In this discovery lab, you will learn how to use [PBR](https://ondemandelearning.cisco.com/cisco-cte/ccnp-route/search?query=PBR&type=terms) to influence path selection. First, you will verify normal traffic paths, as selected by the traditional destination-based routing. Then, you will configure PBR to alter the traffic flow for one client station. Finally, you will verify both the PBR configuration and the new traffic path.

PC and Notebook, as used in this topology, are simulated by routers running Cisco IOS SoftwareYou are using a **US English** keyboard layout. This cannot be changed once the lab has initialized.

Visit [Device Help](https://client.certsite.net/help/devices/index.html) for info about changing the OS keyboard layout and screen resolution after lab initialization.

**You may navigate away from this page once you begin initializing the lab.**

You will be notified once the devices are ready.

**Step 1**

Verify traffic paths from Notebook and PC to the HQ router using traceroute.

**Answer**

|  |
| --- |
| NOTEBOOK# **traceroute 192.168.100.1**  Type escape sequence to abort.  Tracing the route to 192.168.100.1  VRF info: (vrf in name/id, vrf out name/id)  1 192.168.110.1 1 msec 0 msec 1 msec  2 10.10.20.1 1 msec 1 msec \* |
| PC# **traceroute 192.168.100.1**  Type escape sequence to abort.  Tracing the route to 192.168.100.1  VRF info: (vrf in name/id, vrf out name/id)  1 192.168.110.1 2 msec 0 msec 1 msec  2 10.10.20.1 0 msec 1 msec \* |

As you can see from the output, traffic from both clients is going through the faster [WAN](https://ondemandelearning.cisco.com/cisco-cte/ccnp-route/search?query=WAN&type=terms) link (network 10.10.20.0/30) to the HQ router.

**Step 2**

Define [ACL](https://ondemandelearning.cisco.com/cisco-cte/ccnp-route/search?query=ACL&type=terms) on BR1 to match traffic coming from the Notebook client.

**Answer**

Create a new named access list PBR-ACL to match all [IP](https://ondemandelearning.cisco.com/cisco-cte/ccnp-route/search?query=IP&type=terms) traffic coming from the Notebook client.

|  |
| --- |
| BR1# **configure terminal**  BR1(config)# **ip access-list extended PBR-ACL**  BR1(config-ext-nacl)# **permit ip host 192.168.110.10 any** |

**Step 3**

Create a route map that matches traffic using the defined ACL.

**Answer**

Create a new route map PBR-RP, and match the access list PBR-ACL.

|  |
| --- |
| BR1(config)# **route-map PBR-RP**  BR1(config-route-map)# **match ip address PBR-ACL** |

**Step 4**

Alter the traffic path for the Notebook client.

**Answer**

To alter the traffic path, change the next hop with the set ip next-hop route-map subcommand.

|  |
| --- |
| BR1(config-route-map)# **set ip next-hop 10.10.10.1** |

By setting the next hop to 10.10.10.1, router BR1 will forward all matched traffic over the serial link.

**Step 5**

Verify the configured route map.

**Answer**

Use the show route-mapglobal-levelcommand to verify the configured route maps.

|  |
| --- |
| BR1# **show route-map**  route-map PBR-RP, permit, sequence 10  Match clauses:  ip address (access-lists): PBR-ACL  Set clauses:  ip next-hop 10.10.10.1  Policy routing matches: 0 packets, 0 bytes |

The route map output states that incoming traffic defined with the access list PBR-ACL is forwarded to 10.10.10.1, which corresponds to the IP address of HQ configured on the serial link.

**Step 6**

Apply the route map to the inbound interface.

**Answer**

Apply the route map with the ip policy interface-level command. The command should be applied to the inbound interface, that is, to an interface where the traffic is entering the router.

|  |
| --- |
| BR1(config)# **interface ethernet 0/1**  BR1(config-if)# **ip policy route-map PBR-RP** |

If you want to alter the traffic path for locally generated traffic, that is, traffic that is generated by the local router itself, you need to attach the route map with the global configuration mode command ip local policy route-map route-map-name.

**Step 7**

Verify the configured policy.

**Answer**

Use the show ip policy global-level command to verify the configured policies.

|  |
| --- |
| BR1# **show ip policy**  Interface Route map  Ethernet0/1 PBR-RP |

As you can see from the output, the route map PBR-RP is applied to the Ethernet0/1 interface.

**Step 8**

Verify traffic paths again from Notebook and PC to the HQ router using traceroute.

**Answer**

|  |
| --- |
| NOTEBOOK# **traceroute 192.168.100.1**  Type escape sequence to abort.  Tracing the route to 192.168.100.1  VRF info: (vrf in name/id, vrf out name/id)  1 192.168.110.1 1 msec 1 msec 1 msec  2 10.10.10.1 4 msec 6 msec \* |
| PC# **traceroute 192.168.100.1**  Type escape sequence to abort.  Tracing the route to 192.168.100.1  VRF info: (vrf in name/id, vrf out name/id)  1 192.168.110.1 1 msec 1 msec 0 msec  2 10.10.20.1 1 msec 1 msec \* |

The traffic path for the PC client remained the same—the traffic is going through the [WAN](https://ondemandelearning.cisco.com/cisco-cte/ccnp-route/search?query=WAN&type=terms) link (network 10.10.20.0/30).

On the other hand, traffic for the Notebook client is now going through the serial link (network 10.10.10.0/30).

**Step 9**

Enable PBR debugging on BR1 to analyze the detailed policy-based routing operation. Initiate traffic from the Notebook and the PC to the HQ router using ICMP.

**Answer**

To verify the actual traffic path in the lab environment, you can use the debug ip policy command. For the command to display the output, traffic must be present on the network. If that is not the case, you must generate traffic, with the ping command, for example.

|  |
| --- |
| BR1# **debug ip policy**  Policy routing debugging is on |
| NOTEBOOK# **ping 192.168.100.1**  Type escape sequence to abort.  Sending 5, 100-byte ICMP Echos to 192.168.100.1, timeout is 2 seconds:  !!!!!  Success rate is 100 percent (5/5), round-trip min/avg/max = 4/4/4 ms |
| BR1#  \*Apr 17 10:21:39.012: IP: s=192.168.110.10 (Ethernet0/1), d=192.168.100.1, len 100, FIB policy match  \*Apr 17 10:21:39.012: IP: s=192.168.110.10 (Ethernet0/1), d=192.168.100.1, len 100, PBR Counted  \*Apr 17 10:21:39.012: IP: s=192.168.110.10 (Ethernet0/1), d=192.168.100.1, g=10.10.10.1, len 100, FIB policy routed  \*Apr 17 10:21:39.016: IP: s=192.168.110.10 (Ethernet0/1), d=192.168.100.1, len 100, FIB policy match  \*Apr 17 10:21:39.016: IP: s=192.168.110.10 (Ethernet0/1), d=192.168.100.1, len 100, PBR Counted  \*Apr 17 10:21:39.016: IP: s=192.168.110.10 (Ethernet0/1), d=192.168.100.1, g=10.10.10.1, len 100, FIB policy routed  <... output omitted ...> |
| PC# **ping 192.168.100.1**  Type escape sequence to abort.  Sending 5, 100-byte ICMP Echos to 192.168.100.1, timeout is 2 seconds:  !!!!!  Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms |
| BR1#  \*Apr 17 10:22:47.401: IP: s=192.168.110.20 (Ethernet0/1), d=192.168.100.1, len 100, FIB policy rejected(no match) - normal forwarding  \*Apr 17 10:22:47.402: IP: s=192.168.110.20 (Ethernet0/1), d=192.168.100.1, len 100, FIB policy rejected(no match) - normal forwarding  \*Apr 17 10:22:47.402: IP: s=192.168.110.20 (Ethernet0/1), d=192.168.100.1, len 100, FIB policy rejected(no match) - normal forwarding  \*Apr 17 10:22:47.403: IP: s=192.168.110.20 (Ethernet0/1), d=192.168.100.1, len 100, FIB policy rejected(no match) - normal forwarding  \*Apr 17 10:22:47.404: IP: s=192.168.110.20 (Ethernet0/1), d=192.168.100.1, len 100, FIB policy rejected(no match) - normal forwarding |

The first part of the output displays information about traffic from the Notebook client. You can see that traffic flow information is displayed in the output block, beginning with "IP: ..", which includes source, incoming interface, and destination. The output clearly states that there is a policy match and that traffic is forwarded to 10.10.10.1.

The second part of the output displays information about traffic from the PC client. After traffic flow information, the output clearly states that there is no policy match and that traffic is treated with normal (destination-based) forwarding.