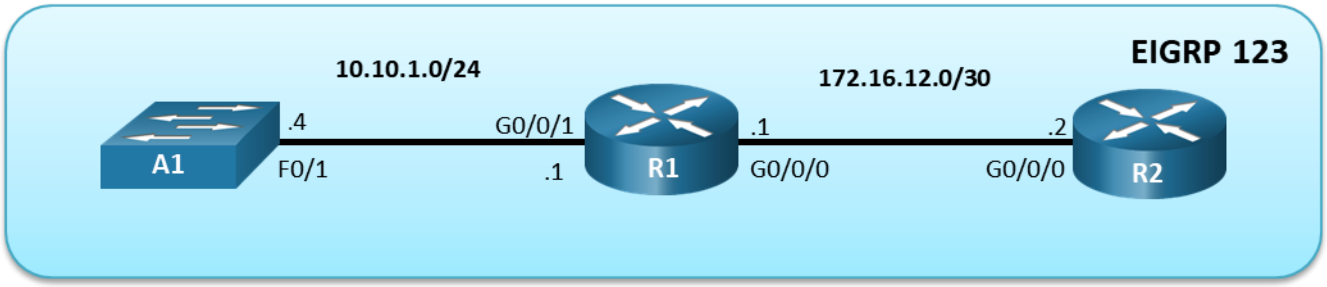
# Lab - Implement CoPP (Instructor Version)

**Instructor Note**: Red font color or gray highlights indicate text that appears in the instructor copy only.

## Answers: 26.1.6 Lab - Implement CoPP

## Topology



## Addressing Table

| Device | Interface | IP Address | Subnet Mask |
|---|---|---|---|
| R1 | G0/0/0 | 172.16.12.1 | 255.255.255.252 |
| | G0/0/1 | 10.10.1.1 | 255.255.255.0 |
| R2 | G0/0/0 | 172.16.12.2 | 255.255.255.252 |
| A1 | VLAN 1 | 10.10.1.4 | 255.255.255.0 |

## Objectives

**Part 1: Build the Network and Configure Basic Device Settings**

**Part 2: Verify Initial Connectivity**

**Part 3: Implement a CoPP Policy on R1**

**Part 4: Verify the CoPP Policy on R1**

**Part 5: (Challenge) Further Classify Default Traffic**

## Background / Scenario

Control Plane Policing (CoPP) is a protection feature for the router's control plane CPU. CoPP can granularly permit, drop, or rate-limit traffic to or from the CPU using a Modular QoS CLI (MQC) policy. The CoPP policy is applied to a dedicated control-plane interface which protects the CPU from unexpected extreme rates of traffic that could impact the stability of the router.

CoPP handles all process-switched traffic, such as packets logged by an ACL or IP packets with header (TTL) options. Other types of traffic directed to the control plane include routing updates, (OSPF, EIGRP and BGP) as well as management traffic, including Telnet, SNMP, SSH, NTP, and HTTP etc.

The focus of this lab is using the Cisco IOS Modular QoS CLI (MQC) to implement CoPP.

**Note**: This lab is an exercise in configuring CoPP policies and does not necessarily reflect network best practices.

**Note**: The routers used with CCNP hands-on labs are Cisco 4221 with Cisco IOS XE Release 16.9.4 (universalk9 image). The switch used in the lab is a Cisco Catalyst 2960 with Cisco IOS Release 15.2(2) (lanbasek9 image). Other routers, switches, and Cisco IOS versions can be used. Depending on the model and Cisco IOS version, the commands available and the output produced might vary from what is shown in the labs. Refer to the Router Interface Summary Table at the end of the lab for the correct interface identifiers.

**Note**: Make sure that the routers and switches have been erased and have no startup configurations. If you are unsure, contact your instructor.

**Instructor Note**: Refer to the Instructor Lab Manual for the procedures to initialize and reload devices.

## Required Resources

- 2 Routers (Cisco 4221 with Cisco IOS XE Release 16.9.4 universal image or comparable)
- 1 Switch (Cisco 2960 with Cisco IOS Release 15.2(2) lanbasek9 image or comparable)
- 1 PC (Choice of operating system with terminal emulation program and a packet capture utility installed)
- Console cables to configure the Cisco IOS devices via the console ports
- Ethernet cables as shown in the topology

## Part 1: Build the Network and Configure Basic Device Settings

In Part 1, you will set up the network topology and configure basic settings and interface addressing on devices.

### Step 1: Cable the network as shown in the topology.

Attach the devices as shown in the topology diagram, and cable as necessary.

### Step 2: Configure basic settings for each device.

a. Console into each device, enter global configuration mode, and apply the basic settings. The startup configurations for each device are provided below.

**Router R1**

```
hostname R1
no ip domain lookup
ip domain name CCNPv8.CoPP.Lab
username admin privilege 15 algorithm-type scrypt secret cisco123
banner motd # R1, Control Plane Policing  #
line con 0
 exec-timeout 0 0
 logging synchronous
 exit
interface g0/0/1
 ip address 10.10.1.1 255.255.255.0
 no shutdown
 exit
interface g0/0/0
 ip address 172.16.12.1 255.255.255.252
```

```
  no shutdown
  exit
 router eigrp 123
   eigrp router-id 0.0.0.1
   network 172.16.12.0 0.0.0.3
   network 10.10.1.0 0.0.0.255
   exit
 line vty 0 4
  login local
  transport input telnet ssh
 exit
 crypto key generate rsa modulus 1024
  end
```

**Router R2**

```
hostname R2
no ip domain lookup
ip domain name CCNPv8.CoPP.Lab
username admin privilege 15 algorithm-type scrypt secret cisco123
banner motd # R2, Control Plane Policing  #
line con 0
 exec-timeout 0 0
 logging synchronous
 exit
interface g0/0/0
 ip address 172.16.12.2 255.255.255.252
 no shutdown
 exit
router eigrp 123
  eigrp router-id 0.0.0.2
  network 172.16.12.0 0.0.0.3
  exit
line vty 0 4
 login local
 transport input telnet ssh
exit
crypto key generate rsa modulus 1024
 end
```

**Switch A1**

```
hostname A1
no ip domain lookup
username admin privilege 15 algorithm-type scrypt secret cisco123
ip domain name CCNPv8.COPP.LAB
banner motd # A1, Control Plane Policing  #
spanning-tree mode rapid-pvst
```

```
 line con 0
 exec-timeout 0 0
 logging synchronous
 exit
line vty 0 15
 login local
 transport input telnet ssh
interface range f0/1-24, g0/1-2
 shutdown
 exit
interface range f0/1
switchport mode access
 no shutdown
 exit
interface vlan 1
 ip address 10.10.1.4 255.255.255.0
 no shut
 exit
 ip default-gateway 10.10.1.1
 crypto key generate rsa modulus 1024
 end
```

b. Set the clock on each device to UTC time.

c. Save the running configuration to startup-config.

d. Verify ICMP connectivity between the devices.

## Part 2: Verify Initial Connectivity

It is always advisable to test network connectivity and services before applying any policies. This ensures that the network is fully functional, and that the loss of connectivity or functionality is due to the applied policy and not a pre-existing network issue.

When testing TCP connectivity, some services will prompt for a username/password. The pre-configured username is **admin** and password is **cisco123**.

### Step 1: Test Telnet connectivity.

From A1, test Telnet connectivity to R1 and R2. When prompted for a username / password use **admin** and **cisco123**. You should be successful. Troubleshoot as needed.

### Step 2: Test SSH connectivity.

From A1, test SSH connectivity to R1 and R2. When prompted for a username / password use **admin** and **cisco123**. You should be successful. Troubleshoot as needed.

## Part 3: Implement a CoPP Policy on R1

In this part, you will configure ACLs to identify traffic flows. You will then use these ACLs in class maps to classify the traffic for CoPP. Next, you will bind the class maps to a CoPP policy map. This policy map will apply the policing and actions to take for each class map. Finally, you will apply the policy map to the control plane.

### Step 1: Configure named ACLs to identify traffic flows.

Understanding what is needed for traffic classification can be achieved from network protocol analysis. Cisco NetFlow and Embedded Packet Capture (EPC) can be used to classify network traffic.

After the traffic has been identified, ACLs can be built for matching the identified traffic. The definition of these ACLs is one of the most critical steps in the CoPP process. MQC uses these ACLs to define the traffic classes. Appropriate granularity in the classification of these protocols within these ACLs allows for better protection of the control plane CPU.

a. Configure an extended ACL using the name **TELNET** to identify Telnet traffic.

```
R1(config)# ip access-list extended TELNET
R1(config-ext-nacl)# permit tcp any any eq 23
R1(config-ext-nacl)# exit
```

b. Configure an extended ACL using the name **EIGRP** to identify all EIGRP traffic.

```
R1(config)# ip access-list extended EIGRP
R1(config-ext-nacl)# permit eigrp any any
R1(config-ext-nacl)# exit
```

c. Configure an extended ACL using the name **SSH** to identify all SSH traffic.

```
R1(config)# ip access-list extended SSH
R1(config-ext-nacl)# permit tcp any any eq 22
R1(config-ext-nacl)# exit
```

d. Configure an extended ACL using the name **ICMP** to identify ICMP traffic.

```
R1(config)# ip access-list extended ICMP
R1(config-ext-nacl)# permit icmp any any
R1(config-ext-nacl)# exit
```

**Note:** Packets not matching any permit statements will be sent to the control plane CPU on R1.

### Step 2: Configure class maps for CoPP on R1.

Class maps are used to classify traffic for CoPP. The ACLs defined previously specify which IP packets belong to which classes. The mapping of ACLs to class maps completes the traffic classification process.

Class maps support multiple match criteria; however, in this configuration a single-match criteria will satisfy basic functionality.

a. Configure a class map named **CM–TELNET** to match IP packets that should be dropped and never reach the control plane CPU.

```
R1(config)# class-map match-all CM-TELNET
R1(config-cmap)# match access-group name TELNET
R1(config-cmap)# exit
```

b. Configure a class map named **CM–EIGRP** to match EIGRP packets. This will allow you to apply a policy to these routing packets. The same could be done for OSPF or BGP packets if those routing protocols were active.

```
R1(config)# class-map match-all CM-EIGRP
R1(config-cmap)# match access-group name EIGRP
R1(config-cmap)# exit
```

c. Configure a class map named **CM-SSH** to match IP packets in the ACL named SSH. These IP packets are SSH packets destined to the management plane.

```
R1(config)# class-map match-all CM-SSH
R1(config-cmap)# match access-group name SSH
R1(config-cmap)# exit
```

d. Configure a class map named **CM–ICMP** to match IP packets in the ACL named **ICMP**. These IP packets are ICMP packets destined to the router.

```
R1(config)# class-map match-all CM-ICMP
R1(config-cmap)# match access-group name ICMP
R1(config-cmap)# exit
```

## Step 3: Configure a policy map for CoPP on R1.

The policy map defines the “baseline” service policy for each traffic classification. For each traffic class previously configured, the policy map applies the control plane policing rates and actions.

Finding the optimal police rate without impacting network stability requires understanding traffic flows over time. To guarantee that CoPP does not introduce stability issues, the violate action should be set to transmit for all vital classes. When the CoPP policy is confirmed to be operationally effective, that would be the time to convert the appropriate “exceed-action” transmit actions to drop actions.

a. Configure a policy map named **PM–COPP** on R1.

```
R1(config)# policy-map PM-COPP
```

b. Associate the traffic class maps to the policy map and configure the following policing and action policies:

- For Telnet packets, configure policing at 8 kbps. Then set the conform action to drop and the exceed action to drop. This will effectively drop all Telnet packets that match the class map.
- For EIGRP packets, configure policing at 20 packets per second (pps). Then set the conform action to transmit and the exceed action to drop.
- For SSH packets, configure the policing at 50 kbps, set the conform action to transmit and the exceed action to transmit. This will effectively transmit all SSH packets that match the class map.
- For ICMP packet, configure policing at 10pps. Then set the conform action to transmit and the exceed action to drop.

```
R1(config-pmap)# class CM-TELNET
R1(config-pmap-c)# police 8000 conform-action drop exceed-action drop
R1(config-pmap-c-police)# exit
R1(config-pmap-c)# class CM-EIGRP
R1(config-pmap-c)# police rate 10 pps conform-action transmit exceed-action transmit
R1(config-pmap-c-police)# exit
R1(config-pmap-c)# class CM-SSH
R1(config-pmap-c)# police 50000 conform-action transmit exceed-action transmit
R1(config-pmap-c-police)# exit
R1(config-pmap-c)# class CM-ICMP
R1(config-pmap-c)# police rate 10 pps conform-action transmit exceed-action drop
R1(config-pmap-c-police)# exit
```

c. Associate the default class map with the traffic policy. Set the class to **class-default** and configure policing at 12 kbps. Then set the conform action to transmit and the exceed action to transmit.

```
R1(config-pmap-c)# class class-default
R1(config-pmap-c)# police 12000 conform-action transmit exceed-action transmit
```

```
R1(config-pmap-c-police)# end
```

**Note**: The class **class-default** is automatically placed at the end of the policy map. By the nature of CoPP-matching mechanisms, certain traffic types will always end up falling into the default class. This includes Layer 2 keepalives and non-IP traffic. Because these traffic types are required to maintain the network control plane, the **class-default** should never be policed with both conform and exceed actions set to drop.

### Step 4: Apply the CoPP policy to the control plane virtual interface on R1.

The policy map is applied to the control plane virtual interface in the inbound direction using the **service-policy** command. Only traffic destined for the router's control plane will be affected by the CoPP policy.

a. Enter control plane configuration mode to apply a CoPP policy.

```
R1# conf t
R1(config)# control-plane
```

b. Next, attach the policy map to the control plane interface using the **service-policy input** command and specify the policy map named PM–COPP on the control plane virtual interface.

```
R1(config-cp)# service-policy input PM-COPP
R1(config-cp)# end
```

## Part 4: Verify the CoPP Policy on R1.

Within a few moments of the CoPP policy being applied, dynamic information can be seen about the actual policy including rate information and the number of bytes and packets that conformed to or exceeded the configured policies.

### Step 1: Issue the show access-list command.

The **show access-list** command identifies specific traffic for each ACL.

```
R1# show access-lists
Extended IP access list EIGRP
    10 permit eigrp any any
Extended IP access list ICMP
    10 permit icmp any any
Extended IP access list SSH
    10 permit tcp any any eq 22
Extended IP access list TELNET
    10 permit tcp any any eq telnet
```

### Step 2: Issue the show class-map command.

The only class maps configured are those used for CoPP. Notice the **match-any** statement under the **class-default** class map. The **match** statement was automatically added when the class map was included in the policy map configuration.

```
R1# show class-map
  Class Map match-all CM-TELNET (id 1)
   Match access-group name TELNET

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

 Class Map match-all CM-ICMP (id 5)
```

```
      Match access-group name ICMP

   Class Map match-all CM-EIGRP (id 2)
      Match access-group name EIGRP

    Class Map match-all CM-SSH (id 3)
      Match access-group name SSH
```

### Step 3: Issue the show policy-map command.

Notice the policy map shows the policing rates that were configured for each class map, including the **class–default** class map.

```
R1# show policy-map
  Policy Map PM-COPP
    Class CM-TELNET
     police cir 8000 bc 1500
       conform-action drop
       exceed-action drop
    Class CM-EIGRP
     police rate 10 pps
       conform-action transmit
       exceed-action transmit
    Class CM-SSH
     police cir 50000 bc 1562
       conform-action transmit
       exceed-action transmit
        Class CM-ICMP
     police rate 10 pps
       conform-action transmit
       exceed-action drop
        Class class-default
     police cir 12000 bc 1500
       conform-action transmit
       exceed-action transmit
```

### Step 4: Issue the show policy-map control-plane command.

This is the most useful command for verifying CoPP functionality. Notice that the EIGRP packets conform to the policing rule. Because no packets exceeded the defined rates for the EIGRP class map, all packets are transmitted to the control plane for processing. Note that your packet count will vary from the example below.

```
R1# show policy-map control-plane
Control Plane

  Service-policy input: PM-COPP

    Class-map: CM-TELNET (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: access-group name TELNET
      police:
          cir 8000 bps, bc 1500 bytes
        conformed 0 packets, 0 bytes; actions:
          drop
        exceeded 0 packets, 0 bytes; actions:
          drop
        conformed 0000 bps, exceeded 0000 bps

    Class-map: CM-EIGRP (match-all)
      34 packets, 2516 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: access-group name EIGRP
      police:
          rate 10 pps, burst 2 packets
        conformed 34 packets, 2516 bytes; actions:
          transmit
        exceeded 0 packets, 0 bytes; actions:
          transmit
        conformed 0 pps, exceeded 0 pps

    Class-map: CM-SSH (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: access-group name SSH
      police:
          cir 50000 bps, bc 1562 bytes
        conformed 0 packets, 0 bytes; actions:
          transmit
        exceeded 0 packets, 0 bytes; actions:
          transmit
        conformed 0000 bps, exceeded 0000 bps

    Class-map: CM-ICMP (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: access-group name ICMP
      police:
          rate 10 pps, burst 2 packets
        conformed 0 packets, 0 bytes; actions:
          transmit
        exceeded 0 packets, 0 bytes; actions:
          drop
        conformed 0 pps, exceeded 0 pps

    Class-map: class-default (match-any)
      333 packets, 19788 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: any
      police:
          cir 12000 bps, bc 1500 bytes
        conformed 333 packets, 19788 bytes; actions:
          transmit
        exceeded 0 packets, 0 bytes; actions:
          transmit
        conformed 0000 bps, exceeded 0000 bps
```

## Step 5: From R2, ping R1.

From R2, use the command **ping 172.16.12.1 repeat 20** to simulate a DoS attach on the control plane on R1. Notice the success rate of 70 percent and the failed pings as indicated by the period symbol (.) among the (successful) exclamation marks. From the output above, this indicates that R1 is dropping every third ICMP packet, based on the policy for rate limiting ICMP traffic to the control plane on R1.

```
R2# ping 172.16.12.1 repeat 20
Type escape sequence to abort.
Sending 20, 100-byte ICMP Echos to 172.16.12.1, timeout is 2 seconds:
!!.!!.!!.!!.!!.!!.!!
Success rate is 70 percent (14/20), round-trip min/avg/max = 1/1/2 ms
```

## Step 6: From R2, access R1 via Telnet.

From R2 enter the **telnet 172.17.12.1** command. Notice that all Telnet traffic to R1 is not successful. This is because our policy is to drop all Telnet traffic directed to the control plane on R1.

```
R2# telnet 172.16.12.1
Trying 172.16.12.1 ...
% Connection timed out; remote host not responding
```

## Step 7: From R2, access A1 via Telnet.

From R2, enter the **telnet 10.10.1.4** command. When prompted for a username / password enter **admin** and **cisco123**. Notice that Telnet traffic from R2 through R1 to A1 is successful.

```
R2# telnet 10.10.1.4
Trying 10.10.1.4 ... Open
 A1, Control Plane Policing

User Access Verification

Username: admin
Password:
A1# exit


[Connection to 10.10.1.4 closed by foreign host]
```

## Step 8: SSH from R2 to R1.

From R2 SSH into R1 using the **ssh -l admin 172.16.12.1** command. When prompted for a password enter **cisco123**. Then enter the **show running-config** command to generate SSH packets.

```
R2# ssh -l admin 172.16.12.1
Password:
 R1, Control Plane Policing
```

```
R1# show running-config
Building configuration...
(output omitted)

R1# exit

[Connection to 172.16.12.1 closed by foreign host]
```

## Step 9: Investigate the CoPP policy on R1.

a. On R1, repeat the **show policy-map control-plane** command to investigate policing of ICMP, Telnet, SSH, EIGRP and other packets.

**Note:** Your packet counts will vary from the following output.

```
R1# show policy-map control-plane
Control Plane

  Service-policy input: PM-COPP

    Class-map: CM-TELNET (match-all)
      4 packets, 232 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: access-group name TELNET
      police:
          cir 8000 bps, bc 1500 bytes
        conformed 4 packets, 232 bytes; actions:
          drop
        exceeded 0 packets, 0 bytes; actions:
          drop
        conformed 0000 bps, exceeded 0000 bps

    Class-map: CM-EIGRP (match-all)
      487 packets, 36038 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: access-group name EIGRP
      police:
          rate 10 pps, burst 2 packets
        conformed 487 packets, 36038 bytes; actions:
          transmit
        exceeded 0 packets, 0 bytes; actions:
          transmit
        conformed 0 pps, exceeded 0 pps

    Class-map: CM-SSH (match-all)
      68 packets, 5776 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: access-group name SSH
      police:
          cir 50000 bps, bc 1562 bytes
        conformed 68 packets, 5776 bytes; actions:
          transmit
        exceeded 0 packets, 0 bytes; actions:
          transmit
        conformed 0000 bps, exceeded 0000 bps

    Class-map: CM-ICMP (match-all)
      10 packets, 1000 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: access-group name ICMP
      police:
          rate 10 pps, burst 2 packets
        conformed 7 packets, 700 bytes; actions:
          transmit
        exceeded 3 packets, 300 bytes; actions:
          drop
        conformed 0 pps, exceeded 0 pps

    Class-map: class-default (match-any)
      4701 packets, 281816 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: any
      police:
          cir 12000 bps, bc 1500 bytes
        conformed 4701 packets, 281816 bytes; actions:
          transmit
        exceeded 0 packets, 0 bytes; actions:
          transmit
        conformed 0000 bps, exceeded 0000 bps
```

b. From the output above note the following:

- Notice that all four Telnet packets were dropped.
- All EIGRP packets conform to the policing rule and are forwarded the control plane for processing.
- All SSH packets also conform to the policing rule and are forwarded to the control plane for processing.
- Ping packets are rate limited to 10 pps. This allows a “burst” of two packets, with every third packet being dropped.

## Part 5: (Challenge) Further Classify Default Traffic

Notice in the output of the **show policy-map control-plane** command that there is a substantial number of packets that are being matched to the **class-default** class map. One solution to help identify the source of these packets would be to configure a “CATCH-ALL” class map. This new class map would collect any remaining traffic that have not matched previous class maps and are destined to the device’s control plane. This CATCH-ALL class would prevent packets from ending up in the **class-default** class map. For example, an ACL called CATCH-ALL could be defined as follows:

```
R1(config)# ip access-list extended CATCH-ALL
R1(config-ext-nacl)# permit icmp any any
R1(config-ext-nacl)# permit udp any any
```

```
R1(config-ext-nacl)# permit tcp any any
R1(config-ext-nacl)# permit ip any any
R1(config-ext-nacl)# exit
```

This ACL could be matched to a class map called CM-CATCH-ALL that could be policed to 50 kbps with a confirm action of transmit and an exceed action of drop.

Implement this new class map in your policy map, repeat the verifications in Part 4, and then view the output of the **show policy-map control-plane** command to see where this default traffic is classified.

## Reflection Questions

1. Why is R2 able to use Telnet through R1 to A1?

**Telnet traffic from R2 through R1 to A1 is seen as data plane traffic and forwarded to A1.**

2. When initially deploying CoPP, how can you prevent disruption of legitimate control plane traffic?

**Initially deploy COPP in “monitoring mode” without any aggressive drop actions. Configure transmit actions, even for traffic exceeding the police rate of your specific class.**

## Router Interface Summary Table

| Router Model | Ethernet Interface #1 | Ethernet Interface #2 | Serial Interface #1 | Serial Interface #2 |
|---|---|---|---|---|
| 1800 | Fast Ethernet 0/0 (F0/0) | Fast Ethernet 0/1 (F0/1) | Serial 0/0/0 (S0/0/0) | Serial 0/0/1 (S0/0/1) |
| 1900 | Gigabit Ethernet 0/0 (G0/0) | Gigabit Ethernet 0/1 (G0/1) | Serial 0/0/0 (S0/0/0) | Serial 0/0/1 (S0/0/1) |
| 2801 | Fast Ethernet 0/0 (F0/0) | Fast Ethernet 0/1 (F0/1) | Serial 0/1/0 (S0/1/0) | Serial 0/1/1 (S0/1/1) |
| 2811 | Fast Ethernet 0/0 (F0/0) | Fast Ethernet 0/1 (F0/1) | Serial 0/0/0 (S0/0/0) | Serial 0/0/1 (S0/0/1) |
| 2900 | Gigabit Ethernet 0/0 (G0/0) | Gigabit Ethernet 0/1 (G0/1) | Serial 0/0/0 (S0/0/0) | Serial 0/0/1 (S0/0/1) |
| 4221 | Gigabit Ethernet 0/0/0 (G0/0/0) | Gigabit Ethernet 0/0/1 (G0/0/1) | Serial 0/1/0 (S0/1/0) | Serial 0/1/1 (S0/1/1) |
| 4300 | Gigabit Ethernet 0/0/0 (G0/0/0) | Gigabit Ethernet 0/0/1 (G0/0/1) | Serial 0/1/0 (S0/1/0) | Serial 0/1/1 (S0/1/1) |

**Note**: To find out how the router is configured, look at the interfaces to identify the type of router and how many interfaces the router has. There is no way to effectively list all the combinations of configurations for each router class. This table includes identifiers for the possible combinations of Ethernet and Serial interfaces in the device. The table does not include any other type of interface, even though a specific router may contain one. An example of this might be an ISDN BRI interface. The string in parenthesis is the legal abbreviation that can be used in Cisco IOS commands to represent the interface.

## Device Configs

### Router R1

```
R1# show running-config
Building configuration...


Current configuration : 2772 bytes
!
version 16.9
service timestamps debug datetime msec
service timestamps log datetime msec
platform qfp utilization monitor load 80
no platform punt-keepalive disable-kernel-core
!
hostname R1
!
boot-start-marker
boot-end-marker
!
vrf definition Mgmt-intf
 !
 address-family ipv4
 exit-address-family
 !
 address-family ipv6
 exit-address-family
!
no aaa new-model
!
no ip domain lookup
ip domain name CCNPv8.CoPP.Lab
!
login on-success log
!
subscriber templating
!
multilink bundle-name authenticated
!
spanning-tree extend system-id
!
username admin privilege 15 secret 9
$9$DeeR16VslSqQwM$DHvIvMYqMZ4z2iBOCR6xXZsWavU3BRSA4HLZG.B8NPE
!
redundancy
 mode none
!
class-map match-all CM-TELNET
 match access-group name TELNET
```

```
class-map match-all CM-ICMP
 match access-group name ICMP
class-map match-all CM-EIGRP
 match access-group name EIGRP
class-map match-all CM-SSH
 match access-group name SSH
!
policy-map PM-COPP
 class CM-TELNET
  police 8000 conform-action drop  exceed-action drop
 class CM-EIGRP
  police rate 10 pps conform-action transmit  exceed-action transmit
 class CM-SSH
  police 50000 conform-action transmit  exceed-action transmit
 class CM-ICMP
  police rate 10 pps conform-action transmit  exceed-action drop
 class class-default
  police 12000 conform-action transmit  exceed-action transmit
!
interface GigabitEthernet0/0/0
 ip address 172.16.12.1 255.255.255.252
 negotiation auto
!
interface GigabitEthernet0/0/1
 ip address 10.10.1.1 255.255.255.0
 negotiation auto
!
interface Serial0/1/0
 no ip address
!
interface Serial0/1/1
 no ip address
!
router eigrp 123
 network 10.10.1.0 0.0.0.255
 network 172.16.12.0 0.0.0.3
 eigrp router-id 0.0.0.1
!
ip forward-protocol nd
no ip http server
ip http secure-server
ip tftp source-interface GigabitEthernet0
!
ip access-list extended CATCH-ALL
 permit icmp any any
 permit udp any any
 permit tcp any any
 permit ip any any
ip access-list extended EIGRP
 permit eigrp any any
```

```
ip access-list extended ICMP
 permit icmp any any
ip access-list extended SSH
 permit tcp any any eq 22
ip access-list extended TELNET
 permit tcp any any eq telnet
!
control-plane
 service-policy input PM-COPP
!
banner motd ^C R1, Control Plane Policing  ^C
!
line con 0
 exec-timeout 0 0
 logging synchronous
 transport input none
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login local
 transport input telnet ssh
!
end
```

## Router R2

```
R2# show running-config
Building configuration...


Current configuration : 1810 bytes
!
version 16.9
service timestamps debug datetime msec
service timestamps log datetime msec
platform qfp utilization monitor load 80
no platform punt-keepalive disable-kernel-core
!
hostname R2
!
boot-start-marker
boot-end-marker
!
no aaa new-model
!
no ip domain lookup
ip domain name CCNPv8.CoPP.Lab
!
login on-success log
```

```
!
subscriber templating
!
multilink bundle-name authenticated
!
spanning-tree extend system-id
!
username admin privilege 15 secret 9
$9$BpMc23YNSFD1Gr$4x3z8z07FbIZxPnS/0mK1e0qGLzi7GnmnqI3e0beNcA
!
redundancy
 mode none
!
!
interface GigabitEthernet0/0/0
 ip address 172.16.12.2 255.255.255.252
 negotiation auto
!
interface GigabitEthernet0/0/1
 no ip address
 negotiation auto
!
!
router eigrp 123
 network 172.16.12.0 0.0.0.3
 eigrp router-id 0.0.0.2
!
ip forward-protocol nd
no ip http server
ip http secure-server
ip tftp source-interface GigabitEthernet0
!
!
control-plane
!
banner motd ^C R2, Control Plane Policing  ^C
!
line con 0
 exec-timeout 0 0
 logging synchronous
 transport input none
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login local
 transport input telnet ssh
!
end
```

## Switch A1

```
A1# show running-config brief
Building configuration...

Current configuration : 2659 bytes
!
version 15.2
no service pad
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname A1
!
boot-start-marker
boot-end-marker
!
!
username admin privilege 15 secret 9
$9$q93nz/NOzZfxYL$PldFC0CJSzBd9qGi1CM6YhxEiB.d1hALRz1ItxMGaUs
no aaa new-model
system mtu routing 1500
!
no ip domain-lookup
ip domain-name CCNPv8.COPP.LAB
!
spanning-tree mode rapid-pvst
spanning-tree extend system-id
!
vlan internal allocation policy ascending
!
!
interface FastEthernet0/1
 switchport mode access
!
interface FastEthernet0/2
 shutdown
!
interface FastEthernet0/3
 shutdown
!
interface FastEthernet0/4
 shutdown
!
interface FastEthernet0/5
 shutdown
!
interface FastEthernet0/6
 shutdown
```

```
!
interface FastEthernet0/7
 shutdown
!
interface FastEthernet0/8
 shutdown
!
interface FastEthernet0/9
 shutdown
!
interface FastEthernet0/10
 shutdown
!
interface FastEthernet0/11
 shutdown
!
interface FastEthernet0/12
 shutdown
!
interface FastEthernet0/13
 shutdown
!
interface FastEthernet0/14
 shutdown
!
interface FastEthernet0/15
 shutdown
!
interface FastEthernet0/16
 shutdown
!
interface FastEthernet0/17
 shutdown
!
interface FastEthernet0/18
 shutdown
!
interface FastEthernet0/19
 shutdown
!
interface FastEthernet0/20
 shutdown
!
interface FastEthernet0/21
 shutdown
!
interface FastEthernet0/22
 shutdown
!
interface FastEthernet0/23
```

```
 shutdown
!
interface FastEthernet0/24
 shutdown
!
interface GigabitEthernet0/1
 shutdown
!
interface GigabitEthernet0/2
 shutdown
!
interface Vlan1
 ip address 10.10.1.4 255.255.255.0
!
ip default-gateway 10.10.1.1
ip http server
ip http secure-server
!
no vstack
banner motd ^C A1, Control Plane Policing  ^C
!
line con 0
 exec-timeout 0 0
 logging synchronous
line vty 0 4
 login local
 transport input telnet ssh
line vty 5 15
 login local
 transport input telnet ssh
!
end
```