Lab Assignment on Unit IV: (Mandatory Assignment)

Aim: Configure RIP/OSPF/BGP using packet Tracer.

Requirements: Wireshark Packet Analyzer Tool.

Theory:

Dynamic Routing Protocols

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 for each network interface to 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 between 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. 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 (intra) and between autonomous systems (inter) 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).

SETUP FOR LAB

• This lab involves 4 routers, 4 PCs and 4 hubs.

• The goal of this lab is to observe how the dynamic routing protocols, RIP, OSPF and BGP, work. You will observe the different types of packets used by the RIP, OSPF and BGP protocols.

• In Parts 1, 2 3 and 4, you will set up the Cisco Routers as RIP routers.

o Observe how the routing tables are dynamically modified, and eventually find the shortest path to all machines.

o Add one more router (Cisco Router) to provide a better path and observe how the routing tables adapt to the change.

o “Break”the topology and observe the change again.

o Add one more router (Cisco Router) to observe the count-to-infinity problem when using the RIP protocol.

• In Part 5, you will run the same experiments with the OSPF routing protocol.

o Configure the Cisco Router and as OSPF routers.
o Observe as the OSPF protocol floods the link state information of each router across the whole network.
o Observe the modification in the link state database of the routers when there is a break in the ring topology.
• In Part 6, you will partition an autonomous system (AS) into two areas and observe how OSPF operates with intra-area routing and inter-area routing.
• In Part 7, you will build a simple network topology with three different autonomous systems (AS) and configure each of the routers to run the BGP protocol and observe the packet exchange between ASs.

PART 1. Configuring RIP on CISCO ROUTER

This lab starts with the same network topology as used in Part 5 of Lab 3. Different from Lab 3, where the routing tales were configured manually, here you run the routing protocol RIP to perform the same task. In Part 1, you configure RIP on the Cisco routers. In Part 2, you configure RIP on the Linux PCs. Figure 4.1 and Table 4.1 describe the network configuration for this part of the lab.

RIP is one of the oldest dynamic routing protocols on the Internet that is still in use. This lab uses the latest revision of RIP, RIPv2 (RIP version 2).RIP is an intradomain routing protocol that uses a distance vector approach to determine the paths between routers. RIP minimizes the number of hops on each path, where each point-to-point link or LAN constitutes a *hop*.

Each RIP-enabled router periodically sends the content of its routing table to all 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 associated prefix) and the distance to the destination measured in hops. When a router receives an update message from a neighboring router, it updates its own routing table.

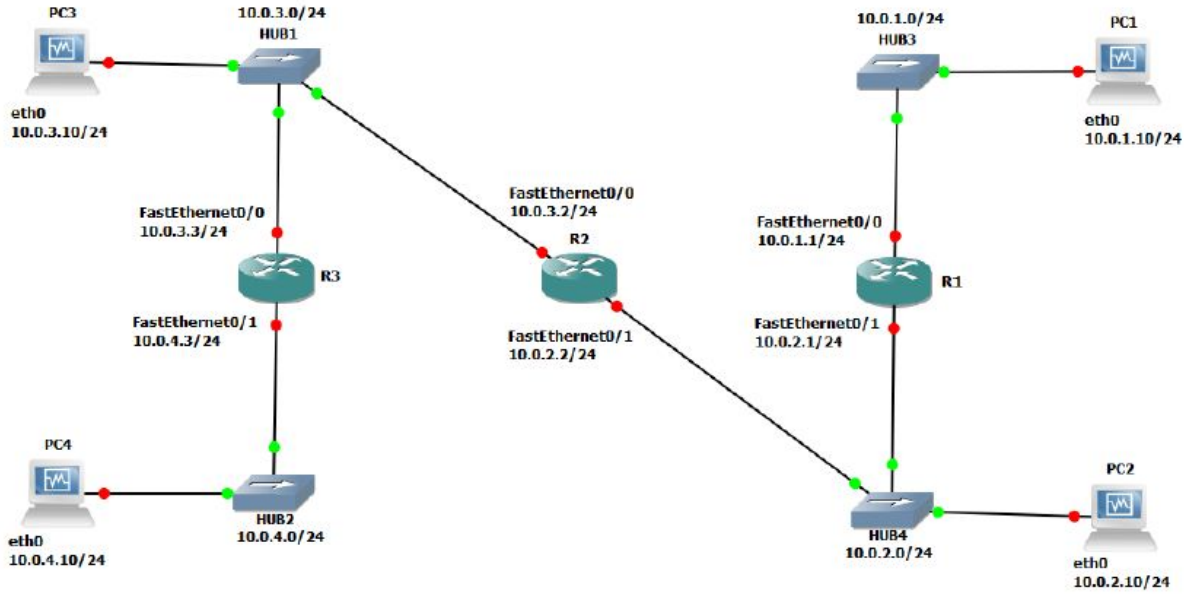

Figure 4.1 Network topology for Part 1 and Part 2

Linux PC	Ethernet Interface eth0	Ethernet Interface eth1
PC1	10.0.1.10 / 24	Disabled
PC2	10.0.2.10 / 24	Disabled
PC3	10.0.3.10 / 24	Disabled
PC4	10.0.4.10 / 24	Disabled

Cisco Routers	Ethernet Interface FastEthernet 0/0	Ethernet Interface FastEthernet 0/1
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 IP addresses of the Cisco routers and Linux PCs

PART 1. Configuring RIP on Cisco routers

In this exercise, you will configure all the routers to run RIP. After the configuration, all the routers should be able to ping all the other routers. Following is a brief overview of the basic commands used to configure RIP on a Cisco router. Make sure you type in the command in the correct command mode (note the prompt).

IOS MODE: GLOBAL CONFIGURATION

Enables or disables RIP on the local router.

IOS MODE: PRIVILEGED EXEC

Enables or disables a debugging mode where the router displays a message for each received RIP packet.

IOS MODE: ROUTER CONFIGURATION

Associates or disables the network IP addressesNetaddr with RIP. RIP sends updates only on interfaces on which the network address has been associated with RIP.

Sets or disables the interface Iface in RIP passive mode. On an interface in passive mode, the router processes incoming RIP packets but does not transmit RIP packages.

Increases the metric (hop count) of incoming RIP packages that arrive or outgoing RIP packets that are sent on interface Iface by value.

update: The time interval between transmissions of RIP update messages (default: 30 seconds).

invalid: The time interval after which a route, which has not been updated, is declared

timers basic **update invalid hold-down flush**

offset-list 0 in **valueIface** offset-list 0 out **valueIface**

passive-interface **Iface** no passive-interface **Iface**

network **Netaddr** no network **Netaddr**

debug ip rip no debug ip rip

router rip no router rip

invalid (default: 180 seconds).

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 packets with routing updates

after a link failure (default: 180 seconds).
flush: The amount of time that must pass before a route that has not been updated is removed from the routing tables (default: 240 seconds).
Sets the router not to perform triggered updates, when the next transmission of routing updates is due in time. If time is set to the same value as the update timer, then triggered update are disabled. In RIP, a triggered update means that a router sends a RIP packet with a routing update, whenever one of its routing table entries changes.
1. Connect the PCs and the Cisco Routers as shown in Figure 4.1. The PCs and routers are connected with Ethernet hubs.
2. Start Routers by clicking the right button and select Start; then, open a terminal by clicking the right button and select Console.
3. On Router1, Router2, and Router3, configure the IP addresses as shown in Table 4.1, and enable the routing protocol RIP. The commands to set up Router 1 are as follows:
flash-update-threshold **time**

Router1>enable
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)#interface FastEthernet0/0
Router1(config-if)#no shutdown
Router1(config-if)#ip address 10.0.1.1 255.255.255.0
Router1(config-if)#interface FastEthernet0/1
Router1(config-if)#no shutdown
Router1(config-if)#ip address 10.0.2.1 255.255.255.0
Router1(config-if)#end
Router1#clear ip route *
4. After you have configured the routers, check the routing table at each router with the show ip route command. Each router should have four entries in the routing table: two entries for directly connected networks and two other entries for remote networks that were added by RIP.
5. From each router, issue a ping command to the IP address of interfaces FastEthernet0/0 and FastEthernet0/1 on all remote routers. Once you can successfully contact the IP addresses of all routers, proceed to the next exercise.

PART 2. Configuring RIP on a LINUX PC
In this part of the lab, you continue with the network configuration in Figure 4.1 and Table 4.1 and configure RIP on the Linux PCs.
In Figure 4.1, all Linux PCs are set up as hosts. Since hosts do not perform IP forwarding, they need not send routing messages. Therefore, when a routing protocol is configured on a host, the protocol is set to run in *passive mode*, in which a host receives and processes incoming routing messages but not transmit routing messages. We note that normally, routing protocols

are not enabled on hosts. Instead, one generally configures a static routing table entry for the *default gateway*. Obviously, when a routing protocol is enabled, there is no need to configure a default gateway.
The configuration of routing protocols on Linux PCs in Lab 4 is done with the routing software package *Quagga*. Quagga is a software package that manages the routing tables of a Linux system and that provides the ability to execute a variety of routing protocols such as BGP, OSPF, and RIP. The process quagga must be started prior to starting and configuring any of the routing protocols. After that, the processes bgpd, ripd, and ospfd can be invoked to be configured. In Lab 1, Part 3, Exercise 3, step 4, you installed Quagga. Below we take you through the steps to configure it.

Configuring RIP on Linux Pcs with Quagga

1. For each PC be sure to configure the Quagga directory to include zebra, ripd, and ospfd conf files as follows:
Make sure the files are created by checking the directory:
The 3 files should be shown along with 2 other files named "daemons" and "debain.conf."
2. Next we will edit the daemons file to enable zebra, ospfd, and ripd. Do the following:
Once you are in vi mode for the file "daemons". Go down and change zebra, ospfd, ripd from "no" to "yes"

PC1% cp /usr/share/doc/quagga/examples/zebra.conf.sample /etc/quagga/zebra.conf
PC1% cp /usr/share/doc/quagga/examples/ospfd.conf.sample
PC1% cd /etc/quagga
PC1% ls
PC1% cd /etc/quagga
PC1% vi daemons

3. These are some commands to start, stop, and restart the process quagga.
4. Type the following command on PC1 to check the status of the process quagga
Note that the quagga process is automatically initiated with the system. You can configure each process, ripd, ospfd, or bgpd, by establishing a Telnet session to that process. Each process listens on a specific port for incoming requests to establish a Telnet session. The port numbers of the processes are given below.

Routing Protocol Routing process TCP port number

Quagga quagga 2601
RIPv1 and RIPv2 ripd 2602
OSPFv2 and OSPFv3 ospfd 2604
BGPv4+ bgpd 2605

5. Establish a Telnet session to the ripd process on the local host so that the routing protocol can be set up. You will be asked for a password, enter the password as: zebra
Instead of specifying the port number, you can telent ripd with the following command
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 to IOS, and the syntax of commands is generally the same as the corresponding command in IOS.

6. The Linux PCs, which are configured as hosts, will be set to run RIP in *passive mode*. The commands to enable RIP in passive mode are as follows:
PC1% /etc/init.d/quagga start
PC1% /etc/init.d/quagga status
PC1% telnet localhost 2602
PC1% telnet localhost ripd

a) The show ip rip command displays the routing database of the RIP protocol. This command does not exist in IOS. It may take a few minutes until RIP has built up its routing database. Screenshot the output.
b) Exit from the Telnet session. The Telnet session is terminated with the exit command.
7. On PC1, view the routing table with the command netstat –rn and screenshot the output.
8. Compare the output of netstat –rn to the output of show ip rip. Note the cost metric for each entry.
9. Repeat Steps 1-5 for the other three Linux PCs.
10. Once you can successfully ping all the Linux PCs, capture the route from PC1 to PC4 (10.0.4.10) using traceroute and screenshot the output.
11. Start to capture traffic with Wireshark on all four Linux PCs.
a) What is the destination IP address of RIP packets?
b) Do routers forward RIP packets? In other words, does PC1 receive RIP packets sent by Router3?
c) Which types of routing RIP messages do you observe? The type of a RIP message is indicated by the value of the filed *command*. For each packet type that you observed, explain the role that this message type plays in the RIP protocol.
d) A RIP message may contain multiple routing table entries. How many bytes are consumed in a RIP message for each routing table entry? Which information is transmitted for each message?
12. Stop the traffic on Wireshark and save the content of those RIP messages.
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)# passive-interface eth0
ripd(config-router)# end
ripd# show ip rip
PC1% traceroute 10.0.4.10

PART 3. Reconfiguring The Topology in RIP
In Part 3, you will add Router4 to the network topology of Figure 4.1. The configuration of the network with Router4 is illustrated in Figure 4.2. The IP configuration of Router4 is given in Table 4.2. The purpose of this exercise is to explore how RIP detects changes to the network topology and how long it takes until RIP updates the routing tables.
You will observe how the routing tables reorganize themselves after this change in the topology.

Also you will measure approximately how much time it takes RIP to alter the routing table to reflect the new topology.+

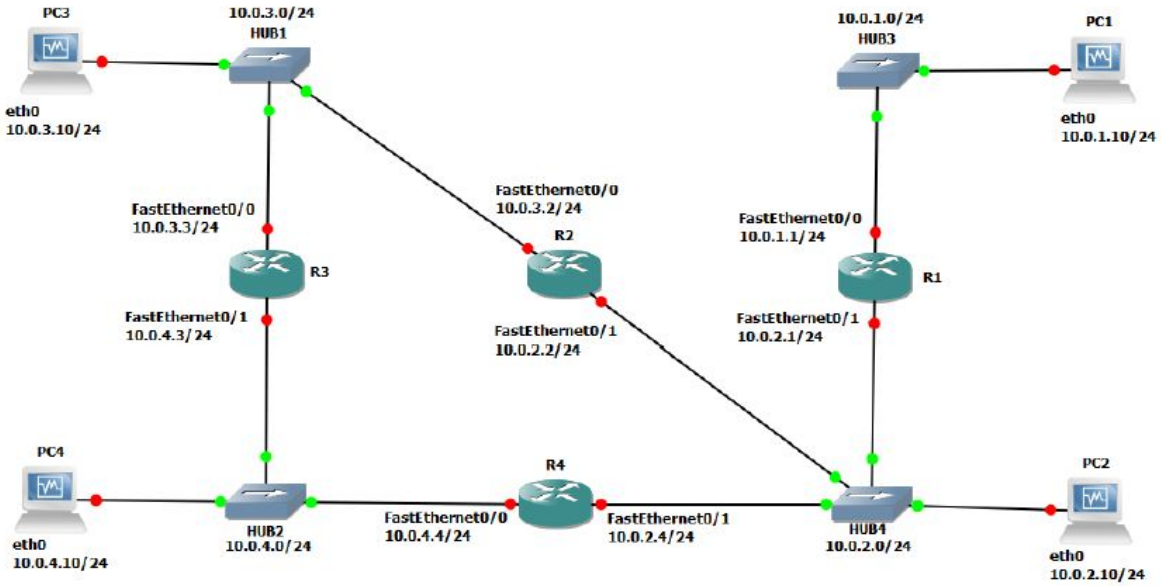

Figure 4.2 Network topology for Part 3

Cisco Router Interface FastEthernet 0/0 Interface FastEthernet 0/1
Router4 10.0.4.4 / 24 10.0.2.4 / 24
Table 4.2 IP addresses of Router4 for Part 3

Exercise 3(A). Updating the routing tables

Add Router4 to the network and observe the routing table updates made by RIP to reflect the new topology.

1. Continue with the network configuration of Part 2. RIP must be enabled on all routers shown in Figure 4.1, and a RIP process must be running (in passive mode) on all Linux PCs.

NOTE: **Do not add Router4 yet to the configuration.**

2. Before attaching Router4, screenshot the routing tables on all four Linux PCs with the command netstat –rn. Also, use the command show ip route on all Routers and save the output.
3. Connect Router4 as shown in Figure 4.2 and assign the IP addresses to the interfaces as shown in Table 4.2.
4. Configure Router 4 to run RIP, following the same steps as in Part 1.
5. Use the command show ip route on the Routers and the command netstat –rn on the Linux PCs to observe how the routing tables are updated. Once the routing tables on the PCs have converged, screenshot the routing tables on all four Linux PCs and save the output of the all four Routers.

Exercise 3(B). Convergence of RIP after a link failure

In this exercise you disconnect the Ethernet cable of interface FastEthernet0/0 on Router4 and observe how much time RIP takes to update the routing table of the Routers and PCs to reflect the new topology.

1. Issue a ping command from PC4 to PC1. Do not terminate the ping command until this

exercise is completed in Step 4.
2. To disconnect the Ethernet cable (remove a link) connected to interface FastEthernet0/0 on Router4 in GNS3, you need to shutdown the interface. Removing the link itself will not work. Now the output of the ping on PC4 should show that the destination network is unreachable.
3. Wait until the ping command is successful again, that is, ICMP Echo Reply messages arrive at PC4. This occurs once an alternate path has been found between PC4 and PC1 and the routing tables have been updated accordingly. This may take several minutes.
4. Stop the ping command with Ctrl-C and screenshot the ping statistics output (i.e, the data that appears at the bottom of the terminal screen when you stop the ping process).
5. Reconnect the interface FastEthernet0/0with the previous configuration on Router4 to proceed to Part 4.

PART 4. COUNT-TO-INFINITY Problem in RIP

Distance vector routing protocols, such as RIP, are susceptible to a convergence problem known as the *count-to-infinity* problem. This problem is a consequence of the fact that distance vector routing protocols exchange routing information only with their neighbors. Here, it may happen that, after the failure of a link, information about routes that use the failed link are propagated a long time after the failure has occurred. This results in a slow convergence of the routing tables. Each time the router exchange RIP packets, the cost of a path that uses the failed link increases, but it takes a long time until all routers realize that reputes through the failed link are unavailable.
The goal of this part of the lab is to observe the count-to-infinity problem. RIP has a number of protocol features that try to avoid the count-to-infinity problem. These fetters will be disabled. Still, since the count-to-infinity problem requires that routing updates occur in a certain order, the count-to-infinity problem is not always observable.
The network configuration is shown in Figure 4.3. Different from the network in Figure 4.2, PC3 is reconfigured and set up as an active RIP router. After the routing tables have converged, you disable the eth1interface on PC3. Following the upcoming steps, this may trigger the occurrence of the count-to-infinity problem.

Exercise 4(A). Adding PC3 to the network configuration

In this exercise, you will configure PC3 as a RIP router, using the Quagga software package. The steps to configure PC3 are almost the same as those to configure RIP on a Cisco router.
1. Connect the interface eth1 of PC3 with an Ethernet cable as shown in Figure 4.3. After connecting PC3, assign the IP addresses to interface eth1 of PC3 as indicated in Table 4.3.

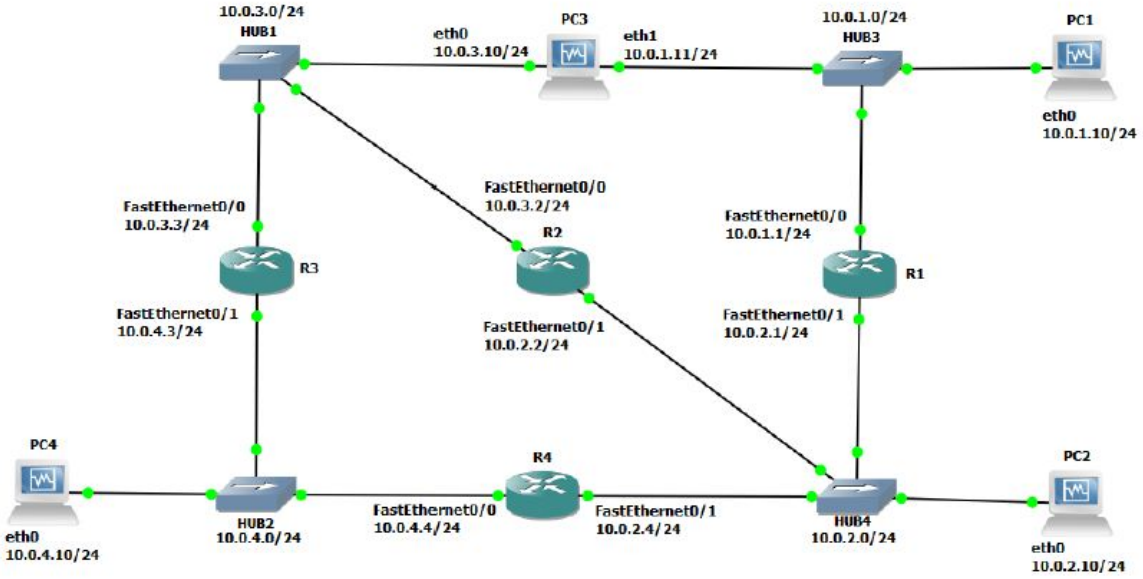

Figure 4.3 Network topology for Part 4

Linux PC	**Interface eth0**	**Interface eth1**
PC3	10.0.3.10 / 24	10.0.1.11 / 24

Table 4.3 IP address of the Cisco routers and VPCSs for Part 3.

2. On PC3, enable RIP forwarding. Log in to the ripd process on PC3 and issue the following commands, which enable RIP on both interfaces of PC3. Here, PC3 is set up as an IP router, and its interfaces are enabled in active mode. Therefore, it must send RIP packets.

3. Enable IP forwarding on PC3 with the following command:
or
PC3% echo “1” > ‘/proc/sys/net/ipv4/ip_forward’
You can check if the ip_forward is set by using this command. If the value is “1” it is set, if “0” you need to enable it:
PC3% sysctl net.ipv4.ip_forward
4. On PC3, start to capture traffic with Wireshark on both interfaces (eth0andeth1). Set a capture or display filter to limit the displayed traffic to RIP packets.
5. Use the netstat –rn command on PC1 and PC4, and use the show ip route command on Router3 to observe that the routing tables have been updated accordingly.
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)# no passive-interface eth0
ripd(config-router)# no passive-interface eth1
ripd(config-router)# redistribute connected

ripd(config-router)# end
ripd# exit
PC3% sysctl -w net.ipv4.ip_forward=1

Exercise 4(B). The count-to-infinity problem

The goal of this exercise is to observe the effects of the count-to-infinity problem for routes to network 10.0.1.0/24. Note that in Figure 4.3, the traffic to network 10.0.1.0/24 passes through either PC3 or Router1. To conduct this exercise, you first ensure that traffic to network 10.0.1.0/24always prefers the path through PC3. This is done by setting the link metric of the interface of Router1's network10.0.1.0/24 to a large value. Then, you will disable the eth1interface of PC3. When the interface is disabled, the next update packet from PC3 will list the cost metric for network 10.0.1.0/24 as 16, which, in RIP, is interpreted as *infinity*. If this information is not distributed quickly, then the other routers in the network send RIP packets that assume that PC3 is still connected to network 10.0.1.0/24. For example, Router3 may still believe that it can reach network 10.0.1.0/24 in two hops. When such an update arrives at PC3, then PC3 wrongly assumes that it can reach network 10.0.1.0/24 via Router3 in three hops. In this situation, it may take the hosts and routes a very long time before they realize that the best path to network 10.0.1.0/24 is through Router1. This slow convergence of the routing tales is the count-to-infinity problem. Please review IOS Mode: Router Configuration from Part 1.

RIP has several protocol features that try to avoid the count-to-infinity problem. One of them, called *triggered updates*, forces a router to immediately issue a RIP update packet whenever the cost metric of a routing table entry changes. In the previous scenario, the link failure triggers an update message, which ensures that information about the link failure of PC3 is quickly propagated to all systems in the network. This prevents routers from continuing to advertise routes that are based on incorrect information about the network topology. Another feature that avoids count-to-infinity is the *hold-down* mechanisms. When a router receives information that a route is unavailable (i.e., having cost metric 16), it puts the route in a hold-down state. In this state, the router ignores routing updates that advertise a better cost metric for a certain period of time (which is given by the hold-down timer).

To increase the likelihood of the count-to-infinity problem occurring, it is necessary to disable triggered updates and to set the hold-down timer to 0. Still, it may happen that the count-to infinity

problem does not manifest itself, since the problem is dependent on the timing of RIP update messages between the routers. Repeating the exercise several times will eventually

exhibit the count-to-infinity problem.

If the *hold-down* timer is set to ***h***, then the RIP process ignores information about an improved route for ***h*** seconds, after a RIP update packet has been received that states that this route is not available. Setting the hold-down timer to 0 means that the RIP process immediately accepts updates to a route. The command to set the hold-down timer to 0 seconds is The *triggered update* feature is controlled by setting the value of the flash-update-threshold timer. Suppose the timer is set to ***h***. Then, whenever the metric of a routing table changes, the router sends a RIP update packet only if the next round of (regularly scheduled) update messages is more than *h* seconds away. Triggered updates are disabled by setting the flashupdate- threshold timer to the same value as the update timer. Assuming that the update timer is set to the default value of 30

seconds, then the command to disable triggered updates is 1. On all Cisco routers, set the *hold-down timer* to 0 and disable *triggered updates*. The configuration for Router2 follows.
2. Next, you make routes to and from network 10.0.1.0/24 through Router1 unattractive by increasing the cost metric of interfaces FastEthernet0/0 and FastEthernet0/1 at Router1 to 10. This simulates a situation in which the hop count from Router1 to network 10.0.1.0/24 is 10 hops.
3. Wait until the routing tables on all routers have converged. Since the cost of the interface on Router1 is set high, you should observe that the traffic from all hosts and routers to network

Router1(config-router)#flash-update-threshold 30
Router1(config-router)#timers basic 30 180 0 240
Router2#enable
Router2#configure terminal
Router2(config)#no ip routing
Router2(config)#ip routing
Router2(config)#router rip
Router2(config-router)#version 2
Router2(config-router)#network 10.0.0.0
Router2(config-router)#timers basic 30 180 0 240
Router2(config-router)#flash-update-threshold 30
Router2(config-router)#end
Router2#clear ip route *
Router1#enable
Router1#configure terminal
Router1(config)#router rip
Router1(config-router)#offset-list 0 out 10 FastEthernet0/0
Router1(config-router)#offset-list 0 out 10 FastEthernet0/1
Router1(config-router)#end
Router1#clear ip route *

10.0.1.0/24 passes through PC3. This can be verified by issuing a traceroute (in Linux) or trace (in IOS) to IP address 10.0.1.10.
4. Start to capture traffic with Wireshark on interface eth0 of PC3. Set a display filter to display only RIP packets.
5. Issue a ping command from PC2 to PC1.
6. On Router3, enable the debugging mode of RIP. In this mode, the router displays all received RIP packets. This mode is enabled by typing:
You can disable the command by typing no debug ip rip.
7. On PC3, disable interface eth1 to break the connection between PC3 and network 10.0.1.0. PC3 will send out a RIP packet indicating that the cost metric to network 10.0.1.0/24 is now *infinity* (16). When the interface is disabled, the pings from PC2 to PC1 will fail. The ping commands to PC1 are again successful, once the routing tables have converged.
8. Now you should observe the slow convergence of the routing tables due to the count-toinfinity problem, by observing the content of the RIP messages that are shown by Wireshark on PC3 and by observing the debugging output on Router3.
9. Save the debug messages on Router3. Then disable RIP debugging on Router3.

10. Save the RIP packets captured by Wireshark on PC3. You need to save only the packets you need to explain the count-to-infinity problem.

Exercise 4(C). Avoiding the count-to-infinity problem

This exercise repeats Exercise 4(B), with the difference that triggered updates are enabled and the hold-down timer is set up a nonzero value. As a result, you observe that the count-to-infinity problem does not occur.

1. On Router1, enable triggered updates by setting the flash-update-threshold timer to 0, and by setting the hold-down timer to twice the value of the update timer. You can use the command show ip protocols to display the value of the update timer. If the update timer is set to the default value of 30 seconds, set the hold-down timer to 60 seconds.
2. Repeat the previous commands on all other Routers.

Router3#debug ip rip

3. Enable eth1 on PC3 with the previous configurations.
4. Now repeat Steps 3-9 from Exercise 4(B). You should observe that the count-to-infinity problem does not occur. In other words, if the connect between PC3 interface eth1 and network 10.0.1.0/24 is broken, you should observe that the routing tables converge much faster. Save the RIP packets captured by Wireshark on PC3.

PART 5.Configuring Open Shortest Path First (OSPF)

Next, you explore the routing protocol Open Shortest Path First (OSPF). OSPF is a link state routing protocol, in which 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 in messages that are called 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. However, each LSA is forwarded only once. Each router maintains a link state database of all received LSAs, which provides the router with complete information about the topology of the network, Routers use their link state databases to run a shortest-path algorithm that computes the shortest paths in the network.

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

OSPF is one of the most important link state routing protocols of 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).

The network configuration is shown in Figure 4.4 and Table 4.4. Note that PC1-4 are set up as routers.

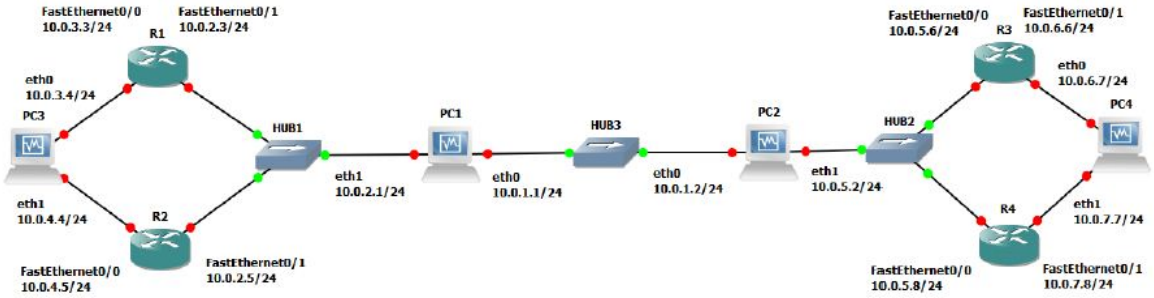

Figure 4.4 Network topology for Part 5

Cisco Routers	**Ethernet Interface FastEthernet 0/0**	**Ethernet Interface FastEthernet 0/1**
Router1	10.0.3.3 / 24	10.0.2.3 / 24
Router2	10.0.4.5 / 24	10.0.2.5 / 24
Router3	10.0.5.6 / 24	10.0.6.6 / 24
Router4	10.0.5.8 / 24	10.0.7.8 / 24
PCs	**Interface eth0**	**Interface eth1**
PC1	10.0.1.1 / 24	10.0.2.1 / 24
PC2	10.0.1.2 / 24	10.0.5.2 / 24
PC3	10.0.3.4 / 24	10.0.4.4 / 24
PC4	10.0.6.7 / 24	10.0.7.7 / 24

Table 4.4. IP addresses of the routers for Part 5

Exercise 5(A). Configuring OSPF on Cisco routers

In this exercise, you configure OSPF on the Cisco routers. A brief description of the basic IOS commands used to configure OSPF on a Cisco router follows. 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. process-id is a number that identifies the process. In this lab, only one OSPF process is started per router, and the process-id value is always set to 1. (The process-id of a router does not need to be the same on all routers). 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: PREVILEGED EXEC

show ipospf

Displays general information about the OSPF configuration

show ipospf database

Displays the link state database.

show ipospf border-routers

Displays the Area Border Router (ABR) and Autonomous System Boundary Router (ASBR).

clear ipospf process-id process

Resets the specified OSPF process.

IOS MODE: ROUTER CONFIGURATION

network NetaddrInvNetmask 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 net mask (InvNetmask). For example, Netaddr=10.0.0.0 and InvNetmask=0.255.255.255 specify the network prefix 10.0.0.0/8 and the broadcast area 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 Router 1 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 packets and 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. This commands can be used to set the value explicitly.

1. Connect the routers as shown in Figure 4.4.
2. Configure the Cisco routers to run OSPF. The following commands are used to configure Router1:

These commands configure the IP addresses of the routers, disable RIP, and enable OSPF for Area 1 and network 10.0.0.0/8. Since no router-id is specified, the highest IP address of Router1, 10.0.3.3, is used as the router-id. The router-id can be verified by issuing the command show ip OSPF.

3. Set up the PCs as OSPF routers. Refer to Figure 4.4 for the connections and to Table 4.4 for the IP addresses. Use the following set of commands.
4. Now configure the PC's similar to the way you configured the routers
5. Enable ip_forwarding on all the PCs.

Router1> enable
Router1#configure terminal
Router1(config)#interface FastEthernet0/0
Router1(config-if)# no shutdown
Router1(config-if)#ip address 10.0.3.3 255.255.255.0

Router1(config-if)#interface FastEthernet0/1
Router1(config-if)# no shutdown
Router1(config-if)#ip address 10.0.2.3 255.255.255.0
Router1(config-if)#exit
Router1(config)# no ip routing
Router1(config)#ip routing
Router1(config)#no router rip
Router1(config)#router ospf 1
Router1(config-router)#network 10.0.0.0 0.255.255.255 area 1
Router1(config-if)#end
Router1#clear ip route *
PC1% telnet localhost 2604
Password:zebra
ospfd>enable
ospfd#configure terminal
ospfd(config)#no router rip
ospfd(config)#router ospf
ospfd(config-router)#network 10.0.0.0/8 area 1
ospfd(config-router)#router-id 10.0.1.1
ospfd(config-router)#no passive-interface eth0
ospfd(config-router)#no passive-interface eth1
ospfd(config-router)#end
ospfd#exit

Exercise 5(B). Observing convergence of OSPF

In comparison to the distance vector protocol RIP, the link state routing protocol OSPF quickly adapts to changes in the network topology. In this exercise, you observe the interactions of OSPF after a change to the network topology.

1. On PC1, start to capture traffic with Wireshark on interface FastEthernet0/0. Set a filter to display only OSPF packets.
2. From PC3, run a trace command to PC4. Confirm from the output and Figure 4.4 whether the path from PC3 to PC4 includes Router3 or Router4.
3. Issue a ping command from PC3 to PC4 (10.0.7.7). Do not terminate the ping command until this exercise is completed.
4. If the path from PC3 to IP address 10.0.7.7 from Step 2 included Router3, then disconnect the Ethernet cable of FastEthernet0/1 interface of Router3. Otherwise, disconnect the Ethernet cable of FastEthernet0/1 interface of Router4.

When the Ethernet cable is disconnected, the ping command on PC3 will show that IP address 10.0.7.7 is not reachable.

5. Now OSPF updates the routing tables. Use the Wireshark window on PC1 to observe the transmitted OSPF messages:

• How quickly are OSPF messages sent after the cable is disconnected?
• How many OSPF messages are sent?
• Which type of OSPF packet is used for flooding link state information?
• Describe the flooding of LSAs to all routers.

• Which type of encapsulation is used for OSPF packets (TCP, UDP, or other)?
• What is the destination address of OSPF packets?
6. Wait until the ping command is successful again, that is, ICMP Echo Reply messages arrive at PC3. This happens when the routing tables have been updated.
7. Stop the ping command and save the ping statics output.
• Count the number of lost packets and calculate the time it took OSPF to update the routing tables. (The ping command issues an ICMP Echo Request message approximately once every second.)
8. Issue another trace command from PC3 to IP address 10.0.7.7 By now, the output should show the new route to PC4.
9. Save the link state database on all Cisco routers to a file, and verify that all routers indeed have the same link state database.
• Compare the output of the command “show ip OSPF database” from the Cisco routers.

10. Stop Wireshark on PC1, and save the different types of OSPF packets captured by Wireshark. Save one copy of each type of OSPF packet that you observed.
a) Pick a single link state advertisement packet captured by Wireshark, and describe how to interpret the information contained in the link state advertisement.

PART 6. Hierarchical Routing in OSPF

The concept of areas in OSPF can be used to construct a hierarchical routing scheme. When the network is partitioned in to multiple areas, then routers must have complete topology information only about routers in the same are and only limited information about other areas. All areas must be connected to *Area 0*, which is a special area called the backbone area. This lab will have you build a two-level hierarchy: The backbone area is at the top of the hierarchy and the other areas are at the bottom of the hierarchy. Traffic between two areas is routed through the backbone area. Routers that connect to two areas are called area border routers (ABRs).
The configuration for this part is shown in Figure 4.5. Here, the network from Part 5 is partitioned into three areas. The area in the middle is the backbone area (Area 0). The IP addresses are the same as in Part 4 and need not be modified. PC1 and PC2 are area border routers.
In the following exercises, you define the areas and then observe how the link state databases are built.

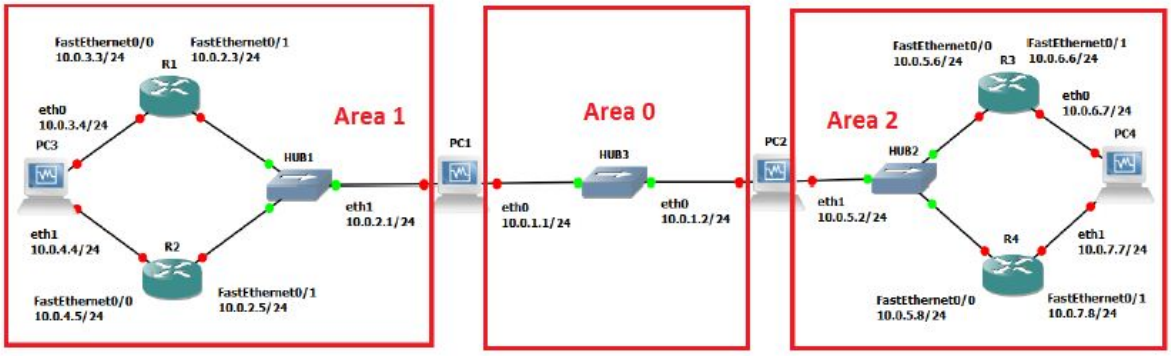

Figure4.5 Network topology for Part 6.

Exercise 6. Defining multiple areas in OSPF

1. Start Wireshark on PC1 and capture traffic on interface eth0
2. Change the Area IDs of the Cisco routers. On each system, the directly connected networks are assigned to an area with a 24 bit prefix. Here are the configurations for PC3 and PC1. The other configurations are similar.

PC3, which belongs to only one area, is configured as follows:

PC1 belongs to two areas and is configured as follows:

Router1 configuration is as follows:

Once the routing tables have converged, test the network configuration with the commands trace and ping on all routers. All routers should be able to communicate with one another.

3. Save the link state database on all routes. Also save the output of the link state database to a file.

• Compare the link state databases to those saved in Part 4. Which differences do you note?

PC3% telnet localhost 2604
Password: zebra
ospfd> enable
ospfd# configure terminal
ospfd(config)# no router ospf
ospfd(config)#router ospf
ospfd(config-router)# router-id 10.0.3.4
ospfd(config-router)# network 10.0.3.0/24 area 1
ospfd(config-router)# network 10.0.4.0/24 area 1
ospfd(config-router)# end
ospfd# exit
PC1% telnet localhost 2604
Password: zebra
ospfd> enable
ospfd# configure terminal
ospfd(config)# no router ospf
ospfd(config)#router ospf
ospfd(config-router)# router-id 10.0.1.1
ospfd(config-router)# network 10.0.2.0/24 area 1
ospfd(config-router)# network 10.0.1.0/24 area 0
ospfd(config-router)# end
ospfd# exit
Router1> enable
Router1# configure terminal
Router1 (config)# no router ospf 1
Router1 (config)#router ospf 1
Router1 (config-router)# network 10.0.3.0 0.255.255.255 area 1
Router1 (config-router)# network 10.0.2.0 0.255.255.255 area 1
Router1 (config-router)# end
Router1# clear ipospf 1 process

PART 7. Configuring The Border Gateway Protocol (BGP)

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

The network configuration for this part is shown in Figure 4.6, and the IP configuration information is given in table 4.5. the network has three autonomous systems with AS numbers 100, 200 and 300. PC4, is used to capture the BGP packets transmitted between the ASs.

Figure 4.6 Network topology for Part 7.

VPCS	**Ethernet Interface eth0**	**Ethernet Interface eth1**
PC1	10.0.1.10 / 24	Disabled
PC2	10.0.2.10 / 24	Disabled
PC3	10.0.3.10 / 24	Disabled
PC4	10.0.4.10 / 24	Disabled

Cisco Routers	**Ethernet Interface FastEthernet 0/0**	**Ethernet Interface FastEthernet 0/1**
Router1	10.0.1.1 / 24	10.0.4.1 / 24
Router2	10.0.2.2 / 24	10.0.4.2 / 24
Router3	10.0.3.3 / 24	10.0.4.3 / 24

Table 4.5 IP addresses of the routers and PCs for Part 7.

Exercise 7(A). Basic BGP configuration

Here, you configure the Cisco routers as BGP routers and you assign routers to autonomous systems. The configuration is completed when you can issue ping commands between any two PCs Next 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: PREVILEGED EXEC

show ipbgp

Displays the BGP routing table.

show ipbgp neighbors

Displays the neighbors, also called peers, of this BGP router.

show ipbgp paths

Displays the BGP path information in the local database.

clear ipbgp *

Deletes BGP routing information

IOS MODE: ROUTER CONFIGURATION
network Netaddr
network Netaddr mask netmask
Specifies a network address that will be advertised by the local BGP process. A network mask maybe added to denote the length of the network prefix.
neighbor IPaddress remote-as ASnumber
Adds a neighbor to the BGP neighbor table. IPaddress is the IP address and ASnumber is the AS number of the neighbor.
timers bgp keepalive holdtime
Sets the values of the keep alive and hold time 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 keep alive 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 is specified by the hold time (default: 180 seconds).
1. Disable all RIP or OSPF processes that are running on the Cisco routers. Use the following commands:
Router1# no router ospf 1
Router1# no router rip
2. Disable all RIP or OSPF processes running on the Linux PCs using the following command. For PC1, on the console at the prompt type:
PC1% /etc/init.d/quagga stop
3. Assign the IP addresses to Ethernet interface eth0 of each PC as indicated in Table 4.5
4. Disable eth1 on the Linux PCs using the following command as shown in Table 4.5.
For PC1, on the console at the prompt type:
PC1% ifconfig eth1 down
5. Add a default gateway to PC1, PC2, and PC3 as follows:
PC1% route add default gw 10.0.1.1/24
PC2% route add default gw 10.0.2.2/24
PC3% route add default gw 10.0.3.3/24

6. Start Wireshark on PC4 and set a display filter to capture only BGP packets.
7. 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. Following is the configuration for Router2. Router 2 is in AS 200 and neighbors are AS 100 and AS 300.
8. On PC1, issue a ping command to PC3. The command succeeds when BGP has converged.
9. Once the routing tables have converged, you see all the other AS entries in the BGP routing table. On each Cisco router, save the output of the following commands:
Router1# show ip route
Router1# show ip bgp
Router1# show ip bgp paths
• Describe the different types of BGP messages that you observe in the Wireshark window on PC4.
• Notice that BGP transmits messages over TCP connections. What is a reason that BGP

uses TCP to transmit its messages?
• What is the IP address of the next-hop attribute for AS 100 on Router 2?
• What are the BGP peers in this topology?
10. Stop the Wireshark traffic capture on PC4 and save the BGP packets captured by Wireshark.
a) Use the output to provide answers to the questions in Step 7.
Router2> enable
Router2# configure terminal
Router2(config)#no ip routing
Router2(config)# ip routing
Router2(config)# interface FastEthernet0/0
Router2(config-if)# no shutdown
Router2(config-if)# ip address 10.2.2 255.255.255.0
Router2(config-if)# interface FastEthernet0/1
Router2(config-if)# no shutdown
Router2(config-if)# ip address 10.0.4.2 255.255.255.240
Router2(config-if)#router bgp 200
Router2(config-router)# neighbor 10.0.4.1 remote-as 100
Router2(config-router)# neighbor 10.0.4.3 remote-as 300
Router2(config-router)# network 10.0.2.0 mask 255.255.255.0
Router2(config-router)# end
Router2# clear ip bgp *

b) Which BGP message(s) contain(s) the AS-PATH information? Use a BGP message to illustrate your answer.
c) Use the saved output to provide a brief explanation of how the routers find the proper path between the autonomous systems.

Exercise 7(B). BGP convergence

Disconnect one of the links between two BGP peers and observe how the BGP protocol reconfigures the paths.
1. After completing Exercise 6(A), save the output of the command show ip BGP neighbors on Router2. Pay attention to the neighbor AS information.
2. On PC4, run Wireshark and set a display filter for BGP. Observe the flow of BGP packets between the autonomous systems.
3. On all routers, change the keepalive timer to 10 seconds and the holdtime timer to 30 seconds. This speeds up the convergence time by a factor of 6 as compared to the default values. The following are the commands for Router2:
4. Disconnect the cable of interface FastEthernet0/1 on Router1.
• From the output you saved, describe how the BGP routers learn that a link is down. (Hint: Look at the BGP State field)
• Which BGP messages indicate that there is a link problem? Use a BGP message to answer the question.
5. Use the command show ip BGP neighbors on Router2 and Router3 to obtain the neighbor information. Save the output.
6. Wait until BGP converges. Save the routing tables on Router2 and Router3. What can you say?

Router2# configure terminal
Router2(config)#router bgp 200
Router2(config-router)# timers bgp 10 30
Router2(config-router)#end
Router2#clear ip bgp *