

Cisco RIP Lab

What you will learn from this lab:

- How to implement RIP on Cisco hardware.
- Effects of RIP parameters on performance.
- Distance-Vector routing protocol operation.
- Convergence time of RIP
- Small-scale network vs. Large Scale Performance

Table of Contents

Cisco RIP Lab	1
0.0 Preparation for Cisco RIP Lab	3
0.1 Introduction and Review	3
0.2 Distance-Vector vs. Link State Table	3
0.3 Classful vs. Classless Routing	3
1.0 RIP Within the Pod	5
1.1 Enabling RIP on Core Router-	5
1.2 Investigating How RIP Handles Network Changes	8

Supplementary Materials Attached:

- ✓ RFC 1058
- ✓ Cisco Command Summary Worksheet

0.0 Preparation for Cisco RIP Lab

0.1 Introduction and Review

For this lab, you will initially begin configuring RIP in your pod of three routers. Obtain proper IP addressing for the links from the lab instructor. This will be very important because after studying the performance of the protocol in a smaller area, each group's pod will be joined to the backbone to simulate a much larger network. We will then be able to study the performance on this much larger scale.

Read RFC 1058 on RIP. Specific sections are highlighted below. Focus on the methods of route selection, methods by which routers update their neighbors, metrics, and convergence.

Read the attached sheet of related Cisco RIP commands. Familiarize yourself with Cisco's nomenclature, and command sequences. Also, be sure to make the distinction between configuration commands, and monitoring commands.

Some important differences between Distance Vector and Link-State protocols are highlighted in the following table as a quick reference. As we will see, RIP is very demonstrative of all these properties.

Be sure you understand the methods by which a routing protocol looks to prevent loops. These include poison reverse updates, hold-down timers, and split horizon rules.

0.2 Distance-Vector vs. Link State Table

<i>Distance Vector</i>	<i>Link-State</i>
When a router learns about a change in the network, the router updates it's routing table with the change. The change is propagated when the router broadcasts it's periodic messages.	When a router learns about a change in the network, it updates it's link-state table and sends an update only about the changed entries in the network.
Neighboring routers incorporate the received information into their routing table (including the new info). When their next update process occurs, they broadcast this new information along with ALL other entries in their routing table.	Each router receives the update and adds it to the link-state table
The process continues until all routers converge.	The routers then run an algorithm to determine the best available path.
If there is no change in the network at a periodic interval, each router sends out it's routing table to it's neighbors anyway.	If no change occurs in the network, the routers will send updates only for those route entries that have not been updated periodically – from 30 minutes to 2 hours, depending upon protocol

0.3 Classful vs. Classless Routing

As a protocol, RIP does not handle Variable Length Subnet Masks (VLSM's), and therefore, is regarded as a classful routing protocol. Generally, a protocol that

does not support VLSM's is restricted because it does not send subnet masks along with it's routing updates. Upon receiving a packet, these type of protocols do one of the following to determine the network address:

- *if the routing update information about the same network number is configured on the receiving interface, the router applies the subnet mask that is configured on the receiving interface.*
- *if the router receives information about a network address that is not the same as the one configured on the receiving interface, it will apply the default (by class) subnet mask, which is why RIP1 networks are referred to as "classful networks". RIP1 route updates do not have a subnet mask field*
"Advanced Cisco Router Configuration", pg 166

This concept is extremely important when discussing, designing, and implementing RIP. Continuity of subnet masks throughout the network is imperative, or an extremely convoluted and difficult to troubleshoot series of symptoms arises.

1.0 RIP Within the Pod

As discussed in the lab overview, our first step is to enable RIP within each of our pods. It will provide a contained testbed for us to study the performance of RIP, and will also become a building block of a larger network.

Start on your core router (7000 series). Begin by addressing the interfaces connected to your user routers with the proper commands. A short example is listed below. All user commands are listed in bold italics (*command*)

```
LAGOS_WAN#conf t
Enter configuration commands, one per line. End with CNTL/Z.
LAGOS_WAN(config)#int s3/6
LAGOS_WAN(config-if)#ip address 172.31.50.1 255.255.255.0
LAGOS_WAN(config-if)#description **56k line to Milwaukee WAN**
LAGOS_WAN(config-if)#no shut
LAGOS_WAN(config-if)#
%LINK-3-UPDOWN: Interface Serial3/6, changed state to down
LAGOS_WAN(config-if)#exit
LAGOS_WAN(config)#exit
LAGOS_WAN#
```

Go to each of the area routers (2500 series), and repeat these steps. You should be able to ping from one end of a link to the other (but not across routers yet, we have not enabled routing).

Once complete, we have IP connectivity from each router to another. We may now begin the process of enabling routing.

1.1 Enabling RIP on Core Router-

1. Obtain access to your core router.
2. Enter configuration mode (*config term*)
3. Enter the command “*ip routing*”
4. Enter the command “*router rip*”. At this point, we are in IP RIP configuration mode, type a “?” to see available options.

Router configuration commands:

auto-summary	Enable automatic network number summarization
default-metric	Set metric of redistributed routes
distance	Define an administrative distance
distribute-list	Filter networks in routing updates
exit	Exit from routing protocol configuration mode
help	Description of the interactive help system
maximum-paths	Forward packets over multiple paths
neighbor	Specify a neighbor router
network	Enable routing on an IP network
no	Negate or set default values of a command
offset-list	Add or subtract offset from IGRP or RIP metrics
output-delay	Interpacket delay for RIP updates
passive-interface	Suppress routing updates on an interface
redistribute	Redistribute information from another routing protocol
timers	Adjust routing timers
validate-update-source	Perform sanity checks against source address of routing updates
version	Set routing protocol version

5. Use the “*network <network>*” command to enable RIP routing for our network. **Note: RIP does not support VLSM (Variable Length Subnet**

Masking), so it will pull the network address back to the Class A,B,C equivalent. In our example, entering “network 172.31.50.0” would enable RIP routing for network 172.31.0.0. RIP will recognize the subnets, but will assume they are all part of the much larger class B space.

6. Exit configuration mode. (CTRL-Z).
7. Use “show ip protocol” to verify that RIP is up and running.

```
LAGOS_WAN#show ip protocols
Routing Protocol is "rip"
  Sending updates every 30 seconds, next due in 4 seconds
  Invalid after 180 seconds, hold down 180, flushed after 240
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Redistributing: rip
  Default version control: send version 1, receive any version
  Routing for Networks:
    172.31.0.0
  Routing Information Sources:
    Gateway         Distance      Last Update
  Distance: (default is 120)
```

Protocol that is running

Interval between updates

“Invalid after” – if a route update for a route is not heard for X seconds, it’s declared invalid.

Networks RIP is operating for

8. If RIP came up correctly (output resembles above), we may move on to enabling it on the other routers in the pod.
9. Obtain access to a connected router.
10. Before entering configuration mode, enter the following commands “debug ip rip” and “debug ip rip events”.
11. For the next step, we will enable RIP on this router. However, also pay attention to the debugging output. Go ahead and enable RIP for our network, and exit configuration mode.

12. Some debugging output should appear on the screen, reflecting the IP addresses assigned to your pod:

```
RIP: sending general request on Serial0 to 255.255.255.255
RIP: sending general request on Serial1 to 255.255.255.255
RIP: received v1 update from 172.31.60.1 on Serial0
    172.31.60.0 in 1 hops
RIP: Update contains 1 routes
RIP: sending v1 update to 255.255.255.255 via Serial0 (172.31.60.2)
    subnet 172.31.70.0, metric 1
RIP: Update contains 1 routes
RIP: Update queued
RIP: Update sent via Serial0
RIP: sending v1 update to 255.255.255.255 via Serial1 (172.31.70.1)
    subnet 172.31.60.0, metric 1
RIP: Update contains 1 routes
RIP: Update queued
RIP: Update sent via Serial1
RIP: sending v1 update to 255.255.255.255 via Serial0 (172.31.60.2)
    subnet 172.31.70.0, metric 1
RIP: Update contains 1 routes
RIP: Update queued
RIP: Update sent via Serial0
RIP: sending v1 update to 255.255.255.255 via Serial1 (172.31.70.1)
    subnet 172.31.60.0, metric 1
RIP: Update contains 1 routes
RIP: Update queued
RIP: Update sent via Serial1
```

First set of updates is received from the network.

Every 30 seconds, the router generates a new update and sends it out to the network. (Remember with show ip proto, we saw update interval!)

When RIP first comes up, it sends out general requests on all interfaces

13. Disable debugging with the “undebug all” command.
14. Continue to load RIP on the last remaining router in the pod. Once you’ve completed this, verify that all routes within the pod are viewable to all routers (show ip route).

Questions-

- ✓ Who does the general request go to? Who does it go to on a point-to-point link? What about on Ethernet?
- ✓ Why do you think that RIP sends out a general request on first initialization? Can you think of any potential benefits?
- ✓ What happens if you DO NOT want to route a specific subnet of an otherwise routed classful address? Is it possible using the commands we have discussed. (The answer is no, but why?)
- ✓ Cisco has a command called "ip classless". Look up it's use and results. Why would you use this command?

1.2 Investigating How RIP Handles Network Changes

In this section, we will investigate how RIP handles a dynamic network. Verify once again that the entire network is seen from each router. Use the “show ip route” command to view the routing tables. You should see at least one RIP route on each router.

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
U - per-user static route, o - ODR

Gateway of last resort is not set

172.31.0.0/24 is subnetted, 3 subnets

```
R    172.31.50.0 [120/1] via 172.31.60.1, 00:00:07, Serial0
C    172.31.60.0 is directly connected, Serial0
C    172.31.70.0 is directly connected, Serial1
```

Code in first column is “R” for RIP.
Route via 172.31.60.1

Route last heard from

1. Obtain access to the “core” router.
2. Verify RIP parameters with “show ip protocol” command

Routing Protocol is “rip”

Sending updates every 30 seconds, next due in 14 seconds

Invalid after 180 seconds, hold down 180, flushed after 240

Outgoing update filter list for all interfaces is not set

Incoming update filter list for all interfaces is not set

Redistributing: rip

Default version control: send version 1, receive any version

Interface	Send	Recv	Key-chain
-----------	------	------	-----------

Serial3/6	1	1 2	
-----------	---	-----	--

Serial3/7	1	1 2	
-----------	---	-----	--

Routing for Networks:

172.31.0.0

Routing Information Sources:

Gateway	Distance	Last Update
172.31.60.2	120	00:05:38

Distance: (default is 120)

Update frequency

If not heard from for 3 minutes (180sec), the route is marked invalid, and held. If still not heard from after 4min (240 sec), it is flushed from the table.

Note: this means it will take 4 minutes for a router to flag a route as down!!

3. Issue the following debugging commands: “debug ip rip”, “debug ip rip events”
4. Obtain access to a “user” router.
5. Issue the same debugging commands.
6. Enter configuration mode, and remove the IP address from the interface connected between user routers in the pod. “no ip address <ip_add>”
7. Watch the debugging output on both screens. It will take 3*holdtime for the core to recognize that the link is down because it has not heard an update, so be patient! The following output has been broken up into two sections, core router, and client router:

Core:

```
RIP: sending v1 update to 255.255.255.255 via Serial3/7 (172.31.60.1)
      subnet 172.31.50.0, metric 1
RIP: Update contains 1 routes
RIP: Update queued
RIP: Update sent via Serial3/7
RIP: sending v1 update to 255.255.255.255 via Serial3/6 (172.31.50.1)
      subnet 172.31.60.0, metric 1
      subnet 172.31.70.0, metric 2
RIP: Update contains 2 routes
RIP: Update queued
RIP: Update sent via Serial3/6
```

Regular
update
traffic

4 minutes after the link has been "unaddressed"-

```
RIP: sending v1 update to 255.255.255.255 via Serial3/6 (172.31.50.1)
      subnet 172.31.60.0, metric 1
      subnet 172.31.70.0, metric 16
RIP: Update contains 2 routes
RIP: Update queued
RIP: Update sent via Serial3/6
RIP: sending v1 update to 255.255.255.255 via Serial3/7 (172.31.60.1)
      subnet 172.31.50.0, metric 1
      subnet 172.31.70.0, metric 16
RIP: Update contains 2 routes
RIP: Update queued
RIP: Update sent via Serial3/7
```

Notice the metric now applied to the downed route!

Client:

```
RIP: received v1 update from 172.31.60.1 on Serial0
      172.31.50.0 in 1 hops
RIP: Update contains 1 routes
RIP: sending v1 update to 255.255.255.255 via Serial0 (172.31.60.2) - suppressing
      null update
RIP: received v1 update from 172.31.60.1 on Serial0
      172.31.50.0 in 1 hops
RIP: Update contains 1 routes
RIP: sending v1 update to 255.255.255.255 via Serial0 (172.31.60.2) - suppressing
      null update
RIP: received v1 update from 172.31.60.1 on Serial0
      172.31.50.0 in 1 hops
      172.31.70.0 in 16 hops (inaccessible)
RIP: Update contains 2 routes
RIP: sending v1 update to 255.255.255.255 via Serial0 (172.31.60.2) - suppressing
      null update
RIP: received v1 update from 172.31.60.1 on Serial0
      172.31.50.0 in 1 hops
      172.31.70.0 in 16 hops (inaccessible)
RIP: Update contains 2 routes
RIP: sending v1 update to 255.255.255.255 via Serial0 (172.31.60.2) - suppressing
      null update
```

This example demonstrates the use of many different routing protocol methods of preventing network loops. Poison reverse is the most easily seen, but hold-down timers and split horizon also play a role in this example. Think about each process and answer the following methods.

Questions-

- ✓ Imagine you could flip a switch on a device to turn off split horizon for RIP. Before we try this experiment, we turn off split horizon on the core router. What difference would it make in the output, (i.e. what updates/information would appear or disappear as a result)?
- ✓ Suppose we could do the same thing with poison reverse. Now we turn off poison reverse on the core, in addition to turning off split horizon. The link between the client routers goes down, what is different about the client/core relationship?
- ✓ What are the potential problems with each of these theoretical situations? (You do not need to answer for each individually, just a quantitative analysis of possible behavior, a short summary of what would happen).
- ✓ We have yet to discuss hold-down timers. The role they play is slightly harder to see, but no less important. What purpose do they serve?

- ✓ RIP is being phased out of many large networks due to its slow convergence time. Calculate the maximum convergence time for our 3-node network (assuming all changes are dependent upon regular updates). Calculate the maximum convergence time for a network of 6 routers arranged linearly:

