Lab

5

LAN Switching

What you will learn in this lab:

- How to configure a Cisco Router and a Linux PC as a LAN switch.
- How LAN switches update their forwarding tables.
- How LAN switches run a spanning tree protocol for loop free routing..

Updated: November 2020

Table of Content

Study Material for Lab 5

1. **Bridging**: Read about LAN switching and bridging at

   https://en.wikipedia.org/wiki/Bridging_(networking)

2. **Transparent Bridges and Spanning Tree Protocol**: Read about transparent bridges and the spanning tree protocol at https://searchnetworking.techtarget.com/definition/spanning-tree-protocol

3. **Bridge Protocol Data Unit (BPDU):** Familiarize yourself with the format of bridge protocol data units (BPDUs) by reading the information at https://en.wikipedia.org/wiki/Bridge_Protocol_Data_Unit

4. **Configuring a PC as a Bridge**: Explore the website https://wiki.debian.org/BridgeNetworkConnections, which describes the *bridge-utils* software package for configuring a Linux PC as a bridge.

Prelab 5

1. Describe the difference between a LAN switch/bridge and a router?
2. What is the difference between an Ethernet switch and an Ethernet hub? Which is more suitable for a network with a high traffic load, a switch or a hub? Explain.
3. What motivates the use of the term "*transparent*" in transparent bridges?
4. Which role does the spanning tree protocol play when interconnecting LAN switches/bridges?
5. In the context of the IEEE 802.1d specification of the spanning tree protocol, define the following terms:
   - Root Bridge
   - Root Port
   - Designated Bridge
   - Designated Port
   - Blocked Port
6. In the spanning tree protocol, how does a LAN switch/bridge decide which ports are in a blocking state?

Lab 5

A *bridge or LAN switch* is a device that interconnects two or more *Local Area Networks* (*LANs)* and forwards packets between these networks. Different from IP routers, bridges and LAN switches operate at the data link layer. For example, bridges and LAN switches forward packets based on MAC addresses, whereas IP routers forward packets based on IP addresses.

LAN switches are widely deployed in enterprise networks, including university campus networks. Many enterprise networks primarily use LAN switches, and use IP routers only to connect the enterprise network to the public Internet.

The term `bridge' was coined in the early 1980s. Today, when referring to data-link layer interconnection devices, the terms `LAN switch' or `Ethernet switch' (in the context of Ethernet) are much more common. Since many of the concepts, configuration commands, and protocols for LAN switches in Lab 5 use the old term *`bridge', we* will, with few exceptions, refer to LAN switches as bridges.

This lab covers the main concepts of LAN switching in Ethernet networks: how packets are forwarded between LANs and how the routes of packets are determined. In the first and second parts of Lab 5, you learn how to configure a Linux PC and a Cisco router as a bridge. Parts 3, 4, and 5 explore how forwarding tables of bridges are set up. You learn about the concepts of *learning bridges* and *transparent bridges*, as well as the operation of the spanning tree protocol that enables loop free routing between interconnected LANs. The last part of the lab explores issues that arise when IP routers and bridges operate in the same network.

Setup for Lab 5

The configuration of the equipment in Lab 5 is changed several times during the course of the lab. With exception of the last part, the IPv4 address configuration of the Linux PCs is as shown in Table 1. Note that all IP addresses have the same netmask.

Linux PC	Interface *eth0*	Interface *eth1*
PC1	10.0.1.11/24	–
PC2	10.0.1.21/24	10.0.1.22/24
PC3	10.0.1.31/24	–
PC4	10.0.1.41/24	–

Table 1. IPv4 addresses of the PCs

Part 1. Configuring a Linux PC as a bridge

The exercises in this lab show how to configure a Linux PC as a bridge. Ethernet bridging functionality is integrated in all recent versions of Linux. The configuration of bridging functions in Linux is done with configuration commands and tools. In this lab, we provide the bridge configuration tool *brctl*.

The network configuration for Part 1 is shown in Figure 1. Here, PC1 and PC3 act as hosts and PC2 is set up as a bridge. Note that you can set up this configuration without using any hub.

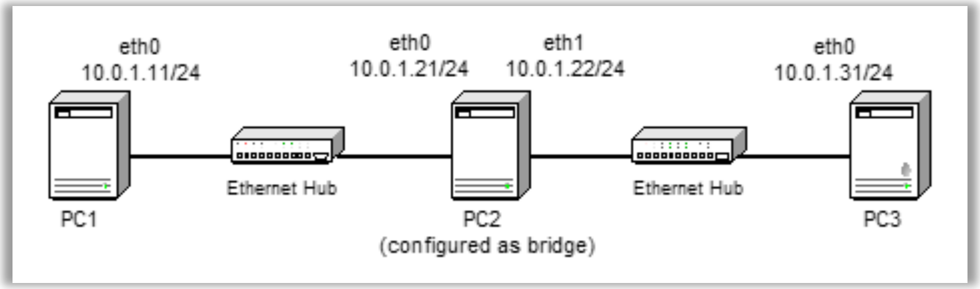

Figure 1. Network Topology for Part 1

Exercise 1.a- IP configuration of Linux PCs

Step 1: Set up the network configuration as shown in Figure 1.

Step 2: Configure the interfaces of PC1, PC2, and PC3, with the IP addresses given in Table 1. Disable the interfaces that are not used in the configuration, that is, disable interface eth*1* on both PC1 and PC3.

Step 3: Since, throughout this lab, you frequently work with MAC addresses, you should record the MAC addresses of the Linux PCs. Open consoles on the PCs and record the MAC addresses of both Ethernet interfaces with the command `ip addr show`. Enter the MAC addresses in Table 2. Save the table.

Linux PC	MAC address of interface *eth0*	MAC address of interface *eth1*
PC1		
PC2		
PC3		
PC4		

Table 2. MAC addresses of the Linux PCs

Step 4: Execute the following ping commands on PC1

```
PC1$ ping -c 1 10.0.1.21
```

```
PC1$ ping -c 1 10.0.1.22
PC1$ ping -c 1 10.0.1.31
```

Take a screenshot of the output. For each command, state why it succeeds or fails.

Lab Questions/Report

1. Include the table of the MAC addresses from Step 3.
2. Include the screenshot from Step 4. For each command, provide an explanation of the success or failure.

.

Exercise 1.b- Configure a Linux PC as a bridge

In this exercise, you configure PC2 as a bridge that forwards packets between the two Ethernet segments shown in Figure 1. The bridge configuration on the Linux PCs is done with the tool *brctl.*

Step 1: Creating a bridge with `brctl`: Each bridge is assigned a name and is associated with a set of interfaces. Here, you configure one bridge on PC2 and assign the bridge the name *Bridge1*.

```
PC2$ sudo brctl addbr Bridge1
```

Step 2: Configuring a bridge with `brctl`*:* After the bridge is created, the bridge is configured in the following steps:

a. The first part of the configuration is the assignment of network interfaces to the bridge.

```
PC2$ sudo brctl addif Bridge1 eth0
PC2$ sudo brctl addif Bridge1 eth1
```

b. The next part of the configuration sets the parameters of the spanning tree protocol (STP). In Part 1, the spanning tree protocol is not used. Therefore, you need to disable the spanning tree protocol

```
PC2$ sudo brctl stp Bridge1 off
```

Step 3: Displaying and activating the bridge with `brctl`: In the last part of the bridge configuration, you activate the bridge *Bridge1* from a terminal window. On a Linux PC, each created bridge is represented as a network interface. Therefore, if you type the command `ip address show` on PC2, the command shows an interface *Bridge1*, in addition to the other interfaces *eth0*, *eth1*, and *lo*.

a. Display the interfaces to see the presence of the bridge.

```
PC2$ ip address show
```

Take a screen snapshot of the output.

b. You can also get information about the bridge using the command

```
PC2$ brctl show Bridge1
```

Take a screen snapshot of the output.

Note that the command lists the interfaces that are part of Bridge1.

c. The bridge is activated by enabling the interface associated with the bridge. This is done with the following command:

```
PC2$ sudo ip link set dev Bridge1 up
```

Lab Questions/Report

1. Include the screenshots from Step 3.

Exercise 1.c- Observing a bridge in operation

When the bridge configuration of PC2 is complete, PC2 forwards packets between PC1 and PC3. This exercise asks you to observe the forwarding.

> **Note:**
> Adding an interface with an IP configuration to a bridge disables IP. Hence, it is no longer possible to issue ping commands to such an interface.

Step 1: Start *Wireshark* on PC1(eth0) and PC3 (eth0), and capture traffic on interface *eth0* on both systems.

Step 2: When bridging is activated on PC2, the configured IP addresses on PC2 are disabled. To verify this, issue a ping command to interfaces *eth0* and *eth1* of PC2 from PC1 and PC3 and take a screenshot of the output.

```
PC1$ ping -c 1 10.0.1.21
PC3$ ping -c 1 10.0.1.22
```

If PC2 is configured as a bridge, these ping commands should fail.

Step 3: Clear the ARP caches on PC1 and PC3, using the command `sudo ip neighbor flush all’. Display ARP caches on PC1 and PC2 using the command ‘ip neighbor show’. On PC1 the commands are:

```
PC1$ sudo ip neigh flush all
PC1$ ip neigh show
```

Step 4: Issue a ping command from PC1 to PC3 with

```
PC1$ ping -c 3 10.0.1.31
```

and take a screenshot the output.

Observe that PC2 forwards the packets between PC1 and PC3.

From the Wireshark windows, capture a packet before and after it is forwarded by PC2. Make sure that the capture shows the Ethernet and IP headers.

1. Does PC2 modify the source and destination MAC and IP addresses? What would be different if PC2 was configured as an IP router?

2. Does PC2 manipulate any fields in the MAC and IP headers?

Step 5: Run a `traceroute` command from PC1 to PC3 with

```
PC1$ traceroute 10.0.1.31
```

and take a screenshot the output.

Here, you should observe that PC2 does not appear in the output of `traceroute`.

1. Why is PC2 not visible from PC1?

2. If PC2 was configured as an IP router, how would the output differ?

Step 6: Stop the instances of *Wireshark*.

Lab Questions/Report:

1. Include the screenshot from Step 2. Explain why the ping command fails.
2. Include the screenshots from Step 4, and answer the posed questions.
3. Include the screenshot from Step 5, and answer the posed questions.

Exercise 1.d- Manipulating a PC bridge

This exercise familiarizes you with a few tasks related to running `brctl` on a Linux PC. You learn how to display the MAC table, how to delete the contents of the MAC table, and, finally, how to turn off the bridging functions. All tasks are performed on PC2.

Step 1: **Displaying the MAC table:** The MAC table of a bridge plays the same role as the routing table of an IP router. To view the contents of the MAC table of *Bridge1* on PC2, use the following command:

```
PC2$ brctl showmacs Bridge1 eth0
PC2$ brctl showmacs Bridge1 eth1
```

Step 2: **Clearing the MAC table of a bridge:** The `brctl` tool does not have a convenient way to clear the contents of the MAC table. You can delete the (non local) entries by deactivating and then re-activating the bridge. This is done with the commands

```
PC2$ sudo ip link set dev Bridge1 down
PC2$ sudo ip link set dev Bridge1 up
```

- Display the MAC table again. You should now see only the MAC addresses of the local interfaces at PC2.

Step 3: **Disabling a bridge.** Disabling a bridge on a Linux PC is done in two steps. We describe these steps for disabling *Bridge1* on PC2; first you deactivate the bridge and then you delete it, as shown below:.

1. Deactivate the interface associated with *Bridge1*. This is done by typing

   ```
   PC2$ sudo ip link set dev Bridge1 down
   ```

2. Delete the bridge by typing:

   ```
   PC2$ sudo brctl delbr Bridge1
   ```

   You can verify that the bridge is disabled as follows:

   - Verify that PC2 is operating as a normal host. To do this, issue a `ping` command to interfaces *eth0* and *eth1* of PC2 from PC1 and PC3.

     ```
     PC1$ ping -c 1 10.0.1.21
     PC3$ ping -c 1 10.0.1.22
     ```

     Note that the `ping` commands are successful.

   - Verify that PC2 does not forward packets by issuing the following *ping* command:

     ```
     PC1$ ping -c 1 10.0.1.31
     ```

     The `ping` command should not be successful.

Part 2. Configuring a Cisco Router as a bridge

Next you learn how to configure a Cisco Router as a bridge. The topology for this part is shown in Figure 2. Router1 is configured as a bridge that connects the two Ethernet segments.

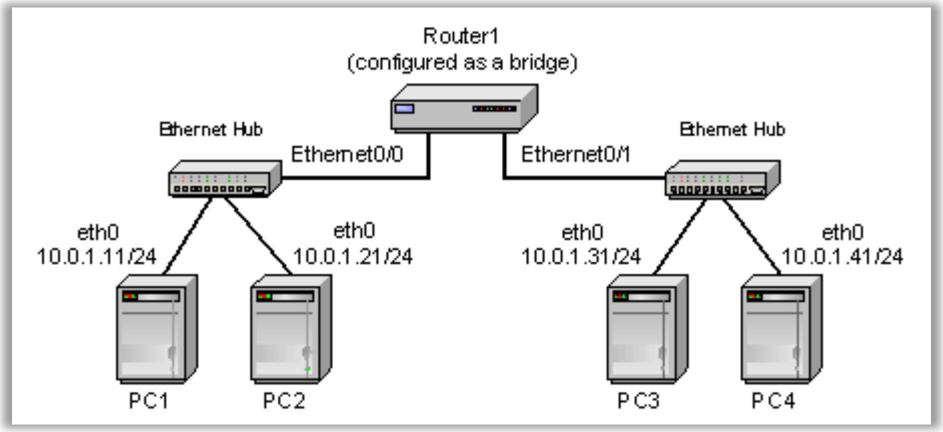

Figure 2. Network Topology for Parts 2 and 3-b

Exercise 2.a- Setup of network configuration

After the network is configured, as in Exercise 1-a above for the Linux PCs, you are asked to record the MAC addresses of the Cisco routers.

Step 1: Set up the network topology of PCs and Router1 with Ethernet switches as shown in Figure 2.

Step 2: Configure the *eth0* interfaces of the Linux PCs with the IP addresses given in Table 1. (The IP addresses of PC1, PC2, and PC3 are the same as in Part 1). Disable the *eth1* interfaces of all Linux PCs.

Exercise 2.b- Configuring a Cisco Router as a bridge

Next you configure *Router1* as a bridge. A Cisco router is configured as a bridge by disabling IP forwarding functions (with the command `no ip routing`) and by enabling bridging functions.

Similar as on the Linux PCs, a Cisco router can be configured to perform the functions of multiple independently operating bridges. This is done by defining a bridge group, which is identified by a number, and associating two or more network interfaces with each bridge group. Packets are forwarded only between interfaces that are assigned to the same bridge group. Since the exercises in Lab 5 only use one bridge group, we always use 1 to identify the group.

IOS Mode: Interface Configuration

`bridge 1 protocol ieee`

Defines a bridge group and assigns the spanning tree protocol as defined in IEEE 802.1d to bridge group *1*. After the command is issued, the Cisco router forwards packets between all interfaces that are assigned to bridge group *1*. A bridge group can be any number between 1 and 63. After defining a bridge group, one can assign network interfaces to the bridge group. It is possible to define multiple bridge groups. In Lab 5, only one bridge group (with identifier 1) is used.

`bridge 1 priority 128`

Assigns the priority *128* to bridge group *1*. The priority of a bridge group plays a role in the spanning tree protocol, which is covered in Part 5.

`debug ip rip`

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

Each interface is individually configured to participate in a bridge group. This is done with the following commands:

IOS Mode: Interface Configuration

`bridge-group N`

Assigns this network interface to bridge group *N.*

`no bridge-group N`

Removes this network interface from bridge group *N*.

`bridge-group N spanning-disabled`

Disables the spanning tree protocol on this interface for bridge group *N*.

`no bridge-group N spanning-disabled`

Enables the spanning tree protocol on this interface for bridge group *N*.

Once a Cisco router is configured as a bridge, the following commands can be used to display the status of the bridge:

IOS Mode: Privileged Exec

show bridge *1*

Displays the entries of the MAC table for interfaces that are assigned to bridge group *1*.

show spanning-tree

Displays the spanning tree topology information known to this bridge.

show interfaces

Displays statistics of all interfaces, including the MAC addresses of all interfaces.

The following commands disable bridging functions on a Cisco router:

IOS Mode: Privileged Exec

no bridge *1*

Deletes the defined bridge group. After the command is issued, the Cisco router stops forwarding packets between interfaces that are assigned to bridge group *1*.

clear bridge

Removes all entries from the MAC table.

clear arp-cache

Clears the ARP table.

Step 1: Configure *a* Cisco Router as a Bridge: Open a console on Router1. Use the above commands to configure Router1 as a bridge. On Router1, type the following commands:

```
Router1# configure terminal
Router1(config)# no ip routing
Router1(config)# bridge 1 protocol ieee
Router1(config)# bridge 1 priority 128
Router1(config)# interface Ethernet0/0
Router1(config-if)# bridge-group 1
Router1(config-if)# bridge-group 1 spanning-disabled
Router1(config-if)# no shutdown
Router1(config-if)# interface Ethernet0/1
Router1(config-if)# bridge-group 1
Router1(config-if)# bridge-group 1 spanning-disabled
Router1(config-if)# no shutdown
Router1(config-if)# end
Router1# clear bridge
Router1# clear arp-cache
```

The commands disable IP forwarding and set up Router1 as a bridge that runs with priority 128. Both Ethernet interfaces are assigned to the bridge, but the spanning tree protocol is disabled.

Step 2: Delete all entries in the ARP caches of PC1 and PC3.

Step 3: Issue a `ping` and a traceroute command from PC1 to PC3 with

```
PC1$ ping -c2 10.0.1.31
PC1$ traceroute 10.0.1.31
```

Take a screenshot of the output.

- Compare the results to the outcome of the *traceroute* command in Exercise 1.c.
- Why is it not possible to issue a *ping* command to Router1?

Lab Questions/Report:

1. Include the screen capture of the output of the *ping* and *traceroute* commands.
2. Provide the answers to the questions in Step 3.

Part 3. Learning Bridges

Each bridge has a MAC table that determines the port where a packet is transmitted from. When a packet arrives, the bridge looks up the destination MAC address of the packet in its MAC table, and retrieves the outgoing port for this packet. If the destination MAC address is not found in the forwarding table, the bridge floods the packet on all ports, with exception of the port where the packet arrived.

Bridges update their MAC table using what is called a *learning algorithm*, which works as follows. A bridge examines the source MAC address of each packet that arrives on a particular port, and memorizes that the source address is reachable via that port. This is done by adding the source MAC address and the port to the MAC table. The next time the bridge receives a packet which has this MAC address as destination, the bridge finds the outgoing port in its forwarding table. Bridges that run this algorithm are referred to as *learning bridges*. All currently deployed Ethernet switches execute the learning algorithm.

An entry in the MAC table is deleted if is not used (looked up) for a certain amount of time. The maximum time that a MAC address can stay in the forwarding table without a lookup is determined by the *Ageing* value, which is a configuration parameter.

Here you investigate the learning algorithm of bridges. The network configuration is shown in Figure 3.

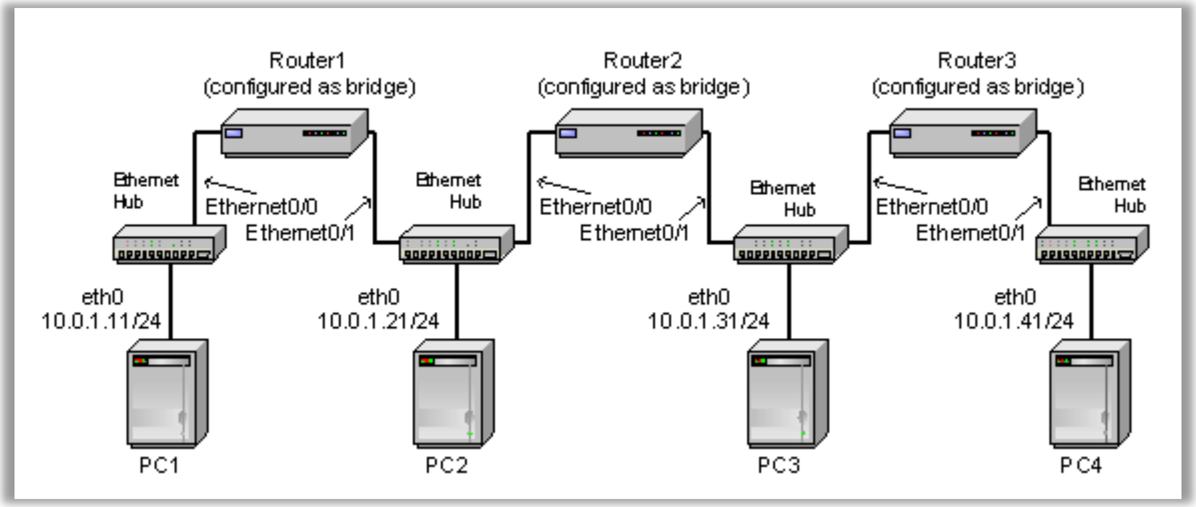

Figure 3. Network Topology for Part 4

Exercise 3.a- Exploring the learning algorithm of bridges

In this exercise you study how bridges set up their MAC tables from the network traffic.

Step 1: Set up the network configuration as shown in Figure 3. The IP addresses of the Linux PCs is the same as in Part 2.

Step 2: Open consoles on the Cisco routers (Router1, Router2, and Router3). Configure each Cisco router in the same way as shown for Router1 in Exercise 2-b.

Each Cisco router is then configured as a bridge with the spanning tree protocol disabled. In the following we refer to the Cisco routers that are configured in this fashion as “bridges.”

Step 3: Verify that PC2 is not running as a bridge. If necessary, follow the instructions in Exercise 1-d to disable the bridging functions on PC2.

~~Also, verify that on each PC only interface *eth0* is enabled.~~

Step 4: Start to capture traffic with *Wireshark* on the *eth0* interfaces of PC1, PC2, PC3, and PC4.

Step 5: Clear the ARP caches on all PCs.

Step 6: Now, issue a set of *ping* commands. After each command:

- Take a screenshot of the MAC table at each bridge with the command *show bridge*.
- Observe how the *ICMP Echo Request* and *Reply* packets (and related ARP packets) are forwarded by the bridges.
- Explain how the traffic generated by the command (which includes the ARP packets) change the MAC tables of the bridges.

Here are the *ping* commands.

```
PC1$ ping -c 2 10.0.1.21
PC2$ ping -c 2 10.0.1.11
PC2$ ping -c 2 10.0.1.41
PC3$ ping -c 2 10.0.1.21
```

Step 7: Stop the traffic capture on the PCs.

Lab Questions/Report

1. For each *ping* command in Step 6, include the screenshots of the MAC tables and explain how the traffic generated by the command changed the MAC tables of the bridges.

Exercise 3.b- Learning about new locations of hosts

Learning bridges adapt their MAC tables automatically when the location of a host changes. Due to the learning algorithm, the time it takes to adapt to a change depends on the network traffic and on the value of the *Ageing* parameter. This is illustrated in the following exercise.

Step 1: Continue with the configuration of the previous exercise. First, create or refresh entries in the MAC table at the bridges by issuing the following commands from PC1:

```
PC1$ ping -c 3 10.0.1.31
PC1$ ping -c 3 10.0.1.41
```

Now, connect PC2 to the same switch that PC4 is connected to. Record the current time with the command

```
PC1$ date
```

Step 2: Issue a *ping* command from PC1 that continuously sends *ICMP Echo Request* packets to PC2

```
PC1$ ping 10.0.1.21
```

Since Router2 does not know that PC2 has moved, it does not forward the *ICMP Echo Request* packet, and packet does not reach PC2. As a result, the ARP requests and the *ping* are unsuccessful. Eventually, since the MAC forwarding entry for PC2 is not refreshed at Router2 and Router3, the entry is deleted. When the entry is removed, the next *ICMP Echo Request* from PC1 is flooded on all ports, thus reaching PC2. When PC2 responds, all bridges update their MAC table using the source MAC address of PC2.

- Record the amount of time until the *ping* from PC1 to PC2 is again successful after PC2 has been moved to a different hub.

Step 3: Next, connect PC3 to the same hub as PC4.

Step 4: Issue a *ping* command from PC1 to PC3 that continuously sends ICMP *Echo Request* packets to PC3 with

```
PC1$ ping 10.0.1.31
```

Step 5: Then, generate a single *ICMP Echo Request* packet from PC3 to PC1 with the command

```
PC3$ ping -c 1 10.0.1.11
```

- Now, if you look on PC1, you notice that the *ping* command is successful again. Explain this outcome and compare it to the outcome of Step 2.

Lab Questions/ Report

1. Include the time that you recorded in Step 2 (with an explanation).
2. Explain the outcome of Steps 4 and 5. That is, explain why the ping issued by PC3 has the effect that the ping from PC1 to PC3 (in Step 5) quickly becomes successful. Compare the outcome with the outcome in Step 2.

Part 4. Spanning Tree Protocol

The learning algorithm from Part 3 builds the MAC tables of bridges, without the need for a routing protocol. However, since learning bridges flood a packet on all ports when a destination is not known, it may happen that packets are forwarded in a cycle and loop indefinitely. The spanning tree protocol for bridges, standardized in the IEEE 802.1d specification, prevents such forwarding loops from occurring. This is done by organizing the bridges in a spanning tree topology. Learning bridges that run the spanning tree protocol are called *transparent bridges*.

The spanning tree protocol, which is used by virtually all Ethernet switches, works as follows. One bridge, called the *root bridge*, is elected to be the root of the tree. Each bridge determines which of its ports, has the best path to the root bridge. This is the *root port* of the bridge. On each LAN, the bridges elect one bridge, called the *designated bridge*, which, among all bridges on the same LAN, has the best path to the root bridge. The port that connects a bridge to the LAN where it is a designated bridge is called *designated port*. Then, all bridges disable all ports that are not root ports or designated ports. What results is a spanning tree of bridges. Since a tree topology does not have a loop, forwarding packets along the edges of the tree guarantees that forwarding loops are entirely avoided.

This part of the lab has three components: (1) You set up a new network configuration; (2) You verify that bridges without the spanning tree result in forwarding loops; And (3) you configure the spanning tree protocol and observe how it prevents loops from occurring.

Exercise 4.a- Configuring a topology that results in forwarding loops

Step 1: Set up the network topology as shown in Figure 4. Configure Router1, Router2, Router3, and Router4 as done in Exercise 2-b. Also, configure PC2 as a bridge as in Exercise 1-b. Configure PC1, PC3, and PC4 as hosts with IP addresses on the eth0 interfaces as shown in Table 1.

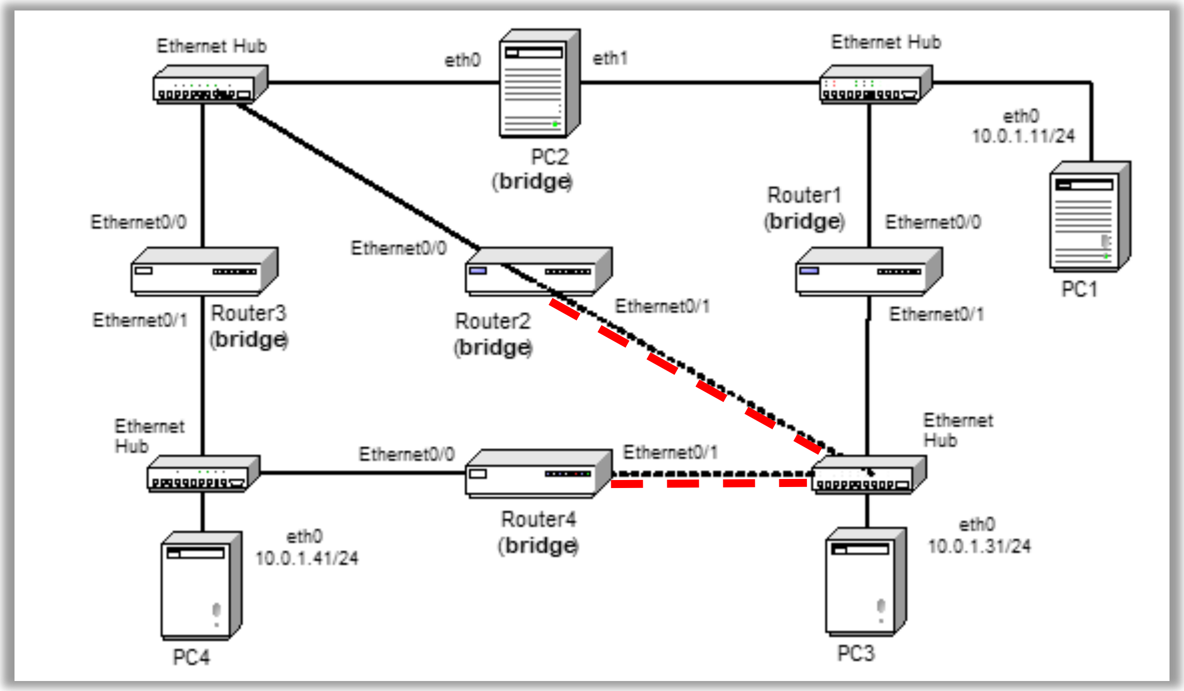

Figure 4. Network Topology for Part 4.

Note:
For the time being, do not connect interfaces Ethernet0/1 of Router2 and Router4 (shown as red dashed line Figure 4). Adding these connections will create in a forwarding loop. The connections will be made when you have set up the tools that allow you to observe the forwarding loop.

Step 2: Verify that the MAC addresses of the Linux PCs and the Cisco routers are recorded in Table 2 and **Error! Reference source not found.**. You need to refer to these tables frequently in the following exercises and in the lab report.

Step 3: Since ARP traffic interferes with the forwarding and bridge learning operations that you need to observe, pre-configure the ARP tables with static entries so that no ARP *Request* packets need to be sent.

First clear the non-permanent entries in the ARP tables of PC1, PC3 and PC4, and replace them with permanent (static) entries for interface *eth0* of PC1, PC3, and PC4, with the following commands.

```
PC1$ sudo ip neigh del 10.0.1.31 dev eth0
PC1$ sudo ip neigh del 10.0.1.41 dev eth0
PC1$ sudo ip neigh add 10.0.1.31 lladdr MACaddress dev eth0
PC1$ sudo ip neigh add 10.0.1.41 lladdr MACaddress dev eth0
```

```
PC3$ sudo ip neigh del 10.0.1.11 dev eth0
PC3$ sudo ip neigh del 10.0.1.41 dev eth0
PC3$ sudo ip neigh add 10.0.1.11 lladdr MACaddress dev eth0
PC3$ sudo ip neigh add 10.0.1.41 lladdr MACaddress dev eth0

PC4$ sudo ip neigh del 10.0.1.11 dev eth0
PC4$ sudo ip neigh del 10.0.1.31 dev eth0
PC4$ sudo ip neigh add 10.0.1.11 lladdr MACaddress dev eth0
PC4$ sudo ip neigh add 10.0.1.31 lladdr MACaddress dev eth0
```

Note that lladr in the above command is "(lowercase L)(lowercase L)addr". In the above commands, *MACaddress* is the 48-bit MAC address of the corresponding interface from Table 2. The MAC addresses are entered in hexadecimal notation, where each byte is separated by a colon, e.g., 00:a0:24:71:e4:44.

Exercise 4.b- Observing forwarding loops

The problem with learning bridges that do not run the spanning tree algorithm is that bridges flood a packet on all outgoing links, when a destination MAC address is not found in the MAC table. If this results in a forwarding loop, the packet will be flooded over and over again, with each reception of the packet at a bridge generating a new round of copies. Thus, not only are packets forwarded in an infinite loop, in addition, the number of packets that are forwarded increases due to the repeated flooding of the same packet in each round of the loop.

In this exercise you observe that the bridges in the topology of Figure 4 forward packets in a loop.

Step 1: Once the network topology of Figure 4 is set up, verify that the spanning tree protocol (STP) is disabled on all bridges as follows:

a. On the Cisco routers: Type 'show running-config' and verify that the line `bridge-group 1 spanning-disabled' is displayed for both Ethernet interfaces.

b. On PC2: Issue `brctl show' and verify that STP enabled is set to 'no'

Step 2: Clear the contents of the forwarding tables on all bridges with the following commands:

a. On the Cisco routers: Execute the command clear bridge.

b. On PC2: Follow the instructions for `Clearing the MAC table of a bridge' in Exercise 1.d.

Step 3: Start *Wireshark* on the eth0 interfaces of PC1, PC3 and PC4.

Step 4: Now complete the topology in Figure 4 by connecting interface *Ethernet0/1* of Router2 and Router4 to the switches.

Step 5: Issue a *ping* command from PC1 to PC4.

```
PC1$ ping -c 1 10.0.1.41
```

Use the *Wireshark* output to observe the route of the *ICMP Echo Request* and *Reply* packets. You should be able to observe that the ICMP packet is looping. (You may see other packets than *ICMP Echo Request* and *Reply.* The main observation is that packets are looping).

Step 6: Wait for several seconds and break the loop by disconnecting the *Ethernet0/1* interfaces of Router2 and Router4. Otherwise, the packets will loop forever and cause the bridges to become overloaded and stop forwarding packets.

Step 7: Stop the traffic capture with *Wireshark* on all PCs. Save the captured traffic for one of the PCs. (In the lab report, you will be asked to identify a packet that is in a forwarding loop.)

Step 8: Take a screenshot of the MAC table of all bridges, that is, Router1, Router2, Router3, Router4, and PC2.

Lab Questions/Report:

1. Use the Wireshark data saved from Step 7 to document for a single packet that the packet is forwarded in a loop.
2. Use the output from *Wireshark* and the MAC tables to explain why some packets are looping indefinitely?

Exercise 4.c- Enabling the spanning tree protocol

Next, you enable the spanning tree protocol on the bridges in Figure 4, and in this way, complete forwarding loops. Before starting the exercise, we provide a brief description of the spanning tree protocol.

An Overview of the Spanning Tree Protocol

The IEEE 802.1d spanning tree protocol (STP) organizes bridges in a tree topology without any central coordination. Every bridge has only a limited view of the spanning tree, and no bridge has complete knowledge of the complete spanning tree.

Bridge ID: In the spanning tree protocol, each bridge has a unique identifier, called the bridge ID. The bridge ID has a length of 8 bytes. The first two bytes are the bridge priority and the remaining six bytes are the bridge MAC address. The bridge priority is a configuration parameter. The bridge MAC address is set to the lowest MAC address of any of the ports of the bridge. In the spanning tree protocol, the bridge with the lowest bridge ID is elected as the root bridge.

BPDUs: Bridges build the spanning tree by exchanging *Bridge Protocol Data Units (BPDU)*. Bridges send BDPUs approximately once every 4 seconds. BPDUs are only exchanged between bridges that are connected to the same LAN.

Table 4 presents the four fields of a BPDU that are relevant to the spanning tree protocol. The BPDU of a bridge advertises the best path from this bridge to the root bridge. Specifically, a BPDU (*R, C, B, P*) where R is the value of the root ID, C is the value of the root path cost, B is the bridge ID, and P is the port ID, is interpreted as follows: "*I am bridge B and I am sending from my port P. I believe R to be the root bridge, and the cost of my path to the root bridge is C.*"

Table 3. Fields of a BPDU relevant to the spanning tree construction

Length (in Bytes)	Field Name	Content
8	Root ID (*R*)	Identifies the root bridge (as seen by the sender of this BPDU).
4	Root path cost (*C*)	Cost of the path from the sender of this BPDU to the root bridge.
8	Bridge ID (*B*)	Identifies the sender of this BPDU.
2	Port ID (*P*)	Identifies the network interface (port) where this BPDU is sent.

Operations of the spanning tree protocol

Each bridge listens on all its ports to BPDUs sent by other bridges. If a bridge receives a BPDU that advertises a 'better' path than advertised in its own BPDU, the bridge updates its BPDU. To determine if a received BPDU advertises a better path, the bridge compares the received BPDU to its own BPDU. If the root ID in the received BPDU is smaller than the root ID of the bridge, the received BPDU is seen as advertising a better path. If the root IDs are identical, the BPDU with the lower root path cost advertises a better path. If both the root ID and root path cost are identical, then the BPDU with the lower bridge ID is seen as advertising a better path cost. Finally, if root ID, root path cost, and bridge ID are all identical, then the BPDU with the lowest port ID is interpreted as advertising a better BPDU.

When a bridge with BPDU *(R1, C1, B1, P1)* receives a BPDU *(R2, C2, B2, P2)* and the received BPDU advertises a better path, the bridge updates its own BPDU to *(R2, C2+increment, B1, P1).* The increment value is a configuration parameter that accounts for the cost increase of the path due to bridge *B1*. When the increment value is set to 1 on all bridges, then the bridges establish a minimum hop route to the root bridge. The *increment* can also be set to account for the data rate of a LAN. For example, to make a path on a 100 Mbps LAN more desirable than on a 10 Mbps LAN, the 10 Mbps LAN can be assigned a larger increment value.

A bridge transmits its BPDU on a port only if its BPDU advertises a better route than any of the BPDUs received on that port. In this case, the bridge assumes that it is the *designated bridge* for the LAN to which the port connects, and the port that connects the bridge to the LAN is called the *designated port* of the LAN. A bridge that is not the designated bridge for a LAN does not send BPDUs on that LAN.

On each bridge, the port where the bridge receives the BPDU that advertises the best route is labeled the *root port*.

Constructing the spanning tree: Each bridge locally decides which of its ports are part of the spanning tree. Only the root port and the designated ports of a bridge are part of the spanning tree, all other ports are not part of the spanning tree. (One can reconstruct the complete tree by connecting, for each LAN, the root ports that connect to this LAN with the designated port of the LAN).

Each bridge forwards packets only on ports that are part of the spanning tree, that is, if it is received on the root port or on its designated ports. These ports are said to be in a *Forwarding* state. All other ports are said to be in a *Blocking* state. In this way, packets are forwarded only along the edges of the spanning tree. As a result, since a tree topology does not have a loop, the forwarding of packets does not result in loops.

Initializing the spanning tree protocol: When a bridge, say, with bridge ID B, is started, it assumes that it is the root bridge. It sends a BPDU (*B, 0, B, p*) on all its ports *p*. The root path cost is set to zero, since B believes itself to be root. Within a short amount of time, the bridge learns about better paths, and the protocol quickly convergences to a new spanning tree.

Your task in this exercise is to set up the spanning tree protocol. You capture the BPDUs in the network of Figure 4 (including the connections between eth1 of routers Router2 and Router4 to the hub as shown in the figure) and trace how the spanning tree is constructed.

Step 1: Delete all entries in the MAC tables on all bridges (Router1, Router2, Router3, Router4, and PC2), as done in Step 2 of Exercise 1.d.

Step 2: If you disconnected interface *Ethernet0/1* of Router2 and Router4 from the switches, reconnect them.

Step 3: Next, enable the spanning tree protocol on all bridges (Router1, Router2, Router3, Router4, PC2).

a. On the Cisco routers, the following commands, shown here for Router1, must be executed:

```
Router1# configure terminal
Router1(config)# interface Ethernet0/0
Router1(config-if)# bridge-group 1
Router1(config-if)# no bridge-group 1 spanning-disabled
Router1(config-if)# no shutdown
Router1(config-if)# interface Ethernet0/1
Router1(config-if)# bridge-group 1
Router1(config-if)# no bridge-group 1 spanning-disabled
Router1(config-if)# no shutdown
Router1(config-if)# exit
```

b. On PC2, the spanning tree protocol is enabled with *brctl*, following the instructions given in Exercise 1.b and Exercise 1.d. spanning tree is enabled by the following command:

```
PC2$ sudo brctl stp Bridge1 on
```

Step 4: Set the bridge priority to *128* on all bridges. Table 5 shows valid priority values which could be used by *brctl*.

a. On the Cisco routers, you can view the bridge priority by typing

```
Router1# show spanning-tree
```

The output should include a line that states: “Bridge Identifier has priority 128 ...”. If the priority is different from 128, use the following IOS global configuration command to change the priority:

```
Router1(config)# bridge 1 priority 128
```

b. On PC2, set the bridge priority to 128 using the following command:

```
PC2$ sudo brctl setbridgeprio Bridge1 128
```

Table 4. Valid bridge priority values for *brctl*.

Input Value in Bridge Priority Field of *brctl*	Bridge Priority Value Displayed by *brctl* (hexadecimal)	Bridge Priority Value (decimal)
16	0008	8
32	0010	16
64	0020	32
256	0040	64
512	0080	128
1024	0100	256

Step 5: Wait until the spanning tree stabilizes. The spanning tree has stabilized when all interfaces of the bridges are either in state *Blocking* or *Forwarding*. You can obtain the state of the interfaces of a bridge by typing:

- Display the spanning tree information on the Cisco routers. For Router1, the command is

```
Router1# show spanning–tree
```

- Display the spanning tree information on PC2 by typing

```
PC2$ brctl showstp Bridge1
```

After the spanning tree has stabilized, take screenshots of the output of the above commands.

Study the output of the commands to derive the spanning tree configuration.

1. Provide a drawing of the network topology that shows, for each bridge, the root port, the designated ports, and the blocked ports.
2. Use the drawing to show the spanning tree.
3. Determine the path cost (designated cost) of each bridge?

Step 6: Delete the ARP cache on all Linux PCs, including the static entries that you added in Exercise 4.a.

Step 7: Run *Wireshark* to capture traffic on the *eth0* interfaces of all PCs.

Step 8: In each Wireshark window, select the BPDUs of the spanning tree protocol by setting the display filter “stp”.

Explore the BPDUs captured in all Wireshark windows. **For each window**:

1. After convergence of the spanning tree algorithm, the BPDU messages captured by each instance of Wireshark are identical. Explain why this is the expected outcome of the spanning tree algorithm.
2. Create a screen snapshot of the details of a BPDU. The message should show:
    - All fields of the Ethernet header
    - Root bridge identifier
    - Root path cost
    - Identifier of the sending bridge
    - Identifier of the port where the BDPU was transmitted
3. Reconcile the content of the BPDU messages with the spanning tree information displayed by the bridges.
4. Inspect some details of the BPDU messages, in particular,
    - the MAC destination addresses
    - the Ethernet header.

Step 9: Issue a *ping* command from PC1 to PC4 by typing

```
PC1$ ping -c 1 10.0.1.41
```

Confirm that the ping is successful.

1. Use the Wireshark traffic captures to determine the path of packets that are generated when issuing the ping command. Record your findings.
2. Confirm your answer with the spanning tree information from Step 5.

Step 10: Terminate all *Wireshark* applications.

Lab Questions/Report

1. Provide the screenshots from Step 5 displaying the spanning tree information of the bridges.
2. Use a drawing of the network topology and the spanning tree information to indicate, for each bridge, the bridge identifier, the root port, the designated ports, and the blocked ports. Make sure that ports are labelled. Determine the path cost (designated cost) of each bridge.
3. Use the above drawing (or a new drawing) to indicate the path of packets that are sent between PC1 and PC4 in Step 8. Explain why this path is unique.
4. Provide the screen captures of the BPDUs from Step 8.
5. Reconcile the content of the BPDU messages with the spanning tree information displayed by the bridges.
6. Explain the MAC destination address of the BPDU messages.
7. Present your observations of the Ethernet header of the BPDU messages.
8. Use a drawing as in Figure 4 to trace the path of the packets that are sent as a result of the *ping* command (e.g. ARP Request and Reply, ICMP Echo Request and Reply). Explain how the path of packets can be explained by the outcome of the spanning tree algorithm.

> **Note:**
> The next part of the lab continues with the network configuration from Figure 4, and partially relies on the results in Exercise 4.c- You may want to directly continue to Part 5.

Part 5. Reconfigurations of the Spanning Tree

The spanning tree protocol adapts to changes in the network topology. Here you observe how the spanning tree topology adapts when a link fails.

Exercise 5.a- Reconfiguration of the spanning tree

You simulate a failed link in the network shown in Figure 4 and observe how the bridges adjust to a change in the network topology.

Step 1: Continue with the network configuration at the end of Exercise 4-c in Figure 4.
Make sure that the spanning tree protocol is enabled on Router1, Router2, Router3, Router4, and PC2, and that the priorities of all bridges are set to 128.

Step 2: Start to capture traffic with *Wireshark* on the *eth0* interfaces of PC1, PC3, and PC4.

Step 3: Issue a **ping** command from PC1 to PC4 that sends an *ICMP Echo Request* every second:

```
PC1$ ping -i 1  10.0.1.41
```

Step 4: If you did not *just* complete Part 4, Determine the root bridge of the spanning tree. You can infer this from the spanning tree information of each bridge. The You can do this with the command `**show spanning-tree’** on  a Cisco router and with `**brctl showstp Bridge1**’ on a Linux PC.

Step 5: Disconnect one of the interfaces of the root bridge from the switch.

- From the display of *Wireshark* on the PCs, observe the BPDUs of the bridges that build a new spanning tree.
- Observe that the *ping* command at PC1 fails for a certain period of time, but is successful again when the new spanning tree is built.
- When the *ping* command works again, terminate the *ping* command on PC1 with *Ctrl-C.* Use the statistics that are displayed when the *ping* command is terminated, to estimate the amount of time it took to build the new spanning tree.

Step 6: Display the spanning tree information on the bridges (see Step 5 in Exercise 4-c) and take screenshots of the information.

- Provide a drawing of the network topology that shows, for each bridge, the root port, the designated ports, and the blocked ports.
- Use the drawing to show the new spanning tree.

Step 7: Terminate all Wireshark sessions.

Lab Questions/Report

1. Include the screenshots from Step 6.
2. Provide a drawing of the network topology that shows, for each bridge, the root port, the designated ports, and the blocked ports, and draw the new spanning tree.

Exercise 5.b- Forcing a bridge to become the root bridge

Here you force a certain bridge to become the root bridge. Recall that the priority of a bridge is used in the first two bytes of the bridge ID. Thus, the bridge with the lowest priority value has the lowest bridge ID. Since the spanning tree protocol elects the bridge with the lowest bridge ID as the root bridge, the root bridge can be fixed by modifying the priority field.

Being able to fix a certain bridge as the root bridge provides some control over the spanning tree topology. For example, one can select the device with the highest capacity to become the root bridge.

Step 1: Continue with the topology at the end of Exercise 5.a- (Do not reconnect the interface of the root bridge was disconnected in Step 5 of Exercise 6.a).

Step 2: Select one of the bridges from Figure 4, other than the bridge that was the root bridge in Exercise 4.c. Change the priority of this bridge to 64. Since all other bridges in the network have a bridge priority of 128, the selected bridge becomes the root bridge.

Step 3: If you selected Router1, then the commands to set the bridge priority are as follows:

```
Router1# configure terminal
Router1(config)# bridge 1 priority 64
Router1(config)# end
```

(On a Linux PC, you can change the bridge priority to 64 through *brctl*.)

Step 4: Display the spanning tree information on the bridges (see Step 5 in Exercise 4-c) and take screenshots of the information.

- Provide a drawing of the network topology that shows, for each bridge, the root port, the designated ports, and the blocked ports.
- Use the drawing to show the new spanning tree.

Lab Questions/Report:

1. Include the screenshots from Step 4.
2. Provide a drawing of the network topology that shows, for each bridge, the root port, the designated ports, and the blocked ports, and draw the new spanning tree.

Part 6. Mixed Router and Bridge Configuration

In this part of the lab you set up a network topology that contains bridges as well as IP routers. Both bridges and routers are devices that connect networks and forward packets between networks. Bridges make forwarding decisions based on destination MAC addresses. IP routers make forwarding decisions based on destination IP. In a properly configured network, bridges and IP routers coexist without causing network problems. Sometimes, however, the forwarding of packets in a network with bridges and IP routers can be difficult to trace. The following exercises explore such a scenario.

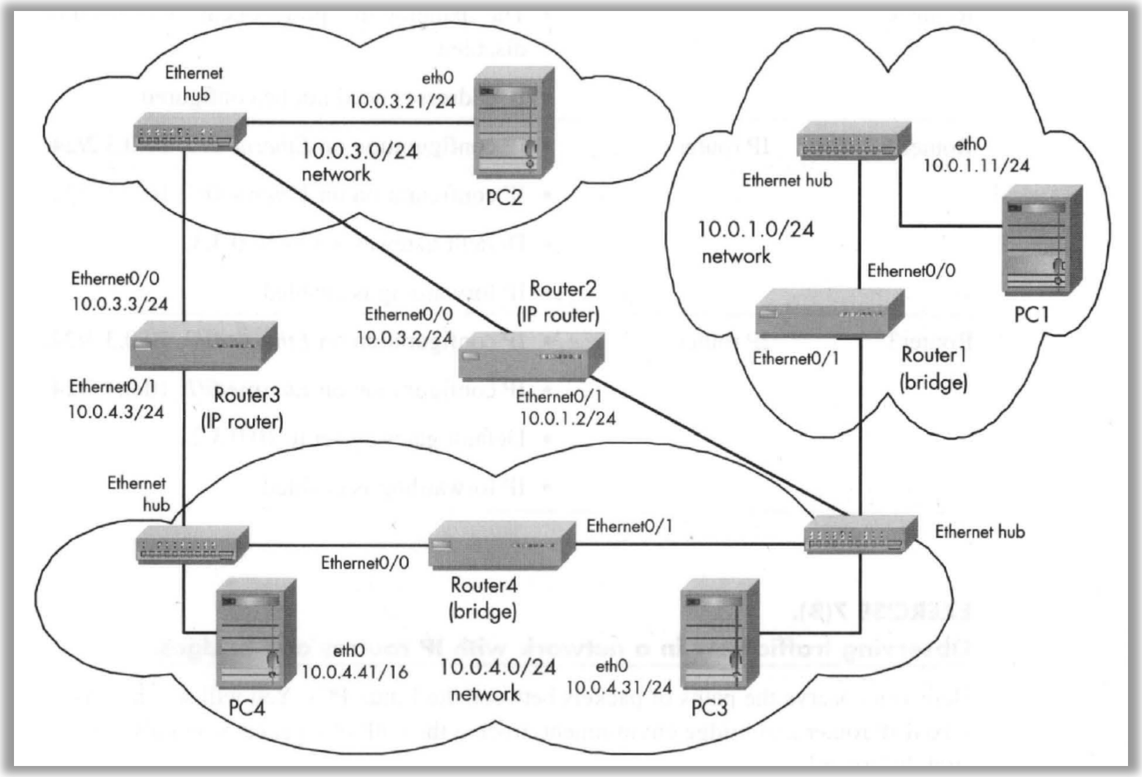

Figure 5. Network topology for Part 6.

Exercise 6.a- Setting up the network configuration

Step 1: Create a network topology of Linux PCs and Cisco routers as shown in Figure 5.

Step 2: Skip this step, unless you continue from Part 5.

Disable bridging on PC2, Router2, and Router3.

a. On Router2 and Router3, you disable the bridging functions by typing

```
Router1# configure terminal
Router1(config)# no bridge 1
Router1(config)# end
```

b. On PC2, the commands are

```
PC2$ sudo ip link set dev Bridge1 down
```

```
PC2$ sudo brctl delbr Bridge1
```

Step 3: Skip this step, if you continue from Part 5 (where Router1 and Router 4 are configured as bridges).
Configure Router 1 and Router 4 as bridges using the commands from Exercise 2-b (Step 1).

Step 4: Complete the IP configuration of the Linux PCs, Router2, and Router3 as given in Tables 6 and 7.

a. If you continue from Part 5, you delete the IP address of interface eth0 on PC3 and PC4 before adding the address given in Table 6.

b. Note that PC5 has a /16 prefix. All other configured addresses use a /24 prefix.

c. Configure Router2 and Router3 as IP routers (ip routing). Also, disable Proxy ARP on all interfaces of the IP routers.

d. Here are the configuration commands for Router2:

```
Router2# configure terminal
Router2(config)# no ip routing
Router2(config)# ip routing
Router2(config)# ip route 0.0.0.0  0.0.0.0  10.0.3.3
Router2(config)# interface ethernet0/0
Router2(config-if)# ip address 10.0.3.2 255.255.255.0
Router2(config-if)# no ip proxy-arp
Router2(config-if)# no shutdown
Router2(config-if)# exit
Router2(config)# interface ethernet0/1
Router2(config-if)# no ip proxy-arp
Router2(config-if)# ip address 10.0.1.2 255.255.255.0
Router2(config-if)# no shutdown
Router2(config-if)# end
```

Table 5. IP configuration of the Linux PCs for Part 6.

Linux PC	Ethernet Interface *eth0*	Ethernet Interface *eth1*	Default Gateway
PC1	10.0.1.11/24	Disabled	10.0.1.2
PC2	10.0.3.21/24	Disabled	10.0.3.2
PC3	10.0.4.31/24	Disabled	10.0.4.3
PC4	10.0.4.41/16	Disabled	10.0.4.3

Table 6. IP configuration of of the Cisco routers for Part 6.

Cisco Router	Configured As	Configuration information
Router1 Router4	Bridge	Enable bridging on both interfaces *Ethernet0/0* and *Ethernet0/1* The spanning tree protocol can be enabled or disabled IP addresses need not be configured
Router2	IP router	IP configuration on *Ethernet0/0*: 10.0.3.2/24 IP configuration on *Ethernet0/1*: 10.0.1.2/24 Default gateway set to 10.0.3.3 IP forwarding is enabled
Router3	IP router	IP configuration on *Ethernet0/0*: 10.0.3.3/24 IP configuration on *Ethernet0/1:* 10.0.4.3/24 Default gateway set to 10.0.3.2 IP forwarding is enabled

Exercise 6.b- Observing traffic flow in a network with IP routers and bridges

Here you observe the route of traffic between the Linux PCs. You will see that in a mixed IP router and bridge environment, tracing the routes of traffic is not always straightforward.

Step 1: Clear the forwarding tables on all bridges and clear the ARP cache on all hosts and IP routers.

```
Router1# clear bridge
```

```
Router2# clear ip route *
```

Step 2: Run *Wireshark* to capture traffic on the *eth0* interface of each Linux PC.

Step 3: Issue the following set of *ping* commands. Use the output of Wireshark to determine the path of the *ICMP Echo Request* and *Rep*ly packets. Note that not all *ping* commands will be successful.

a. On PC1:

```
PC1$ ping -c 3 10.0.4.31
```

b. On PC1:

```
PC1$ ping -c 3 10.0.4.41
```

c. On PC4:

```
PC4$ ping -c 3 10.0.1.11
```

d. On PC1:

```
PC1$ ping -c 3 10.0.3.21
```

- Determine which of the *ping* commands are successful and which fail.
- Use the data displayed by *Wireshark* to determine the route of *the ICMP Echo Request* and *Reply* packets (e.g., PC1 → Router1 → Router2 → Router4 → PC4).
- The Wireshark sessions do not allow you to observe the traffic between the Cisco routers. To trace the path of the packets, you can use other sources of information:
    - The MAC addresses in Ethernet packet headers can be matched against the MAC addresses of the Cisco routers.
    - The Identification field in the IP header can be used to uniquely tag a packet. If a packet with the same tag is observed at two observation points, and the path between the observation points is unique, the packet has traversed the unique path.
- For each path, provide an explanation why a certain route is taken by the *ICMP Echo Request* and *Reply* packets.

Step 4: On each PCs, save enough ICMP packets so that you can address the problems raised in Step 3. You can save the ICMP packets as screen snapshots. Alternatively, you can save the Wirehshark data as a plaintext file.

Step 5: Stop the *Wireshark* packet captures on all PCs.

Lab Questions/Report

1. Describe which of the *ping* commands are successful and which fail. Use the data that you captured to determine the route of the *ICMP Echo Request* and *Reply* packets. For each route, provide an explanation why the path is taken for each of the *ping* commands above.