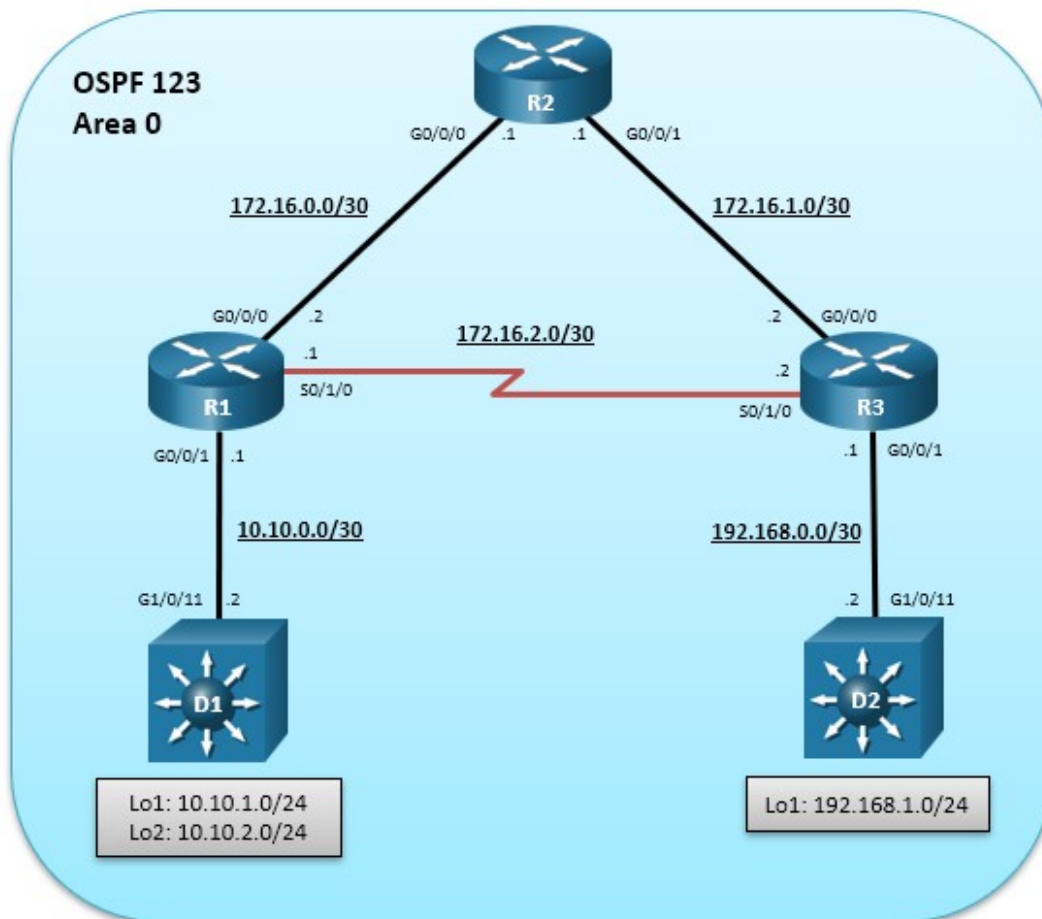


## Lab - Path Control Using PBR (Instructor Version)

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

### Topology



### Addressing Table

Device	Interface	IPv4 Address	Subnet Mask
D1	G0/0/11	10.10.0.2	255.255.255.252
	Loopback 1	10.10.1.1	255.255.255.0
	Loopback 2	10.10.2.1	255.255.255.0
R1	G0/0/0	172.16.0.2	255.255.255.252
	G0/0/1	10.10.0.1	255.255.255.252
	S0/1/0	172.16.2.1	255.255.255.252

Device	Interface	IPv4 Address	Subnet Mask
R2	G0/0/0	172.16.0.1	255.255.255.252
	G0/0/1	172.16.1.1	255.255.255.252
R3	G0/0/0	172.16.1.2	255.255.255.252
	G0/0/1	192.168.0.1	255.255.255.252
	S0/1/0	172.16.2.2	255.255.255.252
D2	G0/0/11	192.168.0.2	255.255.255.252
	Loopback 1	192.168.1.1	255.255.255.0

## Objectives

In this lab, you will learn how to use policy-based routing to influence path selection.

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

### Part 2: Configure and Verify Routing

### Part 3: Configure PBR to Provide Path Control

### Part 4: Configure Local PBR to Provide Path Control

## Background / Scenario

In this scenario, you want to experiment with policy-based routing (PBR) to see how it is implemented and study how it could be used to influence path selection.

Your task is to connect and configure a 3 router and 2 Layer 3 switch OSPF routing domain and verify normal path selection. You will alter the traffic flow for PC2 going to PC3 using PBR. You will also use Cisco IOS IP SLA with PBR to achieve dynamic path control. Finally, you will configure PBR for traffic originating locally on router R1.

**Note:** This lab is an exercise in configuring distribute lists and redistribution using ALCs, prefix lists, and route maps.

**Note:** The routers used with CCNP hands-on labs are Cisco 4221 with Cisco IOS XE Release 16.9.4 (universalk9 image). The switches used in the labs are Cisco Catalyst 3650 with Cisco IOS XE Release 16.9.4 (universalk9 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 devices 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

- 3 Routers (Cisco 4221 with Cisco IOS XE Release 16.9.4 universal image or comparable)
- 2 Switches (Catalyst 3650s with Cisco IOS XE Release 16.9.4 universalk9 image).
- 1 PC (Choice of operating system with terminal emulation program installed)

### Instructions

#### 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 routers.

##### 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
line con 0
  logging sync
  exec-time 0 0
exit
banner motd # This is R1, Path Control Using PBR #
interface G0/0/0
  description Connection to R2
  ip add 172.16.0.2 255.255.255.252
  no shut
exit
interface S0/1/0
  description Serial Connection to R3
  ip add 172.16.2.1 255.255.255.252
  no shut
exit
interface G0/0/1
  description Connection to D1
  ip add 10.10.0.1 255.255.255.252
  no shut
exit
```

###### Router R2

```
hostname R2
no ip domain lookup
line con 0
  logging sync
  exec-time 0 0
exit
banner motd # This is R2, Path Control Using PBR #
interface G0/0/0
```

```
description Connection to R1
ip add 172.16.0.1 255.255.255.252
no shut
exit
interface GigabitEthernet0/0/1
description Connection to R3
ip address 172.16.1.1 255.255.255.252
no shut
exit
```

### Router R3

```
hostname R3
no ip domain lookup
line con 0
logging sync
exec-time 0 0
exit
banner motd # This is R3, Path Control Using PBR #
interface G0/0/0
description Connection to R2
ip add 172.16.1.2 255.255.255.252
no shut
exit
interface S0/1/0
description Serial Connection to R1
ip add 172.16.2.2 255.255.255.252
no shut
exit
interface G0/0/1
description Connection to D2
ip add 192.168.0.1 255.255.255.252
no shut
exit
```

### Switch D1

```
hostname D1
no ip domain lookup
line con 0
exec-timeout 0 0
logging synchronous
exit
banner motd # This is D1, Path Control Using PBR #
interface G1/0/11
no switchport
description Connects to R1
ip address 10.10.0.2 255.255.255.252
```

```
no shut
exit
interface Loopback 1
description Interface simulates network
ip ospf network point-to-point
ip address 10.10.1.1 255.255.255.0
exit
interface Loopback 2
description Interface simulates network
ip ospf network point-to-point
ip address 10.10.2.1 255.255.255.0
exit
```

### Switch D2

```
hostname D2
no ip domain lookup
line con 0
logging sync
exec-time 0 0
exit
banner motd # This is D2, Path Control Using PBR #
interface G1/0/11
no switchport
description Connects to R3
ip address 192.168.0.2 255.255.255.252
no shut
exit
interface Loopback 1
description Interface simulates network
ip ospf network point-to-point
ip address 192.168.1.1 255.255.255.0
exit
```

- b. Save the running configuration to startup-config.

## Part 2: Configure and Verify Routing

In Part 2, you will implement OSPF routing for the routing domain and verify end to end routing.

### Step 1: Configure Routing.

In this step, you will configure OSPF.

- a. On D1, advertise the connected networks using OSPF process ID 123. Also assign D1 the router ID of 1.1.1.2 and set the reference bandwidth to recognize Gigabit Ethernet interfaces.

```
D1(config)# ip routing
D1(config)# router ospf 123
D1(config-router)# router-id 1.1.1.2
D1(config-router)# auto-cost reference-bandwidth 1000
```

```
% OSPF: Reference bandwidth is changed.
Please ensure reference bandwidth is consistent across all routers.
D1(config-router)# network 10.10.0.0 0.0.0.3 area 0
D1(config-router)# network 10.10.1.0 0.0.0.255 area 0
D1(config-router)# network 10.10.2.0 0.0.0.255 area 0
D1(config-router)# end
```

- b. On R1, advertise the connected networks using OSPF process ID 123. Also assign R1 the router ID of 1.1.1.1 and set the reference bandwidth to recognize Gigabit Ethernet interfaces.

```
R1(config)# router ospf 123
R1(config-router)# router-id 1.1.1.1
R1(config-router)# auto-cost reference-bandwidth 1000
% OSPF: Reference bandwidth is changed.
Please ensure reference bandwidth is consistent across all routers.
R1(config-router)# network 10.10.0.0 0.0.0.3 area 0
R1(config-router)# network 172.16.0.0 0.0.0.3 area 0
R1(config-router)# network 172.16.2.0 0.0.0.3 area 0
R1(config-router)# end
R1#
*Feb 19 17:00:40.661: %OSPF-5-ADJCHG: Process 123, Nbr 1.1.1.2 on
GigabitEthernet0/0/1 from LOADING to FULL, Loading Done
```

- c. On R2, advertise the connected networks using OSPF process ID 123. Also assign R2 the router ID of 2.2.2.1 and set the reference bandwidth to recognize Gigabit Ethernet interfaces.

```
R2(config)# router ospf 123
R2(config-router)# router-id 2.2.2.1
R2(config-router)# auto-cost reference-bandwidth 1000
% OSPF: Reference bandwidth is changed.
Please ensure reference bandwidth is consistent across all routers.
*Feb 19 17:02:34.016: %OSPF-6-DFT_OPT: Protocol timers for fast convergence
areEnabled.172.
R2(config-router)# network 172.16.0.0 0.0.0.3 area 0
R2(config-router)# network 172.16.1.0 0.0.0.3 area 0
R2(config-router)# end
R2#
*Feb 19 17:02:42.460: %OSPF-5-ADJCHG: Process 123, Nbr 1.1.1.1 on
GigabitEthernet0/0/0 from LOADING to FULL, Loading Done
```

- d. On R3, advertise the connected networks using OSPF process ID 123. Also assign R3 the router ID of 3.3.3.1 and set the reference bandwidth to recognize Gigabit Ethernet interfaces.

```
R3(config)# router ospf 123
R3(config-router)# router-id 3.3.3.1
R3(config-router)# auto-cost reference-bandwidth 1000
% OSPF: Reference bandwidth is changed.
Please ensure reference bandwidth is consistent across all routers.
R3(config-router)# network 192.168.0.0 0.0.0.3 area 0
R3(config-router)# network 172.16.1.0 0.0.0.3 area 0
R3(config-router)# network 172.16.2.0 0.0.0.3 area 0
```

```
R3(config-router)# end
R3#
*Feb 19 17:03:56.362: %OSPF-5-ADJCHG: Process 123, Nbr 1.1.1.1 on Serial0/1/0
from LOADING to FULL, Loading Done
R3#
*Feb 19 17:09:38.978: %OSPF-5-ADJCHG: Process 123, Nbr 2.2.2.1 on
GigabitEthernet0/0/0 from LOADING to FULL, Loading Done
```

- e. On D2, advertise the connected networks using OSPF process ID 123. Also assign D2 the router ID of 3.3.3.2 and set the reference bandwidth to recognize Gigabit Ethernet interfaces.

```
D2(config)# ip routing
D2(config)# router ospf 123
D2(config-router)# router-id 3.3.3.2
D2(config-router)# auto-cost reference-bandwidth 1000
% OSPF: Reference bandwidth is changed.
    Please ensure reference bandwidth is consistent across all routers.
D2(config-router)# network 192.168.0.0 0.0.0.3 area 0
D2(config-router)# network 192.168.1.0 0.0.0.255 area 0
D2(config-router)# end
D2#
*Feb 19 17:29:46.627: %OSPF-5-ADJCHG: Process 123, Nbr 3.3.3.1 on
GigabitEthernet1/0/11 from LOADING to FULL, Loading Done
```

### Step 2: Verify OSPF routing

- a. Before configuring PBR, verify the current routing table on all devices. All routing tables look accurate.

```
D1# show ip route ospf | begin Gateway
Gateway of last resort is not set

    172.16.0.0/30 is subnetted, 3 subnets
O       172.16.0.0 [110/2] via 10.10.0.1, 00:11:36, GigabitEthernet1/0/11
O       172.16.1.0 [110/3] via 10.10.0.1, 00:10:10, GigabitEthernet1/0/11
O       172.16.2.0 [110/497] via 10.10.0.1, 00:11:36, GigabitEthernet1/0/11
    192.168.0.0/30 is subnetted, 1 subnets
O       192.168.0.0 [110/4] via 10.10.0.1, 00:09:57, GigabitEthernet1/0/11
O       192.168.1.0/24 [110/14] via 10.10.0.1, 00:09:57, GigabitEthernet1/0/11

D2# show ip route ospf | begin Gateway
Gateway of last resort is not set

    10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
O       10.10.0.0/30 [110/4] via 192.168.0.1, 00:03:55, GigabitEthernet1/0/11
O       10.10.1.0/24 [110/5] via 192.168.0.1, 00:03:55, GigabitEthernet1/0/11
O       10.10.2.0/24 [110/5] via 192.168.0.1, 00:03:55, GigabitEthernet1/0/11
    172.16.0.0/30 is subnetted, 3 subnets
O       172.16.0.0 [110/3] via 192.168.0.1, 00:03:55, GigabitEthernet1/0/11
O       172.16.1.0 [110/2] via 192.168.0.1, 00:03:55, GigabitEthernet1/0/11
O       172.16.2.0 [110/499] via 192.168.0.1, 00:03:55, GigabitEthernet1/0/1
```

R1# **show ip route ospf | begin Gateway**

Gateway of last resort is not set

```
10.0.0.0/8 is variably subnetted, 4 subnets, 3 masks
O      10.10.1.0/24 [110/11] via 10.10.0.2, 00:12:21, GigabitEthernet0/0/1
O      10.10.2.0/24 [110/11] via 10.10.0.2, 00:12:21, GigabitEthernet0/0/1
172.16.0.0/16 is variably subnetted, 5 subnets, 2 masks
O      172.16.1.0/30 [110/2] via 172.16.0.1, 00:10:55, GigabitEthernet0/0/0
192.168.0.0/30 is subnetted, 1 subnets
O      192.168.0.0 [110/3] via 172.16.0.1, 00:10:42, GigabitEthernet0/0/0
O      192.168.1.0/24 [110/13] via 172.16.0.1, 00:10:42, GigabitEthernet0/0/0
```

R2# **show ip route ospf | begin Gateway**

Gateway of last resort is not set

```
10.0.0.0/8 is variably subnetted, 5 subnets, 3 masks
O      10.10.0.0/30 [110/2] via 172.16.0.2, 00:12:26, GigabitEthernet0/0/0
O      10.10.1.0/24 [110/12] via 172.16.0.2, 00:12:26, GigabitEthernet0/0/0
O      10.10.2.0/24 [110/12] via 172.16.0.2, 00:12:26, GigabitEthernet0/0/0
172.16.0.0/16 is variably subnetted, 5 subnets, 2 masks
O      172.16.2.0/30
        [110/497] via 172.16.0.2, 00:12:26, GigabitEthernet0/0/0
192.168.0.0/30 is subnetted, 1 subnets
O      192.168.0.0 [110/2] via 172.16.1.2, 00:11:22, GigabitEthernet0/0/1
O      192.168.1.0/24 [110/12] via 172.16.1.2, 00:11:22, GigabitEthernet0/0/1
```

R3# **show ip route ospf | begin Gateway**

Gateway of last resort is not set

```
10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
O      10.10.0.0/30 [110/3] via 172.16.1.1, 00:12:56, GigabitEthernet0/0/0
O      10.10.1.0/24 [110/13] via 172.16.1.1, 00:12:56, GigabitEthernet0/0/0
O      10.10.2.0/24 [110/13] via 172.16.1.1, 00:12:56, GigabitEthernet0/0/0
172.16.0.0/16 is variably subnetted, 5 subnets, 2 masks
O      172.16.0.0/30 [110/2] via 172.16.1.1, 00:12:56, GigabitEthernet0/0/0
O      192.168.1.0/24 [110/11] via 192.168.0.2, 00:12:44, GigabitEthernet0/0/1
```

### Step 3: Verify end-to-end connectivity and path taken

- From any device, verify connectivity to all configured destinations using the following TCL script. All pings should be successful. Troubleshoot if necessary.

**tclsh**

```
foreach address {
10.10.0.1
10.10.0.2
10.10.1.1
10.10.2.1
172.16.0.1
```



```
172.16.0.2
172.16.1.1
172.16.1.2
172.16.2.1
172.16.2.2
192.168.0.1
192.168.0.2
192.168.1.1
} { ping $address }
```

- b. On D1, ping the D2 Loopback interface 192.168.1.1 address from the Lo1 interface as shown. The pings should be successful.

```
D1# ping 192.168.1.1 source 10.10.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds:
Packet sent with a source address of 10.10.1.1
!!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 2/2/3 ms
```

- c. Next, identify the path taken to D2 Lo1 interface using the **traceroute** command as shown. Notice that the path taken for the packets sourced from the D1 Lo1 LAN is going through R1 --> R2 --> R3 --> D2.

```
D1# traceroute 192.168.1.1 source 10.10.1.1
Type escape sequence to abort.
Tracing the route to 192.168.1.1
VRF info: (vrf in name/id, vrf out name/id)
 1 10.10.0.1 2 msec 2 msec 2 msec
 2 172.16.0.1 2 msec 2 msec 2 msec
 3 172.16.1.2 2 msec 2 msec 2 msec
 4 192.168.0.2 3 msec * 3 msec
```

Why is the path not taking the shorter R1 --> R3 --> D2 path?

**Because the serial interfaces between routers R1 and R3 are using a slower serial link, giving it a higher metric. All other interfaces are faster Gigabit Ethernet producing lower (and preferred) metrics.**

- d. Now ping and traceroute the D2 Lo1 interface from the D1 Loopback 2 interface as shown. It is also taking the same path.

```
D1# ping 192.168.1.1 source lo 2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds:
Packet sent with a source address of 10.10.2.1
!!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 2/2/4 ms
```

```
D1# traceroute 192.168.1.1 source lo 2
Type escape sequence to abort.
Tracing the route to 192.168.1.1
```

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

```
1 10.10.0.1 2 msec 2 msec 1 msec
2 172.16.0.1 1 msec 2 msec 1 msec
3 172.16.1.2 2 msec 2 msec 2 msec
4 192.168.0.2 3 msec * 3 msec
```

- e. Display the OSPF routes in the routing table of R1. R1 forwards all packets destined to the 192.168.1.0/24 network out of its G0/0/0 interface to R2.

```
R1# show ip route ospf | begin Gateway
```

```
Gateway of last resort is not set
```

```
10.0.0.0/8 is variably subnetted, 4 subnets, 3 masks
O    10.10.1.0/24 [110/2] via 10.10.0.2, 00:19:56, GigabitEthernet0/0/1
O    10.10.2.0/24 [110/2] via 10.10.0.2, 00:19:56, GigabitEthernet0/0/1
172.16.0.0/16 is variably subnetted, 5 subnets, 2 masks
O    172.16.1.0/30 [110/2] via 172.16.0.1, 00:18:50, GigabitEthernet0/0/0
192.168.0.0/30 is subnetted, 1 subnets
O    192.168.0.0 [110/3] via 172.16.0.1, 00:18:37, GigabitEthernet0/0/0
O    192.168.1.0/24 [110/4] via 172.16.0.1, 00:18:33, GigabitEthernet0/0/0
```

- f. Display how R1 learned about the 192.168.1.0 network. R1 learned of the network from R2 (i.e., 172.16.0.1) who originally learned it from D2 (i.e., 3.3.3.2).

```
R1# show ip route 192.168.1.0
```

```
Routing entry for 192.168.1.0/24
```

```
Known via "ospf 123", distance 110, metric 4, type intra area
```

```
Last update from 172.16.0.1 on GigabitEthernet0/0/0, 00:20:27 ago
```

```
Routing Descriptor Blocks:
```

```
* 172.16.0.1, from 3.3.3.2, 00:20:27 ago, via GigabitEthernet0/0/0
```

```
Route metric is 4, traffic share count is 1
```

## Part 3: Configure PBR to Provide Path Control

Recall that route maps can be used for:

- **Redistribution** - Route maps provide more options and flexibility to the **redistribute** command.
- **Policy-based routing (PBR)** - PBR allows an administrator to define routing policy other than basic destination-based routing using the routing table. The route map is applied to an interface using the **ip policy route-map** interface configuration command.
- **BGP** - Route maps are the primary tools for implementing BGP policy and allows an administrator to do path control and provide sophisticated manipulation of BGP path attributes. The route map is applied using the BGP **neighbor** router configuration command.

In this part, you will use PBR to configure source-based IP routing. Specifically, you will override the default IP routing decision based on the OSPF-acquired routing information for selected IP source-to-destination flows and apply a different next-hop router.

Recall that routers normally forward packets to destination addresses based on information in their routing table. By using PBR, you can implement policies that selectively cause packets to take different paths based on source address, protocol type, or application type. Therefore, PBR overrides the router's normal routing behavior.

Configuring PBR involves configuring a route map with **match** and **set** commands and then applying the route map to the interface.

The steps required to implement path control include the following:

- Choose the path control tool to use. Path control tools manipulate or bypass the IP routing table. For PBR, **route-map** commands are used.
- Implement the traffic-matching configuration, specifying which traffic will be manipulated. The **match** commands are used within route maps.
- Define the action for the matched traffic using **set** commands within route maps.
- Apply the route map to incoming traffic.

### Step 1: Configure PBR on R1.

As a test, you will configure the following policy on router R1:

- All traffic sourced from D1 Lo1 LAN must take the R1 --> R2 --> R3 --> D2 path.
  - All traffic sourced from D1 Lo2 LAN must take the R1 --> R3 --> D2 path.
- a. On R1, create a standard named ACL called **Lo2-ACL** to identify the D1 Loopback 2 (i.e., 10.10.2.0/24) LAN.

```
R1(config)# ip access-list standard Lo2-ACL
R1(config-std-nacl)# remark ACL matches D1 Lo2 traffic
R1(config-std-nacl)# permit 10.10.2.0 0.0.0.255
R1(config-std-nacl)# exit
```

- b. Create a route map called **R1-to-R3** that matches Lo2-ACL and sets the next-hop interface to the R3 serial 0/1/0 interface.

```
R1(config)# route-map R1-to-R3 permit
R1(config-route-map)# description RM to forward Lo2 traffic to R3
R1(config-route-map)# match ip address Lo2-ACL
R1(config-route-map)# set ip next-hop 172.16.2.2
R1(config-route-map)# exit
```

- c. Apply the R1-to-R3 route map to the G0/0/1 interface using the **ip policy route-map** command.

```
R1(config)# interface g0/0/1
R1(config-if)# ip policy route-map R1-to-R3
R1(config-if)# end
```

- d. On R1, display the policy and matches using the **show route-map** command.

```
R1# show route-map
route-map R1-to-R3, permit, sequence 10
  Match clauses:
    ip address (access-lists): Lo2-ACL
  Set clauses:
    ip next-hop 172.16.2.2
  Policy routing matches: 0 packets, 0 bytes
```

**Note:** There are currently no matches because no packets matching the ACL have passed through R1 G0/0/1.

- e. On R1, verify that the R1-to-R3 route map has been applied to the G0/0/1 interface.

```
R1# show ip policy
Interface      Route map
Gi0/0/1        R1-to-R3
```

### Step 2: Test the policy.

Now you are ready to test the policy configured on R1.

- a. From D1, test the policy with the **traceroute** command, using D1 Lo1 interface as the source network.

```
D1# traceroute 192.168.1.1 source lo 1
Type escape sequence to abort.
Tracing the route to 192.168.1.1
VRF info: (vrf in name/id, vrf out name/id)
 1 10.10.0.1 2 msec 1 msec 1 msec
 2 172.16.0.1 2 msec 2 msec 2 msec
 3 172.16.1.2 2 msec 4 msec 2 msec
 4 192.168.0.2 3 msec * 3 msec
```

Notice the path taken for the packet sourced from D1 Lo 1 LAN A is still going through R1 --> R2 --> R3 --> D2.

Why is the traceroute traffic not using the R3 --> R1 path as specified in the R1-to-R3 policy?

**It does not take the PBR-specified path because LAN A does not meet the criteria specified in the Lo1-ACL access list.**

- b. Now test the policy with the **traceroute** command, using D1 Lo2 interface as the source network. Now the path taken for the packet sourced from D1 Lo 2 LAN is R1 --> R3 --> D2, as expected.

```
D1# traceroute 192.168.1.1 source lo 2
Type escape sequence to abort.
Tracing the route to 192.168.1.1
VRF info: (vrf in name/id, vrf out name/id)
 1 10.10.0.1 2 msec 2 msec 1 msec
 2 172.16.2.2 3 msec 2 msec 3 msec
 3 192.168.0.2 3 msec * 4 msec
```

- c. On R1, display the policy and matches using the **show route-map** command.

**Note:** There are now matches to the policy because packets matching the ACL have passed through R1 G0/0/1 interface. The number of packet and bytes may differ in your implementation.

```
R1# show route-map
route-map R1-to-R3, permit, sequence 10
Match clauses:
  ip address (access-lists): Lo2-ACL
Set clauses:
  ip next-hop 172.16.2.2
Policy routing matches: 5 packets, 210 bytes
```

### Part 4: Configure Local PBR to Provide Path Control

How would you policy route packets generated by a router? The answer is to configure Local PBR.

Local PBR is a feature to policy route locally generated traffic. Local PBR policies are applied to the router with the **ip local policy route-map** global config command.

In this part, you will configure R1 to policy route all router generated traffic over the R1 to R3 link.

### Step 1: Configure Local PBR on R1.

- a. Verify the path that R1 currently takes without local PBR configured. R1 sends traffic to R2 then R3 and finally D2 as expected.

```
R1# traceroute 192.168.1.1
Type escape sequence to abort.
Tracing the route to 192.168.1.1
VRF info: (vrf in name/id, vrf out name/id)
 1 172.16.0.1 1 msec 1 msec 1 msec
 2 172.16.1.2 1 msec 2 msec 1 msec
 3 192.168.0.2 2 msec * 2 msec
```

- b. On R1, create a named extended ACL called **R1-TRAFFIC** which matches all IP generated packets from R1 and destined to the D2 192.162.1.0/24 network.

```
R1(config)# ip access-list extended R1-TRAFFIC
R1(config-ext-nacl)# permit ip any 192.168.1.0 0.0.0.255
R1(config-ext-nacl)# exit
```

- c. On R1, create a route map called **LOCAL-PBR** that permits traffic matching the R1-TRAFFIC ACL and redirects it to the R3 172.16.2.2 interface.

```
R1(config)# route-map LOCAL-PBR permit
R1(config-route-map)# match ip address R1-TRAFFIC
R1(config-route-map)# set ip next-hop 172.16.2.2
R1(config-route-map)# exit
```

- d. Create a local PBR policy that matches the LOCAL-PBR route map.

```
R1(config)# ip local policy route-map LOCAL-PBR
R1(config)# exit
```

### Step 2: Test Local PBR on R1.

- a. Verify the path taken by R1 to reach the 192.168.1.0/24 LAN. The traffic generated by R1 and going to 192.168.1.0/24 is now policy routed directly to R3 (i.e., 172.16.2.2).

```
R1# traceroute 192.168.1.1
Type escape sequence to abort.
Tracing the route to 192.168.1.1
VRF info: (vrf in name/id, vrf out name/id)
 1 172.16.2.2 2 msec 1 msec 2 msec
 2 192.168.0.2 2 msec * 2 msec
```

- b. Verify the path taken by R1 to reach other networks. The traffic takes the normal OSPF generated path and is not policy routed.

```
R1# traceroute 192.168.0.2
Type escape sequence to abort.
Tracing the route to 192.168.0.2
VRF info: (vrf in name/id, vrf out name/id)
 1 172.16.0.1 1 msec 1 msec 1 msec
 2 172.16.1.2 1 msec 2 msec 1 msec
 3 192.168.0.2 4 msec * 2 msec
```

- c. Verify the route-map counters. The local PBR policy has matched packets.

**Note:** The number of packets and bytes may differ in your implementation.

```
R1# show route-map
route-map R1-to-R3, permit, sequence 10
  Match clauses:
    ip address (access-lists): Lo2-ACL
  Set clauses:
    ip next-hop 172.16.2.2
  Policy routing matches: 6 packets, 252 bytes
route-map LOCAL-PBR, permit, sequence 10
  Match clauses:
    ip address (access-lists): ICMP-TRAFFIC
  Set clauses:
    ip next-hop 172.16.2.2
  Policy routing matches: 32 packets, 2384 bytes
```

### 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 – Final

#### Router R1

```
R1# show run
Building configuration...
```

```
Current configuration : 4180 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  
!  
no aaa new-model  
!  
no ip domain lookup  
!  
login on-success log  
!  
subscriber templating  
!  
multilink bundle-name authenticated  
!  
!  
spanning-tree extend system-id  
!  
redundancy  
mode none  
!  
interface GigabitEthernet0/0/0  
description Connection to R2  
ip address 172.16.0.2 255.255.255.252  
negotiation auto  
!  
interface GigabitEthernet0/0/1  
description Connection to D1  
ip address 10.10.0.1 255.255.255.252  
ip policy route-map R1-to-R3  
negotiation auto  
!  
interface Serial0/1/0  
description Serial Connection to R3  
ip address 172.16.2.1 255.255.255.252  
!  
interface Serial0/1/1  
!  
router ospf 123  
router-id 1.1.1.1  
auto-cost reference-bandwidth 1000  
network 10.10.0.0 0.0.0.3 area 0  
network 172.16.0.0 0.0.0.3 area 0
```

## Lab - Path Control Using PBR

---

```
network 172.16.2.0 0.0.0.3 area 0
!
ip local policy route-map LOCAL-PBR
ip forward-protocol nd
ip http server
ip http authentication local
ip http secure-server
!
ip access-list standard Lo2-ACL
remark ACL matches D1 Lo2 traffic
permit 10.10.2.0 0.0.0.255
!
ip access-list extended R1-TRAFFIC
permit ip any 192.168.1.0 0.0.0.255
!
route-map R1-to-R3 permit 10
description RM to forward Lo2 traffic to R3
match ip address Lo2-ACL
set ip next-hop 172.16.2.2
!
route-map LOCAL-PBR permit 10
match ip address R1-TRAFFIC
set ip next-hop 172.16.2.2
!
control-plane
!
banner motd ^C This is R1, Path Control Using PBR ^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
!
end
```

### Router R2

```
R2# show run
Building configuration...

Current configuration : 3565 bytes
!
version 16.9
service timestamps debug datetime msec
```



## Lab - Path Control Using PBR

---

```
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
!
login on-success log
!
subscriber templating
!
multilink bundle-name authenticated
!
spanning-tree extend system-id
!
redundancy
mode none
!
interface GigabitEthernet0/0/0
description Connection to R1
ip address 172.16.0.1 255.255.255.252
negotiation auto
!
interface GigabitEthernet0/0/1
description Connection to R3
ip address 172.16.1.1 255.255.255.252
negotiation auto
!
router ospf 123
router-id 2.2.2.1
auto-cost reference-bandwidth 1000
network 172.16.0.0 0.0.0.3 area 0
network 172.16.1.0 0.0.0.3 area 0
!
ip forward-protocol nd
ip http server
ip http authentication local
ip http secure-server
!
control-plane
!
banner motd ^C This is R2, Path Control Using PBR ^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
!
end
```

### Router R3

```
R3# show run
```

```
Building configuration...
```

```
Current configuration : 3782 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 R3
!
boot-start-marker
boot-end-marker
!
no aaa new-model
!
no ip domain lookup
!
login on-success log
!
subscriber templating
!
multilink bundle-name authenticated
!
spanning-tree extend system-id
!
redundancy
  mode none
!
interface GigabitEthernet0/0/0
  description Connection to R2
  ip address 172.16.1.2 255.255.255.252
  negotiation auto
!
```

## Lab - Path Control Using PBR

---

```
interface GigabitEthernet0/0/1
description Connection to D2
ip address 192.168.0.1 255.255.255.252
negotiation auto
!
interface Serial0/1/0
description Serial Connection to R1
ip address 172.16.2.2 255.255.255.252
!
interface Serial0/1/1
no ip address
shutdown
!
router ospf 123
router-id 3.3.3.1
auto-cost reference-bandwidth 1000
network 172.16.1.0 0.0.0.3 area 0
network 172.16.2.0 0.0.0.3 area 0
network 192.168.0.0 0.0.0.3 area 0
!
ip forward-protocol nd
ip http server
ip http authentication local
ip http secure-server
!
control-plane
!
banner motd ^C This is R3, Path Control Using PBR ^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
!
end
```

### Switch D1

```
D1# show run
Building configuration...

Current configuration : 4285 bytes
!
version 16.9
no service pad
```

## Lab - Path Control Using PBR

---

```
service timestamps debug datetime msec
service timestamps log datetime msec
no platform punt-keepalive disable-kernel-core
!
hostname D1
!
vrf definition Mgmt-vrf
!
  address-family ipv4
  exit-address-family
!
  address-family ipv6
  exit-address-family
!
no aaa new-model
switch 1 provision ws-c3650-24ps
!
ip routing
!
no ip domain lookup
!
login on-success log
!
license boot level ipservicesk9
!
diagnostic bootup level minimal
!
spanning-tree mode rapid-pvst
spanning-tree extend system-id
!
redundancy
  mode sso
!
transceiver type all
  monitoring
!
class-map match-any system-cpp-police-topology-control
  description Topology control
class-map match-any system-cpp-police-sw-forward
  description Sw forwarding, L2 LVX data, LOGGING
class-map match-any system-cpp-default
  description Inter FED, EWLC control, EWLC data
class-map match-any system-cpp-police-sys-data
  description Learning cache ovfl, High Rate App, Exception, EGR Exception,
NFLSAMPLED DATA, RPF Failed
class-map match-any system-cpp-police-punt-webauth
  description Punt Webauth
class-map match-any system-cpp-police-l2lvx-control
  description L2 LVX control packets
class-map match-any system-cpp-police-forus
```

```
    description Forus Address resolution and Forus traffic
class-map match-any system-cpp-police-multicast-end-station
    description MCAST END STATION
class-map match-any system-cpp-police-multicast
    description Transit Traffic and MCAST Data
class-map match-any system-cpp-police-l2-control
    description L2 control
class-map match-any system-cpp-police-dot1x-auth
    description DOT1X Auth
class-map match-any system-cpp-police-data
    description ICMP redirect, ICMP_GEN and BROADCAST
class-map match-any system-cpp-police-stackwise-virt-control
    description Stackwise Virtual
class-map match-any non-client-nrt-class
class-map match-any system-cpp-police-routing-control
    description Routing control and Low Latency
class-map match-any system-cpp-police-protocol-snooping
    description Protocol snooping
class-map match-any system-cpp-police-dhcp-snooping
    description DHCP snooping
class-map match-any system-cpp-police-system-critical
    description System Critical and Gold Pkt
!
policy-map system-cpp-policy
!
interface Loopback1
    description Interface simulates network
    ip address 10.10.1.1 255.255.255.0
    ip ospf network point-to-point
!
interface Loopback2
    description Interface simulates network
    ip address 10.10.2.1 255.255.255.0
    ip ospf network point-to-point
!
interface GigabitEthernet0/0
    vrf forwarding Mgmt-vrf
    no ip address
    negotiation auto
!
interface GigabitEthernet1/0/1
!
interface GigabitEthernet1/0/2
!
interface GigabitEthernet1/0/3
!
interface GigabitEthernet1/0/4
!
interface GigabitEthernet1/0/5
!
```

```
interface GigabitEthernet1/0/6
!
interface GigabitEthernet1/0/7
!
interface GigabitEthernet1/0/8
!
interface GigabitEthernet1/0/9
!
interface GigabitEthernet1/0/10
!
interface GigabitEthernet1/0/11
  description Connects to R1
  no switchport
  ip address 10.10.0.2 255.255.255.252
!
interface GigabitEthernet1/0/12
!
interface GigabitEthernet1/0/13
!
interface GigabitEthernet1/0/14
!
interface GigabitEthernet1/0/15
!
interface GigabitEthernet1/0/16
!
interface GigabitEthernet1/0/17
!
interface GigabitEthernet1/0/18
!
interface GigabitEthernet1/0/19
!
interface GigabitEthernet1/0/20
!
interface GigabitEthernet1/0/21
!
interface GigabitEthernet1/0/22
!
interface GigabitEthernet1/0/23
!
interface GigabitEthernet1/0/24
!
interface GigabitEthernet1/1/1
!
interface GigabitEthernet1/1/2
!
interface GigabitEthernet1/1/3
!
interface GigabitEthernet1/1/4
!
interface Vlan1
```

## Lab - Path Control Using PBR

---

```
no ip address
!
router ospf 123
router-id 1.1.1.2
auto-cost reference-bandwidth 1000
network 10.10.0.0 0.0.0.3 area 0
network 10.10.1.0 0.0.0.255 area 0
network 10.10.2.0 0.0.0.255 area 0
!
ip forward-protocol nd
ip http server
ip http secure-server
!
control-plane
service-policy input system-cpp-policy
!
banner motd ^C This is D1, Path Control Using PBR ^C
!
line con 0
exec-timeout 0 0
logging synchronous
stopbits 1
line aux 0
stopbits 1
line vty 0 4
login
line vty 5 15
login
!
end
```

### Switch D2

```
D2# show run
Building configuration...

Current configuration : 4126 bytes
!
version 16.9
no service pad
service timestamps debug datetime msec
service timestamps log datetime msec
no platform punt-keepalive disable-kernel-core
!
hostname D2
!
vrf definition Mgmt-vrf
!
address-family ipv4
exit-address-family
```

```
!  
address-family ipv6  
exit-address-family  
!  
no aaa new-model  
switch 1 provision ws-c3650-24ps  
!  
ip routing  
!  
no ip domain lookup  
!  
login on-success log  
!  
license boot level ipservicesk9  
!  
diagnostic bootup level minimal  
!  
spanning-tree mode rapid-pvst  
spanning-tree extend system-id  
!  
redundancy  
mode sso  
!  
transceiver type all  
monitoring  
!  
class-map match-any system-cpp-police-topology-control  
description Topology control  
class-map match-any system-cpp-police-sw-forward  
description Sw forwarding, L2 LVX data, LOGGING  
class-map match-any system-cpp-default  
description Inter FED, EWLC control, EWLC data  
class-map match-any system-cpp-police-sys-data  
description Learning cache ovfl, High Rate App, Exception, EGR Exception,  
NFLSAMPLED DATA, RPF Failed  
class-map match-any system-cpp-police-punt-webauth  
description Punt Webauth  
class-map match-any system-cpp-police-l2lvx-control  
description L2 LVX control packets  
class-map match-any system-cpp-police-forus  
description Forus Address resolution and Forus traffic  
class-map match-any system-cpp-police-multicast-end-station  
description MCAST END STATION  
class-map match-any system-cpp-police-multicast  
description Transit Traffic and MCAST Data  
class-map match-any system-cpp-police-l2-control  
description L2 control  
class-map match-any system-cpp-police-dot1x-auth  
description DOT1X Auth  
class-map match-any system-cpp-police-data
```



```
    description ICMP redirect, ICMP_GEN and BROADCAST
class-map match-any system-cpp-police-stackwise-virt-control
    description Stackwise Virtual
class-map match-any non-client-nrt-class
class-map match-any system-cpp-police-routing-control
    description Routing control and Low Latency
class-map match-any system-cpp-police-protocol-snooping
    description Protocol snooping
class-map match-any system-cpp-police-dhcp-snooping
    description DHCP snooping
class-map match-any system-cpp-police-system-critical
    description System Critical and Gold Pkt
!
policy-map system-cpp-policy
!
interface Loopback1
    description Interface simulates network
    ip address 192.168.1.1 255.255.255.0
    ip ospf network point-to-point
!
interface GigabitEthernet0/0
    vrf forwarding Mgmt-vrf
    no ip address
    negotiation auto
!
interface GigabitEthernet1/0/1
!
interface GigabitEthernet1/0/2
!
interface GigabitEthernet1/0/3
!
interface GigabitEthernet1/0/4
!
interface GigabitEthernet1/0/5
!
interface GigabitEthernet1/0/6
!
interface GigabitEthernet1/0/7
!
interface GigabitEthernet1/0/8
!
interface GigabitEthernet1/0/9
!
interface GigabitEthernet1/0/10
!
interface GigabitEthernet1/0/11
    description Connects to R3
    no switchport
    ip address 192.168.0.2 255.255.255.252
!
```

```
interface GigabitEthernet1/0/12
!
interface GigabitEthernet1/0/13
!
interface GigabitEthernet1/0/14
!
interface GigabitEthernet1/0/15
!
interface GigabitEthernet1/0/16
!
interface GigabitEthernet1/0/17
!
interface GigabitEthernet1/0/18
!
interface GigabitEthernet1/0/19
!
interface GigabitEthernet1/0/20
!
interface GigabitEthernet1/0/21
!
interface GigabitEthernet1/0/22
!
interface GigabitEthernet1/0/23
!
interface GigabitEthernet1/0/24
!
interface GigabitEthernet1/1/1
!
interface GigabitEthernet1/1/2
!
interface GigabitEthernet1/1/3
!
interface GigabitEthernet1/1/4
!
interface Vlan1
no ip address
!
router ospf 123
router-id 3.3.3.2
auto-cost reference-bandwidth 1000
network 192.168.0.0 0.0.0.3 area 0
network 192.168.1.0 0.0.0.255 area 0
!
ip forward-protocol nd
ip http server
ip http secure-server
!
control-plane
service-policy input system-cpp-policy
!
```

## Lab - Path Control Using PBR

---

```
banner motd ^C This is D2, Path Control Using PBR ^C
!  
line con 0  
  exec-timeout 0 0  
  logging synchronous  
  stopbits 1  
line aux 0  
  stopbits 1  
line vty 0 4  
  login  
line vty 5 15  
  login  
!  
end
```