Lab

4

**Dynamic Routing Protocols**

**(RIP, OSPF and BGP)**

What you will learn in this lab:

* How to configure the routing protocols RIP, OSPF, and BGP on a Linux PC and a Cisco router.
* How those routing protocols update the routing tables after a change in the network topology.
* How OSPF achieves a hierarchical routing scheme through the use of multiple areas.
* How to set up and route traffic between autonomous systems with BGP and how to configure a routing policy.

Updated: February 2021

**Table of Content**

[Study Material for Lab 4 3](#_Toc63550522)

[Prelab 4 4](#_Toc63550523)

[Lab 4 - Dynamic Routing Protocols 5](#_Toc63550524)

[Part 1. Configuring RIP on Cisco Routers 7](#_Toc63550525)

[Exercise 1-a. Network setup 7](#_Toc63550526)

[Exercise 1-b. Configuring RIP on Cisco routers 7](#_Toc63550527)

[Exercise 1-c. Adding another router and observing route updates 9](#_Toc63550528)

[Exercise 1-d. Configuring RIP on a Linux host 12](#_Toc63550529)

[Exercise 1-e. Convergence of RIP 15](#_Toc63550530)

[Part 2 Configuring Open Shortest Path First (OSPF) 18](#_Toc63550531)

[Exercise 2-a. Network setup 20](#_Toc63550532)

[Exercise 2-b. Configuring OSPF on Cisco routers 20](#_Toc63550533)

[Exercise 2-c. Configuring OSPF on a Linux host 23](#_Toc63550534)

[Exercise 2-d. Convergence of OSPF 25](#_Toc63550535)

[Part 3. OSPF Metrics and Traffic Engineering 26](#_Toc63550536)

[Exercise 3-a. Changing routing metrics 27](#_Toc63550537)

[Part 4. Configuring the Border Gateway Protocol (BGP) 29](#_Toc63550538)

[Exercise 4-a. Network setup and IPv4 configuration 30](#_Toc63550539)

[Exercise 4-b. eBGP configuration of Cisco routers 30](#_Toc63550540)

[Exercise 4-c. iBGP configuration 33](#_Toc63550541)

[Exercise 4-d. Routing policy (selective transit) 36](#_Toc63550542)

[Appendix: An Introduction to Quagga 39](#_Toc63550543)

[1. Starting and stopping quagga processes 39](#_Toc63550544)

[2. Configuring the routing protocol processes 40](#_Toc63550545)

[3. Typing configuration commands 41](#_Toc63550546)

# Study Material for Lab 4

1. **Distance Vector and Link State Routing Protocols**: The following article (in PDF) <https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKEwj5ofvRl4TrAhUVlHIEHZ87BPQQFjALegQIAxAB&url=https%3A%2F%2Fptgmedia.pearsoncmg.com%2Fimages%2F9781587132063%2Fsamplechapter%2F1587132060_03.pdf&usg=AOvVaw2x7ZuazBLseG6JSLySw46_> provides an introduction to dynamic routing protocols. Review your knowledge of interdomain and intradomain routing, distance vector routing, and link state routing.
2. **Quagga**: Information on the *Quagga* routing software package is found at  
   <https://www.nongnu.org/quagga/docs/quagga.html>   
   Read the documents on *zebra*, *ripd*, *ospfd*and*bgpd*.
3. **RIP**: An overview of the Routing Information Protocol (RIP) is   
   <https://en.wikipedia.org/wiki/Routing_Information_Protocol>
4. Commands to configure RIP on a Cisco router are explained at   
   <https://www.cisco.com/c/en/us/td/docs/ios/iproute_rip/command/reference/irr_book/irr_rip.html>
5. **OSPF**: Here is an overview of Open Shortest Path First (OSPF) routing protocol:  
   <https://en.wikipedia.org/wiki/Open_Shortest_Path_First>
6. Commands to configure OSPF on a Cisco router are explained at   
   <https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/iproute_ospf/configuration/xe-16/iro-xe-16-book/iro-cfg.html>
7. **BGP**: This is an overview of the Border Gateway Protocol (BGP):  
   <https://en.wikipedia.org/wiki/Border_Gateway_Protocol>
8. For an explanation of commands to configure BGP on a Cisco router go to   
   <https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/iproute_bgp/configuration/xe-16/irg-xe-16-book/configuring-a-basic-bgp-network.html>

Here are additional pointers to online course material on RIP, OSPF, and BGP:

* RIP: <https://networklessons.com/rip>
* OSPF: <https://networklessons.com/ospf>
* BGP: <https://networklessons.com/bgp>

## Prelab 4

1. List one example each of a currently used distance vector routing protocol and a link state routing protocol for IPv4. For each protocol, list the full name, the acronym of the protocol, and version number (if applicable).
2. List one example each of a currently used distance vector routing protocol and a link state routing protocol for IPv6. For each protocol, list the full name, the acronym of the protocol, and version number (if applicable).
3. Provide a list of all routing protocols supported by the software package Quagga?
4. Which role does the *zebra* process play in Quagga?
5. Describe how a Linux user accesses the processes of Quagga (zebra, ripd, ospfd, bgpd) to configure routing parameters?
6. Explain what it means to "run RIP in passive mode".
7. Explain the meaning and the purpose of "triggered updates" in RIP.
8. What is the role of a designated router in OSPF? How is a designated router determined?
9. Provide the Cisco IOS command to display the link state database of OSPF.
10. If multiple routing protocols run simultaneously on a Cisco IOS router, how can you tell which routing table entries were created by the different routing protocols.

# Lab 4 - Dynamic Routing Protocols

In the previous lab, you learned how to configure routing table entries manually. This was referred to as *static routing*. The topic of Lab 4 is *dynamic routing,* where *dynamic routing protocols* (from now on, called *routing protocols*) set the routing tables automatically without human intervention. Routers and hosts that run a routing protocol, exchange routing protocol messages related to network paths and node conditions, and use these messages to compute paths between routers and hosts.

Most routing protocols implement a shortest-path algorithm, which, for a given set of routers, determines the shortest paths between the routers. Some routing protocols allow that each network interface be assigned a *cost metric*. In this case, routing protocols compute paths with least cost. Based on the method used to compute the shortest or least-cost paths, one distinguishes *distance vector* and *link state* routing protocols. In a distance vector routing protocol, *neighboring* routers send the content of their routing tables to each other and update the routing tables based on the received routing tables. In a link state routing protocol, each router advertises the cost of each of its interfaces to *all* routers in the network. Thus, all routers have complete knowledge of the network topology, and can locally run a shortest-path (or least-cost) algorithm to determine their own routing tables.

The notion of an *autonomous system* (AS) is central to the understanding of routing protocols on the Internet. An autonomous system is a group of IP networks under the authority of a single administration, and the entire Internet is carved up into a large number of autonomous systems. Examples of autonomous systems are the campus network of a university and the backbone network of a global network service provider. Each autonomous system is assigned a globally unique identifier, called the *AS number*. On the Internet, dynamic routing within an autonomous system and between autonomous systems is handled by different types of routing protocols. A routing protocol that is concerned with routing within an autonomous system is called an *intradomain routing protocol* or *interior* *gateway protocol (IGP).* A routing protocol that determines routes between autonomous systems is called an *interdomain* *routing protocol* or *exterior gateway protocol (EGP).*

In this lab, you study two intradomain protocols, namely, the Routing Information Protocol (RIP) and the Open Shortest Path First (OSPF) protocol. Parts 1–3 of this lab deal with RIP, and Parts 4–5 are about OSPF. In Part 6-7, you are exposed to a few features of the Border Gateway Protocol (BGP), which is the interdomain routing protocol of the Internet.

For the network configurations, we start with the topology in Figure 4.1, to which we add additional PCs and routers in Part 2 (Figure 4.2) and Part 4 (Figure 4.3). For Part 6, we need a completely new topology (Figure 4.4).



Figure 4.1. Network topology for Part 1.

|  |  |  |
| --- | --- | --- |
| Cisco Router | Interface FastEthernet0/0 | Interface FastEthernet1/0 |
| Router1 | 10.0.1.1/24 | 10.0.2.1/24 |
| Router2 | 10.0.3.2/24 | 10.0.2.2/24 |
| Router3 | 10.0.3.3/24 | 10.0.4.3/24 |

Table 4.1. IPv4 addresses of Cisco routers.

|  |  |  |  |
| --- | --- | --- | --- |
| Linux PC | Interface *eth0* | Interface *eth1* | Default gateway |
| PC1 | 10.0.1.11/24 | - | 10.0.1.1 |
| PC4 | 10.0.4.44/24 | - | 10.0.4.3 |

Table 4.2. IPv4 addresses of PCs.

## Part 1. Configuring RIP on Cisco Routers

In this part of the lab, you configure the routing protocol RIP on the routers in Figure 4.1.

RIP is one of the oldest dynamic routing protocols that is still in use. RIP is an intradomain routing protocol that uses a distance vector approach to resolve the paths between routers. RIP minimizes the number of hops of each path, where the traversal of each subnet counts as a *hop*.

Two versions of RIP are in use today *RIPv2* (RIP Version 2) for IPv4 routers and *RIPng* (RIP next generation) for IPv6. In this lab, we only work with RIPv2. RIP does not work well in large networks, the convergence of routing tables after a change in the network is quite slow. RIP is very easy to configure.

Every RIP-enabled router periodically sends routing table entries to each of its neighboring routers in an update message. For each routing table entry, the router sends the destination (host IP address or network IP address) and the distance to that destination measured in hops. When a router receives an update message from a neighboring router, it updates its own routing table.

### Exercise 1-a. Network setup

1. Connect the Ethernet interfaces of the Linux PCs and the Cisco routers as shown in Figure 4.1.
2. Configure the IP addresses and default gateways of *PC1* and *PC4* as shown in Table 4.2. Refer to Lab 3 for commands to configure a default gateway. Check the configuration by displaying the IP addresses (ip addr) and the routing table (netstat -rn).

### Exercise 1-b. Configuring RIP on Cisco routers

Configure all the Cisco routers to run the routing protocol RIP. Once the configuration is completed, the Cisco routers and the PCs can issue *ping* commands to each other. Below, we give a brief overview of the basic commands used to configure RIP on a Cisco router.

|  |
| --- |
| **IOS mode: privileged EXEC**  show ip protocols  Displays parameters of the currently configured routing protocol.  debug ip rip  Enables a debugging mode where the router displays a message for each received RIP message.  no debug ip rip  Disables the debugging feature.  **IOS mode: global configuration**  router rip  Enables the routing protocol RIP and enters the router configuration mode with the following prompt:  Router1(config-router)#  You return from the router configuration mode to the global configuration mode by typing the command exit.  no router rip  Disables RIP.  **IOS mode: RIP router configuration**  version 2  Sets the RIP version to RIPv2.  network *10.0.2.0*  Associates the subnet 10.0.2.0/24 with RIP. RIP sends updates for a network only if the network address has been associated with RIP. Note that the prefix length is not entered.  no network *10.0.2.0*  Disables RIP for the specified network address.  passive-interface *Ethernet0*  Sets interface *Ethernet0* in RIP passive mode. If an interface is in passive mode, the router processes incoming RIP messages, but does not transmit RIP messages on that interface.  no passive-interface *Ethernet0*  Disables RIP passive mode on interface *Ethernet0*. This means that RIP messages are transmitted on this interface. |

1. Configure the Cisco Routers with IPv4 addresses as shown in Table 4.1 and enable the routing protocol RIP. The commands to set up *Router1* are as follows:

Router1# configure terminal

Router1(config)# no ip routing

Router1(config)# ip routing

Router1(config)# router rip

Router1(config-router)# version 2

Router1(config-router)# network 10.0.0.0

Router1(config-router)# timers basic 10 60 60 80

Router1(config)# interface FastEthernet0/0

Router1(config-if)# ip address 10.0.1.1 255.255.255.0

Router1(config-if)# no shutdown

Router1(config)# interface FastEthernet1/0

Router1(config-if)# ip address 10.0.2.1 255.255.255.0

Router1(config-if)# no shutdown

Router1(config-if)# end

Router1# clear ip route \*

The command ‘no ip routing’is used to reset all previous configurations related to IP forwarding and routing protocols (RIP, OSPF, etc.). The command ‘clear ip route \*’ deletes all entries in the routing table (except the networks of the configured interfaces).

You can check the configured IP addresses and routing protocols with the commands

Router1# show ip interface

Router1# show ip interface brief

Router1# show ip protocols

1. After the routers are configured, you can issue ping commands between any pair of routers or PCs. Try to ping *PC4* from *PC1* with the command

PC1:~$ ping 10.0.4.44

You can view the routing path by typing

PC1:~$ traceroute 10.0.4.44

1. Check the routing table at the router by typing, e.g., on *Router1*,

Router1# show ip route

Every router should have one routing table entry for each subnet. Two entries are created by the configuration of IP addresses on the two interfaces. The other entries are created by RIP. Take a snapshot of the routing table of one of the routers.

1. On *Router2*, the router in the middle, display the RIP updates with the command.

Router2# debug ip rip

From the output you can infer how frequently RIP routers send distance vector information. Record how frequently the following events occur:

* *Router2* transmits a RIP update on the same interface;
* *Router2* receives a RIP update from Router1;
* *Router2* receives a RIP update from Router3.

Take a snapshot of the output of the debug ip rip command which contains two of each of the update events.

1. Once you can successfully *ping* the IP addresses of all routers, proceed to Part 2.

Lab Question/Report

1. Include the snapshot of the routing table taken in Step 3. Indicate which routing table entries were created by RIP.
2. Answer the questions on the frequency of RIP updates from Step 4. Use the snapshot taken in Step 4 to support your answer.

### Exercise 1-c. Adding another router and observing route updates

The topology for this part, shown in Figure 4.2, adds *Router4* to the network. In this you will observe RIP exchanges as the network configuration changes and how it impacts the router routing tables.

Diagram

Description automatically generated

Figure 4.2. Network topology for Part 1, Ex 1-c to 1-f.

|  |  |  |  |
| --- | --- | --- | --- |
| Cisco Router | Interface *FastEthernet0/0* | Interface *FastEthernet1/0* | Default Gateway |
| Router4 | 10.0.1.33/24 | 10.0.3.33/24 | – |

Icon

Description automatically generatedTable 4.3. IPv4 addresses of Router4.

1. Icon

   Description automatically generatedBefore you proceed with connecting *Router4*, show the routing table at all 3 routers (1,2, and 3). Take a screenshot.
2. On *PC4,* perform a traceroute to *PC1* and repeat for *PC1* to *PC4*. Take a screenshot of the outputs.
3. The network topology is shown in Figure 4.2. Attach the interfaces of *Router4* as shown in the Figure 4.2. The *FastEthernet0/0* interface of *Router4* connects to the switch, which is connected to *PC1* and *Router1*. The *FastEthernet1/0* interface of *Router4* is connected to the switch, where *Router2* and *Router3* connect to.
4. The configuration of *PC1, PC4, Router1, Router2*, and *Router3* is as in Part 1. In particular, RIP is running on the Cisco routers.
5. Start two Wireshark sessions to capture the traffic on **both** interfaces of *Router4.*
6. Configure *Router4* with the IPv4 addresses in Table 4.3, enable IPv4 forwarding, and set up RIP forwarding as shown

Router4# configure terminal  
Router4(config)# no ip routing  
Router4(config)# ip routing   
Router4(config)# router rip  
Router4(config-router)# version 2  
Router4(config-router)# network 10.0.0.0

Router4(config-router)# timers basic 10 60 60 80

Router4(config)# interface FastEthernet0/0  
Router4(config-if)# ip address 10.0.1.33 255.255.255.0

Router4(config-if)# no shutdown  
Router4(config)# interface FastEthernet1/0  
Router4(config-if)# ip address 10.0.3.33 255.255.255.0

Router4(config-if)# no shutdown

Icon

Description automatically generatedRouter4(config-if)# end  
Router4# clear ip route \*

1. Wait 30secs or so, and then check the 4 the routing tables on the 4 routers (1-4). Take a screenshot.
2. On *PC4*, issue a traceroute to *PC1*

PC4:~$ traceroute 10.0.1.11

Icon

Description automatically generatedWith *Router4* in the configuration, we expect that the route from PC4 to PC1 will change. Since RIP performs shortest path routing, you will eventually see that the route from *PC4* to *PC1* switches to include *Router4.* Repeat the traceroute command till you see the switch in routes.

Take a screen shot of the traceroute output that shows the new route.

1. Icon

   Description automatically generatedPerform a traceroute from *PC1* to *PC4*.

PC1:~$ traceroute 10.0.4.44

Take a screen of the traceroute output.

1. By default, “redirects” are disabled on Linux hosts. We will enable redirects and see if you see a difference in the traceroute output.

PC1:~$sudo sysctl -w net.ipv4.conf.all.accept\_redirects=1

Icon

Description automatically generatedPC1:~$ traceroute 10.0.4.44

![Icon

Description automatically generated with medium confidence]()Take a screenshot of the output. You should see a difference in the hops. If not repeat a few times till you observe a difference. Display the route cache on *PC1* with the command “ip route show cache”. Take a screenshot.

1. Disable redirects on PC1 and flush the route cache and neigh table

PC1:~$sudo sysctl -w net.ipv4.conf.all.accept\_redirects=0

PC1:~$sudo ip route flush cache

PC1:~$sudo ip neigh flush all

1. Stop the Wireshark capture and save the output to answer the questions in the Lab Report.

Lab Questions/Report

1. Include the screenshots from Steps 2, 8, and 9 above and provide a brief explanation of the change you observe in the output using the Wireshark output and the routing tables of the routers before and after *Router4* was added (screenshots for Step 1 and Step 7).
   1. Were the traceroutes issued from *PC1* and *PC4* any different? What did you observe? Explain in your own words what you think is going. Is that what you expected? If not, express why and what you thought should have been the outcome.
   2. When you “enable redirects” on *PC1*, did you see a different route? Is that what you were expecting?
2. In the saved Wireshark captures, set a display filter to “rip” so that only RIPv2 messages are displayed. Explore the messages and answer the following questions:

* What type of RIPv2 messages do you see?
* What is the source IP address of RIPv2 messages?
* What is the destination IP address of RIPv2 messages? What is the significance of this address? Who are the receivers of these messages?
* How are RIP messages encapsulated? What are the port numbers?
* Most RIPv2 messages that are captured are “RIP Response” messages, but you should see at least one RIP Request message. Who sent this message and when was it sent?
* How frequently does *Router4* send a RIP Response message?
* Take a screenshot of the details of a request packet and a response packet and Include here. What do you see? Compare to the routing table of the router that sent the response packet. Explain what you see.

### Exercise 1-d. Configuring RIP on a Linux host

As an alternative to a default route on PCs, we can configure Linux hosts so that they listen to RIPv2 messages, but do not transmit them (i.e., no IP forwarding). This is achieved by running the RIP protocol in ***passive mode*** on a host.

The configuration of routing protocols on Linux PCs is done with the routing software package *Quagga*. Before starting the exercise, read the tutorial on the Quagga software package in the appendix. The tutorial focuses on the features used in the lab exercises and omits many features that are available in *Quagga*. The *Zebra* process is also required on Linux hosts to enable Quagga to be configured.

Below is a list of the most relevant commands.

|  |
| --- |
| **Zebra Process**  The zebra process must be running if you want to configure routing protocols with Quagga.  sudo service zebra start  Starts the zebra process. The zebra process must be running when you use Quagga.  sudo service zebra status  Shows the status of the zebra process.  sudo service zebra stop  Terminates the zebra process.  sudo service zebra restart  Stops the zebra process, if it is running, and then starts the process.  **Ripd Process**  The ripd process runs the RIP routing protocol. The commands to control the process are analogous to those of the zebra process.  sudo service ripd start  sudo service ripd status  sudo service ripd stop  sudo service ripd restart  telnet localhost 2602  Access the command line interface of the ripd process. This requires a password (The default password is ‘zebra’. The password is found in /etc/quagga/ripd.conf). The command line interface shows the prompt  ripd>  Once the Telnet session is established you can type configuration commands. The commands and command modes are similar to those of a Cisco router. The commands are discussed in the exercises.  Typing  ripd> exit  closes the Telnet session.  show ip rip  Displays the RIP routing table, referred to as RIP table. These are the routes that are learned through RIP.  show ip rip status  Displays the status of the RIP configuration. |

In this exercise you will configure ***passive*** RIP on *PC1*. RIP will now automatically create a routing table for *PC1*.

When RIP runs in **passive mode** on a host, it only receives and processes incoming RIP messages, but does not transmit them to neighbors. Unfortunately, a bug in the PC software setup causes it to crash when there is a configuration change in the network. Under normal circumstances, a change would result in an updated routing table at PC1. We have therefore temporarily removed that part of the exercise till we resolve this issue.

1. Continue with the configuration from the previous exercise. On *PC1*, display the IPv4 routing table with

PC1:~$ netstat –rn

Take a screenshot of the output.

1. On *PC1*, run a traceroute to *PC4* and to *Router2* with

PC1:~$ traceroute 10.0.4.44

PC1:~$ traceroute 10.0.2.2

Take a screen capture of the output.

1. On *PC1*, remove the default route by typing

PC1:~$ sudo ip route flush default

You may want to check that the default route in the routing table is removed (netstat -rn)

Start *quagga* and the *ripd* by typing

PC1:~$ sudo service zebra restart

PC1:~$ sudo service ripd restart

Connect to the ripd process via Telnet*.*

PC1:~$ telnet localhost 2602

The system will prompt you for a password. Enter the default password ‘zebra’, or the password that is in /etc/quagga/rpid.conf*.*

Issue the following commands to enable RIP in passive mode on interface *eth0*:

ripd> enableripd# configure terminalripd(config)# router ripripd(config-router)# version 2  
ripd(config-router)# network 10.0.0.0/8  
ripd(config-router)# passive-interface eth0  
ripd(config-router)# endripd# show ip rip

Here, the RIP configuration sets interface eth0 into passive mode, meaning that *PC1* listens to RIPv2 messages, but does not transmit them.

The‘show ip rip’ command displays the routing database of the RIP protocol. Pay attention to the Metric values in the table and match them with the length of the routes. Repeat the ‘show ip rip’ command a few times until you see an entry for each subnet in Figure 4.2. Then, exit the *Telnet* session with the command.

ripd# exit

1. Continue on *PC1*. View the *IPv4* routing table with the command

PC1:~$ netstat –rn

Take a screenshot of the output.

1. Perform a traceroute from *PC1* to *PC4*, and *Router2*

PC1:~$ traceroute 10.0.4.44

PC1:~$ traceroute 10.0.2.2

Take a screenshot of the output.

1. Leave the configuration in place for the next exercise 1-e.

Lab Questions/Report

1. Include the screen snapshot of the IPv4 routing tables from Step 1 and Step 4, and compare.
2. Include the screenshots of the traceroutes from Step 2 and Step 5, and compare.

### Exercise 1-e. Convergence of RIP

This exercise continues with the topology as shown in Figure 4.2 with PC1 setup as in Table 4.2.

Icon

Description automatically generatedHere you disconnect the Ethernet connection of interface *FastEthernet0/0* on a *Router4*that is on a traceroute path between *PC4* and *PC1,* and observe how RIP updates the routing tables to reflect the new topology.

1. If you still have the screenshots of the routing tables of the 4 routers saved from part 2-a, hold on to them. If not show the routing tables and screenshot the outputs.
2. On *PC1* check the routing table with “netstat -rn”. Take a screenshot.
3. On *PC4,* run a traceroute to *PC1* (10.0.1.11). You will see *Router4* on the path.
4. Issue a ping command from *PC4* to *PC1*. Do not terminate the ping command until this exercise is completed in Step 7.

PC4:~$ ping 10.0.1.11

1. On *Router4*, disable the *FastEthernet1/0* interface with the command.

Router4# **configure terminal**

Router4(config)# **interface FastEthernet1/0**

Icon

Description automatically generatedRouter4(config-if)# **shutdown**

Router4(config)# **end**

1. The ping will hang. Wait around 2mins, the ping should be successful again.
2. Icon

   Description automatically generatedStop the ping command on *PC4*.
3. To show the new route between *PC4* and *PC1* run a traceroute from *PC4* to *PC1*. Take a screenshot. Verify that *Router4* is no longer on the route.
4. On *PC1* check the routing table with “netstat -rn”. Take a screenshot. Do you see a difference.
5. Show the routing tables of the 4 routers. Screenshot the outputs. Do you see a difference?
6. On *PC1*, terminate the *zebra* and *ospfd* processes and add the default route from Table 4.2, by typing

PC1:~$ sudo service zebra stop

PC1:~$ sudo service ripd stop

1. On all Cisco routers, clean up the routing tables and disable RIP. The commands for *Router1* are:

Router1# configure terminal  
Router1(config)# no ip routing  
Router1(config)# ip routing

Router1(config)# end

 Repeat the commands on *Router2*, *Router3*, and *Router4.*

Lab Questions/Report

1. From the screenshots of the traceroutes on *PC4*, what do you observe happened?
2. Using the routing table outputs of the routers, comment on what happened after *Router4* f0/0 was shutdown.
   1. How does *Router4* now reach network 10.0.1.0?

Some extra Cisco RIP commands.

|  |
| --- |
| **IOS mode: RIP router configuration**  **Offset commands**  In RIP, by default each router increases the cost of a path by one. The offset command can be used to artificially increase the cost by a given value.  offset-list *0* in *10* *Ethernet0*  Increases the metric (hop count) of incoming RIP messages that arrive on interface *Ethernet0* by value *10*.  offset-list *0* out *10* *Ethernet0*  Increases the metric of outgoing RIP messages that are sent on interface *Ethernet0* by value *10*.  no offset-list *0* in *10* *Ethernet0*  Disables the specified offset-list command for incoming RIP messages for interface *Ethernet0*.  no offset-list *0* out *10* *Ethernet*  Disables the specified offset-list command for outgoing RIP messages.  **Hold-down timer**  The timers command can be used to manipulate four timers of the RIP protocol.  timers basic *<update>* *<invalid>* *<hold-down>* *<flush>*  Sets the values of the timers in the RIP protocol. The timers are measured in seconds:  *<update>*  – The time interval between transmissions of RIP update messages  (Default: 30 sec).  *<invalid>* – The time interval after which a route, which has not been updated,  is declared invalid (Default: 180 sec).  *<hold-down>* – Determines how long after a route has been updated as unavailable,  a router will wait before accepting a new route with a lower metric. This introduces a delay for processing incoming RIP messages with routing updates after a link failure (Default: 180 sec). Setting the hold-down timer to 0 means that the RIP process immediately accepts updates to a route.  *<flush>* – The amount of time that must pass before a route that has not been  updated is removed from the routing table (Default: 240 sec).  **Triggered update**  In RIP, a triggered update means that a router sends a RIP message with a routing update, whenever one of its routing table entries changes. The triggered update feature is controlled by setting the value of the flash-update-threshold timer. Triggered updates are disabled by setting the flash-update-threshold timer to the same value as the update timer. Assuming that the update timer is set to the default value of 30 seconds, the command to disable triggered updates is  flash-update-threshold *<time>*  Whenever the metric of a routing table changes, the router sends a RIP Response packet only if the next regularly scheduled Response messages is more than <time> seconds away. Hence, if *<time>* is set to the same value as the update timer, then triggered updates are disabled. |

## Part 2 Configuring Open Shortest Path First (OSPF)

Next, you explore the routing protocol Open Shortest Path First (OSPF). OSPF is a link state routing protocol, where each router sends information on the cost metric of its network interfaces to all other routers in the network. The information about the interfaces is sent using Link State Advertisements (LSAs). LSAs are disseminated using flooding, that is, a router sends its LSAs to all its neighbors, which, in turn, forward the LSAs to their neighbors, and so on. Each router maintains a link state database (LSDB) of all received LSAs, which provides the router with complete information about the topology of the network. A router uses its LSDB to run a shortest path algorithm that computes the best paths in the network.

Unlike distance vector routing protocols, link state routing protocols do not have convergence problems, such as count-to-infinity. This is seen as a significant advantage of link state protocols over distance vector protocols.

OSPF is the most widely used link state routing protocol on the Internet. The functionality of OSPF is rich, and the lab exercises highlight only a small portion of the OSPF protocol. The Internet Lab uses OSPF version 2 (OSPFv2). OSPF version 3 (version 3) is a more recent version, which is mostly used for routing of IPv6 traffic.

This part of the lab uses the network topology shown in Figure 4.3 below. The IPv4 addresses of the interfaces are shown in Table 4.4 and Table 4.5. In this part, the Cisco routers are set up as OSPF routers. The two host PCs, *PC1* and *PC4* are **not set up with a default route**. We will use OSPF to configure the routing table at the hosts. Here we use ***no passive mode*** when configuring OSPF in Quagga. Passive mode does not work with OSPF.

Diagram

Description automatically generated

Figure 4.3 Network Topology for Part 2.

|  |  |
| --- | --- |
|  | **Point-to-point links vs. stub networks vs. transit networks**  OSPF distinguishes between point-to-point links between routers and routers that are connected by a multi-access network (e.g., Ethernet). For multi-access networks, OSPF classifies those with a single router as *stub networks* and all others as *transit networks*. (This may be counterintuitive: OSPF labels all point-to-point links as stub networks.)  **Information on different types of link states in OSPF**  <https://networklessons.com/ospf/ospf-lsa-types-explained> |

|  |  |
| --- | --- |
|  | **Avoiding a combinatorial problem** Consider the subnet 10.0.2.0/24 in Figure 4.3, which is a transit network with three routers. If OSPF advertises the links in this subnet there should be a total of 3 LSAs, one link for each pair of routers. However, this does not scale. For a transit network with *N* routers, OSPF would require *N(N-1)/2* LSAs. To prevent this, for each transit network, OSPF elects one router as the Designated Router (DR), e.g., the router with the largest router ID. This router sends out a single *Network LSA* for the multiaccess network, which contains the list of all routers connected to this network. In addition, each router sends out a *Router LSA.* So, with *N* routers on a subnet there are at most *N+1* LSAs. Note once a DR is selected for a subnet, the DR is not changed even when a router with a larger router ID is added at a later time. |

|  |  |  |
| --- | --- | --- |
| Cisco Router | Interface Ethernet0 | Interface Ethernet1 |
| Router1 | 10.0.1.1/24 | 10.0.2.1/24 |
| Router2 | 10.0.3.2/24 | 10.0.2.2/24 |
| Router3 | 10.0.3.3/24 | 10.0.4.3/24 |
| Router4 | 10.0.1.33/24 | 10.0.3.33/24 |
| Router5 | 10.0.2.4/24 | 10.0.4.4/24 |

Table 4.4. IPv4 addresses of Cisco routers.

|  |  |  |  |
| --- | --- | --- | --- |
| Linux PC | Interface *eth0* | Interface *eth1* | Default gateway |
| PC1 | 10.0.1.11/24 | - |  |
| PC4 | 10.0.4.44/24 | - |  |

Table 4.5 IPv4 address of PCs.

### Exercise 2-a. Network setup

In this exercise we set up the network configuration as shown in Figure 4.3. This configuration is different from Figure 4.2 in that we have added *Router5 a*nd we will enable ***no passive*** OSPF on *PC1* and *PC4*

1. Connect the Ethernet interfaces of *Router5* as shown in Figure 4.3. Configure *Router5* with the following commands

Router5# configure terminal

Router5(config)# no ip routing

Router5(config)# ip routing

Router5(config)# interface Ethernet0

Router5(config-if)# ip address 10.0.2.4 255.255.255.0

Router5(config-if)# no shutdown

Router5(config)# interface Ethernet1

Router5(config-if)# ip address 10.0.4.4 255.255.255.0

Router5(config-if)# no shutdown

Router5(config-if)# end

Router5# clear ip route \*

1. Verify that the IPv4 configuration of the Cisco Routers is as shown in Table 4.3 and that *PC1* and *PC4* are configured as shown in Table 4.4
2. On both *PC1* and *PC4*, make sure there is no default route by typing

PC1:~$ sudo ip route flush default

PC4:~$ sudo ip route flush default

You may want to check that the default route in the routing table is removed (netstat -rn).

### Exercise 2-b. Configuring OSPF on Cisco routers

Here, you configure OSPF on the Cisco routers. Below we give a brief description of the basic IOS commands used to configure OSPF on a Cisco router. As usual, each command must be issued in a particular IOS command mode.

|  |
| --- |
| **IOS mode: Global configuration**  router ospf *<process-id>*  Enables an OSPF routing process. Each router can execute multiple OSPF processes. The *<process-id>* is a number that identifies the process. In this lab, only one OSPF process is started per router, where the <process-id> value is always set to one. The command enters the router configuration mode, which has the following command prompt:  **Router1(config-router)#**  no router ospf *<process-id>*  Disables the specified OSPF process.  **IOS mode: privileged EXEC**  show ip ospf  Displays general information about the OSPF configuration.  show ip route  Displays the routing table  show ip route ospf  Displays the routing table entries related to OSPF.  show ip protocols  Displays IP protocols running and active in the router. If OSPF is running, it displays details information about the process. This is a good way to determine if the commands that you have entered are correct and being effective or not.  show ip ospf database  Displays a summary of entries in the LSDB listing the router and network link states.  show ip ospf database network  Displays detailed information of the network link states in the LSDB. For each network link state, all OSPF routers on this network are listed.  show ip ospf database router  Displays detailed information of the router link states in the LSDB. For each router link state, both connected transit and stub networks are shown.  **IOS mode: Router configuration**  network *<Netaddr>* *<InvNetmask>* area *<AreaID>*  Associates a network prefix with OSPF and associates an OSPF area to the network address. The prefix is specified with an IP address *<Netaddr>* and an inverse netmask *<InvNetmask>*. For example, Netaddr=10.0.0.0 and InvNetmask=0.255.255.255 specify the network prefix 10.0.0.0/8. The AreaID is a number that associates an area with the address range. Area 0 is reserved to specify the backbone area.  Example: To run OSPF on Router1 for the address range 10.0.0.0/8 and assign it to Area 1, type  Router1(config-router)# network 10.0.0.0 0.255.255.255 area 1  no network *<Netaddr>* *<InvNetmask>* area *<AreaID>*  Disables OSPF for the specified network area.  passive-interface *<Iface>*  Sets interface *<Iface>* into passive mode. In passive mode, the router only receives and processes OSPF messages, but does not transmit OSPF messages.  no passive-interface *<Iface>*  Sets interface *<Iface>* into active mode. In active mode, the router receives and transmits OSPF messages.  router-id *<IPaddress>*  Assigns the IP address *<IPaddress>* as the router identifier (router-id) of the local OSPF router. In OSPF, the router-id is used in LSA messages to identify a router. In IOS, by default, a router selects the highest IP address as the router-id. The above command can be used to set the value explicitly |

1. Configure Router1 to run OSPF by typing

Router1# configure terminal

Router1(config)# no ip routing

Router1(config)# ip routing

Router1(config)# router ospf 1

Router1(config-router)# network 10.0.0.0 0.255.255.255 area 1

Router1(config-router)# end

The above commands (1) reset IP forwarding by disabling and then enabling it, (2) enable the OSPF routing protocol, and (3) link subnet 10.0.0.0/8 with OSPF Area 1. Since no *router-id* is specified, the highest IP address of Router1, 10.0.2.1, is used as the *router-id*. In (1), disabling IP forwarding also terminates the RIP routing process that was configured in earlier parts of the lab.

1. Display the status of the OSPF configuration with the commands:

Router1# show ip protocols

Router1# show ip ospf

1. Next, configure the other Cisco routers. The commands for configuring OSPF are the same as in Step 1. Each time you set up a new OSPF router, the LDBS at each router will grow. To follow the evolution of the LSDB, display the LSDB entries at **Router1** each time **before** you configure one of the other routers with the command

Router1# show ip ospf database

The command gives an overview of the entries in the LSDB, but does not show the details of the entries. Observe that there are two types of link states: router link states and network link states.

Observe that the link state database at *Router1* grows, as you add OSPF routers.

Take a screenshot of the LSDB on *Router1* after all Cisco routers have been **configured** as OSPF routers.

1. When all five Cisco routers have been configured, **wait a minute** for OSPF to converge, then display the details of the LSDB at *Router1* by typing

Router1# show ip ospf database

Router1# show ip ospf database network

Router1# show ip ospf database router

The second command displays details of network link states and the third command displays details of router link states.

Save the output. (Taking screenshots is an alternative, but the output is quite long.)

1. Display the routing table entries at *Router1* that were created by OSPF with

Router1# show ip route ospf

Take a screenshot.

1. In a network with OSPF routing, all routers have the same link state database. Confirm this by performing Step 4 on one of the other Cisco routers (e.g., *Router2*) and comparing the output. There is no need to save the output.



Lab Questions/Report

1. Provide the saved LSDB summary from Step 3 and the outputs from Step 4.
2. **?? Show how the information in the LSDB of *Router1* (from Step 4) can be used to construct parts of the network topology from Figure 4.3, including the Cisco routers, networks that connect two or more Cisco routers, and networks that connect to only one Cisco router.**
3. **?? Explain how *Router1* can use the information of its LSDB to create routing table entries displayed in Step 5.**

### Exercise 2-c. Configuring OSPF on a Linux host

Next you configure *non passive* OSPF on host *PC1*.

1. On *PC1* and make sure there is no default route by typing

PC1:~$ sudo ip route flush default

You may want to check that the default route in the routing table is removed (netstat -rn).

1. On *PC1,* restart the zebra and OSPF processes with

PC1:~$ sudo service zebra restart

PC1:~$ sudo service ospfd restart

Establish a *Telnet* session to the *ospfd* process using

PC1:~$ telnet localhost 2604

As password, use the default password “zebra”. If this does not work, lookup the password in the OSPF configuration file by typing

PC1:~$ sudo more /etc/quagga/ospfd.conf

1. On *PC1*, issue the following commands to enable OSPF in ***no passive*** mode on interface eth0.

ospfd> enable

ospfd# configure terminal

ospfd(config)# router ospf

ospfd(config-router)# network 10.0.0.0/8 area 1

ospfd(config-router)# **no** passive-interface eth0

ospfd(config-router)# end

ospfd# show ip ospf interface

The‘show ip ospf’ command displays the routing database of the OSPF protocol. Pay attention to the Metric values in the table and match them with the length of the routes. Repeat the ‘show ip ospf’ command a few times until you see an entry for each subnet in Figure 4.3. Then, exit the *Telnet* session with the command.

ospfd# exit

1. Repeat the above steps (Step 1 – 3) to configure ***no passive*** OSPF on *PC4*.
2. Icon

   Description automatically generatedOn *PC1’s* console view the *IPv4* routing table with the command

PC1:~$ netstat –rn

Take a screen snapshot of the output.

1. Repeat for *PC4*.
2. Icon

   Description automatically generatedPerform a traceroute from *PC1* to *PC4*

PC1:~$ traceroute 10.0.4.44

Take a screen snapshot of the output.

1. Icon

   Description automatically generatedPerform a traceroute from *PC4* to *PC1*

PC4:~$ traceroute 10.0.1.11

Take a screen snapshot of the output.

1. Leave the configuration in place for the next Exercise 4-d.
2. On *PC1*, terminate the *zebra* and *ospfd* processes with the following commands

PC1:~$ sudo service zebra stop

PC1:~$ sudo service ospfd stop

PC1:~$ sudo ip route add default via 10.0.1.1

1. Repeat Step 9 for *PC4* with its original default router10.0.4.44(typo original was 10.0.4.3).

Lab Questions/Report

1. **Confirm? Include the screenshot of the IPv4 routing table from Step 5 and Step 6. From the routing table, can you tell which router will *PC1* use to reach *PC4*. Likewise, for *PC4*, which router will it use to reach *PC1.***
   1. No, from entry 10.0.4.0 10.0.1.1 255.255.255.0 UG 0 0 0 eth0, but traceroute shows it takes 10.0.1.33
   2. No, from entry 10.0.1.0 10.0.4.3 255.255.255.0 UG 0 0 0 eth0, but from traceroute it takes 10.0.4.4
2. Include the screenshot of the traceroute from Step 7 and Step 8. Are the routes the same?
   1. Nope for PC1->PC4 its PC1->R4->R3-PC4 but for for PC4->PC1 it’s PC4->R5->R1->PC1.

### Exercise 2-d. Convergence of OSPF

In this exercise we will create a link failure by disabling *Router4* interface f0/0 and show how OSPF, like RIP, converges to find new routes to the destination. We continue with the same configuration and setup for the routers 1-5. BUT here we will set the **default routes** of *PC1* and *PC4* as shown in **Table 4.2** (i.e., for *PC1* via 10.0.1.1 and for *PC4* via 1.0.0.4.3)

1. Set the default routes for *PC1* and *PC4* as shown

PC1:~$ sudo ip route add default via 10.0.1.1

PC4:~$ sudo ip route add default via 10.0.4.3

1. Run a traceroute from *PC4* to *PC1*.

Icon

Description automatically generatedPC4:~$ traceroute 10.0.1.11

Take a screenshot of the output and save. You should see *Router4* on the path to *PC1*.

1. Issue a ping command from *PC4* to *PC1*.

PC4:~$ ping 10.0.1.11

1. On *Router4* disable interface FastEthernet0/0.

Router4# **configure terminal**

Router4(config)# **interface FastEthernet0/0**

Router4(config-if)# **shutdown**

Router4(config)# **end**

1. On *PC4* wait until the ping command is successful again.
2. Perform a traceroute from *PC4* to *PC1*Icon

   Description automatically generated.

PC4:~$ traceroute 10.0.1.11

Take a screenshot of the output. Verify that *Router4* is no longer listed on the route.

1. On *Router4* bring interface FastEthernet0/0 back up again by repeating the commands in Step 4 with “no shutdown”.

Lab Questions/Report

1. Did you notice a break in the pings from *PC4* to *PC1*? yup
2. From the screenshots of the traceroutes on *PC4,* what do you observe happened? It tried first route via R4 then it switched to route via R5.

## Part 3. OSPF Metrics and Traffic Engineering

An OSPF router uses the content of the LSDB to generate a map of the network topology in the form of a directed graph. Each directed edge in the graph is assigned a cost, which is generally referred to as *metric*. The router then runs Dijkstra’s shortest path algorithm to all known destinations and sets the routing table based on the result.

Figure 4.4 illustrates the construction of the directed graph. For the network in Figure 4.4(a) where three routers are connected by a switch, the resulting directed graph is shown in Figure 4.4(b). The metric of a directed edge is configured at the interface of the router where the edge originates. For example, the metric of the edges from *Router1* to *Router5* and from *Router1* to *Router3* is configured at the *FastEthernet0/1* interface of *Router1*.

|  |  |
| --- | --- |
| Diagram  Description automatically generated   1. Three routers connected to a switch. | A picture containing diagram  Description automatically generated   1. Directed graph. |

Figure 4.4. Construction of the directed graph.

The interfaces of OSPF routers are configured with a default metric. By changing the values of the metrics, network operators can control the path of network traffic. In practice, network operators periodically adjust the metrics of routers to achieve given objectives, e.g., load balancing or minimizing delays. This is referred to as *traffic engineering*.

|  |  |
| --- | --- |
|  | **Default metric of Cisco routers** On Cisco routers the default metric of an interface in OSPF is obtained by rounding down the ratio of 100 Mbps and the link rate of the interface. If the ratio is less than 1, the metric is set to 1.  In our Labs, we assume that FastEthernet interfaces of the Cisco routers have a rate of 100 Mbps or more, which results in a default metric of 1. (Cisco uses the interface names FastEthernet0/0, FastEthernet1/0, etc. for 100 Mbps Ethernet). |

|  |  |
| --- | --- |
| A picture containing light  Description automatically generated | **Viewing the link metrics**  To view the link metrics, use the command `show ip ospf interface’. The link metric is listed as a “cost” as shown below:  Router1#show ip ospf interface  FastEthernet1/0 is up, line protocol is up  Internet Address 10.0.2.1/24, Area 1  Process ID 1, Router ID 10.0.2.1, Network Type BROADCAST, **Cost: 1**  <snip> |

### Exercise 3-a. Changing routing metrics

1. Continue with the configuration from Part 2 , Exercise 2-e(typo 2-d), with default routes for *PC1* and *PC4* as set in Table 4.2.
2. Run a traceroute from *PC1* to *PC4* with

PC1:~$ traceroute 10.0.4.44

and a traceroute from *PC4* to *PC1* with

PC4:~$ traceroute 10.0.1.11 

Take screen captures of the output of both commands.

1. To explain the routes observed in Step 2, investigate the link metrics. Use the following command to show the link cost on a Cisco router interface(s), shown here for *Router4*

Router4# show ip ospf interface

1. Record the metrics displayed for the links of the Cisco routers.
2. Use the metric obtained in Step 4 to construct a graph of the network. Label the interfaces of the OSPF routers (Router1, Router2, Router3, Router4, Router5) with their metric. **All** **COST: 1**
3. Next, change the cost metric of the FastEthernet0/0 interface of *Router4*.

Router4(config)# int f0/0

Icon

Description automatically generatedRouter4(config-if)# ip ospf cost 100

1. After some time (10-30 seconds), all other OSPF routers should have received the change of the metric. Confirm the change of the metric by repeating the command `ip ospf database router’ for *Router4*. Screenshot the output.
2. Repeat Step 2 and take screenshots of the output of both commands.
3. **Repeat Step 3 - 5 showing the new cost metric on *Router4* and the new? network graph?.**
4. Change the metric *cost* from “100 “back to “1” on *Router4* FastEthernet0/0, using the commands in Step 6.
5. On all Cisco routers, clean up the routing tables and disable OSPF. The commands for *Router1* are

Router1# configure terminal  
Router1(config)# no ip routing  
Router1(config)# ip routing

Router1(config)# end

Repeat the commands on *Router2*, *Router3*, *Router4* and *Router5.*

Lab Questions/Report

1. Include the screenshots of the *traceroute* commands from Step 2 (before the metric was changed) and from Step 8 (after the change).
2. Include the graph from Step 5 and Step 9, and use it to explain the difference in the outcomes of the traceroute commands in Step 2 and Step 8.
   1. For traceroute pc1->pc4 no change, but for pc4->pc1 the new route via r5 is used as r4 metric as 100.

## Part 4. Configuring the Border Gateway Protocol (BGP)

The last part of this lab provides some exposure to the interdomain Border Gateway Protocol (BGP), which determines paths between autonomous systems on the Internet. The exercises in this lab only cover the basics of BGP. Essentially, you learn how to set up an autonomous system and observe BGP traffic between autonomous systems. BGP is a distance vector protocol that uses a path vector algorithm, where routers exchange full path information of a route. An important feature of BGP is that it permits to define *routing policies*, which can be used by a network to specify which type of traffic it is willing to allow to pass through it. The version of BGP used in the following exercise, is BGP version 4 (BGP-4).

The network configuration for this part is shown in Figure 4.5, and the IP configuration information is given in Tables 4.6 and 4.7. The network has four autonomous systems with AS numbers 100, 200, 300, and 400.

Diagram

Description automatically generated

Figure 4.5. Network topology for Part 4

BGP routers exchange routing information over a TCP connection. BGP routers that have a TCP connection established are called *BGP peers*, or simply *peers*, and the connection is called a *BGP session*. A BGP session between peers in different autonomous systems is said to run *external BGP (eBGP*). A BGP session between peers that are in the same autonomous system is said to run *internal BGP (iBGP).*

|  |  |  |
| --- | --- | --- |
| Cisco Router | Interface FastEthernet0/0 | Interface FastEthernet1/0 |
| Router1 | 10.0.1.1/24 | 10.0.10.1/24 |
| Router2 | 10.0.2.2/24 | 10.0.10.2/24 |
| Router3 | 10.0.3.3/24 | 10.0.20.3/24 |
| Router4 | 10.0.4.4/24 | 10.0.20.4/24 |
| Router5 | 10.0.3.33/24 | 10.0.10.33/24 |

Table 4.6. IPv4 addresses of Cisco routers.

|  |  |  |  |
| --- | --- | --- | --- |
| Linux PC | Interface *eth0* | Interface *eth1* | Default gateway |
| PC1 | 10.0.1.11/24 | - | 10.0.1.1 |
| PC2 | 10.0.2.22/24 | - | 10.0.2.2 |
| PC4 | 10.0.4.44/24 | - | 10.0.4.4 |

Table 4.7. IPv4 addresses of PCs.

### Exercise 4-a. Network setup and IPv4 configuration

The network topology is quite different from earlier parts of the Labs. If you have not stopped GNS3 and Quit after Part3, do so now. The network topology has different configurations. It is recommended that you start from scratch, that way all interfaces will be cleared and you will not have to delete routes, IP addresses and caches.

1. Connect the Ethernet interfaces of the Linux PCs and the Cisco routers as shown in Figure 4.5. The network topology is quite different from earlier parts of this lab. Many PCs and Cisco routers are directly connected without a switch. Using a switch is an alternative setup which does not impact the ability to communicate.
2. Configure *PC1, P2*, and *PC4* with the IP addresses and default gateways listed in Table 4.6.
3. Next, set up the IPv4 addresses of the Cisco routers according to Table 4.5.

The commands for *Router1* are given below. The configuration of the other Cisco routers is done accordingly.

Router1# configure terminal

Router1(config)# no ip routing  
Router1(config)# ip routing

Router1(config-router)# interface Ethernet0  
Router1(config-if)# ip address 10.0.1.1 255.255.255.0

Router1(config-if)# no shutdown  
Router1(config-if)# interface Ethernet1  
Router1(config-if)# ip address 10.0.10.1 255.255.255.0

Router1(config-if)# no shutdown  
Router1(config-if)# end

1. At this time, PCs and Cisco routers that are directly connected or are connected to the same switch should be able to *ping* each other. Verify that this is the case and correct the configuration if a *ping* fails.

### 

### Exercise 4-b. eBGP configuration of Cisco routers

Here, you configure the Cisco routers as BGP routers. You assign routers to autonomous systems and establish eBGP sessions.

Below we summarize the Cisco IOS commands that are used to enable BGP.

|  |
| --- |
| **IOS mode: global configuration**  router bgp *<ASnumber>*  Enables the BGP routing protocol, and sets the autonomous system number to *<ASnumber>*. The command enters the router configuration mode with the following prompt:  Router1(config-router)#  no router bgp *<ASnumber>*  Disables the BGP routing process.  **IOS mode: privileged EXEC**  show ip bgp  Displays the BGP routing table.  show ip bgp neighbors  Displays the neighbors, also called peers, of this BGP router.  show ip bgp paths  Displays the BGP path information in the local database.  clear ip bgp \*  Deletes BGP routing information.  **IOS mode: router configuration**  network *10.0.1.0* mask *255.255.255.0*  Specifies that subnet *10.0.1.0/24* will be advertised by the *local* BGP process.  neighbor *10.0.10.2* remote-as *2*  Adds the BGP router with IP address *10.0.10.2* of AS 2 as a neighbor to the BGP neighbor table.  timers bpg *<keepalive>* *<holdtime>*  Sets the values of the *<keepalive>* and *<holdtime>* timers of the BGP process. BGP routers exchange periodic messages to confirm that the connection between the routers is maintained. The interval between these messages is *<keepalive>* seconds (default: 60 seconds). The number of seconds that a BGP router waits for any BGP message before it decides that a connection is down (default: 180 seconds). |

1. Start a Wireshark capture on interface 0/0 and 1/0 of *Router5*.
2. Configure the Cisco routers to run BGP with the autonomous system numbers shown in Figure 4.6. The routers must know the AS number of their neighbors. Below is the configuration for *Router1. Router1* is in AS 100, and has neighbors in AS 200 and in AS 300.

Router1# configure terminal

Router1(config)# router bgp 100

Router1(config-router)# neighbor 10.0.10.2 remote-as 200

Router1(config-router)# neighbor 10.0.10.33 remote-as 300

Router1(config-router)# network 10.0.1.0 mask 255.255.255.0

Router1(config-router)# end

Router1# clear ip bgp \*

The first command starts BGP and assigns *Router1* to AS 100. The next two commands configure *Router2* in AS 200 and *Router5* in AS 300 as neighbors. The next command configures the router to advertise the prefix 10.0.1.0/24. The last command cleans up the routing table.

WHAT IS THE NETWORK FOR OTHER ROUTERS?

Configure *Router2*, *Router3*, *Router4* and *Router5* as BGP routers following the instructions above. Set up the following neighbor relationships (typo missing the network)

|  |  |
| --- | --- |
| Router | Neighbors |
| *Router2* | *Router1* (AS 100), *Router5* (AS 300) |
| *Rotuer5* | *Router1* (AS 100), *Router2* (AS 200) |
| *Router3* | *Router4* (AS 400) |
| *Router4* | *Router3*(AS 300) |

Do not set up neighbor relationships between *Router3* and *Router5*! This will be later configured as an iBGP session.

1. Once the Cisco routers have been configured, you should be able to ping *PC2* from *PC1*. On *PC1*, issue a *ping* command to *PC2*.

PC1:~$ ping -c3 10.0.2.22

You should also be able to do a ping from *PC4* to IP address 10.0.3.3 but NOT to 10.0.3.33.

PC4:~$ ping -c3 10.0.3.3

PC4:~$ ping -c3 10.0.3.33

* The ping from PC4 to 10.0.3.33 is not successful. Why?

1. Once the BGP peers have exchanged routing information, you can display the BGP routing table and other BGP information. On each Cisco router, save the output of the following commands:

Router1# **show ip route**

Router1# **show ip bgp**

Router1# **show ip bgp summary**

The first command is the usual IP routing table display. The second command shows the table of the BGP protocol, which includes information about the routing paths. The last command displays the BGP sessions to peers.

Take a screen capture of the output of these commands at *Router1*.

1. Next take a look at the BGP protocol messages that were captured by *Wireshark*. You only see BGP messages on the FastEthernet1/0 interface of *Router5*. Explore the different types of BGP messages, and try to infer their meanings. The following are the message types that you should observe:

* OPEN message,
* KEEP ALIVE message, 60 seconds peer keep alive message
* UPDATE message, update route message by giving path info.

1. Do NOT terminate the *Wireshark* captures.

At this time, AS 100, AS 200, and AS 300 can exchange traffic, as can AS 300 and AS 400, but the routing across AS 300 is not yet configured. This will be done in the remaining exercises.

Lab Questions/Report

1. Using the data from Step 5. (You can use the details of the UPDATE messages saved in Step 6).

Answer the following questions:

1. Which type of encapsulation is used for BGP messages (TCP, UDP or other)? TCP
2. Are these messages sent in regular time intervals or only when certain events happen? In the former case, what is the time interval? In the latter case, what is the event?
   * Regular 60 seconds keep alive
   * UPDATE with update route paths.

Inspect the “Path attributes” and the “Network Layer Reachability Information (NLRI)” information in the UPDATE messages that are sent and received by *Router5*.

1. Create a table for the captured UPDATE messages (from Step 6), with one row for each UPDATE message. The columns of the table are: source IP address, destination IP address, advertised network prefix (NLRI), and AS PATH.
   1. Source destination NLRI AS PATH

10.0.3.3 10.0.3.33 10.0.4.0/24 400

### Exercise 4-c. iBGP configuration

You next establish an iBGP session between *Router5* and *Router3* so that prefixes that are advertised to *Router5* are sent to *Router3*, and vice versa. As we will see, advertising the routes is not sufficient and another step must be taken. To observe the need for this additional step, the iBGP configuration is done in two phases.

**Phase 1: Enabling iBGP**

1. Make sure that the two Wireshark sessions for the traffic on *Router5* are active.
2. Take a look at the routing tables at *Router3* and *Router5*.

On *Router3*, the command is

Router3# show ip route

Repeat for *Router5*. Take screenshots of the output.

The routing tables do not have entries for some of the subnets in Figure 4.5. Note which ones are missing and explain why they are missing. ONLY ATTACHED routes.

1. On *Router3*, configure an iBGP session to *Router5* with the commands

Router3# show ip bgp

Router3# configure terminal

Router3(config)# router bgp 300

Router3(config-router)# neighbor 10.0.3.33 remote-as 300

Router3(config-router)# end

1. On *Router5*, repeat the commands but with the following parameters

Router5# show ip bgp

Router5# configure terminal

Router5(config)# router bgp 300

Router5(config-router)# neighbor 10.0.3.3 remote-as 300

Router5(config-router)# end

1. There are now three things to observe:  
   First, observe the BGP UPDATE messages that are sent between *Router3* and *Router5*.These messages contain updates where the two BGP peers send each other all of their known routes.
2. The second observation concerns the BGP routing table. On *Router3*, issue the command

Router3# show ip bgp

Take a screen snapshot of the output.

1. Repeat for *Router5*

Router5# show ip bgp

Take a screen snapshot of the output.

The BGP routing tables now have entries for subnets 10.0.1.0/24, 10.0.2.0/24, 10.0.3.0/24, and 10.0.4.0/24. The entries that have been added through the iBGP session are marked with an “*i*”. Take note of the *Next Hop* value of these entries.

1. The third observation is made when looking at the IPv4 routing table at *Router5* and *Router3*.

On Router3, type

Router3# show ip route

Repeat for *Router5*. You will observe that the routing tables have not changed from Step 2. Take screenshots of both outputs.

|  |  |
| --- | --- |
|  | **Why have the routing tables of Router5 and Router3 not changed?**  The reason is found in the BGP routing tables. In the output of the ‘show ip bgp’ command at *Router3* for subnet 10.0.1.0/24, the Next Hop entry shows *10.0.10.1.* But *Router3* does not know how to reach 10.0.10.1 since it has no match for this address in its routing table. Without a valid *Next Hop* address, *Router3* cannot create an entry for the IPv4 routing table entry for network 10.0.1.0/24. |

Lab Questions/Report

1. Include the captures of the IPv4 routing tables from Step 2 (before the configuration of the iBGP sessions).
2. Include the captures of the BGP routing tables from Step 6 and Step 7 (after the configuration of the iBGP sessions).
3. Include the captures of the IPv4 routing tables from Step 8 (after the configuration of the iBGP sessions).
4. Summarize your observations about the captured IPv4 and BGP routing tables.

**Phase 2: Resolving the Next Hop address**

This issue with iBGP that the routing tables could not be updated, highlights a difficulty of configuring BGP routing involving iBGP sessions. There are multiple solutions to this issue, all of which are found in practice. The following instructions pursue a solution that is the most straightforward for the topology in Figure 4.5.

The root cause of the missing routing table entries is as follows: *iBGP* peers forward BGP messages, which they received over an *eBGP* session to each other, without modifying them. In particular, the *NEXT\_HOP* attribute is not updated. Then, when an *iBGP* peer receives such a forwarded BGP message it may not know how to reach the node listed in the *NEXT\_HOP* attribute.

A solution is to force an iBGP peer to replace the NEXT\_HOP attribute with its own IP address (where it uses the IP address that connects to the same subnet as the other iBGP peer). The replacement is done for all UPDATE messages that are received on an eBGP session before the messages are forwarded to an iBGP peer.

The above can be accomplished with a single command. Once the commands are issued there will be IPv4 routing table updates at all BGP routers in Figure 4.5.

1. On *Router3*, configure an iBGP session to *Router5* with the commands

Router3# conf term

Router3(config)# router bgp 300

Router3(config-router)# neighbor 10.0.3.33 next-hop-self

Router3(config-router)# end

1. Repeat for *Router5*

Router5# conf term

Router5(config)# router bgp 300

Router5(config-router)# neighbor 10.0.3.3 next-hop-self

Router5(config-router)# end

1. Display the BGP routing table on both *Router3* and *Router5*. Also, display the IPv4 routing tables on both *Router3* and *Router5*.

Take screenshots of the BGP and IPv4 routing tables.

* Confirm that the Next Hop entries in the BGP routing table of *Router3* and *Router5* are all reachable.
* Confirm that *Router5* and *Router3* have IPv4 table entries for the subnets in AS 100, AS 200, and AS 400.
* Check the NEXT\_HOP attribute in the UPDATE messages that are sent by *Router5* (on interface *FastEthernet0/0*) to *Router3*. The display filter to show only captured UPDATE messages is “*bgp.type==2*”. Screenshot and save the details of the captured UPDATE traffic.

1. Icon

   Description automatically generatedObserve that *Router5* has sent UPDATE messages to *Router1* and *Router2*, which contain advertisements for the prefix 10.0.4.0/24. With these messages *Router1* and *Router2* will create entries for network 10.0.4.0/24 in the BGP routing table and in the IPv4 routing table.

To confirm this, display both tables at *Router1* and take a screen snapshot.

1. Now *PC4* can exchange messages with *PC1* and *PC4*. Confirm this by issuing ping commands from *PC4*

PC4:~$ ping -c3 10.0.1.11

PC4:~$ ping -c3 10.0.2.22

1. On *Router4*, run a *ping* command from *Router4* to *PC1* and *PC4*

Router4# ping 10.0.1.11

Router4# ping 10.0.2.22

1. Why do the *ping* commands from *Router4* to 10.0.1.11 and 10.0.2.22 fail?
2. What needs to be done to fix this?
3. Save the traffic capture of *Wireshark* but do not terminate the Wireshark sessions.

Lab Questions/Report

1. Include the BGP routing table and IPv4 routing table of *Router5* and *Router3* from Step 3. Use the tables to support the observations in the first two bullets in Step 3.
2. Include the screen captures of the BGP and IPv4 routing tables of *Router1* (Step 4) to confirm that *Router1* has received updates for AS 400 from *Router5*.
3. Provide your answers to the questions in Step 6.

### Exercise 4-d. Routing policy (selective transit)

A major task of BGP is to realize routing policies between autonomous systems. Routing policies are implemented by either rejecting certain network prefixes in received UPDATE messages or by not including certain prefixes in transmitted UPDATE messages.

In this exercise you implement a simple routing policy. Suppose AS 300 is a service provider, which has AS 100 and AS 400 as its customers. Since AS 200 is not a customer, AS 300 does not want to accept traffic to and from AS 200. This is referred to as *selective transit*, i.e., AS 300 will transport traffic between AS 100 and AS 400, but not between AS 200 and AS 400,

With the configuration from the previous exercises, all autonomous systems accept traffic and routing updates from all other autonomous systems.

1. Check that the *Wireshark* sessions for the traffic on the interfaces of *Router5* are still running.
2. On *PC1* and *PC2*, run the following *ping* commands

PC1:~$ ping 10.0.4.44

PC2:~$ ping 10.0.4.44

Let the *ping* commands run until the end of this exercise.

1. The remaining configuration is done exclusively on *Router5*.

Router5# conf term

Router5(config)# ip prefix-list MYLIST permit 10.0.1.0/24

The last command defines a prefix list that is assigned the name *“MYLIST”*. The argument ‘permit 10.0.1.0/24’ states that all traffic from 10.0.1.0/24 is permitted. Everything that is not permitted is blocked. (Note: A prefix list can be extended to contain multiple prefixes and ranges of prefixes. For example, ‘permit 0.0.0.0/0 le 32’) permits all IPv4 addresses.

1. Next, apply the prefix list to the BGP configuration of *Router5*. The commands continue as follows:

Router5(config)# router bgp 300

Router5(config-router)# neighbor 10.0.10.1 prefix-list MYLIST in

Router5(config-router)# neighbor 10.0.10.2 prefix-list MYLIST in

Router5(config)# end

Here, the list is applied to both 10.0.10.1 and 10.0.10.2. This is necessary, since both routers advertise the 10.0.2.0/24 prefix.

1. The last thing to do is to force a refresh of the BGP sessions with 10.0.10.1 and 10.0.10.2. This is done with the commands

Router5# clear ip bgp 10.0.10.1 in prefix-filter

Router5# clear ip bgp 10.0.10.2 in prefix-filter

1. Now, check the output of the ping commands started in Step 2. The ping from *PC1* to *PC4* continues, but the ping form PC2 to PC4 has stopped.
2. Analyze the impact of the configurations from Steps 3–5. Check the BGP routing table and the IPv4 routing table at *Router5* with the commands

Router5# show ip bgp

Router5# **show ip route**

You should see that the routing table entries for network 10.0.2.0/24 have disappeared.

1. Now check the UPDATE messages that are sent between *Router5, Router2 and Router1*. The most recent UPDATE messages were exchanged after the commands in Step 5. Recall that the display filter to show only captured UPDATE messages is “*bgp.type==2*”.

* Take a closer look at the UPDATE messages. Identify the UPDATE messages, where *Router3* withdraws the route to 10.0.2.0/24.

1. Since *Router3* has no longer a routing table entry for 10.0.2.0/24 it does not send traffic to this network. How about the traffic that is sent by 10.0.2.0/24 into AS 300?

* First check the routing table at *Router2* (show ip route). Verify that there are still entries for networks 10.0.3.0/24 and 10.0.4.0/24.
* Next, change the display filters of the *Wireshark* applications to “icmp and ip.addr==10.0.2.22”. This captures the traffic due to the ping command at *PC2*. The *Wireshark* on *Router5(FastEthernet1/0)* shows the *ICMP Echo Reply* messages, but the *Wireshark* on *Router5(FastEthernet0/0)* does not show any traffic from 10.0.2.22. This shows that *Router5* no longer forwards traffic from 10.0.2.0/24.

# Appendix: An Introduction to Quagga

Quagga is a network routing software suite providing implementations of Open Shortest Path First (OSPF), Routing Information Protocol (RIP), Border Gateway Protocol (BGP) and IS-IS for Linux systems. The Quagga architecture consists of a routing manager (*zebra*) that manages the routing tables of various routing protocols and communicates with the Linux kernel.

The routing processes used in this lab and the routing protocols they manage are as follows:

|  |  |
| --- | --- |
| Routing Process | Routing Protocol |
| zebra | Routing manager (not a routing protocol) |
| bgpd | BGP-4 |
| ripd | RIPv2 |
| ospfd | OSPFv2 |

To run Quagga, you must start the zebra service. You also need to start the service for each routing protocol that you configure.



### 1. Starting and stopping quagga processes

You start the *zebra* process by typing

PC1:~$ sudo service zebra start

You can verify if a *zebra* process is already running by typing

PC1:~$ sudo service zebra status

You terminate the *zebra* process with the command

PC1:~$ sudo service zebra stop

You can stop and restart the *zebra* process in a single command by typing

PC1:~$ sudo service zebra restart

To set up a routing process, you must first start the *zebra* process, and then start the routing protocol process. For example to start the process that runs the RIP routing protocol you type

PC1:~$ sudo service zebra start

PC1:~$ sudo service ripd start

As with the *zebra* process, you can query the status of the RIP process with the command `*ripd* *status’* and you can stop the process with the command `*ripd* *stop’*. When you type `*zebra* *stop’*, then all routing protocol processes are stopped as well.

For the *zebra* process and all other routing processes, there is a configuration file which is read when the process is started. The configuration files are located in the directory /etc/quagga, and have names zebra.conf, ripd.conf*,* etc. The configuration files look similar to the configuration files of Cisco IOS, and contain commands that are executed when the process is started.

### 2. Configuring the routing protocol processes

After you start the services for zebra or a routing protocol (e.g., ripd) you can configure the services. To configure any of the routing processes, establish a *Telnet* session with the TCP port number of the service process. Each service listens on a specific port for incoming requests to establish a *Telnet* session. The port numbers s are as follows:

|  |  |
| --- | --- |
| Routing Process | TCP Port number |
| zebra | 2601 |
| ripd | 2602 |
| ospfd | 2604 |
| bgpd | 2605 |

If you establish a *Telnet* session to a routing process, you are asked for a password. The password is If the password is correct, a command prompt is displayed. You can find the password in the configuration file of the routing process, e.g., /etc/quagga/ripd.conf. Since the configuration files can be accessed only by users with sudo privileges, you can list the configuration file by typing

PC1:~$ sudo more /etc/quagga/zebra.conf

To access the ripd process, that is running on PC1, from PC1 you type

PC1:~$ sudo telnet localhost 2602

This results in the following output:

labuser@PC1:~$ sudo telnet localhost 2602

Trying 127.0.0.1...

Connected to localhost.

Escape character is '^]'.

Hello, this is Quagga (version 0.99.24.1).

Copyright 1996-2005 Kunihiro Ishiguro, et al.

User Access Verification

Password: <enter password>

ripd>

At the prompt, you may type configuration commands. The *Telnet* session is terminated with the command.

ripd> exit

### 3. Typing configuration commands

Once you have established a *Telnet* session to a routing process, you can configure the routing protocol of that process. The command line interface of the routing processes emulates the IOS command line interface, that is, the processes have similar command modes as Cisco IOS, and the syntax of commands is generally the same as the corresponding commands in Cisco IOS.

For example, the following commands configure the RIP routing protocol for network 10.0.0.0/8 on a Linux PC.

ripd> enable

ripd# configure terminal

ripd(config)# router rip

ripd(config-router)# version 2

ripd(config-router)# **network 10.0.0.0/8**

ripd(config-router)# end

ripd# exit

Here, in the default configuration of the *Quagga* processes, an enable password is not required.