# How to configure Policy Based Routing

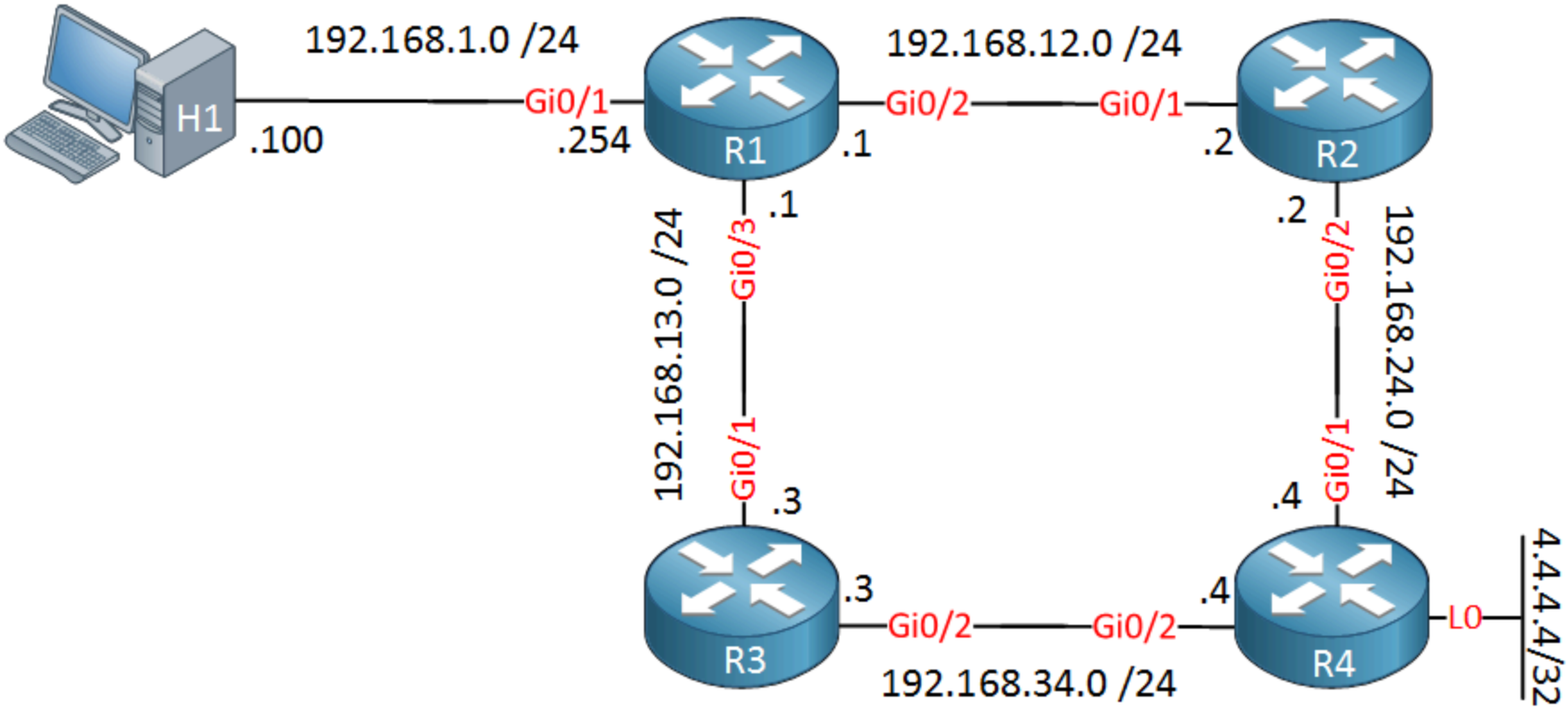
**Lesson Contents**



Policy-based routing can be used to change the next hop IP address for traffic matching certain criteria. This can be useful to overrule your routing table for certain traffic types. I will show you how to configure policy based routing.

## Configuration

here’s the topology that we will use:




Take a look at the topology picture above. OSPF is configured on all routers. Since we are using Gigabit interfaces everywhere, traffic from R1 destined to 4.4.4.4 would normally be load balanced between R2 and R3. However, I changed the cost on the Gigabit Ethernet 0/3 interface of R1 so that all traffic will go from R1 > R2 > R4.

Configurations | H1 | R1 | R2 | R3 | R4

Want to try this for yourself? Here you will find the startup configuration of each device.

Let's verify this:

```
R1#show ip ospf interface GigabitEthernet 0/2 | include Cost:
  Process ID 1, Router ID 192.168.13.1, Network Type BROADCAST, Cost: 1
```

```
R1#show ip ospf interface GigabitEthernet 0/3 | include Cost:
  Process ID 1, Router ID 192.168.13.1, Network Type BROADCAST, Cost: 1000
```

Above you can see the increased cost. Let's try a quick traceroute from H1:

```
H1#traceroute 4.4.4.4 probe 1
Type escape sequence to abort.
Tracing the route to 4.4.4.4
VRF info: (vrf in name/id, vrf out name/id)
  1 192.168.1.254 7 msec
  2 192.168.12.2 6 msec
  3 192.168.24.4 8 msec
```

Now let's say I want to use the link between R1 and R3 to reach 4.4.4.4. I could influence the metric for OSPF, but this applies to **all traffic**. What if I wanted to use this link for certain traffic only?

We could use the link between R1/R2 for the majority of our traffic and use the link between R1/R3 only for certain traffic. This can be very useful. For example, imagine that the link between R1/R3 is a dedicated link that offers QoS for VoIP traffic.

This is something we can achieve with **PBR (Policy Based Routing).** Let me show you how!

Right now, all traffic is sent toward R2:

```
R1#show ip route | include 4.4.4.4
O        4.4.4.4 [110/3] via 192.168.12.2, 00:16:48, GigabitEthernet0/2
```

Now let's say that we want all ICMP traffic from H1 destined for 4.4.4.4 to cross the link between R1/R3. Here's how to do this:

```
R1(config)#ip access-list extended ICMP_H1
R1(config-ext-nacl)#permit icmp host 192.168.1.100 host 4.4.4.4
```

First, I create an access-list that matches my traffic. Now we have to create a route-map:

```
R1(config)#route-map PBR_H1 permit 10
R1(config-route-map)#match ip address ICMP_H1
R1(config-route-map)#set ip next-hop 192.168.13.3
```

Whenever the traffic matches the access-list, we will change the next hop to 192.168.13.3 (R3).

Last but not least, let's activate it:

```
R1(config)#interface GigabitEthernet 0/1
R1(config-if)#ip policy route-map PBR_H1
```

Let's see if it works. To see it in action, I will enable a debug on R1:

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

Now let's send a ping from H1:

```
H1#ping 4.4.4.4 repeat 1
Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 4.4.4.4, timeout is 2 seconds:
!
Success rate is 100 percent (1/1), round-trip min/avg/max = 13/13/13 ms
```

The ping is working. Let's see what R1 thinks of it:

```
R1#
IP: s=192.168.1.100 (GigabitEthernet0/1), d=4.4.4.4, len 100, FIB policy match
IP: s=192.168.1.100 (GigabitEthernet0/1), d=4.4.4.4, len 100, PBR_H1 Counted
IP: s=192.168.1.100 (GigabitEthernet0/1), d=4.4.4.4, g=192.168.13.3, len 100, FIB policy routed
```

Above, you can see that it has been policy routed towards 192.168.13.3. We can also verify this by looking at the route-map:

```
R1#show route-map PBR_H1
route-map PBR_H1, permit, sequence 10
  Match clauses:
    ip address (access-lists): ICMP_H1
  Set clauses:
    ip next-hop 192.168.13.3
Nexthop tracking current: 0.0.0.0
192.168.13.3, fib_nh:0,oce:0,status:0

  Policy routing matches: 1 packets, 114 bytes
```

Let's try some traffic that doesn't match our access-list. Telnet, for example:

```
H1#telnet 4.4.4.4
Trying 4.4.4.4 ... Open
```

H1 can connect but it's not policy routed:

```
R1#
IP: s=192.168.1.100 (GigabitEthernet0/1), d=4.4.4.4, len 40, FIB policy rejected(no match) - normal forwarding
```

As you can see above, this telnet traffic is routed using the normal path.

There is one more thing I'd like to show you. With policy-based routing, there is a difference between traffic going **through the router** and traffic o**riginating from the router**.

The example above is for traffic that went through our router. What if we want to policy route traffic that originated from R1? We will have to use another command to activate it. Let's create another route-map:

```
R1(config)#ip access-list extended ICMP_R1
R1(config-ext-nacl)#permit icmp host 192.168.12.1 host 4.4.4.4
R1(config-ext-nacl)#permit icmp host 192.168.13.1 host 4.4.4.4

R1(config)#route-map PBR_R1 permit 10
R1(config-route-map)#match ip address ICMP_R1
R1(config-route-map)#set ip next-hop 192.168.13.3
```

The route-map above will redirect all traffic from R1 to 4.4.4.4 toward R3. To activate this, we need to use another command:

```
R1(config)#ip local policy route-map PBR_R1
```

This time, we need to use the **`ip local policy`** command. Let's test this:

```
R1#ping 4.4.4.4 repeat 1
Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 4.4.4.4, timeout is 2 seconds:
!
Success rate is 100 percent (1/1), round-trip min/avg/max = 19/19/19 ms
```

```
R1#
IP: s=192.168.12.1 (local), d=4.4.4.4, len 100, policy match
IP: route map PBR_R1, item 10, permit
IP: s=192.168.12.1 (local), d=4.4.4.4 (GigabitEthernet0/3), len 100, policy routed
IP: local to GigabitEthernet0/3 192.168.13.3
```

Great, our traffic from R1 is policy routed.

Configurations | H1 | R1 | R2 | R3 | R4

Want to take a look for yourself? Here you will find the final configuration of each device.

# Conclusion

Policy-based routing allows us to overrule the global routing table for traffic that matches our access-list. This can be a great way to redirect traffic for certain applications. To activate PBR, there is a difference for traffic that is going through the router or that is originated by the router:

- Use the **`ip policy`** command under the interface for traffic that is going through the router.
- Use the **`ip local policy`** command globally for traffic that originated by the router.