Chapter 15: Route Maps and Conditional Forwarding

Instructor Materials

CCNP Enterprise: Advanced
Routing

Chapter 15 Content

This chapter covers the following content:

- **Conditional Matching -** This section provides an overview of how network prefixes can be conditionally matched with ACLs or prefix lists.

- **Route Maps -** This section explains the structure of a route map and how conditional matching and conditional actions can be combined to filter or manipulate routes.

- **Conditional Forwarding of Packets -** This section explains how a router forwards packets down different paths based on the network traffic.

- **Trouble Tickets -** This section provides three trouble tickets that demonstrate how a structured troubleshooting process can be used to solve a reported problem.

Conditional Matching

- Applying bulk changes to routes does not allow for easy tuning of the network.
- This section reviews some of the common techniques used to conditionally match a route—using access control lists (ACLs) and prefix lists.

Conditional Matching

Access Control Lists (ACLs)

- ACLs provide packet classification for a variety of features, such as quality of service (QoS), or for identifying networks within routing protocols.
- ACLs are composed of access control entries (ACEs), which are entries in the ACL that identify the action to be taken (permit or deny) and the relevant packet classification.
- Packet classification starts at the top (lowest sequence) and proceeds down (higher sequence) until a matching pattern is identified. When a match is found, the appropriate action (permit or deny) is taken, and processing stops.
- At the end of every ACL is an implicit deny ACE, which denies all packets that did not match earlier in the ACL.

Conditional Matching

Access Control Lists (Cont.)

ACLs are classified into two categories:

- **Standard ACLs -** Define packets based solely on the source network. Standard ACLS use a numbered entry 1–99 or 1300–1999 or a named ACL.

- **Extended ACLs -** Define packets based on the source, destination, protocol, port, or a combination of other packet attributes. Extended ACLs use a numbered entry 100–199 or 2000–2699 or a named ACL.

Named ACLs provide relevance to the functionality of an ACL, can be used with standard or extended ACLs, and are generally preferred.

Note: ACE placement within an ACL is important, and unintended consequences may result from ACEs being out of order.

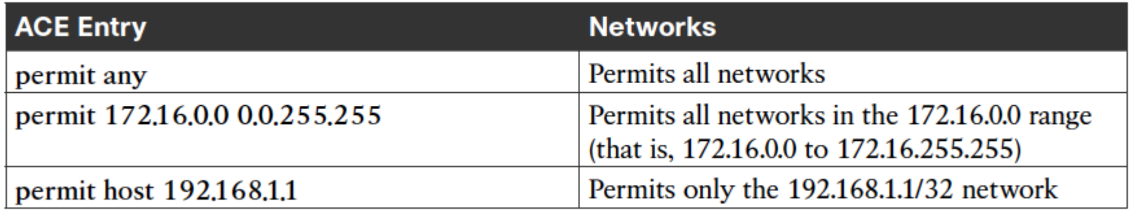
Conditional Matching

Standard ACLs

The process for defining a standard ACL is as follows:
Step 1. Define the ACL by using the command **ip access-list standard** {*acl-number* | *acl-name*} and placing the CLI in ACL configuration mode.

Step 2. Configure the specific ACE entry with the command [*sequence*] {**permit | deny**} *source source-wildcard*. In lieu of using *source source-wildcard*, the keyword **any** replaces *0.0.0.0 0.0.0.0*, and use of the **host** keyword refers to a /32 IP address so that the *source-wildcard* can be omitted.

Table 15-2 Standard ACL-to-Network Entries



ACE Entry	Networks
permit any	Permits all networks
permit 172.16.0.0 0.0.255.255	Permits all networks in the 172.16.0.0 range (that is, 172.16.0.0 to 172.16.255.255)
permit host 192.168.1.1	Permits only the 192.168.1.1/32 network

Table 15-2 provides sample ACL entries from within the ACL configuration mode and specifies the networks that would match with a standard ACL.

Conditional Matching

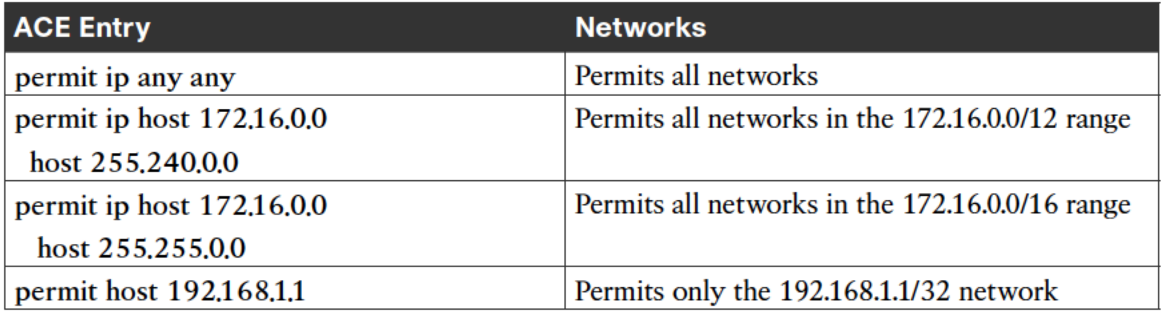
Extended ACLs

The process for defining an extended ACL is as follows:

Step 1. Define the ACL by using the command **ip access-list extended** {*acl-number* | *acl-name*} and placing the CLI in ACL configuration mode.

Step 2. Configure the specific ACE entry with the command [*sequence*] {**permit | deny**} *protocol source source-wildcard destination destination-wildcard*.

Table 15-3 Extended ACL for IGP Route Selection



ACE Entry	Networks
permit ip any any	Permits all networks
permit ip host 172.16.0.0 host 255.240.0.0	Permits all networks in the 172.16.0.0/12 range
permit ip host 172.16.0.0 host 255.255.0.0	Permits all networks in the 172.16.0.0/16 range
permit host 192.168.1.1	Permits only the 192.168.1.1/32 network

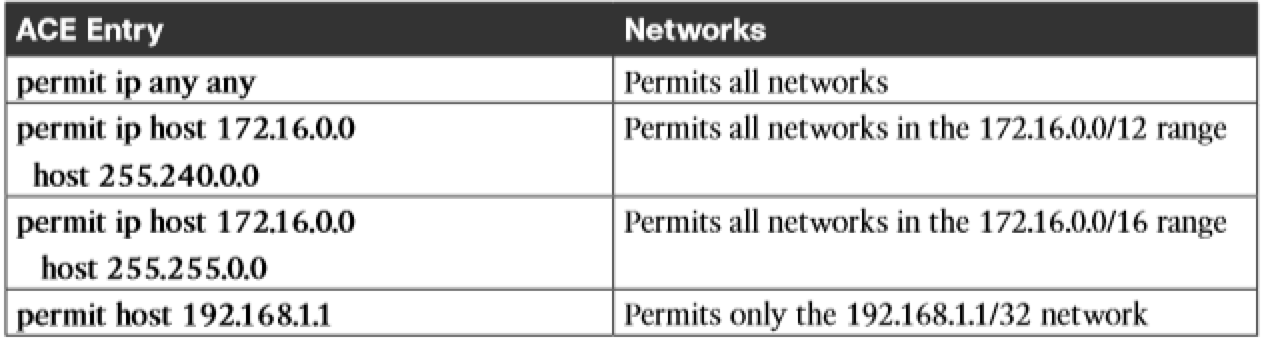
Table 15-3 provides sample ACL entries from within the ACL configuration mode and specifies the networks that would match with the extended ACL.

Conditional Matching

IGP Network Selection

When ACLS are used for IGP network selection, the source fields of the ACL are used to identify the network, and the destination fields identify the smallest prefix length allowed in the network range. Table 15-3 provides sample ACL entries from within the ACL configuration mode and specifies the networks that would match with the extended ACL.

Table 15-3 Extended ACL for IGP Route Selection



ACE Entry	Networks
permit ip any any	Permits all networks
permit ip host 172.16.0.0 host 255.240.0.0	Permits all networks in the 172.16.0.0/12 range
permit ip host 172.16.0.0 host 255.255.0.0	Permits all networks in the 172.16.0.0/16 range
permit host 192.168.1.1	Permits only the 192.168.1.1/32 network

Conditional Matching

BGP Network Selection

Extended ACLs react differently when matching BGP routes than when matching IGP routes. The source fields match against the network portion of the route, and the destination fields match against the network mask, as shown in Figure 15-1. Until the introduction of prefix lists, extended ACLs were the only match criteria used with BGP.

Table 15-4 demonstrates the concept of the wildcard for the network mask and subnet mask.

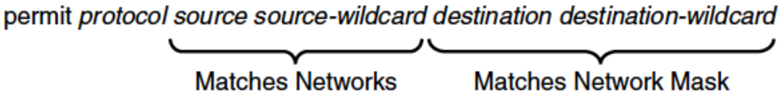
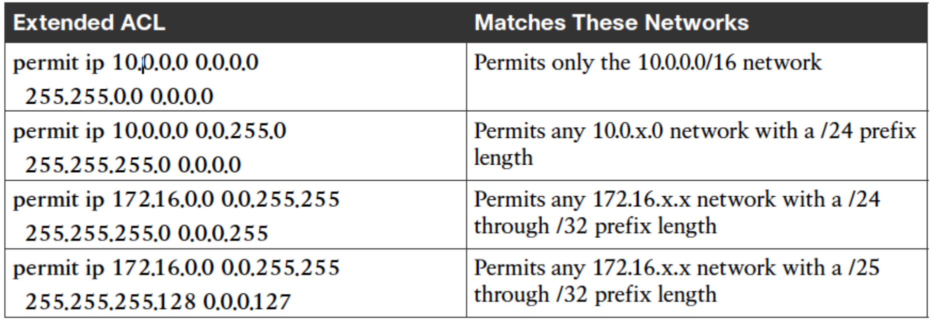

permit *protocol source source-wildcard destination destination-wildcard*
Matches Networks
Matches Network Mask

Figure 15-1 *BGP Extended ACL Matches*

Table 15-4 Extended ACL for BGP Route Selection



Extended ACL	Matches These Networks
permit ip 10.0.0.0 0.0.0.0 255.255.0.0 0.0.0.0	Permits only the 10.0.0.0/16 network
permit ip 10.0.0.0 0.0.255.0 255.255.255.0 0.0.0.0	Permits any 10.0.x.0 network with a /24 prefix length
permit ip 172.16.0.0 0.0.255.255 255.255.255.0 0.0.0.255	Permits any 172.16.x.x network with a /24 through /32 prefix length
permit ip 172.16.0.0 0.0.255.255 255.255.255.128 0.0.0.127	Permits any 172.16.x.x network with a /25 through /32 prefix length

Conditional Matching

Prefix Matching

Prefix lists provide another method of identifying networks in a routing protocol. A prefix list identifies a specific IP address, network, or network range and allows for the selection of multiple networks with a variety of prefix lengths, using a prefix match specification.

A prefix match specification contains two parts: a high-order bit pattern and a high-order bit count, which determines the high-order bits in the bit pattern that are to be matched. Some documentation refers to the high-order bit pattern as the address or network and the high-order bit count as the length or mask length.

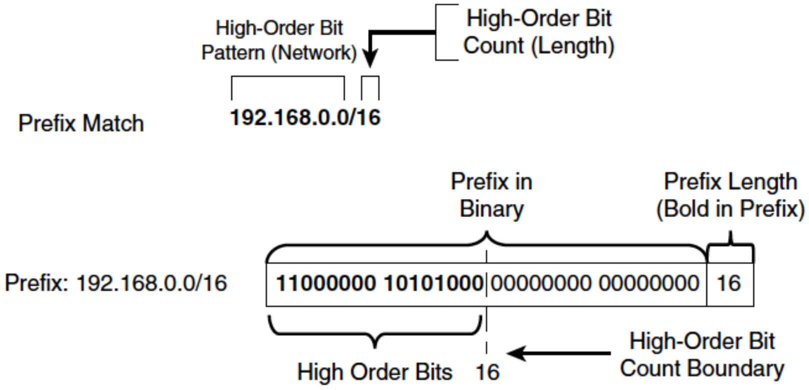

High-Order Bit Pattern (Network)
High-Order Bit Count (Length)
Prefix Match
192.168.0.0/16
Prefix in Binary
Prefix Length (Bold in Prefix)
Prefix: 192.168.0.0/16
11000000 10101000 00000000 00000000
16
High Order Bits
16
High-Order Bit Count Boundary

Figure 15-2 *Basic Prefix Match Pattern*

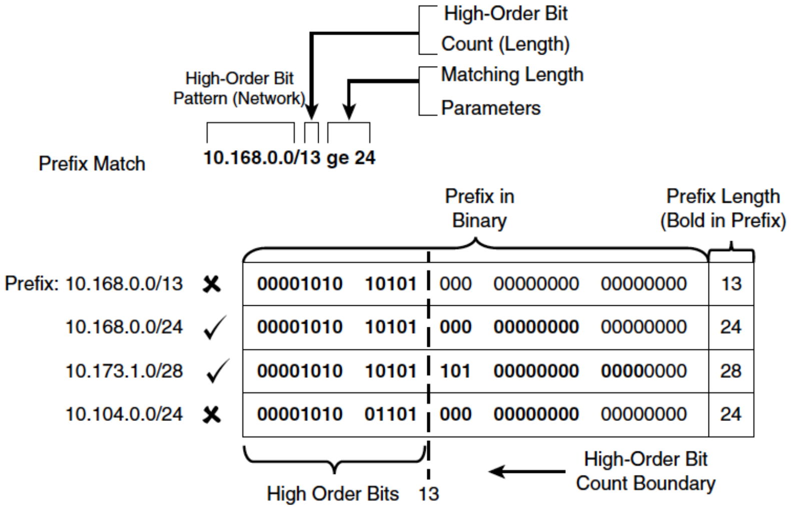
In Figure 15-2, the prefix match specification has the high-order bit pattern 192.168.0.0 and the high-order bit count 16.

Conditional Matching

Prefix Matching (Cont.)

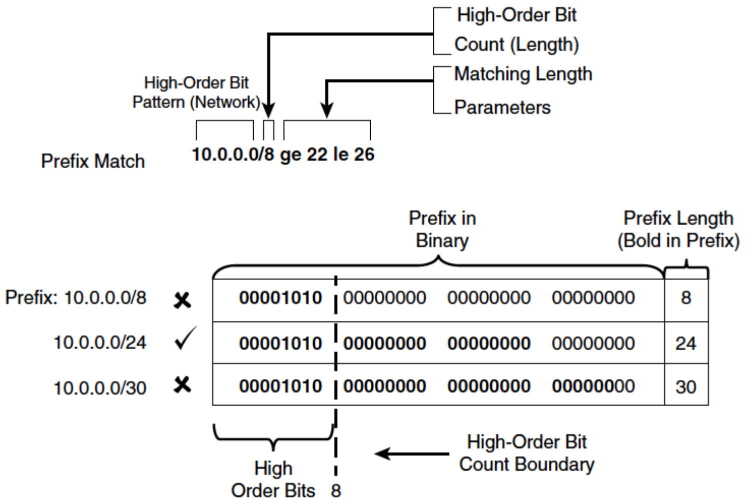
Figure 15-3 demonstrates the prefix match specification with a high-order bit pattern of 10.168.0.0 and high-order bit count of 13, where the matching length of the prefix must be greater than or equal to 24.

Figure 15-4 demonstrates a prefix match specification with a high order bit pattern of 10.0.0.0 and high-order bit count of 8, where the matching length must be between 22 and 26.



		Prefix in Binary					Prefix Length (Bold in Prefix)
Prefix: 10.168.0.0/13	✗	00001010	10101	000	00000000	00000000	13
10.168.0.0/24	✓	00001010	10101	000	00000000	00000000	24
10.173.1.0/28	✓	00001010	10101	101	00000000	00000000	28
10.104.0.0/24	✗	00001010	01101	000	00000000	00000000	24

Figure 15-3 *Prefix Match Pattern with Matching Length Parameters*



		Prefix in Binary				Prefix Length (Bold in Prefix)
Prefix: 10.0.0.0/8	✗	00001010	00000000	00000000	00000000	8
10.0.0.0/24	✓	00001010	00000000	00000000	00000000	24
10.0.0.0/30	✗	00001010	00000000	00000000	00000000	30

Figure 15-4 *Prefix Match with Ineligible Matched Prefixes*

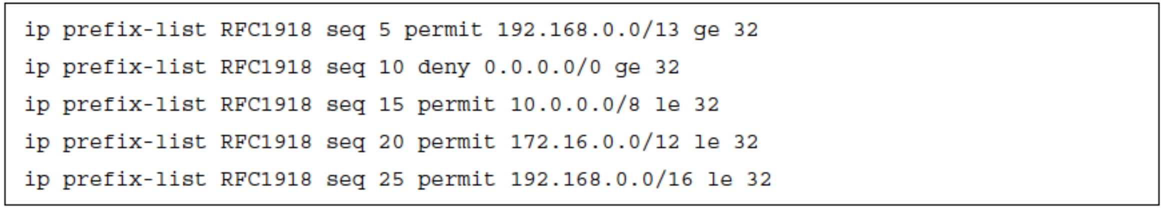
Conditional Matching

Prefix Lists

A prefix list can have multiple prefix matching specification entries that contain permit or deny actions. Prefix lists process in sequential order in a top-down fashion, and the first prefix match processes with the appropriate permit or deny action. Prefix lists are configured with the following global configuration command syntax:

ip prefix-list *prefix-list-name* [**seq** *sequence-number*] {**permit | deny**} *high-order-bit-pattern/high-order-bit-count* [**ge** *ge-value*] [**le** *le-value*]

Example 15-1 *Sample Prefix List*



```
ip prefix-list RFC1918 seq 5 permit 192.168.0.0/13 ge 32
ip prefix-list RFC1918 seq 10 deny 0.0.0.0/0 ge 32
ip prefix-list RFC1918 seq 15 permit 10.0.0.0/8 le 32
ip prefix-list RFC1918 seq 20 permit 172.16.0.0/12 le 32
ip prefix-list RFC1918 seq 25 permit 192.168.0.0/16 le 32
```

If a sequence is not provided, the sequence number auto-increments by 5, based on the highest sequence number. The first entry is 5. Example 15-1 provides a sample prefix list named RFC1918 for all the networks in the RFC 1918 address range.

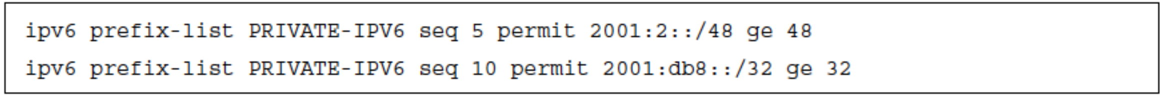
Conditional Matching

IPv6 Prefix Lists

The prefix matching logic works exactly the same for IPv6 networks as for IPv4 networks.
The most important thing to remember is that IPv6 networks are notated in hex and not in binary when identifying ranges. IPv6 prefix lists are configured with the following global configuration command syntax:

ipv6 prefix-list *prefix-list-name* [**seq** *sequence-number*] {**permit** | **deny**} *high-order-bit-pattern*/*high-order-bit-count* [**ge** *ge-value*] [**le** *le-value*]

Example 15-2 *Sample IPv6 Prefix List*



```
ipv6 prefix-list PRIVATE-IPV6 seq 5 permit 2001:2::/48 ge 48
ipv6 prefix-list PRIVATE-IPV6 seq 10 permit 2001:db8::/32 ge 32
```

Example 15-2 provides a sample prefix list named PRIVATE-IPV6 for all the networks in the documentation and benchmarking IPv6 space.

Route Maps

- Route maps provide many different features to a variety of routing protocols.
- At the simplest level, route maps can filter networks much the same way as ACLs, but they also provide additional capability through the addition or modification of network attributes.
- To influence a routing protocol, a route map must be referenced from the routing protocol.

Route Maps

Route maps are critical to BGP because they are the main component in modifying a unique routing policy on a neighbor-by-neighbor basis. Route maps have four components:

- **Sequence number** - Dictates the processing order of the route map.
- **Conditional matching criterion** - Identifies prefix characteristics (network, BGP path attribute, next hop, and so on) for a specific sequence.
- **Processing action** - Permits or denies the prefix.
- **Optional action** - Allows for manipulations dependent on how the route map is referenced on the router. Actions can include modification, addition, or removal of route characteristics.

Route Maps

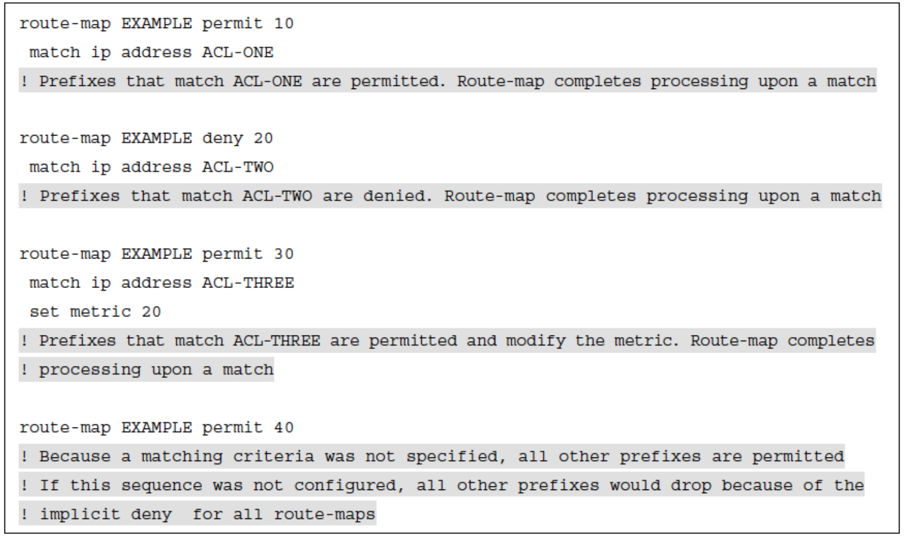
Route Map Configuration

A route map use the following command syntax:
route-map *route-map-name* [**permit** | **deny**] [*sequence-number*]

The following rules apply to route-map statements:

- If a processing action is not provided, the default value of permit is used.
- If a sequence number is not provided, the sequence number increments by 10 automatically.
- If a matching statement is not included, an implied all prefixes is associated with the statement.
- Processing within a route map stops after all optional actions have been processed (if configured) after matching a matching criterion.

Example 15-3 *Sample Route Map*



```
route-map EXAMPLE permit 10
 match ip address ACL-ONE
! Prefixes that match ACL-ONE are permitted. Route-map completes processing upon a match

route-map EXAMPLE deny 20
 match ip address ACL-TWO
! Prefixes that match ACL-TWO are denied. Route-map completes processing upon a match

route-map EXAMPLE permit 30
 match ip address ACL-THREE
 set metric 20
! Prefixes that match ACL-THREE are permitted and modify the metric. Route-map completes
! processing upon a match

route-map EXAMPLE permit 40
! Because a matching criteria was not specified, all other prefixes are permitted
! If this sequence was not configured, all other prefixes would drop because of the
! implicit deny  for all route-maps
```

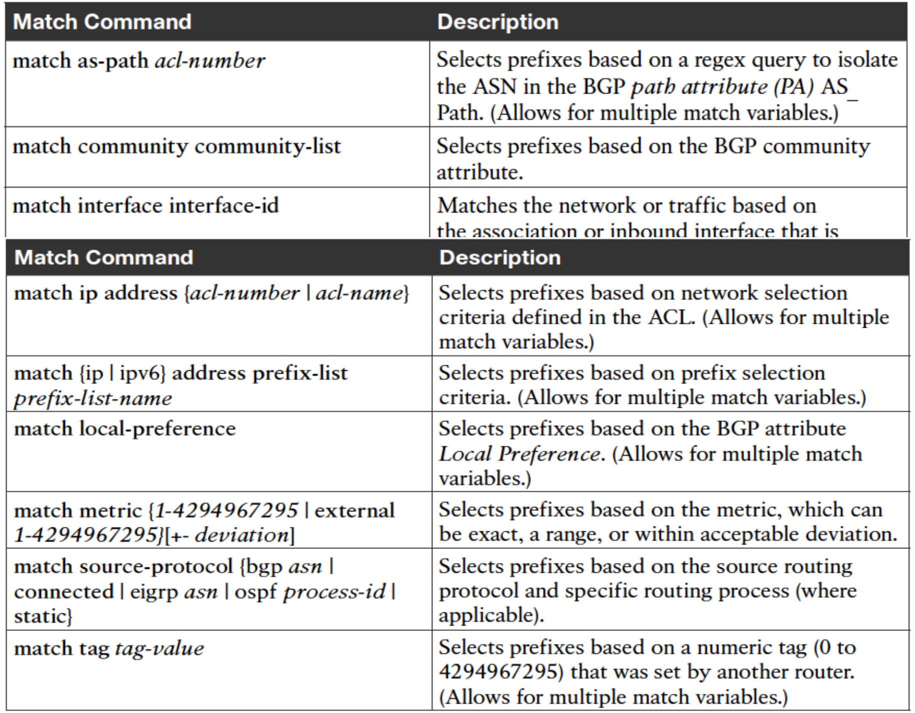
Example 15-3 provides a sample route map to demonstrate the four components of a route map.

Route Maps

Conditional Matching Commands

Table 15-5 provides the command syntax for the most common methods for conditionally matching prefixes and describes their usage

Table 15-5 Conditional Match Options



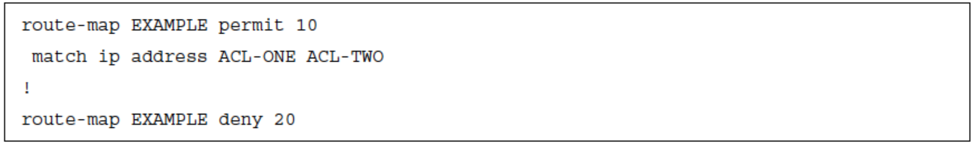
Match Command	Description
match as-path *acl-number*	Selects prefixes based on a regex query to isolate the ASN in the BGP *path attribute (PA)* AS_Path. (Allows for multiple match variables.)
match community community-list	Selects prefixes based on the BGP community attribute.
match interface interface-id	Matches the network or traffic based on the association or inbound interface that is
match ip address {*acl-number* | *acl-name*}	Selects prefixes based on network selection criteria defined in the ACL. (Allows for multiple match variables.)
match {ip | ipv6} address prefix-list *prefix-list-name*	Selects prefixes based on prefix selection criteria. (Allows for multiple match variables.)
match local-preference	Selects prefixes based on the BGP attribute *Local Preference*. (Allows for multiple match variables.)
match metric {*1-4294967295* | external *1-4294967295*}[+- *deviation*]	Selects prefixes based on the metric, which can be exact, a range, or within acceptable deviation.
match source-protocol {bgp *asn* | connected | eigrp *asn* | ospf *process-id* | static}	Selects prefixes based on the source routing protocol and specific routing process (where applicable).
match tag *tag-value*	Selects prefixes based on a numeric tag (0 to 4294967295) that was set by another router. (Allows for multiple match variables.)

Route Maps

Multiple Conditional Match Conditions

If multiple variables (ACLs, prefix lists, tags, and so on) are configured for a specific route map sequence, only one variable must match for the prefix to qualify. The Boolean logic uses an *or* operator for this configuration. If multiple match options are configured for a specific route map sequence, both match options must be met for the prefix to qualify for that sequence. The Boolean logic uses an *and* operator for this configuration.

Example 15-4 *Multiple Match Variables Sample Route Map*



```
route-map EXAMPLE permit 10
 match ip address ACL-ONE ACL-TWO
!
route-map EXAMPLE deny 20
```

```
route-map EXAMPLE permit 10
match ip address ACL-ONE
match metric 550 +- 50
```

In Example 15-4, sequence 10 requires that a prefix pass ACL-ONE or ACL-TWO. Notice that sequence 20 does not have a match statement, so all prefixes that are not passed in sequence 10 qualify and are denied.

In the above snippet, sequence 10 requires that the prefix match ACL ACL-ONE and that the metric be a value between 500 and 600.

Route Maps

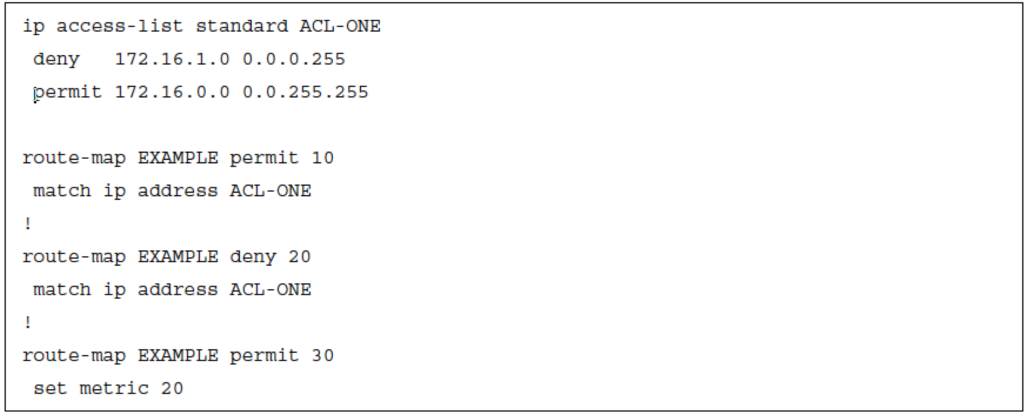
Complex Matching

Some network engineers find route maps too complex if the conditional matching criteria use an ACL, an AS_Path ACL, or a prefix list that contains a **deny** statement. Example 15-5 shows a configuration in which the ACL uses a **deny** statement for the 172.16.1.0/24 network range.

Configurations should follow the sequence order first and conditional matching criteria second, and only after a match occurs should the processing action and optional action be used.

Matching a **deny** statement in the conditional match criteria excludes the route from that sequence in the route map.

Example 15-5 *Complex Matching Route Maps*



```
ip access-list standard ACL-ONE
 deny   172.16.1.0 0.0.0.255
 permit 172.16.0.0 0.0.255.255

route-map EXAMPLE permit 10
 match ip address ACL-ONE
!
route-map EXAMPLE deny 20
 match ip address ACL-ONE
!
route-map EXAMPLE permit 30
 set metric 20
```

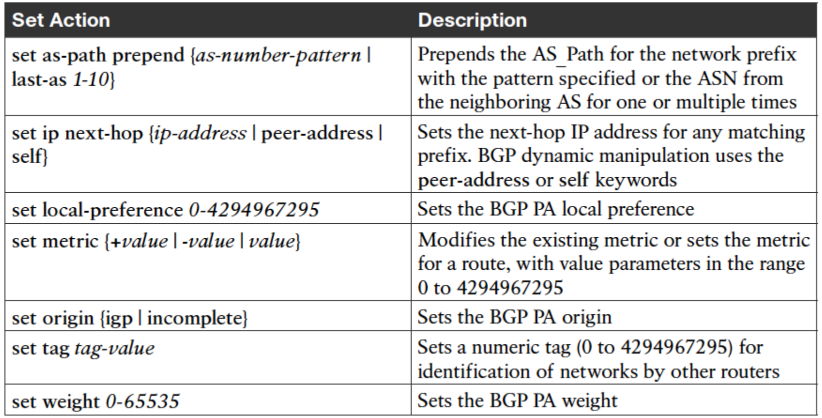
Route Maps

Optional Actions

In addition to permitting a prefix to pass, a route map can modify route attributes. Table 15-6 provides a brief overview of the most popular attribute modifications.

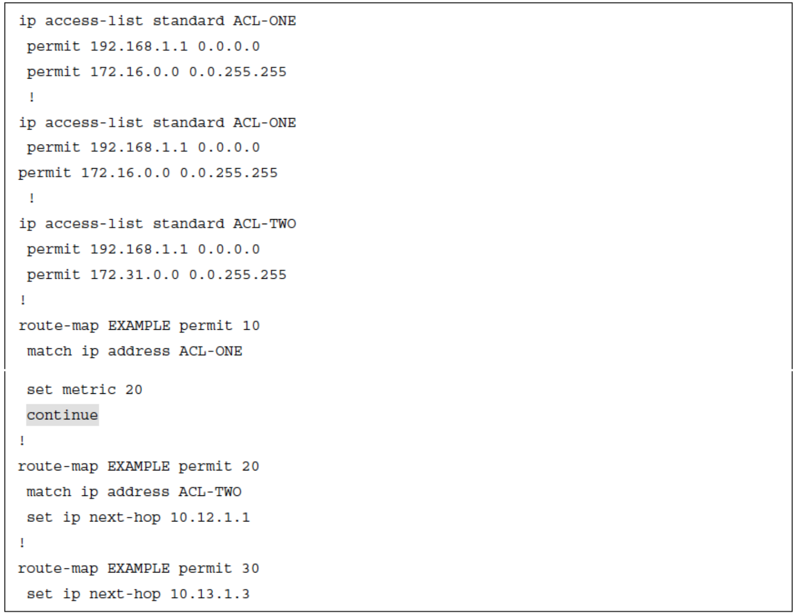
Adding the keyword **continue** to a route map allows the route map to continue processing other route map sequences. Example 15-6 provides a basic configuration.

Table 15-6 Route Map Set Actions



Set Action	Description
set as-path prepend {*as-number-pattern* | last-as *1-10*}	Prepends the AS_Path for the network prefix with the pattern specified or the ASN from the neighboring AS for one or multiple times
set ip next-hop {*ip-address* | peer-address | self}	Sets the next-hop IP address for any matching prefix. BGP dynamic manipulation uses the peer-address or self keywords
set local-preference *0-4294967295*	Sets the BGP PA local preference
set metric {+*value* | -*value* | *value*}	Modifies the existing metric or sets the metric for a route, with value parameters in the range 0 to 4294967295
set origin {igp | incomplete}	Sets the BGP PA origin
set tag *tag-value*	Sets a numeric tag (0 to 4294967295) for identification of networks by other routers
set weight *0-65535*	Sets the BGP PA weight

Example 15-6 *Route Map Configuration with the continue Keyword*



```
ip access-list standard ACL-ONE
 permit 192.168.1.1 0.0.0.0
 permit 172.16.0.0 0.0.255.255
  !
ip access-list standard ACL-ONE
 permit 192.168.1.1 0.0.0.0
permit 172.16.0.0 0.0.255.255
  !
ip access-list standard ACL-TWO
 permit 192.168.1.1 0.0.0.0
 permit 172.31.0.0 0.0.255.255
!
route-map EXAMPLE permit 10
 match ip address ACL-ONE
 set metric 20
 continue
!
route-map EXAMPLE permit 20
 match ip address ACL-TWO
 set ip next-hop 10.12.1.1
!
route-map EXAMPLE permit 30
 set ip next-hop 10.13.1.3
```

Conditional Forwarding of Packets

- A router makes forwarding decisions based on the destination addresses IP packets.
- Some scenarios accommodate other factors, such as packet length or source address, when deciding where the router should forward a packet.

Conditional Forwarding of Packets

Policy-Based Routing (PBR)

Policy-based routing (PBR) allows for conditional forwarding of packets based on packet characteristics besides the destination IP address. PBR provides the following capabilities:

- Routing by protocol type (ICMP, TCP, UDP, and so on)
- Routing by source IP address, destination IP address, or both
- Manual assignment of different network paths to the same destination, based on tolerance for latency, link speed, or utilization for specific transient traffic

Some of the drawbacks of conditional routing include the following:

- Administrative burden in scalability
- Lack of network intelligence
- Troubleshooting complexity

PBR Configuration

PBR configuration uses a route map with **match** and **set** statements that are attached to the inbound interface. It involves the following steps:

Step 1. Configure the route map by using the command **route-map** *route-map-name* [**permit** | **deny**] [*sequence-number*].

Step 2. Identify the conditional match criteria. The conditional match criteria can be based on packet length with the command **match length** *minimum-length maximum-length* or can be based on the packet IP address fields with an ACL using the command **match ip address** {*access-list-number* | *acl-name*}.

Step 3. Use the command **set ip** [**default**] **next-hop** *ip-address* [... *ip-address*] to specify one or more next hops for packets that match the criteria. The optional default keyword changes the behavior so that the next-hop address specified by the route map is used only if the destination address does not exist in the RIB. If a viable route exists in the RIB, then that is the next-hop address that is used for forwarding the packet.

Step 4. Apply the route map to the inbound interface by using the interface parameter command **ip policy route-map** *route-map-name*.

Conditional Forwarding of Packets

PBR Configuration (Cont.)

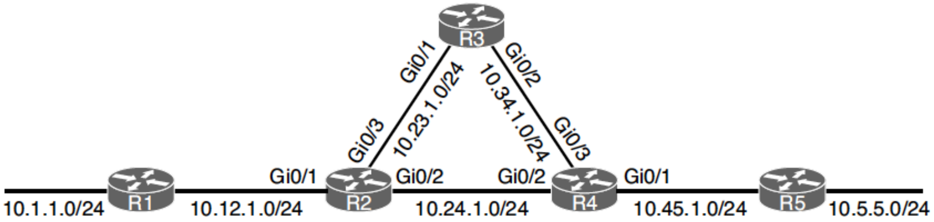
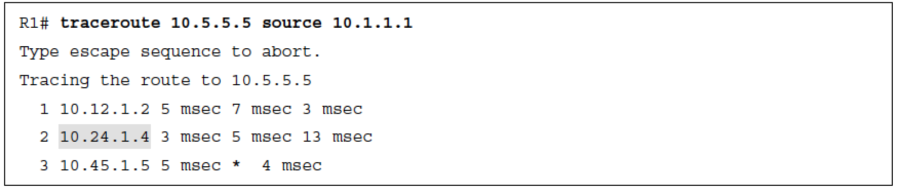


Figure 15-5 *PBR Next-Hop Topology*

Figure 15-5 provides a sample topology for illustrating PBR concepts. R1, R2, R3, R4, and R5 are all configured with OSPF.

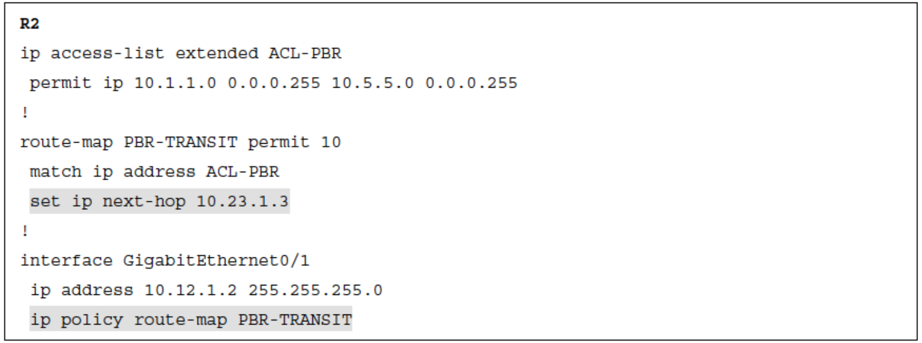
Example 15-7 *traceroute for Normal Traffic Flow*



```
R1# traceroute 10.5.5.5 source 10.1.1.1
Type escape sequence to abort.
Tracing the route to 10.5.5.5
  1 10.12.1.2 5 msec 7 msec 3 msec
  2 10.24.1.4 3 msec 5 msec 13 msec
  3 10.45.1.5 5 msec *  4 msec
```

Example 15-7 shows the normal traffic path using **traceroute** between the 10.1.1.0/24 and 10.5.5.0/24 networks without PBR configured.

Example 15-8 *Configuring Policy-Based Routing*



```
R2
ip access-list extended ACL-PBR
 permit ip 10.1.1.0 0.0.0.255 10.5.5.0 0.0.0.255
!
route-map PBR-TRANSIT permit 10
 match ip address ACL-PBR
 set ip next-hop 10.23.1.3
!
interface GigabitEthernet0/1
 ip address 10.12.1.2 255.255.255.0
 ip policy route-map PBR-TRANSIT
```

Example 15-8 shows the PBR configuration on R2 for network traffic from 10.1.1.0/24 destined for the 10.5.5.0/24 network to route traffic to 10.23.1.3 (R3).

Conditional Forwarding of Packets

PBR Configuration (Cont.)

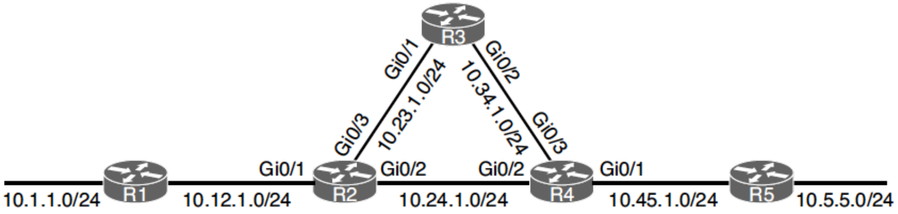
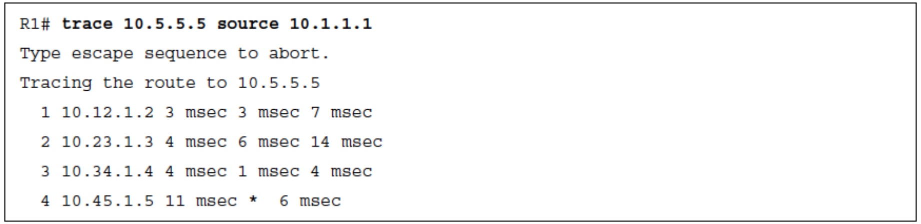


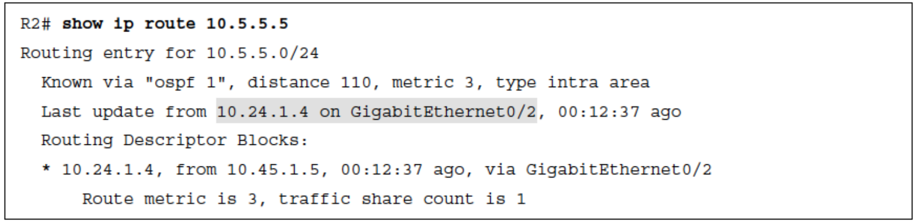
Figure 15-5 *PBR Next-Hop Topology*

Example 15-9 *R1 to R5 Paths Demonstrating PBR*



```
R1# trace 10.5.5.5 source 10.1.1.1
Type escape sequence to abort.
Tracing the route to 10.5.5.5
  1 10.12.1.2 3 msec 3 msec 7 msec
  2 10.23.1.3 4 msec 6 msec 14 msec
  3 10.34.1.4 4 msec 1 msec 4 msec
  4 10.45.1.5 11 msec *  6 msec
```

Example 15-10 *R2 Routing Table for the 10.5.5.0/24 Network*



```
R2# show ip route 10.5.5.5
Routing entry for 10.5.5.0/24
  Known via "ospf 1", distance 110, metric 3, type intra area
  Last update from 10.24.1.4 on GigabitEthernet0/2, 00:12:37 ago
  Routing Descriptor Blocks:
  * 10.24.1.4, from 10.45.1.5, 00:12:37 ago, via GigabitEthernet0/2
      Route metric is 3, traffic share count is 1
```

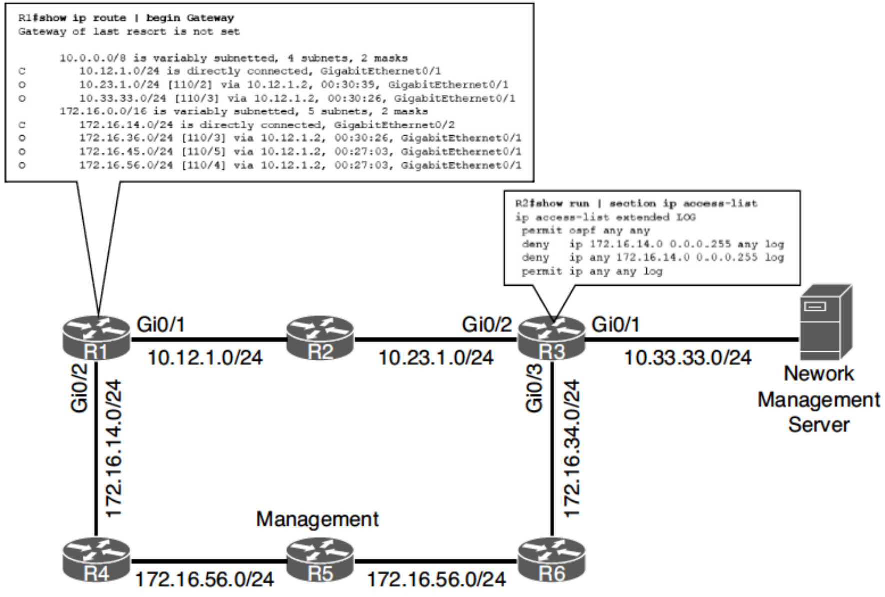
Example 15-9 shows the traffic path between the 10.1.1.0/24 and 10.5.5.0/24 networks after the conditional route forwarding policies are applied.

Example 15-10 shows that applying a PBR configuration does not modify the routing table.

Conditional Forwarding of Packets

Local PBR

Packets originated by the router are not policy routed. There is a feature for policy routing of locally generated traffic through local PBR. Local PBR policies are applied to the router with the global configuration command **ip local policy** *route-map-name*.



```
R1#show ip route | begin Gateway
Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
C        10.12.1.0/24 is directly connected, GigabitEthernet0/1
O        10.23.1.0/24 [110/2] via 10.12.1.2, 00:30:39, GigabitEthernet0/1
O        10.33.33.0/24 [110/3] via 10.12.1.2, 00:30:26, GigabitEthernet0/1
      172.16.0.0/16 is variably subnetted, 5 subnets, 2 masks
C        172.16.14.0/24 is directly connected, GigabitEthernet0/2
O        172.16.36.0/24 [110/3] via 10.12.1.2, 00:30:26, GigabitEthernet0/1
O        172.16.45.0/24 [110/5] via 10.12.1.2, 00:27:03, GigabitEthernet0/1
O        172.16.56.0/24 [110/4] via 10.12.1.2, 00:27:03, GigabitEthernet0/1
```

```
R2#show run | section ip access-list
ip access-list extended LOG
 permit ospf any any
 deny   ip 172.16.14.0 0.0.0.255 any log
 deny   ip any 172.16.14.0 0.0.0.255 log
 permit ip any any log
```

Figure 15-6 *Local PBR Topology*

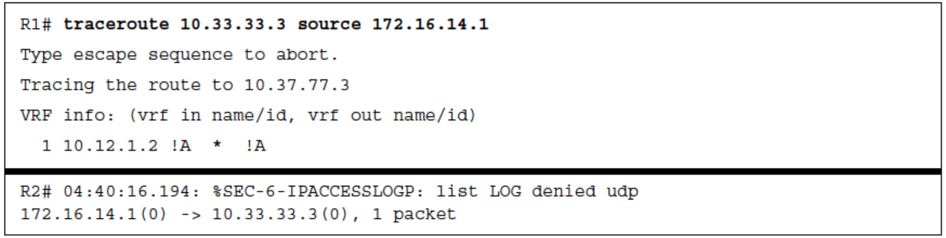
Figure 15-6 demonstrates a scenario where R1 is managed by an interface specifically for out-of-band management on the 172.16.14.0/24 network.

Conditional Forwarding of Packets

Local PBR Configuration

Example 15-11 demonstrates that R1 forwards traffic from its GigabitEthernet 0/2 interface through 10.12.1.0/24 to the network management system on the 10.33.33.0/24 network.

Example 15-11 *Traffic Is Not Sent Out of the Interface on Which It Was Received*

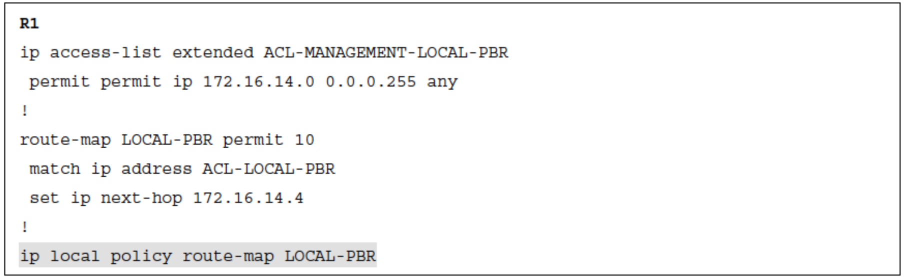


```
R1# traceroute 10.33.33.3 source 172.16.14.1
Type escape sequence to abort.
Tracing the route to 10.37.77.3
VRF info: (vrf in name/id, vrf out name/id)
  1 10.12.1.2 !A  *  !A
```

```
R2# 04:40:16.194: %SEC-6-IPACCESSLOGP: list LOG denied udp
172.16.14.1(0) -> 10.33.33.3(0), 1 packet
```

If you configure a local PBR on R1, it will only modify the next-hop IP address on traffic that is sourced locally from the 172.16.14.0/24 network by modifying the next-hop IP address to 172.16.14.4. Example 15-12 shows the local PBR configuration for R1.

Example 15-12 *Local PBR Configuration*



```
R1
ip access-list extended ACL-MANAGEMENT-LOCAL-PBR
 permit permit ip 172.16.14.0 0.0.0.255 any
!
route-map LOCAL-PBR permit 10
 match ip address ACL-LOCAL-PBR
 set ip next-hop 172.16.14.4
!
ip local policy route-map LOCAL-PBR
```

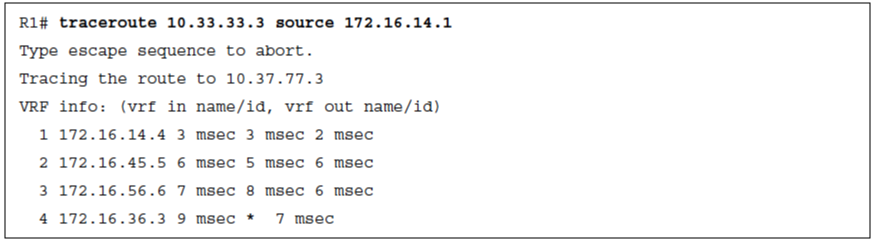
Conditional Forwarding of Packets

Local PBR Verification

With the local PBR policy placed on R1, Example 15-13 verifies that network traffic between 172.16.14.1 and 10.33.33.0/24 will use the out-of-band networks (172.16.0.0/16).

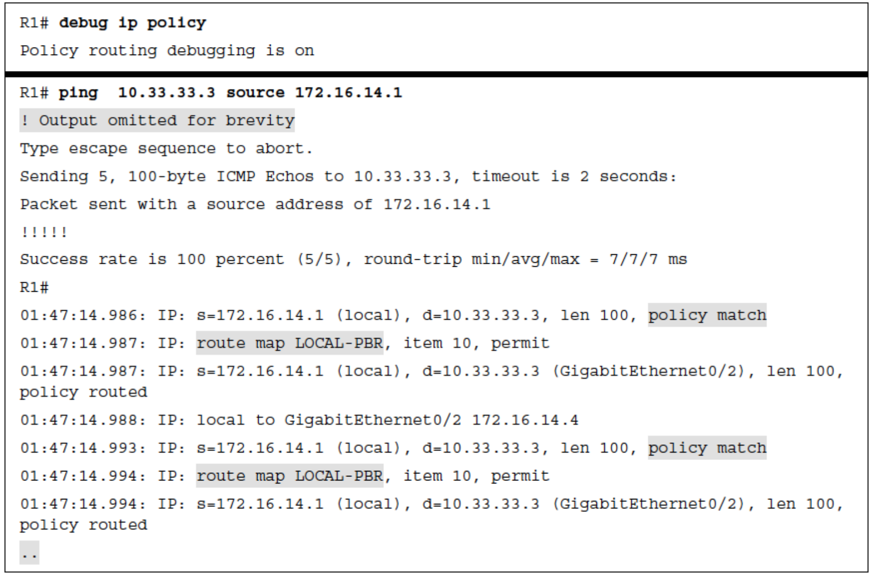
You can view policy-based decisions by enabling debugging functionality on policy-based routing with the command **debug ip policy**. Example 15-14 shows the use of the debug command and the initiation of traffic that matches the PBR to display the output.

Example 15-13 *Local PBR Verification*



```
R1# traceroute 10.33.33.3 source 172.16.14.1
Type escape sequence to abort.
Tracing the route to 10.37.77.3
VRF info: (vrf in name/id, vrf out name/id)
  1 172.16.14.4 3 msec 3 msec 2 msec
  2 172.16.45.5 6 msec 5 msec 6 msec
  3 172.16.56.6 7 msec 8 msec 6 msec
  4 172.16.36.3 9 msec *  7 msec
```

Example 15-14 *PBR Debugging*



```
R1# debug ip policy
Policy routing debugging is on
```

```
R1# ping  10.33.33.3 source 172.16.14.1
! Output omitted for brevity
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.33.33.3, timeout is 2 seconds:
Packet sent with a source address of 172.16.14.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 7/7/7 ms
R1#
01:47:14.986: IP: s=172.16.14.1 (local), d=10.33.33.3, len 100, policy match
01:47:14.987: IP: route map LOCAL-PBR, item 10, permit
01:47:14.987: IP: s=172.16.14.1 (local), d=10.33.33.3 (GigabitEthernet0/2), len 100,
policy routed
01:47:14.988: IP: local to GigabitEthernet0/2 172.16.14.4
01:47:14.993: IP: s=172.16.14.1 (local), d=10.33.33.3, len 100, policy match
01:47:14.994: IP: route map LOCAL-PBR, item 10, permit
01:47:14.994: IP: s=172.16.14.1 (local), d=10.33.33.3 (GigabitEthernet0/2), len 100,
policy routed
..
```

Trouble Tickets

- This section presents various trouble tickets relating to the topics discussed in this chapter.
- The purpose of trouble tickets is to show a process that you can follow when troubleshooting in the real world or in an exam environment.

Trouble Tickets

Trouble Ticket 15-1

Problem: Traffic from 10.1.4.0/24 to 10.1.1.0/24 is routed through R2 using Gi3/0, but it should be routed directly to R1 using Fa1/0. You begin troubleshooting by verifying the problem with a trace from a PC in 10.1.4.0/24 with the destination 10.1.1.1.

Refer to your text for next steps and examples to troubleshoot and resolve this trouble ticket.

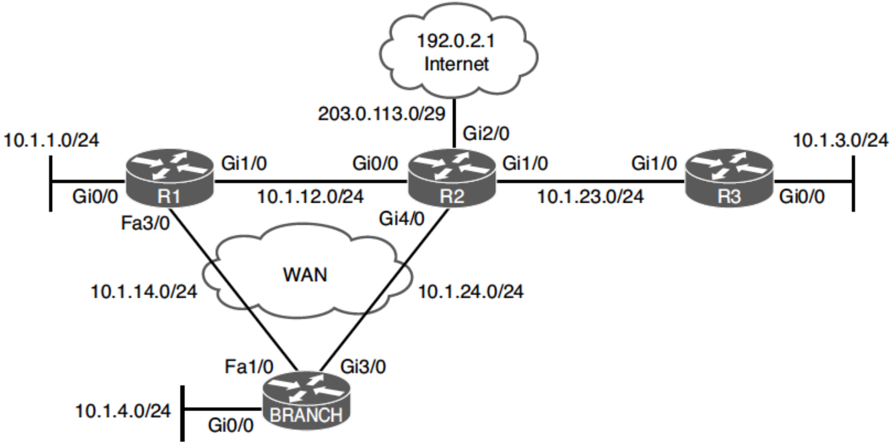


Figure 15-7 *PBR Trouble Tickets Topology*

Trouble Ticket 15-2

Problem: Traffic from 10.1.4.0/24 to 10.1.1.0/24 is routed though R2 using Gi3/0, but it should be routed directly to R1 using Fa1/0. You begin troubleshooting by verifying the problem with a trace from a PC in 10.1.4.0/24 (Branch) with the destination 10.1.1.1.

Refer to your text for next steps and examples to troubleshoot and resolve this trouble ticket.

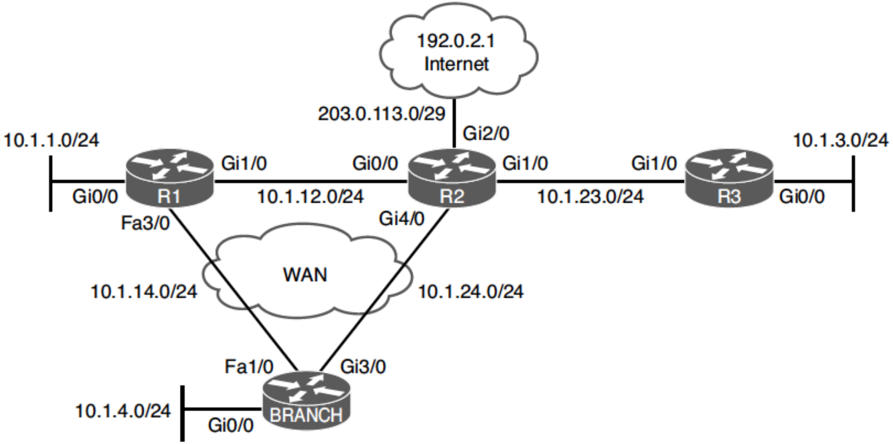


Figure 15-7 *PBR Trouble Tickets Topology*

Trouble Ticket 15-3

Problem: Traffic from 10.1.4.0/24 to 10.1.1.0/24 is routed though R2 using Gi3/0, but it should be routed directly to R1 using Fa1/0. You begin troubleshooting by verifying the problem with a trace from a PC in 10.1.4.0/24 with the destination 10.1.1.1.

Refer to your text for next steps and examples to troubleshoot and resolve this trouble ticket.

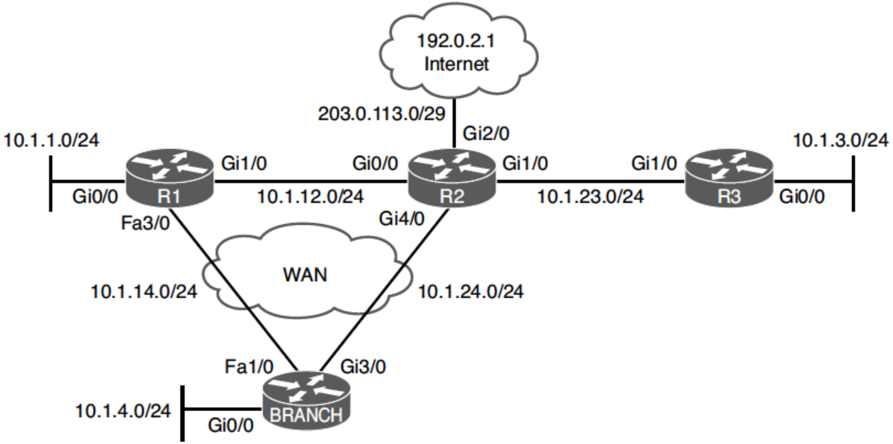


Figure 15-7 *PBR Trouble Tickets Topology*

Prepare for the Exam

Prepare for the Exam

Key Topics for Chapter 15

Description	
Access control lists (ACLs)	Route maps
Extended ACLs for IGP route selection	Conditional matching
Extended ACLs for BGP route selection	Multiple conditional match conditions
Prefix matching specification	Complex matching
Prefix match high-order bit pattern	Optional actions
Prefix match with length parameters	Policy-based routing (PBR)
Prefix lists	Local PBR
IPv6 prefix lists	

Prepare for the Exam
Key Terms for Chapter 15

Key Terms
prefix list
policy-based routing
route map

Prepare for the Exam

Command Reference for Chapter 15

Task	Command Syntax
Configure a prefix list	{**ip** | **ipv6**} **prefix-list** *prefix-list-name* [**seq** *sequence-number*] {**permit** | **deny**} *high-order-bit-pattern/high-order-bit-count* [*ge ge-value*] [*le le-value*]
Create a route map entry	**route-map** *route-map-name* [**permit** | **deny**] [*sequence-number*]
Conditionally match in a route map using AS_Path	**match as-path** *acl-number*
Conditionally match in a route map using an ACL	**match ip address** {*acl-number* | *acl-name*}
Conditionally match in a route map using a prefix list	**match ip address prefix-list** *prefix-list-name*
Conditionally match in a route map using local preference	**match local-preference** *local-preference*

Prepare for the Exam

Command Reference for Chapter 15 (Cont.)

Task	Command Syntax
Prepend the AS_Path for the network prefix with the pattern specified or from multiple iterations from neighboring autonomous systems	**set as-path prepend** {*as-number-pattern* | **last-as** *1-10*}
Set the next-hop IP address for any matching prefix	**set ip next-hop** *ip-address*
Set the BGP PA local preference	**set local-preference** *0-4294967295*
Set a numeric tag (0 through 4294967295) for identification of networks by other routers	**set tag** *tag-value*

cisco