Based on
Mastering Networks - An Internet Lab Manual
by Jörg Liebeherr and Magda Al Zarki

Adapted for
'Labo Computernetwerken'
by Johan Bergs, Nicolas Letor, Michael Voorhaen and Kurt Smolderen

Completed by
Johan Bergs

April 23, 2017

Lab 4

Dynamic Routing Protocols (RIP and OSPF)

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 the count-to-infinity problem in RIP can be avoided.
- How OSPF achieves a hierarchical routing scheme through the use of multiple areas.

Prelab 4

Routing protocols

- *Distance Vector and Link State Routing Protocols*: Go to the website http://docwiki.cisco.com/wiki/Internetworking_Technology_Handbook and read the article about dynamic routing protocols. Review your knowledge of interdomain and intradomain routing, distance vector routing, and link state routing.
- *Zebra*: Go to the website of the Zebra fork Quagga at http://www.nongnu.org/quagga/ and study the information on the Quagga routing protocol software for Linux systems. Also find and read the man pages on zebra, ripd, ospfd and bgpd. Note: Quagga is a fork of the GNU Zebra project.
- *RIP*: Read the overview of the Routing Information Protocol (RIP) and study the commands to configure RIP on a Cisco router at http://www.routeralley.com/guides/rip.pdf.
- *OSPF*: Read the overview of Open Shortest Path First (OSPF) routing protocol and study the commands to configure OSPF on a Cisco router at http://www.routeralley.com/guides/ospf.pdf.

Prelab Questions

Question 1)

Provide the command that configures a Linux PC as an IP router (see Lab 3).

```
echo "1" >/proc/sys/net/ipv4/ip_forward
```

When rebooting, these changes will be reset. So when you want the ip forwarding to be permanent, you'll have to use:

```
sysctl -w net.ipv4.ip_forward=1
```

Question 2)

What are the main differences between a distance vector routing protocol and a link state routing protocol? Give examples for each type of protocol.
A distance vector routing protocol informs its neighbours of changes in the network topology. A link state routing protocol informs all the nodes in a network of topology changes.
RIP (Routing Information Protocol) is an example of a distance vector routing protocol. OSPF (Open Shortest Path First) is an example of a link state routing protocol.

Question 3)

What are the differences between an intradomain routing protocol (also called interior gateway protocol or IGP) and an interdomain routing protocol (also called exterior gateway protocol or EGP)? Give examples for each type of protocol.
An intradomain routing protocol will route packets within a domain or autonomous system. Information will be exchanged between nodes within this domain or autonomous system. Two examples of intradomain routing protocols are RIP and OSPF. An interdomain routing protocol will route packets between domains or autonomous systems. Information will be exchanged between nodes from different domains or autonomous systems.
An example of and interdomain routing protocol is the Border Gateway Protocol.

Question 4)

Which routing protocols are supported by the software package Zebra?
OSPFv2, OSPFv3, RIP v1, RIP v2, RIPng and BGP-4

Question 5)

In the Zebra software package, the processes ripd, ospfd, and bgpd deal, respectively, with the routing protocols RIP, OSPF, and BGP. Which role does the process zebra play?
Zebra manages routing protocols and provides an easy user interface for each routing protocol.

Question 6)

Describe how a Linux user accesses the processes of Zebra (zebra, ripd, ospfd, bgpd) processes to configure routing algorithm parameters?
You must enable zebra by adjusting the file /etc/quagga/daemons. You must change/add the line "zebra=yes".
In the same file, you must also activate the protocol. This is also done by changing/adding the line "X=yes" where X is ripd, ospfd or bgpd. Afterwards, you'll have to type the following command: /etc/init.d/quagga start. In case quagga was already running, you shouldn't start quagga, you should obviously restart it. This is done by using the command: /etc/init.d/quagga restart.

Question 7)

What is the main difference between RIP version 1 (RIPv1) and RIP version 2 (RIPv2)?

RIPv1 is a distance vector routing protocol. RIPv2 is a hybrid routing protocol, it has characteristics of both distance vector routing protocols and link state routing protocols.

Question 8)

Explain what it means to “run RIP in passive mode”.

Passive routers will receive and process updates, but they will not send updates to other nodes.

Question 9)

Explain the meaning of “triggered updates” in RIP.

Every router on which RIP is enabled will have a send-timer. Once the timer expires, he will send an update and reset his send-timer. By default, this interval is 30 seconds.

When a router changes the metric for a route, he will send a triggered update. This will not reset his send-timer.

This is because if he wouldn't send the updates immediately, the other nodes would have outdated information for 30 seconds (at most).

Question 10)

Explain the concept of split-horizon in RIP?

Split-horizon tries to avoid loops in an RIP protocol.

There's one rule: Do not advertise addresses through the interface from which you received the initial advertisement.

For example: If you have node A connected to node B and node B connected to node C. Node A will advertise it's route for node C to every other node that is not on the same interface as node B because he got the initial advertisement from node B.

So if node A was connected to node D (via another interface), he would advertise the route for node C to node D but not to node B.

If there were another node E connected to A on the same interface as node B is connected to A, he would not advertise the route for C to node E.

Question 11)

What is an autonomous system (AS)? Which roles do autonomous systems play in the Internet?

An autonomous system (or domain) is a collection of connected nodes with a clearly defined routing policy. It is under control of a single administravtive entity.

These autonomous systems communicate internally. They are also responsible for routing packets to the nodes within this system.

But, these domains also communicate with each other. So it's possible for a node in autonomous system A to send packets to autonomous system B.

Question 12)

What is the AS number of your institution? Which autonomous system has AS number 1?

The University of Antwerp has AS number 2611.

Level 3 Communications Inc. has AS number 1. It's an American multinational telecommunications and internet service provider company.

Question 13)

Explain the terms: Stub AS, Multi-homed AS and Transit AS?

A stub AS sends and receives packets whose source or destination are one of its own hosts.

A transit AS provides services for other AS's, i.e. forwarding the packets (whose source and destination don't belong to his own AS) to other AS's.

A multi-homed AS is connected to two or more transit AS's.

Lab 4

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, neighbouring 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 the two most common 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.

This lab uses two different network configurations. The first network configuration, shown in Figure 4.1, is used in Parts 1-2, and is modified in Part 3 (Figure 4.3). The network configuration in Parts 4 and 5 is shown in Figure 4.4.

Configuring RIP on a 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 tables 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.

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 of 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 neighbouring 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 neighbouring router, it updates its own routing table.

Figure 4.1 and Table 4.1 describe the network configuration for this part of the lab.

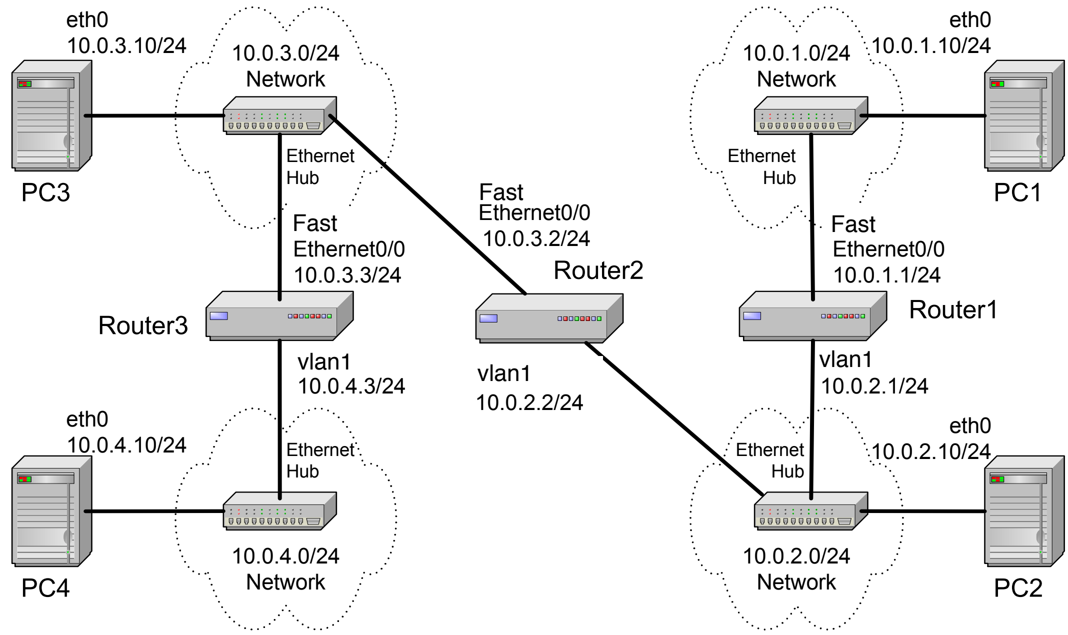

Figure 4.1: Network configuration for Parts 1 and 2.

Exercise 1. Configuring RIP on Cisco routers

Configure all three Cisco routers to run the routing protocol RIP. Once the configuration is completed, all Cisco routers 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.

The following can can be executed in the Global Configuration mode.

Linux PC	eth0	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 Router	**FastEthernet0/0**	**vlan1**
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

- Enable the routing protocol RIP on the local router, and enters the router configuration mode from the following prompt:

  ```
  Router1(config-router)#
  ```

  You return from the router configuration command to the global configuration command by typing the command `exit`.

  ```
  router rip
  ```

- Disable RIP on the local router.

  ```
  no router rip
  ```

The following can can be executed in the Privileged EXEC mode.

- Enable a debugging mode where the router displays a message for each received RIP packet.

  ```
  debug ip rip
  ```

- Disable the debugging feature

  ```
  o debug ip rip
  ```

The following can can be executed in the Router Configuration mode.

- Associate the network IP address *Netaddr* with RIP. RIP sends updates only on interfaces where the network address has been associated with RIP.

  ```
  etwork Netaddr
  ```

- Disable RIP for the specified network address.

  ```
  no network Netaddr
  ```

- Set the interface *Iface* in RIP passive mode. On an interface in passive mode, the router processes incoming RIP packets, but does not transmit RIP packets.

  ```
  passive-interface Iface
  ```

- Enable active mode on interface *Iface*. This means that RIP packets are transmitted on this interface.

```
no passive-interface Iface
```

- Increase the metric (hop count) of incoming RIP packets that arrive on interface *Iface* by *value*, where *value* is a number.

```
offset-list 0 in value Iface
```

- Increase the metric of outgoing RIP packets that are sent on interface *Iface* by *value*.

```
offset-list 0 out value Iface
        \end{verbatim}
\item Disable the specified offset-list command for incoming RIP packets.
        \begin{cmdblock}
no offset-list 0 in value Iface
```

- Disable the specified offset-list command for outgoing RIP packets.

```
no offset-list 0 out value Iface
```

- Set the RIP version to RIPv2.

```
version 2
```

- Set the values of the timers in the RIP protocol. The timers are measured in seconds.

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

`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 packets with routing updates after a link failure (Default: 180 sec).

`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).

Example:

```
Router1(config-router)# timers basic 30 180 180 240
```

- Set the router to not 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 updates 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.

```
flash-update-threshold time
```

1. Connect the the Linux PCs and the Cisco routers as shown in Figure 4.1. The PCs and routers are connected with Ethernet hubs.

2. Verify that the serial interfaces of the PCs are connected to the console port of the routers. PC1 should be connected to Router1, PC2 to Router2, and so on. Once the serial cables are connected, establish a minicom session from each PC to the connected router.

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 Router1 are as follows:

```
Router1> enable Password: <enable secret>
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)# interface vlan1
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 *
```

The command `no ip routing` is used to reset all previous configurations related to routing (RIP, OSPF, etc). The command `clear ip route *` deletes all entries in the routing table. Make sure that all static routing entries are removed, since, in IOS, RIP does not overwrite static routing entries.

4. After you have configured the routers, check the routing table at each router by typing

```
Router1# show ip route
```

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 addresses of interfaces *FastEthernet0/0* and *vlan1* on all remote routers. For example, to issue a ping from Router1 to interface *FastEthernet0/0* on Router2, type

```
Router1# ping 10.0.3.2
```

Once you can successfully contact the IP addresses of all routers, proceed to the next exercise.

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, where a host receives and processes incoming routing messages, but does 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. Before starting the exercise, we give a brief tutorial on the Quagga software package. The tutorial focuses on the features used in the lab exercises and omits many interesting features of Zebra.

An Introduction to 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. For this course we make use of Quagga, which is a fork of the GNU Zebra project and while the project has a new name, many of the references to Zebra still remain, e.g. there is still a `zebra` control process.

The Quagga architecture, shown in Figure 4.2, consists of a set of processes. The process `zebra` updates the routing tables and exchanges routes between different routing protocols. Each routing protocol has a separate process, and each routing process can be started, stopped, configured, and upgraded independently of the other routing processes. The process `zebra` must be invoked prior to starting and configuring any of the routing protocols. The routing processes used in this lab and the routing protocols they manage are shown in table 4.2.

Routing Process	Routing Protocol
ripd	RIPv1 and RIPv2
ospfd	OSPFv2 (Version 2)

Table 4.2: Quagga routing processes used for this lab.

(a) Adding the directory with Quagga commands to the search path
On Ubuntu systems, the script to start, stop and control the `zebra` process and its routing processes is located in directory `/etc/init.d`.

```
PC1% /etc/init.d/quagga start
```

(b) Starting and stopping Quagga processes

`/etc/init.d/quagga start`
Start the Quagga processes.

`/etc/init.d/quagga stop`
Terminate the Quagga processes.

`/etc/init.d/quagga restart`
Stop and restart the Quagga processes.

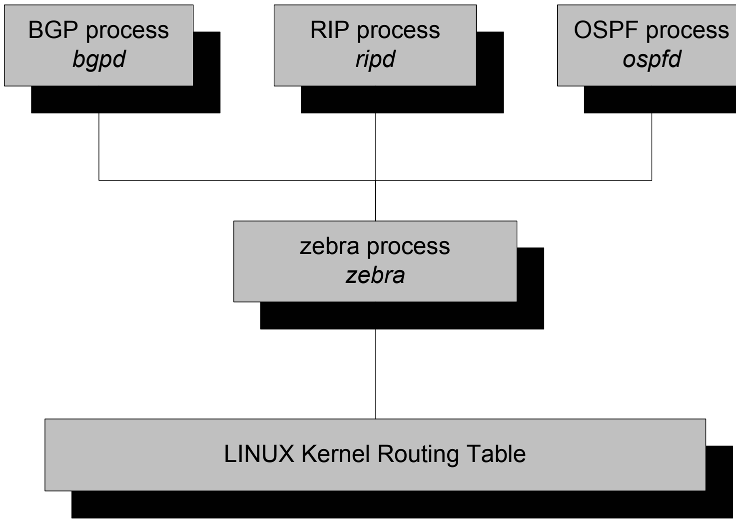

Figure 4.2: Quagga processes

To set up a routing process, you must enable the routing daemon in the Quagga configuration file `/etc/quagga/daemons` and than start the `quagga` service. For example to start the RIP routing protocol daemon, your `daemons` file should look as shown below. Afterwards, you can start the `zebra` and the `ripd` daemons by running `/etc/init.d/quagga start` or `/etc/init.d/quagga restart` in case Quagga was already running.

```
zebra=yes
bgpd=no
ospfd=no
ospf6d=no
ripd=yes
ripngd=no
isisd=no
```

Make sure the `zebra` daemon is always enabled as the other routing daemons depend on this process. When you type `/etc/init.d/quagga 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 `/usr/local/etc` or `/etc/quagga`, and have names `zebra.conf`, `ripd.conf`, etc. The configuration files look similar to the configuration files of IOS, and contain commands that are executed when the process is started.

(c) Configuring the `zebra` process and the routing protocol processes

After starting the `zebra` process or any of the routing protocol processes, you can configure each process 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 as follows:

- 2601 - Zebra
- 2602 - ripd
- 2604 - ospfd

If you establish a Telnet session to a routing process, you are asked for a password. If the password is correct, a command prompt is displayed. For example, to access the ripd process on the local host you type:

```
PC1% telnet localhost 2602
```

This results in the following output.

```
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.

Hello, this is Quagga (version 0.99.20.1).
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
```

(d) 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 IOS, and the syntax of commands is generally the same as the corresponding commands in 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
```

The password and enable password for all Quagga deamons (ripd and ospfd) is set to ‘mvkbj1n‘.

After this brief tutorial, you can now complete the configuration of RIP on the Linux PCs.

Exercise 2. Configuring RIP on Linux PCs with Quagga

Enable RIP on all Linux PCs. Since all Linux PCs are running as hosts, RIP is set to passive mode, where the PCs receive and process incoming RIP packets, but do not transmit RIP packets. The following guidelines describe the configuration of PC1. Repeat the steps on each PC.

1. On PC1, start the `zebra` and the`ripd` daemons by typing

```
PC1% /etc/init.d/quagga start
```

Make sure your daemons configuration file is correctly configured.

2. To configure the RIP routing process on PC1, connect to the ripd process via Telnet.

```
PC1% telnet localhost 2602
```

The system will prompt you for a login password. The password should be the same password as the login password on the Cisco routers.

3. 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:

```
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
```

The show ip rip 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. When the routing table has stabilized, that is, the results of the command show ip rip do not change after subsequent rounds of update messages, save the output of the command, and exit the Telnet session with the command.

```
ripd# exit
```

4. On PC1, view the routing table with the command

```
PC1% netstat -rn
```

and save the output to a file. Compare the output of netstat -rn to the output of show ip rip. Note the cost metric for each entry.

5. Repeat Steps 1-5 for the other three Linux PCs.

6. Once you can successfully issue a ping from each Linux PCs to every other Linux PC, display the route from PC1 to PC4 (10.0.4.10) with the traceroute command and save the result to a file:

```
PC1% traceroute 10.0.4.10
```

7. Start to capture traffic with Wireshark on all four Linux PCs. Set a capture filter or display filter to display only RIP packets.

8. Stop the traffic Wireshark capture on the PCs and save the traces for your report to a pcap file. Save the content of those RIP messages, needed to answer the questions in Part 8 (Select the Print details option).

Question 2.1)
Use the captured data of a single RIP packet and explain the fields in a RIP message.

```
      3 0.809577151    10.0.1.1              10.0.1.10             RIPv2    106
          Response

Frame 3: 106 bytes on wire (848 bits), 106 bytes captured (848 bits) on interface 0
    Interface id: 0 (eth0)
    Encapsulation type: Ethernet (1)
    Arrival Time: Mar 16, 2017 15:00:53.305377492 CET
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1489672853.305377492 seconds
    [Time delta from previous captured frame: 0.004589418 seconds]
    [Time delta from previous displayed frame: 0.004589418 seconds]
    [Time since reference or first frame: 0.809577151 seconds]
    Frame Number: 3
    Frame Length: 106 bytes (848 bits)
    Capture Length: 106 bytes (848 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ethertype:ip:udp:rip]
    [Coloring Rule Name: UDP]
    [Coloring Rule String: udp]
Ethernet II, Src: 00:0d:65:17:01:29, Dst: 68:05:ca:36:33:a0
    Destination: 68:05:ca:36:33:a0
        Address: 68:05:ca:36:33:a0
        .... ..0. .... .... .... .... = LG bit: Globally unique address (factory
            default)
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Source: 00:0d:65:17:01:29
        Address: 00:0d:65:17:01:29
        .... ..0. .... .... .... .... = LG bit: Globally unique address (factory
            default)
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: IPv4 (0x0800)
Internet Protocol Version 4, Src: 10.0.1.1, Dst: 10.0.1.10
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes
    Differentiated Services Field: 0xc0 (DSCP: CS6, ECN: Not-ECT)
        1100 00.. = Differentiated Services Codepoint: Class Selector 6 (48)
        .... ..00 = Explicit Congestion Notification: Not ECN-Capable Transport (0)
    Total Length: 92
    Identification: 0x0000 (0)
    Flags: 0x00
        0... .... = Reserved bit: Not set
        .0.. .... = Don't fragment: Not set
        ..0. .... = More fragments: Not set
    Fragment offset: 0
    Time to live: 255
    Protocol: UDP (17)
    Header checksum: 0xa4c6 [validation disabled]
        [Good: False]
        [Bad: False]
    Source: 10.0.1.1
    Destination: 10.0.1.10
    [Source GeoIP: Unknown]
    [Destination GeoIP: Unknown]
User Datagram Protocol, Src Port: 520 (520), Dst Port: 520 (520)
    Source Port: 520
    Destination Port: 520
    Length: 72
    Checksum: 0xbf32 [validation disabled]
        [Good Checksum: False]
        [Bad Checksum: False]
    [Stream index: 2]
Routing Information Protocol
    Command: Response (2)
    Version: RIPv2 (2)
    IP Address: 10.0.2.0, Metric: 1
        Address Family: IP (2)
```

```
            Route Tag: 0
            IP Address: 10.0.2.0
            Netmask: 255.255.255.0
            Next Hop: 0.0.0.0
            Metric: 1
        IP Address: 10.0.3.0, Metric: 2
            Address Family: IP (2)
            Route Tag: 0
            IP Address: 10.0.3.0
            Netmask: 255.255.255.0
            Next Hop: 0.0.0.0
            Metric: 2
        IP Address: 10.0.4.0, Metric: 3
            Address Family: IP (2)
            Route Tag: 0
            IP Address: 10.0.4.0
            Netmask: 255.255.255.0
            Next Hop: 0.0.0.0
            Metric: 3
```

the captured RIP packet.

The RIP message starts with the command that the protocol issues. In this case, the command is Response, as a reply to the Request command of the previous packet. Following that is the version of RIP that is used (version 2 in this packet). After this you have a list of all the IP addresses that the host/router knows, providing routing information to the host/router that sent the request.

Question 2.2)

For PC1, include the output of the commands `show ip rip` and `netstat -rn` from Steps 4 and 5. Discuss the differences in the output of the commands.

```
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
10.0.1.0        0.0.0.0         255.255.255.0   U         0 0          0 eth0
10.0.2.0        10.0.1.1        255.255.255.0   UG        0 0          0 eth0
10.0.3.0        10.0.1.1        255.255.255.0   UG        0 0          0 eth0
10.0.4.0        10.0.1.1        255.255.255.0   UG        0 0          0 eth0
```

traces/2.4.Netstat.PC1.txt

```
Codes: R - RIP, C - connected, S - Static, O - OSPF, B - BGP
Sub-codes:
      (n) - normal, (s) - static, (d) - default, (r) - redistribute,
      (i) - interface

     Network            Next Hop         Metric From            Tag Time
C(i) 10.0.1.0/24        0.0.0.0               1 self              0
R(n) 10.0.2.0/24        10.0.1.1              2 10.0.1.1          0 02:55
R(n) 10.0.3.0/24        10.0.1.1              3 10.0.1.1          0 02:55
R(n) 10.0.4.0/24        10.0.1.1              4 10.0.1.1          0 02:55
```

traces/2.4.Rip.PC1.txt

The netstat command displays a few more statistics of the routing table. First of all it displays the MSS (Maximum Segment Size) for the different routes (0 meaning no changes should be made to the packet size). It also displays the Window, which specifies the default window size. The irtt column specifies the initial round trip time. Lastly the netstat command also shows on which interface the packets should be placed, depending on the destination ip address.

The other way around, the "show ip rip" command displays some other additional properties/statistics. This is specifically the tag for the different routes, the timeout time for the routes (Time), and the cost that is associated with the route (Metric).

Question 2.3)

Include the output of `traceroute` from Step 7.

```
traceroute to 10.0.4.10 (10.0.4.10), 30 hops max, 60 byte packets
 1   10.0.1.1 (10.0.1.1)  2.038 ms  2.410 ms  2.780 ms
 2   10.0.2.2 (10.0.2.2)  3.020 ms  4.495 ms  4.980 ms
 3   10.0.3.3 (10.0.3.3)  4.453 ms  4.465 ms  4.694 ms
 4   10.0.4.10 (10.0.4.10)  3.419 ms  3.658 ms  3.666 ms
```

traces/2.6.txt

Question 2.4.a)

What is the destination IP address of RIP packets?

The destination ip address for RIP request packets is $224.0.0.9$. RIP response packets either have a destination address of $224.0.0.9$ or the unicast address of the host that has sent a request.

Question 2.4.b)

Do routers forward RIP packets? In other words, does PC1 receive RIP packets sent by Router3?

Routers do not forward RIP packets. Routers only send RIP packets with information of their own routing tables. Because of this, PC1 does not receive RIP packets sent by Router 3.

Question 2.4.c)

Which types of routing RIP messages do you observe? The type of a RIP message is indicated by the value of the field command. For each packet type that you observed, explain the role that this message type plays in the RIP protocol.

We observed 2 types of RIP messages: Requests and Responses. A router that connects to the network the first time (or enables RIP at an arbitrary time), sends a request message to all its neighbouring routers by sending a multicast packet. The routers which receive this request answer with a Response message, containing their respective routing tables. Response messages are also sent periodically.

Question 2.4.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?

Each routing table entry consumes 20 bytes. They contain information about the Address Family (in our case IP), the Route Tag (in order to differentiate between different protocols if necessary), the IP address of the routing table entry, the netmask associated with the IP address, the next hop address and the metric for that route (hop count).

Reconfiguring the topology in RIP

In Part 3, you add Router4 to the network topology of Figure 4.1. The configuration of the network with Router4 is illustrated in Figure 4.3. The IP configuration of Router4 is given in Table 4.3. 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.

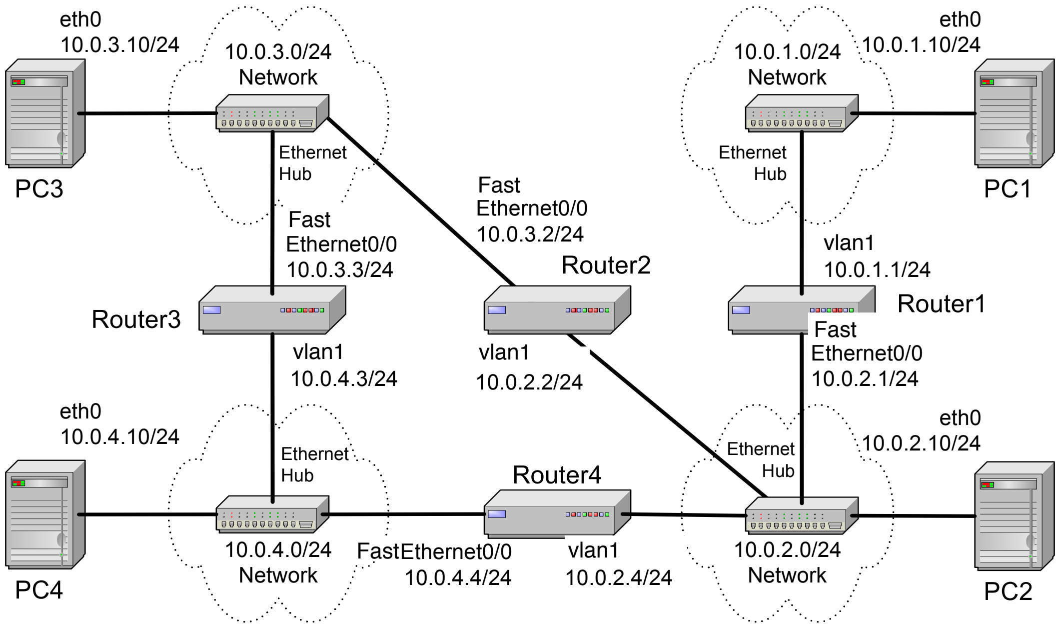

Figure 4.3: Network configuration for Part 3.

Cisco Router	FastEthernet0/0	vlan1
Router4	10.0.4.4/24	10.0.2.4/24

Table 4.3: IP addresses of Router4

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.
2. Before attaching Router4, save the routing tables on all four Linux PCs with the command `netstat -rn`.
3. Connect Router4 as shown in Figure 4.3 and assign the IP addresses to the interfaces as shown in Table 4.3.
4. Configure Router4 to run RIP, following the same steps as in Part 1.

5. Use 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, save the routing tables on all four Linux PCs.

Question 3.A)

Include the routing tables of the Linux PCs before the topology was changed (Step 2) and after Router4 has been added and the routing tables have been updated (Step 5). Discuss the time it took to update the routing tables.

Before the attachment of Router 4:

```
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
10.0.1.0        0.0.0.0         255.255.255.0   U         0 0          0 eth0
10.0.2.0        10.0.1.1        255.255.255.0   UG        0 0          0 eth0
10.0.3.0        10.0.1.1        255.255.255.0   UG        0 0          0 eth0
10.0.4.0        10.0.1.1        255.255.255.0   UG        0 0          0 eth0
```

traces/3–A.2.PC1.txt

```
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
10.0.0.0        0.0.0.0         255.0.0.0       U         0 0          0 eth0
10.0.1.0        10.0.2.1        255.255.255.0   UG        0 0          0 eth0
10.0.3.0        10.0.2.2        255.255.255.0   UG        0 0          0 eth0
10.0.4.0        10.0.2.2        255.255.255.0   UG        0 0          0 eth0
```

traces/3–A.2.PC2.txt

```
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
10.0.1.0        10.0.3.2        255.255.255.0   UG        0 0          0 eth0
10.0.2.0        10.0.3.2        255.255.255.0   UG        0 0          0 eth0
10.0.3.0        0.0.0.0         255.255.255.0   U         0 0          0 eth0
10.0.4.0        10.0.3.3        255.255.255.0   UG        0 0          0 eth0
```

traces/3–A.2.PC3.txt

```
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
10.0.1.0        10.0.4.3        255.255.255.0   UG        0 0          0 eth0
10.0.2.0        10.0.4.3        255.255.255.0   UG        0 0          0 eth0
10.0.3.0        10.0.4.3        255.255.255.0   UG        0 0          0 eth0
10.0.4.0        0.0.0.0         255.255.255.0   U         0 0          0 eth0
```

traces/3–A.2.PC4.txt

After the attachment of Router 4:

```
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
10.0.1.0        0.0.0.0         255.255.255.0   U         0 0          0 eth0
10.0.2.0        10.0.1.1        255.255.255.0   UG        0 0          0 eth0
10.0.3.0        10.0.1.1        255.255.255.0   UG        0 0          0 eth0
10.0.4.0        10.0.1.1        255.255.255.0   UG        0 0          0 eth0
```

traces/3–A.5.PC1.txt

```
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
10.0.1.0        10.0.2.1        255.255.255.0   UG        0 0          0 eth0
```

```
10.0.2.0        0.0.0.0         255.255.255.0   U     0 0          0 eth0
10.0.3.0        10.0.2.2        255.255.255.0   UG    0 0          0 eth0
10.0.4.0        10.0.2.4        255.255.255.0   UG    0 0          0 eth0
```

traces/3–A.5.PC2.txt

```
Kernel IP routing table
Destination     Gateway         Genmask         Flags MSS Window  irtt Iface
10.0.1.0        10.0.3.2        255.255.255.0   UG    0 0          0 eth0
10.0.2.0        10.0.3.2        255.255.255.0   UG    0 0          0 eth0
10.0.3.0        0.0.0.0         255.255.255.0   U     0 0          0 eth0
10.0.4.0        10.0.3.3        255.255.255.0   UG    0 0          0 eth0
```

traces/3–A.5.PC3.txt

```
Kernel IP routing table
Destination     Gateway         Genmask         Flags MSS Window  irtt Iface
10.0.1.0        10.0.4.4        255.255.255.0   UG    0 0          0 eth0
10.0.2.0        10.0.4.4        255.255.255.0   UG    0 0          0 eth0
10.0.3.0        10.0.4.3        255.255.255.0   UG    0 0          0 eth0
10.0.4.0        0.0.0.0         255.255.255.0   U     0 0          0 eth0
```

traces/3–A.5.PC4.txt

The updating of the routing tables happened very quickly (it was already updated the moment we checked the routing tables after connecting Router 4). As soon as Router 4 connected to the network, it sent a RIP request message. After receiving the responses from the other routers, Router 4 informed the other routers and hosts about his (updated) routing table.

Exercise 3-B. Convergence of RIP after a link failure

Next you disconnect the Ethernet cable of interface Ethernet0/0 on Router4 and observe how much time RIP takes to update the routing table of the Linux 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.

   ```
   PC4% ping 10.0.1.10
   ```

2. Disconnect the Ethernet cable connected to interface *FastEthernet0/0* on Router4. Now, the output of 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 save the ping statistics output (i.e. the data that appears at the bottom of the terminal screen when you stop the ping process).
5. Count the number of lost packets and calculate the time it took RIP to update the routing tables. (The `ping` command issues an ICMP Echo Request message approximately once every second.)

Question 3.B)

Include your answer on the convergence time from Step 4. Count the number of lost packets and calculate the time it took RIP to update the routing tables. (The ping command issues an ICMP Echo Request message approximately once every second.)

```
--- 10.0.1.10 ping statistics ---
235 packets transmitted, 47 received, +116 errors, 80% packet loss, time 235264ms
rtt min/avg/max/mdev = 1.582/1.895/2.320/0.312 ms, pipe 3
```

traces/3–B.4.txt

Approximately 188 $(235 - 47)$ packets were lost after disconnecting the cable. According to this, it took RIP about 3 minutes to update its routing tables (which also conforms with the previous timeout value seen in exercise 2.2).

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 in messages that are called link state advertisements (LSAs). LSAs are disseminated using flooding, that is, a router sends its LSAs to all its neighbours, which, in turn, forward the LSAs to their neighbours, 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 the most important 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). The network configuration is shown in Figure 4.4 and Table 4.4. Note that some Linux PCs and routers are connected with crossover cables.

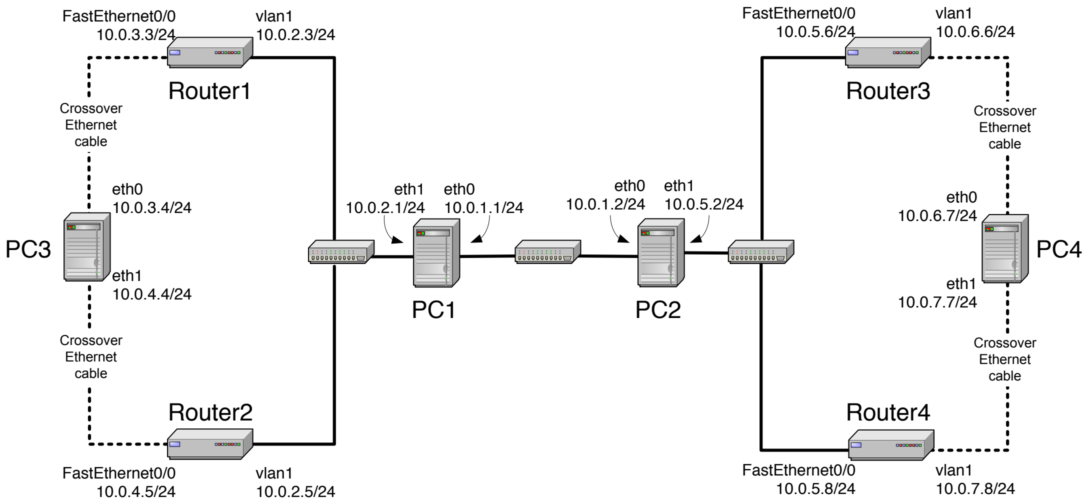

Figure 4.4: Network configuration for Part 4.

Linux PC	**eth0**	**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
Cisco Router	**FastEthernet0/0**	**vlan1**
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

Table 4.4: IP addresses for Part 5

Exercise 4-A. 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.

1. Connect the routers as shown in Figure 4.4. Some of the interfaces are connected with crossover cables or with hubs in between them.

2. Configure the Cisco routers to run OSPF. The following set of commands are used to configure Router1.

```
Router1> enable
Password: <enable secret>
Router1# configure terminal
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-router)# interface FastEthernet0/0
Router1(config-if)# ip address 10.0.3.3 255.255.255.0
Router1(config-if)# interface vlan1
Router1(config-if)# ip address 10.0.2.3 255.255.255.0
Router1(config-if)# end
Router1# clear ip route *
```

   The above commands disable RIP, enable OSPF for Area 1 and network 10.0.0.0/8, and configure the IP addresses of the routers. 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. Repeat the configuration on the other routers. Refer to Figure 4.4 for the connections, and to Table 4.4 for the IP addresses.

Exercise 4-B. Configuring OSPF on Linux PCs

On the Linux PCs, OSPF is configured using the Quagga package. The syntax of the Quagga commands is essentially identical to the corresponding IOS commands. All PCs are set up as IP routers. The following describes the configuration of PC1.

1. Connect PC1 as shown in Figure 4.4.

2. Enable IP forwarding on PC1 by typing

```
PC1% echo "1" > /proc/sys/net/ipv4/ip_forward
```

3. Terminate the existing ripd process and disable the `ripd` daemon in the `daemons` configuration file:

```
PC1% /etc/init.d/quagga stop
```

4. Disabel the `ripd` and enable the `ospfd` daemon in the `daemons` configuration file:

```
zebra=yes
bgpd=no
ospfd=yes
ospf6d=no
ripd=no
ripngd=no
isisd=no
```

5. Restart Quagga

```
PC1% /etc/init.d/quagga start
```

6. Set the OSPF configuration on PC1. Note that the commands for configuring OSPF in Quagga are very similar to the IOS commands:

```
PC1% telnet localhost 2604 Password: <login password>
ospfd> enable
ospfd# configure terminal
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
```

   Note that the command to enable OSPF (`router ospf`) does not use a process-id. Also, there is an explicit command to set the router-id. The latter is necessary since Quagga does not assign a default value for the router-id. In Quagga, the router-id must be explicitly set. In this exercise we use the IP address of the Ethernet interface *eth0* as the router-id for the Linux PCs.

7. Repeat the OSPF configuration in Steps 1-6 for all other Linux PCs.

8. When the OSPF configuration is complete, all hosts and routers should be able to communicate with each other. You can test the network configuration by running `traceroute` and `ping` commands on a Linux PC (or trace and ping commands on a Cisco router). When you have verified that the network connection is correct, proceed with the next step.

Exercise 4-C. 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 *eth0*. Set a filter to only display OSPF packets.

2. From PC3, run a `traceroute` command to PC4

```
PC3% traceroute 10.0.7.7
```

   Confirm from the output and Figure 4.4, whether the path from PC3 to PC4 includes Router 3 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.

```
PC3% ping 10.0.7.7
```

4. If the path from PC3 to IP address 10.0.7.7 from Step 2 included Router3, then disconnect the Ethernet cable of the *Ethernet0/1* interface of Router3. Otherwise, disconnect the Ethernet cable of the *Ethernet0/1* 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:

Question 4.C.1.a)

How quickly are OSPF messages sent after the cable is disconnected?

If we take a look at the pcap (traces/4-C.10.pcapng), we can see that the ping from packet 58 and onwards does not receive a reply. This happened at ≈ 44.6 seconds into capturing. The first OSPF update packet captured after this is at ≈ 52.8 seconds. It took OSPF approximately 8 seconds to notice the disconnection and start sending updates.

Question 4.C.1.b)

How many OSPF messages are sent?

We captured 7 update packets, 8 hello packets and 5 acknowledgement packets before the pings were succesful again.

Question 4.C.1.c)

Which type of OSPF packet is used for flooding link state information?

The LS update packets (LS standing for Link State) are responsible for flooding the link state information on the network.

Question 4.C.1.d)

Describe the flooding of LSAs to all routers.

LSA's are flooded over the network via multicast. These packets get acknowledged by the other routers (LS Acknowledge).

Question 4.C.1.e)

Which type of encapsulation is used for OSPF packets (TCP, UDP or other)?

OSPF does not use either TCP or UDP. It uses its own type of transport layer correction and error detection. The data for OSPF is directly encapsulated into IP packets, using its own header after the standard ipv4 header.

Question 4.C.1.f)

What is the destination address of OSPF packets?

The destination address for OSPF packets is the multicast address $224.0.0.5$.

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 with `Ctrl-c` and save the ping statistics output (i.e. the data that appears at the bottom of the terminal screen when you stop the ping process).

Question 4.C.2)

Include your answer on the convergence time from Step 7. 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.)

```
--- 10.0.7.7 ping statistics ---
53 packets transmitted, 13 received, +4 errors, 75% packet loss, time 52304ms
rtt min/avg/max/mdev = 2.022/2.098/2.209/0.074 ms
```

traces/4–C.7.txt

In total, there were 40 packets lost $(53 - 13)$. This means that OSPF took ≈ 40 seconds to update the routing tables.

8. Issue another `traceroute` 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 and on all Linux PCs, and verify that all routers indeed have the same link state database. On the Linux PCs, open a Telnet session to the ospfd process, and then type

```
ospfd# show ip ospf database router
```

On the Cisco routers, simply type

```
Router1# show ip ospf database
```

Save the output of the link state databases to a file.

Question 4.C.3)

Can you confirm that the link state databases are identical? Compare the output of the command `show ip ospf database` from the Cisco routers and the Linux PCs.

```

       OSPF Router with ID (10.0.1.1)


                Router Link States (Area 0.0.0.1)


  LS age: 466
  Options: 0x2  : *|-|-|-|-|-|E|*
  LS Flags: 0x3
  Flags: 0x0
  LS Type: router-LSA
  Link State ID: 10.0.1.1
  Advertising Router: 10.0.1.1
  LS Seq Number: 8000000c
  Checksum: 0xfbde
  Length: 48
   Number of Links: 2

    Link connected to: a Transit Network
     (Link ID) Designated Router address: 10.0.1.2
     (Link Data) Router Interface address: 10.0.1.1
      Number of TOS metrics: 0
       TOS 0 Metric: 10

    Link connected to: a Transit Network
     (Link ID) Designated Router address: 10.0.2.1
     (Link Data) Router Interface address: 10.0.2.1
      Number of TOS metrics: 0
       TOS 0 Metric: 10


  LS age: 294
  Options: 0x2  : *|-|-|-|-|-|E|*
```

```
LS Flags: 0x6
Flags: 0x0
LS Type: router-LSA
Link State ID: 10.0.1.2
Advertising Router: 10.0.1.2
LS Seq Number: 8000000c
Checksum: 0x0dbe
Length: 48
 Number of Links: 2

   Link connected to: a Transit Network
    (Link ID) Designated Router address: 10.0.1.2
    (Link Data) Router Interface address: 10.0.1.2
     Number of TOS metrics: 0
      TOS 0 Metric: 10

   Link connected to: a Transit Network
    (Link ID) Designated Router address: 10.0.5.6
    (Link Data) Router Interface address: 10.0.5.2
     Number of TOS metrics: 0
      TOS 0 Metric: 10


LS age: 224
Options: 0x22 : *|-|DC|-|-|-|E|*
LS Flags: 0x6
Flags: 0x0
LS Type: router-LSA
Link State ID: 10.0.3.3
Advertising Router: 10.0.3.3
LS Seq Number: 80000006
Checksum: 0xfcc3
Length: 48
 Number of Links: 2

   Link connected to: a Transit Network
    (Link ID) Designated Router address: 10.0.2.1
    (Link Data) Router Interface address: 10.0.2.3
     Number of TOS metrics: 0
      TOS 0 Metric: 1

   Link connected to: a Transit Network
    (Link ID) Designated Router address: 10.0.3.4
    (Link Data) Router Interface address: 10.0.3.3
     Number of TOS metrics: 0
      TOS 0 Metric: 1


LS age: 417
Options: 0x2  : *|-|-|-|-|-|E|*
LS Flags: 0x6
Flags: 0x0
LS Type: router-LSA
Link State ID: 10.0.3.4
Advertising Router: 10.0.3.4
LS Seq Number: 80000007
Checksum: 0x09b9
Length: 48
 Number of Links: 2

   Link connected to: a Transit Network
    (Link ID) Designated Router address: 10.0.3.4
    (Link Data) Router Interface address: 10.0.3.4
     Number of TOS metrics: 0
      TOS 0 Metric: 10

   Link connected to: a Transit Network
```

```
     (Link ID) Designated Router address: 10.0.4.4
     (Link Data) Router Interface address: 10.0.4.4
      Number of TOS metrics: 0
       TOS 0 Metric: 10


  LS age: 299
  Options: 0x22 : *|-|DC|-|-|-|E|*
  LS Flags: 0x6
  Flags: 0x0
  LS Type: router-LSA
  Link State ID: 10.0.4.5
  Advertising Router: 10.0.4.5
  LS Seq Number: 80000004
  Checksum: 0x4d69
  Length: 48
   Number of Links: 2

    Link connected to: a Transit Network
     (Link ID) Designated Router address: 10.0.2.1
     (Link Data) Router Interface address: 10.0.2.5
      Number of TOS metrics: 0
       TOS 0 Metric: 1

    Link connected to: a Transit Network
     (Link ID) Designated Router address: 10.0.4.4
     (Link Data) Router Interface address: 10.0.4.5
      Number of TOS metrics: 0
       TOS 0 Metric: 1


  LS age: 298
  Options: 0x22 : *|-|DC|-|-|-|E|*
  LS Flags: 0x6
  Flags: 0x0
  LS Type: router-LSA
  Link State ID: 10.0.6.6
  Advertising Router: 10.0.6.6
  LS Seq Number: 80000009
  Checksum: 0x2671
  Length: 48
   Number of Links: 2

    Link connected to: a Transit Network
     (Link ID) Designated Router address: 10.0.6.7
     (Link Data) Router Interface address: 10.0.6.6
      Number of TOS metrics: 0
       TOS 0 Metric: 1

    Link connected to: a Transit Network
     (Link ID) Designated Router address: 10.0.5.6
     (Link Data) Router Interface address: 10.0.5.6
      Number of TOS metrics: 0
       TOS 0 Metric: 1


  LS age: 393
  Options: 0x2  : *|-|-|-|-|-|E|*
  LS Flags: 0x6
  Flags: 0x0
  LS Type: router-LSA
  Link State ID: 10.0.6.7
  Advertising Router: 10.0.6.7
  LS Seq Number: 80000003
  Checksum: 0x5052
  Length: 48
   Number of Links: 2
```

```

    Link connected to: a Transit Network
     (Link ID) Designated Router address: 10.0.6.7
     (Link Data) Router Interface address: 10.0.6.7
      Number of TOS metrics: 0
       TOS 0 Metric: 10

    Link connected to: a Transit Network
     (Link ID) Designated Router address: 10.0.7.7
     (Link Data) Router Interface address: 10.0.7.7
      Number of TOS metrics: 0
       TOS 0 Metric: 10


LS age: 64
Options: 0x22 : *|-|DC|-|-|-|E|*
LS Flags: 0x6
Flags: 0x0
LS Type: router-LSA
Link State ID: 10.0.7.8
Advertising Router: 10.0.7.8
LS Seq Number: 80000007
Checksum: 0x0abf
Length: 36
 Number of Links: 1

    Link connected to: a Transit Network
     (Link ID) Designated Router address: 10.0.7.7
     (Link Data) Router Interface address: 10.0.7.8
      Number of TOS metrics: 0
       TOS 0 Metric: 1
LS age: 393
Options: 0x2  : *|-|-|-|-|-|E|*
LS Flags: 0x6
Flags: 0x0
LS Type: router-LSA
Link State ID: 10.0.6.7
Advertising Router: 10.0.6.7
LS Seq Number: 80000003
Checksum: 0x5052
Length: 48
 Number of Links: 2

    Link connected to: a Transit Network
     (Link ID) Designated Router address: 10.0.6.7
     (Link Data) Router Interface address: 10.0.6.7
      Number of TOS metrics: 0
       TOS 0 Metric: 10

    Link connected to: a Transit Network
     (Link ID) Designated Router address: 10.0.7.7
     (Link Data) Router Interface address: 10.0.7.7
      Number of TOS metrics: 0
       TOS 0 Metric: 10


LS age: 64
Options: 0x22 : *|-|DC|-|-|-|E|*
LS Flags: 0x6
Flags: 0x0
LS Type: router-LSA
Link State ID: 10.0.7.8
Advertising Router: 10.0.7.8
LS Seq Number: 80000007
Checksum: 0x0abf
Length: 36
 Number of Links: 1
```

```
    Link connected to: a Transit Network
     (Link ID) Designated Router address: 10.0.7.7
     (Link Data) Router Interface address: 10.0.7.8
      Number of TOS metrics: 0
       TOS 0 Metric: 1
```

traces/4–C.9.PC1.txt

```
            OSPF Router with ID (10.0.3.3) (Process ID 1)

                Router Link States (Area 1)

Link ID         ADV Router      Age         Seq#       Checksum Link count
10.0.1.1        10.0.1.1        473         0x8000000C 0x00FBDE 2
10.0.1.2        10.0.1.2        301         0x8000000C 0x000DBE 2
10.0.3.3        10.0.3.3        228         0x80000006 0x00FCC3 2
10.0.3.4        10.0.3.4        422         0x80000007 0x0009B9 2
10.0.4.5        10.0.4.5        305         0x80000004 0x004D69 2
10.0.6.6        10.0.6.6        304         0x80000009 0x002671 2
10.0.6.7        10.0.6.7        400         0x80000003 0x005052 2
10.0.7.8        10.0.7.8        70          0x80000007 0x000ABF 1

                Net Link States (Area 1)

Link ID         ADV Router      Age         Seq#       Checksum
10.0.1.2        10.0.1.2        556         0x80000003 0x005FC5
10.0.2.1        10.0.1.1        393         0x80000005 0x00C544
10.0.3.4        10.0.3.4        382         0x80000004 0x00759E
10.0.4.4        10.0.3.4        2311        0x80000001 0x00957D
10.0.5.6        10.0.6.6        305         0x80000001 0x00519A
10.0.6.7        10.0.6.7        2015        0x80000001 0x009F5F
10.0.7.7        10.0.6.7        402         0x80000001 0x00B941
```

traces/4–C.9.Router1.txt

Other than the age of the different routes, the Router Link States are the same for all the different databases. What is different though, is that the Router databases also contain information about the Net Link States (non point to point links).

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 (Selecting the Print Detail option).

Question 4.C.4)

From your saved Wireshark output, include one packet from each of the different OSPF packet types that you have observed. (Include only one packet from each type!)

Hello packet

```
      1 0.000000000     10.0.1.2              224.0.0.5             OSPF     82
          Hello Packet

Frame 1: 82 bytes on wire (656 bits), 82 bytes captured (656 bits) on interface 0
    Interface id: 0 (eth0)
    Encapsulation type: Ethernet (1)
    Arrival Time: Mar 23, 2017 14:49:14.126448537 CET
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1490276954.126448537 seconds
    [Time delta from previous captured frame: 0.000000000 seconds]
    [Time delta from previous displayed frame: 0.000000000 seconds]
    [Time since reference or first frame: 0.000000000 seconds]
```

```
    Frame Number: 1
    Frame Length: 82 bytes (656 bits)
    Capture Length: 82 bytes (656 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ethertype:ip:ospf]
    [Coloring Rule Name: Routing]
    [Coloring Rule String: hsrp || eigrp || ospf || bgp || cdp || vrrp || carp ||
        gvrp || igmp || ismp]
Ethernet II, Src: 68:05:ca:36:31:f0, Dst: 01:00:5e:00:00:05
    Destination: 01:00:5e:00:00:05
        Address: 01:00:5e:00:00:05
        .... ..0. .... .... .... .... = LG bit: Globally unique address (factory
            default)
        .... ...1 .... .... .... .... = IG bit: Group address (multicast/broadcast)
    Source: 68:05:ca:36:31:f0
        Address: 68:05:ca:36:31:f0
        .... ..0. .... .... .... .... = LG bit: Globally unique address (factory
            default)
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: IPv4 (0x0800)
Internet Protocol Version 4, Src: 10.0.1.2, Dst: 224.0.0.5
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes
    Differentiated Services Field: 0xc0 (DSCP: CS6, ECN: Not-ECT)
        1100 00.. = Differentiated Services Codepoint: Class Selector 6 (48)
        .... ..00 = Explicit Congestion Notification: Not ECN-Capable Transport (0)
    Total Length: 68
    Identification: 0xcbda (52186)
    Flags: 0x00
        0... .... = Reserved bit: Not set
        .0.. .... = Don't fragment: Not set
        ..0. .... = More fragments: Not set
    Fragment offset: 0
    Time to live: 1
    Protocol: OSPF IGP (89)
    Header checksum: 0x01c0 [validation disabled]
        [Good: False]
        [Bad: False]
    Source: 10.0.1.2
    Destination: 224.0.0.5
    [Source GeoIP: Unknown]
    [Destination GeoIP: Unknown]
Open Shortest Path First
    OSPF Header
        Version: 2
        Message Type: Hello Packet (1)
        Packet Length: 48
        Source OSPF Router: 10.0.1.2
        Area ID: 0.0.0.1
        Checksum: 0xd093 [correct]
        Auth Type: Null (0)
        Auth Data (none): 0000000000000000
    OSPF Hello Packet
        Network Mask: 255.255.255.0
        Hello Interval [sec]: 10
        Options: 0x02 ((E) External Routing)
            0... .... = DN: Not set
            .0.. .... = O: Not set
            ..0. .... = (DC) Demand Circuits: Not supported
            ...0 .... = (L) LLS Data block: Not Present
            .... 0... = (N) NSSA: Not supported
            .... .0.. = (MC) Multicast: Not capable
            .... ..1. = (E) External Routing: Capable
            .... ...0 = (MT) Multi-Topology Routing: No
        Router Priority: 1
        Router Dead Interval [sec]: 40
```

```
        Designated Router: 10.0.1.2
        Backup Designated Router: 10.0.1.1
        Active Neighbor: 10.0.1.1
```

LS Update packet

```
      2 1.756152656    10.0.1.1              224.0.0.5             OSPF     110
          LS Update

Frame 2: 110 bytes on wire (880 bits), 110 bytes captured (880 bits) on interface 0
    Interface id: 0 (eth0)
    Encapsulation type: Ethernet (1)
    Arrival Time: Mar 23, 2017 14:49:15.882601193 CET
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1490276955.882601193 seconds
    [Time delta from previous captured frame: 1.756152656 seconds]
    [Time delta from previous displayed frame: 1.756152656 seconds]
    [Time since reference or first frame: 1.756152656 seconds]
    Frame Number: 2
    Frame Length: 110 bytes (880 bits)
    Capture Length: 110 bytes (880 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ethertype:ip:ospf]
    [Coloring Rule Name: OSPF State Change]
    [Coloring Rule String: ospf.msg != 1]
Ethernet II, Src: 68:05:ca:36:33:a0, Dst: 01:00:5e:00:00:05
    Destination: 01:00:5e:00:00:05
        Address: 01:00:5e:00:00:05
        .... ..0. .... .... .... .... = LG bit: Globally unique address (factory
            default)
        .... ...1 .... .... .... .... = IG bit: Group address (multicast/broadcast)
    Source: 68:05:ca:36:33:a0
        Address: 68:05:ca:36:33:a0
        .... ..0. .... .... .... .... = LG bit: Globally unique address (factory
            default)
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: IPv4 (0x0800)
Internet Protocol Version 4, Src: 10.0.1.1, Dst: 224.0.0.5
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes
    Differentiated Services Field: 0xc0 (DSCP: CS6, ECN: Not-ECT)
        1100 00.. = Differentiated Services Codepoint: Class Selector 6 (48)
        .... ..00 = Explicit Congestion Notification: Not ECN-Capable Transport (0)
    Total Length: 96
    Identification: 0xca0d (51725)
    Flags: 0x00
        0... .... = Reserved bit: Not set
        .0.. .... = Don't fragment: Not set
        ..0. .... = More fragments: Not set
    Fragment offset: 0
    Time to live: 1
    Protocol: OSPF IGP (89)
    Header checksum: 0x0372 [validation disabled]
        [Good: False]
        [Bad: False]
    Source: 10.0.1.1
    Destination: 224.0.0.5
    [Source GeoIP: Unknown]
    [Destination GeoIP: Unknown]
Open Shortest Path First
    OSPF Header
        Version: 2
        Message Type: LS Update (4)
        Packet Length: 76
        Source OSPF Router: 10.0.1.1
```

```
        Area ID: 0.0.0.1
        Checksum: 0x039a [correct]
        Auth Type: Null (0)
        Auth Data (none): 0000000000000000
    LS Update Packet
        Number of LSAs: 1
        LSA-type 1 (Router-LSA), len 48
            .000 0000 0000 0010 = LS Age (seconds): 2
            0... .... .... .... = Do Not Age Flag: 0
            Options: 0x22 ((DC) Demand Circuits, (E) External Routing)
                0... .... = DN: Not set
                .0.. .... = O: Not set
                ..1. .... = (DC) Demand Circuits: Supported
                ...0 .... = (L) LLS Data block: Not Present
                .... 0... = (N) NSSA: Not supported
                .... .0.. = (MC) Multicast: Not capable
                .... ..1. = (E) External Routing: Capable
                .... ...0 = (MT) Multi-Topology Routing: No
            LS Type: Router-LSA (1)
            Link State ID: 10.0.3.3
            Advertising Router: 10.0.3.3
            Sequence Number: 0x80000006
            Checksum: 0xfcc3
            Length: 48
            Flags: 0x00
                .... .0.. = (V) Virtual link endpoint: No
                .... ..0. = (E) AS boundary router: No
                .... ...0 = (B) Area border router: No
            Number of Links: 2
            Type: Transit   ID: 10.0.2.1        Data: 10.0.2.3        Metric: 1
                Link ID: 10.0.2.1 - IP address of Designated Router
                Link Data: 10.0.2.3
                Link Type: 2 - Connection to a transit network
                Number of Metrics: 0 - TOS
                0 Metric: 1
            Type: Transit   ID: 10.0.3.4        Data: 10.0.3.3        Metric: 1
                Link ID: 10.0.3.4 - IP address of Designated Router
                Link Data: 10.0.3.3
                Link Type: 2 - Connection to a transit network
                Number of Metrics: 0 - TOS
                0 Metric: 1
```

LS Acknowledgment packet

```
      3 2.191991367    10.0.1.2             224.0.0.5            OSPF     78
          LS Acknowledge

Frame 3: 78 bytes on wire (624 bits), 78 bytes captured (624 bits) on interface 0
    Interface id: 0 (eth0)
    Encapsulation type: Ethernet (1)
    Arrival Time: Mar 23, 2017 14:49:16.318439904 CET
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1490276956.318439904 seconds
    [Time delta from previous captured frame: 0.435838711 seconds]
    [Time delta from previous displayed frame: 0.435838711 seconds]
    [Time since reference or first frame: 2.191991367 seconds]
    Frame Number: 3
    Frame Length: 78 bytes (624 bits)
    Capture Length: 78 bytes (624 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ethertype:ip:ospf]
    [Coloring Rule Name: OSPF State Change]
    [Coloring Rule String: ospf.msg != 1]
Ethernet II, Src: 68:05:ca:36:31:f0, Dst: 01:00:5e:00:00:05
```

```
    Destination: 01:00:5e:00:00:05
        Address: 01:00:5e:00:00:05
        .... ..0. .... .... .... .... = LG bit: Globally unique address (factory
            default)
        .... ...1 .... .... .... .... = IG bit: Group address (multicast/broadcast)
    Source: 68:05:ca:36:31:f0
        Address: 68:05:ca:36:31:f0
        .... ..0. .... .... .... .... = LG bit: Globally unique address (factory
            default)
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: IPv4 (0x0800)
Internet Protocol Version 4, Src: 10.0.1.2, Dst: 224.0.0.5
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes
    Differentiated Services Field: 0xc0 (DSCP: CS6, ECN: Not-ECT)
        1100 00.. = Differentiated Services Codepoint: Class Selector 6 (48)
        .... ..00 = Explicit Congestion Notification: Not ECN-Capable Transport (0)
    Total Length: 64
    Identification: 0xcbdd (52189)
    Flags: 0x00
        0... .... = Reserved bit: Not set
        .0.. .... = Don't fragment: Not set
        ..0. .... = More fragments: Not set
    Fragment offset: 0
    Time to live: 1
    Protocol: OSPF IGP (89)
    Header checksum: 0x01c1 [validation disabled]
        [Good: False]
        [Bad: False]
    Source: 10.0.1.2
    Destination: 224.0.0.5
    [Source GeoIP: Unknown]
    [Destination GeoIP: Unknown]
Open Shortest Path First
    OSPF Header
        Version: 2
        Message Type: LS Acknowledge (5)
        Packet Length: 44
        Source OSPF Router: 10.0.1.2
        Area ID: 0.0.0.1
        Checksum: 0x39c8 [correct]
        Auth Type: Null (0)
        Auth Data (none): 0000000000000000
    LSA-type 1 (Router-LSA), len 48
        .000 0000 0000 0010 = LS Age (seconds): 2
        0... .... .... .... = Do Not Age Flag: 0
        Options: 0x22 ((DC) Demand Circuits, (E) External Routing)
            0... .... = DN: Not set
            .0.. .... = O: Not set
            ..1. .... = (DC) Demand Circuits: Supported
            ...0 .... = (L) LLS Data block: Not Present
            .... 0... = (N) NSSA: Not supported
            .... .0.. = (MC) Multicast: Not capable
            .... ..1. = (E) External Routing: Capable
            .... ...0 = (MT) Multi-Topology Routing: No
        LS Type: Router-LSA (1)
        Link State ID: 10.0.3.3
        Advertising Router: 10.0.3.3
        Sequence Number: 0x80000006
        Checksum: 0xfcc3
        Length: 48
```

Question 4.C.5)

Include the output of the link state database of PC2.

```
       OSPF Router with ID (10.0.1.2)


                Router Link States (Area 0.0.0.1)

  LS age: 597
  Options: 0x2  : *|-|-|-|-|-|E|*
  LS Flags: 0x6
  Flags: 0x0
  LS Type: router-LSA
  Link State ID: 10.0.1.1
  Advertising Router: 10.0.1.1
  LS Seq Number: 8000000c
  Checksum: 0xfbde
  Length: 48
   Number of Links: 2

    Link connected to: a Transit Network
     (Link ID) Designated Router address: 10.0.1.2
     (Link Data) Router Interface address: 10.0.1.1
      Number of TOS metrics: 0
       TOS 0 Metric: 10

    Link connected to: a Transit Network
     (Link ID) Designated Router address: 10.0.2.1
     (Link Data) Router Interface address: 10.0.2.1
      Number of TOS metrics: 0
       TOS 0 Metric: 10


  LS age: 423
  Options: 0x2  : *|-|-|-|-|-|E|*
  LS Flags: 0x3
  Flags: 0x0
  LS Type: router-LSA
  Link State ID: 10.0.1.2
  Advertising Router: 10.0.1.2
  LS Seq Number: 8000000c
  Checksum: 0x0dbe
  Length: 48
   Number of Links: 2

    Link connected to: a Transit Network
     (Link ID) Designated Router address: 10.0.1.2
     (Link Data) Router Interface address: 10.0.1.2
      Number of TOS metrics: 0
       TOS 0 Metric: 10

    Link connected to: a Transit Network
     (Link ID) Designated Router address: 10.0.5.6
     (Link Data) Router Interface address: 10.0.5.2
      Number of TOS metrics: 0
       TOS 0 Metric: 10


  LS age: 354
  Options: 0x22 : *|-|DC|-|-|-|E|*
  LS Flags: 0x6
  Flags: 0x0
  LS Type: router-LSA
  Link State ID: 10.0.3.3
  Advertising Router: 10.0.3.3
  LS Seq Number: 80000006
  Checksum: 0xfcc3
  Length: 48
   Number of Links: 2
```

```
      Link connected to: a Transit Network
       (Link ID) Designated Router address: 10.0.2.1
       (Link Data) Router Interface address: 10.0.2.3
        Number of TOS metrics: 0
         TOS 0 Metric: 1

      Link connected to: a Transit Network
       (Link ID) Designated Router address: 10.0.3.4
       (Link Data) Router Interface address: 10.0.3.3
        Number of TOS metrics: 0
         TOS 0 Metric: 1


  LS age: 548
  Options: 0x2  : *|-|-|-|-|-|E|*
  LS Flags: 0x6
  Flags: 0x0
  LS Type: router-LSA
  Link State ID: 10.0.3.4
  Advertising Router: 10.0.3.4
  LS Seq Number: 80000007
  Checksum: 0x09b9
  Length: 48
   Number of Links: 2

      Link connected to: a Transit Network
       (Link ID) Designated Router address: 10.0.3.4
       (Link Data) Router Interface address: 10.0.3.4
        Number of TOS metrics: 0
         TOS 0 Metric: 10

      Link connected to: a Transit Network
       (Link ID) Designated Router address: 10.0.4.4
       (Link Data) Router Interface address: 10.0.4.4
        Number of TOS metrics: 0
         TOS 0 Metric: 10


  LS age: 430
  Options: 0x22 : *|-|DC|-|-|-|E|*
  LS Flags: 0x6
  Flags: 0x0
  LS Type: router-LSA
  Link State ID: 10.0.4.5
  Advertising Router: 10.0.4.5
  LS Seq Number: 80000004
  Checksum: 0x4d69
  Length: 48
   Number of Links: 2

      Link connected to: a Transit Network
       (Link ID) Designated Router address: 10.0.2.1
       (Link Data) Router Interface address: 10.0.2.5
        Number of TOS metrics: 0
         TOS 0 Metric: 1

      Link connected to: a Transit Network
       (Link ID) Designated Router address: 10.0.4.4
       (Link Data) Router Interface address: 10.0.4.5
        Number of TOS metrics: 0
         TOS 0 Metric: 1


  LS age: 426
  Options: 0x22 : *|-|DC|-|-|-|E|*
  LS Flags: 0x6
  Flags: 0x0
```

```
LS Type: router-LSA
Link State ID: 10.0.6.6
Advertising Router: 10.0.6.6
LS Seq Number: 80000009
Checksum: 0x2671
Length: 48
 Number of Links: 2

   Link connected to: a Transit Network
    (Link ID) Designated Router address: 10.0.6.7
    (Link Data) Router Interface address: 10.0.6.6
     Number of TOS metrics: 0
      TOS 0 Metric: 1

   Link connected to: a Transit Network
    (Link ID) Designated Router address: 10.0.5.6
    (Link Data) Router Interface address: 10.0.5.6
     Number of TOS metrics: 0
      TOS 0 Metric: 1


LS age: 522
Options: 0x2  : *|-|-|-|-|-|E|*
LS Flags: 0x6
Flags: 0x0
LS Type: router-LSA
Link State ID: 10.0.6.7
Advertising Router: 10.0.6.7
LS Seq Number: 80000003
Checksum: 0x5052
Length: 48
 Number of Links: 2

   Link connected to: a Transit Network
    (Link ID) Designated Router address: 10.0.6.7
    (Link Data) Router Interface address: 10.0.6.7
     Number of TOS metrics: 0
      TOS 0 Metric: 10

   Link connected to: a Transit Network
    (Link ID) Designated Router address: 10.0.7.7
    (Link Data) Router Interface address: 10.0.7.7
     Number of TOS metrics: 0
      TOS 0 Metric: 10


LS age: 192
Options: 0x22 : *|-|DC|-|-|-|E|*
LS Flags: 0x6
Flags: 0x0
LS Type: router-LSA
Link State ID: 10.0.7.8
Advertising Router: 10.0.7.8
LS Seq Number: 80000007
Checksum: 0x0abf
Length: 36
 Number of Links: 1

   Link connected to: a Transit Network
    (Link ID) Designated Router address: 10.0.7.7
    (Link Data) Router Interface address: 10.0.7.8
     Number of TOS metrics: 0
      TOS 0 Metric: 1
```

traces/4–C.9.PC2.txt

Question 4.C.6)

Pick a single link state advertisement packet captured by Wireshark, and describe how to interpret the information contained in the link state advertisement.

The OSPF part of the LS Update packet from question 4.C.4.

```
Open Shortest Path First
    OSPF Header
        Version: 2
        Message Type: LS Update (4)
        Packet Length: 76
        Source OSPF Router: 10.0.1.1
        Area ID: 0.0.0.1
        Checksum: 0x039a [correct]
        Auth Type: Null (0)
        Auth Data (none): 0000000000000000
    LS Update Packet
        Number of LSAs: 1
        LSA-type 1 (Router-LSA), len 48
            .000 0000 0000 0010 = LS Age (seconds): 2
            0... .... .... .... = Do Not Age Flag: 0
            Options: 0x22 ((DC) Demand Circuits, (E) External Routing)
                0... .... = DN: Not set
                .0.. .... = O: Not set
                ..1. .... = (DC) Demand Circuits: Supported
                ...0 .... = (L) LLS Data block: Not Present
                .... 0... = (N) NSSA: Not supported
                .... .0.. = (MC) Multicast: Not capable
                .... ..1. = (E) External Routing: Capable
                .... ...0 = (MT) Multi-Topology Routing: No
            LS Type: Router-LSA (1)
            Link State ID: 10.0.3.3
            Advertising Router: 10.0.3.3
            Sequence Number: 0x80000006
            Checksum: 0xfcc3
            Length: 48
            Flags: 0x00
                .... .0.. = (V) Virtual link endpoint: No
                .... ..0. = (E) AS boundary router: No
                .... ...0 = (B) Area border router: No
            Number of Links: 2
            Type: Transit   ID: 10.0.2.1        Data: 10.0.2.3        Metric: 1
                Link ID: 10.0.2.1 - IP address of Designated Router
                Link Data: 10.0.2.3
                Link Type: 2 - Connection to a transit network
                Number of Metrics: 0 - TOS
                0 Metric: 1
            Type: Transit   ID: 10.0.3.4        Data: 10.0.3.3        Metric: 1
                Link ID: 10.0.3.4 - IP address of Designated Router
                Link Data: 10.0.3.3
                Link Type: 2 - Connection to a transit network
                Number of Metrics: 0 - TOS
                0 Metric: 1
```

The OSPF part of the packet is divided into 2 parts: the header and the update packet itself. The OSPF header contains information about the used version, the message type, the packets length, the ip address of the source router, the area ID, the checksum and authentication data.

The update packet itself contains the LSAs itself. In this packet 1 LSA is included. The LSA type for this particular advertisement is Router-LSA, meaning it contains information about directly connected links to the router in the area of that router. As for options, demand circuits and external routing is specified. Demand circuits surpresses periodic hello and update packets. The external routing option means that the source router is capable of accepting external LSAs.

After the options, the Link State ID and Advertising Router are specified (in this case $10.0.3.3$). Following this we have the sequence number, the checksum and the length of the packet.
Lastly, the connected links are specified. The link ID, link data, link type, number of metrics and metric are specified for each link. The link ID and link data have different meanings depending on the link type. In our case, for link type 2, the link ID is the IP address of the Designated Router. The link data is the IP address of the interface (to the network) from the Advertising Router (e.g. for the first link: interface vlan1 from Router1). The metric for the links is a cost calculated by ospf, depending on the bandwidth of the interface.

Hierarchical Routing in OSPF

The concept of areas in OSPF can be used to construct a hierarchical routing scheme. When the network is partitioned into multiple areas, then routers must have complete topology information only about routers in the same area, 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 builds 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.

The configuration in this part is shown in Figure 4.5. Here, the network from Part 4 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.

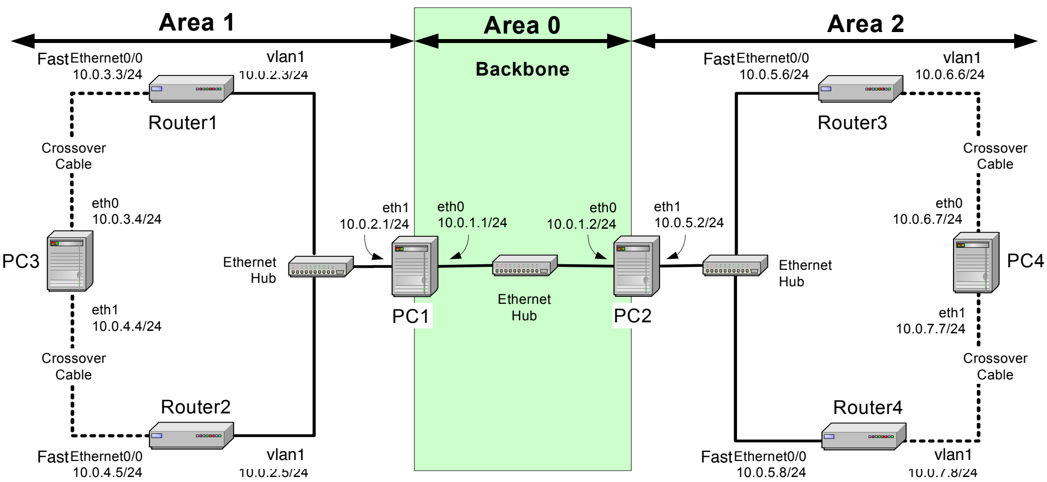

Figure 4.5: Network configuration for Part 5

In the following exercises you define the areas, and then observe how the link state databases are built.

Exercise 5. Defining multiple areas in OSPF

1. Restart the `zebra` and `ospfd` processes on all four Linux PCs. Use the same `daemons` configuration file as used in the previous exercise.

```
PC1% /etc/init.d/quagga restart
```

2. Start Wireshark on PC1 and capture traffic on interface *eth0*.

3. Change the Area IDs of the Cisco routers and the PCs. On each system, the directly connected networks are assigned to an area with a 24-bit prefix. Here are the configurations for PC3, PC1, and Router 1. The other configurations are similar. PC3, which belongs to only one area, is configured as follows:

```
PC3% telnet localhost 2604 Password: <login password>
ospfd> enable
```

```
ospfd# configure terminal
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, belongs to two areas, and is configured as follows:

```
PC1% telnet localhost 2604 Password: <login password>
ospfd> enable
ospfd# configure terminal
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
```

The configuration of Router 1 is as follows:

```
Router1# configure terminal
Router1(config)# no router ospf 1
Router1(config)# router ospf 1
Router1(config-router)# network 10.0.3.0 0.0.0.255 area 1
Router1(config-router)# network 10.0.2.0 0.0.0.255 area 1
Router1(config-router)# end
Router1# clear ip ospf 1 process
```

4. Once the routing tables have converged, test the network configuration with the commands `traceroute` and `ping` on the Linux PCs, and the commands trace and ping on the Cisco routers. All hosts and routers should be able to communicate with each other.

5. Save the link state database on all Cisco routers and on all Linux PCs. On the Linux PCs, open a Telnet session to the ospfd process, and then type

```
ospfd# show ip ospf database router
```

On the Cisco routers, type

```
Router1# show ip ospf database
```

Save the output of the link state databases to a file.

Question 5.1.a)

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

```
       OSPF Router with ID (10.0.1.1)


                Router Link States (Area 0.0.0.0)

  LS age: 385
  Options: 0x2  : *|-|-|-|-|-|E|*
  LS Flags: 0x3
  Flags: 0x1  : ABR
```

```
LS Type: router-LSA
Link State ID: 10.0.1.1
Advertising Router: 10.0.1.1
LS Seq Number: 80000004
Checksum: 0x51c4
Length: 36
 Number of Links: 1

  Link connected to: a Transit Network
   (Link ID) Designated Router address: 10.0.1.1
   (Link Data) Router Interface address: 10.0.1.1
    Number of TOS metrics: 0
     TOS 0 Metric: 10


LS age: 383
Options: 0x2  : *|-|-|-|-|-|E|*
LS Flags: 0x6
Flags: 0x1 : ABR
LS Type: router-LSA
Link State ID: 10.0.1.2
Advertising Router: 10.0.1.2
LS Seq Number: 80000004
Checksum: 0x4fc3
Length: 36
 Number of Links: 1

  Link connected to: a Transit Network
   (Link ID) Designated Router address: 10.0.1.1
   (Link Data) Router Interface address: 10.0.1.2
    Number of TOS metrics: 0
     TOS 0 Metric: 10



               Router Link States (Area 0.0.0.1)

LS age: 465
Options: 0x2  : *|-|-|-|-|-|E|*
LS Flags: 0x3
Flags: 0x1 : ABR
LS Type: router-LSA
Link State ID: 10.0.1.1
Advertising Router: 10.0.1.1
LS Seq Number: 80000005
Checksum: 0x7997
Length: 36
 Number of Links: 1

  Link connected to: a Transit Network
   (Link ID) Designated Router address: 10.0.2.3
   (Link Data) Router Interface address: 10.0.2.1
    Number of TOS metrics: 0
     TOS 0 Metric: 10


LS age: 367
Options: 0x22 : *|-|DC|-|-|-|E|*
LS Flags: 0x6
Flags: 0x0
LS Type: router-LSA
Link State ID: 10.0.3.3
Advertising Router: 10.0.3.3
LS Seq Number: 80000004
Checksum: 0xfec2
Length: 48
 Number of Links: 2
```

```

    Link connected to: a Transit Network
     (Link ID) Designated Router address: 10.0.2.3
     (Link Data) Router Interface address: 10.0.2.3
      Number of TOS metrics: 0
       TOS 0 Metric: 1

    Link connected to: a Transit Network
     (Link ID) Designated Router address: 10.0.3.3
     (Link Data) Router Interface address: 10.0.3.3
      Number of TOS metrics: 0
       TOS 0 Metric: 1


  LS age: 362
  Options: 0x2  : *|-|-|-|-|-|E|*
  LS Flags: 0x6
  Flags: 0x0
  LS Type: router-LSA
  Link State ID: 10.0.3.4
  Advertising Router: 10.0.3.4
  LS Seq Number: 80000006
  Checksum: 0x17ac
  Length: 48
   Number of Links: 2

    Link connected to: a Transit Network
     (Link ID) Designated Router address: 10.0.3.3
     (Link Data) Router Interface address: 10.0.3.4
      Number of TOS metrics: 0
       TOS 0 Metric: 10

    Link connected to: a Transit Network
     (Link ID) Designated Router address: 10.0.4.5
     (Link Data) Router Interface address: 10.0.4.4
      Number of TOS metrics: 0
       TOS 0 Metric: 10


  LS age: 352
  Options: 0x22 : *|-|DC|-|-|-|E|*
  LS Flags: 0x6
  Flags: 0x0
  LS Type: router-LSA
  Link State ID: 10.0.4.5
  Advertising Router: 10.0.4.5
  LS Seq Number: 80000005
  Checksum: 0x753d
  Length: 48
   Number of Links: 2

    Link connected to: a Transit Network
     (Link ID) Designated Router address: 10.0.2.3
     (Link Data) Router Interface address: 10.0.2.5
      Number of TOS metrics: 0
       TOS 0 Metric: 1

    Link connected to: a Transit Network
     (Link ID) Designated Router address: 10.0.4.5
     (Link Data) Router Interface address: 10.0.4.5
      Number of TOS metrics: 0
       TOS 0 Metric: 1
```

traces/5.5.PC1.txt

```
            OSPF Router with ID (10.0.3.3) (Process ID 1)
```

```
                 Router Link States (Area 1)

Link ID          ADV Router       Age         Seq#        Checksum Link count
10.0.1.1         10.0.1.1         344         0x80000005  0x007997 1
10.0.3.3         10.0.3.3         243         0x80000004  0x00FEC2 2
10.0.3.4         10.0.3.4         239         0x80000006  0x0017AC 2
10.0.4.5         10.0.4.5         229         0x80000005  0x00753D 2

                 Net Link States (Area 1)

Link ID          ADV Router       Age         Seq#        Checksum
10.0.2.3         10.0.3.3         354         0x80000002  0x00AB3B
10.0.3.3         10.0.3.3         243         0x80000001  0x00A94F
10.0.4.5         10.0.4.5         230         0x80000001  0x00945B

                 Summary Net Link States (Area 1)

Link ID          ADV Router       Age         Seq#        Checksum
10.0.1.0         10.0.1.1         344         0x80000001  0x00DA61
10.0.5.0         10.0.1.1         255         0x80000001  0x00131B
10.0.6.0         10.0.1.1         255         0x80000001  0x00121A
10.0.7.0         10.0.1.1         256         0x80000001  0x000724
```

traces/5.5.Router1.txt

The router databases now also contain summary net link states, because of the addition of more than 1 area in the network topology. These entries are constructed by the border routers.
The PC databases are adjusted to contain information about the area they belong to, as well as information about the border routers in their area.

Question 5.1.b)

Which information do routers in Area 1 have about Area 2? Which information do they have about the backbone area (Area 0)?

Routers in Area 1 only have information about the subnets from area 2 and area 0. It should be noted that they do not know which subnets belong to which area, they just know how to direct traffic to them. This can be seen by looking at the summary net link states. If they want to send traffic to those subnets, they need to send the traffic to advertising router 10.0.1.1 (PC1).

Question 5.1.c)

How much information do the routers in the backbone area (Area 0) have about the topology of Area 1 and Area 2?
The routers in the backbone area only have information about the backbone area, and the other area they border with. In other words, PC1 has information about area 0 and 1, PC2 has information about area 0 and 2.

Question 5.1.d)

How do the IP routers in Area 1 know how to forward traffic to Area 2?
The IP routers in Area 1 know how to forward traffic according to the summary net link states. These link states are provided by the border routers. According to those entries they know how to route packets to certain subnets (via the ADV router).

6. Display the area routers known to Router 1 from Area 1, with the command

```
Router1# show ip ospf border-routers
```

Save the output to a file.

7. Save the Wireshark output of OSPF packet types (selecting the Print Detail option) that you did not observe in Part 4. Only include one packet of each type.

Question 5.2)

Include the Wireshark output in your report showing, if any, the different types of OSPF packets that you did not observe in Part 5.

```
No.      Time            Source                Destination           Protocol Length
     Info
      1 0.000000000     10.0.1.1              10.0.1.2              OSPF     66
           DB Description

Frame 1: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
    Interface id: 0 (eth0)
    Encapsulation type: Ethernet (1)
    Arrival Time: Mar 23, 2017 15:32:53.067332018 CET
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1490279573.067332018 seconds
    [Time delta from previous captured frame: 0.000000000 seconds]
    [Time delta from previous displayed frame: 0.000000000 seconds]
    [Time since reference or first frame: 0.000000000 seconds]
    Frame Number: 1
    Frame Length: 66 bytes (528 bits)
    Capture Length: 66 bytes (528 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ethertype:ip:ospf]
    [Coloring Rule Name: OSPF State Change]
    [Coloring Rule String: ospf.msg != 1]
Ethernet II, Src: 68:05:ca:36:33:a0, Dst: 68:05:ca:36:31:f0
    Destination: 68:05:ca:36:31:f0
        Address: 68:05:ca:36:31:f0
        .... ..0. .... .... .... .... = LG bit: Globally unique address (factory
            default)
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Source: 68:05:ca:36:33:a0
        Address: 68:05:ca:36:33:a0
        .... ..0. .... .... .... .... = LG bit: Globally unique address (factory
            default)
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: IPv4 (0x0800)
Internet Protocol Version 4, Src: 10.0.1.1, Dst: 10.0.1.2
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes
    Differentiated Services Field: 0xc0 (DSCP: CS6, ECN: Not-ECT)
        1100 00.. = Differentiated Services Codepoint: Class Selector 6 (48)
        .... ..00 = Explicit Congestion Notification: Not ECN-Capable Transport (0)
    Total Length: 52
    Identification: 0xdc64 (56420)
    Flags: 0x00
        0... .... = Reserved bit: Not set
        .0.. .... = Don't fragment: Not set
        ..0. .... = More fragments: Not set
    Fragment offset: 0
    Time to live: 1
    Protocol: OSPF IGP (89)
    Header checksum: 0xc64a [validation disabled]
        [Good: False]
        [Bad: False]
    Source: 10.0.1.1
    Destination: 10.0.1.2
    [Source GeoIP: Unknown]
    [Destination GeoIP: Unknown]
Open Shortest Path First
    OSPF Header
```

```
            Version: 2
            Message Type: DB Description (2)
            Packet Length: 32
            Source OSPF Router: 10.0.1.1
            Area ID: 0.0.0.0 (Backbone)
            Checksum: 0x9a9b [correct]
            Auth Type: Null (0)
            Auth Data (none): 0000000000000000
        OSPF DB Description
            Interface MTU: 1500
            Options: 0x02 ((E) External Routing)
                0... .... = DN: Not set
                .0.. .... = O: Not set
                ..0. .... = (DC) Demand Circuits: Not supported
                ...0 .... = (L) LLS Data block: Not Present
                .... 0... = (N) NSSA: Not supported
                .... .0.. = (MC) Multicast: Not capable
                .... ..1. = (E) External Routing: Capable
                .... ...0 = (MT) Multi-Topology Routing: No
            DB Description: 0x00
                .... 0... = (R) OOBResync: Not set
                .... .0.. = (I) Init: Not set
                .... ..0. = (M) More: Not set
                .... ...0 = (MS) Master: No
            DD Sequence: 1490286481

No.     Time           Source                Destination           Protocol Length
    Info
      2 0.000008337    10.0.1.1              10.0.1.2              OSPF     70
          LS Request

Frame 2: 70 bytes on wire (560 bits), 70 bytes captured (560 bits) on interface 0
    Interface id: 0 (eth0)
    Encapsulation type: Ethernet (1)
    Arrival Time: Mar 23, 2017 15:32:53.067340355 CET
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1490279573.067340355 seconds
    [Time delta from previous captured frame: 0.000008337 seconds]
    [Time delta from previous displayed frame: 0.000008337 seconds]
    [Time since reference or first frame: 0.000008337 seconds]
    Frame Number: 2
    Frame Length: 70 bytes (560 bits)
    Capture Length: 70 bytes (560 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ethertype:ip:ospf]
    [Coloring Rule Name: OSPF State Change]
    [Coloring Rule String: ospf.msg != 1]
Ethernet II, Src: 68:05:ca:36:33:a0, Dst: 68:05:ca:36:31:f0
    Destination: 68:05:ca:36:31:f0
        Address: 68:05:ca:36:31:f0
        .... ..0. .... .... .... .... = LG bit: Globally unique address (factory
            default)
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Source: 68:05:ca:36:33:a0
        Address: 68:05:ca:36:33:a0
        .... ..0. .... .... .... .... = LG bit: Globally unique address (factory
            default)
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: IPv4 (0x0800)
Internet Protocol Version 4, Src: 10.0.1.1, Dst: 10.0.1.2
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes
    Differentiated Services Field: 0xc0 (DSCP: CS6, ECN: Not-ECT)
        1100 00.. = Differentiated Services Codepoint: Class Selector 6 (48)
        .... ..00 = Explicit Congestion Notification: Not ECN-Capable Transport (0)
    Total Length: 56
```

```
    Identification: 0xdc65 (56421)
    Flags: 0x00
        0... .... = Reserved bit: Not set
        .0.. .... = Don't fragment: Not set
        ..0. .... = More fragments: Not set
    Fragment offset: 0
    Time to live: 1
    Protocol: OSPF IGP (89)
    Header checksum: 0xc645 [validation disabled]
        [Good: False]
        [Bad: False]
    Source: 10.0.1.1
    Destination: 10.0.1.2
    [Source GeoIP: Unknown]
    [Destination GeoIP: Unknown]
Open Shortest Path First
    OSPF Header
        Version: 2
        Message Type: LS Request (3)
        Packet Length: 36
        Source OSPF Router: 10.0.1.1
        Area ID: 0.0.0.0 (Backbone)
        Checksum: 0xdcd2 [correct]
        Auth Type: Null (0)
        Auth Data (none): 0000000000000000
    Link State Request
        LS Type: Router-LSA (1)
        Link State ID: 10.0.1.2
        Advertising Router: 10.0.1.2

No.     Time           Source                Destination           Protocol Length Info
      3 0.000484803    10.0.1.1              224.0.0.5             OSPF     182    LS Update

Frame 3: 182 bytes on wire (1456 bits), 182 bytes captured (1456 bits) on interface 0
    Interface id: 0 (eth0)
    Encapsulation type: Ethernet (1)
    Arrival Time: Mar 23, 2017 15:32:53.067816821 CET
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1490279573.067816821 seconds
    [Time delta from previous captured frame: 0.000476466 seconds]
    [Time delta from previous displayed frame: 0.000476466 seconds]
    [Time since reference or first frame: 0.000484803 seconds]
    Frame Number: 3
    Frame Length: 182 bytes (1456 bits)
    Capture Length: 182 bytes (1456 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ethertype:ip:ospf]
    [Coloring Rule Name: OSPF State Change]
    [Coloring Rule String: ospf.msg != 1]
Ethernet II, Src: 68:05:ca:36:33:a0, Dst: 01:00:5e:00:00:05
    Destination: 01:00:5e:00:00:05
        Address: 01:00:5e:00:00:05
        .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
        .... ...1 .... .... .... .... = IG bit: Group address (multicast/broadcast)
    Source: 68:05:ca:36:33:a0
        Address: 68:05:ca:36:33:a0
        .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: IPv4 (0x0800)
Internet Protocol Version 4, Src: 10.0.1.1, Dst: 224.0.0.5
    0100 .... = Version: 4
```

```
        .... 0101 = Header Length: 20 bytes
        Differentiated Services Field: 0xc0 (DSCP: CS6, ECN: Not-ECT)
            1100 00.. = Differentiated Services Codepoint: Class Selector 6 (48)
            .... ..00 = Explicit Congestion Notification: Not ECN-Capable Transport (0)
        Total Length: 168
        Identification: 0xdc66 (56422)
        Flags: 0x00
            0... .... = Reserved bit: Not set
            .0.. .... = Don't fragment: Not set
            ..0. .... = More fragments: Not set
        Fragment offset: 0
        Time to live: 1
        Protocol: OSPF IGP (89)
        Header checksum: 0xf0d0 [validation disabled]
            [Good: False]
            [Bad: False]
        Source: 10.0.1.1
        Destination: 224.0.0.5
        [Source GeoIP: Unknown]
        [Destination GeoIP: Unknown]
Open Shortest Path First
    OSPF Header
        Version: 2
        Message Type: LS Update (4)
        Packet Length: 148
        Source OSPF Router: 10.0.1.1
        Area ID: 0.0.0.0 (Backbone)
        Checksum: 0x4feb [correct]
        Auth Type: Null (0)
        Auth Data (none): 0000000000000000
    LS Update Packet
        Number of LSAs: 4
        LSA-type 1 (Router-LSA), len 36
            .000 0000 0010 1001 = LS Age (seconds): 41
            0... .... .... .... = Do Not Age Flag: 0
            Options: 0x02 ((E) External Routing)
                0... .... = DN: Not set
                .0.. .... = O: Not set
                ..0. .... = (DC) Demand Circuits: Not supported
                ...0 .... = (L) LLS Data block: Not Present
                .... 0... = (N) NSSA: Not supported
                .... .0.. = (MC) Multicast: Not capable
                .... ..1. = (E) External Routing: Capable
                .... ...0 = (MT) Multi-Topology Routing: No
            LS Type: Router-LSA (1)
            Link State ID: 10.0.1.1
            Advertising Router: 10.0.1.1
            Sequence Number: 0x80000003
            Checksum: 0xce54
            Length: 36
            Flags: 0x01 ((B) Area border router)
                .... .0.. = (V) Virtual link endpoint: No
                .... ..0. = (E) AS boundary router: No
                .... ...1 = (B) Area border router: Yes
            Number of Links: 1
            Type: Stub     ID: 10.0.1.0        Data: 255.255.255.0    Metric: 10
                Link ID: 10.0.1.0 - IP network/subnet number
                Link Data: 255.255.255.0
                Link Type: 3 - Connection to a stub network
                Number of Metrics: 0 - TOS
                0 Metric: 10
        LSA-type 3 (Summary-LSA (IP network)), len 28
            .000 0000 0101 0000 = LS Age (seconds): 80
            0... .... .... .... = Do Not Age Flag: 0
            Options: 0x02 ((E) External Routing)
                0... .... = DN: Not set
                .0.. .... = O: Not set
```

```
                ..0. .... = (DC) Demand Circuits: Not supported
                ...0 .... = (L) LLS Data block: Not Present
                .... 0... = (N) NSSA: Not supported
                .... .0.. = (MC) Multicast: Not capable
                .... ..1. = (E) External Routing: Capable
                .... ...0 = (MT) Multi-Topology Routing: No
            LS Type: Summary-LSA (IP network) (3)
            Link State ID: 10.0.2.0
            Advertising Router: 10.0.1.1
            Sequence Number: 0x80000001
            Checksum: 0xcf6b
            Length: 28
            Netmask: 255.255.255.0
            TOS: 0
            Metric: 10
        LSA-type 3 (Summary-LSA (IP network)), len 28
            .000 0000 0101 0000 = LS Age (seconds): 80
            0... .... .... .... = Do Not Age Flag: 0
            Options: 0x02 ((E) External Routing)
                0... .... = DN: Not set
                .0.. .... = O: Not set
                ..0. .... = (DC) Demand Circuits: Not supported
                ...0 .... = (L) LLS Data block: Not Present
                .... 0... = (N) NSSA: Not supported
                .... .0.. = (MC) Multicast: Not capable
                .... ..1. = (E) External Routing: Capable
                .... ...0 = (MT) Multi-Topology Routing: No
            LS Type: Summary-LSA (IP network) (3)
            Link State ID: 10.0.3.0
            Advertising Router: 10.0.1.1
            Sequence Number: 0x80000001
            Checksum: 0xce6a
            Length: 28
            Netmask: 255.255.255.0
            TOS: 0
            Metric: 11
        LSA-type 3 (Summary-LSA (IP network)), len 28
            .000 0000 0101 0000 = LS Age (seconds): 80
            0... .... .... .... = Do Not Age Flag: 0
            Options: 0x02 ((E) External Routing)
                0... .... = DN: Not set
                .0.. .... = O: Not set
                ..0. .... = (DC) Demand Circuits: Not supported
                ...0 .... = (L) LLS Data block: Not Present
                .... 0... = (N) NSSA: Not supported
                .... .0.. = (MC) Multicast: Not capable
                .... ..1. = (E) External Routing: Capable
                .... ...0 = (MT) Multi-Topology Routing: No
            LS Type: Summary-LSA (IP network) (3)
            Link State ID: 10.0.4.0
            Advertising Router: 10.0.1.1
            Sequence Number: 0x80000001
            Checksum: 0xc374
            Length: 28
            Netmask: 255.255.255.0
            TOS: 0
            Metric: 11
```

traces/5.7.packets.txt

In this exercise, we also observed DB Description and LS Request packets. LS Update packets were observed in the previous exercise, but we included this one since some also contained a different type of LSA (Summary LSA).

Question 5.3)

Include the output of the link state databases saved in Step 5.

```
       OSPF Router with ID (10.0.1.1)


                Router Link States (Area 0.0.0.0)

  LS age: 385
  Options: 0x2  : *|-|-|-|-|-|E|*
  LS Flags: 0x3
  Flags: 0x1 : ABR
  LS Type: router-LSA
  Link State ID: 10.0.1.1
  Advertising Router: 10.0.1.1
  LS Seq Number: 80000004
  Checksum: 0x51c4
  Length: 36
   Number of Links: 1

    Link connected to: a Transit Network
     (Link ID) Designated Router address: 10.0.1.1
     (Link Data) Router Interface address: 10.0.1.1
      Number of TOS metrics: 0
       TOS 0 Metric: 10


  LS age: 383
  Options: 0x2  : *|-|-|-|-|-|E|*
  LS Flags: 0x6
  Flags: 0x1 : ABR
  LS Type: router-LSA
  Link State ID: 10.0.1.2
  Advertising Router: 10.0.1.2
  LS Seq Number: 80000004
  Checksum: 0x4fc3
  Length: 36
   Number of Links: 1

    Link connected to: a Transit Network
     (Link ID) Designated Router address: 10.0.1.1
     (Link Data) Router Interface address: 10.0.1.2
      Number of TOS metrics: 0
       TOS 0 Metric: 10



                Router Link States (Area 0.0.0.1)

  LS age: 465
  Options: 0x2  : *|-|-|-|-|-|E|*
  LS Flags: 0x3
  Flags: 0x1 : ABR
  LS Type: router-LSA
  Link State ID: 10.0.1.1
  Advertising Router: 10.0.1.1
  LS Seq Number: 80000005
  Checksum: 0x7997
  Length: 36
   Number of Links: 1

    Link connected to: a Transit Network
     (Link ID) Designated Router address: 10.0.2.3
     (Link Data) Router Interface address: 10.0.2.1
      Number of TOS metrics: 0
       TOS 0 Metric: 10


  LS age: 367
  Options: 0x22 : *|-|DC|-|-|-|E|*
```

```
LS Flags: 0x6
Flags: 0x0
LS Type: router-LSA
Link State ID: 10.0.3.3
Advertising Router: 10.0.3.3
LS Seq Number: 80000004
Checksum: 0xfec2
Length: 48
 Number of Links: 2

   Link connected to: a Transit Network
    (Link ID) Designated Router address: 10.0.2.3
    (Link Data) Router Interface address: 10.0.2.3
     Number of TOS metrics: 0
      TOS 0 Metric: 1

   Link connected to: a Transit Network
    (Link ID) Designated Router address: 10.0.3.3
    (Link Data) Router Interface address: 10.0.3.3
     Number of TOS metrics: 0
      TOS 0 Metric: 1


LS age: 362
Options: 0x2  : *|-|-|-|-|-|E|*
LS Flags: 0x6
Flags: 0x0
LS Type: router-LSA
Link State ID: 10.0.3.4
Advertising Router: 10.0.3.4
LS Seq Number: 80000006
Checksum: 0x17ac
Length: 48
 Number of Links: 2

   Link connected to: a Transit Network
    (Link ID) Designated Router address: 10.0.3.3
    (Link Data) Router Interface address: 10.0.3.4
     Number of TOS metrics: 0
      TOS 0 Metric: 10

   Link connected to: a Transit Network
    (Link ID) Designated Router address: 10.0.4.5
    (Link Data) Router Interface address: 10.0.4.4
     Number of TOS metrics: 0
      TOS 0 Metric: 10


LS age: 352
Options: 0x22 : *|-|DC|-|-|-|E|*
LS Flags: 0x6
Flags: 0x0
LS Type: router-LSA
Link State ID: 10.0.4.5
Advertising Router: 10.0.4.5
LS Seq Number: 80000005
Checksum: 0x753d
Length: 48
 Number of Links: 2

   Link connected to: a Transit Network
    (Link ID) Designated Router address: 10.0.2.3
    (Link Data) Router Interface address: 10.0.2.5
     Number of TOS metrics: 0
      TOS 0 Metric: 1

   Link connected to: a Transit Network
```

```
      (Link ID) Designated Router address: 10.0.4.5
      (Link Data) Router Interface address: 10.0.4.5
       Number of TOS metrics: 0
        TOS 0 Metric: 1
```

traces/5.5.PC1.txt

```
            OSPF Router with ID (10.0.3.3) (Process ID 1)

                Router Link States (Area 1)

Link ID         ADV Router      Age         Seq#       Checksum Link count
10.0.1.1        10.0.1.1        344         0x80000005 0x007997 1
10.0.3.3        10.0.3.3        243         0x80000004 0x00FEC2 2
10.0.3.4        10.0.3.4        239         0x80000006 0x0017AC 2
10.0.4.5        10.0.4.5        229         0x80000005 0x00753D 2

                Net Link States (Area 1)

Link ID         ADV Router      Age         Seq#       Checksum
10.0.2.3        10.0.3.3        354         0x80000002 0x00AB3B
10.0.3.3        10.0.3.3        243         0x80000001 0x00A94F
10.0.4.5        10.0.4.5        230         0x80000001 0x00945B

                Summary Net Link States (Area 1)

Link ID         ADV Router      Age         Seq#       Checksum
10.0.1.0        10.0.1.1        344         0x80000001 0x00DA61
10.0.5.0        10.0.1.1        255         0x80000001 0x00131B
10.0.6.0        10.0.1.1        255         0x80000001 0x00121A
10.0.7.0        10.0.1.1        256         0x80000001 0x000724
```

traces/5.5.Router1.txt

```
       OSPF Router with ID (10.0.1.2)


                Router Link States (Area 0.0.0.0)

  LS age: 379
  Options: 0x2  : *|-|-|-|-|-|E|*
  LS Flags: 0x6
  Flags: 0x1 : ABR
  LS Type: router-LSA
  Link State ID: 10.0.1.1
  Advertising Router: 10.0.1.1
  LS Seq Number: 80000004
  Checksum: 0x51c4
  Length: 36
   Number of Links: 1

    Link connected to: a Transit Network
     (Link ID) Designated Router address: 10.0.1.1
     (Link Data) Router Interface address: 10.0.1.1
      Number of TOS metrics: 0
       TOS 0 Metric: 10


  LS age: 375
  Options: 0x2  : *|-|-|-|-|-|E|*
  LS Flags: 0x3
  Flags: 0x1 : ABR
  LS Type: router-LSA
  Link State ID: 10.0.1.2
  Advertising Router: 10.0.1.2
  LS Seq Number: 80000004
  Checksum: 0x4fc3
```

```
Length: 36
 Number of Links: 1

   Link connected to: a Transit Network
    (Link ID) Designated Router address: 10.0.1.1
    (Link Data) Router Interface address: 10.0.1.2
     Number of TOS metrics: 0
      TOS 0 Metric: 10



               Router Link States (Area 0.0.0.2)

LS age: 375
Options: 0x2  : *|-|-|-|-|-|E|*
LS Flags: 0x3
Flags: 0x1 : ABR
LS Type: router-LSA
Link State ID: 10.0.1.2
Advertising Router: 10.0.1.2
LS Seq Number: 80000005
Checksum: 0xd72d
Length: 36
 Number of Links: 1

   Link connected to: a Transit Network
    (Link ID) Designated Router address: 10.0.5.6
    (Link Data) Router Interface address: 10.0.5.2
     Number of TOS metrics: 0
      TOS 0 Metric: 10


LS age: 322
Options: 0x22 : *|-|DC|-|-|-|E|*
LS Flags: 0x6
Flags: 0x0
LS Type: router-LSA
Link State ID: 10.0.6.6
Advertising Router: 10.0.6.6
LS Seq Number: 80000004
Checksum: 0x2677
Length: 48
 Number of Links: 2

   Link connected to: a Transit Network
    (Link ID) Designated Router address: 10.0.6.6
    (Link Data) Router Interface address: 10.0.6.6
     Number of TOS metrics: 0
      TOS 0 Metric: 1

   Link connected to: a Transit Network
    (Link ID) Designated Router address: 10.0.5.6
    (Link Data) Router Interface address: 10.0.5.6
     Number of TOS metrics: 0
      TOS 0 Metric: 1


LS age: 317
Options: 0x2  : *|-|-|-|-|-|E|*
LS Flags: 0x6
Flags: 0x0
LS Type: router-LSA
Link State ID: 10.0.6.7
Advertising Router: 10.0.6.7
LS Seq Number: 80000006
Checksum: 0x5649
Length: 48
```

```
    Number of Links: 2

      Link connected to: a Transit Network
       (Link ID) Designated Router address: 10.0.6.6
       (Link Data) Router Interface address: 10.0.6.7
        Number of TOS metrics: 0
         TOS 0 Metric: 10

      Link connected to: a Transit Network
       (Link ID) Designated Router address: 10.0.7.8
       (Link Data) Router Interface address: 10.0.7.7
        Number of TOS metrics: 0
         TOS 0 Metric: 10


  LS age: 307
  Options: 0x22 : *|-|DC|-|-|-|E|*
  LS Flags: 0x6
  Flags: 0x0
  LS Type: router-LSA
  Link State ID: 10.0.7.8
  Advertising Router: 10.0.7.8
  LS Seq Number: 80000003
  Checksum: 0x7020
  Length: 48
    Number of Links: 2

      Link connected to: a Transit Network
       (Link ID) Designated Router address: 10.0.7.8
       (Link Data) Router Interface address: 10.0.7.8
        Number of TOS metrics: 0
         TOS 0 Metric: 1

      Link connected to: a Transit Network
       (Link ID) Designated Router address: 10.0.5.6
       (Link Data) Router Interface address: 10.0.5.8
        Number of TOS metrics: 0
         TOS 0 Metric: 1
```

traces/5.5.PC2.txt

```
             OSPF Router with ID (10.0.4.5) (Process ID 1)

                  Router Link States (Area 1)

Link ID         ADV Router      Age         Seq#       Checksum Link count
10.0.1.1        10.0.1.1        496         0x80000005 0x007997 1
10.0.3.3        10.0.3.3        396         0x80000004 0x00FEC2 2
10.0.3.4        10.0.3.4        392         0x80000006 0x0017AC 2
10.0.4.5        10.0.4.5        380         0x80000005 0x00753D 2

                  Net Link States (Area 1)

Link ID         ADV Router      Age         Seq#       Checksum
10.0.2.3        10.0.3.3        507         0x80000002 0x00AB3B
10.0.3.3        10.0.3.3        396         0x80000001 0x00A94F
10.0.4.5        10.0.4.5        380         0x80000001 0x00945B

                  Summary Net Link States (Area 1)

Link ID         ADV Router      Age         Seq#       Checksum
10.0.1.0        10.0.1.1        496         0x80000001 0x00DA61
10.0.5.0        10.0.1.1        407         0x80000001 0x00131B
10.0.6.0        10.0.1.1        407         0x80000001 0x00121A
10.0.7.0        10.0.1.1        408         0x80000001 0x000724
```

traces/5.5.Router2.txt

```

       OSPF Router with ID (10.0.3.4)


                Router Link States (Area 0.0.0.1)

LS age: 405
Options: 0x2  : *|-|-|-|-|-|E|*
LS Flags: 0x6
Flags: 0x1 : ABR
LS Type: router-LSA
Link State ID: 10.0.1.1
Advertising Router: 10.0.1.1
LS Seq Number: 80000005
Checksum: 0x7997
Length: 36
 Number of Links: 1

   Link connected to: a Transit Network
    (Link ID) Designated Router address: 10.0.2.3
    (Link Data) Router Interface address: 10.0.2.1
     Number of TOS metrics: 0
      TOS 0 Metric: 10


LS age: 305
Options: 0x22 : *|-|DC|-|-|-|E|*
LS Flags: 0x6
Flags: 0x0
LS Type: router-LSA
Link State ID: 10.0.3.3
Advertising Router: 10.0.3.3
LS Seq Number: 80000004
Checksum: 0xfec2
Length: 48
 Number of Links: 2

   Link connected to: a Transit Network
    (Link ID) Designated Router address: 10.0.2.3
    (Link Data) Router Interface address: 10.0.2.3
     Number of TOS metrics: 0
      TOS 0 Metric: 1

   Link connected to: a Transit Network
    (Link ID) Designated Router address: 10.0.3.3
    (Link Data) Router Interface address: 10.0.3.3
     Number of TOS metrics: 0
      TOS 0 Metric: 1


LS age: 299
Options: 0x2  : *|-|-|-|-|-|E|*
LS Flags: 0x3
Flags: 0x0
LS Type: router-LSA
Link State ID: 10.0.3.4
Advertising Router: 10.0.3.4
LS Seq Number: 80000006
Checksum: 0x17ac
Length: 48
 Number of Links: 2

   Link connected to: a Transit Network
    (Link ID) Designated Router address: 10.0.3.3
    (Link Data) Router Interface address: 10.0.3.4
     Number of TOS metrics: 0
      TOS 0 Metric: 10
```

```
     Link connected to: a Transit Network
      (Link ID) Designated Router address: 10.0.4.5
      (Link Data) Router Interface address: 10.0.4.4
       Number of TOS metrics: 0
        TOS 0 Metric: 10


  LS age: 290
  Options: 0x22 : *|-|DC|-|-|-|E|*
  LS Flags: 0x6
  Flags: 0x0
  LS Type: router-LSA
  Link State ID: 10.0.4.5
  Advertising Router: 10.0.4.5
  LS Seq Number: 80000005
  Checksum: 0x753d
  Length: 48
   Number of Links: 2

     Link connected to: a Transit Network
      (Link ID) Designated Router address: 10.0.2.3
      (Link Data) Router Interface address: 10.0.2.5
       Number of TOS metrics: 0
        TOS 0 Metric: 1

     Link connected to: a Transit Network
      (Link ID) Designated Router address: 10.0.4.5
      (Link Data) Router Interface address: 10.0.4.5
       Number of TOS metrics: 0
        TOS 0 Metric: 1
```

traces/5.5.PC3.txt

```

            OSPF Router with ID (10.0.6.6) (Process ID 1)

                 Router Link States (Area 2)

Link ID         ADV Router      Age         Seq#       Checksum Link count
10.0.1.2        10.0.1.2        242         0x80000005 0x00D72D 1
10.0.6.6        10.0.6.6        187         0x80000004 0x002677 2
10.0.6.7        10.0.6.7        182         0x80000006 0x005649 2
10.0.7.8        10.0.7.8        174         0x80000003 0x007020 2

                 Net Link States (Area 2)

Link ID         ADV Router      Age         Seq#       Checksum
10.0.5.6        10.0.6.6        241         0x80000002 0x00F5D7
10.0.6.6        10.0.6.6        187         0x80000001 0x00CD13
10.0.7.8        10.0.7.8        174         0x80000001 0x00B81F

                 Summary Net Link States (Area 2)

Link ID         ADV Router      Age         Seq#       Checksum
10.0.1.0        10.0.1.2        242         0x80000001 0x00D466
10.0.2.0        10.0.1.2        236         0x80000001 0x002E02
10.0.3.0        10.0.1.2        236         0x80000001 0x002D01
10.0.4.0        10.0.1.2        239         0x80000001 0x00220B
```

traces/5.5.Router3.txt

```

       OSPF Router with ID (10.0.6.7)


                 Router Link States (Area 0.0.0.2)
```

```

LS age: 291
Options: 0x2   : *|-|-|-|-|-|E|*
LS Flags: 0x6
Flags: 0x1  : ABR
LS Type: router-LSA
Link State ID: 10.0.1.2
Advertising Router: 10.0.1.2
LS Seq Number: 80000005
Checksum: 0xd72d
Length: 36
 Number of Links: 1

   Link connected to: a Transit Network
    (Link ID) Designated Router address: 10.0.5.6
    (Link Data) Router Interface address: 10.0.5.2
     Number of TOS metrics: 0
      TOS 0 Metric: 10


LS age: 236
Options: 0x22  : *|-|DC|-|-|-|E|*
LS Flags: 0x6
Flags: 0x0
LS Type: router-LSA
Link State ID: 10.0.6.6
Advertising Router: 10.0.6.6
LS Seq Number: 80000004
Checksum: 0x2677
Length: 48
 Number of Links: 2

   Link connected to: a Transit Network
    (Link ID) Designated Router address: 10.0.6.6
    (Link Data) Router Interface address: 10.0.6.6
     Number of TOS metrics: 0
      TOS 0 Metric: 1

   Link connected to: a Transit Network
    (Link ID) Designated Router address: 10.0.5.6
    (Link Data) Router Interface address: 10.0.5.6
     Number of TOS metrics: 0
      TOS 0 Metric: 1


LS age: 231
Options: 0x2   : *|-|-|-|-|-|E|*
LS Flags: 0x3
Flags: 0x0
LS Type: router-LSA
Link State ID: 10.0.6.7
Advertising Router: 10.0.6.7
LS Seq Number: 80000006
Checksum: 0x5649
Length: 48
 Number of Links: 2

   Link connected to: a Transit Network
    (Link ID) Designated Router address: 10.0.6.6
    (Link Data) Router Interface address: 10.0.6.7
     Number of TOS metrics: 0
      TOS 0 Metric: 10

   Link connected to: a Transit Network
    (Link ID) Designated Router address: 10.0.7.8
    (Link Data) Router Interface address: 10.0.7.7
     Number of TOS metrics: 0
```

```
        TOS 0 Metric: 10


  LS age: 222
  Options: 0x22 : *|-|DC|-|-|-|E|*
  LS Flags: 0x6
  Flags: 0x0
  LS Type: router-LSA
  Link State ID: 10.0.7.8
  Advertising Router: 10.0.7.8
  LS Seq Number: 80000003
  Checksum: 0x7020
  Length: 48
   Number of Links: 2

    Link connected to: a Transit Network
     (Link ID) Designated Router address: 10.0.7.8
     (Link Data) Router Interface address: 10.0.7.8
      Number of TOS metrics: 0
       TOS 0 Metric: 1

    Link connected to: a Transit Network
     (Link ID) Designated Router address: 10.0.5.6
     (Link Data) Router Interface address: 10.0.5.8
      Number of TOS metrics: 0
       TOS 0 Metric: 1
```

traces/5.5.PC4.txt

```
            OSPF Router with ID (10.0.7.8) (Process ID 1)

                Router Link States (Area 2)

Link ID         ADV Router      Age         Seq#       Checksum Link count
10.0.1.2        10.0.1.2        243         0x80000005 0x00D72D 1
10.0.6.6        10.0.6.6        189         0x80000004 0x002677 2
10.0.6.7        10.0.6.7        184         0x80000006 0x005649 2
10.0.7.8        10.0.7.8        173         0x80000003 0x007020 2

                Net Link States (Area 2)

Link ID         ADV Router      Age         Seq#       Checksum
10.0.5.6        10.0.6.6        243         0x80000002 0x00F5D7
10.0.6.6        10.0.6.6        189         0x80000001 0x00CD13
10.0.7.8        10.0.7.8        173         0x80000001 0x00B81F

                Summary Net Link States (Area 2)

Link ID         ADV Router      Age         Seq#       Checksum
10.0.1.0        10.0.1.2        243         0x80000001 0x00D466
10.0.2.0        10.0.1.2        237         0x80000001 0x002E02
10.0.3.0        10.0.1.2        237         0x80000001 0x002D01
10.0.4.0        10.0.1.2        240         0x80000001 0x00220B
```

traces/5.5.Router4.txt

Question 5.4)

Explain the output of the command “show ip ospf border-routers” in Step 6.

```
OSPF Process 1 internal Routing Table

Codes: i - Intra-area route, I - Inter-area route

i 10.0.1.1 [1] via 10.0.2.1, Vlan1, ABR, Area 1, SPF 11
```

traces/5.6.txt

This command shows all the border routers known to Router 1. In case of our topology, the only border router Router 1 knows of is PC1. PC1's Link State ID is 10.0.1.1, which can be reached by sending to address 10.0.2.1 (interface eth1 of PC1) via Router 1's vlan1 interface. The route is an intra-area route, meaning that the route doesn't leave the area (area 1 in this case).