Experiment # 8

IP Routing: Configuring Static And Dynamic (RIP) Routing Protocol

Lab Objectives

In this Lab, you will learn, how to configure and verify IP routing with Cisco routers. The following objectives will be covered in this Lab:

- Understanding and Configuring Static Routing
- Understanding and Configuring Default Routing
- Understanding and Configuring Dynamic Routing Protocol, RIP.

Routing Basics

In this Lab, we are going to discuss the IP routing process. IP routing is the process of moving packets from one network to another network using routers.

Routers don't really care about hosts, they only care about networks and the best path to each network. The network address of the destination host is used to get packets to a network through a routed network, then the hardware address of the host is used to deliver the packet from a router to the correct destination host.

Routers route traffic to all the networks in internetwork. To be able to route packets, a router must know, at a minimum, the following:

- Destination address
- Neighbor routers from which it can learn about remote networks
- Possible routes to all remote networks
- The best route to each remote network
- How to maintain and verify routing information

The router learns about remote networks from neighbor routers or from an administrator. The router then builds a routing table that describes how to find the remote networks. If a network is directly connected, then the router already knows how to get to it. If a network isn't connected, the router must learn how to get to the remote network in two ways: by using static routing or dynamic routing.

Static Routing

Static routing occurs when you manually add routes in each router's routing table. Here is the command syntax use to add a static route to a routing table:

```
Router(config)#ip route [destination_network] [mask] [next-
hop_address or exitinterface] [administrative_distance]
[permanent]
```

This list describes each command in the string:

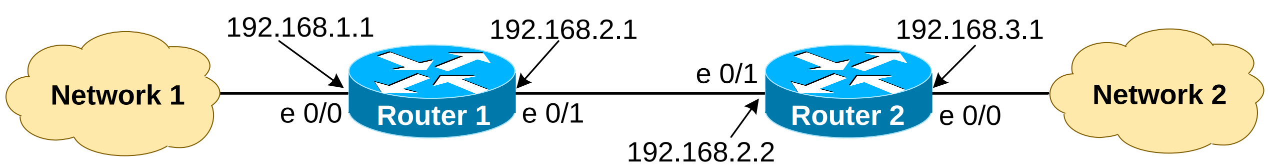
ip route The command used to create the static route.

destination_network The network you're placing in the routing table.

mask The subnet mask being used on the network.
next-hop_address The address of the next-hop router that will receive the packet and forward it to the remote network.
exitinterface You can use it in place of the next-hop address if you want, but it's got to be on a point-to-point link, such as a WAN. This command won't work on a LAN such as Ethernet.
administrative_distance By default, static routes have an administrative distance of 1. You can change the default value by adding an administrative weight at the end of the command.
permanent If the interface is shut down, or the router can't communicate to the next-hop router, the route will automatically be discarded from the routing table. Choosing the permanent option keeps the entry in the routing table no matter what happens.

Configuring Static Route using ip route

To learn, how to configure ip route, lets consider the following example. In the figure below, there are two routers shown, Router 1 and Route 2. Router don't have information about Network 2 and Router 2 don't know about Network 1. We need to configure ip route (static routes) on Router 1 for Network 2 and on Router 2 for Network 1. The subnet mask for the following example is 255.255.255.0




```
Router1(config)#ip route 192.168.3.0 255.255.255.0
192.168.2.2
```

The ip route command tells us simply that it is a static route.

- `192.168.3.0` is the remote network we want to send packets to.
- 255.255.255.0 is the mask of the remote network.
- 192.168.2.2 is the next hop, or router, we will send packets to.

For Router 2, the ip route can be configured as:

```
Router2(config)#ip route 192.168.1.0 255.255.255.0
192.168.2.1
```

The static route can be deleted by using no ip route command as:

```
Router1(config)#no ip route 192.168.3.0 255.255.255.0
192.168.2.2
```

Default Routing (Gateway of Last Resort)

We use default routing to send packets with a remote destination network not in the

routing table to the next-hop router. You can only use default routing on stub networks — those with only one exit path out of the network.
To configure a default route, you use wildcards in the network address and mask locations of a static route. In fact, you can just think of a default route as a static route that uses wildcards instead of network and mask information.
Router 1 is directly connected to network 1 (192.168.1.0) and 192.168.2.0. The routing table needs to know about network 2 (192.168.3.0). To configure the router to route to the Network 2, I placed a static routes in the routing table. By using a default route, you can just create one static route entry instead. You must first delete the existing static routes from the router and then add the default route.

```
Router1(config)#ip route 0.0.0.0 0.0.0.0 192.168.2.2
```

This above default route is also known as Gateway of Last Resort.
For Router 2, the default route can be configured as:

```
Router2(config)#ip route 0.0.0.0 0.0.0.0 192.168.2.1
```

The static route can be deleted by using no ip route command as:

```
Router1(config)#no ip route 0.0.0.0 0.0.0.0 192.168.2.2
```

There is another command you can use to configure a gateway of last resort — the ip default-network command. Here are three solutions (all providing the same solution) for adding a gateway of last resort on the router to the ISP.

```
Router(config)#ip route 0.0.0.0 0.0.0.0 192.168.2.2
Router(config)#ip route 0.0.0.0 0.0.0.0 e0/1
Router(config)#ip default-network 192.168.2.2
```

Dynamic Routing

In **dynamic routing**, a protocol on one router communicates with the same protocol running on neighbor routers. The routers then update each other about all the networks they know about and place this information into the routing table. If a change occurs in the network, the dynamic routing protocols automatically inform all routers about the event. If static routing is used, the administrator is responsible for updating all changes by hand into all routers. Typically, in a large network, a combination of both dynamic and static routing is used.

Routing Information Protocol (RIP)

Routing Information Protocol (RIP) is a true distance-vector routing protocol. It sends the complete routing table out to all active interfaces every 30 seconds. RIP only uses hop count to determine the best way to a remote network, but it has a maximum allowable hop count of 15 by default, meaning that 16 is deemed unreachable. RIP works well in small networks, but it's inefficient on large networks with slow WAN links or on networks with a large number of routers installed.
RIP version 1 uses only classful routing, which means that all devices in the network must use the same subnet mask. This is because RIP version 1 doesn't send updates with subnet mask information in tow. RIP version 2 provides something called prefix routing,

and does send subnet mask information with the route updates. This is called classless routing.

Configuring RIP Routing

To configure RIP routing, just turn on the protocol with the router rip command and tell the RIP routing protocol which networks to advertise. That's it. Let's configure our two-router internetwork (shown earlier) with RIP routing and practice that.
You can add the RIP routing protocol by using the router rip command and the network command. The network command tells the routing protocol which network to advertise.
Look at the Router 1 configuration:

```
Router1(config)#router rip
Router1(config-router)#network 192.168.1.0
Router1(config-router)#network 192.168.2.0
Router1(config-router)#exit
Router1#
```

Note the fact that you need to type in every directly connected network that you want RIP to advertise. But because they're not directly connected we're going to leave out Network 2 — it's RIP's job to find them and populate the routing table.

Look at the Router 2 configuration:

```
Router2(config)#router rip
Router2(config-router)#network 192.168.2.0
Router2(config-router)#network 192.168.3.0
Router2(config-router)#exit
Router2#
```

The RIP routing protocol can be completely removed from the router by using:

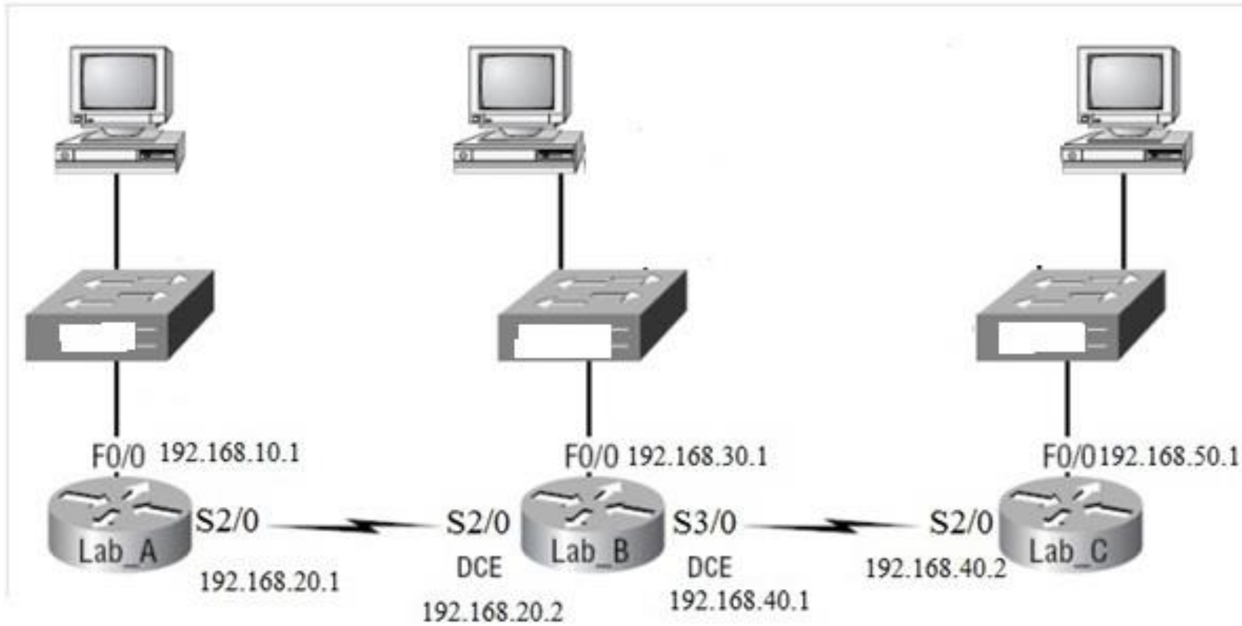
```
Router(config)#no router rip
```

If you want to remove only one entry of a network, it can be done as:

```
Router(config-router)#no network 192.168.2.0
```

Hands-on Exercise 1: Configuring Routers' Interfaces

Let's consider the following on IP Routing. Figure below shows three routers: Lab_A, Lab_B, and Lab_C, connected via a WAN. Each router also has an Ethernet network connected. The idea is that each router must know about all five networks.



Use Cisco Packet Tracer network simulator, and use Generic Router model, which has four FastEthernet ports and two serial ports.

The first step is to configure each router correctly. Table below shows the IP address scheme, which we are going to use to configure the network. After we go over how the network is configured, we will cover how to configure IP routing. Each network in the following table has a default Class C 24-bit subnet mask (255.255.255.0).

TABLE: Network Addressing for the IP Network

Router	Network Address	Interface	Address
Lab_A	192.168.10.0	fa0/0	192.168.10.1
Lab_A	192.168.20.0	S2/0	192.168.20.1
Lab_B	192.168.20.0	S2/0	192.168.20.2
Lab_B	192.168.40.0	S3/0	192.168.40.1
Lab_B	192.168.30.0	fa0/0	192.168.30.1
Lab_C	192.168.40.0	S2/0	192.168.40.2
Lab_C	192.168.50.0	fa0/0	192.168.50.1

First let’s configure the IP addresses in the network.

Lab_A Configuration

To configure the Lab_A router, you just need to add an IP address to interface FastEthernet 0/0 as well as the serial 2/0. Configuring the hostnames of each router and descriptions on each interface will make identification easier.

```
Router>en
Router#config t
Router(config)#hostname Lab_A
Lab_A(config)#interface fa 0/0
Lab_A(config-if)#ip address 192.168.10.1 255.255.255.0
Lab_A(config-if)#no shut
Lab_A(config-if)#exit
Lab_A(config)#
Lab_A(config)#interface serial 2/0
Lab_A(config-if)#ip address 192.168.20.1 255.255.255.0
Lab_A(config-if)#no shut
Lab_A(config-if)#exit
Lab_A#
```

Lab_B Configuration

It's now time to configure the next router. To configure Lab_B, we have three interfaces to deal with: FastEthernet 0/0, serial 2/0, and serial 3/0. Both serial interfaces are DCE. We will have to add the clock rate command to each interface.
Here is the configuration:

```
Router>en
Router#config t
Router(config)#hostname Lab_B
Lab_B(config)#interface fa 0/0
Lab_B(config-if)#ip address 192.168.30.1 255.255.255.0
Lab_B(config-if)#no shut
Lab_B(config-if)#exit
Lab_B(config)#
Lab_B(config)#interface serial 2/0
Lab_B(config-if)#ip address 192.168.20.2 255.255.255.0
Lab_B(config-if)#clock rate 64000
Lab_B(config-if)#no shut
Lab_B(config-if)#exit
Lab_B(config)#
Lab_B(config)#interface serial 3/0
Lab_B(config-if)#ip address 192.168.40.1 255.255.255.0
Lab_B(config-if)#clock rate 64000
Lab_B(config-if)#no shut
Lab_B(config-if)# exit
Lab_B#
```

Lab_C Configuration

The configuration of Lab_C is similar to the other two routers:

```
Router>en
Router#config t
Router(config)#hostname Lab_C
```

```
Lab_C(config)#interface fa 0/0
Lab_C(config-if)#ip address 192.168.50.1 255.255.255.0
Lab_C(config-if)#no shut
Lab_C(config-if)#exit
Lab_C(config)#
Lab_C(config)#interface serial 2/0
Lab_C(config-if)#ip address 192.168.40.2 255.255.255.0
Lab_C(config-if)#no shut
Lab_C(config-if)# exit
Lab_C#
```

Hands-on Exercise 2: Configuring Static IP Routing in Network

Configuring ip route to Lab_A

Each routing table automatically includes directly connected networks. To be able to route to all networks in the internetwork, the routing table must include information that describes where these other networks are located and how to get there.
The Lab_A router is connected to networks 192.168.10.0 and 192.168.20.0. For the Lab_A router to be able to route to all networks, the following networks have to be configured in its routing table:

- 192.168.30.0
- 192.168.40.0
- 192.168.50.0

The following router output shows the configuration of static routes on the Lab_A router and the routing table after the configuration. For the Lab_A router to find the remote networks, an entry is placed in the routing table describing the network, the mask, and where to send the packets. Notice that each static route sends the packets to 192.168.20.2, which is the Lab_A router's next hop.

```
Lab_A(config)#ip route 192.168.30.0 255.255.255.0 192.168.20.2
Lab_A(config)#ip route 192.168.40.0 255.255.255.0 192.168.20.2
Lab_A(config)#ip route 192.168.50.0 255.255.255.0 192.168.20.2
```

After the router is configured, you can type **show running-config** and **show ip route** to see the static routes:

```
Lab_A#sh ip route
[output cut]
S 192.168.50.0 [1/0] via 192.168.20.2
S 192.168.40.0 [1/0] via 192.168.20.2
S 192.168.30.0 [1/0] via 192.168.20.2
C 192.168.20.0 is directly connected, Serial 0/0
C 192.168.10.0 is directly connected, FastEthernet0/0
```

```
Lab_A#
```

Remember that if the routes don't appear in the routing table, it's because the router cannot communicate with the next-hop address you configured. You can use the permanent parameter to keep the route in the routing table even if the next-hop device can't be contacted.

The S in the routing table entries above means that the network is a static entry. The [1/0] is the administrative distance and metric, which I'll discuss below, to the remote network. Here the next-hop interface is 0, indicating that it's directly connected.

The Lab_A router now has all the information it needs to communicate with the other remote networks. However, if the Lab_B and Lab_C routers are not configured with all the same information, the packets will be discarded at Lab_B and at Lab_C. We need to fix this with static routes.

Configuring ip route to Lab_B

The Lab_B router is connected to the networks 192.168.20.0, 192.168.30.0, and 192.168.40.0. The following static routes must be configured on the Lab_B router:

- 192.168.10.0
- 192.168.50.0

Here's the configuration for the Lab_B router:

```
Lab_B(config)#ip route 192.168.10.0 255.255.255.0 192.168.20.1
Lab_B(config)#ip route 192.168.50.0 255.255.255.0 192.168.40.2
```

By looking at the routing table, you can see that the Lab_B router now understands how to find each network:

```
Lab_B#sh ip route
[output cut]
S 192.168.50.0 [1/0] via 192.168.40.2
C 192.168.40.0 is directly connected, Serial0/1
C 192.168.30.0 is directly connected, FastEthernet 0/0
C 192.168.20.0 is directly connected, Serial0/0
S 192.168.10.0 [1/0] via 192.168.20.1
Lab_B#
```

The Lab_B router now has a complete routing table. As soon as the other routers in the internetwork have all the networks in their routing table, Lab_B can communicate to all remote networks.

Configuring ip route to Lab_C

The Lab_C router is directly connected to networks 192.168.40.0 and 192.168.50.0. Three routes need to be added:

- 192.168.30.0

- 192.168.20.0
- 192.168.10.0

Here's the configuration for the Lab_C router:

```
Lab_C(config)#ip route 192.168.30.0 255.255.255.0 192.168.40.1
Lab_C(config)#ip route 192.168.20.0 255.255.255.0 192.168.40.1
Lab_C(config)#ip route 192.168.10.0 255.255.255.0 192.168.40.1
```

The following output shows the routing table on the Lab_C router:

```
Lab_C#sh ip route
[output cut]
C 192.168.50.0 is directly connected, FastEthernet0/0
C 192.168.40.0 is directly connected, Serial0/0
S 192.168.30.0 [1/0] via 192.168.40.1
S 192.168.20.0 [1/0] via 192.168.40.1
S 192.168.10.0 [1/0] via 192.168.40.1
Lab_C#
```

Lab_C now shows all the networks in the internetwork and can communicate with all routers and networks. All the routers have the correct routing table, and all the routers and hosts should be able to communicate without a problem — for now. But if you add even one more network or another router to the internetwork, you'll have to update all routers' routing tables by hand.

Verifying Your Configuration

Once all the routers' routing tables are configured, they need to be verified. The best way to do this, besides using the show ip route command, is with the Ping program. By pinging from routers Lab_A and Lab_C, the whole internetwork will be tested end-to-end.

Here is the output of a ping to network 192.168.50.0 from the Lab_A router:

```
Lab_A#ping 192.168.50.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.50.1, timeout is
2 seconds:
!!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max =
64/66/68 ms
Lab_A#
```

From router Lab_C, a ping to 192.168.10.0 will test for good IP connectivity. Here is the router output:

```
Lab_C#ping 192.168.10.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.10.1, timeout is
2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max
 = 64/67/72 ms
```

Since we can ping from end-to-end without a problem, our static route configuration was a success!

Default Routing

In the internetworking exercise used above, the only routers that are considered to be in a stub network are Lab_A and Lab_C. If you tried to put a default route on router Lab_B, packets wouldn't be forwarded to the correct networks because they have more than one interface routing to other routers. And even though router Lab_C has two connections, it doesn't have another router on the 192.168.50.0 network that needs packets sent to it. Lab_C will only send packets to 192.168.40.1, which is the serial 2/0 interface of Lab_B. Router Lab_A will only send packets to the 192.168.20.2 interface of Lab_A.

Router Lab_C is directly connected to networks 192.168.40.0 and 192.168.50.0. The routing table needs to know about networks 192.168.10.0, 192.168.20.0, and 192.168.30.0.

To configure the router to route to the other three networks, we can place three static routes in the routing table. By using a default route, you can just create one static route entry instead. You must first delete the existing static routes from the router and then add the default route.

```
Lab_C(config)#no ip route 192.168.10.0 255.255.255.0 192.168.40.1
Lab_C(config)#no ip route 192.168.20.0 255.255.255.0 192.168.40.1
Lab_C(config)#no ip route 192.168.30.0 255.255.255.0 192.168.40.1

Lab_C(config)#ip route 0.0.0.0 0.0.0.0 192.168.40.1
```

If you look at the routing table now, you'll see only the two directly connected networks plus an S*, which indicates that this entry is a candidate for a default route.

```
Lab_C#sh ip route
[output cut]
Gateway of last resort is 192.168.40.1 to network 0.0.0.0
C       192.168.50.0 is directly connected, FastEthernet0/0
C       192.168.40.0 is directly connected, Serial0/0
S*      0.0.0.0/0 [1/0] via 192.168.40.1
Lab_C#
```

Notice also in the routing table that the gateway of last resort is now set.

Hands-on Exercise 3: Configuring RIP Routing

Configuring RIP on Lab_A

To configure RIP on the Lab_A router, you need to remove the three static routes you added from the earlier exercise.

```
Lab_A(config)#no ip route 192.168.30.0 255.255.255.0 192.168.20.2
Lab_A(config)#no ip route 192.168.40.0 255.255.255.0 192.168.20.2
Lab_A(config)#no ip route 192.168.50.0 255.255.255.0 192.168.20.2
```

You can add the RIP routing protocol by using the router rip command and the network command.

```
Lab_A(config)#router rip
Lab_A(config-router)#network 192.168.10.0
Lab_A(config-router)#network 192.168.20.0
Lab_A(config-router)#exit
Lab_A#
```

Configuring RIP on Lab_B

To configure RIP on the Lab_B router, you need to remove the two static routes you added from the earlier exercise.

```
Lab_B(config)#no ip route 192.168.10.0 255.255.255.0 192.168.20.1
Lab_B(config)#no ip route 192.168.50.0 255.255.255.0 192.168.40.2
```

You can add the RIP routing protocol by using the router rip command and the network command.

```
Lab_B(config)#router rip
Lab_B(config-router)#network 192.168.20.0
Lab_B(config-router)#network 192.168.30.0
Lab_B(config-router)#network 192.168.40.0
Lab_B(config-router)#exit
Lab_B#
```

Configuring RIP on Lab_C

We've already removed the static routes on the Lab_C router because we placed a default route on it. So all that's needed here is to remove the default route from the Lab_C router. Then, you can turn on RIP routing for the two directly connected routes:

```
Lab_C(config)#no ip route 0.0.0.0 0.0.0.0 192.168.40.1
Lab_C(config)#router rip
Lab_C(config-router)#network 192.168.40.0
Lab_C(config-router)#network 192.168.50.0
Lab_C(config-router)#exit
Lab_C#
```

Hands-on Exercise 4: Verifying the RIP Routing Tables

Each routing table should now have the routers' directly connected routes as well as RIP-injected routes received from neighboring routers.
This output shows us the contents of the Lab_A routing table:

```
Lab_A#sh ip route
[output cut]
R    192.168.50.0 [120/2] via 192.168.20.2, 00:00:23,
Serial2/0
R    192.168.40.0 [120/1] via 192.168.20.2, 00:00:23,
Serial2/0
R    192.168.30.0 [120/1] via 192.168.20.2, 00:00:23,
Serial2/0
C    192.168.20.0 is directly connected, Serial2/0
C    192.168.10.0 is directly connected, FastEthernet0/0
Lab_A#
```

Looking at this, you can see that the routing table has the same entries that they had when we were using static routes, except for that R. The R means that the networks were added dynamically using the RIP routing protocol. The [120/1] is the administrative distance of the route (120) along with the number of hops to that remote network (1).
The following output displays Lab_B's routing table.

```
Lab_B#sh ip route
[output cut]
R 192.168.50.0 [120/1] via 172.16.40.2, 00:00:11, Serial3/0
C 192.168.40.0 is directly connected, Serial3/0
C 192.168.30.0 is directly connected, FastEthernet0/0
C 192.168.20.0 is directly connected, Serial2/0
R 192.168.10.0 [120/1] via 172.16.20.1, 00:00:21, Serial2/0
Lab_B#
```

Notice that here again the same networks are in the routing table and they weren't added manually. Let's check out Lab_C's routing table:

```
Lab_C#sh ip route
[output cut]
Gateway of last resort is not set
C 192.168.50.0 is directly connected, FastEthernet0/0
C 192.168.40.0 is directly connected, Serial2/0
R 192.168.30.0 [120/1] via 192.168.40.1, 00:00:04,
Serial2/0
R 192.168.20.0 [120/1] via 192.168.40.1,
00:00:26, Serial2/0 R 192.168.10.0 [120/2] via
192.168.40.1, 00:00:04, Serial2/0 Lab_C#
```