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 MAC tables.
* How LAN switches run a spanning tree protocol for loop free routing.

Updated: February 2021

**Table of Content**

[Study Material for Lab 5 3](#_Toc65183855)

[Prelab 5 4](#_Toc65183856)

[Lab 5 5](#_Toc65183857)

[Part 1. Configuring a Cisco Router as a bridge 7](#_Toc65183858)

[Exercise 1-a. Setup of network configuration 7](#_Toc65183859)

[Exercise 1-b. Configuring a Cisco Router as a bridge 8](#_Toc65183860)

[Part 2. Learning Bridges 11](#_Toc65183861)

[Exercise 2-a. Exploring the learning algorithm of bridges 11](#_Toc65183862)

[Exercise 2-b. Learning about new locations of hosts 12](#_Toc65183863)

[Part 3. Spanning Tree Protocol 15](#_Toc65183864)

[Exercise 3-a. Configuring a topology that results in forwarding loops 15](#_Toc65183865)

[Exercise 3-b. Observing forwarding loops 17](#_Toc65183866)

[Exercise 3-c. Enabling the spanning tree protocol 18](#_Toc65183867)

[Part 4. Reconfiguration of the Spanning Tree 23](#_Toc65183868)

[Exercise 4-a. Reconfiguration of the spanning tree 23](#_Toc65183869)

[Exercise 4-b. Forcing a bridge to become the root bridge 24](#_Toc65183870)

[Part 5. Mixed Router and Bridge Configuration 25](#_Toc65183871)

[Exercise 5-a. Setting up the network configuration 25](#_Toc65183872)

[Exercise 5-b. Observing traffic flow in a network with IP routers and bridges 27](#_Toc65183873)

# Study Material for Lab 5

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

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

1. **Transparent Bridges and Spanning Tree Protocol**: Read about transparent bridges and the spanning tree protocol at   
   <https://searchnetworking.techtarget.com/definition/spanning-tree-protocol>
2. **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>
3. **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?
   1. The term “transparent” originates from the fact that, with this method, end systems are oblivious to the existence and configuration of bridges inserted in the network ([Transparent Bridging - an overview | ScienceDirect Topics](https://www.sciencedirect.com/topics/computer-science/transparent-bridging))
   2. They are transparent to networks connected hosts
   3. They don’t require setup
   4. [What is a Transparent Bridge? - Definition from Techopedia](https://www.techopedia.com/definition/3179/transparent-bridge" \l ":~:text=A%20transparent%20bridge%20is%20a,all%20the%20network%27s%20connected%20hosts.&text=Transparent%20bridges%20are%20implemented%20primarily%20in%20Ethernet%20networks.)
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:
6. Root Bridge
7. Root Port
8. Designated Bridge
9. Designated Port
10. Blocked Port
11. 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.

The term “bridge” refers to a Layer-2 switch. Since we are using Ethernet as the Layer 2 technology, the bridge configurations of the Cisco routers turned these devices into Ethernet switches.

The exercises in this part explore how the switches in the network configurations relate to the concept of a “bridge”. Depending on the type of device that plays the role of an Ethernet switch in the network configurations, different outcomes are possible.

* A **hub** is a relatively simple device that floods a packet received on one of its ports to all other ports. A hub does not maintain a MAC table and does not learn addresses by observing traffic. Even though a hub connects devices on an Ethernet network, it is strictly not correct to classify it as an Ethernet switch. Today, hubs are pretty much extinct.
* An **unmanaged switch** is a low-cost Ethernet switch that does not require any configuration. In fact, the attributed “unmanaged” indicates that the switch does not offer any configuration options. Ethernet switches in home networks or small offices are usually unmanaged switches. The functionality offered by unmanaged switches varies a lot. All unmanaged switches support the learning algorithm seen in Part 2 of this lab. Unmanaged switches may, but generally do not, support the spanning tree protocol for loop prevention, which will be covered in detail in Part 5.
* A **managed switch** is fully configurable. The configuration commands on a managed switch are similar to those on a Cisco router, with the difference that the commands relate to Layer 2 configurations. If fact, the configuration of a Cisco router as a bridge corresponds to the configuration of a managed switch. Most managed switches also can support IP forwarding and even some routing protocols. A switch that can also operate as an IP router is also referred to as Layer 3 switch.

The lab instructions suppose that the Ethernet switches in GNS3 used in the network configurations are unmanaged switches. If the switches used in the labs are actually hubs, the lab exercises still work, but the forwarding behavior is different from an unmanaged switch (due to the absence of the learning algorithm). If the switches used in the labs are actually managed switches, we never take advantage of their configurability. To avoid issues in our exercises, we replace the GNS3 Ethernet Switch with a **Hub** to avoid any issues.

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 part of Lab 5, you learn how to configure a Cisco router as a bridge. Parts 2 and 3 explore how MAC 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.

The configuration of the equipment in Lab 5 is changed several times during the course of the lab. The IPv4 address configuration of the Linux PCs is as shown in Table 5.1. Note that all IP addresses have the same netmask.

|  |  |
| --- | --- |
| Linux PC | Interface  *eth0* |
| *PC1* | 10.0.1.11/24 |
| *PC2* | 10.0.1.22/24 |
| *PC3* | 10.0.1.33/24 |
| *PC4* | 10.0.1.44/24 |

Table 5.1. IPv4 addresses of the PCs.

## Part 1. 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 5.1. *Router1* is configured as a bridge that connects the two Ethernet segments.

Diagram

Description automatically generated

Figure 5.1. Network Topology for Parts 1 and 2-b.

### Exercise 1-a. Setup of network configuration

After the network is configured, as I Figure 5.1 above, for the Linux PCs, you are asked to record the MAC addresses of the Cisco routers.

1. Set up the network topology of PCs and *Router1* as shown in Figure 5.1.
2. Configure the *eth0* interfaces of the Linux PCs with the IP addresses given in Table 5.1.
3. ![A picture containing text, iPod, electronics

   Description automatically generated]()Since, throughout this part, you frequently work with MAC addresses, you should record the MAC addresses of the Linux PCs at the start of each part. Open consoles on the PCs and record the MAC address of Ethernet interface eth0 with the command ip addr show. Enter the MAC addresses of eth0 for each PC in Table 5.2. Save the table.

|  |  |
| --- | --- |
| Linux PC | MAC address of interface *eth0* |
| PC1 | **aa:bb:cc:dd:ee:11** |
| PC2 | **aa:bb:cc:dd:ee:22** |
| PC3 | **aa:bb:cc:dd:ee:33** |
| PC4 | **aa:bb:cc:dd:ee:44** |

Table 5.2. MAC addresses of the PCs.

1. Disable IPv6 on all PCs.
2. Disable IPv6 on all PCs. On *PC1*, the commands are

PC1$ sudo sysctl -w net.ipv6.conf.all.disable\_ipv6=1

PC1$ sudo sysctl -w net.ipv6.conf.default.disable\_ipv6=1

The reason for disabling IPv6 is that IPv6 enabled hosts send periodic ICMP queries, that interfere with the experiments in this Lab.

1. On *Router1*, use the command show interfacesto display the MAC addresses of the FastEthernet interfaces. Record the MAC addresses in Table 5.3 below:

|  |  |  |
| --- | --- | --- |
| Router | MAC address of interface *FastEthernet0/0* | MAC address of interface *FastEthernet1/0* |
| Router1 | **cc01.4b70.0000** | **cc01.4b70.0010** |

Table 5.3. MAC addresses of the FastEthernet interfaces on Cisco Router1.

### Exercise 1-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.

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 6.  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. |

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)# no cdp run

Router1(config)# bridge 1 protocol ieee

Router1(config)# bridge 1 priority 128

Router1(config)# interface FastEthernet0/0

Router1(config-if)# **no keepalive**

Router1(config-if)# bridge-group 1

Router1(config-if)# bridge-group 1 spanning-disabled

Router1(config-if)# no shutdown

Router1(config-if)# interface FastEthernet1/0

Router1(config-if)# **no keepalive**

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, gets rid of the CDP messages “**no cdp run**”, and set up *Router1* as a bridge that runs with priority 128. Both FastEthernet interfaces are assigned to the bridge with the **no keepalive** command enabled to stop the LOOP messages. Here the spanning tree protocol is disabled.

1. Delete all entries in the neighbor caches of *PC1* and *PC3*.
2. Issue a ping and a traceroute command from *PC1* to *PC3* with

PC1$ ping –c 2 10.0.1.33

PC1$ traceroute 10.0.1.33

 Take a screenshot of the output.

Lab Questions/Report:

1. Include the screen capture of the output of the ping and *traceroute* commands.
2. Why is it not possible to issue a ping command to *Router1*? NO IP SETUP FOR R1 INTERFACEs.

## Part 2. 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 MAC 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 5.2.

Diagram

Description automatically generated

Figure 5.2. Network Topology for Part 2.

### Exercise 2-a. Exploring the learning algorithm of bridges

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

1. Set up the network configuration as shown in Figure 5.2.
2. Configure the PCs using Table 5.1 for the IP addresses.
3. Record the Mac addresses of the PCs and the routers and save in the table below:

|  |  |
| --- | --- |
| Linux PC | MAC address of interface *eth0* |
| PC1 | **aa:bb:cc:dd:ee:11** |
| PC2 | **aa:bb:cc:dd:ee:22** |
| PC3 | **aa:bb:cc:dd:ee:33** |
| PC4 | **aa:bb:cc:dd:ee:44** |

Table 5.4

1. Disable IPv6 on all PCs. The commands are shown in Exercise 1-a (Step 5).
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 1-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.”
3. Start to capture traffic with *Wireshark* on the *eth0* interfaces of *PC1, PC2, PC3*, and *PC4*.
4. Clear the neighbor caches on all PCs.
5. Now, issue a set of ping commands.

PC1$ ping –c 2 10.0.1.22

PC2$ ping –c 2 10.0.1.11

PC2$ ping –c 2 10.0.1.44

PC3$ ping –c 2 10.0.1.22

1. After each command:
   * Take a screenshot of the MAC table at each bridge with the command *show bridge*.
   * ![A picture containing text, iPod, electronics

     Description automatically generated]()From the Wireshark data, observe how the *ICMP Echo Request* and *Reply* packets (and related ARP packets) are forwarded by the bridges.
2. Stop the Wireshark traffic capture on the PCs and save.

Lab Questions/Report

1. For each ping command in Step 8, include the screenshots of the MAC tables.
2. Using the Wireshark output, explain the entries in the bridge tables. Explain how the traffic generated by the ping command changed the MAC tables of the bridges overtime using what you have learnt about Learning Bridges. (Use the Mac addresses saved in Table 5.4 to help you figure out what is going on)
   1. First messages are broadcast which creates mac entries for source which are used thereafter.

### Exercise 2-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.

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 ping commands from *PC1*:

PC1$ ping –c 3 10.0.1.22

PC1$ ping –c 3 10.0.1.33

1. Now issue a continuous ping from *PC1* to *PC2*.

PC1$ ping 10.0.1.22

![A picture containing text, iPod, electronics

Description automatically generated]()The ping should be successful.

1. Wait 30secs, then disconnect *PC2* from Hub2 and connect it to the same hub that *PC4* is connected to (i.e., disconnect it from the Hub2 and connect it to Hub4). Record and save the *time* when you moved *PC2* with the command

PC4$ date

1. You will observe that the pings on *PC1* will fail – for a while.

Since *Router2* does not know that *PC2* has moved, it does not forward the *ICMP Echo Request* packet, and the 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.

![A picture containing text, iPod, electronics

Description automatically generated]()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*.

1. Record and save the time at which the ping at *PC1* to *PC2* becomes successful again with

PC4$ date

Using this new value for date and what you recorded previously in Step 3, calculate the amount of time*, T1*, until the ping from *PC1* to *PC2* is again successful after *PC2* has been moved to a different hub.

IT TAKE BETWEEN 10-20 seconds

1. Now issue a continuous ping from *PC1* to *PC3*.

PC1$ ping 10.0.1.33

![A picture containing text, iPod, electronics

Description automatically generated]()The ping should be successful.

1. Wait 30secs, then disconnect *PC3* from Hub3 and connect it to the same hub that *PC4* is connected to (i.e., disconnect it from the Hub3 and connect it to Hub4). Record and save the *time* when you moved *PC3* with the command

PC4$ date

1. ![A picture containing text, iPod, electronics

   Description automatically generated]()You will notice that similarly to what happened when you moved *PC2*, the ping command will fail after the move.
2. Immediately you observe the failed pings on PC1, issue two pings from *PC3* to *PC1* with the command

PC3$ ping -c 2 10.0.1.11

1. You will notice that on *PC1*, the ping becomes successful again after the pings were issued in Step 9. Record and save the time when you observe the successful pings.

PC4$ date

Using this new value for date and what you recorded previously in Step 7, calculate the amount of time, *T2*, until the ping from *PC1* to *PC3* is again successful after *PC3* has been moved Hub4.

It takes 10-20 seconds without second ping, but with ping it’s immediate

Lab Questions/ Report

1. Include the two times *T1* and *T2* for the pings to become successful again.
2. Are they different? **Explain what you observe and why.**
   1. **For PC2 it takes 10-20 seconds for PC3 it’s immediate after second ping.**
   2. **Because the route for the PC1->PC3 pings were failing from R3 no forwarding PC1s ping packets because of previous mapping of PC3 to f0/0 but after PC3 pings via Hub4 and via f1/0 of R3 then after that R3 start learns about new PC3 mapping via f1/0 and forwards PC1s pings.**

## Part 3. Spanning Tree Protocol

The learning algorithm from Part 2 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(HUB), 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 3-a. Configuring a topology that results in forwarding loops

1. Set up the network topology as shown in Figure 5.3. Configure *Router1*, *Router2*, *Router3*, *Router4* and *Router5* as done in Exercise 2-b.

|  |  |
| --- | --- |
| A picture containing light  Description automatically generated | **Note:** For the time being, do not connect interface f1/0 of *Router2* and f1/0 of *Router4* (shown as red dashed line Figure 5.3). Adding these connections will create a forwarding loop. The connections will be made when you have set up the tools that allow you to observe the forwarding loop. |

1. Configure *PC1*, *PC3*, and *PC4* as hosts with IP addresses on the *eth0* interfaces as shown in Table 5.1.
2. Disable IPv6 on all PCs. The commands are shown in Exercise 1-a (Step 5).

Timeline

Description automatically generated

Figure 5.3. Network topology for Part 3

1. Check the MAC addresses of the Linux PCs and those of the Cisco routers and record in Table 5.5 and Table 5.6 below.

|  |  |
| --- | --- |
| Linux PC | MAC address of interface *eth0* |
| PC1 | **aa:bb:cc:dd:ee:11** |
| PC2 | **aa:bb:cc:dd:ee:22** |
| PC3 | **aa:bb:cc:dd:ee:33** |
| PC4 | **aa:bb:cc:dd:ee:44** |

Table 5.5 Mac Addresses of PCs

|  |  |  |
| --- | --- | --- |
| Router | MAC address of interface *FastEthernet0/0* | MAC address of interface *FastEthernet1/0* |
| Router1 | **cc01.200c.0000** | **cc01.200c.0010** |
| Router2 | **cc02.10a0.0000** | **cc02.10a0.0010** |
| Router3 | **cc03.4bf8.0000** | **cc03.4bf8.0010** |
| Router4 | **cc04.2f18.0000** | **cc04.2f18.0010** |
| Router5 | **cc05.5198.0000** | **cc05.5198.0010** |

Table 5.6 Mac Addresses of Routers (bridges)

1. Since ARP traffic interferes with the forwarding and bridge learning operations that you need to observe, pre-configure the neighbor caches 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 (i.e., Mac addresses) for interface *eth0* of *PC1, PC3*, and *PC4*, with the following commands.

PC1$ sudo ip neigh del 10.0.1.33 dev eth0

PC1$ sudo ip neigh del 10.0.1.44 dev eth0

PC1$ sudo ip neigh add 10.0.1.33 lladdr MACaddress dev eth0

PC1$ sudo ip neigh add 10.0.1.44 lladdr MACaddress dev eth0

PC3$ sudo ip neigh del 10.0.1.11 dev eth0

PC3$ sudo ip neigh del 10.0.1.44 dev eth0

PC3$ sudo ip neigh add 10.0.1.11 lladdr MACaddress dev eth0

PC3$ sudo ip neigh add 10.0.1.44 lladdr MACaddress dev eth0

PC4$ sudo ip neigh del 10.0.1.11 dev eth0

PC4$ sudo ip neigh del 10.0.1.33 dev eth0

PC4$ sudo ip neigh add 10.0.1.11 lladdr MACaddress dev eth0

PC4$ sudo ip neigh add 10.0.1.33 lladdr MACaddress dev eth0

Note that lladr in the above command is “(lowercase L)(lowercase L)addr”. In the above commands, `*MACaddres’s* is the 48-bit MAC address of the corresponding interface from Table 5.5. 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 3-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 5.3 forward packets in a loop.

1. Once the network topology of Figure 5.3 is set up, verify that the spanning tree protocol (STP) is disabled on all bridges (Cisco routers). Shown here for *Router1*

Router1# **show running-config**

Verify that the line `bridge-group 1 spanning-disabled’ is displayed for both FastEthernet interfaces.

1. Clear the contents of the MAC tables on all bridges (Cisco routers) with the following command:

Router1# **clear bridge**

1. Start *Wireshark* on the eth0 interfaces of PC1, PC3 and PC4.
2. Now complete the topology in Figure 5.3 by connecting interface f1/0 of *Router2* and f1/0 of *Router4* to the hubs.
3. Issue a ping command from *PC1* to *PC4*.

PC1$ ping –c 1 10.0.1.44

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).

1. Wait for several seconds and break the loop by ***shutting down*** the FastEthernet1/0interfaces of both *Router2* and *Router4*. Otherwise, the packets will loop forever and cause the bridges to become overloaded and stop forwarding packets.
2. Stop the traffic capture with *Wireshark o*n 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.)**
3. Take a screenshot of the MAC table of all bridges, that is*, Router1, Router2, Router3, Router4*, and *Router5*.



Lab Questions/Report:

1. Use the Wireshark data saved in Step 7 to find a single packet, that is caught in a forwarding loop. Hint: You should see the same packet on each Wireshark capture.

* The Identification field in the IP header can be used to uniquely tag a packet.

1. Use the output from *Wireshark* and the MAC tables to explain why some packets are looping indefinitely? Hard to observe…. But doable. ☺
   1. Because of the switches multicasting and routers relearning paths to PC1 and PC4.
   2. Can be verified by seeing `show bridge 1` the mac entries which are wrong like below.
   3. PC4’s connected router map the PC1 back to PC4 ports and evidence of looping.
   4. The biggest evidence is the wireshark requests.

### Exercise 3-c. Enabling the spanning tree protocol

Next, you enable the spanning tree protocol on the bridges in Figure 5.3, 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 5.7 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*.” |

|  |  |  |
| --- | --- | --- |
| 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. |

Table 5.7 Fields of a BPDU relevant to the spanning tree construction.

|  |
| --- |
| **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 5.3 (including the connections between f1/0 of routers *Router2* and *Router4* to the hub as shown in the figure) and trace how the spanning tree is constructed.

1. Delete all entries in the MAC tables on all bridges (*Router1, Router2, Router3, Router4*, and *Router5*), as done in Step 2 of Exercise 3-b.
2. If you disconnected interface f1/0 of *Router2* and *Router4* from the hubs, reconnect them.
3. Next, enable the spanning tree protocol on all bridges (*Router1, Router2, Router3, Router4, Router5*). On the Cisco routers, the following commands, shown here for *Router1*, must be executed:

Router1# configure terminal

Router1(config)# interface FastEthernet0/0

Router1(config-if)# bridge-group 1

Router1(config-if)# no bridge-group 1 spanning-disabled

Router1(config-if)# no shutdown

Router1(config-if)# interface FastEthernet1/0

Router1(config-if)# bridge-group 1

Router1(config-if)# no bridge-group 1 spanning-disabled

Router1(config-if)# no shutdown

Router1(config-if)# exit

1. Set the bridge priority to *128* on all bridges. Table 5.7 shows valid priority values which could be used by *brctl*. 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

| 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 |

Table 5.7. Valid bridge priority values for *brctl*.

1. 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 displaying the spanning tree information on the Cisco routers. For *Router1*, the command is

Router1# show spanning–tree

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?
4. Delete the entries in the neighbor caches of all PCs, including the static entries that you added in Exercise 3-a.
5. Run *Wireshark* to capture traffic on the *eth0* interfaces of all PCs.
6. 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. **Because the tree converges and everyone agrees on the root of the tree**
2. Take screenshot 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 : 01:80:c2:00:00:00 (spanning tree for bridges.)
   * the Ethernet header. IEEE802.3 have source destination and padding.
5. Issue a ping command from *PC1* to *PC4* by typing

PC1$ ping –c 1 10.0.1.44

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.
   * I can verify that PC3 intercepts the pings, being in the path
2. Confirm your answer with the spanning tree information from Step 5.

BUT THE PATH IS PC1->R1->R4->PC4

1. Terminate all *Wireshark* applications.



Lab Questions/Report: DONE ALL

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 you made in Step 5 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 (**Part 4**) continues with the network configuration from Figure 5.3, and partially relies on the results in Exercise 3-c. You may want to directly continue to **Part 4.** |

## Part 4. Reconfiguration 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 4-a. Reconfiguration of the spanning tree

You simulate a failed link in the network shown in Figure 5.3 and observe how the bridges adjust to a change in the network topology.

1. Continue with the network configuration you set up in Exercise 3-c for Figure 5.3.   
   Make sure that the spanning tree protocol is enabled on *Router1, Router2, Router3, Router4*, and *Router5*, and that the priorities of all bridges are set to 128.
2. Start to capture traffic with *Wireshark* on the *eth0* interfaces of *PC1, PC3*, and *PC4*.
3. Issue a ping command from *PC1* to *PC4* that sends an *ICMP Echo Request* every second:

PC1$ ping –i 1 10.0.1.44

1. If you did not *just* complete **Part 3**, determine the root bridge of the spanning tree. You can infer this from the spanning tree information of each bridge. You this with the command `show spanning–tree’ on a Cisco router.
2. Disconnect one of the interfaces of the root bridge from the hub.

* 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. **50 Seconds**

1. Display the spanning tree information on the bridges (see Step 5 in Exercise 3-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.

1. Terminate all Wireshark sessions.

Lab Questions/Report

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

### Exercise 4-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.

1. Continue with the topology at the end of Exercise 4-a (Do not reconnect the interface of the root bridge that was disconnected in Step 5 of Exercise 4-a).
2. Select one of the bridges from Figure 5.3, other than the bridge that was the root bridge in Exercise 3-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.
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

1. Display the spanning tree information on the bridges (see Step 5 in Exercise 3-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 5. 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.

Timeline

Description automatically generated

Figure 5.4. Network topology for Part 5.

### Exercise 5-a. Setting up the network configuration

1. Create a network topology of Linux PCs and Cisco routers as shown in Figure 5.4.
2. Disable IPv6 on all PCs. The commands are shown in Exercise 1-a (Step 2).
3. Skip this step, unless you continue from **Part 4**.

Disable bridging on *Router2*, and *Router3*. You disable the bridging functions by typing (shown here for *Router2)*

Router2# configure terminal  
Router2(config)# **no** bridge 1   
Router2(config)# end

1. Skip this step if you continue from **Part 4** (where *Router1* and *Router4* are configured as bridges). Configure *Router1* and *Router4* as bridges using the commands from Exercise 1-b (Step 1).
2. Complete the IP configuration of the Linux PCs, *Router2*, and *Router3* as given in Tables 5.8 and 5.9. Note that *PC4* has a /16 prefix. All other configured IP addresses use a /24 prefix.
3. If you continue from **Part 4**, delete the IP address of interface eth0 on *PC3* and *PC4* before adding the address given in Table 5.8.
4. Configure *Router2* and *Router3* as IP routers (**ip routing**). Also, disable Proxy ARP on all interfaces of the IP routers. 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 FastEthernet0/0  
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 FastEthernet1/0

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

|  |  |  |  |
| --- | --- | --- | --- |
| Linux PC | Ethernet Interface *eth0* | Ethernet Interface *eth1* | Default Gateway |
| *PC1* | 10.0.1.11/24 | Disabled | 10.0.1.2 |
| *PC2* | 10.0.3.22/24 | Disabled | 10.0.3.2 |
| *PC3* | 10.0.4.33/24 | Disabled | 10.0.4.3 |
| *PC4* | 10.0.4.44/16 | Disabled | 10.0.4.3 |

Table 5.8. IP configuration of the Linux PCs for Part 5

| Cisco Router | Configured As | Configuration information |
| --- | --- | --- |
| *Router1 Router4* | Bridge | Enable bridging on both interfaces *FastEthernet0/0* and *FastEthernet1/0*  The spanning tree protocol can be enabled or disabled  IP addresses need not be configured |
| *Router2* | IP router | IP configuration on *FastEthernet0/0*: 10.0.3.2/24  IP configuration on *FastEthernet1/0*: 10.0.1.2/24  Default gateway set to 10.0.3.3  IP forwarding is enabled |
| *Router3* | IP router | IP configuration on *FastEthernet0/0*: 10.0.3.3/24  IP configuration on *FastEthernet1/0:* 10.0.4.3/24  Default gateway set to 10.0.3.2  IP forwarding is enabled |

Table 5.9. IP configuration of the Cisco routers for Part 5

### Exercise 5-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.

1. Clear the MAC tables on all bridges and clear the neighbor cache on all hosts and IP routers. Shown here for *Bridge1* (i.e., *Router1*)

Router1# clear bridge

![A picture containing text, iPod, electronics

Description automatically generated]()Router1# clear ip route \*

1. Check the MAC addresses of the Linux PCs and those of the Cisco routers and record in Table 5.10 and Table 5.11 below.

|  |  |
| --- | --- |
| Linux PC | MAC address of interface *eth0* |
| PC1 | **aa:bb:cc:dd:ee:11** |
| PC2 | **aa:bb:cc:dd:ee:22** |
| PC3 | **aa:bb:cc:dd:ee:33** |
| PC4 | **aa:bb:cc:dd:ee:44** |

Table 5.10 Mac Addresses of PCs

|  |  |  |
| --- | --- | --- |
| Router | MAC address of interface *FastEthernet0/0* | MAC address of interface *FastEthernet1/0* |
| Router1 | **cc01.4fc4.0000** | **cc01.4fc4.0010** |
| Router2 | **cc02.0f30.0000** | **cc02.0f30.0010** |
| Router3 | **cc03.33f8.000** | **cc03.33f8.0010** |
| Router4 | **cc04.4b14.0000** | **cc04.4b14.0010** |

Table 5.11 Mac Addresses of Cisco Routers

1. Run *Wireshark* to capture traffic on the *eth0* interface of each Linux PC.
2. Issue the following set of ping commands. Use the output of Wiresharkto determine the path of the *ICMP Echo Request* and *Rep*ly packets. Note that not all ping commands will be successful.
3. *PC1* 🡪 *PC2*: PC1->(R1)->R2->PC2 (because of default route) also we can verify that PC4 never get pings, and also the R4 does gets ICMP req/reply at f1/0 but it doesn’t forward to f0/0 because it has learned that source and destination ie. PC1 and R2 are at f1/0 only.

PC1$ ping -c 3 10.0.3.22

1. *PC1*🡪*PC3*: PC1->(R1)->R2->R3->(R4)->PC3 (because of default route) the premature broadcast ping request to PC3 by hub3 has the no response seen, PC3 only responds to pings from R3->(R4) via it’s default route.

PC1$ ping -c 3 10.0.4.33

1. *PC1*🡪 *PC4*: PC1->R2->R3->PC4 (because default route) BUT REPLY PC4->(R4)->PC1 from MAC(cos /16)

*PC1*$ *ping* -c 3 10.0.4.44

1. *PC2*🡪 *PC4*: (success request PC2->R2->R3->PC4)(we also got the R2’s icmp redirects to PC2 for R3), fail reply PC4->(arp for 10.0.3.22) cos /16, AND PROXY-ARP is disabled

PC2$ ping –c 3 10.0.4.44

1. *PC4*🡪 *PC2*: FAIL Destnation Host Unreachable, fail request PC4->(arp for 10.0.3.22) cos /16, and proxy ARP is disabled.

![A picture containing text, iPod, electronics

Description automatically generated]()PC4$ ping -c 3 10.0.3.22

1. Save each Wireshark capture to answer the questions in the Lab Report
2. Stop the *Wireshark* packet captures *o*n all PCs.

Lab Questions/Report

1. Determine which of the ping commands are successful and which fail.
2. Use the data displayed by *Wireshark* outputs to determine the route of *the ICMP Echo Request* and *Reply* packets (**e.g.**, *PC1* 🡪 *Router1* 🡪 *Router2* 🡪 *Router4* 🡪 *PC4*).
3. The *Wireshark* sessions may 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:
   1. The MAC addresses in **Ethernet** packet headers can be matched against the MAC addresses of the Cisco routers and the PCs (Table 5.10 and 5.11)
   2. 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.
4. For each path, provide an explanation why a certain route is taken by the *ICMP Echo Request* and *Reply* packets*.*