Chapter 5: Path Control Implementation

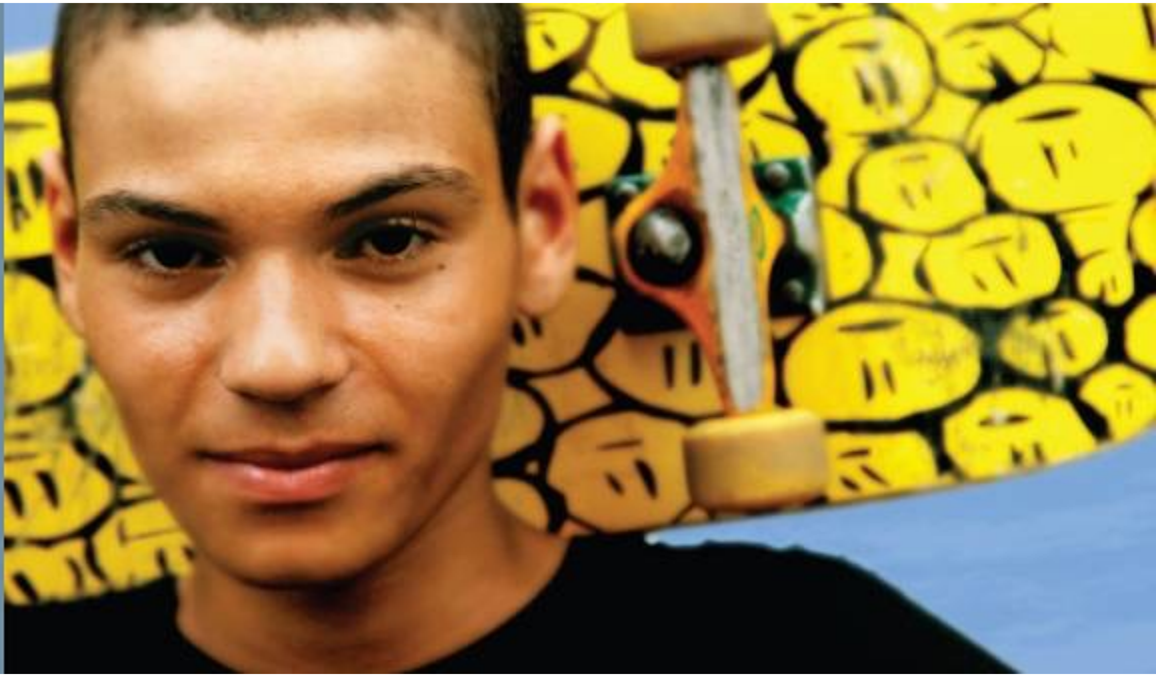
CCNP ROUTE: Implementing IP Routing

Cisco | Networking Academy®
Mind Wide Open™

Chapter 5 Objectives

- Using Cisco Express Forwarding Switching
- Understanding Path Control
- Implementing Path Control Using Policy-Based Routing
- Implementing Path Control Using Cisco IOS IP SLAs

Using Cisco Express Forwarding Switching



Using Cisco Express Forwarding Switching

- Describe the different switching mechanisms that a Cisco router uses
- Describe how Cisco Express Forwarding (CEF) works
- Describe how to verify that CEF is working
- Describe how to verify the content of the CEF tables
- Describe how to enable and disable CEF by interface and globally

Control and Data Plane

- A Layer 3 device employs a distributed architecture in which the control plane and data plane are relatively independent.
- For example, the exchange of routing protocol information is performed in the control plane by the route processor, whereas data packets are forwarded in the data plane by an interface micro-coded processor.

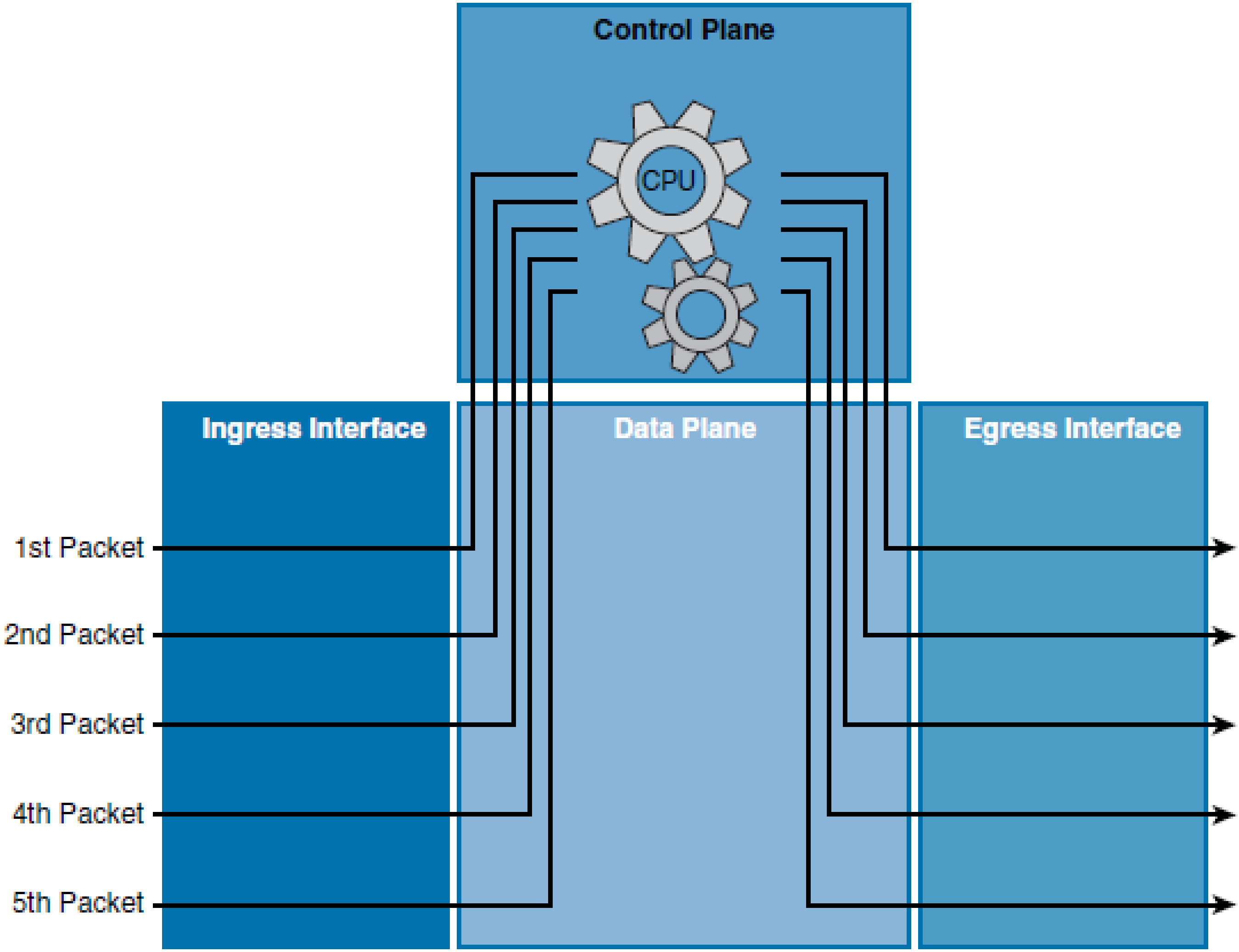
Control and Data Plane

- The main functions of the control layer between the routing protocol and the firmware data plane microcode include the following:
  - Managing the internal data and control circuits for the packet-forwarding and control functions.
  - Extracting the other routing and packet-forwarding-related control information from Layer 2 and Layer 3 bridging and routing protocols and the configuration data, and then conveying the information to the interface module for control of the data plane.
  - Collecting the data plane information, such as traffic statistics, from the interface module to the route processor.
  - Handling certain data packets that are sent from the Ethernet interface modules to the route processor.

Cisco Switching Mechanisms

- Process switching
- Fast switching
- Cisco Express Forwarding

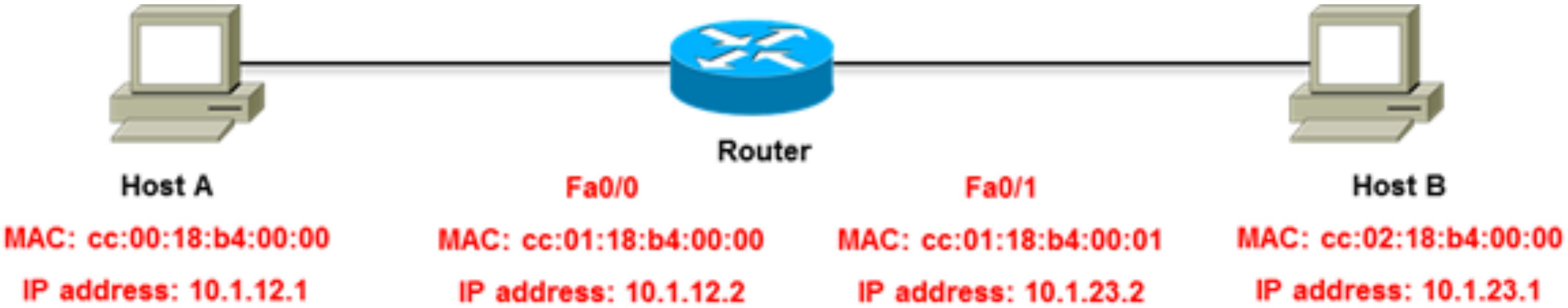
Process switching

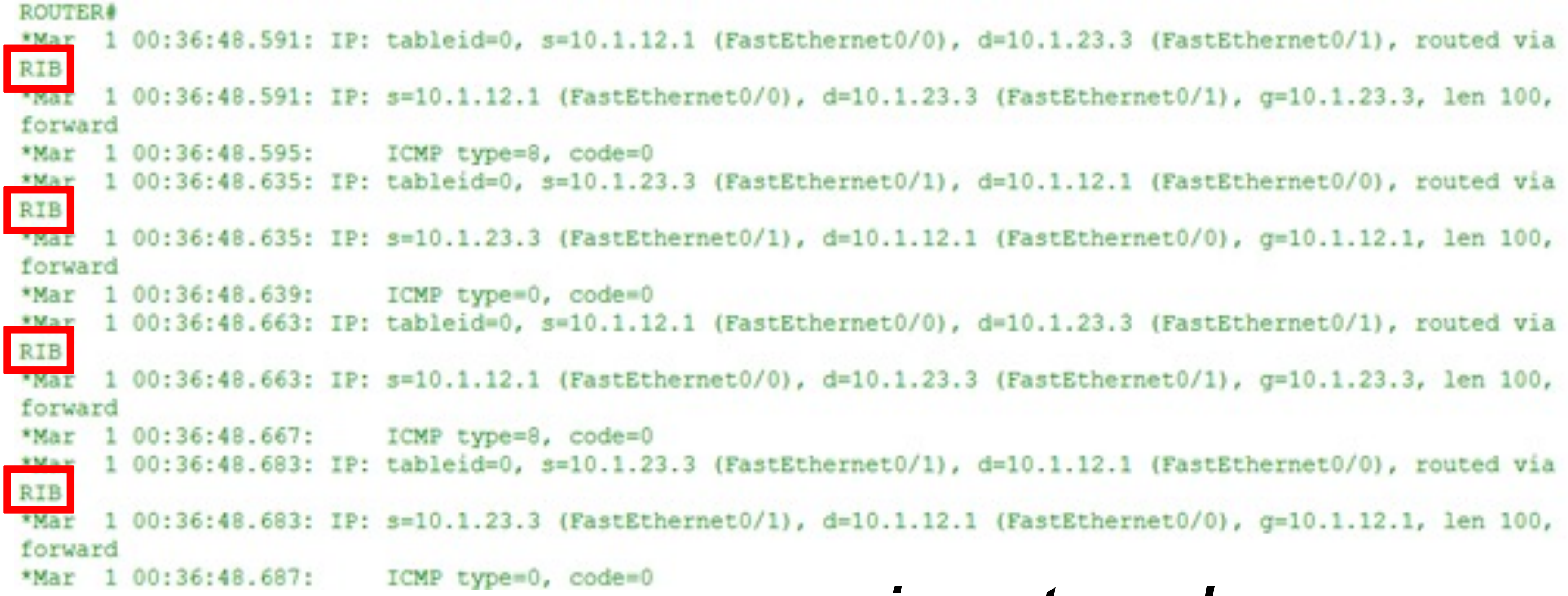


Process switching

- This switching method is the slowest of the three methods.
- Every packet is examined by the CPU in the control plane and all forwarding decisions are made in software.
- When a packet arrives on the ingress interface, it is forwarded to the control plane where the CPU matches the destination address with an entry in its routing table.
- It then determines the exit interface and forwards the packet.
- The router does this for every packet, even if the destination is the same for a stream of packets.
- Process switching is the most CPU-intensive method that is available in Cisco routers. It greatly degrades performance and is generally used only as a last resort or during troubleshooting.

Process switching

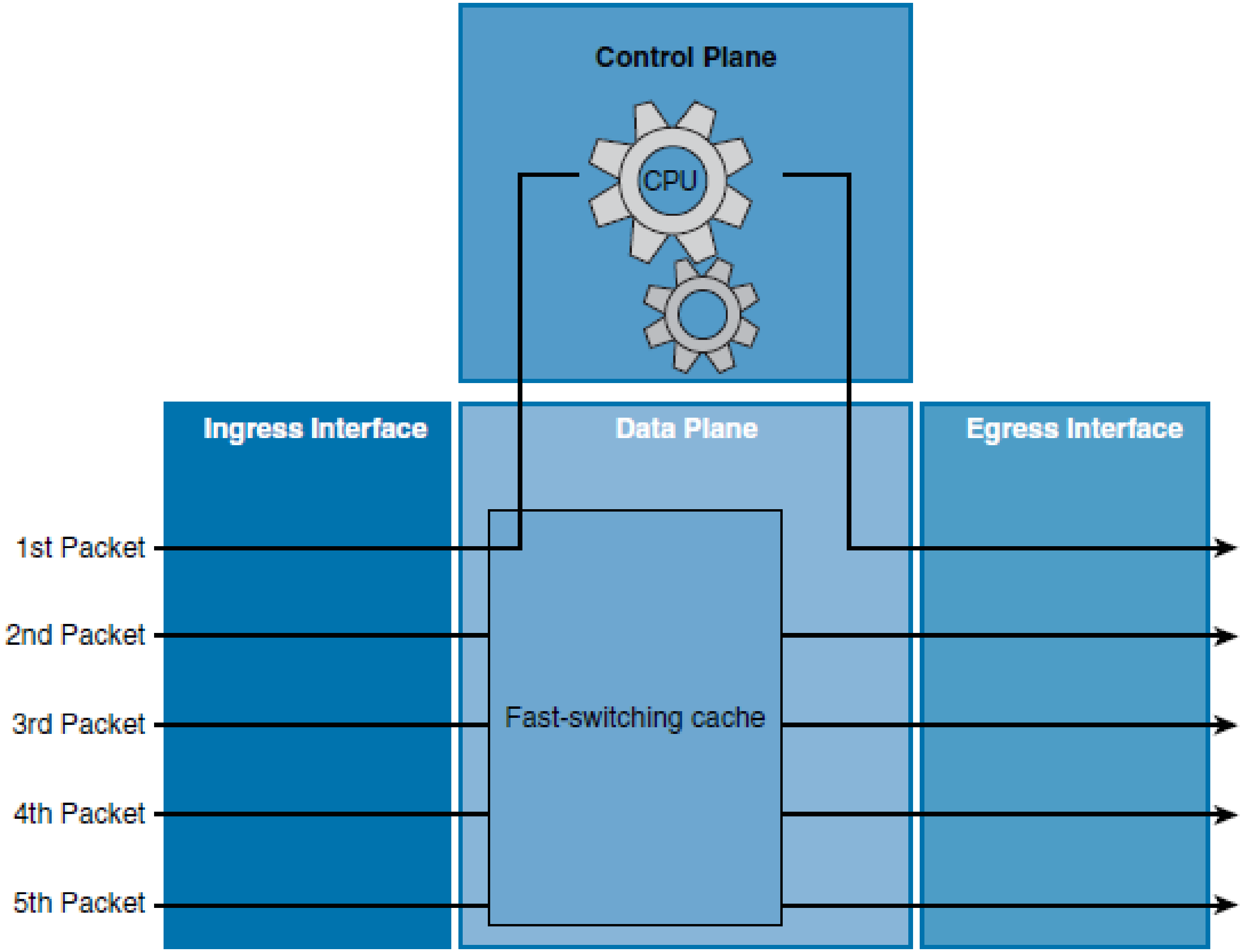




```
ROUTER#
*Mar  1 00:36:48.591: IP: tableid=0, s=10.1.12.1 (FastEthernet0/0), d=10.1.23.3 (FastEthernet0/1), routed via
RIB
*Mar  1 00:36:48.591: IP: s=10.1.12.1 (FastEthernet0/0), d=10.1.23.3 (FastEthernet0/1), g=10.1.23.3, len 100,
forward
*Mar  1 00:36:48.595:     ICMP type=8, code=0
*Mar  1 00:36:48.635: IP: tableid=0, s=10.1.23.3 (FastEthernet0/1), d=10.1.12.1 (FastEthernet0/0), routed via
RIB
*Mar  1 00:36:48.635: IP: s=10.1.23.3 (FastEthernet0/1), d=10.1.12.1 (FastEthernet0/0), g=10.1.12.1, len 100,
forward
*Mar  1 00:36:48.639:     ICMP type=0, code=0
*Mar  1 00:36:48.663: IP: tableid=0, s=10.1.12.1 (FastEthernet0/0), d=10.1.23.3 (FastEthernet0/1), routed via
RIB
*Mar  1 00:36:48.663: IP: s=10.1.12.1 (FastEthernet0/0), d=10.1.23.3 (FastEthernet0/1), g=10.1.23.3, len 100,
forward
*Mar  1 00:36:48.667:     ICMP type=8, code=0
*Mar  1 00:36:48.683: IP: tableid=0, s=10.1.23.3 (FastEthernet0/1), d=10.1.12.1 (FastEthernet0/0), routed via
RIB
*Mar  1 00:36:48.683: IP: s=10.1.23.3 (FastEthernet0/1), d=10.1.12.1 (FastEthernet0/0), g=10.1.12.1, len 100,
forward
*Mar  1 00:36:48.687:     ICMP type=0, code=0
```

no ip route-cache
debug ip packet [detail]

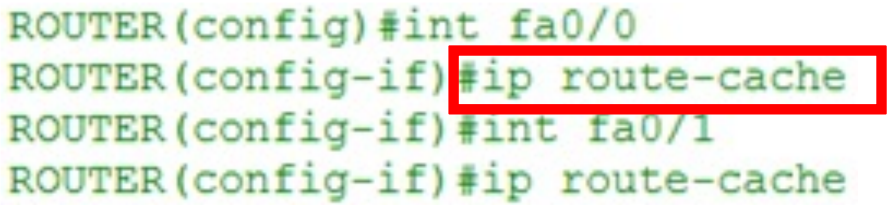
Fast switching



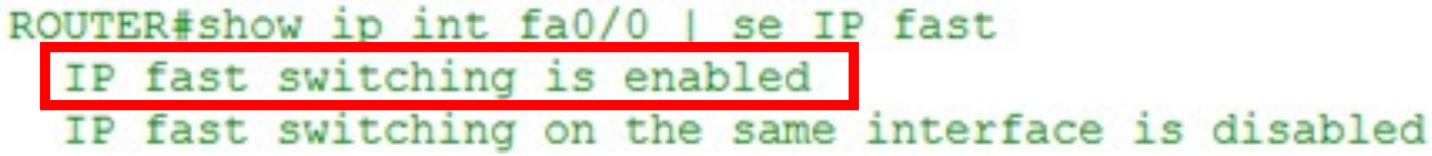
Fast switching

- This switching method is faster than process switching.
- With fast switching, the initial packet of a traffic flow is process switched.
- This means that it is examined by the CPU and the forwarding decision is made in software.
- However, the forwarding decision is also stored in the data plane hardware fast-switching cache.
- When subsequent frames in the flow arrive, the destination is found in the hardware fast-switching cache and the frames are then forwarded without interrupting the CPU.

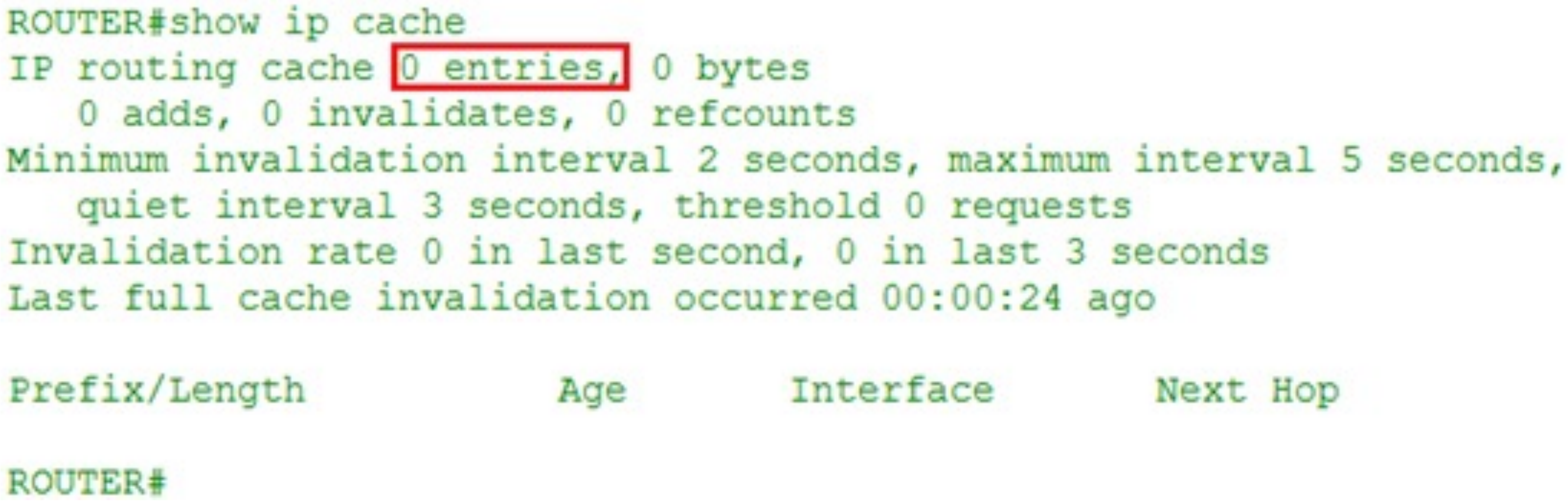
Fast switching



```
ROUTER(config)#int fa0/0
ROUTER(config-if)#ip route-cache
ROUTER(config-if)#int fa0/1
ROUTER(config-if)#ip route-cache
```

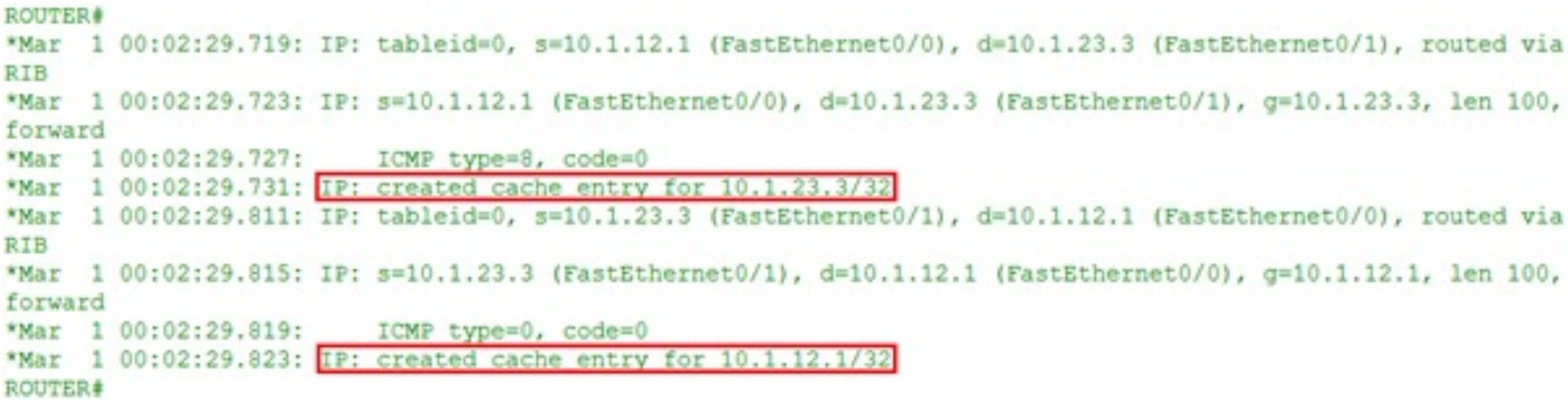


```
ROUTER#show ip int fa0/0 | se IP fast
  IP fast switching is enabled
  IP fast switching on the same interface is disabled
```

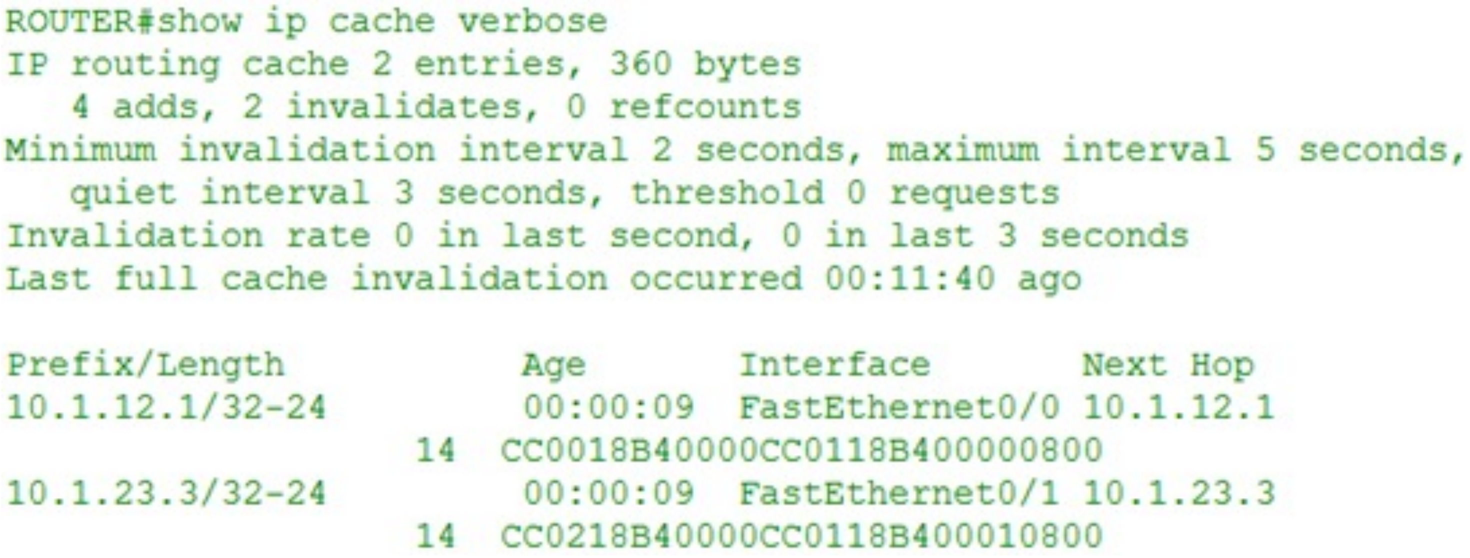


```
ROUTER#show ip cache
IP routing cache 0 entries, 0 bytes
   0 adds, 0 invalidates, 0 refcounts
Minimum invalidation interval 2 seconds, maximum interval 5 seconds,
   quiet interval 3 seconds, threshold 0 requests
Invalidation rate 0 in last second, 0 in last 3 seconds
Last full cache invalidation occurred 00:00:24 ago

Prefix/Length           Age       Interface       Next Hop

ROUTER#
```

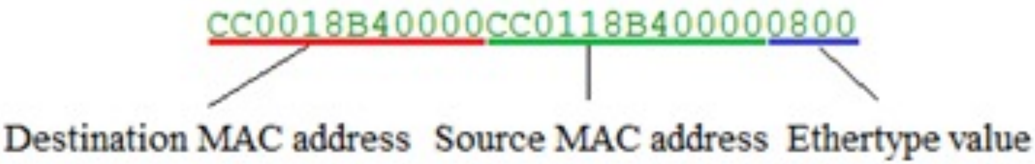
Fast switching



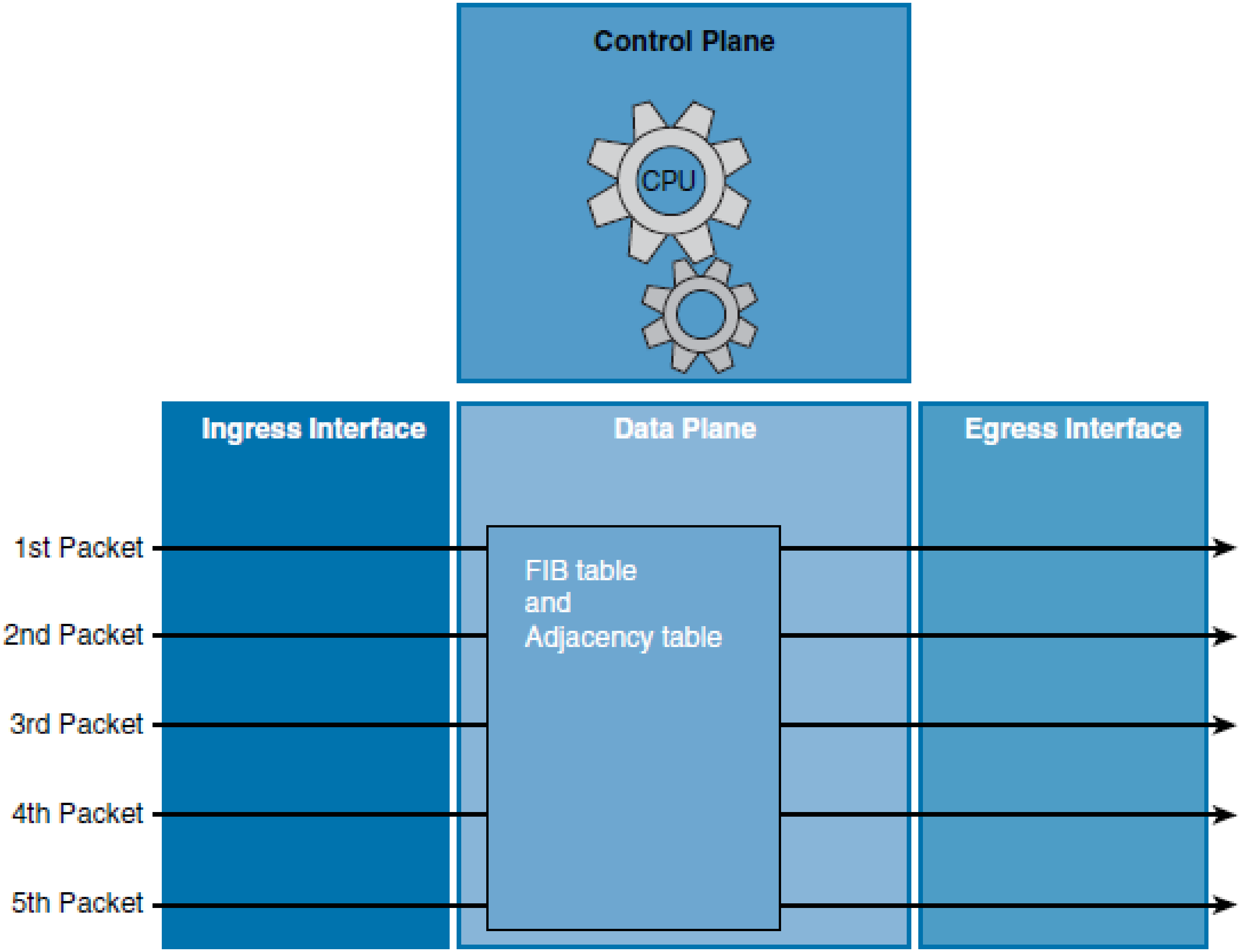
```
ROUTER#
*Mar  1 00:02:29.719: IP: tableid=0, s=10.1.12.1 (FastEthernet0/0), d=10.1.23.3 (FastEthernet0/1), routed via
RIB
*Mar  1 00:02:29.723: IP: s=10.1.12.1 (FastEthernet0/0), d=10.1.23.3 (FastEthernet0/1), g=10.1.23.3, len 100,
forward
*Mar  1 00:02:29.727:     ICMP type=8, code=0
*Mar  1 00:02:29.731: IP: created cache entry for 10.1.23.3/32
*Mar  1 00:02:29.811: IP: tableid=0, s=10.1.23.3 (FastEthernet0/1), d=10.1.12.1 (FastEthernet0/0), routed via
RIB
*Mar  1 00:02:29.815: IP: s=10.1.23.3 (FastEthernet0/1), d=10.1.12.1 (FastEthernet0/0), g=10.1.12.1, len 100,
forward
*Mar  1 00:02:29.819:     ICMP type=0, code=0
*Mar  1 00:02:29.823: IP: created cache entry for 10.1.12.1/32
ROUTER#
```



```
ROUTER#show ip cache verbose
IP routing cache 2 entries, 360 bytes
   4 adds, 2 invalidates, 0 refcounts
Minimum invalidation interval 2 seconds, maximum interval 5 seconds,
   quiet interval 3 seconds, threshold 0 requests
Invalidation rate 0 in last second, 0 in last 3 seconds
Last full cache invalidation occurred 00:11:40 ago

Prefix/Length           Age       Interface       Next Hop
10.1.12.1/32-24         00:00:09  FastEthernet0/0 10.1.12.1
                   14  CC0018B40000CC0118B400000800
10.1.23.3/32-24         00:00:09  FastEthernet0/1 10.1.23.3
                   14  CC0218B40000CC0118B400010800
```



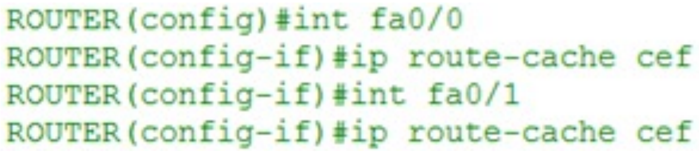
Cisco Express Forwarding



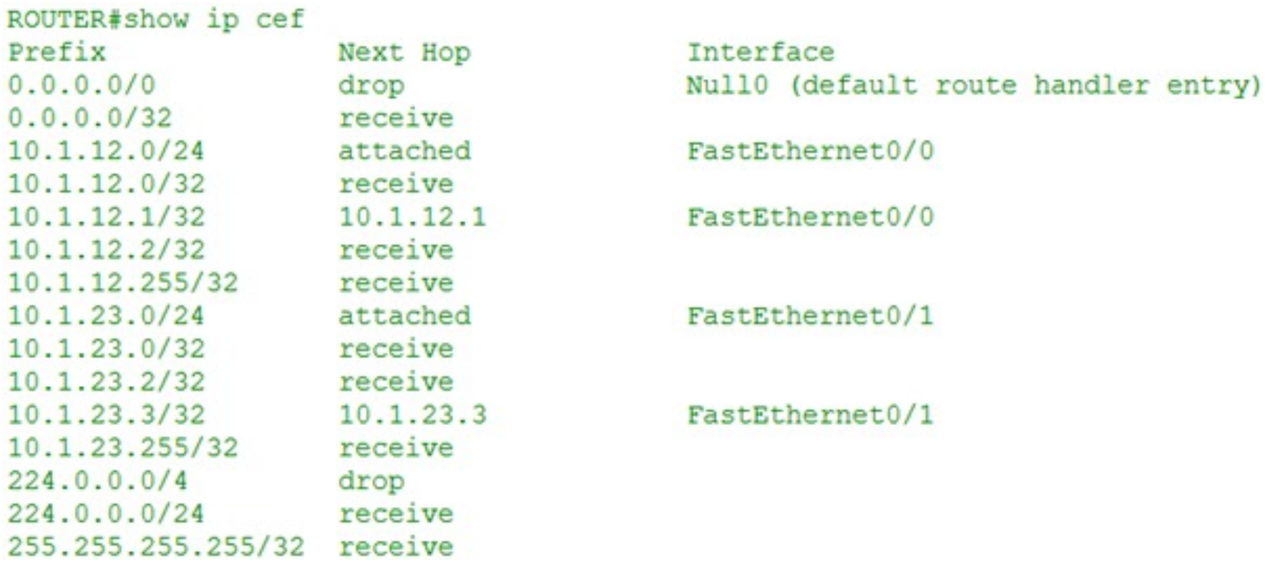
Cisco Express Forwarding

- This switching method is the fastest switching mode and is less CPU-intensive than fast switching and process switching.
- The control plane CPU of a CEF-enabled router creates two hardware-based tables called the Forwarding Information Base (FIB) table and an adjacency table using Layer 3 and 2 tables including the routing and Address Resolution Protocol (ARP) tables.
- When a network has converged, the FIB and adjacency tables contain all the information a router would have to consider when forwarding a packet.
- These two tables are then used to make hardware-based forwarding decisions for all frames in a data flow, even the first frame.
- The FIB contains precomputed reverse lookups and next-hop information for routes, including the interface and Layer 2 information.

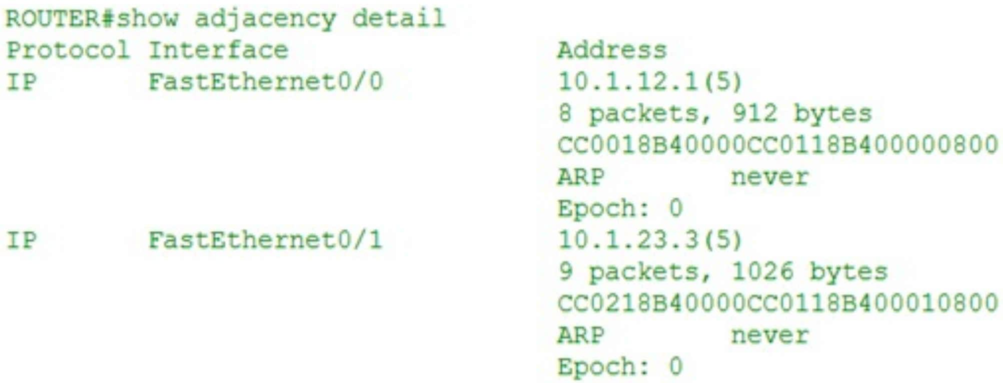
Cisco Express Forwarding



```
ROUTER(config)#int fa0/0
ROUTER(config-if)#ip route-cache cef
ROUTER(config-if)#int fa0/1
ROUTER(config-if)#ip route-cache cef
```

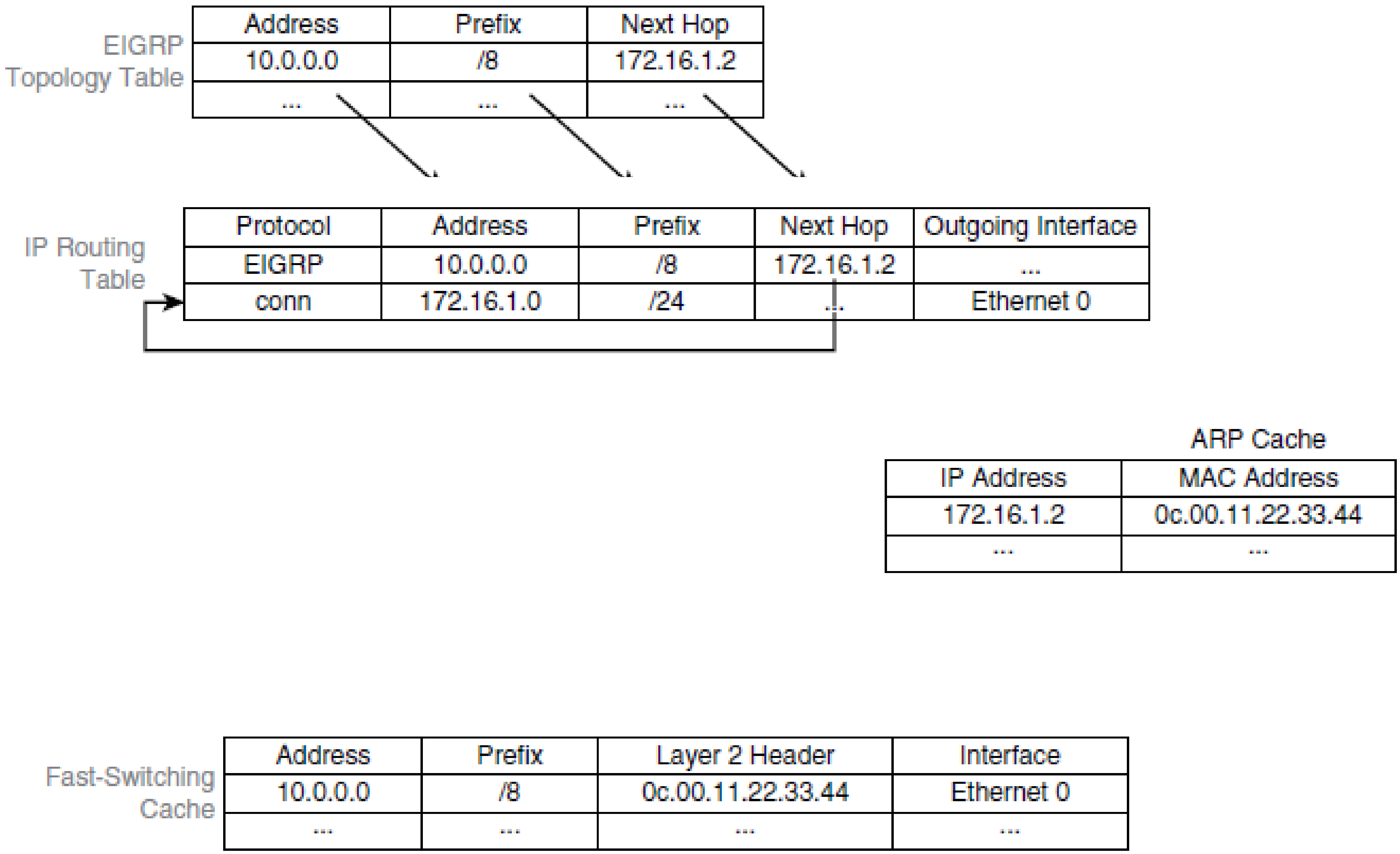


```
ROUTER#show ip cef
Prefix              Next Hop             Interface
0.0.0.0/0           drop                 Null0 (default route handler entry)
0.0.0.0/32          receive
10.1.12.0/24        attached             FastEthernet0/0
10.1.12.0/32        receive
10.1.12.1/32        10.1.12.1            FastEthernet0/0
10.1.12.2/32        receive
10.1.12.255/32      receive
10.1.23.0/24        attached             FastEthernet0/1
10.1.23.0/32        receive
10.1.23.2/32        receive
10.1.23.3/32        10.1.23.3            FastEthernet0/1
10.1.23.255/32      receive
224.0.0.0/4         drop
224.0.0.0/24        receive
255.255.255.255/32  receive
```



```
ROUTER#show adjacency detail
Protocol Interface                 Address
IP       FastEthernet0/0           10.1.12.1(5)
                                   8 packets, 912 bytes
                                   CC0018B40000CC0118B400000800
                                   ARP        never
                                   Epoch: 0
IP       FastEthernet0/1           10.1.23.3(5)
                                   9 packets, 1026 bytes
                                   CC0218B40000CC0118B400010800
                                   ARP        never
                                   Epoch: 0
```

Process and Fast Switching diff



EIGRP Topology Table

Address	Prefix	Next Hop
10.0.0.0	/8	172.16.1.2
...	...	...

IP Routing Table

Protocol	Address	Prefix	Next Hop	Outgoing Interface
EIGRP	10.0.0.0	/8	172.16.1.2	...
conn	172.16.1.0	/24	...	Ethernet 0

ARP Cache

IP Address	MAC Address
172.16.1.2	0c.00.11.22.33.44
...	...

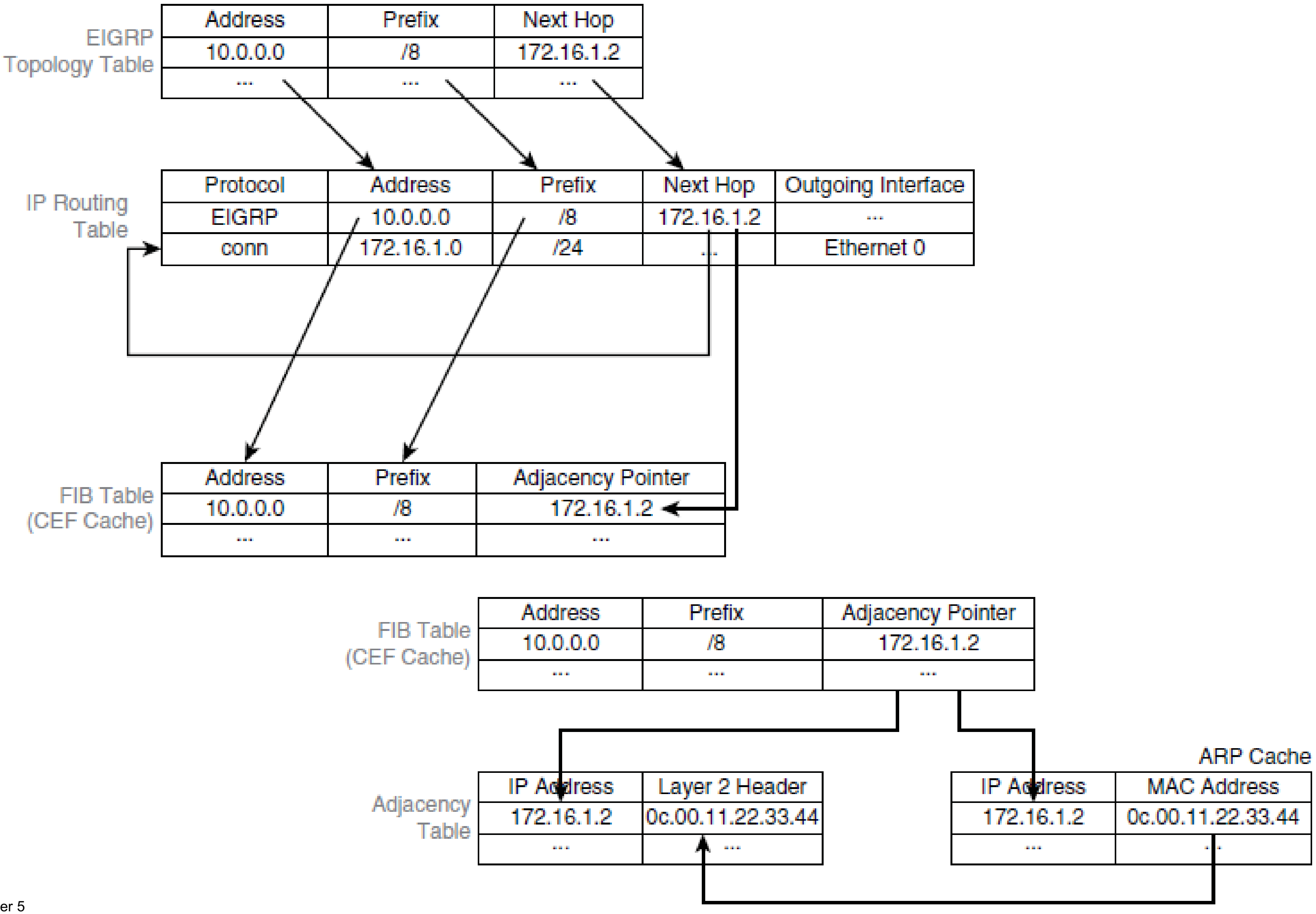
Fast-Switching Cache

Address	Prefix	Layer 2 Header	Interface
10.0.0.0	/8	0c.00.11.22.33.44	Ethernet 0
...	...	...	...

Process and Fast Switching diff

- Specifically, an entry is created in the fast-switching cache to ensure that the subsequent packets for the same destination prefix will be fast switched.
- All subsequent packets for the same destination are fast switched:
    - The switching occurs in the interrupt code. (The packet is processed immediately.)
    - Fast destination lookup is performed (no recursion).
    - The encapsulation uses a pregenerated Layer 2 header that contains the destination IP Address and Layer 2 source MAC address. (No ARP request or ARP cache lookup is necessary.)

Cisco Express Forwarding



EIGRP Topology Table

Address	Prefix	Next Hop
10.0.0.0	/8	172.16.1.2
...	...	...

IP Routing Table

Protocol	Address	Prefix	Next Hop	Outgoing Interface
EIGRP	10.0.0.0	/8	172.16.1.2	...
conn	172.16.1.0	/24	...	Ethernet 0

FIB Table (CEF Cache)

Address	Prefix	Adjacency Pointer
10.0.0.0	/8	172.16.1.2
...	...	...

FIB Table (CEF Cache)

Address	Prefix	Adjacency Pointer
10.0.0.0	/8	172.16.1.2
...	...	...

Adjacency Table

IP Address	Layer 2 Header
172.16.1.2	0c.00.11.22.33.44
...	...

ARP Cache

IP Address	MAC Address
172.16.1.2	0c.00.11.22.33.44
...	...

Cisco Express Forwarding

- CEF separates the control plane software from the data plane hardware, thereby achieving higher data throughput.
- The control plane is responsible for building the FIB table and adjacency tables in software.
- The data plane is responsible for forwarding IP unicast traffic using hardware.

CEF FIB Table

- The FIB is derived from the IP routing table and is arranged for maximum lookup throughput.
- CEF IP destination prefixes are stored from the most-specific to the least specific entry.
- The FIB lookup is based on the Layer 3 destination address prefix (longest match), so it matches the structure of CEF entries. When the CEF FIB table is full, a wildcard entry redirects frames to the Layer 3 engine.
- The FIB table is updated after each network change, but only once, and contains all known routes; there is no need to build a route cache by central-processing initial packets from each data flow.
- Each change in the IP routing table triggers a similar change in the FIB table because it contains all next-hop addresses that are associated with all destination networks.

CEF Adjacency Table

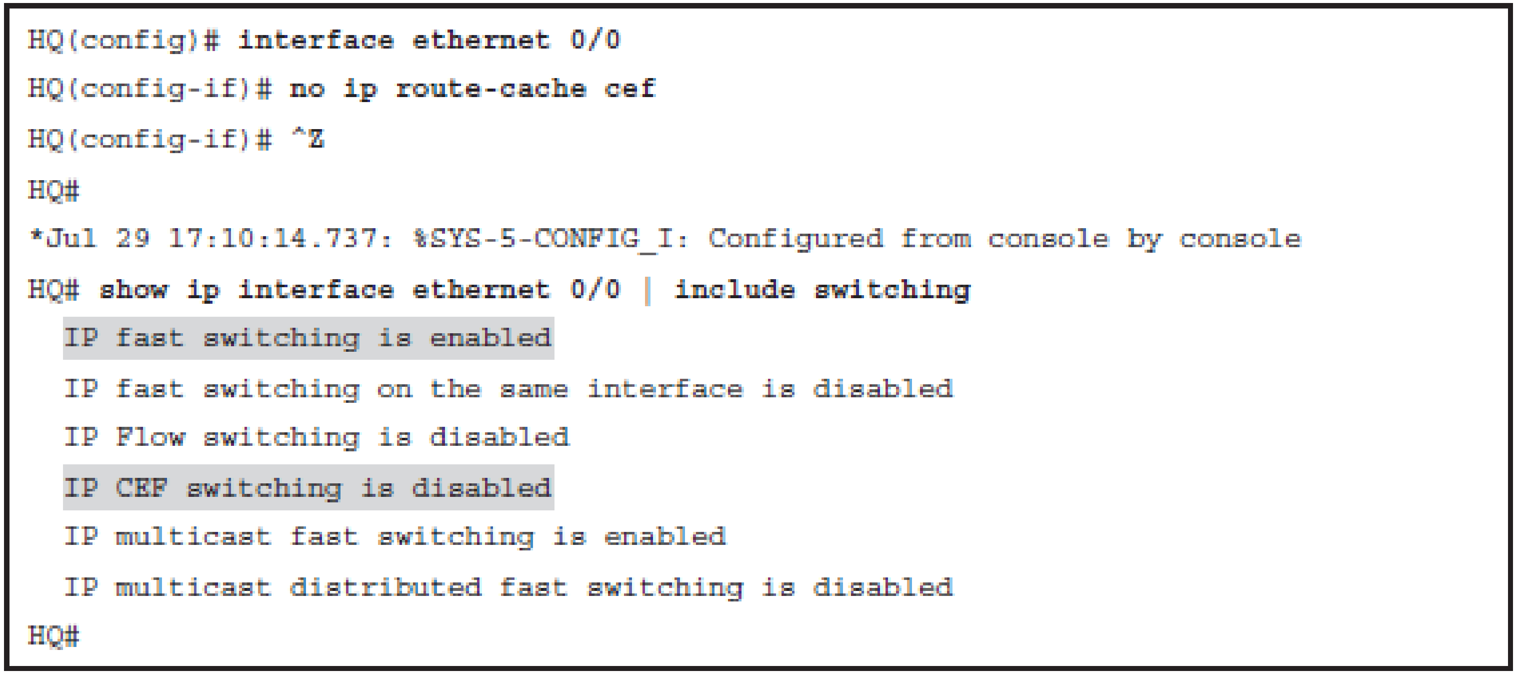
- CEF also caches Layer 2 next-hop addresses and frame header rewrite information for all FIB entries in the adjacency table.
- The adjacency table is derived from the ARP table, and it contains Layer 2 header rewrite (MAC) information for each next hop that is contained in the FIB.
- Each time that an adjacency entry is created (such as through ARP), a link-layer header for that adjacent node is precomputed and is stored in the adjacency table.
- CEF uses a specific process to build forwarding tables in the hardware and then uses the information from those tables to forward packets at line speed.

CEF Exceptions

- Not all packets can be CEF switched and processed in the hardware. When traffic cannot be processed in the hardware, it must be received by software processing of the Layer 3 engine.

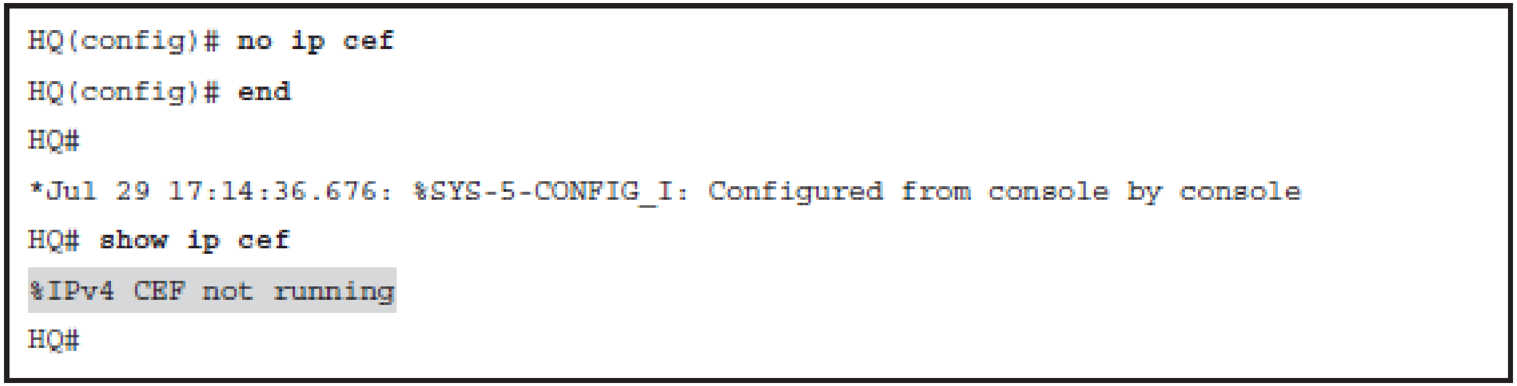
- Some examples of IP exception packets are packets that have the following characteristics:
- They use IP header options.
- They have an expiring IP Time To Live (TTL) counter.
- They are forwarded to a tunnel interface.
- They arrive with unsupported encapsulation types.
- They are routed to an interface with unsupported encapsulation types.
- They exceed the maximum transmission unit (MTU) of an output interface and must be fragmented.

Enable and Disable CEF by Interface



```
HQ(config)# interface ethernet 0/0
HQ(config-if)# no ip route-cache cef
HQ(config-if)# ^Z
HQ#
*Jul 29 17:10:14.737: %SYS-5-CONFIG_I: Configured from console by console
HQ# show ip interface ethernet 0/0 | include switching
  IP fast switching is enabled
  IP fast switching on the same interface is disabled
  IP Flow switching is disabled
  IP CEF switching is disabled
  IP multicast fast switching is enabled
  IP multicast distributed fast switching is disabled
HQ#
```

Enable and Disable CEF Globally



```
HQ(config)# no ip cef
HQ(config)# end
HQ#
*Jul 29 17:14:36.676: %SYS-5-CONFIG_I: Configured from console by console
HQ# show ip cef
%IPv4 CEF not running
HQ#
```

Understanding Path Control

Understanding Path Control

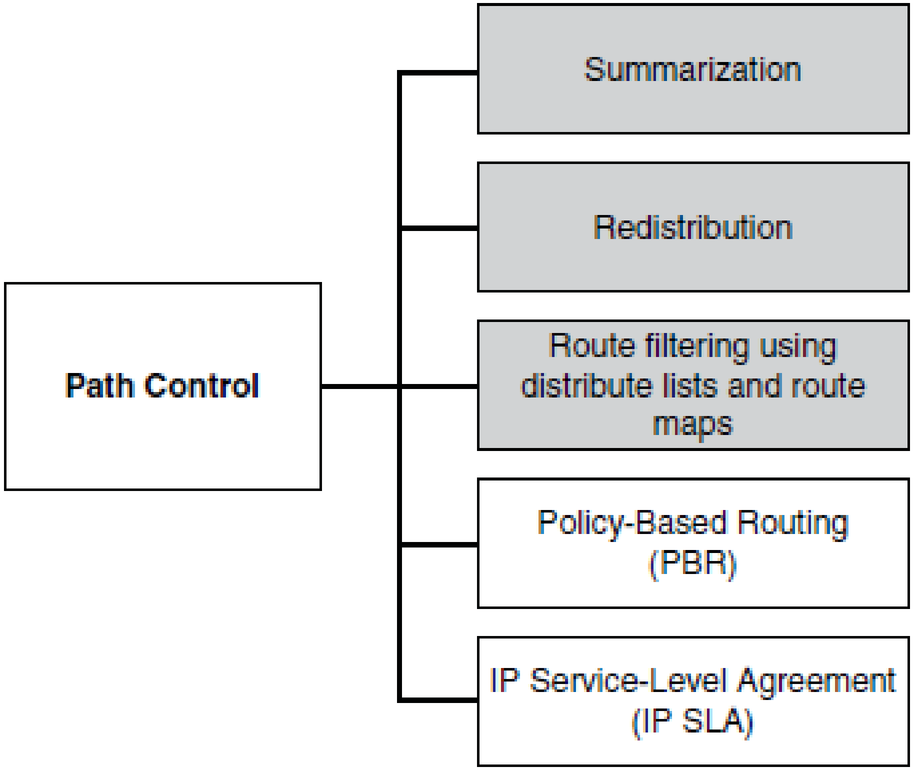
- Identify the need for path control
- Describe how to use policy-based routing (PBR) to control path selection
- Describe how to use IP service-level agreement (IP SLA) to control path selection

The Need for Path Control

- Path control tools can be used to change the default destination forwarding and optimize the path of the packets for some specific application.
- Other examples of path control include switching traffic to the backup link if there is a primary link failure or forwarding some traffic to the backup link if the primary link is congested.
- Path control mechanisms can improve performance in such a situation.
- Similarly, load balancing can divide traffic among parallel paths.
- It is important to provide predictable and deterministic control over traffic patterns.
- Unfortunately, there is not a “one-command” solution to implement path control.

The Need for Path Control

- You can use all these tools as part of an integrated strategy to implement path control.


Path Control
Summarization
Redistribution
Route filtering using distribute lists and route maps
Policy-Based Routing (PBR)
IP Service-Level Agreement (IP SLA)

Implementing Path Control Using Policy-Based Routing

- PBR enables the administrator to define a routing policy other than basic destination-based routing using the routing table.
- With PBR, route maps can be used to match source and destination addresses, protocol types, and end-user applications.
- When a match occurs, a **set** command can be used to define items, such as the interface or next-hop address to which the packet should be sent.

PBR Features

- **Source-based transit-provider selection**
  - PBR policies can be implemented by ISPs and other organizations to route traffic that originates from different sets of users through different Internet connections across the policy routers.
- **QoS**
  - PBR policies can be implemented to provide quality of service (QoS) to differentiated traffic by setting the type of service (ToS) values in the IP packet headers in routers at the periphery of the network and then leveraging queuing mechanisms to prioritize traffic in the network’s core or backbone.
- **Cost savings**
  - PBR policies can be implemented to direct the bulk traffic associated with a specific activity to use a higher-bandwidth, high-cost link for a short time and to continue basic connectivity over a lower-bandwidth, low-cost link for interactive traffic.
- **Load sharing**
  - PBR policies can be implemented based on the traffic characteristics to distribute traffic among multiple paths.

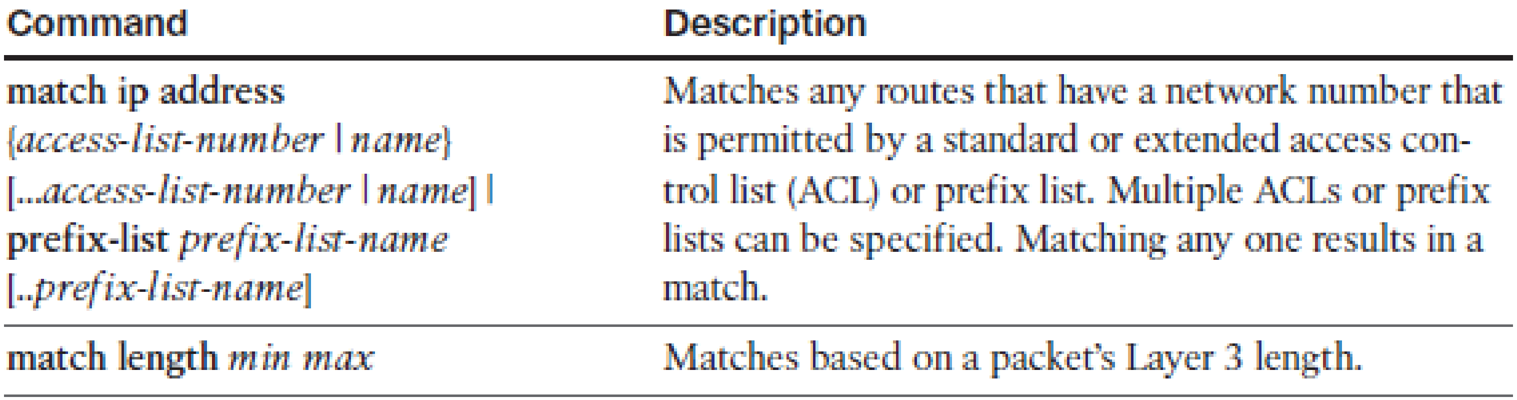
Steps for Configuring PBR

1. Enable PBR by configuring a route map using the **route-map** global configuration command.
2. Implement the traffic-matching configuration, specifying which traffic will be manipulated. This is done using the **match** commands within the route map.
3. Define the action for the matched traffic. This is done using the **set** commands within the route map.
4. Optionally, fast-switched PBR or CEF-switched PBR can be enabled.
5. Apply the route map to incoming traffic or to traffic locally generated on the router using the **ip policy route-map** interface configuration command.

Configuring PBR – Route-Map

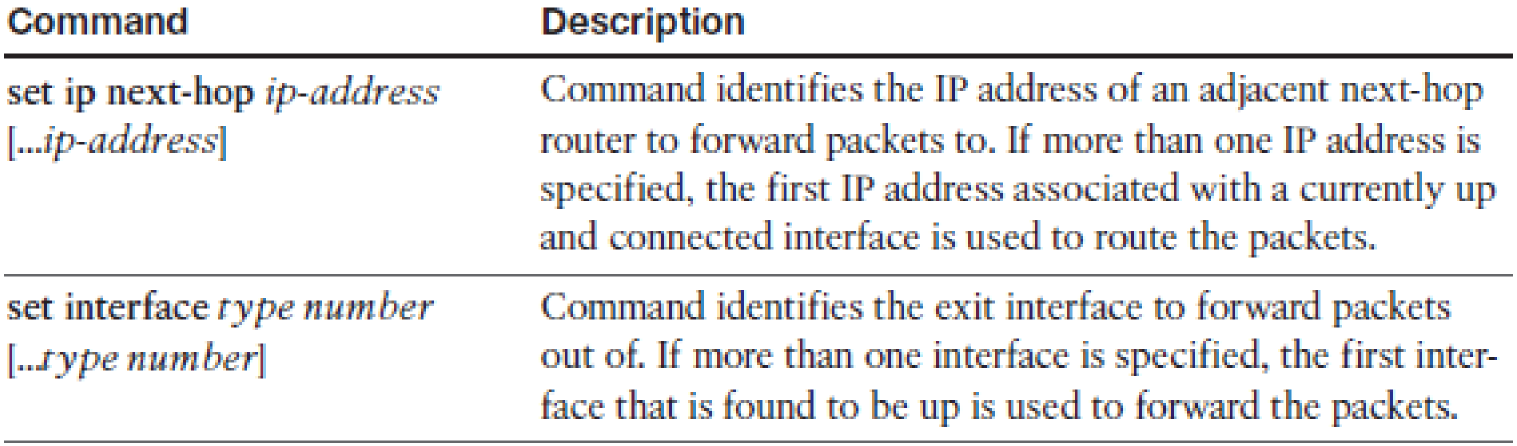
- If the statement is marked as **permit** , such as in **route-map MY-MAP permit 10** , packets that meet all the match criteria are policy-based routed.
- If the statement is marked as **deny** , such as in **route-map MY-MAP deny 10** , a packet meeting the match criteria is not policy-based routed. Instead, it is sent through the normal forwarding channels and destination-based routing is performed.
- If no match is found in the route map, the packet is *not* dropped. It is forwarded through the normal routing channel, which means that destination-based routing is performed.

PBR match Commands



Command	Description
match ip address {*access-list-number* | *name*} [...*access-list-number* | *name*] | **prefix-list** *prefix-list-name* [..*prefix-list-name*]	Matches any routes that have a network number that is permitted by a standard or extended access control list (ACL) or prefix list. Multiple ACLs or prefix lists can be specified. Matching any one results in a match.
match length *min max*	Matches based on a packet's Layer 3 length.

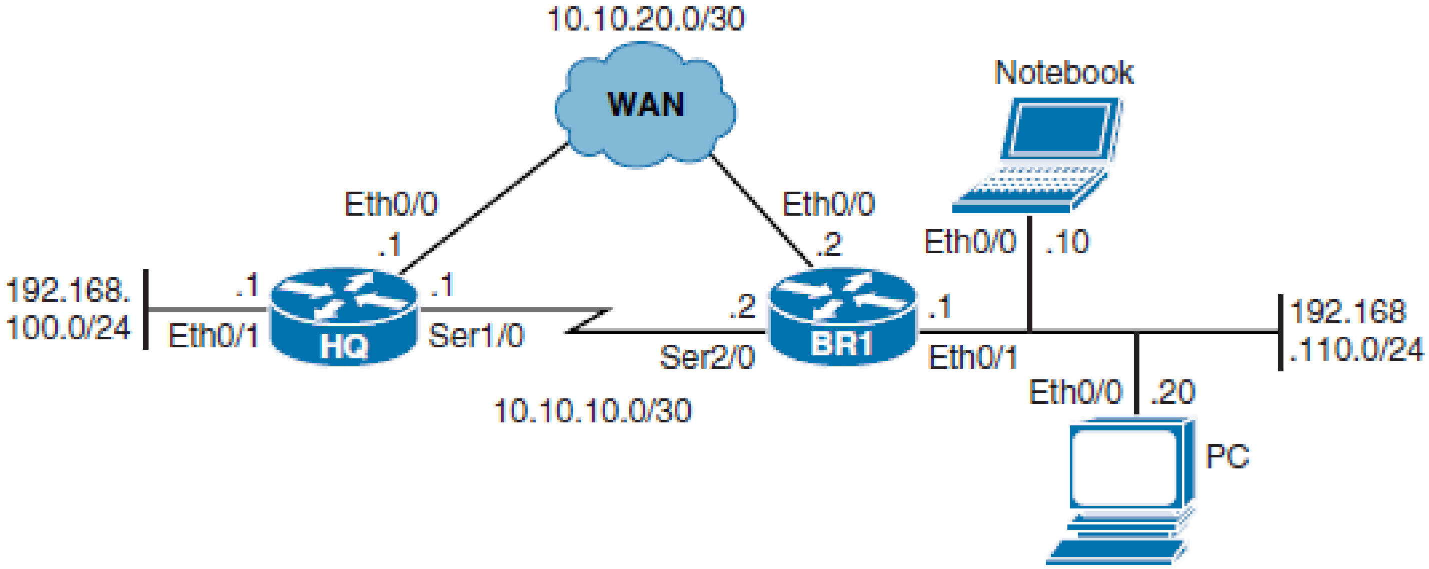
PBR set Commands



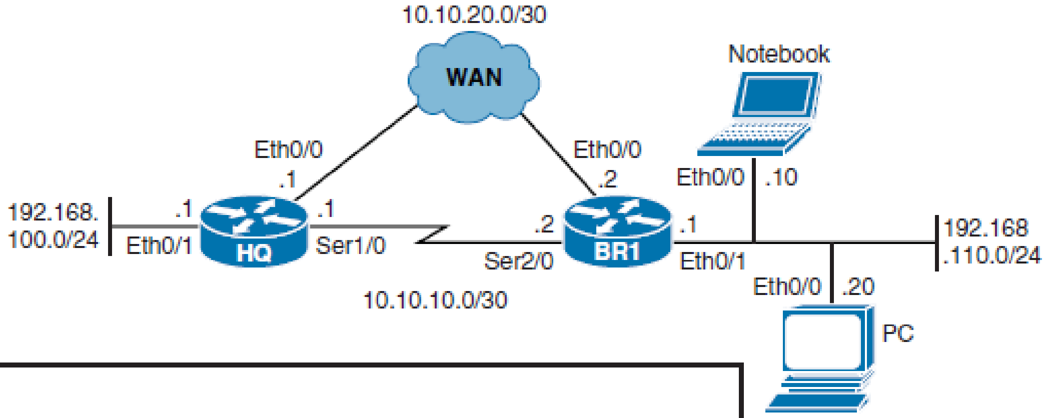
Command	Description
set ip next-hop *ip-address* [...*ip-address*]	Command identifies the IP address of an adjacent next-hop router to forward packets to. If more than one IP address is specified, the first IP address associated with a currently up and connected interface is used to route the packets.
set interface *type number* [...*type number*]	Command identifies the exit interface to forward packets out of. If more than one interface is specified, the first interface that is found to be up is used to forward the packets.

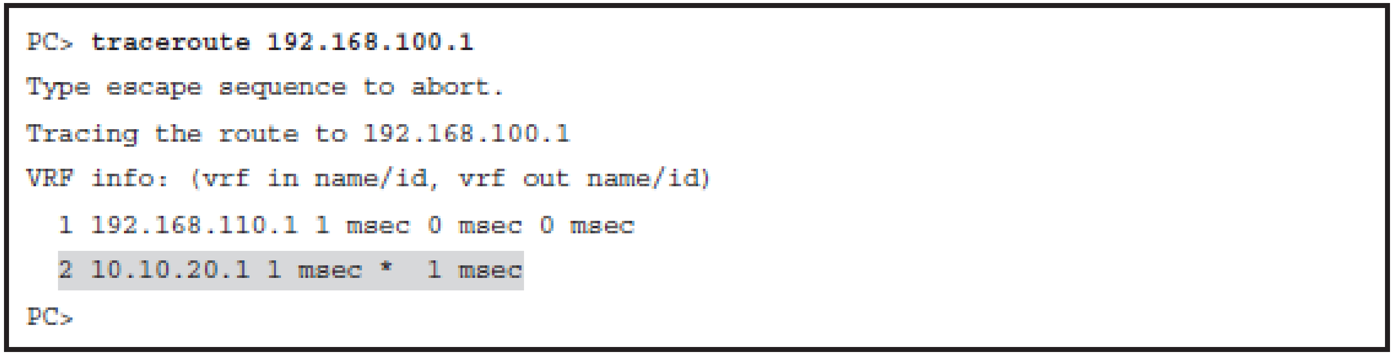
Configuring PBR Example

- Verify normal traffic paths as selected by the traditional destination-based routing
- Configure PBR to alter the traffic flow for one client station
- Verify both the PBR configuration and the new traffic path

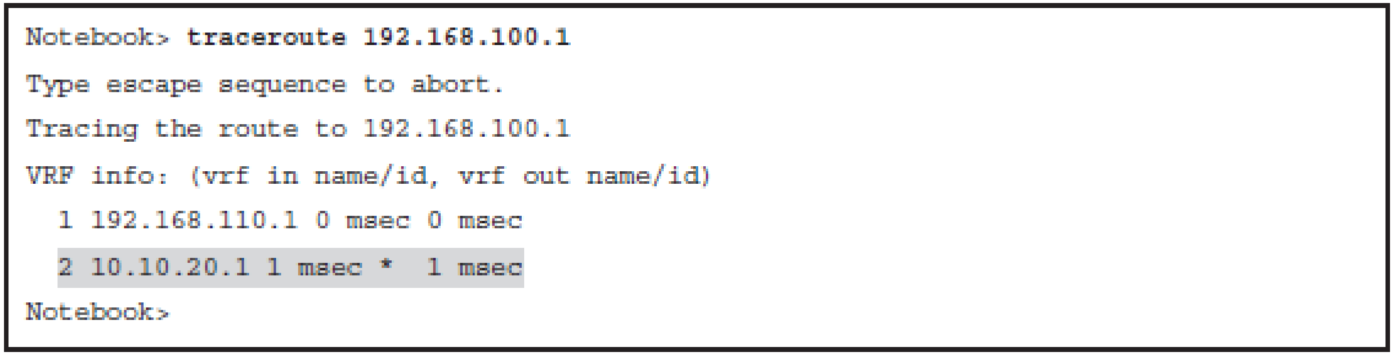


Verify Normal Traffic Paths





```
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 0 msec 0 msec
  2 10.10.20.1 1 msec *  1 msec
PC>
```



```
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 0 msec 0 msec
  2 10.10.20.1 1 msec *  1 msec
Notebook>
```

Configure PBR to Alter the Traffic Flow from the Notebook

1.

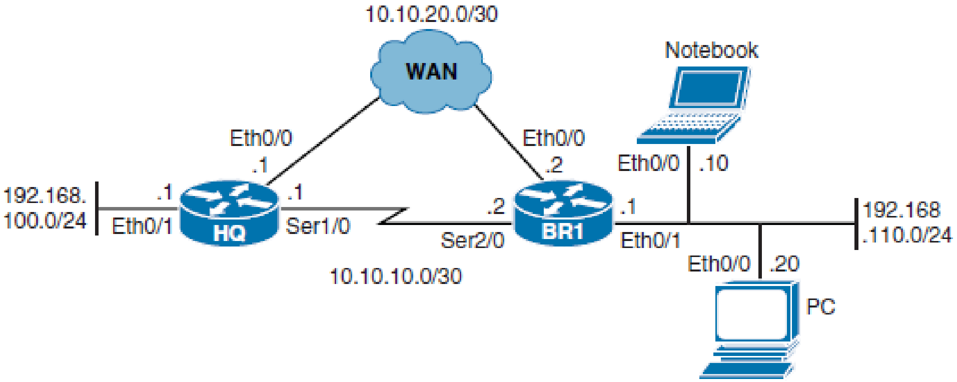
```
BR1(config)# ip access-list extended PBR-ACL
BR1(config-ext-nacl)# permit ip host 192.168.110.10 any
BR1(config-ext-nacl)# exit
```

2.

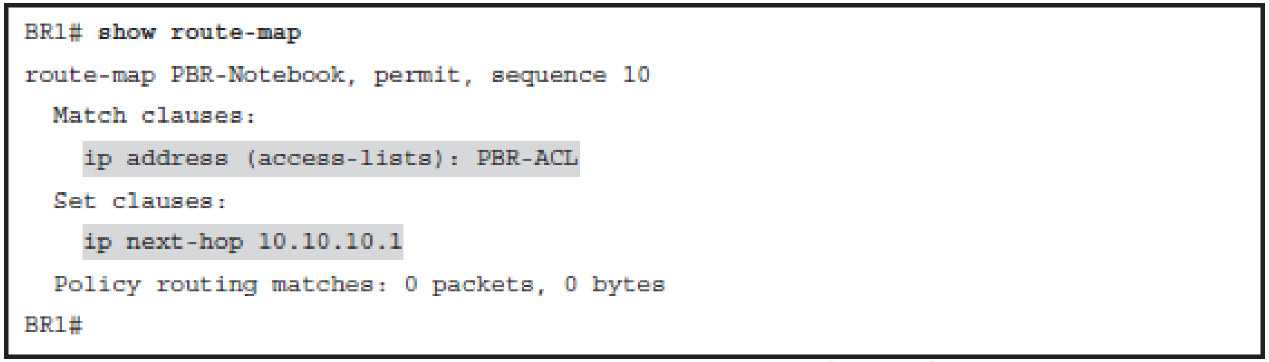
```
BR1(config)# route-map PBR-Notebook
BR1(config-route-map)# match ip address PBR-ACL
BR1(config-route-map)# set ip next-hop 10.10.10.1
BR1(config-route-map)# exit
```

3.

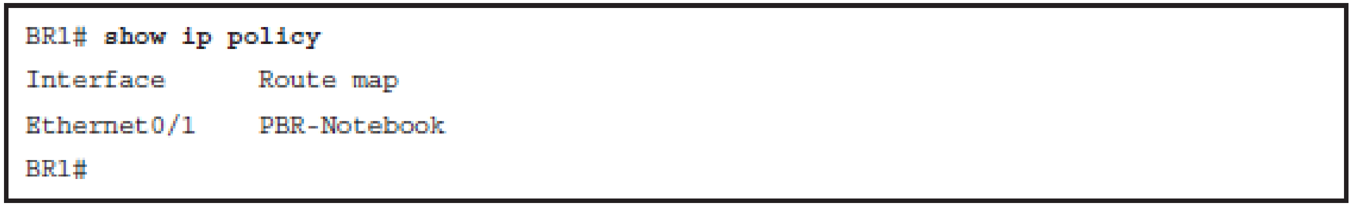
```
BR1(config)# interface ethernet 0/1
BR1(config-if)# ip policy route-map PBR-Notebook
BR1(config-if)# exit
BR1(config)# exit
```



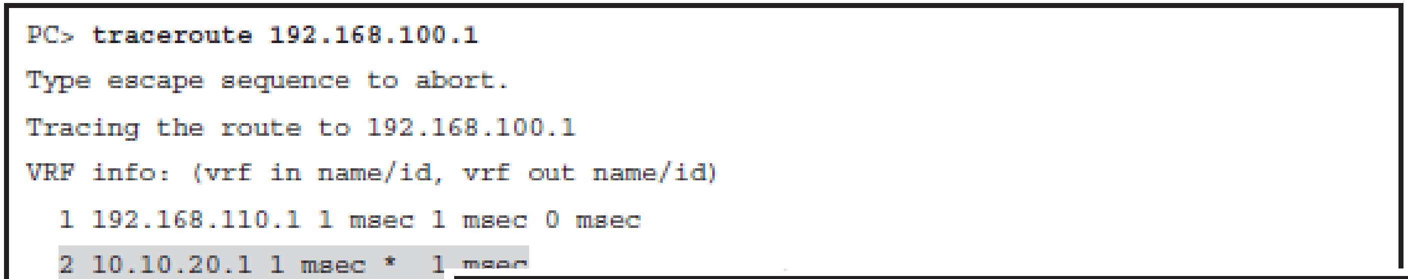
Verify the PBR Configuration and Traffic Path



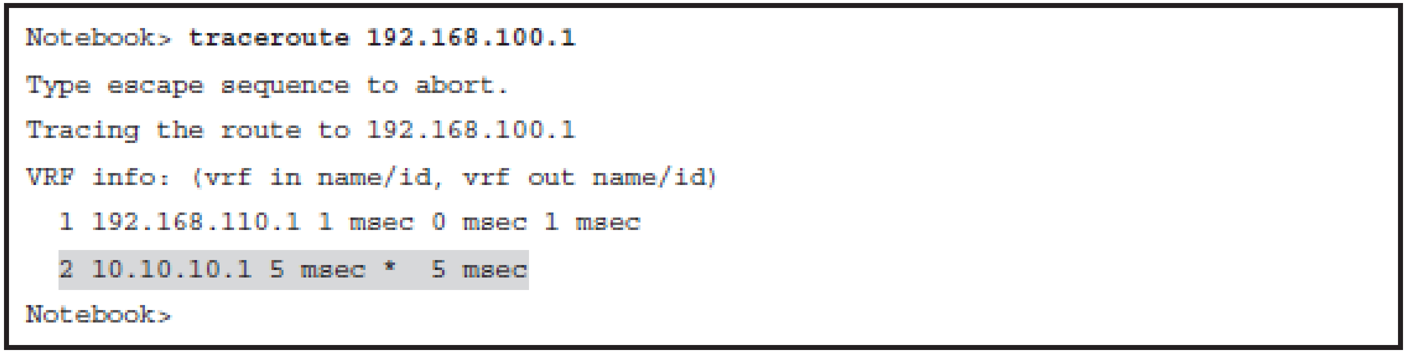
```
BR1# show route-map
route-map PBR-Notebook, permit, sequence 10
  Match clauses:
    ip address (access-lists): PBR-ACL
  Set clauses:
    ip next-hop 10.10.10.1
  Policy routing matches: 0 packets, 0 bytes
BR1#
```



```
BR1# show ip policy
Interface        Route map
Ethernet0/1      PBR-Notebook
BR1#
```



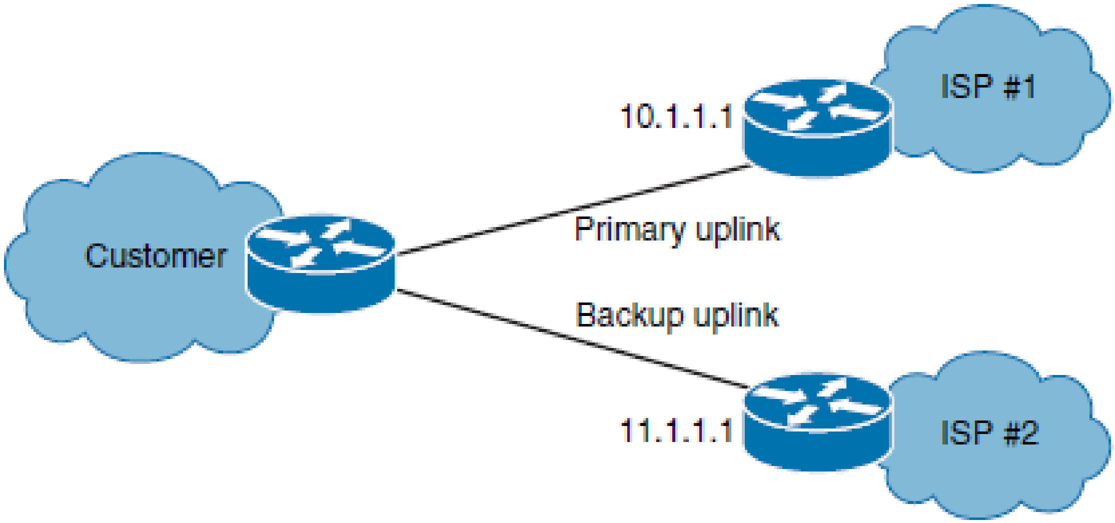
```
PC> traceroute 192.168.100.1
Type escape sequence to abort.
Tracing the route to 192.168.100.1
VRF info: (vrf in name/id, vrf out name/id)
  1 192.168.110.1 1 msec 1 msec 0 msec
  2 10.10.20.1 1 msec *  1 msec
PC>
```



```
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 10.10.10.1 5 msec *  5 msec
Notebook>
```

Implementing Path Control Using Cisco IOS IP SLAs

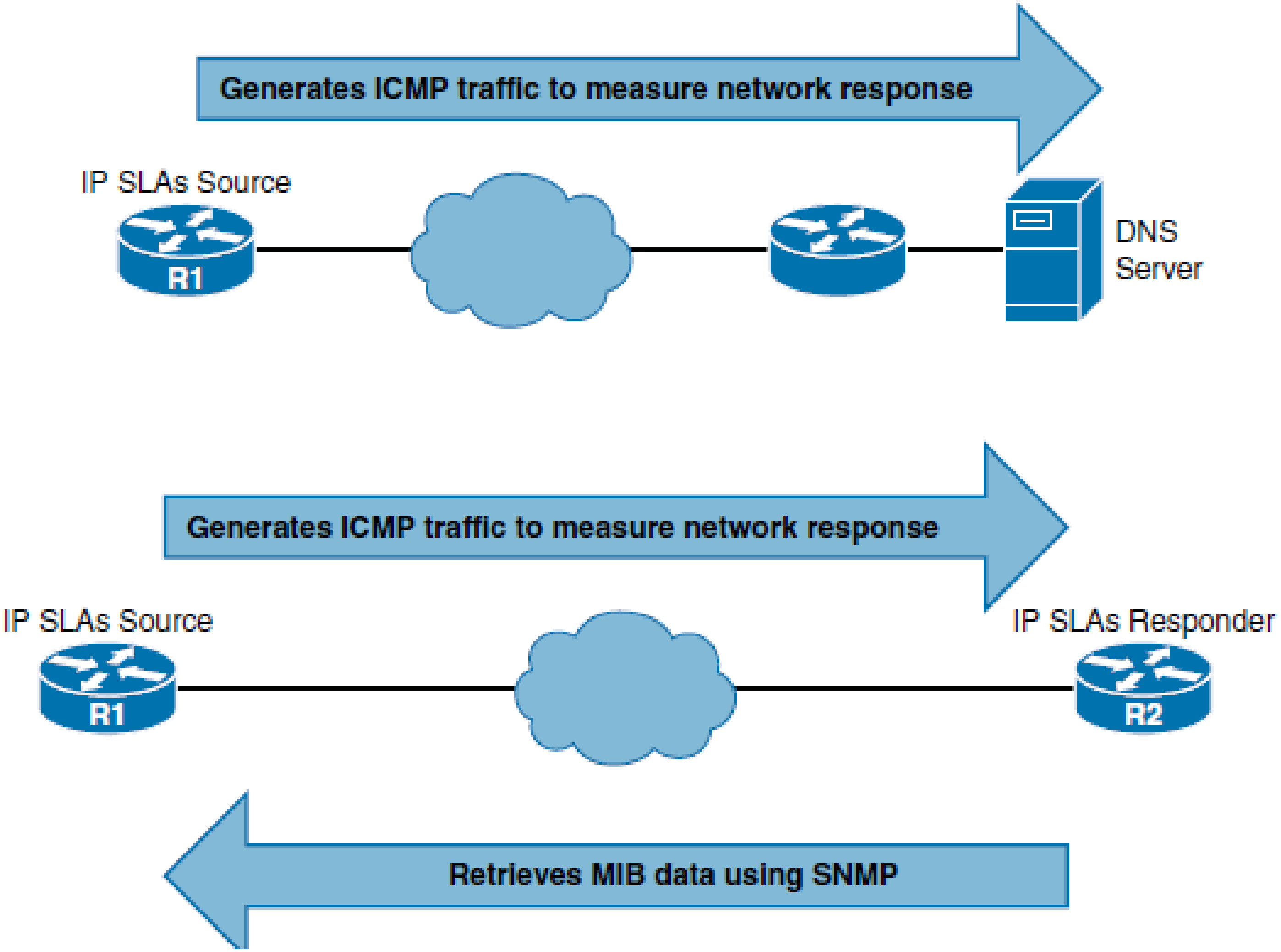
- PBR is a static path control mechanism. It cannot respond dynamically to changes in network health.



IP SLA Features

- Cisco IOS IP SLAs perform network performance measurement within Cisco devices.
- The IP SLAs use active traffic monitoring (generation of traffic in a continuous, reliable, and predictable manner) for measuring network performance.
- Cisco IOS IP SLAs actively send simulated data across the network to measure performance between multiple network locations or across multiple network paths.
- The information collected includes data about response time, one-way latency, jitter, packet loss, voice-quality scoring, network resource availability, application performance, and server response time.
- In its simplest form, Cisco IOS IP SLAs verify whether a network element, such as an IP address on a router interface or an open TCP port on an IP host, is active and responsive.

Cisco IOS IP SLA Sources and Targets

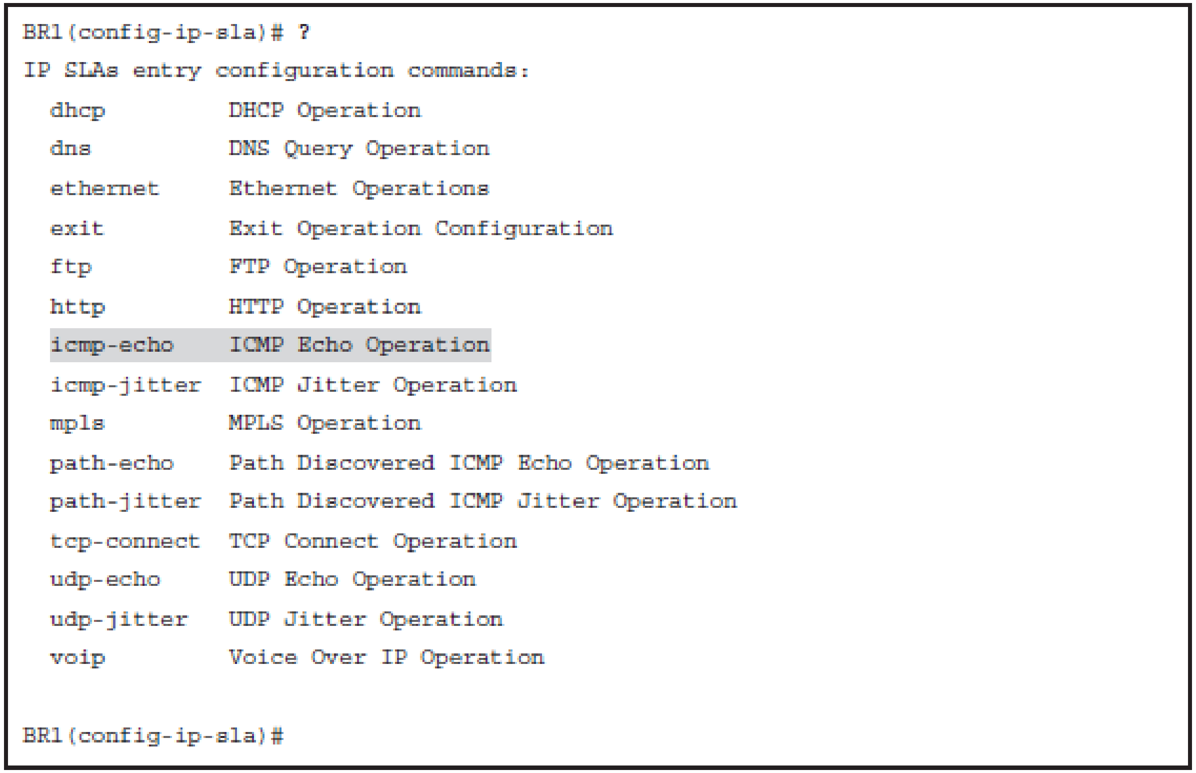

Generates ICMP traffic to measure network response
IP SLAs Source
R1
DNS
Server
Generates ICMP traffic to measure network response
IP SLAs Source
IP SLAs Responder
R1
R2
Retrieves MIB data using SNMP

Steps for Configuring IP SLAs

- **Step 1.** Define one or more IP SLA operations (or probes).
- **Step 2.** Define one or more tracking objects to track the state of IOS IP SLA operations.
- **Step 3.** Define the action associated with the tracking object.

Step 1. Configuring Cisco IOS IP SLA Operations

- Use the **`ip sla`** *`operation-number`* global configuration command to begin configuring a Cisco IOS IP SLA operation and to enter IP SLA configuration mode. The *operation-number* is the identification number of the IP SLA operation to be configured.

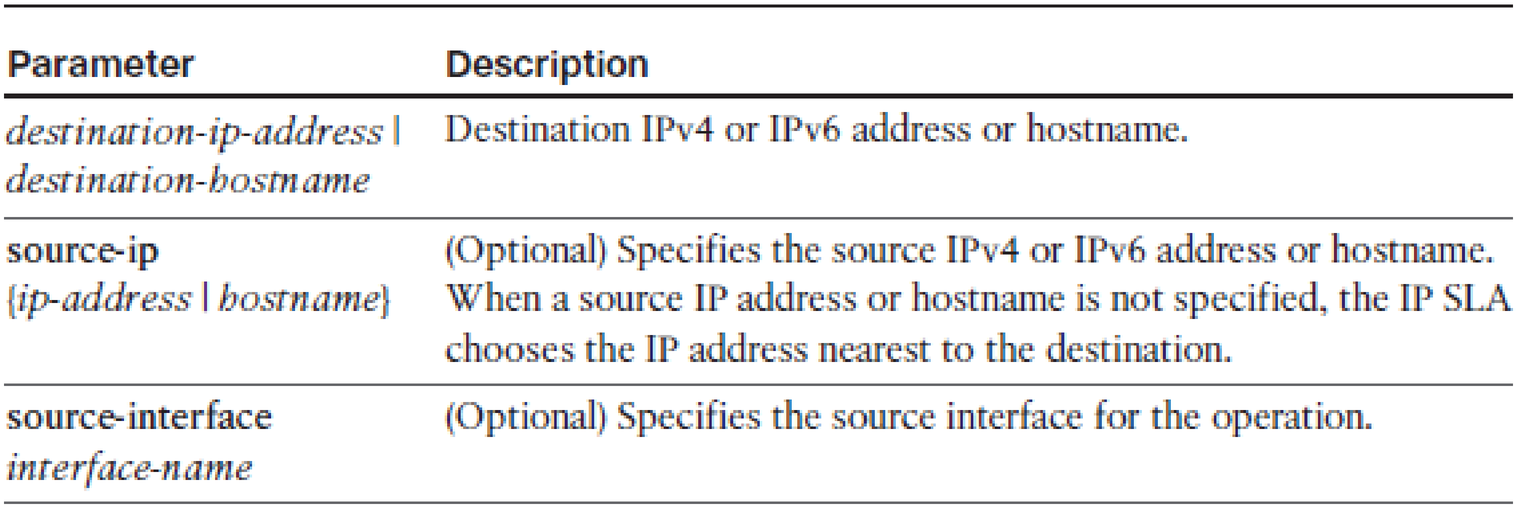


```
BR1(config-ip-sla)# ?
IP SLAs entry configuration commands:
  dhcp          DHCP Operation
  dns           DNS Query Operation
  ethernet      Ethernet Operations
  exit          Exit Operation Configuration
  ftp           FTP Operation
  http          HTTP Operation
  icmp-echo     ICMP Echo Operation
  icmp-jitter   ICMP Jitter Operation
  mpls          MPLS Operation
  path-echo     Path Discovered ICMP Echo Operation
  path-jitter   Path Discovered ICMP Jitter Operation
  tcp-connect   TCP Connect Operation
  udp-echo      UDP Echo Operation
  udp-jitter    UDP Jitter Operation
  voip          Voice Over IP Operation

BR1(config-ip-sla)#
```

IP SLA icmp-echo

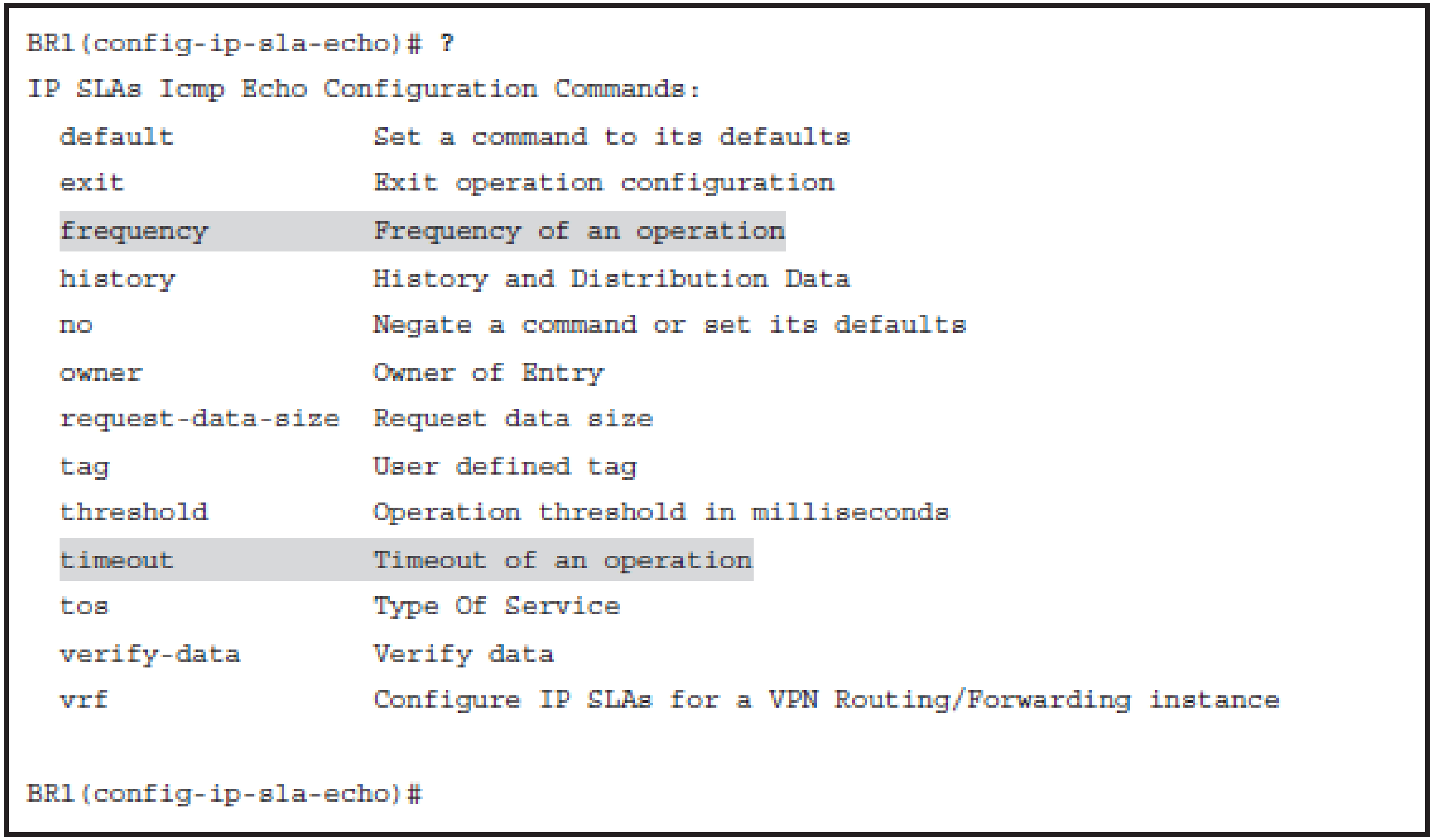
- The complete command syntax is **icmp-echo** { *destination-ip-address* | *destination hostname*} [**source-ip** { *ip -address* | *hostname* } | **source-interface** *interface-name*].

Table 5-3 icmp-echo *Command Parameters*



Parameter	Description
destination-ip-address | *destination-hostname*	Destination IPv4 or IPv6 address or hostname.
source-ip {*ip-address* | *hostname*}	(Optional) Specifies the source IPv4 or IPv6 address or hostname. When a source IP address or hostname is not specified, the IP SLA chooses the IP address nearest to the destination.
source-interface *interface-name*	(Optional) Specifies the source interface for the operation.

IP SLA ICMP Echo Configuration Mode Commands



```
BR1(config-ip-sla-echo)# ?
IP SLAs Icmp Echo Configuration Commands:
  default            Set a command to its defaults
  exit               Exit operation configuration
  frequency          Frequency of an operation
  history            History and Distribution Data
  no                 Negate a command or set its defaults
  owner              Owner of Entry
  request-data-size  Request data size
  tag                User defined tag
  threshold          Operation threshold in milliseconds
  timeout            Timeout of an operation
  tos                Type Of Service
  verify-data        Verify data
  vrf                Configure IP SLAs for a VPN Routing/Forwarding instance

BR1(config-ip-sla-echo)#
```

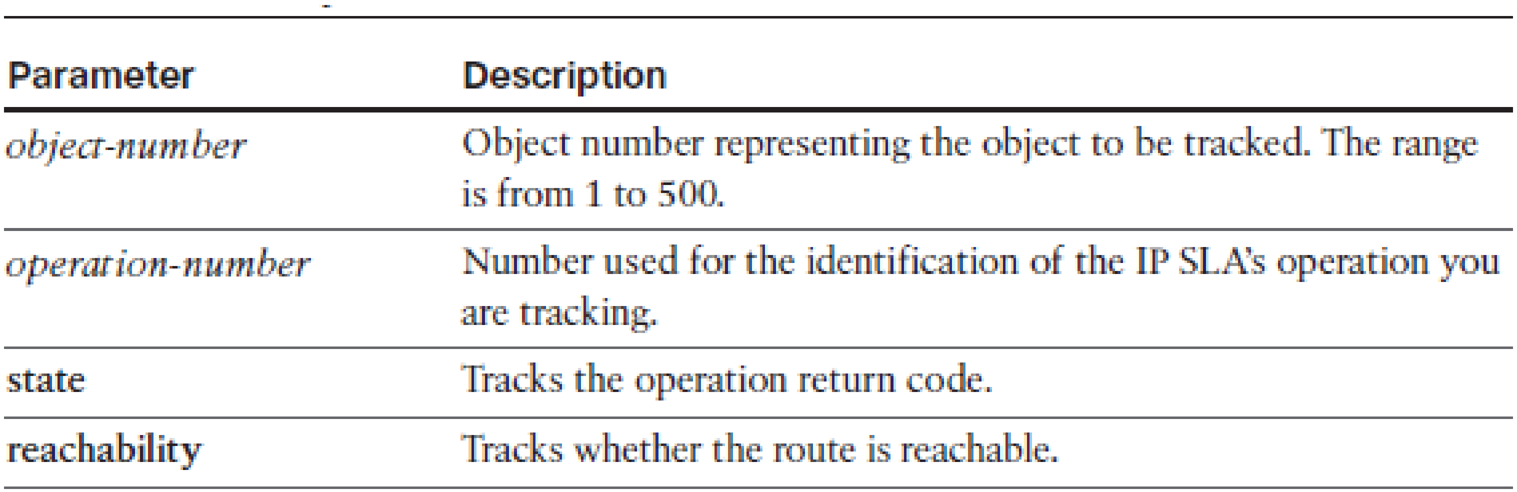
Schedule the IP SLA Operation

- Once a Cisco IP SLA operation is configured, it needs to be scheduled using the **ip sla schedule** global configuration command.

```
ip sla schedule operation-number [ life { forever | seconds
}] [ start-time { hh:mm [ :ss ] [ month day | day month ] |
pending | now | after hh:mm:ss }] [ ageout seconds ] [
recurring ]
```

Step 2: Configuring Cisco IOS IP SLA Tracking Objects

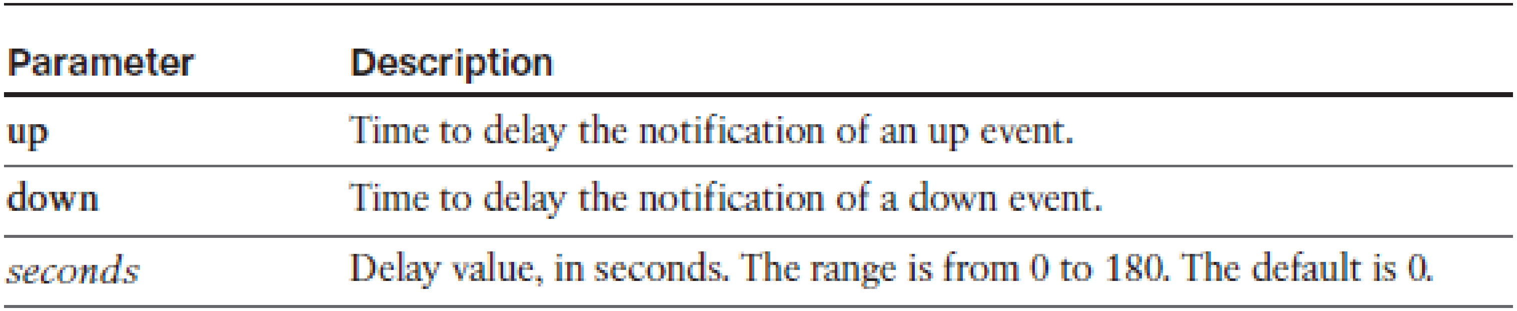
- Use the **`track`** *`object-number`* **`ip sla`** *`operation-number`* `{` **`state`** `|` **`reachability`** `}` global configuration command to track the state of an IOS IP SLA operation, and enter track configuration mode.



Parameter	Description
object-number	Object number representing the object to be tracked. The range is from 1 to 500.
operation-number	Number used for the identification of the IP SLA's operation you are tracking.
state	Tracks the operation return code.
reachability	Tracks whether the route is reachable.

delay Command Parameters

- Once in IP SLA track configuration mode, use the optional **delay** { **up** *seconds* [**down** *seconds*] | [**up** *seconds*] **down** *seconds* } track configuration command to specify a period of time to delay communicating state changes of a tracked object.



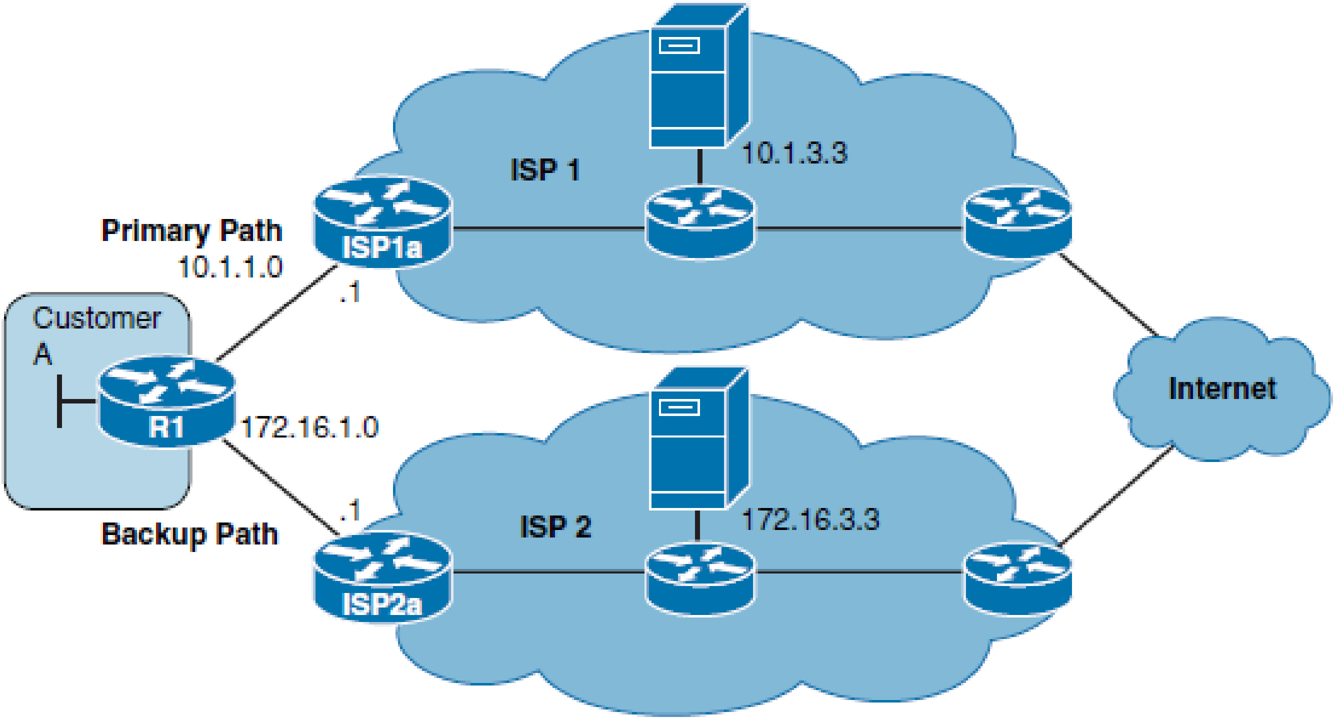
Parameter	Description
up	Time to delay the notification of an up event.
down	Time to delay the notification of a down event.
seconds	Delay value, in seconds. The range is from 0 to 180. The default is 0.

Step 3: Defining an Action Associated with a Tracking Object

- Many types of actions can be associated with a tracked object.
- A simple path control action is to use the **`ip route`** *`prefix mask`* `{` *`ip-address`* `|` *`interface-type interface-number`* `[` *`ip-address`* `]}` `[` **`track`** *`number`* `]` global configuration command.
- The command can be used with the **track** keyword to establish a static route that tracks an object.

Configuring IP SLA Example

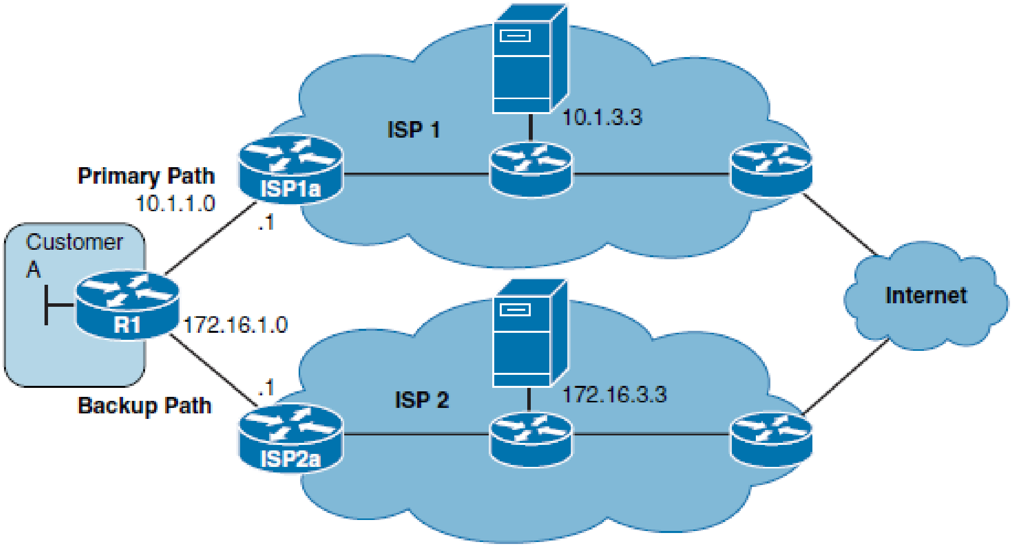
- The static route to ISP1a (ISP-1), which has been assigned an administrative distance of 2
- The static route to ISP2a (ISP-2), which has been assigned an administrative distance of 3



In the example, you will

- Configure an IP SLA operation with the ISP 1 DNS server
- Define a tracking object assign an action
- Configure an IP SLA operation with the ISP 2 DNS server
- Define a tracking object assign an action

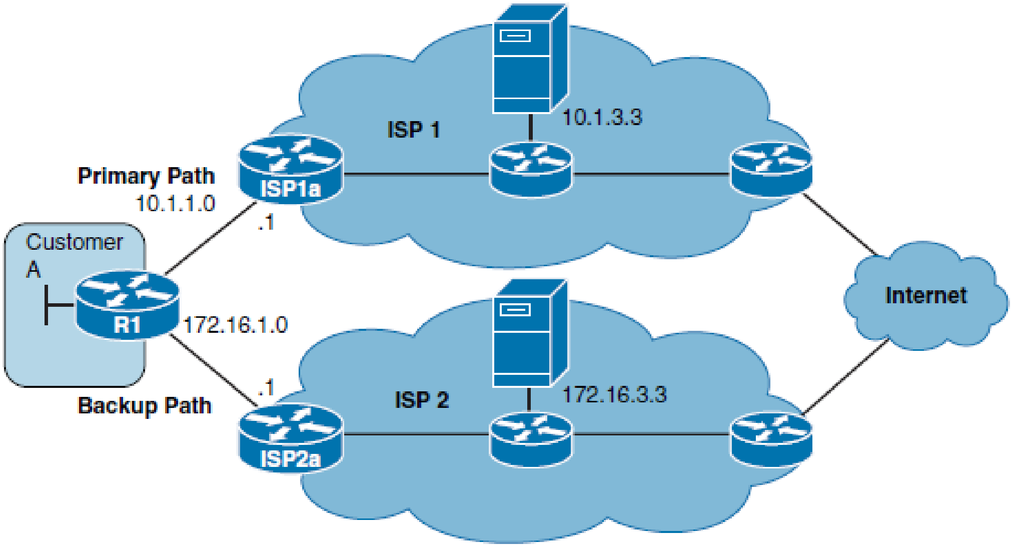
Configure IP SLA and Track Object for ISP 1

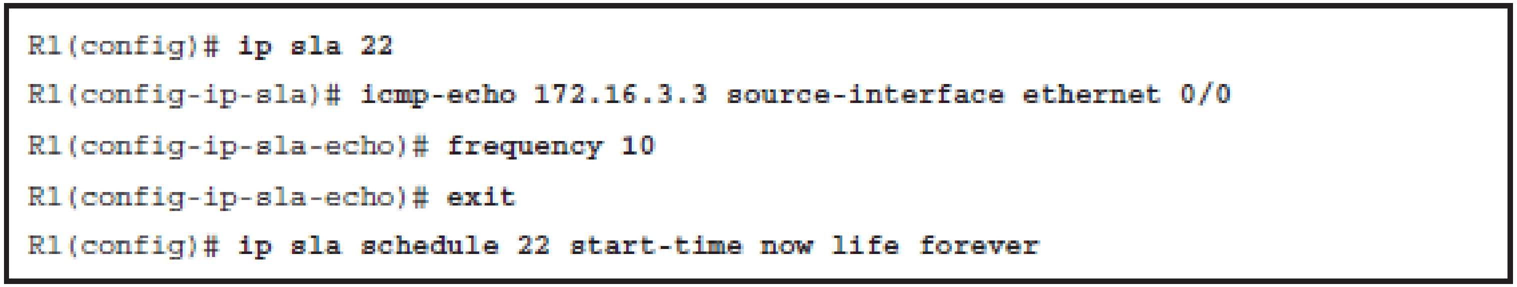


```
R1(config)# ip sla 11
R1(config-ip-sla)# icmp-echo 10.1.3.3 source-interface ethernet 0/0
R1(config-ip-sla-echo)# frequency 10
R1(config-ip-sla-echo)# exit
R1(config)# ip sla schedule 11 start-time now life forever
```

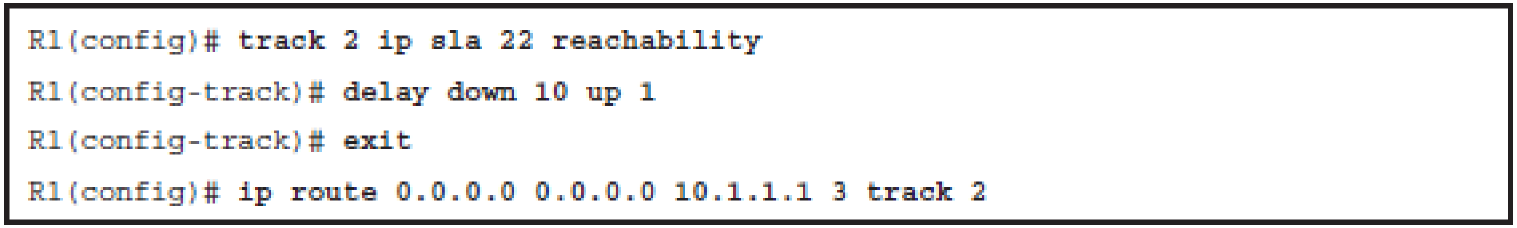
```
R1(config)# track 1 ip sla 11 reachability
R1(config-track)# delay down 10 up 1
R1(config-track)# exit
R1(config)# ip route 0.0.0.0 0.0.0.0 10.1.1.1 2 track 1
```

Configure IP SLA and Track Object for ISP 2





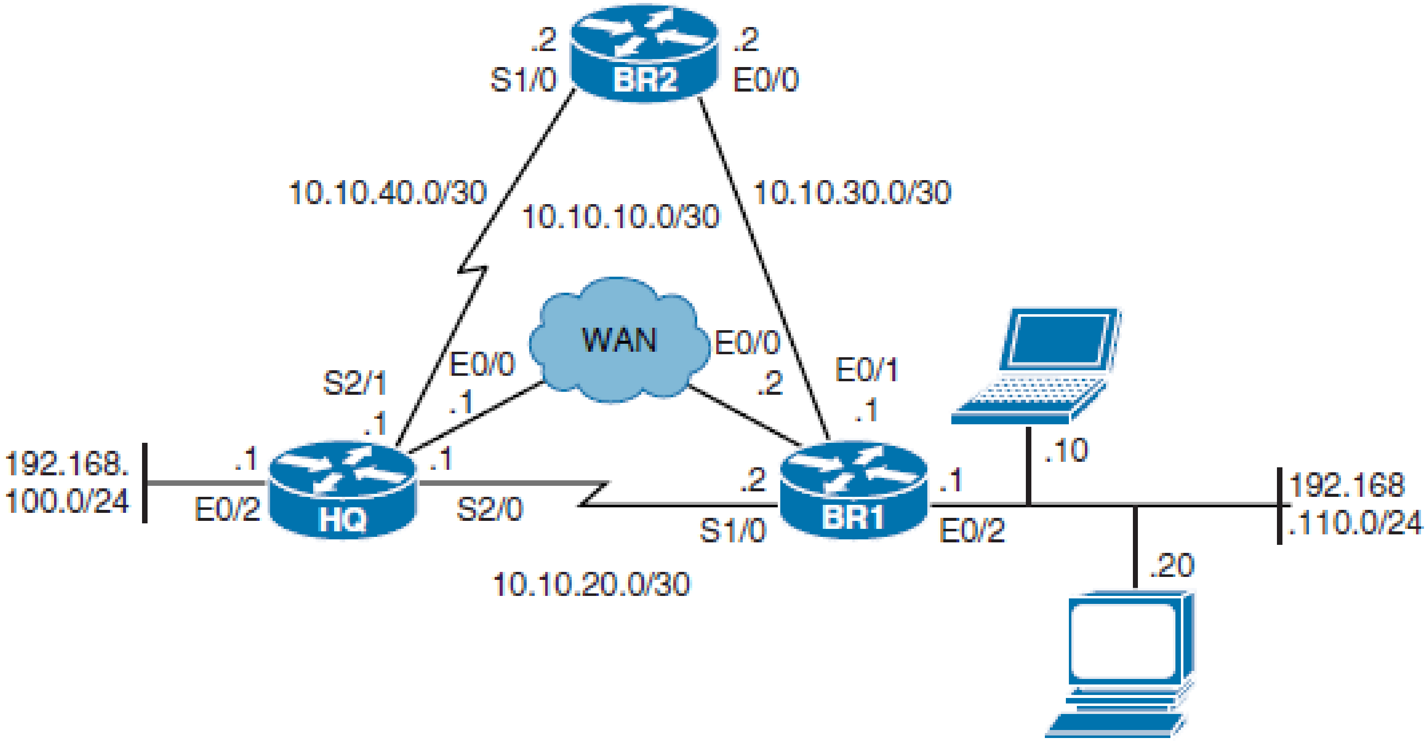
```
R1(config)# ip sla 22
R1(config-ip-sla)# icmp-echo 172.16.3.3 source-interface ethernet 0/0
R1(config-ip-sla-echo)# frequency 10
R1(config-ip-sla-echo)# exit
R1(config)# ip sla schedule 22 start-time now life forever
```



```
R1(config)# track 2 ip sla 22 reachability
R1(config-track)# delay down 10 up 1
R1(config-track)# exit
R1(config)# ip route 0.0.0.0 0.0.0.0 10.1.1.1 3 track 2
```

Configuring PBR and IP SLA Example

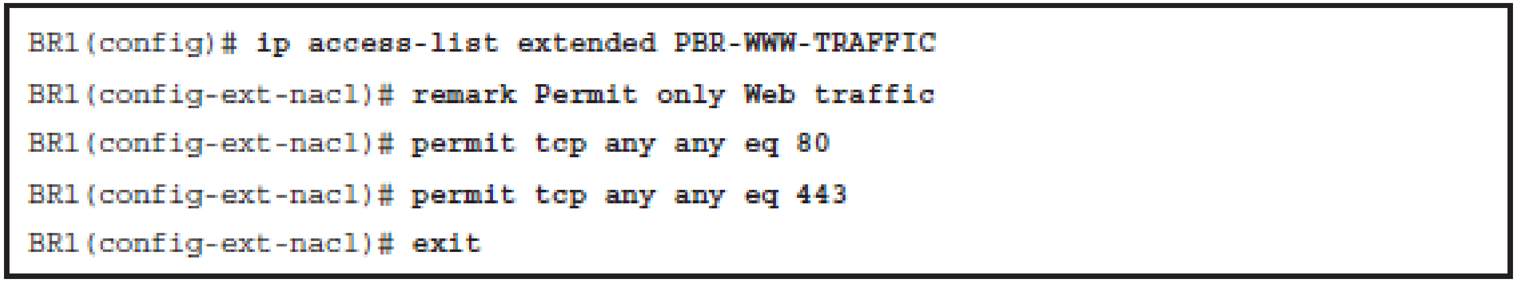
- In this scenario, traffic paths for the clients at first branch office (router BR1) will be optimized using PBR and IP SLA. EIGRP is already configured between HQ and BR1, and all traffic flows over the Ethernet WAN link because it has the lowest EIGRP metric route.



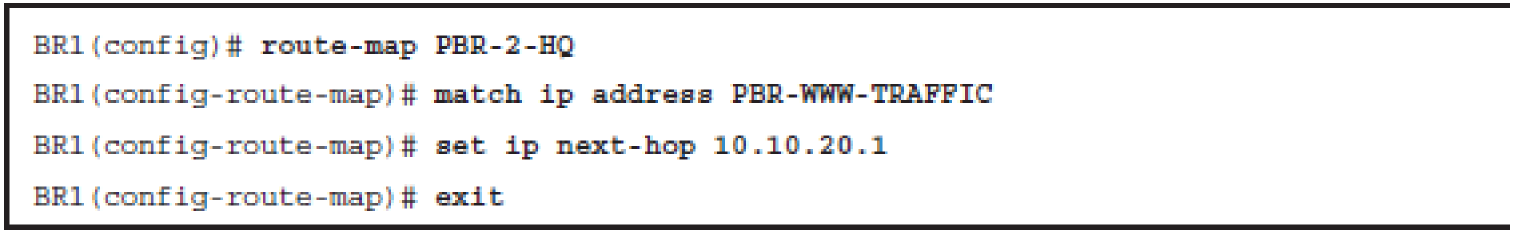
In the example, you will

- The new network policy for BR1 dictates that
  - Web traffic to the HQ site should be redirected over the serial link.
  - All other traffic from Notebook should go via BR2 but only if BR2 is reachable.
- In the example, you will
  - Redirect web traffic from clients on the BR1 router going to the HQ router over the serial link using PBR
  - Ensure that BR2 is reachable by using an IP SLA ICMP echo test to its WAN interface
  - Redirect all other traffic from Notebook to router BR2 if BR2 is reachable

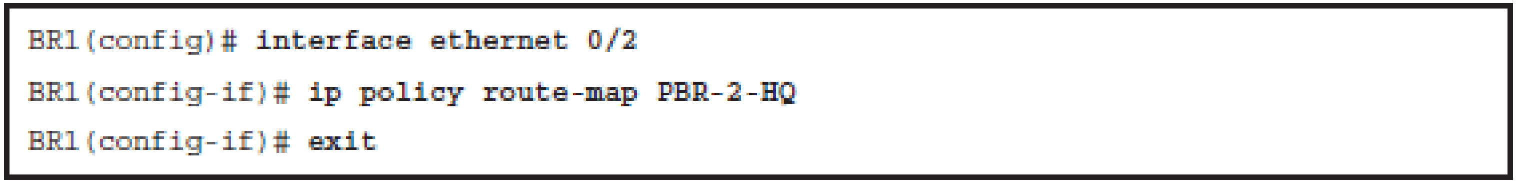
Redirecting Web Traffic from BR1 to HQ Using PBR



```
BR1(config)# ip access-list extended PBR-WWW-TRAFFIC
BR1(config-ext-nacl)# remark Permit only Web traffic
BR1(config-ext-nacl)# permit tcp any any eq 80
BR1(config-ext-nacl)# permit tcp any any eq 443
BR1(config-ext-nacl)# exit
```



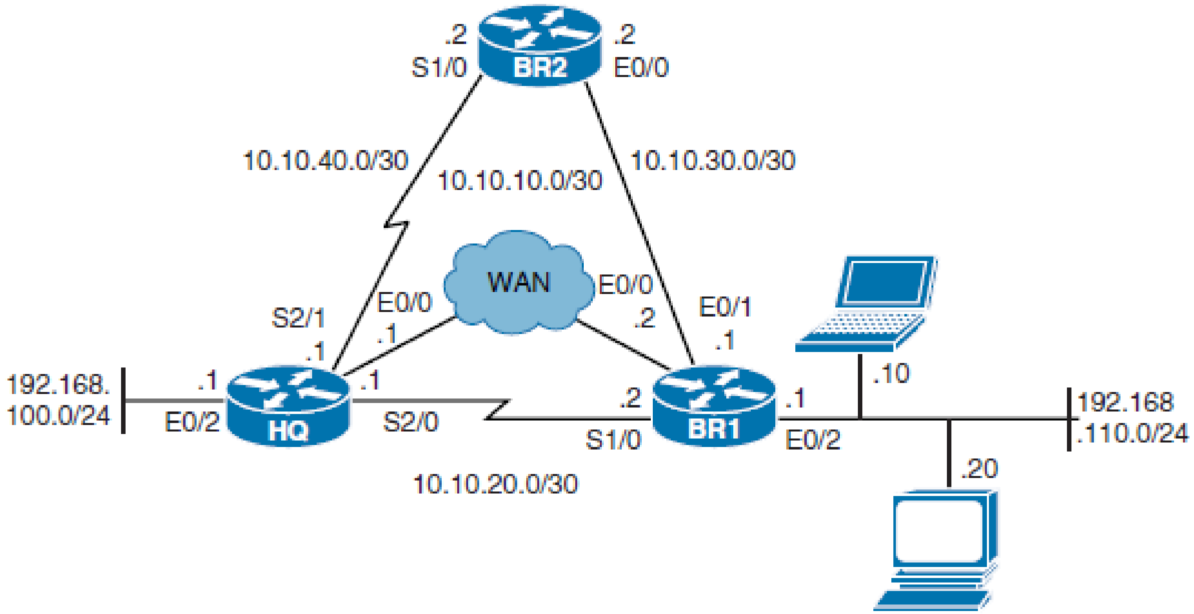
```
BR1(config)# route-map PBR-2-HQ
BR1(config-route-map)# match ip address PBR-WWW-TRAFFIC
BR1(config-route-map)# set ip next-hop 10.10.20.1
BR1(config-route-map)# exit
```



```
BR1(config)# interface ethernet 0/2
BR1(config-if)# ip policy route-map PBR-2-HQ
BR1(config-if)# exit
```

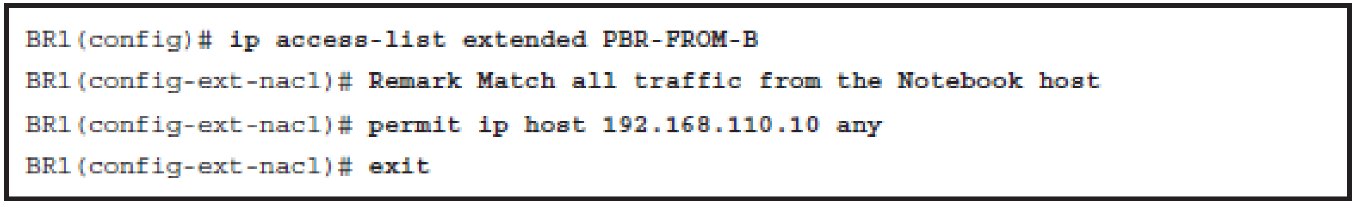
Ensuring That BR2 Is Reachable Using IP SLA

```
BR1(config)# ip sla 1
BR1(config-ip-sla)# icmp-echo 10.10.30.2 source-interface Ethernet 0/1
BR1(config-ip-sla-echo)# frequency 10
BR1(config-ip-sla-echo)# exit
BR1(config)# ip sla schedule 1 start-time now life forever
```

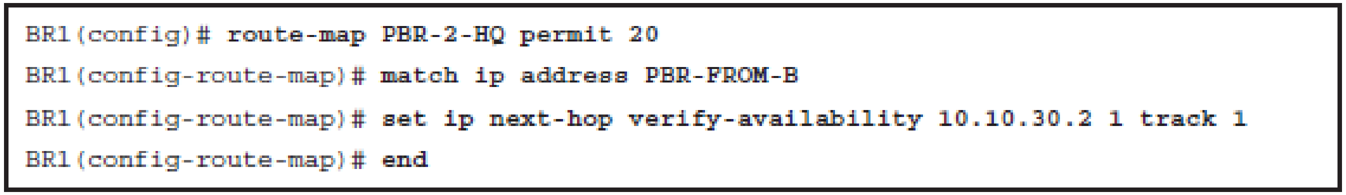


```
BR1(config)# track 1 ip sla 1
BR1(config-track)# delay down 5 up 1
BR1(config-track)# exit
```

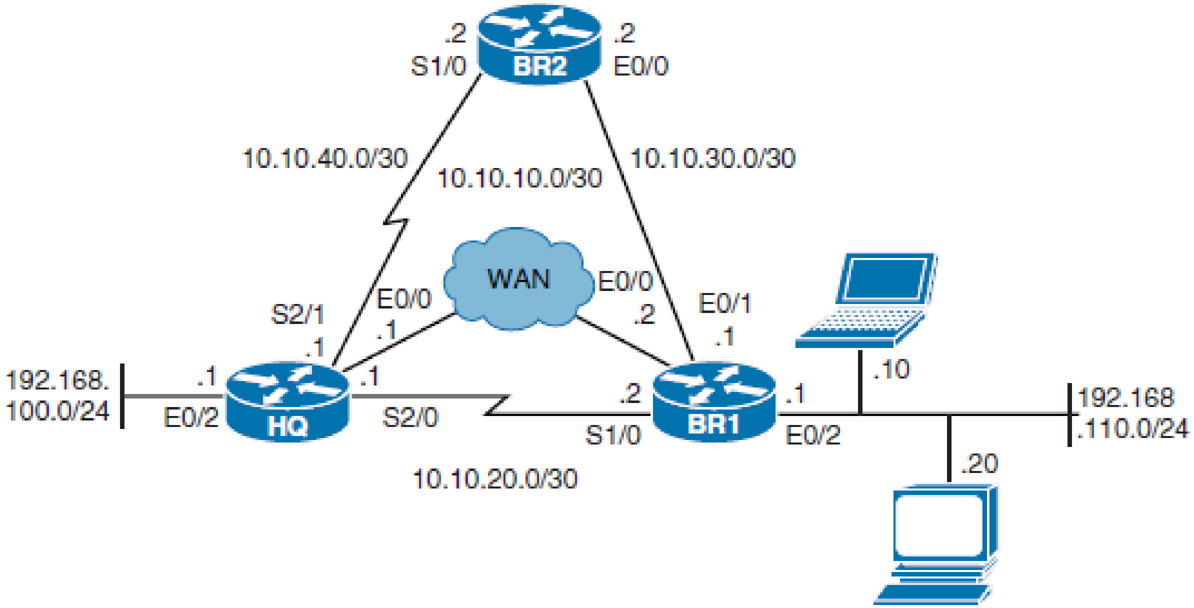
Redirect Traffic from Notebook to BR2 If Reachable



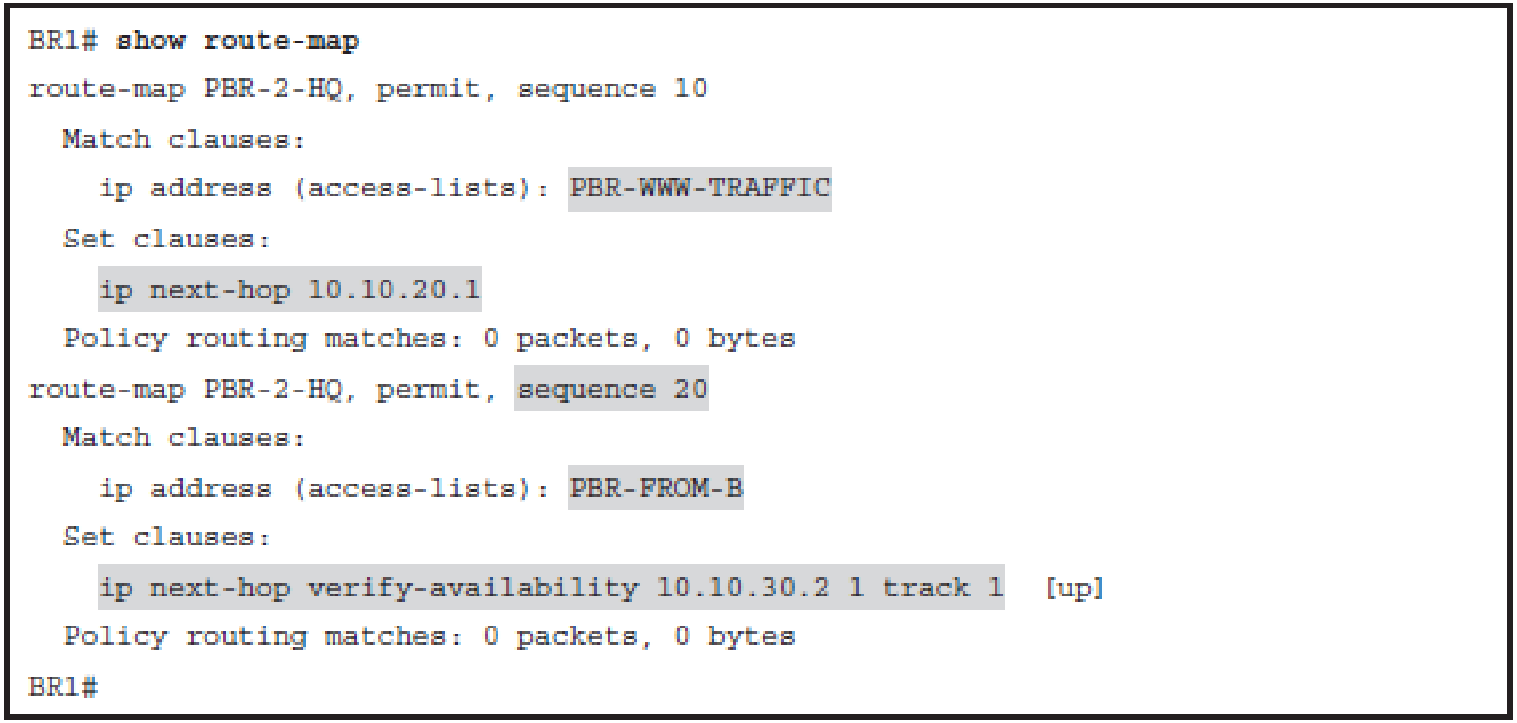
```
BR1(config)# ip access-list extended PBR-FROM-B
BR1(config-ext-nacl)# Remark Match all traffic from the Notebook host
BR1(config-ext-nacl)# permit ip host 192.168.110.10 any
BR1(config-ext-nacl)# exit
```



```
BR1(config)# route-map PBR-2-HQ permit 20
BR1(config-route-map)# match ip address PBR-FROM-B
BR1(config-route-map)# set ip next-hop verify-availability 10.10.30.2 1 track 1
BR1(config-route-map)# end
```

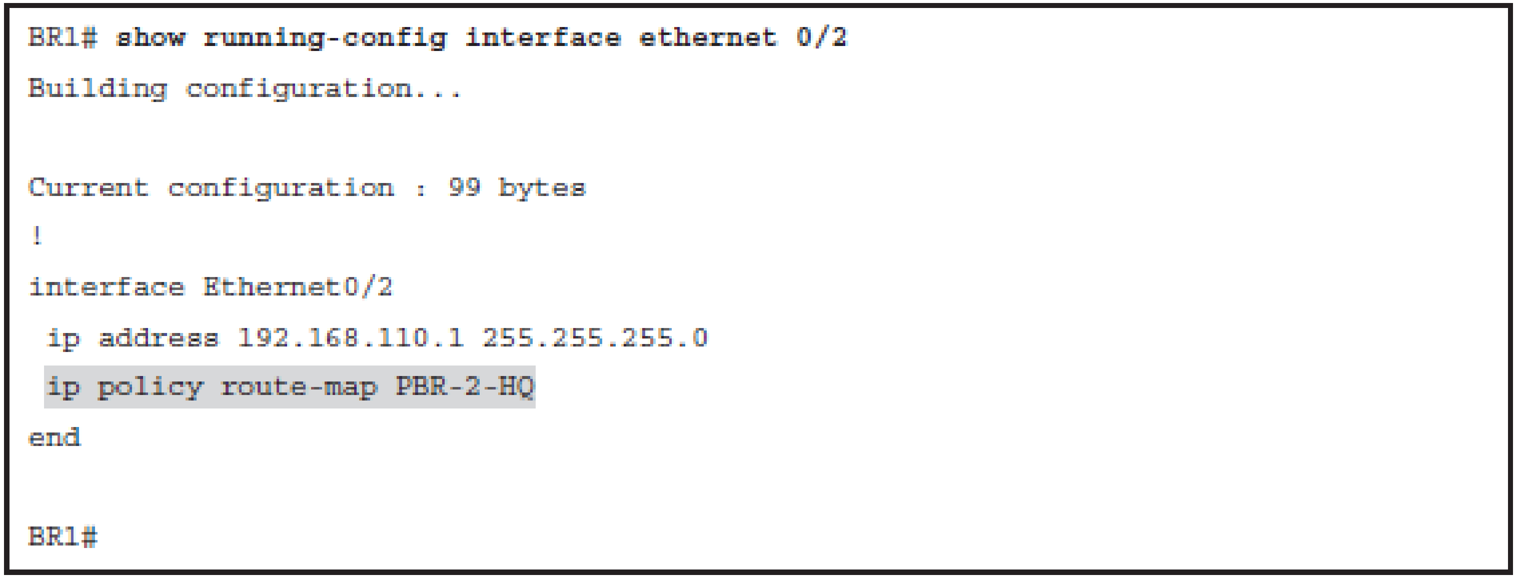


Verify Route Maps



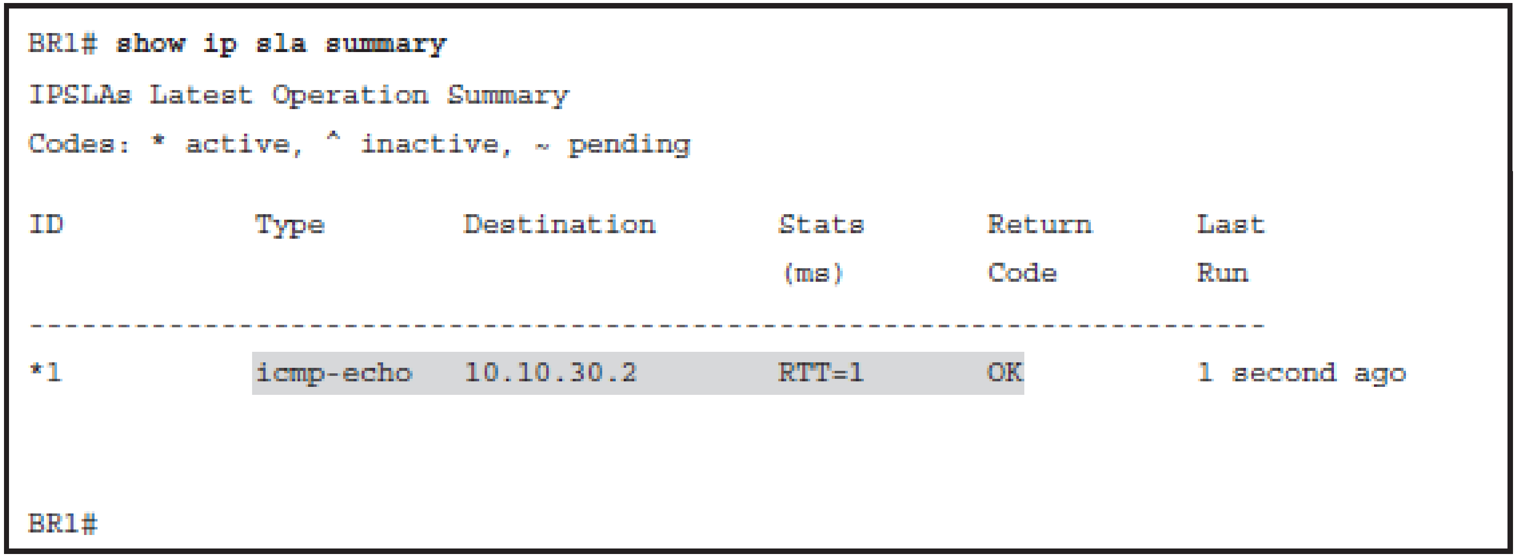
```
BR1# show route-map
route-map PBR-2-HQ, permit, sequence 10
  Match clauses:
    ip address (access-lists): PBR-WWW-TRAFFIC
  Set clauses:
    ip next-hop 10.10.20.1
  Policy routing matches: 0 packets, 0 bytes
route-map PBR-2-HQ, permit, sequence 20
  Match clauses:
    ip address (access-lists): PBR-FROM-B
  Set clauses:
    ip next-hop verify-availability 10.10.30.2 1 track 1  [up]
  Policy routing matches: 0 packets, 0 bytes
BR1#
```

Verify That the Route Map Is Applied



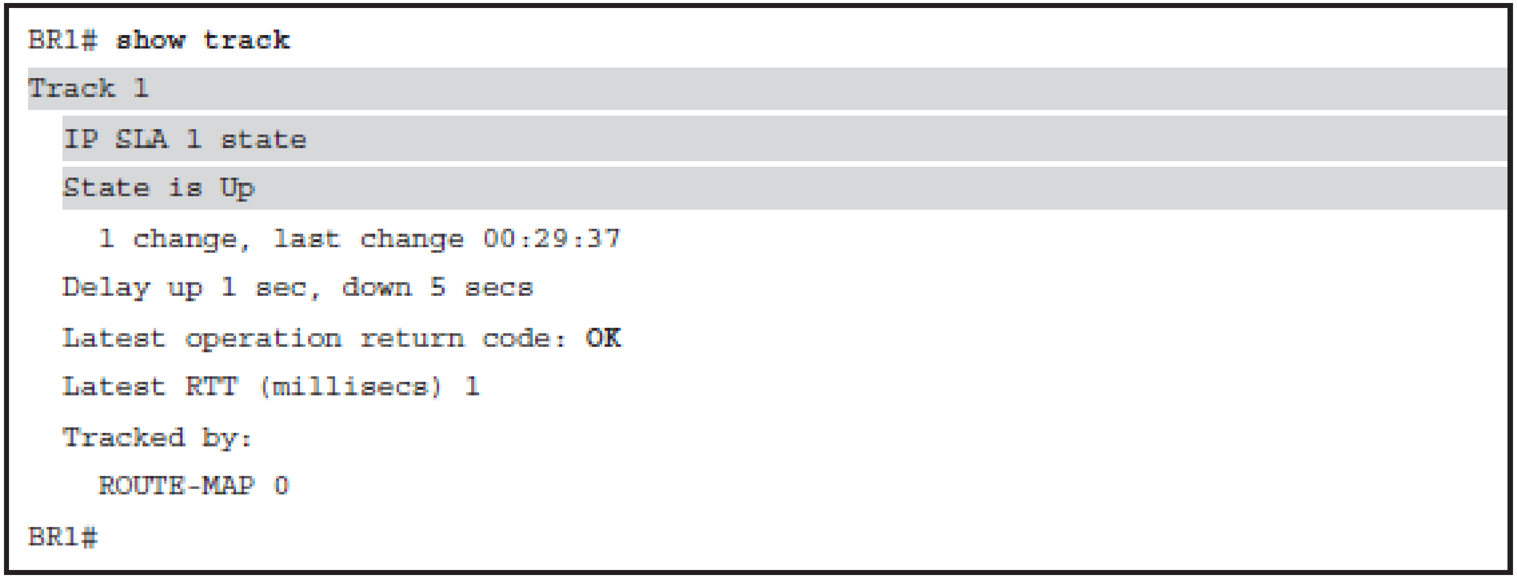
```
BR1# show running-config interface ethernet 0/2
Building configuration...

Current configuration : 99 bytes
!
interface Ethernet0/2
 ip address 192.168.110.1 255.255.255.0
 ip policy route-map PBR-2-HQ
end

BR1#
```

Verify IP SLA Operations



```
BR1# show ip sla summary
IPSLAs Latest Operation Summary
Codes: * active, ^ inactive, ~ pending

ID           Type        Destination      Stats      Return     Last
                                          (ms)       Code       Run
-----------------------------------------------------------------------
*1           icmp-echo   10.10.30.2       RTT=1      OK         1 second ago


BR1#
```

Verify Tracking Objects



```
BR1# show track
Track 1
  IP SLA 1 state
  State is Up
    1 change, last change 00:29:37
  Delay up 1 sec, down 5 secs
  Latest operation return code: OK
  Latest RTT (millisecs) 1
  Tracked by:
    ROUTE-MAP 0
BR1#
```

Chapter 5 Summary

- Packet-switching mechanisms on a Cisco IOS platform, including process switching, fast switching, and CEF switching.
- Overview of path control tools, including PBR and Cisco IOS IP SLAs.
- Using PBR to control path selection, providing benefits including source-based transit provider selection, QoS, cost savings, and load sharing. PBR is applied to *incoming* packets; enabling PBR causes the router to evaluate all packets incoming on the interface using a route map configured for that purpose.
- Configuring and verifying PBR, including the following steps:
  - Choose the path control tool to use; for PBR, **route-map** commands are used
  - Implement the traffic-matching configuration, specifying which traffic will be manipulated; **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 or to traffic locally generated on the router
  - Verify path control results, using **show** commands

Chapter 5 Summary

- Cisco IOS IP SLAs, which use active traffic monitoring, generating traffic in a continuous, reliable, and predictable manner, to measure network performance. IOS IP SLAs can be used in conjunction with other tools, including the following:
  - Object tracking, to track the reachability of specified objects
  - Cisco IOS IP SLAs probes, to send different types of probes toward the desired objects
  - Static routes with tracking options, as a simpler alternative to PBR
  - Route maps with PBR, to associate the results of the tracking to the routing process
- Cisco IOS IP SLA terminology, including the following:
  - All the Cisco IOS IP SLA measurement probe operations are configured on the IP SLA source, either by the CLI or through an SNMP tool that supports IP SLA operation. The source sends probe packets to the target.
  - There are two types of IP SLA operations: those in which the target device is running the IP SLA responder component, and those in which the target device is not running the IP SLA responder component (such as a web server or IP host).
  - An IP SLA operation is a measurement that includes protocol, frequency, traps, and thresholds.
- Configuring and verifying IOS IP SLAs.

Chapter 5 Labs

- **CCNPv7 ROUTE Lab5.1 Path Control Using PBR**
- **CCNPv7 ROUTE Lab 5.2 IP SLA Tracking and Path Control**

Cisco | Networking Academy®

Mind Wide Open™

Acknowledgment

- Some of images and texts are from Implementing Cisco IP Routing (ROUTE) Foundation Learning Guide by Diane Teare, Bob Vachon and Rick Graziani (1587204568)
- Copyright © 2015 – 2016 Cisco Systems, Inc.
- Special Thanks to *Bruno Silva*