next-address 10.0.2.1
```

```
  2: next-address 10.0.1.1
```

```
  3: next-address 10.5.1.1
```

```
PATH Over_P2 (strict source route, path complete, generation 19)
```

```
  1: next-address 10.0.2.1
```

```
  2: next-address 10.5.1.1
```

- When showing current state of newly created MPLS TE tunnel, output should be similar to the following:

```
RP/0/RSP0/CPU0:PE1# show mpls traffic-eng tunnels 2
```

```
<...output omitted...>
```

```
Name: tunnel-te2  Destination: 10.2.1.1
```

```
Status:
```

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

```
path option 1,  type explicit Over_P2 (Basis for Setup, path weight 2)
```

```
<...output omitted...>
```

```
Path info (OSPF 1 area 0):
```

```
Hop0: 192.168.12.2
```

```
Hop1: 192.168.12.2
```

```
Hop2: 192.168.22.20
```

```
Hop3: 10.2.1.1
```

```
<...output omitted...>
```

```

PE2# show mpls traffic-eng tunnels Tunnel 2

Name: PE2_t2                                     (Tunnel2) Destination:
10.1.1.1
Status:
  Admin: up      Oper: up      Path: valid      Signalling:
connected
  path option 1, type explicit Over_P2 (Basis for Setup, path weight
4)
<...output omitted...>
  Explicit Route: 192.168.22.2 192.168.12.2 192.168.12.10 10.5.1.1
Record Route: NONE
Tspec: ave rate=50 kbits, burst=1000 bytes, peak rate=50 kbits
RSVP Resv Info:
<...output omitted...>

```

Task 3: Assign Traffic to the Tunnels (Cisco IOS XE Only)

In this task, you will implement PBR to route different classes of traffic over two paths.

Activity Procedure

Complete these steps:

- Step 1** On the PEy router, create access list **100** that matches all IP traffic with a DSCP value of EF.
- Step 2** Create route map **SelectTunnel** so that traffic matching access list 100 will go through Tunnel1 (PEy – P2 – P1 – PEx) and all other traffic will go through Tunnel2 (PEy – P2 – PEx).
- Step 3** On the PE router, assign a policy map to ingress interface from customer CE.
- Step 4** Clear counters on the Tunnel1 and Tunnel2 interfaces on the PE router.
- Step 5** Check the output packet count for the Tunnel1 and Tunnel2 interface statistics. It should be close to zero.
- Step 6** Log into the CEx (CEy) router and ping the Loopback0 of router PE-y (PE-x) using the following extended ping parameters:
 - Ping count: 1000
 - ToS value: 184 (which equals DSCP value EF)
- Step 7** Check the output packet count for the Tunnel1 and Tunnel2 interface statistics. Write down the packet output packet count for Tunnel1 and Tunnel2.

Tunnel1: _____

Tunnel2: _____

- Step 8** Repeat Step 6 using the default ToS value.

- Step 9** Check the output packet count for the Tunnel1 and Tunnel2 interface statistics. Write down the packet output packet count for Tunnel1 and Tunnel2.

Tunnel1: _____

Tunnel2: _____

Step 10 On your PE, remove the route map **SelectTunnel** from interface GigabitEthernet 0/0/0.

Step 11 Shut down backup MPLS TE tunnel interface 2 on your PE router.

Activity Verification

You have completed this task when you attain this result:

- The configured access list should match all IP traffic with DSCP EF:

```
PE2#show access-lists 100
Extended IP access list 100
 10 permit ip any any dscp ef
```

- The route map should forward all access list 100 IP traffic on Tunnel1 and all other traffic to Tunnel2:

```
PE2# show route-map
route-map SelectTunnel, permit, sequence 10
  Match clauses:
    ip address (access-lists): 100
  Set clauses:
    interface Tunnel1
  Policy routing matches: 0 packets, 0 bytes
route-map SelectTunnel, permit, sequence 20
  Match clauses:
  Set clauses:
    interface Tunnel2
  Policy routing matches: 23 packets, 2162 bytes
route-map permitthis, permit, sequence 10
  Match clauses:
    ipv6 address matchipv6
  Set clauses:
  Policy routing matches: 0 packets, 0 bytes
```

- The route map should be applied the to CE-x (CE-y)-facing interface:

```
PE2# show running-config interface gigabitEthernet 0/0/0

interface GigabitEthernet0/0/0
 ip policy route-map SelectTunnel
end
```

- The interface statistics of Tunnel1 before applying an extended ping command should be as follows:

```
PE2# show interfaces tunnel 1
Tunnel1 is up, line protocol is up
  Hardware is Tunnel
  Interface is unnumbered. Using address of Loopback0
(10.2.1.1)
  MTU 17936 bytes, BW 100 Kbit/sec, DLY 50000 usec,
    reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation TUNNEL, loopback not set
  Keepalive not set
  Tunnel source 10.2.1.1, destination 10.1.1.1
```

```

Tunnel protocol/transport Label Switching
Tunnel transmit bandwidth 8000 (kbps)
Tunnel receive bandwidth 8000 (kbps)
Last input never, output never, output hang never
Last clearing of "show interface" counters 00:00:22
Input queue: 0/375/0/0 (size/max/drops/flushes); Total
output drops: 0
Queueing strategy: fifo
Output queue: 0/0 (size/max)
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
  0 packets input, 0 bytes, 0 no buffer
  Received 0 broadcasts (0 IP multicasts)
  0 runts, 0 giants, 0 throttles
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0
abort
  0 packets output, 0 bytes, 0 underruns
  0 output errors, 0 collisions, 0 interface resets
  0 unknown protocol drops
  0 output buffer failures, 0 output buffers swapped out

```

- The extended ping output with a ToS value of EF should be similar to the following:

```

CE2#ping
Protocol [ip]:
Target IP address: 10.1.1.1
Repeat count [5]: 1000
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Source address or interface:
Type of service [0]: 184
Set DF bit in IP header? [no]:
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 1000, 100-byte ICMP Echos to 10.1.1.1, timeout is 2
seconds:
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!
<...Output omitted...>
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!

```

- The interface statistics of Tunnel1 after applying the extended **ping** command with a ToS value of 184 should look as follows:

```

PE2# show interfaces tunnel 1
Tunnel1 is up, line protocol is up
Hardware is Tunnel

```

```

Interface is unnumbered. Using address of Loopback0
(10.2.1.1)
MTU 17936 bytes, BW 100 Kbit/sec, DLY 50000 usec,
    reliability 255/255, txload 10/255, rxload 1/255
Encapsulation TUNNEL, loopback not set
Keepalive not set
Tunnel source 10.2.1.1, destination 10.1.1.1
Tunnel protocol/transport Label Switching
Tunnel transmit bandwidth 8000 (kbps)
Tunnel receive bandwidth 8000 (kbps)
Last input never, output never, output hang never
Last clearing of "show interface" counters 00:01:38
Input queue: 0/375/0/0 (size/max/drops/flushes); Total
output drops: 0
Queueing strategy: fifo
Output queue: 0/0 (size/max)
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 4000 bits/sec, 4 packets/sec
    0 packets input, 0 bytes, 0 no buffer
    Received 0 broadcasts (0 IP multicasts)
    0 runts, 0 giants, 0 throttles
    0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0
abort
    1000 packets output, 118000 bytes, 0 underruns
    0 output errors, 0 collisions, 0 interface resets
    0 unknown protocol drops
    0 output buffer failures, 0 output buffers swapped out

```

- The output of the Tunnel2 interface statistics should look as follows:

```

PE2# show interfaces tunnel 2
Tunnel2 is up, line protocol is up
Hardware is Tunnel
Interface is unnumbered. Using address of Loopback0
(10.2.1.1)
MTU 17936 bytes, BW 100 Kbit/sec, DLY 50000 usec,
    reliability 255/255, txload 1/255, rxload 1/255
Encapsulation TUNNEL, loopback not set
Keepalive not set
Tunnel source 10.2.1.1, destination 10.1.1.1
Tunnel protocol/transport Label Switching
Tunnel transmit bandwidth 8000 (kbps)
Tunnel receive bandwidth 8000 (kbps)
Last input never, output never, output hang never
Last clearing of "show interface" counters 00:01:41
Input queue: 0/375/0/0 (size/max/drops/flushes); Total
output drops: 0
Queueing strategy: fifo
Output queue: 0/0 (size/max)
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec

```

```
0 packets input, 0 bytes, 0 no buffer
Received 0 broadcasts (0 IP multicasts)
0 runts, 0 giants, 0 throttles
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0
abort
```

```
3 packets output, 212 bytes, 0 underruns
0 output errors, 0 collisions, 0 interface resets
0 unknown protocol drops
0 output buffer failures, 0 output buffers swapped out
```

- The output of the Tunnel2 interface statistics after applying ping with the ToS byte set to the default should look as follows:

```
PE2#show interfaces tunnel 2
Tunnel2 is up, line protocol is up
  Hardware is Tunnel
  Interface is unnumbered. Using address of Loopback0
  (10.2.1.1)
  MTU 17936 bytes, BW 100 Kbit/sec, DLY 50000 usec,
    reliability 255/255, txload 2/255, rxload 1/255
  Encapsulation TUNNEL, loopback not set
  Keepalive not set
  Tunnel source 10.2.1.1, destination 10.1.1.1
  Tunnel protocol/transport Label Switching
  Tunnel transmit bandwidth 8000 (kbps)
  Tunnel receive bandwidth 8000 (kbps)
  Last input never, output never, output hang never
  Last clearing of "show interface" counters 00:03:23
  Input queue: 0/375/0/0 (size/max/drops/flushes); Total
  output drops: 0
  Queueing strategy: fifo
  Output queue: 0/0 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 1000 bits/sec, 2 packets/sec
    0 packets input, 0 bytes, 0 no buffer
    Received 0 broadcasts (0 IP multicasts)
    0 runts, 0 giants, 0 throttles
    0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0
  abort
    1007 packets output, 115972 bytes, 0 underruns
    0 output errors, 0 collisions, 0 interface resets
    0 unknown protocol drops
    0 output buffer failures, 0 output buffers swapped out
```

Lab 4-1: Implement Classification and Marking

Complete this lab activity to practice what you learned in the related module.

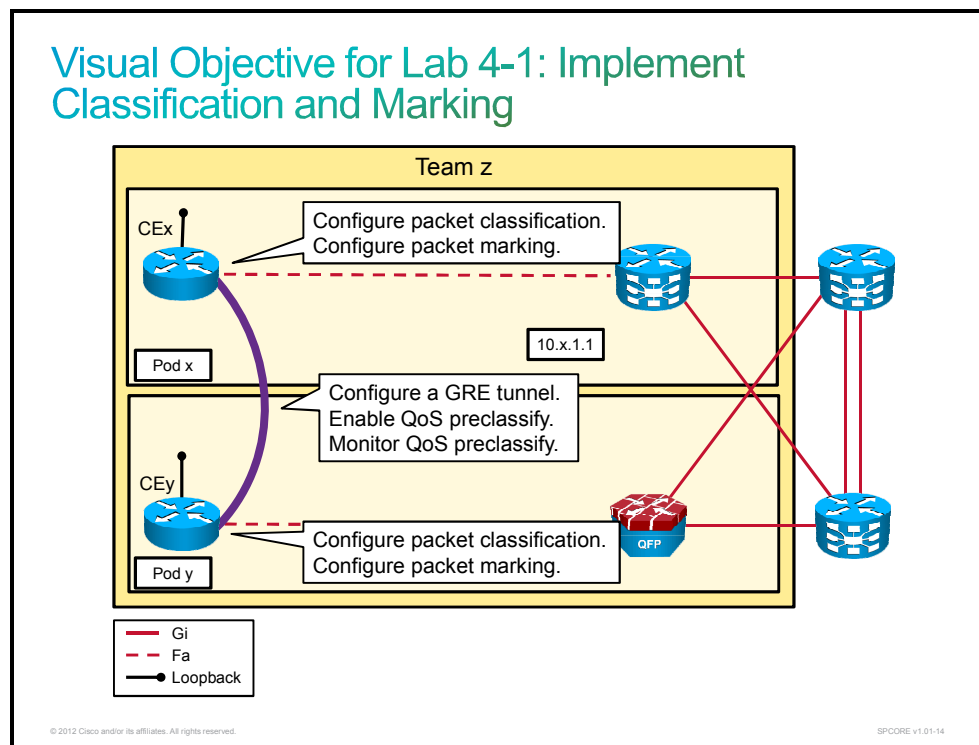
Activity Objective

In this activity, you will configure MQC classification using extended access lists and use class-based marking to mark packets. After that, you will configure a GRE tunnel and monitor the classification process before and after the QoS preclassify feature is applied. After completing this activity, you will be able to meet these objectives:

- Configure MQC classification
- Configure class-based marking
- Configure GRE tunnels
- Configure QoS preclassify
- Monitor QoS preclassify

Visual Objective

The figure illustrates what you will accomplish in this activity.



Required Resources

These are the resources and equipment that are required to complete this activity:

- A PC with access to the Internet
- An SSH client installed on the PC

Command List

Cisco IOS Commands

Command	Description
access-list <i>number</i> {deny permit} <i>protocol</i> source <i>source-wildcard</i> <i>destination destination-wildcard</i> [dscp <i>dscp-value</i>]	Defines an extended access list
configure terminal	Enters configuration mode
class <i>name_of_class</i>	Creates a class map to be used for matching packets to a specified class
class-map <i>class_name</i>	Creates a class to be used with a class map and enters class-map configuration mode. The class map is used for matching packets to the specified class.
clear counters <i>interface</i>	Clears show interface counters and associated statistics on specified interface
enable	Enters privileged EXEC mode
interface <i>interface</i>	Enters interface configuration mode
ip route <i>prefix mask</i> { <i>ip-address</i> <i>interface-type interface-number</i> [<i>ip-address</i>] }	Sets static route for specified destination
tunnel destination <i>ip_address</i>	Sets the destination IP address of the tunnel
tunnel source <i>interface</i>	Sets the source IP interface of the tunnel
permit <i>protocol source source-wildcard destination destination-wildcard</i>	Configures access list entries that permit traffic from source to destination using a specified protocol
ping <i>destination_address size size_of_packet count packets source interface</i>	Pings destination address and specifies packet size, number of packets sent to destination, and source interface
policy-map <i>policy_name</i>	Enters policy-map configuration mode and creates or modifies a policy map that can be attached to one or more interfaces
service-policy {input output} <i>policy-map-name</i>	Attaches service traffic policy to an interface and specifies the direction in which the traffic policy should be applied
set ip precedence <i>ip-precedence-value</i>	Sets the precedence value in the IP header for the class specified (policy-map class configuration mode)
show access-lists	Displays all access lists and their parameters
show interfaces <i>interface</i>	Displays statistics of an interface
show ip interface brief	Displays summarized IP interface status
qos pre-classify	Enables the QoS preclassify feature on a tunnel interface or crypto map

Task 1: Configure MQC Classification

In this task, you will classify different types of traffic by access list and destination addresses.

Activity Procedure

Complete these steps:

- Step 1** Log into the CEx (CEy) router and enter global configuration mode.
- Step 2** Create extended access list **ICMP** matching all ICMP traffic.
- Step 3** Create extended access list **GRE** matching all GRE traffic.
- Step 4** Create first traffic class **ICMP** that matches only access list ICMP.
- Step 5** Create second traffic class **GRE** that matches only access list GRE.

Activity Verification

You have completed this task when you attain these results:

Note	Outputs in the verification section are taken from team 1 and pod 1 and may differ from your outputs.
-------------	---

- Two access lists for matching GRE and ICMP traffic should be defined:

```
CE1# show access-lists
Extended IP access list GRE
  10 permit gre any any
Extended IP access list ICMP
  10 permit icmp any any
```

- When showing configured traffic classes, your output should be similar to the following:

```
CE1# show class-map

Class Map match-all ICMP (id 1)
  Match access-group name ICMP

Class Map match-any class-default (id 0)
  Match any

Class Map match-all GRE (id 2)
  Match access-group name GRE
```

Task 2: Configure Class-Based Marking

In this task, you will configure a policy map that will mark ICMP and GRE traffic with different IP precedence values. You will apply the created policy on the outgoing interface.

Activity Procedure

Complete these steps:

- Step 1** Log into the CEx and CEy router and enter global configuration mode.
- Step 2** Create policy map **MARK_GRE_ICMP**.
- Step 3** In the newly created policy, set the IP precedence value of 6 for traffic class GRE.

- Step 4** Set the IP precedence value of 5 for the traffic class ICMP.
- Step 5** Apply the created policy on the PE-facing interface in the outgoing direction.

Activity Verification

You have completed this task when you attain these results:

Note	Outputs in the verification section are taken from team 1 and pod 1 and may differ from your outputs.
-------------	---

- You should have configured a policy map with the ICMP and GRE traffic classes defined:

```
CE1# show policy-map
      Policy Map MARK_GRE_ICMP
        Class GRE
          set ip precedence 6
        Class ICMP
          set ip precedence 5
```

- The newly created policy map should be applied on PE-facing interface:

```
CE1# show policy-map interface gigabitEthernet 0/0
GigabitEthernet0/0
```

Service-policy output: MARK_GRE_ICMP

```
Class-map: GRE (match-all)
  0 packets, 0 bytes
  5 minute offered rate 0 bps, drop rate 0 bps
Match: access-group name GRE
QoS Set
  precedence 6
  Packets marked 0
```

```
Class-map: ICMP (match-all)
  0 packets, 0 bytes
  5 minute offered rate 0 bps, drop rate 0 bps
Match: access-group name ICMP
QoS Set
  precedence 5
  Packets marked 0
```

```
Class-map: class-default (match-any)
  23 packets, 2024 bytes
  5 minute offered rate 0 bps, drop rate 0 bps
Match: any
```

Task 3: Configure a GRE Tunnel

In this task, you will configure a GRE tunnel and add a static route.

Activity Procedure

Complete these steps:

- Step 1** In global configuration mode of the CEx and CEy routers, create the **Tunnel5** interface.
- Step 2** Set the IP address for the newly created tunnel to **10.xy.xy.1/24** (Pod X) or **10.xy.xy.2/24** (Pod Y).
- Step 3** Set the tunnel source of the CE router of your pod to the PE-facing interface.
- Step 4** Set the tunnel destination to the CE router of the other pod PE-facing interface of your partner pod.
- Step 5** Enter privileged EXEC mode and perform a traceroute to **10.y.10.1** (Pod Y to **10.x.10.1**). You see that traffic is not going through the Tunnel5 interface. This is because an OSPF route is defined over physical links, not through a tunnel interface.
- Step 6** Set the static route to **10.y.10.1** (Pod Y to **10.x.10.1**) through Tunnel5 interface.
- Step 7** Perform the traceroute again. You should see only one hop to the destination.

Activity Verification

You have completed this task when you attain these results:

Note	Outputs in the verification section are taken from team 1 and pod 1 and may differ from your outputs.
-------------	---

- When showing the status of IP interfaces, the Tunnel 5 interface should be in the up/up state:

```
CE1# show ip interface brief
Interface                               IP-Address   OK? Method Status
Protocol
Embedded-Service-Engine0/0             unassigned   YES  unset
administratively down down
GigabitEthernet0/0                     192.168.101.11 YES  manual  up
up
GigabitEthernet0/1                     unassigned   YES  unset  up
up
Loopback0                              10.1.10.1    YES  manual  up
up
Tunnel5                                10.12.12.1   YES  manual  up
up
```

- When performing traceroute before applying the static route, your output should be similar to the following:

```
CE2# traceroute 10.1.10.1
Type escape sequence to abort.
Tracing the route to 10.1.10.1
VRF info: (vrf in name/id, vrf out name/id)
 1 192.168.102.20 0 msec 0 msec 0 msec
 2 192.168.22.2 [MPLS: Label 16033 Exp 0] 0 msec 0 msec 0msec
```

```

3 192.168.1.1 [MPLS: Label 16031 Exp 0] 0 msec 0 msec 0 msec
4 192.168.11.10 0 msec 0 msec 0 msec
5 192.168.101.11 0 msec 0 msec

```

- When performing traceroute after applying the static route, your output should be similar to the following:

```

CE2# traceroute 10.1.10.1
Type escape sequence to abort.
Tracing the route to 10.1.10.1
VRF info: (vrf in name/id, vrf out name/id)
 1 10.12.12.1 0 msec * 0 msec

```

Task 4: Configure QoS Preclassify

In this task, you will configure the QoS preclassify feature on the GRE interface.

Activity Procedure

Complete these steps:

- Step 1** On the CE router, show the policy map statistics on the PE-facing interface and observe the following:
- Class map: ICMP
- Packets: _____
- Marked packets: _____
- Class map: GRE
- Packets: _____
- Marked packets: _____
- Step 2** Ping the destination of your partner pod Loopback 0 interface and write down the following values in the policy map statistics. Observe that the number of marked packets for class map GRE is increasing.
- Class map: ICMP
- Packets: _____
- Marked packets: _____
- Class map: GRE
- Packets: _____
- Marked packets: _____
- Step 3** On the CE router, enter Tunnel 5 configuration mode.
- Step 4** Enable the QoS preclassify feature on that interface.

Activity Verification

You have completed this task when you attain this result:

- The CE policy map statistics before pinging should look similar to the following:

```

CE1# show policy-map interface gigabitEthernet 0/0
GigabitEthernet0/0

```

Service-policy output: MARK-GRE-ICMP

Class-map: ICMP (match-all)

```
0 packets, 0 bytes
5 minute offered rate 0 bps, drop rate 0 bps
Match: protocol icmp
QoS Set
  precedence 5
  Packets marked 0
```

Class-map: GRE (match-all)

```
0 packets, 0 bytes
5 minute offered rate 0 bps, drop rate 0 bps
Match: protocol gre
QoS Set
  precedence 6
  Packets marked 0
```

Class-map: class-default (match-any)

```
57 packets, 5461 bytes
5 minute offered rate 0 bps, drop rate 0 bps
Match: any
```

- Pings to the other pod CE router loopback address should be successful:

```
CE1# ping 10.2.10.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.2.10.1, timeout is 2
seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max =
1/6/20 ms
```

- The CE policy map statistics after pinging should look similar to the following:

```
CE1# show policy-map interface gigabitEthernet 0/0
GigabitEthernet0/0
```

Service-policy output: MARK-GRE-ICMP

Class-map: ICMP (match-all)

```
0 packets, 0 bytes
5 minute offered rate 0 bps, drop rate 0 bps
Match: protocol icmp
QoS Set
  precedence 5
  Packets marked 0
```

Class-map: GRE (match-all)

```
5 packets, 620 bytes
5 minute offered rate 0 bps, drop rate 0 bps
Match: protocol gre
```

```
QoS Set
  precedence 6
  Packets marked 5
```

```
Class-map: class-default (match-any)
  166 packets, 15308 bytes
  5 minute offered rate 0 bps, drop rate 0 bps
  Match: any
```

- Your tunnel interface should have the QoS preclassify feature enabled:

```
CE1# show interfaces tunnel 5
Tunnel5 is up, line protocol is up
  Hardware is Tunnel
  Internet address is 10.12.12.1/24
  MTU 17916 bytes, BW 100 Kbit/sec, DLY 50000 usec,
    reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation TUNNEL, loopback not set
  Keepalive not set
  Tunnel source 192.168.101.11 (GigabitEthernet0/0),
  destination 192.168.102.21
  Tunnel Subblocks:
    src-track:
      Tunnel5 source tracking subblock associated with
      GigabitEthernet0/0
      Set of tunnels with source GigabitEthernet0/0, 2
      members (includes iterators), on interface <OK>
  Tunnel protocol/transport GRE/IP
  Key disabled, sequencing disabled
  Checksumming of packets disabled
  Tunnel TTL 255, Fast tunneling enabled
  Tunnel transport MTU 1476 bytes
  Tunnel transmit bandwidth 8000 (kbps)
  Tunnel receive bandwidth 8000 (kbps)
  Last input never, output never, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output
  drops: 0
  Queueing strategy: fifo (QOS pre-classification)
  Output queue: 0/0 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
    0 packets input, 0 bytes, 0 no buffer
  Received 0 broadcasts (0 IP multicasts)
    0 runts, 0 giants, 0 throttles
    0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0
  abort
    0 packets output, 0 bytes, 0 underruns
    0 output errors, 0 collisions, 0 interface resets
    0 unknown protocol drops
    0 output buffer failures, 0 output buffers swapped out
```

Task 5: Monitor QoS Preclassify

In this task, you will monitor the configured QoS preclassify feature on the tunnel interface.

Activity Procedure

Complete these steps:

- Step 1** On the CE router, ping the CE Loopback0 IP address of the other pod.
- Step 2** Write down the following values in the PE-facing interface policy map statistics. Observe that the number of marked packets for class map ICMP is increasing.
- Class map: ICMP
Packets: _____
Marked packets: _____
- Class map: GRE
Packets: _____
Marked packets: _____
- Step 3** Remove the static route pointing to interface Tunnel 5.

Activity Verification

You have completed this task when you attain this result:

- Pings to the other pod CE router loopback address should be successful:

```
CE1# ping 10.2.10.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.2.10.1, timeout is 2
seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max =
1/6/20 ms
```

- The statistics of the PE-facing interface after applying the QoS preclassify feature should be similar to the following:

```
CE1# show policy-map interface gigabitEthernet 0/0
GigabitEthernet0/0
```

Service-policy output: MARK-GRE-ICMP

```
Class-map: ICMP (match-all)
  5 packets, 500 bytes
  5 minute offered rate 0 bps, drop rate 0 bps
  Match: protocol icmp
  QoS Set
    precedence 5
    Packets marked 5

Class-map: GRE (match-all)
  5 packets, 620 bytes
  5 minute offered rate 0 bps, drop rate 0 bps
  Match: protocol gre
```

```
QoS Set
  precedence 6
  Packets marked 5
```

```
Class-map: class-default (match-any)
  166 packets, 15308 bytes
  5 minute offered rate 0 bps, drop rate 0 bps
  Match: any
```

Lab 5-1: Implement Queuing

Complete this lab activity to practice what you learned in the related module.

Activity Objective

In this lab activity, you will configure and monitor class-based policing.

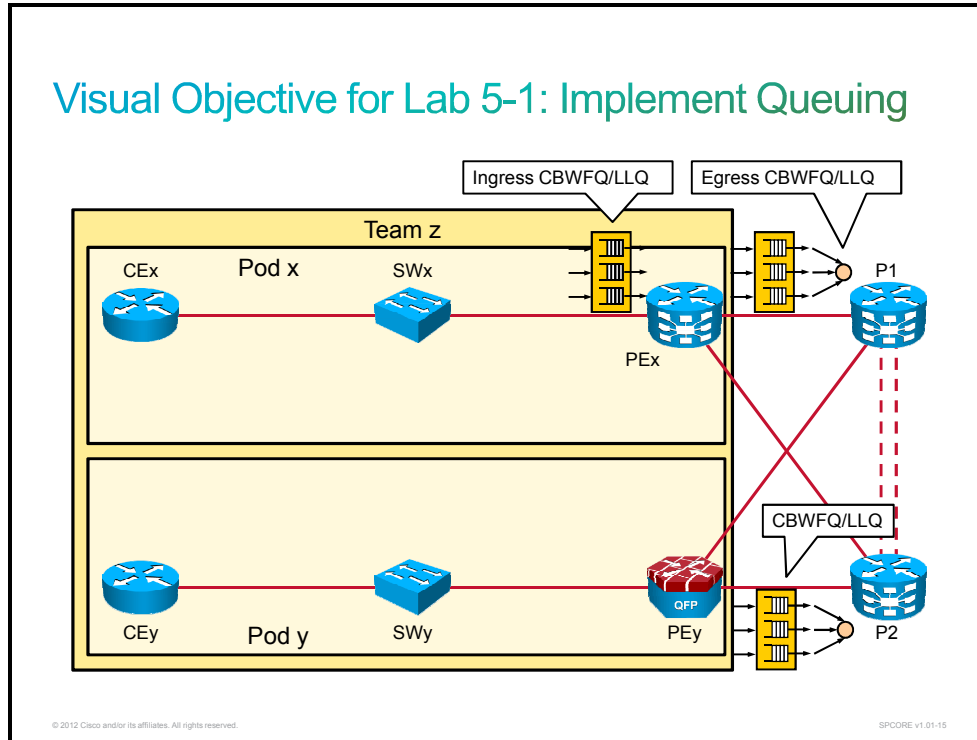
Note Students from two different pods will work in one team. The CE routers in both pods are running Cisco IOS Software. The first pod in the team will work on the PE router that is running Cisco IOS XR Software, and the second pod in the team will work on the PE router that is running Cisco IOS XE Software. Students in the same team should coordinate their lab activity.

In the lab activity, you will work on different Cisco routers running Cisco IOS (c2900), Cisco IOS XE (asr1001), and Cisco IOS XR (asr9k) Software. After completing this activity, you will be able to meet these objectives:

- Configure CBWFQ on Cisco IOS XR and Cisco IOS XE Software
- Monitor CBWFQ on Cisco IOS XR and Cisco IOS XE Software
- Configure LLQ on Cisco IOS XR and Cisco IOS XE Software
- Monitor LLQ on Cisco IOS XR and Cisco IOS XE Software

Visual Objective

The figure illustrates what you will accomplish in this activity.



Command List

The table describes the commands that are used in this lab activity.

Cisco IOS, IOS XE, and IOS XR Policing Commands

Command	Description
<code>class-map</code>	Defines a class map
<code>policy-map</code>	Defines a policy map
<code>class</code>	Defines handling for a traffic class within a policy map
<code>bandwidth</code>	Configures a bandwidth guarantee for a traffic class
<code>service-policy</code>	Applies a policy map to an interface or a child-level policy map to the parent level
<code>police</code>	Sets policing parameters
<code>shape average</code>	Configures shaping to the average rate
<code>conform-action</code>	Defines the conform action of a policer
<code>exceed-action</code>	Defines the exceed action of a policer
<code>violate-action</code>	Defines the violate action of a policer
<code>show policy-map interface</code>	Displays the statistics of a service policy
<code>clear qos counters interface</code>	Clears QoS interface counters in IOS XR Software
<code>clear counters</code>	Clears QoS interface counters in IOS or IOS XE Software

Task 1: Configure IP SLA

In this task, you will configure IP SLA.

Activity Procedure

Complete these steps:

- Step 1** Make sure that your nearest core router is configured as the NTP master using the **ntp master** command.
- Step 2** Configure your CE router as the NTP client of the core router. Use the **ntp server core-router-address** command.
- Step 3** On your nearest P router, make sure that the SLA responder is enabled for the router loopback address, UDP port 3000.

P1:

```
ipsla
 responder
  type udp ipv4 address 10.0.1.1 port 3000
!
commit
```

P2:

```
ipsla
 responder
  type udp ipv4 address 10.0.2.1 port 3000
!
commit
```

- Step 4** On your CE router, configure an IP SLA probe, with a timeout and threshold of 1 millisecond and frequency of 2 seconds. Calculate the ToS value for the DSCP EF (DSCP decimal value 46). What is the ToS value?

CE1:

```
ip sla 1
 udp-jitter 10.0.1.1 3000
 timeout 5000
 threshold 1
 frequency 5000
 tos 184
```

CE2:

```
ip sla 1
 udp-jitter 10.0.2.1 3000
 timeout 5000
 threshold 1
 frequency 5000
 tos 184
```

- Step 5** Schedule the measurement to start immediately and run repetitively on your CE router.

```
ip sla schedule 1 start-time now recurring
!
```

- Step 6** View the SLA statistics using the **show ip sla statistics 1** command. You should see no failures.

Note In this lab environment, where there is plenty of bandwidth and little traffic, you may examine the IP SLA results by configuring a traffic throttle and flooding the network with continuous pings. The IP SLA tool is not well-suited to testing the queuing configuration in this environment.

Activity Verification

You have completed this task when you attain these results:

- You have deployed the IP SLA tool for measuring network performance.
- You have verified the SLA statistics:

```
CE1# show ip sla statistics 1
Dec  5 11:24:20.111: %SYS-5-CONFIG_I: Configured from console by
console
CE1#show ip sla statistics 1
IPSLAs Latest Operation Statistics

IPSLA operation id: 1
Type of operation: udp-jitter
    Latest RTT: 1 milliseconds
Latest operation start time: 11:23:55 UTC Mon Dec 5 2011
Latest operation return code: OK
RTT Values:
    Number Of RTT: 10                      RTT Min/Avg/Max: 1/1/1
milliseconds
Latency one-way time:
    Number of Latency one-way Samples: 0
    Source to Destination Latency one way Min/Avg/Max: 0/0/0
milliseconds
    Destination to Source Latency one way Min/Avg/Max: 0/0/0
milliseconds
Jitter Time:
    Number of SD Jitter Samples: 0
    Number of DS Jitter Samples: 0
    Source to Destination Jitter Min/Avg/Max: 0/0/0 milliseconds
    Destination to Source Jitter Min/Avg/Max: 0/0/0 milliseconds
Packet Loss Values:
    Loss Source to Destination: 0
    Source to Destination Loss Periods Number: 0
    Source to Destination Loss Period Length Min/Max: 0/0
    Source to Destination Inter Loss Period Length Min/Max: 0/0
    Loss Destination to Source: 0
```

Task 2A: Configure CBWFQ on Cisco IOS XR Software

In this task, you will configure and verify CBWFQ on the PEx router running Cisco IOS XR Software.

Activity Procedure

Complete these steps:

Step 1 On your PEx router, configure three traffic classes, matched by IPv4 access lists and DSCP values:

- Class **External** matching **External-podX-ACL** that permits ICMPv4 traffic sourced from your CE router uplink GigabitEthernet interface.
- Class **Internal** matching **Internal-podX-ACL** that permits ICMPv4 traffic sourced from your CE router Loopback 0 interface.
- Class **ef-class** matching all traffic with DSCP set to the EF class.

Note Internal traffic will get a higher bandwidth guarantee than external traffic.

Step 2 On your PEx router, configure a policy map named **cbwfq-policy**. Configure CBWFQ using bandwidth guarantees for two traffic classes:

- **Internal:** 60 percent available bandwidth
- **ef-class:** 35 percent available bandwidth

Note You will see that the Internal class gets more bandwidth than the External class. You do not explicitly configure guaranteed bandwidth for the External class because the minimum configurable bandwidth on Cisco IOS XR Software is 64 kb/s. In this lab environment, you cannot generate that much traffic.

Step 3 On your PEx router, throttle the traffic toward the core to the rate of 200 kb/s. Apply the CBWFQ policy to the throttled traffic.

```
policy-map shape-all
class class-default
  service-policy cbwfq-policy
  shape average 200 kbps
commit
```

Note You will learn about traffic shaping and hierarchical policies in the next module. The throttle is needed in this lab environment to verify the operation of the queuing system.

Step 4 On your PEx router, apply the parent policy (**shape-all**) in the output direction on the core interface.

Step 5 On your PEx router, verify the policy map operation on the core-facing interface. You should see that the number of packets in the ef-class is increasing.

Step 6 Prepare a second terminal session to your CEx router. You may use Telnet to connect to your CEx router from your switch, a P router, or the PEx router. In each session, launch an ICMP echo flood to your nearest P router. One flood should be sourced from the loopback address and the other from the uplink address. Use the commands **ping 10.0.1.1 size 1000 repeat 1000 source loopback 0** and **ping**

10.0.1.1 size 1000 repeat 1000, respectively. You should see that the loopback-sourced flood gets more bandwidth.

Note You must ping your nearest P router. Otherwise, the traffic on the PE core interface will be labeled and the class maps will not match properly.

Step 7 On the PE router, verify the policy map statistics again. You should see that the number of packets in the Internal class has increased.

Note If you examine the policy map statistics while the flood is under way, you will see the corresponding traffic rate in kilobits per second.

Hierarchical CBWFQ

Step 8 Define a policy map (**cbwfq-child**) that allocates the respective amounts of remaining bandwidth:

- Class Internal: 80 percent
- Class External: 20 percent

Step 9 Define two traffic classes that match respective DSCP values:

- Mission-critical: af21, af22, af23, or cs2
- Bulk: af11, af12, af13, cs1, or default

Step 10 Define a policy map (**cbwfq-parent**) that contains two classes with these settings:

- Class Mission-critical:
 - Minimum bandwidth guarantee: 30 percent
 - Applied child policy: cbwfq-child
- Class Bulk
 - Minimum bandwidth guarantee: 40 percent
 - Applied child policy: cbwfq-child

Step 11 Apply the parent policy (cbwfq-parent) to the CE-facing interface in the input direction.

Step 12 Launch the ICMP floods at the same time and check the operation of the service policy. You will not see any drops, but you should see that the respective child classes match traffic.

Step 13 Remove the output and input service policies from PEx.

Activity Verification

You have completed this task when you attain these results:

- You have configured CBWFQ.
- On your PEx router, verify the policy map operation on the core-facing interface. You should see that the number of packets in the ef-class is increasing:

```
RP/0/RSP0/CPU0:PE1# show policy-map interface GigabitEthernet 0/0/0/2
Mon Dec  5 13:53:38.750 UTC
GigabitEthernet0/0/0/2 direction input: Service Policy not installed
```


- Launch the ICMP floods again and check the operation of the service policy. You will not see any drops, but you should see that the respective child classes match traffic.

```
RP/0/RSP0/CPU0:PE1# show policy-map interface gigabitEthernet 0/0/0/0
Mon Dec  5 14:33:22.413 UTC
```

```
GigabitEthernet0/0/0/0 input: cbwfq-parent
```

<...output omitted...>

```
Policy cbwfq-child Class External
Classification statistics          (packets/bytes)      (rate -
kbps)
    Matched                      :          1000/1018000
0
    Transmitted                  :          1000/1029000
0
    Total Dropped                :              0/0
0
Queueing statistics
    Queue ID                     : 268437387
    High watermark (Unknown)
    Inst-queue-len (packets)      : 0
    Avg-queue-len (Unknown)
    Taildropped (packets/bytes)   : 0/0
    Queue (conform)              : 1000/1029000
0
    Queue (exceed)               :              0/0
0
    RED random drops (packets/bytes) : 0/0
```

<...output omitted...>

- You have configured fabric CBWFQ.
- Remove the output and input service policies from PEx.

Task 2B: Configure CBWFQ on Cisco IOS XE Software

In this task, you will configure and verify CBWFQ on the PEx router running Cisco IOS XE Software.

Activity Procedure

Complete these steps:

- Step 14** On your PEx router, configure three traffic classes, matched by IPv4 access lists and DSCP values:
- Class **External** matching **External-podX-ACL** that permits ICMPv4 traffic sourced from your CE router uplink GigabitEthernet interface.
 - Class **Internal** matching **Internal-podX-ACL** that permits ICMPv4 traffic sourced from your CE router Loopback0 interface.

- Class **ef-class** matching all traffic with DSCP set to the EF class.

Note Internal traffic will get higher bandwidth guarantee than external traffic.

Step 15 On your PE router, configure a policy map, named **cbwfq-policy**. Configure CBWFQ using bandwidth guarantees for two traffic classes:

- Internal: 95 percent available bandwidth
- External: 2 percent available bandwidth
- ef-class: 2 percent available bandwidth

Note You will see that the Internal class gets more bandwidth than the External class.

Step 16 On your PEy router, throttle the traffic toward the core to the rate of 100 kb/s. Apply the CBWFQ policy to the throttled traffic.

```
policy-map shape-all
  class class-default
    shape average 100000
  service-policy cbwfq-policy
```

Note You will learn about traffic shaping and hierarchical policies in the next module. The throttle is needed in this lab environment to verify the operation of the queuing system.

Step 17 On your PEy router, apply the parent policy (shape-all) in the output direction on the core interface.

Step 18 On your PEy router, verify the policy map operation on the core-facing interface. You should see that the number of packets in the ef-class is increasing.

Step 19 Prepare a second terminal session to your CEy router. You may use Telnet to connect to your CEy router from your switch, a P router, or the PEy router. In each session, launch an ICMP echo flood. One flood should be sourced from the loopback address and the other from the uplink address. Use commands **ping 10.0.2.1 size 1000 repeat 1000 source loopback 0** and **ping 10.0.2.1 size 1000 repeat 1000**, respectively. You should see that the loopback-sourced flood gets more bandwidth.

Step 20 Remove the shape-all parent policy map from the core-facing interface of PEy.

Activity Verification

You have completed this task when you attain these results:

- On your PE router, verify the policy map operation on the core-facing interface. You should see that the number of packets in the ef-class is increasing.
- Perform an ICMP ping from your CEx router to the loopback IP address of P1 router. One flood should be sourced from the loopback address and the other from the uplink address. Pings should be performed at the same time. You should see that the loopback-sourced flood gets more bandwidth:

```
CE2# ping 10.0.2.1 size 1000 repeat 1000 source loopback 0
Type escape sequence to abort.
Sending 1000, 1000-byte ICMP Echos to 10.0.2.1, timeout is 2 seconds:
Packet sent with a source address of 10.2.10.1
```



```

Class-map: Internal (match-any)
  1000 packets, 2028000 bytes
  5 minute offered rate 16000 bps, drop rate 0000 bps
  Match: access-group name Internal-podY-ACL
  Queueing
    queue limit 64 packets
    (queue depth/total drops/no-buffer drops) 0/0/0
    (pkts output/bytes output) 2000/2028000
    bandwidth 95% (95 kbps)

Class-map: External (match-any)
  1000 packets, 2028000 bytes
  5 minute offered rate 28000 bps, drop rate 0000 bps
  Match: access-group name External-podY-ACL
  Queueing
    queue limit 64 packets
    (queue depth/total drops/no-buffer drops) 0/0/0
    (pkts output/bytes output) 2000/2028000
    bandwidth 2% (2 kbps)
<...output omitted...>

```

Task 3A: Configure LLQ on Cisco IOS XR Software

In this task, you will configure and verify LLQ on the PEx router running Cisco IOS XR Software.

Activity Procedure

Complete these steps:

- Step 1** On your PEx router, create a traffic class named **Top-priority-prec** that matches precedence 5.
- Step 2** On your PEx router, configure a policy map named **llq-policy**. Configure LLQ with this behavior for the traffic classes:
 - Internal: Priority level 1, policing rate 20 percent, maximum queue size 20 ms (higher than the default 10 ms for priority queues)
 - External: Guaranteed 20 percent of available bandwidth, maximum queue size 50 ms (lower than the default 100 ms for regular classes)
 - ef-class: Priority level 1, policing rate 10 percent
 - Top-priority-prec: Priority level 2, policing rate 5 percent

Note Cisco IOS XR Software requires that priority queues are policed. You will learn about policing and implement this technique in the next module.

- Step 3** On your PEx router, apply the LLQ policy in the outbound direction on the core uplink and in the ingress direction on the customer-facing link.
- Step 4** On your PEx router, verify the policy map operation on the core-facing interface. You should see that the number of packets in the ef-class within the llq-policy is increasing.

- Note** Ideally, this task should demonstrate that priority traffic receives better treatment. In this environment, it is impossible to saturate the queues with enough traffic for that purpose.

- ## Verification

- Perform an ICMP ping from your CEx router to the loopback IP address of the P1 router. One flood should be sourced from the loopback address and the other from the uplink address:

Systems, Inc. Lab Guide 75

[illegible]

- You should see that the counters in the Internal and External class have increased:

```
RP/0/RSP0/CPU0:PE1#sh policy-map interface gigabitEthernet 0/0/0/2
```

Tue Dec 6 11:08:35.428 UTC

```
GigabitEthernet0/0/0/2 direction input: Service Policy not installed
```

GigabitEthernet0/0/0/2 output: llq-policy

Class Internal

Classification statistics		(packets/bytes)	(rate - kbps)
Matched	:	4000/4072000	0
Transmitted	:	4000/4072000	0
Total Dropped	:	0/0	0
Policing statistics		(packets/bytes)	(rate - kbps)
Policed(conform)	:	4000/4072000	0
Policed(exceed)	:	0/0	0
Policed(violate)	:	0/0	0
Policed and dropped	:	0/0	
Queueing statistics			
Queue ID	:	264	
High watermark	(Unknown)		
Inst-queue-len	(packets)	:	0
Avg-queue-len	(Unknown)		
Taildropped	(packets/bytes)	:	0/0
Queue(conform)	:	4000/4072000	0
Queue(exceed)	:	0/0	0
RED random drops	(packets/bytes)	:	0/0

Class External

Classification statistics	(packets/bytes)	(rate - kbps)
Matched :	4000/4072000	0
Transmitted :	4000/4072000	0
Total Dropped :	0/0	0
Queueing statistics		
Queue ID	: 266	
High watermark (Unknown)		
Inst-queue-len (packets)	: 0	

```

Avg-queue-len      (Unknown)
Tailedropped (packets/bytes)      : 0/0
Queue (conform)      :      4000/4072000      0
Queue (exceed)      :      0/0      0
RED random drops (packets/bytes) : 0/0
<...output omitted...>

```

Task 3B: Configure LLQ on Cisco IOS XE Software

In this task, you will configure and verify LLQ on the PEy router running Cisco IOS XE Software.

Activity Procedure

Complete these steps:

- Step 1** On your PEy router, configure a policy map named **llq-policy**. Configure LLQ with this behavior for the traffic classes:
 - Internal: Priority queue, 20 percent bandwidth restriction
 - External: No priority, 30 percent bandwidth guarantee
 - ef-class: No priority, 20 percent bandwidth guarantee
- Step 2** On your PEy router, apply the LLQ policy in the outbound direction on the core uplink.
- Step 3** On your PEy router, verify the policy map operation on the core-facing interface. You should see that the number of packets in the ef-class within the llq-policy is increasing.
- Step 4** From your CE router, flood the network with ICMP echo packets sourced from the uplink address (**ping 10.0.2.1 size 1000 repeat 1000 source loopback 0**) and verify the policy statistics on the PE router. You should see that the counters in the Internal class have increased.
- Step 5** From your CE router, flood the network with ICMP echo packets sourced from the uplink address (**ping 10.0.2.1 size 1000 repeat 1000**) and verify the policy statistics on the PE router. You should see that the counters in the External class have increased.
- Step 6** Remove all output service policies from PEy.
- Step 7** Remove the shape-all policy map.

Activity Verification

You have completed this task when you attain these results:

- Perform an ICMP ping from your CEx router to the loopback IP address of P1 router. One flood should be sourced from the loopback address and the other from the uplink address:

```

CE2# ping 10.0.2.1 size 1000 repeat 1000 source loopback 0
Type escape sequence to abort.
Sending 1000, 1000-byte ICMP Echos to 10.0.1.1, timeout is 2 seconds:
Packet sent with a source address of 10.1.10.1
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

```


Priority: 20% (200000 kbps), burst bytes 5000000, b/w exceed
drops: 0

```
Class-map: External (match-any)
  1000 packets, 1014000 bytes
  5 minute offered rate 6000 bps, drop rate 0000 bps
  Match: access-group name External-podY-ACL
  Queueing
  queue limit 1249 packets
  (queue depth/total drops/no-buffer drops) 0/0/0
  (pkts output/bytes output) 1000/1014000
  bandwidth 30% (300000 kbps)
```

<...output omitted...>

Note	Ideally, this task should demonstrate that priority traffic receives better treatment. In this environment, it is impossible to saturate the queues with enough traffic for that purpose.
-------------	---

Lab 5-2: Implement Congestion Avoidance

Complete this lab activity to practice what you learned in the related module.

Activity Objective

In this lab activity, you will configure and monitor class-based policing.

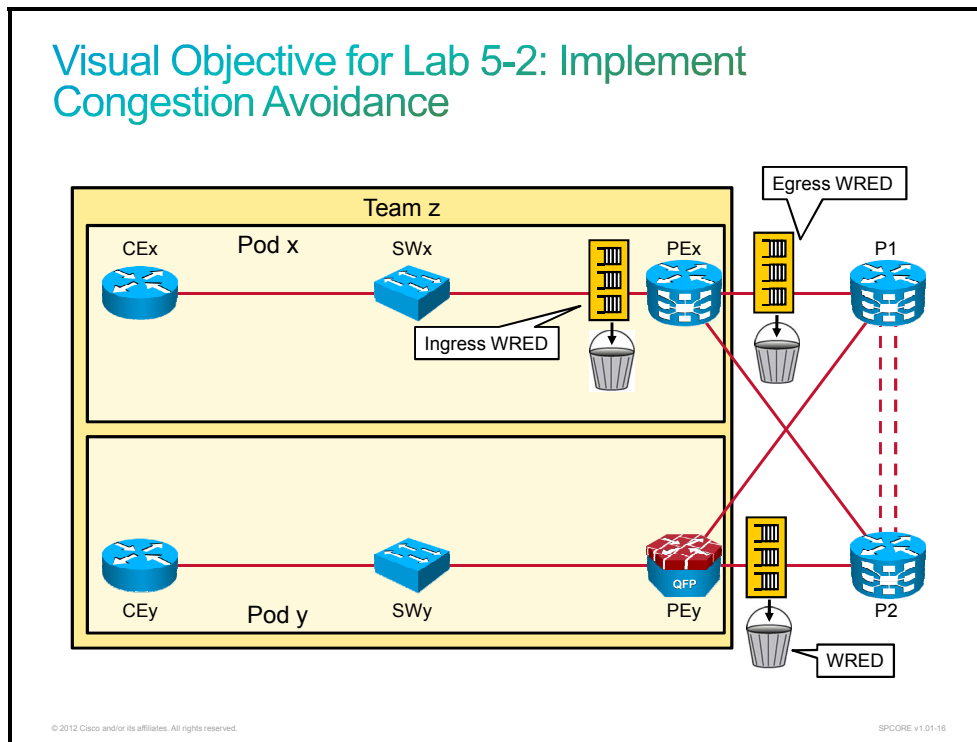
Note Students from two different pods will work in one team. The CE routers in both pods are running Cisco IOS Software. The first pod in the team will work on the PE router that is running Cisco IOS XR Software, and the second pod in the team will work on the PE router that is running Cisco IOS XE Software. Students in the same team should coordinate their lab activity.

In the lab activity, you will work on different Cisco routers running Cisco IOS (c2900), Cisco IOS XE (asr1001), and Cisco IOS XR (asr9k) Software. After completing this activity, you will be able to meet these objectives:

- Enable WRED on Cisco IOS XR and Cisco IOS XE Software
- Customize WRED profiles on Cisco IOS XR and Cisco IOS XE Software
- Monitor WRED on Cisco IOS XR and Cisco IOS XE Software

Visual Objective

The figure illustrates what you will accomplish in this activity.



Command List

The table describes the commands that are used in this lab activity.

Cisco IOS, IOS XE, and IOS XR WRED Commands

Command	Description
<code>class-map</code>	Defines a class map
<code>policy-map</code>	Defines a policy map
<code>class</code>	Defines handling for a traffic class within a policy map
<code>bandwidth</code>	Configures bandwidth guarantee for a traffic class
<code>service-policy</code>	Applies a policy map to an interface or a child-level policy map to the parent level
<code>police</code>	Sets policing parameters
<code>shape average</code>	Configures shaping to the average rate
<code>random-detect default</code>	Enables default WRED profiles on IOS XR Software
<code>random-detect dscp-based</code>	Enables default DSCP-based profiles on IOS XE Software
<code>random-detect dscp</code>	Configures WRED profile for a specific DSCP value
<code>show policy-map interface</code>	Displays the statistics of a service policy
<code>clear qos counters interface</code>	Clears QoS interface counters in IOS XR Software
<code>clear counters</code>	Clears QoS interface counters in IOS or IOS XE Software

Task 1: Configure IP SLA (Optional)

In this task, you will configure IP SLA. If you configured IP SLA in the previous exercise, you need only to reschedule it.

Activity Procedure

Complete these steps:

- Step 1** Make sure that your nearest core router is configured as an NTP master using the **ntp master** command.

Note	Task 1 is only required if not performed in the previous exercise.
-------------	--

- Step 2** Configure your CE router as the NTP client of the core router. Use the **ntp server core-router-address** command.

- Step 3** On your nearest P router, make sure that the SLA responder is enabled for the router loopback address, UDP port 3000.

```
P1:
ipsla
 responder
  type udp ipv4 address 10.0.1.1 port 3000
!
commit

P2:
ipsla
 responder
  type udp ipv4 address 10.0.2.1 port 3000
!
commit
```

- Step 4** On your CE router, configure an IP SLA probe, with timeout and threshold of 1 ms and frequency of 2 seconds. Calculate the ToS value for the DSCP EF (DSCP decimal value 46). What is the ToS value?

```
CE1:
ip sla 1
 udp-jitter 10.0.1.1 3000
 timeout 1
 threshold 1
 frequency 2
 tos 184

CE2:
ip sla 1
 udp-jitter 10.0.2.1 3000
 timeout 1
 threshold 1
 frequency 2
 tos 184
```

- Step 5** Schedule the measurement to start immediately and run repetitively on your CE router.
- ```
ip sla schedule 1 start-time now recurring
!
```
- Step 6** View the SLA statistics using the **show ip sla statistics 1** command. You should see no failures.

---

**Note** In this lab environment, where there is plenty of bandwidth and little traffic, you may examine the IP SLA results by configuring a traffic throttle and flooding the network with continuous pings. The IP SLA tool is not well-suited to testing the queuing configuration in this environment.

---

## Activity Verification

You have completed this task when you attain this results

- You have deployed the IP SLA tool for measuring network performance.

## Task 2A: Enable WRED on Cisco IOS XR Software

In this task, you will configure and verify basic WRED operations on the PEx router running Cisco IOS XR Software.

### Activity Procedure

Complete these steps:

- Step 1** On your PEx router, configure three traffic classes, matched by IPv4 access lists and DSCP values:
- Class **External** matching **External-podX-ACL** that permits ICMPv4 traffic sourced from your CE router uplink GigabitEthernet interface.
  - Class **Internal** matching **Internal-podX-ACL** that permits ICMPv4 traffic sourced from your CE router Loopback0 interface.
  - Class **ef-class** matching all traffic with DSCP set to the EF class.

---

**Note** This step is only required if not performed in the previous exercise.

---

- Step 2** On your PEx router, configure a policy map named **cbwfq-equal**. Configure CBWFQ for three traffic classes: Internal, External, and ef-class. Each class should have a 40 percent bandwidth guarantee.

---

**Note** You have established a baseline in which each class receives equal treatment. Bandwidth allocation has been set to 40 percent (out of 200 kb/s) to exceed the minimal IOS XR value of 64 kb/s.

---

- Step 3** On your PEx router, throttle the traffic toward the core to the rate of 200 kb/s. Apply the CBWFQ policy to the throttled traffic.

```
policy-map shape-all
class class-default
service-policy cbwfq-equal
shape average 200 kbps
commit
```

|                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Note</b>    | You will learn about traffic shaping and hierarchical policies in the next module. The throttle is needed in this lab environment to verify WRED operation.                                                                                                                                                                                                                                                                                                                                                                               |
| <b>Step 4</b>  | On your PEx router, apply the parent policy (shape-all) in the output direction on the core interface.                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <b>Step 5</b>  | Prepare a second terminal session to your CEx router. You may use Telnet to connect to your CE router from your switch, a P router, or the PE router. In each session, launch an ICMP echo flood to your nearest P router. One flood should be sourced from the loopback address and the other from the uplink address at the same time. Use commands <b>ping 10.0.1.1 size 18000 repeat 1000 source loopback 0</b> , and <b>ping 10.0.1.1 size 18000 repeat 1000</b> , respectively. You should receive no or only occasional responses. |
| <b>Note</b>    | You must ping your nearest P router. Otherwise, the traffic on the PE core interface will be labeled and the class maps will not match properly.                                                                                                                                                                                                                                                                                                                                                                                          |
| <b>Step 6</b>  | On your PEx router, verify the policy map operation on the core-facing interface. Why do you not receive any responses? What action does WRED perform?                                                                                                                                                                                                                                                                                                                                                                                    |
| <b>Note</b>    | You should see that all packet drops are tail drops. You should see that WRED does not drop any packets. For each class, you will see only one piece of information about WRED: "RED random drops(packets/bytes) : 0/0."                                                                                                                                                                                                                                                                                                                  |
| <b>Step 7</b>  | Enable the default WRED profile for each class in the cbwfq-equal policy map.                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| <b>Step 8</b>  | Clear the QoS counters on the core-facing interface using the command <b>clear qos counters interface GigabitEthernet 0/0/0/2</b> .                                                                                                                                                                                                                                                                                                                                                                                                       |
| <b>Step 9</b>  | Observe the results of both ICMP echo floods. You should see some responses.                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <b>Step 10</b> | On your PE router, verify the policy map operation on the core-facing interface. What action does WRED perform?                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <b>Note</b>    | You should see that WRED has been enabled and is dropping packets. In each class, the total number of dropped packets is equal to the number of RED random drops. Further, you should see that the WRED profile for Default WRED Curve appears in the output, and its number of dropped packets is equal to the total drop.                                                                                                                                                                                                               |
| <b>Step 11</b> | Remove the default WRED profile configuration from each class.                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |

## Activity Verification

You have completed this task when you attain these results:

- Perform an ICMP ping from your CEx router to the loopback IP address of P1 router. One flood should be sourced from the loopback address and the other from the uplink address. Pings should be performed at the same time. You should receive no response or only occasional responses:

```
CE1# ping 10.0.1.1 size 18000 repeat 1000 source loopback 0
```

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

```
Sending 1000, 18000-byte ICMP Echos to 10.0.1.1, timeout is 2 seconds:
```

```
Packet sent with a source address of 10.1.10.1
```

```
.....
```

Success rate is 0 percent (0/29)Success rate is 39 percent (194/497),  
round-trip min/avg/max = 380/440/756 ms

CE1# **ping 10.0.1.1 size 18000 repeat 1000**

Type escape sequence to abort.

Sending 1000, 18000-byte ICMP Echos to 10.0.1.1, timeout is 2 seconds:

.....

Success rate is 0 percent (0/33)

- On your PEx router, verify the policy map operation on the core-facing interface. You should see that all packet drops are tail drops:

RP/0/RSP0/CPU0:PE1# **show policy-map interface gigabitEthernet 0/0/0/2**

Tue Dec 6 16:19:26.407 UTC

GigabitEthernet0/0/0/2 direction input: Service Policy not installed

GigabitEthernet0/0/0/2 output: shape-all

<...output omitted...>

Policy cbwfq-equal Class Internal

| Classification statistics | (packets/bytes) | (rate - kbps) |
|---------------------------|-----------------|---------------|
| Matched :                 | 4473/6621034    | 0             |
| Transmitted :             | 3566/5347228    | 0             |
| Total Dropped :           | 907/1273806     | 0             |

Queueing statistics

|                                  |               |   |
|----------------------------------|---------------|---|
| Queue ID                         | : 266         |   |
| High watermark (Unknown)         |               |   |
| Inst-queue-len (packets)         | : 0           |   |
| Avg-queue-len (Unknown)          |               |   |
| Taildropped (packets/bytes)      | : 907/1273806 |   |
| Queue (conform) :                | 2055/3089230  | 0 |
| Queue (exceed) :                 | 1511/2257998  | 0 |
| RED random drops (packets/bytes) | : 0/0         |   |

Policy cbwfq-equal Class External

| Classification statistics | (packets/bytes) | (rate - kbps) |
|---------------------------|-----------------|---------------|
| Matched :                 | 3690/5462020    | 0             |
| Transmitted :             | 2898/4349524    | 0             |
| Total Dropped :           | 792/1112496     | 0             |

Queueing statistics

|                                  |               |   |
|----------------------------------|---------------|---|
| Queue ID                         | : 267         |   |
| High watermark (Unknown)         |               |   |
| Inst-queue-len (packets)         | : 0           |   |
| Avg-queue-len (Unknown)          |               |   |
| Taildropped (packets/bytes)      | : 792/1112496 |   |
| Queue (conform) :                | 1637/2455386  | 0 |
| Queue (exceed) :                 | 1261/1894138  | 0 |
| RED random drops (packets/bytes) | : 0/0         |   |

<...output omitted...>

- Perform an ICMP ping from your CEx router to the loopback IP address of P1 router after the default WRED profile for each class in the cbwfq-equal policy map is enabled. You should get more responses than previously:

CE1# **ping 10.0.1.1 size 18000 repeat 1000 source loopback 0**

Type escape sequence to abort.

Sending 1000, 18000-byte ICMP Echos to 10.0.1.1, timeout is 2 seconds:  
 Packet sent with a source address of 10.1.10.1  
 !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!  
 Success rate is 24 percent (11/45), round-trip min/avg/max =  
 632/806/1204 ms

CE1# **ping 10.0.1.1 size 18000 repeat 1000**

Type escape sequence to abort.

Sending 1000, 18000-byte ICMP Echos to 10.0.1.1, timeout is 2 seconds:  
 !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!  
 Success rate is 40 percent (21/52), round-trip min/avg/max =  
 632/809/1140 ms

- On your PEx router, verify the policy map operation on the core-facing interface. You should see that all packet drops are RED drops:

```
RP/0/RSP0/CPU0:PE1#sh policy-map interface gigabitEthernet 0/0/0/2
Tue Dec 6 17:04:13.726 UTC
GigabitEthernet0/0/0/2 direction input: Service Policy not installed
```

```
GigabitEthernet0/0/0/2 output: shape-all
<...output omitted...>
```

```
Policy cbwfq-equal Class Internal
 Classification statistics (packets/bytes) (rate - kbps)
 Matched : 585/831330 0
 Transmitted : 520/775500 0
 Total Dropped : 65/55830 0
 Queueing statistics
 Queue ID : 266
 High watermark (Unknown)
 Inst-queue-len (packets) : 0
 Avg-queue-len (Unknown)
 Taildropped(packets/bytes) : 0/0
 Queue(conform) : 195/289710 0
 Queue(exceed) : 325/485790 0
 RED random drops(packets/bytes) : 65/55830
 WRED profile for Default WRED Curve
 RED Transmitted (packets/bytes) : N/A
 RED random drops(packets/bytes) : 65/55830
 RED maxthreshold drops(packets/bytes): N/A
Policy cbwfq-equal Class External
 Classification statistics (packets/bytes) (rate - kbps)
 Matched : 676/960648 0
 Transmitted : 614/905592 0
 Total Dropped : 62/55056 0
 Queueing statistics
 Queue ID : 267
 High watermark (Unknown)
 Inst-queue-len (packets) : 0
 Avg-queue-len (Unknown)
 Taildropped(packets/bytes) : 0/0
 Queue(conform) : 252/372456 0
 Queue(exceed) : 362/533136 0
```

```

RED random drops (packets/bytes) : 62/55056

WRED profile for Default WRED Curve
RED Transmitted (packets/bytes) : N/A
RED random drops (packets/bytes) : 62/55056
RED maxthreshold drops (packets/bytes): N/A
<...output omitted...>

```

## Task 2B: Enable WRED on Cisco IOS XE Software

In this task, you will configure and verify basic WRED operations on the PE router running Cisco IOS XE Software.

### Activity Procedure

Complete these steps:

**Step 1** On your PE router, configure three traffic classes, matched by IPv4 access lists and DSCP values:

- Class **External** matching **External-podX-ACL** that permits ICMPv4 traffic sourced from your CE router uplink GigabitEthernet interface.
- Class **Internal** matching **Internal-podX-ACL** that permits ICMPv4 traffic sourced from your CE router Loopback0 interface.
- Class **ef-class** matching all traffic with DSCP set to the EF class.

---

**Note** This step is only required if it was not performed in the previous exercise.

---

**Step 2** On your PE router, configure a policy map named **cbwfq-equal**. Configure CBWFQ for three traffic classes: Internal, External, and ef-class. Each class should have a 30 percent bandwidth guarantee.

---

**Note** You have established a baseline in which each class receives equal treatment.

---

**Step 3** On your PE router, throttle the traffic toward the core to the rate of 100 kb/s. Apply the CBWFQ policy to the throttled traffic.

```

policy-map shape-all
class class-default
 shape average 100000
 service-policy cbwfq-equal

```

---

**Note** You will learn about traffic shaping and hierarchical policies in the next module. The throttle is needed in this lab environment to verify WRED operation.

---

**Step 4** On your PEy router, apply the parent policy (shape-all) in the output direction on the core interface.

**Step 5** Prepare a second terminal session to your CEy router. You may use Telnet to connect to your CEy router from your switch, a P router, or the PE router. In each session, launch an ICMP echo flood to your nearest P router. One flood should be sourced from the loopback address and the other from the uplink address. Use commands **ping 10.0.2.1 size 12000 repeat 100000 source loopback 0 timeout 0** and **ping 10.0.2.1 size 12000 repeat 100000 timeout 0**, respectively.

**Step 6** On your PEy router, verify the policy map operation on the core-facing interface. What action does WRED perform?

---

**Note** You should see that the packet drops (if any) are tail drops. There is no mention of WRED. WRED is disabled.

---

**Step 7** Enable default WRED curve for each class in the cbwfq-equal policy map.

**Step 8** On your PEy router, verify the policy map operation on the core-facing interface. What action does WRED perform?

## Activity Verification

You have completed this task when you attain these results:

- Perform an ICMP ping from your CEy router to the loopback IP address of the P2 router. One flood should be sourced from the loopback address and the other from the uplink address. Pings should be performed at the same time:

```
CE2# ping 10.0.2.1 size 12000 repeat 100000 timeout 0
```

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

```
Sending 100000, 12000-byte ICMP Echos to 10.0.2.1, timeout is 0
seconds:
```

```
.....
.....
Success rate is 0 percent (0/777) ms
```

```
CE2# ping 10.0.2.1 size 12000 repeat 100000 source loopback 0 timeout
0
```

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

```
Sending 100000, 12000-byte ICMP Echos to 10.0.2.1, timeout is 0
seconds:
```

```
Packet sent with a source address of 10.2.10.1
```

```
.....
.....
Success rate is 0 percent (0/2276)
```

- On your PEx router, verify the policy map operation on the core-facing interface. You should see that all packet drops are tail drops:

```
PE2#show policy-map interface GigabitEthernet 0/0/3
```

```
GigabitEthernet0/0/3
```

```
Service-policy output: shape-all
```

<...output omitted...>

```
Service-policy : cbwfq-equal
```

```
Class-map: Internal (match-any)
```

```
170640 packets, 240425120 bytes
```

```
5 minute offered rate 4284000 bps, drop rate 4278000 bps
```

```
Match: access-group name Internal-podY-ACL
```

```
Queueing
```

```
queue limit 64 packets
```

```
(queue depth/total drops/no-buffer drops) 61/170461/0
```

```

Class-map: External (match-any)
 6177 packets, 8550658 bytes
 5 minute offered rate 182000 bps, drop rate 176000 bps
 Match: access-group name External-podY-ACL
 Queueing
 queue limit 64 packets
 (queue depth/total drops/no-buffer drops) 46/5998/0
 (pkts output/bytes output) 179/261626
 bandwidth 30% (30 kbps)

```

- Perform an ICMP ping from your CEx router to the loopback IP address of the P2 router after the default WRED profile for each class in the cbwfg-equal policy map is enabled:

Success rate is 1 percent (2/123), round-trip min/avg/max = 940/966/992 ms

Success rate is 0 percent (0/118)

- On your PEy router, verify the policy map operation on the core-facing interface. You should see some RED drops:

```

Class-map: Internal (match-any)
 1062 packets, 1449748 bytes
 5 minute offered rate 2000 bps, drop rate 0000 bps
Match: access-group name Internal-podY-ACL
Queueing
 queue limit 64 packets
 (queue depth/total drops/no-buffer drops) 0/554/0
 (pkts output/bytes output) 508/747672
 bandwidth 30% (30 kbps)
 Exp-weight-constant: 4 (1/16)

```

```

Mean queue depth: 33 packets
dscp Transmitted Random drop Tail drop
Minimum Maximum
 pkts/bytes pkts/bytes pkts/bytes
thresh thresh

16 default 508/747672 51/73194 503/628882
 32 1/10

```

```

Class-map: External (match-any)
 1172 packets, 1515728 bytes
 5 minute offered rate 2000 bps, drop rate 0000 bps
 Match: access-group name External-podY-ACL
 Queueing
 queue limit 64 packets
 (queue depth/total drops/no-buffer drops) 0/563/0
 (pkts output/bytes output) 609/831598
 bandwidth 30% (30 kbps)

```

```

Exp-weight-constant: 4 (1/16)
Mean queue depth: 26 packets
dscp Transmitted Random drop Tail drop
Minimum Maximum
 pkts/bytes pkts/bytes pkts/bytes
thresh thresh

16 default 577/829358 45/68130 485/613690
 32 1/10

16 cs6 32/2240 2/140 31/2170
 32 1/10

```

- You have examined the WRED operation using the default WRED profile. The command output displays WRED statistics for each class, with specific curves for each detected DSCP value. The drops, if any, are marked as WRED drops.

## Task 3A: Customize WRED on Cisco IOS XR Software

In this task, you will customize a WRED profile and verify its operation on the PE router running Cisco IOS XR Software.

### Activity Procedure

Complete these steps:

- Step 1** Configure custom WRED profiles for each class in the cbwfq-equal policy map. This should give preference to the traffic with the default DSCP (000000) from the Internal class over the External class:
- Class Internal:
    - DSCP EF: minimum threshold – 100 KB, maximum threshold – 200 KB
    - DSCP 000000: minimum threshold – 10 KB, maximum threshold – 100 KB
  - Class External:
    - DSCP 000000: minimum threshold – 5 KB, maximum threshold – 12 KB
  - Class ef-class:
    - DSCP EF: minimum threshold – 200 KB, maximum threshold – 300 KB

- Step 2** Clear the QoS counters on the core-facing interface.
- Step 3** Observe the results of both ICMP echo floods. You should see only a few drops in the Internal traffic and many drops in the External flood.
- Step 4** On your PE router, verify the policy map operation on the core-facing interface. What action does WRED perform?

---

**Note** You should see that WRED is dropping more packets for the External class than for the Internal class. The output lists all configured WRED curves for each class.

---

- Step 5** Apply the same policy (shape-all) to the customer-facing interface in the ingress direction.

---

**Note** You will not test ingress WRED, because it works in the same way as in the egress direction. This step demonstrates that Cisco IOS XR Software supports ingress WRED.

---

- Step 6** Remove the policy maps from the core- and customer-facing interfaces.

## Activity Verification

You have completed this task when you attain these results:

- Observe the results of both ICMP echo floods. You should see only a few drops in the Internal traffic and many drops in the External flood:

```
CE1# ping 10.0.1.1 size 18000 repeat 1000 source loopback 0
Type escape sequence to abort.
Sending 1000, 18000-byte ICMP Echos to 10.0.1.1, timeout is 2 seconds:
Packet sent with a source address of 10.1.10.1
!!
!!
Success rate is 87 percent (64/73), round-trip min/avg/max =
632/922/1148 ms
```

```
CE1# ping 10.0.1.1 size 18000 repeat 1000
Type escape sequence to abort.
Sending 1000, 18000-byte ICMP Echos to 10.0.1.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/40)
```

- On your PE router, verify the policy map operation on the core-facing interface.

```
RP/0/RSP0/CPU0:PE1#sh policy-map interface GigabitEthernet 0/0/0/2
Wed Dec 7 09:44:46.378 UTC
GigabitEthernet0/0/0/2 direction input: Service Policy not installed
```

```
GigabitEthernet0/0/0/2 output: shape-all
```

<...output omitted...>

| Policy cbwfq-equal Class Internal |   | (packets/bytes) | (rate - kbps) |
|-----------------------------------|---|-----------------|---------------|
| Classification statistics         |   |                 |               |
| Matched                           | : | 949/1348602     | 0             |
| Transmitted                       | : | 940/1343760     | 0             |
| Total Dropped                     | : | 9/4842          | 0             |

```

Queueing statistics
 Queue ID : 266
 High watermark (Unknown)
 Inst-queue-len (packets) : 0
 Avg-queue-len (Unknown)
 Taildropped(packets/bytes) : 0/0
 Queue(conform) : 352/481416 0
 Queue(exceed) : 588/862344 0
 RED random drops(packets/bytes) : 9/4842

WRED profile for WRED Curve 1
 RED Transmitted (packets/bytes) : N/A
 RED random drops(packets/bytes) : 0/0
 RED maxthreshold drops(packets/bytes) : N/A
WRED profile for WRED Curve 2
 RED Transmitted (packets/bytes) : N/A
 RED random drops(packets/bytes) : 9/4842
 RED maxthreshold drops(packets/bytes) : N/A
Policy cbwfq-equal Class External
 Classification statistics (packets/bytes) (rate - kbps)
 Matched : 520/738960 0
 Transmitted : 264/400752 0
 Total Dropped : 256/338208 0
Queueing statistics
 Queue ID : 267
 High watermark (Unknown)
 Inst-queue-len (packets) : 0
 Avg-queue-len (Unknown)
 Taildropped(packets/bytes) : 0/0
 Queue(conform) : 113/171534 0
 Queue(exceed) : 151/229218 0
 RED random drops(packets/bytes) : 256/338208

WRED profile for WRED Curve 1
 RED Transmitted (packets/bytes) : N/A
 RED random drops(packets/bytes) : 256/338208
 RED maxthreshold drops(packets/bytes) : N/A

```

<...output omitted...>

## Task 3B: Customize WRED on Cisco IOS XE Software

In this task, you will customize a WRED profile and verify its operation on the PE router running Cisco IOS XE Software.

### Activity Procedure

Complete these steps:

- Step 1** Configure custom WRED profiles for each class in the cbwfq-equal policy map. This should give preference to the traffic with the default DSCP (000000) from the Internal class over the External class:

- Class Internal:
  - DSCP EF: minimum threshold – 30 packets, maximum threshold – 100 packets, maximum drop probability 5 percent

- DSCP 000000: minimum threshold – 20 packets, maximum threshold – 100 packets, maximum drop probability – 10 percent
- Class External:
  - DSCP 000000: minimum threshold – 1 packet, maximum threshold – 2 packets, maximum drop probability – 100 percent

**Step 2** Clear the QoS counters on the core-facing interface.

**Step 3** Observe the results of both ICMP echo floods. You should see no or very few drops in the Internal traffic and many drops in the External flood. On your PEy router, verify the policy map operation on the core-facing interface. You should see that WRED is aggressively dropping packets in the External class.

**Step 4** Remove the policy map from the core-facing interface.

## Activity Verification

You have completed this task when you attain these results:

- On your PEy router, verify the policy map operation on the core-facing interface. You should see that WRED is aggressively dropping packets in the External class.

```
PE2# show policy-map interface GigabitEthernet 0/0/3
GigabitEthernet0/0/3
```

```
Service-policy output: shape-all
```

```
Service-policy : cbwfg-equal
```

```
Class-map: Internal (match-any)
 71847 packets, 101283078 bytes
 5 minute offered rate 2677000 bps, drop rate 2665000 bps
 Match: access-group name Internal-podY-ACL
 Queueing
 queue limit 64 packets
 (queue depth/total drops/no-buffer drops) 64/71550/0
 (pkts output/bytes output) 297/417498
 bandwidth 30% (30 kbps)
 Exp-weight-constant: 4 (1/16)
 Mean queue depth: 62 packets
 dscp Transmitted Random drop Tail drop
 Minimum Maximum Mark pkts/bytes pkts/bytes
 thresh thresh prob pkts/bytes pkts/bytes
 20 100 1/10 297/417498 1/1514 0/0
 30 100 1/20 0/0 0/0 0/0
```

```
Class-map: External (match-any)
 2467 packets, 3414778 bytes
 5 minute offered rate 91000 bps, drop rate 85000 bps
 Match: access-group name External-podY-ACL
 Queueing
 queue limit 64 packets
 (queue depth/total drops/no-buffer drops) 3/2336/0
```

```

(pkts output/bytes output) 131/196994
bandwidth 30% (30 kbps)
 Exp-weight-constant: 4 (1/16)
 Mean queue depth: 4 packets
 dscp Transmitted Random drop Tail drop
 Maximum Mark
 pkts/bytes
 thresh prob
 1 default 131/196994 29/38546 2307/3179238
 2 1/1

```

# Lab 6-1: Implement Traffic Policing

Complete this lab activity to practice what you learned in the related module.

## Activity Objective

In this lab activity, you will configure and monitor class-based policing.

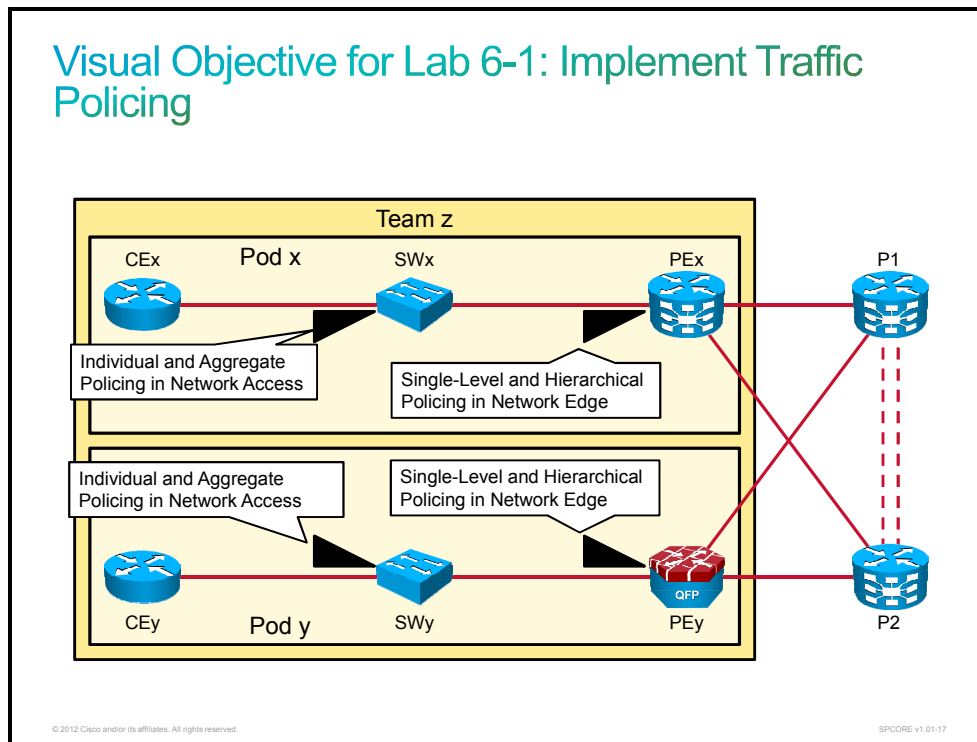
**Note** Students from two different pods will work in one team. The CE routers in both pods are running Cisco IOS Software. The first pod in the team will work on the PE router that is running Cisco IOS XR Software, and the second pod in the team will work on the PE router that is running Cisco IOS XE Software. Students in the same team should coordinate their lab activity.

In the lab activity, you will work on different Cisco routers running Cisco IOS (c2900), Cisco IOS XE (asr1001), and Cisco IOS XR (asr9k) Software. After completing this activity, you will be able to meet these objectives:

- Configure single token bucket class-based policing
- Monitor single token bucket class-based policing
- Configure dual token bucket class-based policing
- Monitor dual token bucket class-based policing
- Configure hierarchical class-based policing
- Monitor hierarchical class-based policing
- Configure LPTS hardware policing
- Monitor LPTS hardware policing

## Visual Objective

The figure illustrates what you will accomplish in this activity.



# Command List

The table describes the commands that are used in this lab activity.

## Cisco IOS, IOS XE, and IOS XR Policing Commands

| Command                                   | Description                                                                           |
|-------------------------------------------|---------------------------------------------------------------------------------------|
| <code>class-map</code>                    | Defines a class map                                                                   |
| <code>policy-map</code>                   | Defines a policy map                                                                  |
| <code>service-policy</code>               | Applies a policy map to an interface, or a child level policy map to the parent level |
| <code>police</code>                       | Sets policing parameters                                                              |
| <code>conform-action</code>               | Defines the conform action of a policer                                               |
| <code>exceed-action</code>                | Defines the exceed action of a policer                                                |
| <code>violate-action</code>               | Defines the violate action of a policer                                               |
| <code>show policy-map interface</code>    | Displays the statistics of a service policy                                           |
| <code>clear qos counters interface</code> | Clears QoS interface counters in IOS XR                                               |
| <code>clear counters</code>               | Clears QoS interface counters in IOS and IOS XE                                       |

## Cisco IOS Commands (Access Switches)

| Command                                | Description                                                                          |
|----------------------------------------|--------------------------------------------------------------------------------------|
| <code>class-map</code>                 | Defines a class map                                                                  |
| <code>policy-map</code>                | Defines a policy map                                                                 |
| <code>service-policy</code>            | Applies a policy map to an interface or a child-level policy map to the parent level |
| <code>police</code>                    | Sets policing parameters                                                             |
| <code>conform-action</code>            | Defines the conform action of a policer                                              |
| <code>exceed-action</code>             | Defines the exceed action of a policer                                               |
| <code>violate-action</code>            | Defines the violate action of a policer                                              |
| <code>policer aggregate</code>         | Defines an aggregate policer                                                         |
| <code>police aggregate</code>          | Applies an aggregate policer to a traffic class inside a policy map                  |
| <code>show policy-map interface</code> | Displays the statistics of a service policy                                          |
| <code>clear counters</code>            | Clears QoS interface counters                                                        |

## Cisco IOS XR Commands (LPTS)

| Command                                               | Description                                                       |
|-------------------------------------------------------|-------------------------------------------------------------------|
| <code>lpts pifib hardware police</code>               | Enters LPTS hardware policing configuration mode                  |
| <code>flow fragment rate</code>                       | Configures a policing rate for fragmented packets                 |
| <code>flow icmp local rate</code>                     | Configures a policing rate for ICMP traffic with local interest   |
| <code>flow icmp default rate</code>                   | Configures a policing rate for all unspecified ICMP traffic types |
| <code>flow http known rate</code>                     | Configures a policing rate for established HTTP traffic           |
| <code>flow http default rate</code>                   | Configures a policing rate for nonestablished HTTP traffic        |
| <code>lpts pifib hardware police location</code>      | Configures the LPTS node                                          |
| <code>show lpts pifib hardware police location</code> | Displays LPTS statistics                                          |

## Task 1A: Configure Single-Level Policing on Cisco IOS XR Software

In this task, you will configure and verify various single-level policing schemes: single-rate single token bucket, single-rate dual token bucket, and dual-rate on the PE router running Cisco IOS XR Software.

### Activity Procedure

Complete these steps:

- Step 1** On your PE router, configure an IPv4 access list that permits traffic from the uplink address of your CE router. You may name the access list **Customer-podX-ACL**, where *X* is your pod number.
- Step 2** On your PE router, create a traffic class that matches the access group defined in the previous step. You may name the class map **Customer**.
- Step 3** On your PE, configure a single token bucket policer for the traffic class defined in the previous step. Use these parameters:
- CIR: 1 MB/s
  - Committed burst duration: 10 ms
  - Possible name: ingress
  - Location: Input direction on the CE-facing interface
- Step 4** On your CE router send a traffic sample to the P1 router consisting of 100 1400-byte ICMP echoes using the **ping 10.0.1.1 repeat 100 size 1400** command.
- Step 5** Verify the statistics of the forwarded and dropped traffic using the **show policy-map int GigabitEthernet 0/0/0/0** command. You should see that 50 percent of packets conform and are forwarded, and the other 50 percent exceed the rate and are dropped.
- Step 6** On your PE router, clear the QoS counters on the CE-facing interface using the **clear qos counters interface GigabitEthernet 0/0/0/0** command.
- Step 7** Change the single token bucket policer to a dual token bucket policer by setting these parameters:
- Excess burst duration: 20 ms
  - Exceed action: Set DSCP scavenger class marker (cs1)

---

**Note** You will have to remove the default exceed action drop to commit the change.

---

- Keep the remaining parameters.
- Step 8** From your CE router, resend the traffic batch (**ping 10.0.1.1 repeat 100 size 1400**).
- Step 9** Verify the policer statistics on your PE router. You should see that the number of conforming, exceeding, and violating packets is (nearly) the same, but the conforming and exceeding packets are forwarded, and thus the drop ratio is 1/3.
- Step 10** On your PE router, clear the QoS counters on the CE-facing interface.
- Step 11** Change the policing scheme to dual-rate. Set the peak rate to 2 MB/s. Keep the burst and peak burst values at their previous values. Keep the same conform, exceed, and violate actions.

- Step 12** From your CE router, resend the traffic batch (**ping 10.0.1.1 repeat 100 size 1400**).
- Step 13** Verify the policer statistics on your PE router. You should see that the conforming packets represent roughly 10 percent of the volume, the exceeding packets 86 percent, and violating packets 4 percent. Thus, 96 percent go through, and 4 percent are dropped.
- Step 14** Remove the service policy from the CE-facing interface.

## Activity Verification

You have completed this task when you attain these results:

- Verify the statistics of the forwarded and dropped traffic using the **show policy-map int GigabitEthernet 0/0/0/0** command when single token bucket policing is configured:

```
RP/0/RSP0/CPU0:PE1# show policy-map interface GigabitEthernet 0/0/0/0
GigabitEthernet0/0/0/0 input: ingress
```

```
Class Customer
 Classification statistics (packets/bytes) (rate - kbps)
 Matched : 100/141800 0
 Transmitted : N/A
 Total Dropped : N/A
 Policing statistics (packets/bytes) (rate - kbps)
 Policed(conform) : 50/70900 0
 Policed(exceed) : 50/70900 0
 Policed(violate) : 0/0 0
 Policed and dropped : 50/70900
Class class-default
 Classification statistics (packets/bytes) (rate - kbps)
 Matched : 12/1016 0
 Transmitted : N/A
 Total Dropped : N/A
GigabitEthernet0/0/0/0 direction output: Service Policy not installed
```

- Verify the statistics of the forwarded and dropped traffic using the **show policy-map int GigabitEthernet 0/0/0/0** command when dual token bucket policing is configured:

```
RP/0/RSP0/CPU0:PE1# show policy-map interface GigabitEthernet 0/0/0/0
GigabitEthernet0/0/0/0 input: ingress
```

```
Class Customer
 Classification statistics (packets/bytes) (rate - kbps)
 Matched : 100/141800 1
 Transmitted : N/A
 Total Dropped : N/A
 Policing statistics (packets/bytes) (rate - kbps)
 Policed(conform) : 34/48212 1
 Policed(exceed) : 33/46794 0
 Policed(violate) : 33/46794 0
 Policed and dropped : 33/46794
Class class-default
 Classification statistics (packets/bytes) (rate - kbps)
```

```

 Matched : 0/0 0
 Transmitted : N/A
 Total Dropped : N/A
GigabitEthernet0/0/0/0 direction output: Service Policy not installed

```

- Verify the statistics of the forwarded and dropped traffic using the **show policy-map int GigabitEthernet 0/0/0/0** command when the policing scheme is changed to dual-rate:

```

RP/0/RSP0/CPU0:PE1# show policy-map interface GigabitEthernet 0/0/0/0
GigabitEthernet0/0/0/0 input: ingress

```

```

Class Customer
 Classification statistics (packets/bytes) (rate - kbps)
 Matched : 100/141800 0
 Transmitted : N/A
 Total Dropped : N/A
 Policing statistics (packets/bytes) (rate - kbps)
 Policed(conform) : 10/14180 0
 Policed(exceed) : 86/121948 0
 Policed(violate) : 4/5672 0
 Policed and dropped : 4/5672
Class class-default
 Classification statistics (packets/bytes) (rate - kbps)
 Matched : 0/0 0
 Transmitted : N/A
 Total Dropped : N/A
GigabitEthernet0/0/0/0 direction output: Service Policy not installed

```

## Task 1B: Configure Single-Level Policing on Cisco IOS XE Software

In this task, you will configure and verify various single-level policing schemes: single-rate single token bucket, single-rate dual token bucket, and dual-rate on the PE router running Cisco IOS XE Software.

### Activity Procedure

Complete these steps:

- Step 1** On your PE router running Cisco IOS XE Software, configure an IPv4 access list that permits traffic from the uplink address of your CE router. You may name the access list **Customer-podY-ACL**, where *Y* is your pod number.
- Step 2** On your PE router, create a traffic class that matches the access group defined in the previous step. You may name the class map **Customer**.

- Step 3** On your PE, configure a single token bucket policer for the traffic class defined in the previous step. Use these parameters:
- CIR: 1 MB/s
  - Committed burst: 1300 bytes
  - Possible name: ingress
  - Location: Input direction on the CE-facing interface
- Step 4** On your CE router, send a traffic sample to the P1 router consisting of 100 1400-byte ICMP echoes using the **ping 10.0.1.1 repeat 100 size 1400** command.
- Step 5** Verify the statistics of the forwarded and dropped traffic using the **show policy-map interface GigabitEthernet 0/0/0** command. You should see that 50 percent of packets conform and are forwarded, and the other 50 percent exceed the rate and are dropped.
- 
- Note** You may see more than 50 percent conforming packets, because control traffic is also considered.
- 
- Step 6** On your PE router, clear the QoS counters on the CE-facing interface using the **clear counters GigabitEthernet 0/0/0** command.
- Step 7** Change the single token bucket policer to a dual token bucket policer by setting these parameters:
- Excess burst size: 2600 bytes
  - Exceed action: Set DSCP scavenger class marker (cs1)
- 
- Note** You will have to remove the default exceed action drop to commit the change.
- 
- Keep the remaining parameters.
- Step 8** From your CE router, resend the ICMP echo batch (**ping 10.0.1.1 repeat 100 size 1400**).
- Step 9** Verify the policer statistics on your PE router. You should see that the number of conforming, exceeding, and violating packets is (nearly) the same, but the conforming and exceeding packets are forwarded, and thus the drop ratio is 1/3.
- 
- Note** You will see more conforming packets, because control traffic is also considered.
- 
- Step 10** On your PE router, clear the QoS counters on the CE-facing interface.
- Step 11** Change the policing scheme to dual-rate. Set the peak rate to 2 MB/s. Keep the burst size of 1300 bytes. Let Cisco IOS Software calculate the peak burst size. Keep the same conform, exceed, and violate actions.
- Step 12** From your CE router, resend the ICMP echo batch (**ping 10.0.1.1 repeat 100 size 1400**).
- Step 13** Verify the policer statistics on your PE router. You should see that the conforming packets represent roughly 11 percent of the volume, the exceeding packets 88 percent, and only occasional packets (roughly 1 out of 100) violate. Thus the bulk of the traffic is forwarded and only a few packets are dropped.
- Step 14** Remove the service policy from the CE-facing interface.

## Activity Verification

You have completed this task when you attain these results:

- Verify the statistics of the forwarded and dropped traffic using the **show policy-map interface GigabitEthernet 0/0/0** command when single token bucket policing is configured.

```
PE2# show policy-map interface GigabitEthernet0/0/0
GigabitEthernet0/0/0
 Service-policy input: ingress

 Class-map: Customer (match-any)
 124 packets, 143640 bytes
 5 minute offered rate 9000 bps, drop rate 4000 bps
 Match: access-group name Customer-pod8-ACL
 police:
 rate 1000000 bps, burst 1300 bytes
 conformed 74 packets, 72940 bytes; actions:
 transmit
 exceeded 50 packets, 70700 bytes; actions:
 drop
 conformed 4000 bps, exceeded 4000 bps

 Class-map: class-default (match-any)
 114 packets, 14645 bytes
 5 minute offered rate 0000 bps, drop rate 0000 bps
 Match: any
```

- Verify the policer statistics on your PE router when single token bucket policing is configured.

```
PE2# show policy-map interface GigabitEthernet 0/0/0
GigabitEthernet0/0/0

 Service-policy input: ingress

 Class-map: Customer (match-any)
 119 packets, 143170 bytes
 5 minute offered rate 7000 bps, drop rate 4000 bps
 Match: access-group name Customer-pod8-ACL
 police:
 rate 1000000 bps, burst 1300 bytes, peak-burst 2600 bytes
 conformed 36 packets, 41664 bytes; actions:
 transmit
 exceeded 29 packets, 41006 bytes; actions:
 set-dscp-transmit cs1
 violated 29 packets, 41006 bytes; actions:
 drop
 conformed 5000 bps, exceeded 5000 bps, violated 5000 bps

 Class-map: class-default (match-any)
 36 packets, 4532 bytes
 5 minute offered rate 0000 bps, drop rate 0000 bps
 Match: any
```

- Verify the policer statistics on your PE router when policing scheme is changed to dual-rate.

```
PE2# show policy-map interface GigabitEthernet0/0/0
GigabitEthernet0/0/0

Service-policy input: ingress
 Class-map: Customer (match-any)
 103 packets, 141666 bytes
 5 minute offered rate 0000 bps, drop rate 0000 bps
 Match: access-group name Customer-pod8-ACL
 police:
 rate 1000000 bps, burst 1300 bytes
 peak-rate 2000000 bps, peak-burst 62500 bytes
 conformed 14 packets, 15820 bytes; actions:
 transmit
 exceeded 88 packets, 124432 bytes; actions:
 set-dscp-transmit cs1
 violated 1 packets, 1414 bytes; actions:
 drop
 conformed 0000 bps, exceeded 0000 bps, violated 0000 bps

 Class-map: class-default (match-any)
 10 packets, 1190 bytes
 5 minute offered rate 0000 bps, drop rate 0000 bps
 Match: any
```

## Task 2: Configure Individual and Aggregate Policers on Cisco IOS Access Switches

In this task, you will configure and verify individual and aggregate policing schemes on Cisco IOS access switches.

### Activity Procedure

Complete these steps:

- Step 1** Verify that there is no policing in the traffic path from your CE router to the P1 router. From your CE router, resend the ICMP echo batch (**ping 10.0.1.1 repeat 100 size 1400**). You should receive immediate responses to all echo requests.
- Step 2** On your access switch SW that provides the interconnection between your CE and PE routers, configure an IPv4 access list that permits traffic from the uplink address of your CE router. You may name the access list **Customer-podX-ACL**, where *X* is your pod number.
- Step 3** On your switch, create a traffic class that matches the access group defined in the previous step. You may name the class map **Customer**.
- Step 4** Configure a dual token bucket policer by setting these parameters:
  - Traffic class: Defined in previous step
  - Possible name: ingress
  - CIR: 1 MB/s
  - PIR: 2 MB/s
  - Committed burst: 8000 bytes

- Conform action: Transmit
- Exceed action: Set DSCP scavenger class marker (cs1)
- Violate action: Drop
- Location: Input direction on the CE-facing interface

---

**Note** You will not configure single token bucket and dual token bucket policing schemes on the switch, although they are also supported. The configuration and operation is equivalent to those that are available on Cisco IOS XE routers.

---

**Step 5** On your CE router, resend the ICMP echo batch using the **ping 10.0.1.1 repeat 100 size 1400** command.

**Step 6** Verify the policing statistics using the **show policy-map interface FastEthernet 0/1** command. You should see that the conforming packets represent roughly 19 percent of the volume, the exceeding packets 80 percent, and only occasional packets (roughly 1 out of 100) violate. Thus, the bulk of the traffic is forwarded and only a few packets are dropped.

---

**Note** You may see more conforming packets because control traffic is also considered.

---

**Step 7** Configure an aggregate policer with the same policing parameters as the individual policer (CIR: 1 MB/s, PIR: 2 MB/s, committed burst: 8000 bytes, conform action: transmit, exceed action: set DSCP scavenger class marker, violate action: drop). Name it **agg-customer-policer**.

---

**Note** In a work situation, you will implement aggregate policers if you have multiple traffic classes inside a single policy map. In this exercise, you will apply the aggregate policer to a single traffic class.

---

**Step 8** Configure the class Customer in your ingress policy map to use the aggregate policer rather than its own specific values.

---

**Note** You will have to remove the individual policer from the class configuration.

---

**Step 9** From your CE router, resend the traffic batch (**ping 10.0.1.1 repeat 100 size 1400**).

**Step 10** Verify the policer statistics on your access switch. You should see the same number of conforming, exceeding, and violating packets as before, and a note about aggregate policing.

**Step 11** On your PE router, clear the QoS counters on the CE-facing interface.

**Step 12** Remove the service policy from the CE-facing interface.

## Activity Verification

You have completed this task when you attain these results:

- Verify the policing statistics using the **show policy-map interface FastEthernet 0/1** command when individual policing is configured:

```
SW1# show policy-map interface FastEthernet0/1
FastEthernet0/1
Service-policy input: ingress
```

```

Class-map: Customer (match-any)
 0 packets
 Match: access-group name Customer-podX-ACL
 police cir 1000000 bc 8000 pir 2000000 be 62500
 conform-action transmit
 exceed-action set-dscp-transmit cs1
 violate-action drop
 conform: 21 (packets) 27104 (bytes)
 exceed: 80 (packets) 113440 (bytes)
 violate: 1 (packets) 1418 (bytes)
 conform: 43107 bps, exceed: 181504 bps, violate: 2268 bps

```

```

Class-map: class-default (match-any)
 0 packets, 0 bytes
 5 minute offered rate 0 bps, drop rate 0 bps
 Match: any

```

- Verify the policer statistics on your access switch when aggregate policing is configured:

```
SW1# show policy-map interface FastEthernet 0/1
```

```
FastEthernet0/1
```

```
Service-policy input: ingress
```

```

Class-map: Customer (match-any)
 0 packets
 Match: access-group name Customer-podX-ACL
 police aggregate agg-customer-policer
 conform: 351 (packets) 53586 (bytes)
 exceed: 81 (packets) 113518 (bytes)
 violate: 1 (packets) 1418 (bytes)
 conform: 23884 bps, exceed: 81801 bps, violate: 0 bps

```

NOTE: Policing statistics for a class configured with an aggregate policer are the same for all classes in the policy-map configured with the same aggregate policer

```

Class-map: class-default (match-any)
 0 packets, 0 bytes
 5 minute offered rate 0 bps, drop rate 0 bps
 Match: any

```

## Task 3: Configure Hierarchical Policing

In this task, you will configure and verify hierarchical policing on your PE router.

### Activity Procedure

Complete these steps:

- Step 1** On your PE router, configure parent-level policing with these parameters:
- Class map Customers should match protocol IPv4.
  - Policy map ingress-all
  - CIR: 2 MB/s
  - Default actions (conform: transmit, exceed: drop)
  - Location: Input direction on the CE-facing interface
- Step 2** On your CE router, resend the ICMP echo batch using the **ping 10.0.1.1 repeat 100 size 1400** command.
- Step 3** Verify the parent-level statistics using the **show policy-map interface** command.

---

**Note** On the Cisco IOS XR router, you should see that 96 percent of the packets conform and are forwarded and 4 percent exceed the rate and are dropped. On the Cisco IOS XE router, 99 percent of the packets are transmitted. The difference results from the different burst implementations on Cisco IOS XR and Cisco IOS XE platforms.

---

- Step 4** On your PE router, clear the QoS counters on the CE-facing interface.
- Step 5** Define a child-level policer with these parameters:
- Policy map name example: **Customer-policer**
  - Reuse the existing Customer class (matching IPv4 ACL)
  - CIR: 50 percent of the parent level
  - PIR: 60 percent of the parent level
  - Conform action: Transmit
  - Exceed action: Mark traffic as scavenger class (DSCP cs1)
  - Violate action: Drop
- Step 6** Apply the child policer to the traffic class (Customers) of the parent policer.
- Step 7** On your PE router, clear the QoS counters.
- Step 8** From your CE router, resend the traffic batch (**ping 10.0.1.1 repeat 100 size 1400**).
- Step 9** Verify the policer statistics on your PE router.

---

**Note** On the Cisco IOS XR router, you should see that the parent policer does not drop packets and the child policer takes precedence. On the Cisco IOS XE router, the statistics will differ from the Cisco IOS XR router.

---

- Step 10** Remove the service policy from the CE-facing interface.

## Activity Verification

You have completed this task when you attain these results:

- Verify the parent-level statistics using the **show policy-map interface** command.

```
RP/0/RSP0/CPU0:PE1# show policy-map interface GigabitEthernet 0/0/0/0
```

```
GigabitEthernet0/0/0/0 input: ingress-all
```

```
Class Customers
```

| Classification statistics | (packets/bytes) | (rate - kbps) |
|---------------------------|-----------------|---------------|
| Matched :                 | 100/141800      | 0             |
| Transmitted :             | N/A             |               |
| Total Dropped :           | N/A             |               |

| Policing statistics   | (packets/bytes) | (rate - kbps) |
|-----------------------|-----------------|---------------|
| Policed(conform) :    | 96/136128       | 0             |
| Policed(exceed) :     | 4/5672          | 0             |
| Policed(violate) :    | 0/0             | 0             |
| Policed and dropped : | 4/5672          |               |

```
Class class-default
```

| Classification statistics | (packets/bytes) | (rate - kbps) |
|---------------------------|-----------------|---------------|
| Matched :                 | 3/254           | 0             |
| Transmitted :             | N/A             |               |
| Total Dropped :           | N/A             |               |

```
GigabitEthernet0/0/0/0 direction output: Service Policy not installed
```

```
PE2# show policy-map interface GigabitEthernet 0/0/0
```

```
GigabitEthernet0/0/0
```

```
Service-policy input: ingress-all
```

```
Class-map: Customers (match-any)
```

```
102 packets, 141588 bytes
```

```
5 minute offered rate 3000 bps, drop rate 0000 bps
```

```
Match: protocol ip
```

```
police:
```

```
rate 2000000 bps, burst 62500 bytes
```

```
conformed 101 packets, 140174 bytes; actions:
```

```
transmit
```

```
exceeded 1 packets, 1414 bytes; actions:
```

```
drop
```

```
conformed 3000 bps, exceeded 0000 bps
```

```
Class-map: class-default (match-any)
```

```
13 packets, 1109 bytes
```

```
5 minute offered rate 0000 bps, drop rate 0000 bps
```

```
Match: any
```

- Verify the policer statistics on your PE router.

```
RP/0/RSP0/CPU0:PE1# show policy-map interface GigabitEthernet 0/0/0/0
```

GigabitEthernet0/0/0/0 input: ingress-all

Class Customers

| Classification statistics | (packets/bytes) | (rate - kbps) |
|---------------------------|-----------------|---------------|
| Matched :                 | 100/141800      | 66            |
| Transmitted :             | N/A             |               |
| Total Dropped :           | N/A             |               |

| Policing statistics   | (packets/bytes) | (rate - kbps) |
|-----------------------|-----------------|---------------|
| Policed(conform) :    | 92/130456       | 61            |
| Policed(exceed) :     | 0/0             | 0             |
| Policed(violate) :    | 0/0             | 0             |
| Policed and dropped : | 0/0             |               |

Policy Customer-policer Class Customer

| Classification statistics | (packets/bytes) | (rate - kbps) |
|---------------------------|-----------------|---------------|
| Matched :                 | 100/141800      | 66            |
| Transmitted :             | N/A             |               |
| Total Dropped :           | N/A             |               |

| Policing statistics   | (packets/bytes) | (rate - kbps) |
|-----------------------|-----------------|---------------|
| Policed(conform) :    | 68/96424        | 45            |
| Policed(exceed) :     | 24/34032        | 16            |
| Policed(violate) :    | 8/11344         | 5             |
| Policed and dropped : | 8/11344         |               |

Policed and dropped(parent policer) : N/A

Policy Customer-policer Class class-default

| Classification statistics | (packets/bytes) | (rate - kbps) |
|---------------------------|-----------------|---------------|
| Matched :                 | 0/0             | 0             |
| Transmitted :             | N/A             |               |
| Total Dropped :           | N/A             |               |

Class class-default

| Classification statistics | (packets/bytes) | (rate - kbps) |
|---------------------------|-----------------|---------------|
| Matched :                 | 0/0             | 0             |
| Transmitted :             | N/A             |               |
| Total Dropped :           | N/A             |               |

GigabitEthernet0/0/0/0 direction output: Service Policy not installed

PE2# **show policy-map interface GigabitEthernet 0/0/0**

GigabitEthernet0/0/0

Service-policy input: ingress-all

Class-map: Customers (match-any)

346 packets, 428584 bytes

5 minute offered rate 2000 bps, drop rate 0000 bps

Match: protocol ip

police:

rate 2000000 bps, burst 62500 bytes

conformed 341 packets, 421514 bytes; actions:

transmit

exceeded 2 packets, 2828 bytes; actions:

drop

conformed 2000 bps, exceeded 0000 bps

Service-policy : Customer-policer

```

Class-map: Customer (match-any)
 107 packets, 142126 bytes
 5 minute offered rate 0000 bps, drop rate 0000 bps
Match: access-group name Customer-podY-ACL
police:
 rate 50 %
 rate 1000000 bps, burst 31250 bytes)
 peak-rate 60 %
 peak-rate 1200000 bps, peak-burst 37500 bytes)
 conformed 89 packets, 116674 bytes; actions:
 transmit
 exceeded 15 packets, 21210 bytes; actions:
 set-dscp-transmit cs1
 violated 3 packets, 4242 bytes; actions:
 drop
 conformed 0000 bps, exceeded 0000 bps, violated 0000 bps

```

## Task 4: Configure LPTS Policing (IOS XR Software Only)

In this task, you will configure and verify LPTS hardware policing on the PE router running Cisco IOS XR Software.

### Activity Procedure

Complete these steps:

**Step 1** On your PE router, set these nondefault LPTS policing values:

- Fragment rate: 1000
- ICMP local rate: 500
- ICMP default rate: 1000
- HTTP known rate: 150
- HTTP default rate: 300

**Step 2** Configure the node for LPTS policing: 0/0/CPU0.

**Step 3** From the CE router, flood the PE router with 1000 ICMP echo requests using the **ping 10.x.1.1 repeat 800** command.

---

**Note** X is your pod number; 10.X.1.1 is the PE loopback address.

---

**Step 4** Verify that the number of policed ICMP local packets increases with each batch of the ICMP echo flood, using the **show lpts pifib hardware police location 0/0/CPU0** command.

**Step 5** From the CE router, flood the PE router with fragmented packets using the **ping 10.1.1.1 repeat 400 size 10000** command.

**Step 6** Verify that the number of policed fragmented packets increases with each flood.

## Activity Verification

You have completed this task when you attain these results:

- Verify that the number of policed ICMP local packets increases with each batch of the ICMP echo flood, using the **show lpts pifib hardware police location 0/0/CPU0** command.

```
RP/0/RSP0/CPU0:PE1# show lpts pifib hardware police location 0/0/CPU0
```

```
Wed Dec 7 16:12:59.085 UTC
```

```

Node 0/0/CPU0:

Burst = 100ms for all flow types

FlowType Policer Type Cur. Rate Def. Rate Accepted Dropped

ICMP-local 112 Static 1500 1500 4031 3
```

- From the CE router, flood the PE router with fragmented packets using the **ping 10.1.1.1 repeat 1000 size 18000** command.
- Verify that the number of policed fragmented packets increases with each flood command.

```
RP/0/RSP0/CPU0:PE1# show lpts pifib hardware police location 0/0/CPU0
```

```
Wed Dec 7 16:12:59.085 UTC
```

```

Node 0/0/CPU0:

Burst = 100ms for all flow types

FlowType Policer Type Cur. Rate Def. Rate Accepted Dropped

Fragment 101 Global 100 2500 29553 809
```

# Lab 6-2: Implement Traffic Shaping

Complete this lab activity to practice what you learned in the related module.

## Activity Objective

In this lab activity, you will configure and monitor class-based traffic shaping.

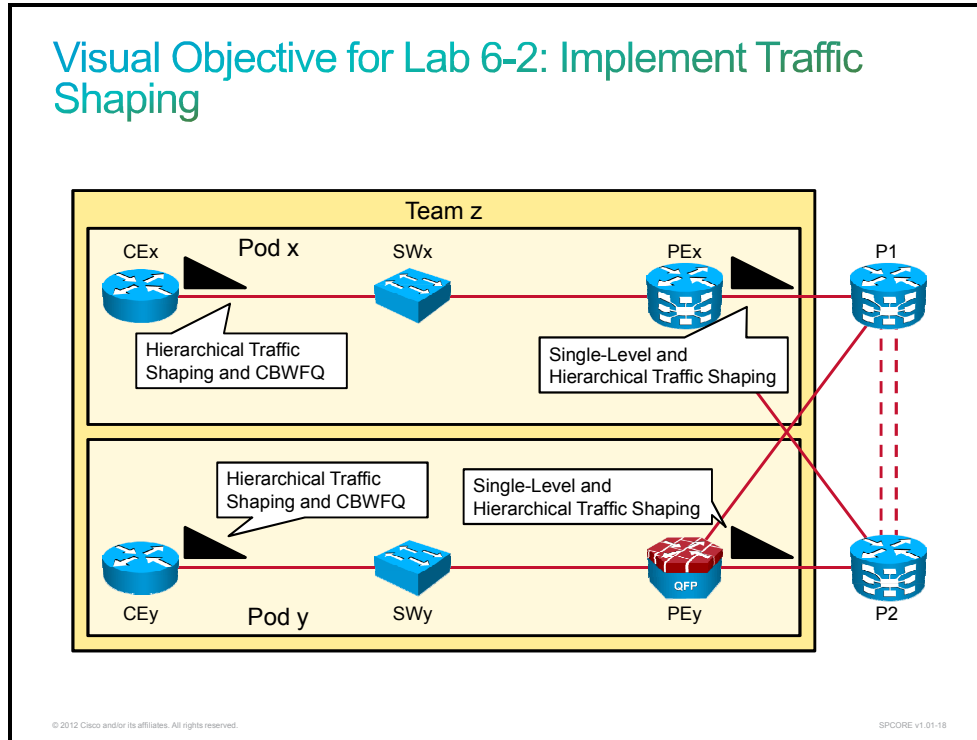
**Note** Students from two different pods will work in one team. The CE routers in both pods are running Cisco IOS Software. The first pod in the team will work on the PE router that is running Cisco IOS XR Software, and the second pod in the team will work on the PE router that is running Cisco IOS XE Software. Students in the same team should coordinate their lab activity.

In the lab activity, you will work on different Cisco routers running Cisco IOS (c2900), Cisco IOS XE (asr1001), and Cisco IOS XR (asr9k) Software. After completing this activity, you will be able to meet these objectives:

- Configure single-level class-based traffic shaping
- Monitor single-level class-based traffic shaping
- Configure hierarchical class-based traffic shaping
- Monitor hierarchical class-based traffic shaping

## Visual Objective

The figure illustrates what you will accomplish in this activity.



## Command List

The table describes the commands that are used in this lab activity.

### Cisco IOS, IOS XE, and IOS XR Policing Commands

| Command                                   | Description                                                                          |
|-------------------------------------------|--------------------------------------------------------------------------------------|
| <code>class-map</code>                    | Defines a class map                                                                  |
| <code>policy-map</code>                   | Defines a policy map                                                                 |
| <code>service-policy</code>               | Applies a policy map to an interface or a child-level policy map to the parent level |
| <code>shape average</code>                | Configures shaping to the average rate                                               |
| <code>conform-action</code>               | Defines the conform action of a policer                                              |
| <code>exceed-action</code>                | Defines the exceed action of a policer                                               |
| <code>violate-action</code>               | Defines the violate action of a policer                                              |
| <code>show policy-map interface</code>    | Displays the statistics of a service policy                                          |
| <code>clear qos counters interface</code> | Clears QoS interface counters in IOS XR Software                                     |
| <code>clear counters</code>               | Clears QoS interface counters in IOS and IOS XE Software                             |

## Task 1A: Configure Single-Level Traffic Shaping on IOS XR Software

In this task, you will configure and verify single-level traffic shaping on Cisco IOS XR Software.

### Activity Procedure

Complete these steps:

- Step 1** On your CEx router, send a traffic sample to the P1 router consisting of 100 1400-byte ICMP echoes. Use the **ping 10.0.1.1 repeat 100 size 1400** command. You should get immediate responses, because no throughput throttles are configured. The round-trip times should be 1/1/4 ms (minimum/average/maximum), respectively.
- Step 2** On your PEx router, configure an IPv4 access list that permits traffic from the uplink address of your CEx router or reuse the access list that you configured in the previous exercise. You may name the access list **Customer-podX-ACL**, where *X* is your pod number.
- Step 3** On your PEx router, create a traffic class that matches the access group defined in the previous step. You may name the class map **Customer** or reuse the class map that you configured in the previous exercise.
- Step 4** On your PEx, configure a shaping policy for the traffic class that was defined in the previous step. Use these parameters:
- Possible name: egress
  - Shape average rate: 500 KB/s
  - Location: Output direction on the core-facing interface

|                |                                                                                                                                                                                                                                                                                                                                                                                                                                |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Note</b>    | In a work situation, you will not configure shaping on core interfaces. In this exercise, you do so to avoid interference with other pods.                                                                                                                                                                                                                                                                                     |
| <b>Step 5</b>  | On your PEx router, clear the QoS counters on the core-facing interface using the <b>clear qos counters interface GigabitEthernet 0/0/0/2</b> command.                                                                                                                                                                                                                                                                         |
| <b>Step 6</b>  | On your CEx router, resend the ICMP echo batch using the <b>ping 10.0.1.1 repeat 100 size 1400</b> command. You should observe longer round-trip times (1/23/28 ms minimum/average/maximum).                                                                                                                                                                                                                                   |
| <b>Note</b>    | You must ping the nearest core router to be able to match the traffic using an IPv4 access list. If you pinged the farther core router, the traffic on the core link would be labeled and therefore not matched by the ACL.                                                                                                                                                                                                    |
| <b>Step 7</b>  | Verify the shaping statistics using the <b>show policy-map interface GigabitEthernet 0/0/0/2</b> command. You should see that the queue packet counters (queue conform and queue exceed on IOS XR Software) for the class Customer increment.                                                                                                                                                                                  |
| <b>Note</b>    | You can disregard class-default, because it catches control traffic and not the ICMP flood.                                                                                                                                                                                                                                                                                                                                    |
| <b>Step 8</b>  | From the CEx router, flood the network with larger packets using the <b>ping 10.0.1.1 repeat 100 size 9000</b> command to cause the shaper to drop some packets. You should get approximate round-trip values (minimum/average/maximum) of 16/86/164 ms and a success rate of about 69 percent. Check the policing statistics on the PE router. You should now see tail drops in addition to conform and exceed queue packets. |
| <b>Step 9</b>  | On the PEx router, lower the excess burst to 1 ms. Keep the same average rate.                                                                                                                                                                                                                                                                                                                                                 |
| <b>Step 10</b> | From the CEx router, flood the network again ( <b>ping 10.0.1.1 repeat 100 size 9000</b> ). Examine the results. You should get approximately a 51 percent success rate, round-trip times minimum/average/maximum of 128/128/164 ms, and see respective amounts of queued and tail-dropped packets on the PEx router.                                                                                                          |
| <b>Step 11</b> | On the PEx router, increase the excess burst to 1000 ms. Keep the same average rate.                                                                                                                                                                                                                                                                                                                                           |
| <b>Step 12</b> | From the CEx router, flood the network again ( <b>ping 10.0.1.1 repeat 100 size 9000</b> ). You should get approximately a 90 percent success rate, round-trip times (minimum/average/maximum) of 1/37/168 ms, and see corresponding shaper statistics.                                                                                                                                                                        |
| <b>Step 13</b> | Remove the output service policy from the core-facing interface.                                                                                                                                                                                                                                                                                                                                                               |

## Activity Verification

You have completed this task when you attain these results:

- On your CEx router, send a traffic sample to the P1 router consisting of 100 1400-byte ICMP echoes. Use the **ping 10.0.1.1 repeat 100 size 1400** command. You should get immediate responses, because no throughput throttles are configured.

```
CE1# ping 10.0.1.1 repeat 100 size 1400
Type escape sequence to abort.
Sending 100, 1400-byte ICMP Echos to 10.0.1.1, timeout is 2 seconds:
!!
!!
Success rate is 100 percent (100/100), round-trip min/avg/max = 1/1/4
ms
```

- Verify the shaping statistics using the **show policy-map interface GigabitEthernet 0/0/0/2** command. You should see that the queue packet counters (queue conform and queue exceed on IOS XR Software) for the class Customer increment.

```
RP/0/RSP0/CPU0:PEX#show policy-map int GigabitEthernet 0/0/0/2
GigabitEthernet0/0/0/2 direction input: Service Policy not installed
GigabitEthernet0/0/0/2 output: egress
```

```
Class Customer
 Classification statistics (packets/bytes) (rate - kbps)
 Matched : 100/141800 0
 Transmitted : 38/53884 0
 Total Dropped : 0/0 0
 Queueing statistics
 Queue ID : 266
 High watermark (Unknown)
 Inst-queue-len (packets) : 0
 Avg-queue-len (Unknown)
 Taildropped (packets/bytes) : 0/0
 Queue (conform) : 21/29778 0
 Queue (exceed) : 17/24106 0
 RED random drops (packets/bytes) : 0/0
```

```
Class class-default
<output omitted>
```

- From the CEx router, flood the network with larger packets using the **ping 10.0.1.1 repeat 100 size 9000** command to cause the shaper to drop some packets.

```
CE1# ping 10.0.1.1 repeat 100 size 9000
Type escape sequence to abort.
Sending 100, 9000-byte ICMP Echos to 10.0.1.1, timeout is 2 seconds:
!!
..!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Success rate is 65 percent (65/100), round-trip min/avg/max =
16/86/164 ms
```

- Check the policing statistics on the PEx router. You should now see tail drops in addition to conform and exceed queue packets.

```
RP/0/RSP0/CPU0:PE1#show policy-map int GigabitEthernet 0/0/0/2
Thu Dec 8 14:03:26.439 UTC
GigabitEthernet0/0/0/2 direction input: Service Policy not installed

GigabitEthernet0/0/0/2 output: egress
```

```
Class Customer
 Classification statistics (packets/bytes) (rate - kbps)
 Matched : 4271/2212074 3
 Transmitted : 4203/2201370 3
 Total Dropped : 68/10704 0
```

```

Queueing statistics
 Queue ID : 266
 High watermark (Unknown)
 Inst-queue-len (packets) : 0
 Avg-queue-len (Unknown)
 Taildropped (packets/bytes) : 68/10704
 Queue (conform) : 3264/1148224 3
 Queue (exceed) : 939/1053146 0
 RED random drops (packets/bytes) : 0/0

```

- From the CEx router, flood the network again (**ping 10.0.1.1 repeat 100 size 9000**) when lowering the excess burst to 1 ms.

```

CE1#ping 10.0.1.1 repeat 100 size 9000
Type escape sequence to abort.
Sending 100, 9000-byte ICMP Echos to 10.0.1.1, timeout is 2 seconds:
!!
!!
Success rate is 51 percent (51/100), round-trip min/avg/max =
128/128/164 ms

```

- From the CEx router, flood the network again (**ping 10.0.1.1 repeat 100 size 9000**) when increasing the excess burst to 1000 ms.

```

CE1# ping 10.0.1.1 repeat 100 size 9000
Type escape sequence to abort.
Sending 100, 9000-byte ICMP Echos to 10.0.1.1, timeout is 2 seconds:
!!
!!
Success rate is 90 percent (90/100), round-trip min/avg/max = 1/37/168
ms

```

- Check the policing statistics on the PEx router when increasing the excess burst to 1000 ms.

```

RP/0/RSP0/CPU0:PE1#show policy-map int GigabitEthernet 0/0/0/2
Thu Dec 8 14:09:43.071 UTC
GigabitEthernet0/0/0/2 direction input: Service Policy not installed

GigabitEthernet0/0/0/2 output: egress

```

```

Class Customer
 Classification statistics (packets/bytes) (rate - kbps)
 Matched : 1253/968726 3
 Transmitted : 1239/962714 3
 Total Dropped : 14/6012 0
 Queueing statistics
 Queue ID : 266
 High watermark (Unknown)
 Inst-queue-len (packets) : 0
 Avg-queue-len (Unknown)
 Taildropped (packets/bytes) : 14/6012
 Queue (conform) : 643/202994 3
 Queue (exceed) : 596/759720 0
 RED random drops (packets/bytes) : 0/0

```

# Task 1B: Configure Single-Level Traffic Shaping on IOS XE Software

In this task, you will configure and verify single-level traffic shaping on Cisco IOS XE Software.

## Activity Procedure

Complete these steps:

- Step 1** On your CEy router, send a traffic sample to the P2 router consisting of 100 1400-byte ICMP echoes. Use the **ping 10.0.2.1 repeat 100 size 1400** command. You should get immediate responses, because no throughput throttles are configured. The round-trip times should be 1/1/4 ms (minimum/average/maximum).
- Step 2** On your PEy router, configure an IPv4 access list that permits traffic from the uplink address of your CEy router or reuse the access list that you configured in the previous exercise. You may name the access list **Customer-podY-ACL**, where *Y* is your pod number.
- Step 3** On your PEy router, create a traffic class that matches the access group defined in the previous step. You may name the class map **Customer** or reuse the class map that you configured in the previous exercise.
- Step 4** On your PEy router, configure a shaping policy for the traffic class defined in the previous step. Use these parameters:
- Possible name: egress
  - Shape average rate: 500 KB/s
  - Location: Output direction on the core-facing interface

---

**Note** In a work situation, you will not configure shaping on core interfaces. In this exercise, you do so to avoid interference with other pods.

---

**Step 5** On your PEy router, clear the QoS counters on the core-facing interface using the **clear counters Gigabit 0/0/3** command.

**Step 6** On your CEy router, resend the ICMP echo batch using the **ping 10.0.2.1 repeat 100 size 1400** command. You should observe longer round-trip times (1/22/48 ms).

---

**Note** You must ping the nearest core router to be able to match the traffic using an IPv4 access list. If you pinged the farther core router, the traffic on the core link would be labeled and therefore not matched by the ACL.

---

**Step 7** Verify the shaping statistics using the **show policy-map interface GigabitEthernet 0/0/3**. You should see that the queue packet counters for the class Customer increment.

```
PEY#show policy-map interface GigabitEthernet 0/0/3
GigabitEthernet0/0/3
```

```
Service-policy output: egress
```

```
Class-map: Customer (match-any)
 100 packets, 141400 bytes
```

```

5 minute offered rate 0000 bps, drop rate 0000 bps
Match: access-group name Customer-podY-ACL
Queueing
queue limit 64 packets
(queue depth/total drops/no-buffer drops) 0/0/0
(pkts output/bytes output) 100/141400
shape (average) cir 500000, bc 2000, be 2000
target shape rate 500000

```

```

Class-map: class-default (match-any)
<output truncated>

```

---

**Note** You can disregard class-default, because it catches control traffic and not the ICMP flood.

---

**Step 8** Remove the output service policy from the core-facing interface.

---

**Note** You will not examine the shaping behavior with different burst sizes, because Cisco IOS and IOS XE Software do not offer such configuration options.

---

## Activity Verification

You have completed this task when you attain these results:

- On your CEy router, resend the ICMP echo batch using the **ping 10.0.2.1 repeat 100 size 1400** command when shaping is enabled. You should observe longer round-trip times (1/22/48 ms).

```

CE2# ping 10.0.2.1 repeat 100 size 1400
Type escape sequence to abort.
Sending 100, 1400-byte ICMP Echos to 10.0.2.1, timeout is 2 seconds:
!!
!!
Success rate is 100 percent (100/100), round-trip min/avg/max =
1/22/48 ms

```

- Verify the shaping statistics using the **show policy-map interface GigabitEthernet 0/0/3**. You should see that the queue packet counters for the class Customer increment.

```

PE2#show policy-map interface GigabitEthernet 0/0/3
GigabitEthernet0/0/3

```

Service-policy output: egress

```

Class-map: Customer (match-any)
 100 packets, 141400 bytes
 5 minute offered rate 0000 bps, drop rate 0000 bps
Match: access-group name Customer-podY-ACL
Queueing
queue limit 64 packets
(queue depth/total drops/no-buffer drops) 0/0/0
(pkts output/bytes output) 100/141400
shape (average) cir 500000, bc 2000, be 2000
target shape rate 500000

```

## Task 2: Configure Hierarchical Traffic Shaping with CBWFQ on the Child Level

In this task, you will configure and verify hierarchical traffic shaping with CBWFQ on the child level. You will configure this functionality on your CE router running Cisco IOS.

### Activity Procedure

Complete these steps:

- Step 1** On your CEy router, send a traffic sample to a core router consisting of 100 10000-byte ICMP echoes. Use the **ping 10.0.1.1 repeat 100 size 10000** command. You should get immediate responses, because no throughput throttles are configured.
- Step 2** On your CEy router, configure two traffic classes, matched by IPv4 access lists:
- Class **External** matching **External-podX-ACL** that permits ICMPv4 traffic sourced from the uplink GigabitEthernet interface.
  - Class **Internal** matching **Internal-podX-ACL** that permits ICMPv4 traffic sourced from the Loopback0 interface.

---

**Note** Internal traffic will get higher bandwidth guarantee than external traffic.

---

- Step 3** Configure a child-level policy (**child-cbwfq**) that implements CBWFQ to guarantee 80 percent of the available bandwidth to the Internal class and 10 percent to the External class.
- Step 4** Define a parent-level policy (**shape-all**) that shapes all traffic from the default class using the average rate of 100 KB/s. Apply the child-level CBWFQ policy to the parent policy.
- Step 5** Apply the parent-level policy to the uplink interface in output direction.
- Step 6** From your PEy router, use Telnet to connect to the CEy router. This will establish a second session, in addition to the console connection. You need two sessions to compare the QoS guarantees assigned to the Internal and External traffic.

---

**Note** You may have to remove an ACL filter from the VTY lines and create a local user to establish a Telnet session to the CE router.

---

- Step 7** In each session, launch an ICMP flood with appropriate source address:
- **ping 10.0.2.1 repeat 100 size 10000**
  - **ping 10.0.2.1 repeat 100 size 10000 source loopback 0**
- Step 8** Observe how much bandwidth is available to each traffic class. Observe how much bandwidth is used if only one traffic type is being transmitted.

---

**Note** CBWFQ allocates the guaranteed amounts of traffic only if there is congestion within the shaped rate. If there is no congestion, the active flow gets all available bandwidth.

---

- Step 9** Remove the service policy from the CEy router uplink interface.

## Activity Verification

You have completed this task when you attain these results:

- You have verified that the transmission rate of the ICMP packets sourced from the loopback interface is eight times higher than the external traffic rate when both traffic classes are competing for bandwidth.

## Task 3: Configure Hierarchical Traffic Shaping with Shaping on the Child Level

In this task, you will configure and verify hierarchical traffic shaping on the PE routers. You will shape traffic both on the parent and on the child level.

### Activity Procedure

Complete these steps:

- Step 1** On your pod CE router, send a traffic sample to the nearest core router consisting of 100 5000-byte ICMP echoes. Use the **ping 10.0.X.1 repeat 100 size 5000** (or **ping 10.0.Y.1 repeat 100 size 5000**) command. You should get immediate responses, because no throughput throttles have been configured.
- Step 2** On your PE router, configure two traffic classes, matched by IPv4 access lists:
- Class **External** matching **External-podX-ACL** that permits ICMPv4 traffic sourced from the CE uplink GigabitEthernet interface.
  - Class **Internal** matching **Internal-podX-ACL** that permits ICMPv4 traffic sourced from the CE Loopback0 interface.

---

**Note** The ACLs and class maps are almost identical to those that were configured in the previous task, but the syntax on IOS XR Software is slightly different.

---

- Step 3** Configure a child-level policy (**child-shaper**) that implements shaping to shape the internal class to 80 percent of the parent bandwidth and shape the external class to 10 percent of the parent bandwidth.
- Step 4** Define a parent-level policy (**shape-all**) that shapes all traffic from the default class using the average rate of 700 KB/s. Apply the child-level shaper policy to the parent policy.
- Step 5** Apply the parent-level policy to the core interface in the output direction.
- Step 6** From your CE router, launch two ICMP floods with the appropriate source addresses, one at a time (consecutively):
- **ping 10.0.1.1 repeat 100 size 5000**
  - **ping 10.0.1.1 repeat 100 size 5000 source loopback 0**
- Step 7** Observe how much bandwidth is available to each traffic class.

---

**Note** The difference between CBWFQ and shaping on the child level is that CBWFQ limits the subclass bandwidth only if there is congestion. Shaping limits the subclass bandwidth even without congestion.

---

## Activity Verification

You have completed this task when you attain these results:

- You have verified that the transmission rate of the ICMP packets sourced from the loopback interface is eight times higher than the external traffic rate, even when the traffic belongs to a single class.
- You have examined the shaping statistics using appropriate **show** commands.
- At the end, you have removed the service policy from the PE router core interface.

# Answer Key

The correct answers and expected solutions for the lab activities that are described in this guide appear here.

## Lab 1-1 Answer Key: Implement MPLS in the Service Provider Core

When you complete this lab activity, device configuration and device outputs will be similar to the results shown here, with differences that are specific to your pod.

### Task 1: Configure and Verify OSPF

There are no configuration tasks in this task.

### Task 2: Configure Basic MPLS

**Step 1** Enable MPLS for interfaces on the path PEx – P1 – P2 – PEy and set the LDP router ID for all routers on that path to the IP address of the loopback:

PE1 (Cisco IOS XR):

```
mpls ldp
router-id 10.1.1.1
interface GigabitEthernet0/0/0/2
!
commit
```

PE2 (Cisco IOS XE):

```
mpls ldp router-id Loopback0
interface GigabitEthernet0/0/3
mpls ip
end
```

P1 (Cisco IOS XR):

```
mpls ldp
router-id 10.0.1.1
!
interface GigabitEthernet0/0/0/0
!
interface GigabitEthernet0/0/0/1
!
interface GigabitEthernet0/0/0/2
!
interface GigabitEthernet0/0/0/3
!
interface GigabitEthernet0/0/0/4
!
interface GigabitEthernet0/0/0/5
!
interface GigabitEthernet0/0/0/6
!
interface GigabitEthernet0/0/0/7
!
interface GigabitEthernet0/0/0/8
!
interface GigabitEthernet0/0/0/9
```

```

!
commit
P2 (Cisco IOS XR):
mpls ldp
router-id 10.0.2.1
!
interface GigabitEthernet0/0/0/0
!
interface GigabitEthernet0/0/0/1
!
interface GigabitEthernet0/0/0/2
!
interface GigabitEthernet0/0/0/3
!
interface GigabitEthernet0/0/0/4
!
interface GigabitEthernet0/0/0/5
!
interface GigabitEthernet0/0/0/6
!
interface GigabitEthernet0/0/0/7
!
interface GigabitEthernet0/0/0/8
!
interface GigabitEthernet0/0/0/9
!
commit

```

### Task 3: Disable IP TTL Propagation

**Step 1** Hide the P1 and P2 core routers from a **traceroute** command only for forwarded packets by disabling IP TTL propagation:

PE1:

```

!
mpls ip-ttl-propagate disable forwarded
commit
!
no mpls ip-ttl-propagate disable forwarded
commit
!

```

PE2:

```

no mpls ip propagate-ttl forwarded
!
mpls ip propagate-tt
!

```

### Task 4: Configure Conditional Label Distribution

**Step 1** Configure conditional label distribution on PE1 router:

PE1:

```

mpls ldp
 label
 advertise
 disable
 for PFX to PEER
 !
 !
!
ipv4 access-list PFX
 10 permit ipv4 host 10.1.1.1 any
!
ipv4 access-list PEER
 10 permit ipv4 any any
!

```

**Step 2** Configure conditional label distribution on PE2 router:

PE2:

```

no mpls ldp advertise-labels
mpls ldp advertise-labels for 20 to 21
!
access-list 20 permit 10.2.1.1
access-list 21 permit any

```

## Lab 2-1 Answer Key: Implement MPLS TE

When you complete this lab activity, device configuration and device outputs will be similar to the results shown here, with differences that are specific to your pod.

### Task 1: Log Existing Traffic Flow

**Step 1** Enable interface GigabitEthernet0/0/0/3 on your pod router PEx and interface GigabitEthernet0/0/0/2 on your pod router PEy:

PE1 (Cisco IOS XR):

```

interface GigabitEthernet 0/0/0/3
 no shutdown
!
commit

```

PE2 (Cisco IOS XE):

```

interface GigabitEthernet 0/0/2
 no shutdown
!

```

**Step 2** Enable MPLS on interface GigabitEthernet0/0/0/3 on your pod router PEx and interface GigabitEthernet0/0/0/2 on your pod router PEy:

PE1 (Cisco IOS XR):

```

mpls ldp
interface GigabitEthernet0/0/0/3
!
commit

```

PE2 (Cisco IOS XE):

```

interface GigabitEthernet0/0/2

```

```
mpls ip
end
```

**Step 3** If it is not already enabled, enable MPLS on all interfaces of core devices P1 and P2 connecting to your PE routers:

**P1 (Cisco IOS XR):**

```
mpls ldp
interface GigabitEthernet0/0/0/0
!
interface GigabitEthernet0/0/0/1
!
interface GigabitEthernet0/0/0/2
!
interface GigabitEthernet0/0/0/3
!
interface GigabitEthernet0/0/0/4
!
interface GigabitEthernet0/0/0/5
!
interface GigabitEthernet0/0/0/6
!
interface GigabitEthernet0/0/0/7
!
interface GigabitEthernet0/0/0/8
!
interface GigabitEthernet0/0/0/9
!
commit
```

**P2 (Cisco IOS XR):**

```
mpls ldp
interface GigabitEthernet0/0/0/0
!
interface GigabitEthernet0/0/0/1
!
interface GigabitEthernet0/0/0/2
!
interface GigabitEthernet0/0/0/3
!
interface GigabitEthernet0/0/0/4
!
interface GigabitEthernet0/0/0/5
!
interface GigabitEthernet0/0/0/6
!
interface GigabitEthernet0/0/0/7
!
interface GigabitEthernet0/0/0/8
!
interface GigabitEthernet0/0/0/9
!
Commit
```

## Task 2: Configure Support for MPLS TE

- Step 1** Enable MPLS TE functionality on all interfaces of core devices P1 and P2 connecting to your PE routers. Enable MPLS functionality also for the P1-P2 interconnection:

P1 (Cisco IOS XR):

```
mpls traffic-eng
interface GigabitEthernet0/0/0/1
!
interface GigabitEthernet0/0/0/2
!
interface GigabitEthernet0/0/0/3
!
interface GigabitEthernet0/0/0/4
!
interface GigabitEthernet0/0/0/5
!
interface GigabitEthernet0/0/0/6
!
interface GigabitEthernet0/0/0/7
!
interface GigabitEthernet0/0/0/8
!
interface GigabitEthernet0/0/0/9
!
commit
```

P2 (Cisco IOS XR):

```
mpls traffic-eng
interface GigabitEthernet0/0/0/1
!
interface GigabitEthernet0/0/0/2
!
interface GigabitEthernet0/0/0/3
!
interface GigabitEthernet0/0/0/4
!
interface GigabitEthernet0/0/0/5
!
interface GigabitEthernet0/0/0/6
!
interface GigabitEthernet0/0/0/7
!
interface GigabitEthernet0/0/0/8
!
interface GigabitEthernet0/0/0/9
!
commit
```

- Step 2** Enable MPLS TE functionality on all interfaces of the PEx and PEy routers connecting to the core devices P1 and P2:

PE1 (Cisco IOS XR):

```

mpls traffic-eng
interface GigabitEthernet0/0/0/2
!
interface GigabitEthernet0/0/0/3
!
!
commit

```

**PE2 (Cisco IOS XE):**

```

mpls traffic-eng tunnels
interface GigabitEthernet0/0/2
 mpls traffic-eng tunnels
!
interface GigabitEthernet0/0/3
 mpls traffic-eng tunnels
!

```

**Step 3** Enable RSVP on all interfaces participating in MPLS TE. Set the maximum amount of bandwidth that may be allocated by RSVP flows to 1 Mb/s and the maximum amount of bandwidth that may be allocated to a single flow to 200 kb/s.

**P1 (Cisco IOS XR):**

```

rsvp
interface GigabitEthernet0/0/0/1
 bandwidth 2000 200
!
interface GigabitEthernet0/0/0/2
 bandwidth 2000 200
!
interface GigabitEthernet0/0/0/3
 bandwidth 2000 200
!
interface GigabitEthernet0/0/0/4
 bandwidth 2000 200
!
interface GigabitEthernet0/0/0/5
 bandwidth 2000 200
!
interface GigabitEthernet0/0/0/6
 bandwidth 2000 200
!
interface GigabitEthernet0/0/0/7
 bandwidth 2000 200
!
interface GigabitEthernet0/0/0/8
 bandwidth 2000 200
!
interface GigabitEthernet0/0/0/9
 bandwidth 2000 200
!
!
commit

```

## P2 (Cisco IOS XR):

```
rsvp
 interface GigabitEthernet0/0/0/1
 bandwidth 2000 200
 !
 interface GigabitEthernet0/0/0/2
 bandwidth 2000 200
 !
 interface GigabitEthernet0/0/0/3
 bandwidth 2000 200
 !
 interface GigabitEthernet0/0/0/4
 bandwidth 2000 200
 !
 interface GigabitEthernet0/0/0/5
 bandwidth 2000 200
 !
 interface GigabitEthernet0/0/0/6
 bandwidth 2000 200
 !
 interface GigabitEthernet0/0/0/7
 bandwidth 2000 200
 !
 interface GigabitEthernet0/0/0/8
 bandwidth 2000 200
 !
 interface GigabitEthernet0/0/0/9
 bandwidth 2000 200
 !
 !
commit
```

## PE1 (Cisco IOS XR):

```
rsvp
 interface GigabitEthernet0/0/0/2
 bandwidth 2000 200
 !
 interface GigabitEthernet0/0/0/3
 bandwidth 2000 200
 !
```

```
commit
```

PE2 (Cisco IOS XE):

```
interface GigabitEthernet0/0/2
```

```
ip rsvp bandwidth 2000 200
```

```
!
```

```
interface GigabitEthernet0/0/3
```

```
ip rsvp bandwidth 2000 200
```

```
!
```

**Step 4** Enable MPLS TE support in the OSPF routing protocol on all routers participating in MPLS TE. Specify that the traffic engineering router identifier for all nodes is the IP address associated with the Loopback0 interface.

P1 (Cisco IOS XR):

```
router ospf 1
```

```
mpls traffic-eng router-id Loopback0
```

```
area 0
```

```
mpls traffic-eng
```

```
!
```

```
!
```

```
commit
```

P2 (Cisco IOS XR):

```
router ospf 1
```

```
mpls traffic-eng router-id Loopback0
```

```
area 0
```

```
mpls traffic-eng
```

```
!
```

```
!
```

```
commit
```

PE1 (Cisco IOS XR):

```
router ospf 1
```

```
mpls traffic-eng router-id Loopback0
```

```
area 0
```

```
mpls traffic-eng
```

```
!
```

```
!
```

```
commit
```

PE2 (Cisco IOS XE):

```
router ospf 1
```

```
mpls traffic-eng router-id Loopback0
```

```
mpls traffic-eng area 0
```

```
!
```

### Task 3: Build an MPLS TE Tunnel

**Step 1** Build an MPLS TE tunnel between routers PEx and PEy:

PE1 (Cisco IOS XR):

```
interface tunnel-te1
 ipv4 unnumbered Loopback0
 signalled-bandwidth 100
 autoroute announce
 destination 10.2.1.1
 path-option 1 explicit name Explicit
!
explicit-path name Explicit
 index 1 next-address strict ipv4 unicast 10.0.1.1
 index 2 next-address strict ipv4 unicast 10.0.2.1
 index 3 next-address strict ipv4 unicast 10.2.1.1
!
commit
```

**Step 2** Build an MPLS TE tunnel between routers PEy and PEx:

PE2 (Cisco IOS XE):

```
interface Tunnel1
 ip unnumbered Loopback0
 tunnel mode mpls traffic-eng
 tunnel destination 10.1.1.1
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng priority 0 0
 tunnel mpls traffic-eng bandwidth 100
 tunnel mpls traffic-eng path-option 1 explicit name Explicit
!
ip explicit-path name Explicit
 next-address 10.0.2.1
 next-address 10.0.1.1
 next-address 10.1.1.1
!
```

### Task 4: Configure a Backup Tunnel

**Step 1** Enable the fast reroute feature for the existing primary MPLS TE tunnel on your PE router:

PE1:

```
interface tunnel-te1
 fast-reroute
!
```

PE2:

```
interface Tunnel1
 tunnel mpls traffic-eng fast-reroute
end
```

**Step 2** Configure a backup MPLS TE tunnel on your PE router for the existing primary MPLS TE tunnel:

PE1:

```
interface tunnel-te1000
```

```

ipv4 unnumbered Loopback0
signalled-bandwidth 500
destination 10.2.1.1
path-option 1 explicit name backup
!
explicit-path name backup
index 1 next-address strict ipv4 unicast 10.0.2.1
index 2 next-address strict ipv4 unicast 10.2.1.1
!
mpls traffic-eng
interface GigabitEthernet0/0/0/2
backup-path tunnel-te 1000
!
!
PE2:
interface Tunnel1000
ip unnumbered Loopback0
tunnel mode mpls traffic-eng
tunnel destination 10.1.1.1
tunnel mpls traffic-eng bandwidth 500
tunnel mpls traffic-eng path-option 1 explicit name backup
!
ip explicit-path name backup enable
next-address 10.0.1.1
next-address 10.1.1.1
!
interface GigabitEthernet0/0/3
mpls traffic-eng backup-path Tunnel1000
!

```

## Lab 3-1 Answer Key: Use QoS Mechanisms

When you complete this activity, your configuration will be similar to the results here, with differences that are specific to your device or workgroup.

### Task 1: Explore Network Topology and Test Network Connectivity

The answer key of this task is part of verification activity. The table with IP subnets between routers should look like the one shown here. (This example is for PE1.)

|    | CE               | P1              | P2              |
|----|------------------|-----------------|-----------------|
| PE | 192.168.101.0/24 | 192.168.11.0/24 | 192.168.12.0/24 |

### Task 2: Monitor Interface Statistics Before and After Network Congestion

During this task, you need to enter the following commands:

**Step 1** Clear the interface statistics on the P router-facing interface on the PE router.

(Cisco IOS XR)

```
clear counters gigabitethernet 0/0/0/2
```

(Cisco IOS XE)

```
clear counters gigabitethernet 0/0/3
```

**Step 2** Answer these questions:

Q: How does round-trip time differ before and after applying the traffic shaping policy?

A: Round-trip time is increased after applying the traffic shaping policy.

**Step 3** Answer these questions:

Q: How do the two traffic flows affect the round-trip time and success rate in the case of traffic shaping?

A: Traffic has longer round-trip time because two ping flows pass through the same interface on which the policy is applied.

**Step 4** On the PE router, clear the interface statistics counters on the P router-facing interface.

(Cisco IOS XR)

```
clear counters gigabitEthernet 0/0/0/2
```

(Cisco IOS XE)

```
clear counters gigabitethernet 0/0/3
```

**Step 5** Edit policy on PE router:

(Cisco IOS XR)

```
policy-map shapeIPv4
class IPv4traffic
no shape average 64 kbps
police rate 64 kbps
end
```

(Cisco IOS XE)

```
policy-map shapeIPv4
class IPv4traffic
no shape average 64000
police rate 64000
end
```

**Step 6** Answer these questions:

Q: What is the effect on the round-trip time and success rate in the case of traffic policing?

A: The success rate is below 100 percent and the round-trip time for transmitted packets is decreased.

**Step 7** Answer these questions:

Q: How does traffic policing affect the packet drops on the output interface?

A: Traffic policing discards packets exceeding the preconfigured rate, therefore the number of interface drops will increase.

**Step 8** Remove the service policy from the interface, and clear the policy map and class map:

(Cisco IOS XR)

```
configure terminal
```

```

interface gigabitethernet 0/0/0/2
 no service-policy output shapeIPv4
 exit
no policy-map shapeIPv4
no class-map IPv4traffic
end
copy running-config startup-config

```

(Cisco IOS XE)

```

configure terminal
interface gigabitethernet 0/0/3
 no service-policy output shapeIPv4
 exit
no policy-map shapeIPv4
no class-map IPv4traffic
end

```

## Lab 3-2 Answer Key: Implement MPLS Support for QoS

When you complete this activity, your configuration will be similar to the results here, with differences that are specific to your device or workgroup:

### Task 1: Update Existing Tunnel Parameters

During this task, you need to enter the following commands:

(Cisco IOS and IOS XE)

```

configure terminal
interface tunnell
 tunnel mpls traffic-eng priority 0 0
 tunnel mpls traffic-eng bandwidth 40
!
no interface tunnel 1000

```

(Cisco IOS XR)

```

configure terminal
interface tunnel-te1
 priority 0 0
 signalled-bandwidth 40
!
no interface tunnel-te 1000
!

```

### Task 2: Create a DS-TE Tunnel

During this task, you need to enter the following commands:

(Cisco IOS and IOS XE)

```

configure terminal
ip explicit-path name Over_P2 enable
next-address 10.0.2.1
next-address 10.1.1.1

```

```

interface Tunnel2
 ip unnumbered Loopback0
 tunnel mode mpls traffic-eng
 tunnel destination 10.1.1.1
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng priority 0 0
 tunnel mpls traffic-eng bandwidth 50
 tunnel mpls traffic-eng path-option 1 explicit name Over_P2

```

(Cisco IOS XR)

```

configure terminal
explicit-path name Over_P2
 index 1 next-address strict ipv4 unicast 10.0.2.1
 index 2 next-address strict ipv4 unicast 10.2.1.1
exit
interface tunnel-te2
 ipv4 unnumbered Loopback0
 priority 0 0
 signalled-bandwidth 50
 autoroute announce
 destination 10.2.1.1
 path-option 1 explicit name Over_P2

```

### Task 3: Assign the Traffic to the Tunnels

During this task, you need to enter the following commands:

(Cisco IOS and IOS XE only)

```

configure terminal
access-list 100 permit ip any any dscp ef
route-map SelectTunnel permit 10
 match ip address 100
 set interface Tunnel1
route-map SelectTunnel permit 20
 set interface Tunnel2
exit
interface gigabitethernet 0/0/0
ip policy route-map SelectTunnel
!

interface gigabitethernet 0/0/0
no ip policy route-map SelectTunnel
!
interface tunnel 2
shutdown
!

```

## Lab 4-1 Answer Key: Implement Classification and Marking

When you complete this activity, your configuration will be similar to the results here, with differences that are specific to your device or workgroup:

### Task 1: Configure MQC Classification

During this task, you need to enter the following commands:

CE1 and CE2:

```
configure terminal
 ip access-list extended GRE
 permit gre any any
 ip access-list extended ICMP
 permit icmp any any
 !
 class-map match-all ICMP
 match access-group name ICMP
 class-map match-all GRE
 match access-group name GRE
```

### Task 2: Configure Class-Based Marking

During this task, you need to enter the following commands:

CE1 and CE2:

```
configure terminal
 policy-map MARK_GRE_ICMP
 class GRE
 set ip precedence 6
 class ICMP
 set ip precedence 5
 interface GigabitEthernet0/0
 service-policy output MARK_GRE_ICMP
```

### Task 3: Configure a GRE Tunnel

During this task, you need to enter the following commands:

CE1:

```
configure terminal
 interface Tunnel5
 ip address 10.12.12.1 255.255.255.0
 tunnel source GigabitEthernet0/0
 tunnel destination 192.168.102.21
 ip route 10.2.10.1 255.255.255.255 Tunnel 5
```

CE2:

```
configure terminal
 interface Tunnel5
 ip address 10.12.12.2 255.255.255.0
 tunnel source GigabitEthernet0/0
 tunnel destination 192.168.101.11
```

```
ip route 10.1.10.1 255.255.255.255 Tunnel 5
```

#### Task 4: Configure QoS Preclassify

During this task, you need to enter the following commands:

CE1 and CE2:

```
configure terminal
interface tunnel 5
qos pre-classify
```

#### Task 5: Monitor QoS Preclassify

CE1:

```
configure terminal
no ip route 10.2.10.1 255.255.255.255 Tunnel 5
!
```

CE2:

```
configure terminal
no ip route 10.1.10.1 255.255.255.255 Tunnel5
!
```

### Lab 5-1 Answer Key: Implement Queuing

When you complete this lab activity, device configuration and device outputs will be similar to the results shown here, with differences that are specific to your pod.

#### Task 2A: Configure CBWFQ on IOS XR Software

```
ipv4 access-list External-podX-ACL
10 permit icmp 192.168.10X.0/24 any
!
ipv4 access-list Internal-podX-ACL
10 permit icmp 10.X.10.0/24 any
!
class-map match-any External
match access-group ipv4 External-podX-ACL
!
class-map match-any Internal
match access-group ipv4 Internal-podX-ACL
!
class-map match-any ef-class
match dscp ef
!
policy-map cbwfq-policy
class Internal
bandwidth percent 60
!
class ef-class
```

```

 bandwidth percent 35
!
policy-map shape-all
 class class-default
 service-policy cbwfq-policy
 shape average 200 kbps
!
interface GigabitEthernet0/0/0/2
 service-policy output shape-all
commit

```

## Hierarchical CBWFQ

```

class-map match-any Mission-critical
 match dscp af21 af22 af23 cs2
class-map match-any Bulk
 match dscp af11 af12 af13 cs1 default
!
policy-map cbwfq-parent
 class Mission-critical
 service-policy cbwfq-child
 bandwidth percent 30
 class Bulk
 service-policy cbwfq-child
 bandwidth percent 40
!
policy-map cbwfq-child
 class Internal
 bandwidth remaining percent 80
 !
 class External
 bandwidth remaining percent 20
!
interface GigabitEthernet0/0/0/0
 service-policy input cbwfq-parent
commit

```

## Task 2B: Configure CBWFQ on IOS XE Software

```

ip access-list extended External-podY-ACL
 permit icmp 192.168.10Y.0 0.0.0.255 any
!
ip access-list extended Internal-podY-ACL
 permit icmp 10.Y.10.0 0.0.0.255 any
!
class-map match-any External
 match access-group name External-podY-ACL
!
class-map match-any Internal
 match access-group name Internal-podY-ACL

```

```

!
class-map match-all ef-class
 match dscp ef
!
policy-map cbwfq-policy
 class Internal
 bandwidth percent 95
 class External
 bandwidth percent 2
 class ef-class
 bandwidth percent 2
policy-map shape-all
 class class-default
 shape average 100000
 service-policy cbwfq-policy
!
interface GigabitEthernet0/0/3
 service-policy output shape-all

```

### Task 3A: Configure LLQ on IOS XR Software

```

policy-map llq-policy
 class Internal
 priority level 1
 police rate percent 20
 queue-limit 20 ms
 !
 class External
 bandwidth percent 30
 queue-limit 50 ms
 !
 class ef-class
 priority level 2
 police rate percent 10
 !
interface GigabitEthernet0/0/0/2
 no service-policy output
 service-policy output llq-policy
!
interface GigabitEthernet0/0/0/0
 no service-policy output
 service-policy output llq-policy
commit

class-map match-any Top-priority-prec
 match precedence 5
!
commit

```

```

!
interface GigabitEthernet0/0/0/0
 no service-policy output llq-policy
!
no policy-map shape-all
!
commit

```

### Task 3B: Configure LLQ on IOS XE Software

```

policy-map llq-policy
 class Internal
 priority percent 20
 !
 class External
 bandwidth percent 30
 !
 class ef-class
 bandwidth percent 20
 !
!
interface GigabitEthernet0/0/3
 service-policy output llq-policy
!

interface GigabitEthernet0/0/3
 no service-policy output llq-policy
!
no policy-map shape-all

```

## Lab 5-2 Answer Key: Implement Congestion Avoidance

When you complete this lab activity, device configuration and device outputs will be similar to the results shown here, with differences that are specific to your pod.

### Task 2A: Enable WRED on Cisco IOS XR Software

```

ipv4 access-list External-podX-ACL
 10 permit icmp 192.168.10X.0/24 any
!
ipv4 access-list Internal-podX-ACL
 10 permit icmp 10.X.10.0/24 any
!
class-map match-any External
 match access-group ipv4 External-podX-ACL
!
class-map match-any Internal
 match access-group ipv4 Internal-podX-ACL
!
class-map match-any ef-class
 match dscp ef

```

```

!
policy-map cbwfq-equal
 class Internal
 random-detect default
 bandwidth percent 40
!
 class External
 bandwidth percent 40
 random-detect default
!
 class ef-class
 bandwidth percent 40
 random-detect default
!
policy-map shape-all
 class class-default
 service-policy cbwfq-equal
 shape average 200 kbps
!
interface GigabitEthernet0/0/0/2
 service-policy output shape-all
!
commit

```

## Task 2B: Enable WRED on Cisco IOS XE Software

```

ip access-list extended External-podY-ACL
 permit icmp 192.168.10Y.0 0.0.0.255 any
!
ip access-list extended Internal-podY-ACL
 permit icmp 10.Y.10.0 0.0.0.255 any
!
class-map match-any External
 match access-group name External-podY-ACL
!
class-map match-any Internal
 match access-group name Internal-podY-ACL
!
class-map match-all ef-class
 match dscp ef
!
policy-map cbwfq-equal
 class Internal
 bandwidth percent 30
 random-detect dscp-based
 class External
 bandwidth percent 30
 random-detect dscp-based
 class ef-class
 bandwidth percent 30
 random-detect dscp-based

```

```

!
policy-map shape-all
 class class-default
 shape average 100000
 service-policy cbwfq-equal
!
interface GigabitEthernet0/0/3
 service-policy output shape-all

```

### Task 3A: Customize WRED on Cisco IOS XR Software

```

no policy-map cbwfq-equal
policy-map cbwfq-equal
 class Internal
 random-detect dscp ef 100 kbytes 200 kbytes
 random-detect dscp default 10 kbytes 100 kbytes
 bandwidth percent 40
 !
 class External
 bandwidth percent 40
 random-detect dscp default 2 kbytes 12 kbytes
 !
 class ef-class
 bandwidth percent 40
 random-detect dscp ef 200 kbytes 300 kbytes
 !
interface GigabitEthernet0/0/0/0
 service-policy input shape-all
commit

interface GigabitEthernet0/0/0/0
 no service-policy input shape-all
interface GigabitEthernet0/0/0/2
 no service-policy output shape-all
commit

```

### Task 3B: Customize WRED on Cisco IOS XE Software

```

policy-map cbwfq-equal
 class Internal
 bandwidth percent 30
 random-detect dscp-based
 random-detect dscp 0 20 100
 random-detect dscp 46 30 100 20
 class External
 bandwidth percent 30
 random-detect dscp-based
 random-detect dscp 0 1 2 1
 !
interface GigabitEthernet0/0/3
 no service-policy output shape-all

```

!

## Lab 6-1 Answer Key: Implement Traffic Policing

When you complete this lab activity, device configuration and device outputs will be similar to the results shown here, with differences that are specific to your pod.

### Task 1A: Configure Single-Level Policing on Cisco IOS XR Software

PE1 (Cisco IOS XR):

```
ipv4 access-list Customer-podX-ACL
 10 permit ipv4 192.168.10X.0/24 any
!
class-map match-any Customer
 match access-group ipv4 Customer-podX-ACL
!
policy-map ingress
 class Customer
 police rate 1 mbps burst 10 ms
 conform-action transmit
 exceed-action drop
!
interface GigabitEthernet0/0/0/0
 service-policy input ingress
commit
```

#### Dual token bucket policing:

```
policy-map ingress
 class Customer
 police rate 1 mbps burst 10 ms peak-burst 20 ms
 conform-action transmit
 no exceed-action drop
 exceed-action set dscp cs1
 violate-action drop
commit
```

#### Dual-rate policing:

```
policy-map ingress
 class Customer
 police rate 1 mbps burst 10 ms peak-rate 2 mbps peak-burst 20 ms
 conform-action transmit
 exceed-action set dscp cs1
 violate-action drop
commit
!
interface GigabitEthernet0/0/0/0
 no service-policy input ingress
commit
```

## Task 1B: Configure Single-Level Policing on Cisco IOS XE Software

PE2 (Cisco IOS):

### Single token bucket policing:

```
ip access-list extended Customer-podY-ACL
 permit ip 192.168.10Y.0 0.0.0.255 any
!
class-map match-any Customer
 match access-group name Customer-podY-ACL
!
policy-map ingress
 class Customer
 police rate 1 m burst 1300
 conform-action transmit
 exceed-action drop
!
interface GigabitEthernet0/0/0
 service-policy input ingress
!
```

### Dual token bucket policing:

```
policy-map ingress
 class Customer
 police rate 1 m burst 1300 peak-burst 2600
 no exceed-action drop
 conform-action transmit
 exceed-action set-dscp-transmit cs1
 violate-action drop
!
```

### Dual-rate policing:

```
policy-map ingress
 class Customer
 police rate 1 m burst 1300 peak-rate 2 m
 conform-action transmit
 exceed-action set-dscp-transmit cs1
 violate-action drop
!
interface GigabitEthernet0/0/0
 no service-policy input ingress
!
```

## Task 2: Configure Individual and Aggregate Policers on Cisco IOS Access Switches

SW1 (Cisco IOS):

### Individual policer:

```
ip access-list extended Customer-podX-ACL
 permit ip 192.168.10X.0 0.0.0.255 any
!
class-map match-any Customer
```

```

 match access-group name Customer-podX-ACL
!
policy-map ingress
 class Customer
 police cir 1 m bc 8000 pir 2 m
 conform-action transmit
 exceed-action set-dscp-transmit cs1
 violate-action drop
!
interface FastEthernet 0/1
 service-policy input ingress

```

#### Aggregate policer:

```

policer aggregate agg-customer-policer cir 1000000 bc 8000 pir 2000000
conform-action transmit exceed-action set-dscp-transmit cs1 violate-
action drop
policy-map ingress
 class Customer
 no police cir 1000000 bc 8000 pir 2000000
 police aggregate agg-customer-policer
!
interface FastEthernet 0/1
 no service-policy input ingress
!

```

#### SW2 (Cisco IOS):

##### Individual policer:

```

ip access-list extended Customer-podX-ACL
 permit ip 192.168.10Y.0 0.0.0.255 any
!
class-map match-any Customer
 match access-group name Customer-podY-ACL
!
policy-map ingress
 class Customer
 police cir 1 m bc 8000 pir 2 m
 conform-action transmit
 exceed-action set-dscp-transmit cs1
 violate-action drop
!
interface FastEthernet 0/1
 service-policy input ingress
!

```

##### Aggregate policer:

```

policer aggregate agg-customer-policer cir 1000000 bc 8000 pir 2000000
conform-action transmit exceed-action set-dscp-transmit cs1 violate-
action drop
policy-map ingress

```

```

class Customer
 no police cir 1000000 bc 8000 pir 2000000
 police aggregate agg-customer-policer
!
interface FastEthernet 0/1
 no service-policy input ingress
!

```

### Task 3: Configure Hierarchical Policing

PE1 (Cisco IOS XR):

#### Cisco IOS XR:

```

class-map match-any Customers
 match protocol ipv4
!
policy-map Customer-policer
 class Customer
 police rate percent 50 peak-rate percent 60
 conform-action transmit
 exceed-action set dscp cs1
 violate-action drop
!
policy-map ingress-all
 class Customers
 police rate 2 mbps
 conform-action transmit
 exceed-action drop
 service-policy Customer-policer
!
interface GigabitEthernet0/0/0/0
 service-policy input ingress-all
!
commit

!
interface GigabitEthernet0/0/0/0
 no service-policy input ingress-all
!
commit

```

PE2 (Cisco IOS):

#### Cisco IOS XE:

```

class-map match-any Customers
 match protocol ip
!
policy-map Customer-policer
 class Customer
 police rate percent 50 peak-rate percent 60

```

```

 conform-action transmit
 exceed-action set-dscp-transmit cs1
 violate-action drop
 !
policy-map ingress-all
 class Customers
 police rate 2 m
 conform-action transmit
 exceed-action drop
 service-policy Customer-policer
 !
interface GigabitEthernet0/0/0
 service-policy input ingress-all

interface GigabitEthernet0/0/0
 no service-policy input ingress-all

```

#### Task 4: Configure LPTS Hardware Policing

```

PE1 (Cisco IOS XR):
lpts pifib hardware police
 flow fragment rate 100
 flow icmp local rate 500
 flow icmp default rate 1000
 flow http known rate 150
 flow http default rate 300
!
lpts pifib hardware police location 0/0/CPU0
!
commit

```

## Lab 6-2 Answer Key: Implement Traffic Shaping

When you complete this lab activity, device configuration and device outputs will be similar to the results shown here, with differences that are specific to your pod.

#### Task 1A: Configure Single-Level Traffic Shaping on IOS XR Software

```

PE1 (Cisco IOS XR):
ipv4 access-list Customer-podX-ACL
 10 permit ipv4 192.168.101.0/24 any
!
class-map match-any Customer
 match access-group ipv4 Customer-podX-ACL
!
policy-map egress
 class Customer
 shape average 500 kbps
 !
interface GigabitEthernet0/0/0/2
 service-policy output egress

```

```
commit
```

Excess burst configuration:

```
policy-map egress
 class Customer
 shape average 500 kbps 1000 ms
commit
```

```
interface GigabitEthernet0/0/0/2
 no service-policy output egress
commit
```

## Task 1B: Configure Single-Level Traffic Shaping on IOS XE Software

PE2 (Cisco IOS):

```
ip access-list extended Customer-podY-ACL
 permit ip 192.168.10Y.0 0.0.0.255 any
!
class-map match-any Customer
 match access-group name Customer-podY-ACL
!
policy-map egress
 class Customer
 shape average 500 k
!
interface GigabitEthernet0/0/3
 service-policy output egress
interface GigabitEthernet0/0/3
 no service-policy output egress
!
```

## Task 2: Configure Hierarchical Traffic Shaping with CBWFQ on the Child Level

PE2 (Cisco IOS):

```
ip access-list extended External-podY-ACL
 permit icmp 192.168.10Y.0 0.0.0.255 any
!
ip access-list extended Internal-podY-ACL
 permit icmp 10.Y.10.0 0.0.0.255 any
!
class-map match-any External
 match access-group name External-podY-ACL
!
class-map match-any Internal
 match access-group name Internal-podY-ACL
!
policy-map child-cbwfq
 class Internal
 bandwidth percent 80
```

```

class External
 bandwidth percent 10
!
policy-map shape-all
 class class-default
 shape average 500 k
 service-policy child-cbwfq
!
interface GigabitEthernet0/0/0
 service-policy output shape-all

```

### Task 3: Configure Hierarchical Traffic Shaping with Shaping on the Child Level

PE1 (Cisco IOS XR):

```

ipv4 access-list External-podX-ACL
 10 permit icmp 192.168.101.0/24 any
!
ipv4 access-list Internal-podX-ACL
 10 permit icmp 10.1.10.0/24 any
!
class-map match-any External
 match access-group ipv4 External-podX-ACL
!
class-map match-any Internal
 match access-group ipv4 Internal-podX-ACL
!
policy-map child-shaper
 class Internal
 shape average percent 80
 class External
 shape average percent 10
!
policy-map shape-all
 class class-default
 shape average 700 kbps
 service-policy child-shaper
!
interface GigabitEthernet0/0/0/2
 service-policy output shape-all
!
commit

```

PE2 (Cisco IOS XE):

```

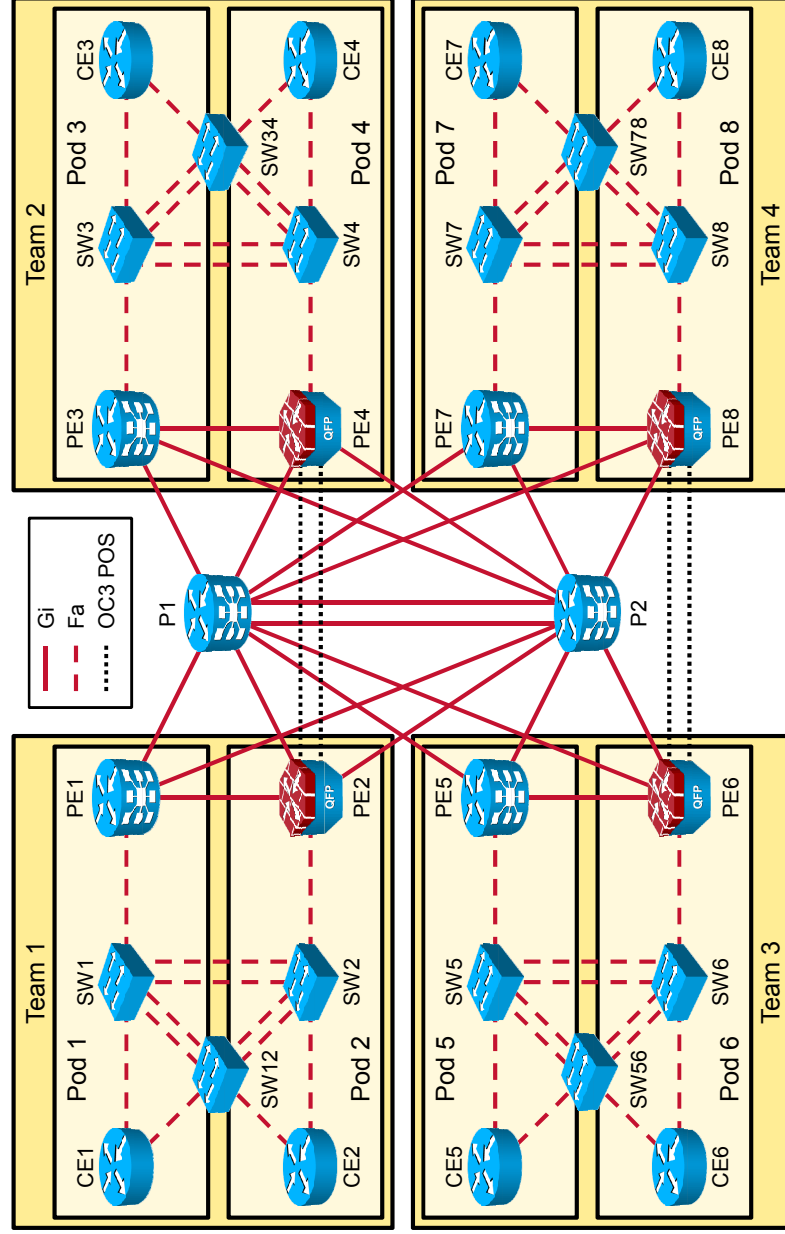
ip access-list extended External-podY-ACL
 permit icmp 192.168.10Y.0 0.0.0.255 any
!
ip access-list extended Internal-podY-ACL
 permit icmp 10.Y.10.0 0.0.0.255 any
!

```

```
class-map match-any External
 match access-group name External-podY-ACL
!
class-map match-any Internal
 match access-group name Internal-podY-ACL
!
policy-map child-shaper
 class Internal
 shape average percent 80
 class External
 shape average percent 10
!
policy-map shape-all
 class class-default
 shape average 700 k
 service-policy child-shaper
!
interface GigabitEthernet0/0/3
 service-policy output shape-all
```

# Appendix A: Lab Topology

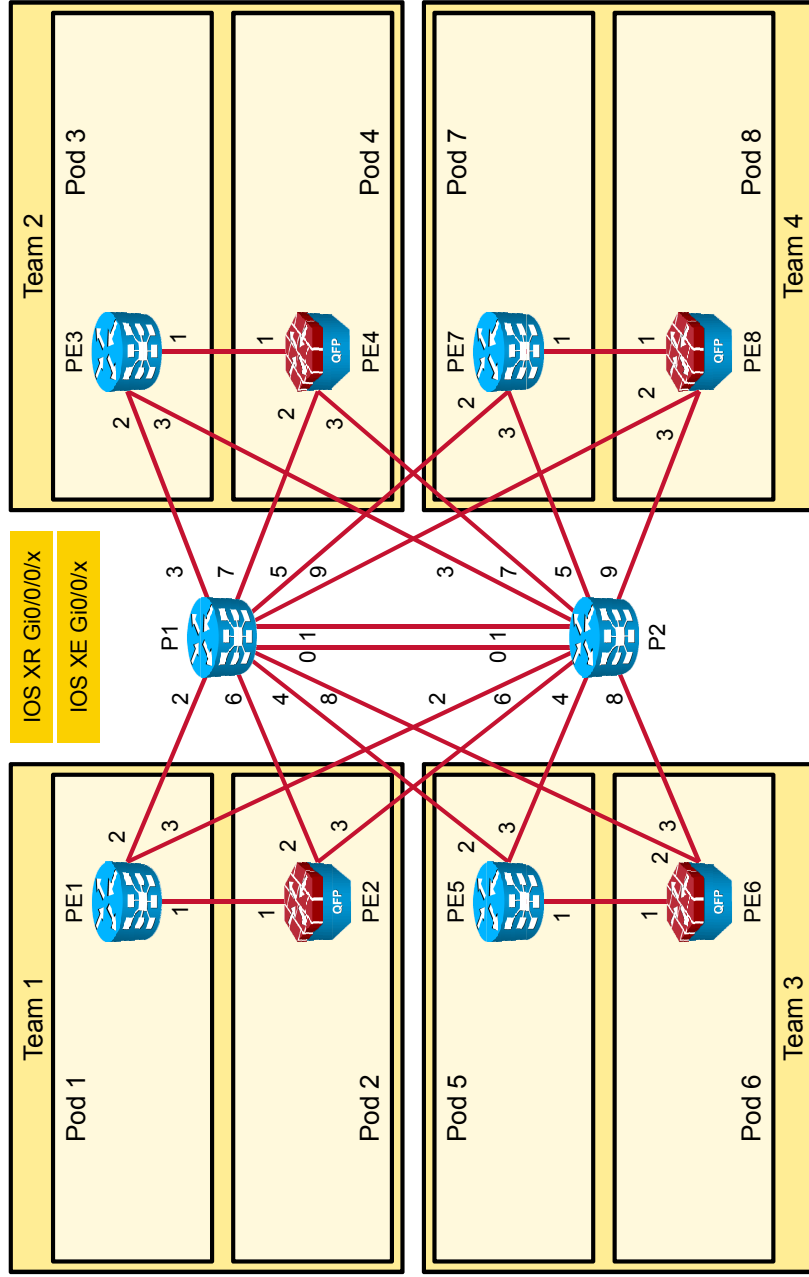
## Lab Topology



© 2012 Cisco and/or its affiliates. All rights reserved.

SPCORE v1.01-4

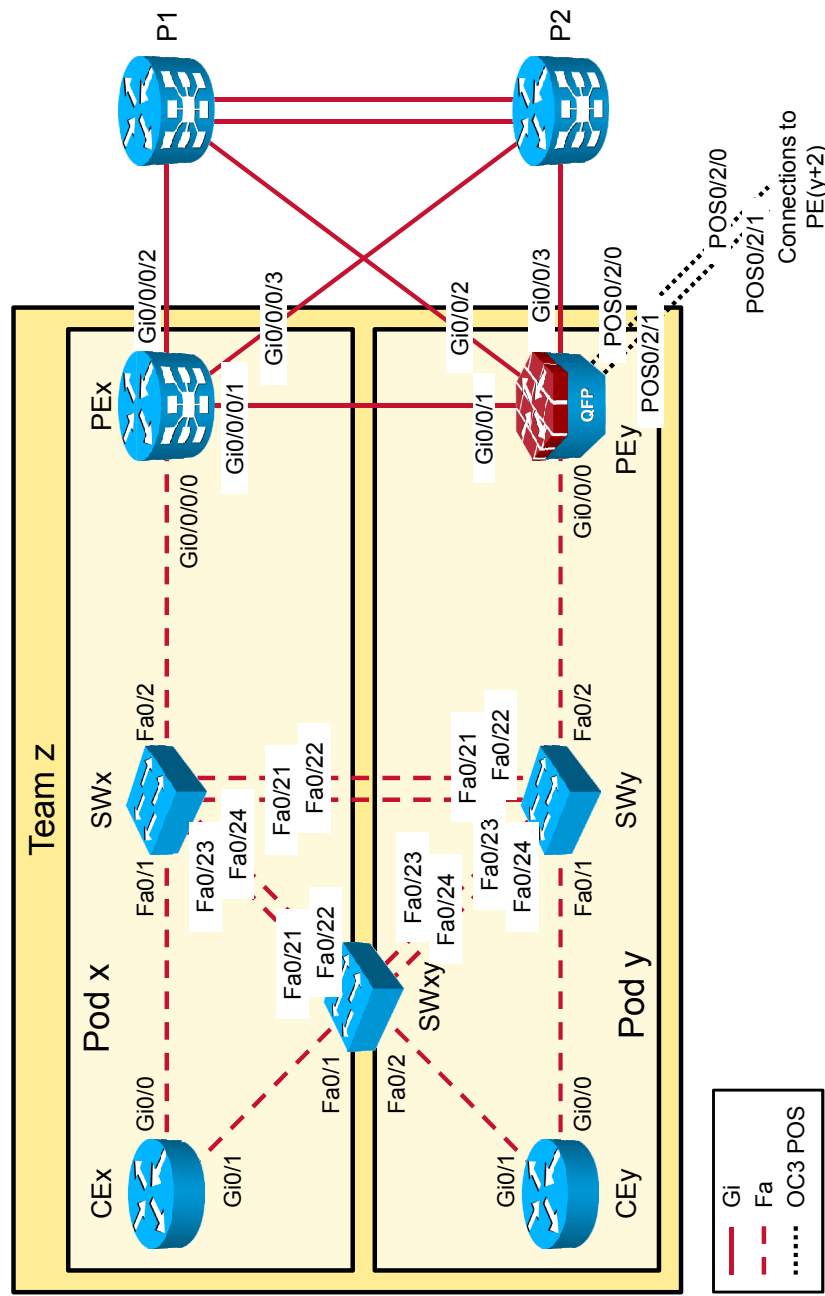
# Lab Interface Identification—Core View



© 2012 Cisco and/or its affiliates. All rights reserved.

SPCORE v1.01-5

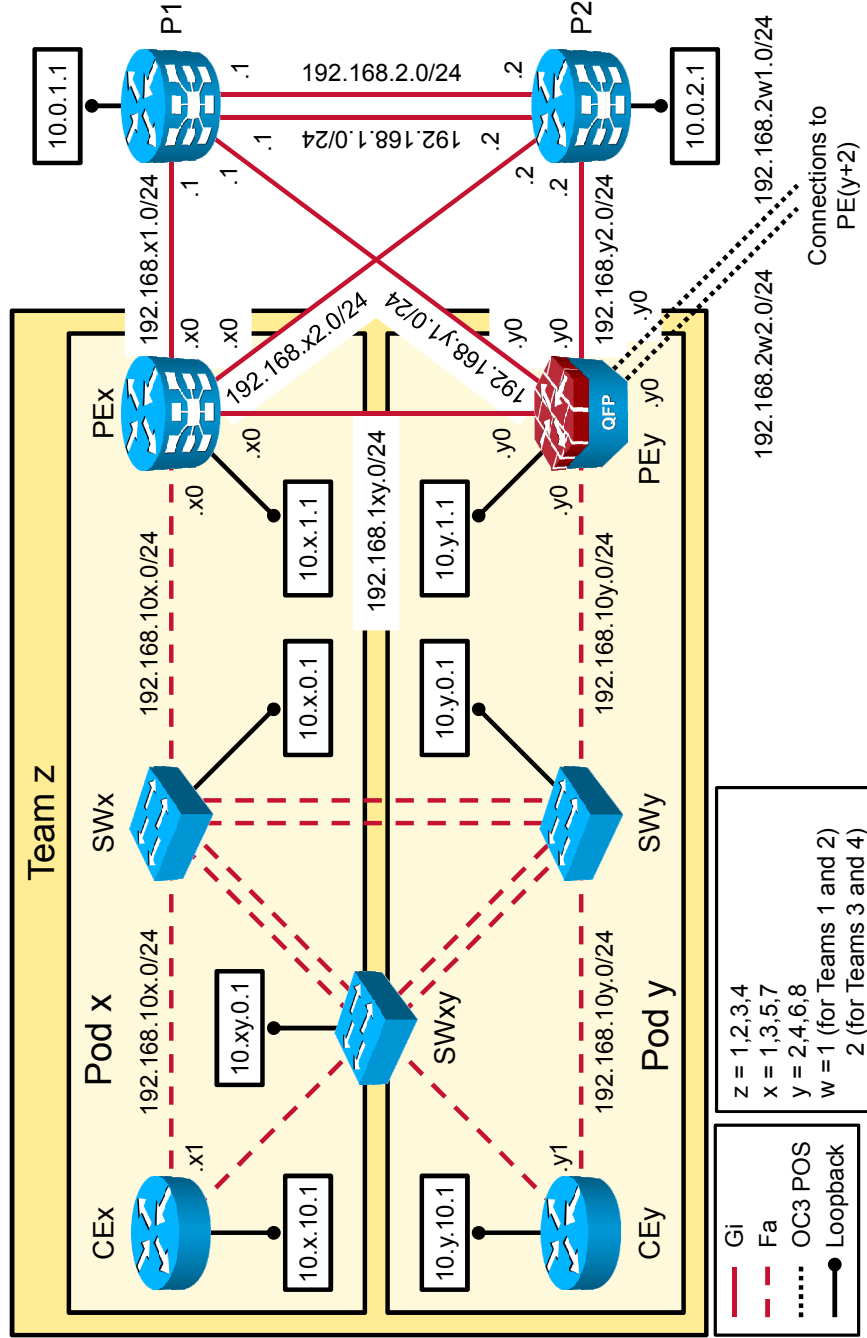
## Lab Interface Identification—Team View



© 2012 Cisco and/or its affiliates. All rights reserved.

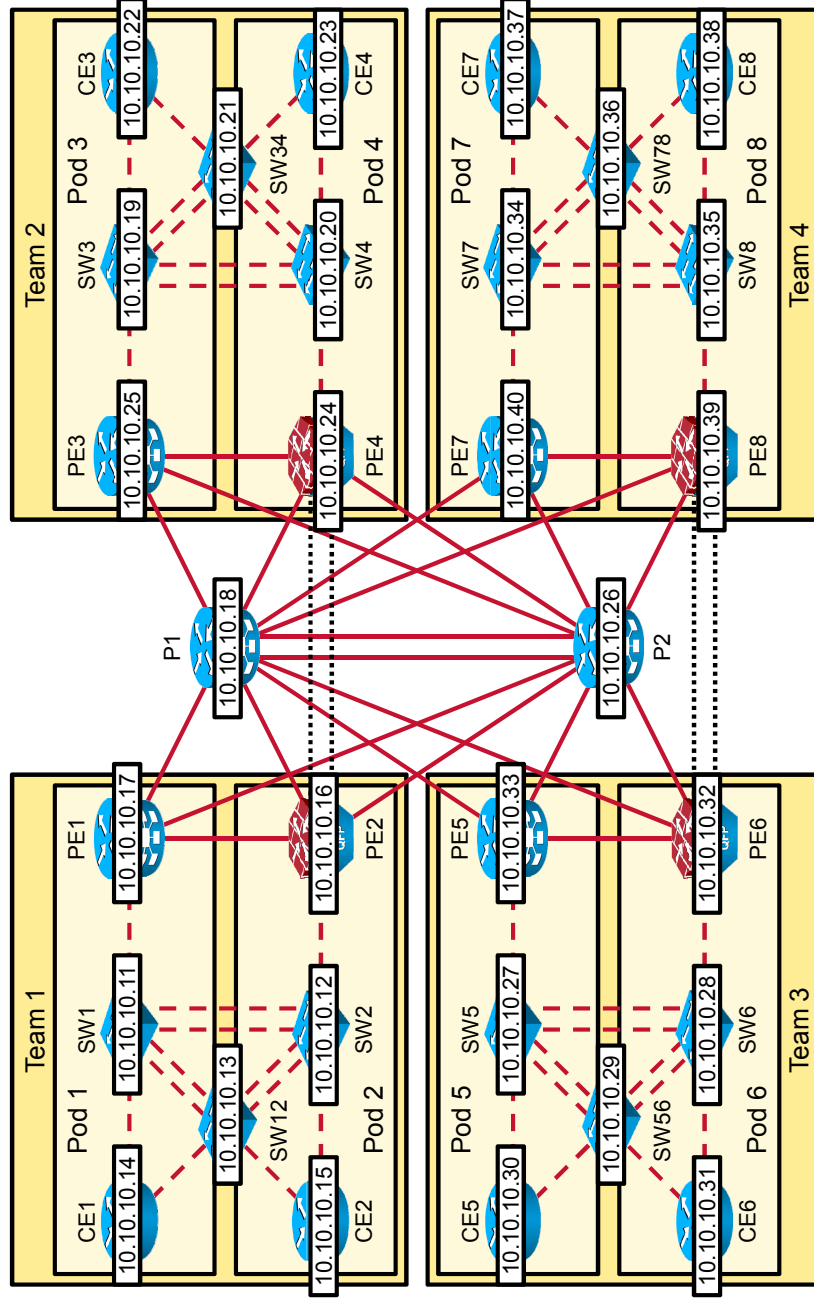
SPCORE v1.01-6

# Lab IP Addressing—Team View



© 2012 Cisco and/or its affiliates. All rights reserved. SPCORE v1.01-7

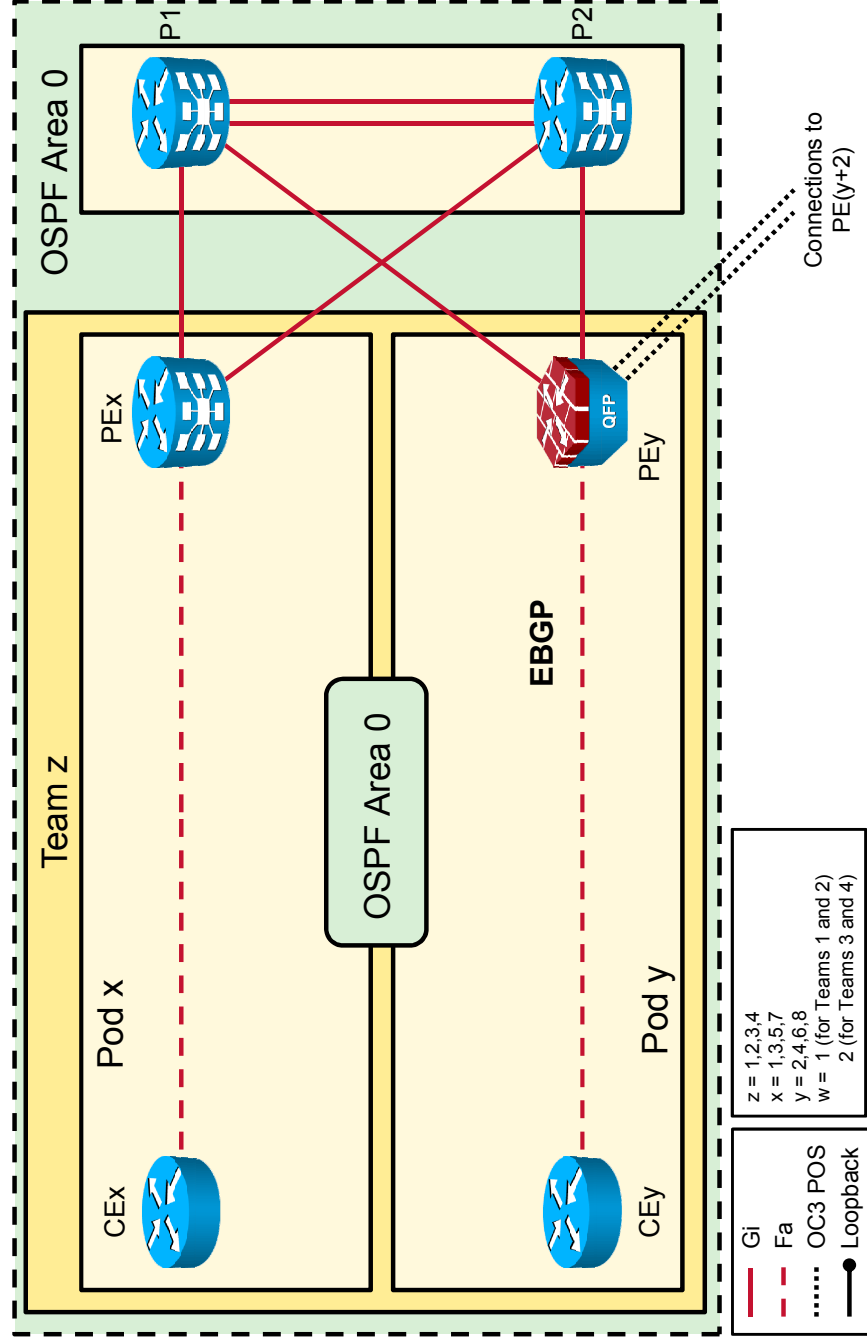
# Management IP Addresses



© 2012 Cisco and/or its affiliates. All rights reserved.

SF-CORE v1.01-3

# Existing Routing—Team View



SPCORE v1.01-9

© 2012 Cisco and/or its affiliates. All rights reserved.