HACETTEPE UNIVERSITY
COMPUTER ENGINEERING DEPARTMENT
COMPUTER NETWORKS LABORATORY

EXPERIMENT

Local Area Networks (LANs) and Ethernet

AIM

In this lab, you are going to create a LAN using HUBs and Switches. You will learn preparation of straight-through and cross-over network cables, to connect related devices. You will learn differences of HUB and Switch in Local Area Network using packet analyzer.

INTRODUCTION

A local area network (LAN) is a computer network that interconnects computers within a limited area such as a residence, school, laboratory, university campus or office building[1] and has its network equipment and interconnects locally managed. By contrast, a wide area network (WAN), not only covers a larger geographic distance, but also generally involves leased telecommunication circuits or Internet links.

Ethernet and Wi-Fi are the two most common transmission technologies in use for local area networks. Historical technologies include ARCNET, Token ring, and AppleTalk.

Reading part from course text book, Chapter 5

5.4 Switched Local Area Networks

Having covered broadcast networks and multiple access protocols in the previous section, let’s turn our attention next to switched local networks. Figure 5.15 shows a switched local network connecting three departments, two servers and a router with four switches. Because these switches operate at the link layer, they switch link-layer frames (rather than network-layer datagrams), don’t recognize network-layer addresses, and don’t use routing algorithms like RIP or OSPF to determine paths through the network of layer-2 switches. Instead of using IP addresses, we will soon see that they use link-layer addresses to forward linklayer frames through the network of switches.

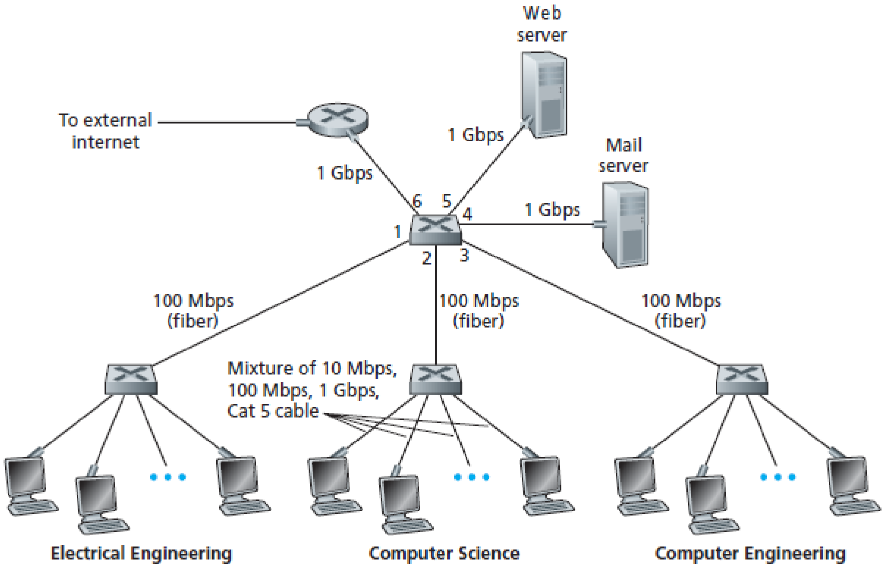


Figure 5.15 ♦ An institutional network connected together by four switches

We'll begin our study of switched LANs by first covering link-layer addressing (Section 5.4.1). We then examine the celebrated Ethernet protocol (Section 5.5.2). After examining link-layer addressing and Ethernet, we'll look at how link-layer switches operate (Section 5.4.3), and then see (Section 5.4.4) how these switches are often used to build large-scale LANs.

5.4.1 Link-Layer Addressing and ARP

Hosts and routers have link-layer addresses. Now you might find this surprising, recalling from Chapter 4 that hosts and routers have network-layer addresses as well. You might be asking, why in the world do we need to have addresses at both the network and link layers? In addition to describing the syntax and function of the link-layer addresses, in this section we hope to shed some light on why the two layers of addresses are useful and, in fact, indispensable. We'll also cover the Address Resolution Protocol (ARP), which provides a mechanism to translate IP addresses to link-layer addresses.

MAC Addresses

In truth, it is not hosts and routers that have link-layer addresses but rather their adapters (that is, network interfaces) that have link-layer addresses. A host or router with multiple network interfaces will thus have multiple link-layer addresses associated with it, just as it would also have multiple IP addresses associated with it. It's important to note, however, that link-layer switches do not have link-layer addresses associated with their interfaces that connect to hosts and routers. This is because the job of the link-layer switch is to carry datagrams between hosts and routers; a switch does this job transparently, that is, without the host or router having to explicitly address the frame to the intervening switch. This is illustrated in Figure 5.16. A link-layer address is variously called a **LAN address**, a **physical address**, or a **MAC address**. Because MAC address seems to be the most popular term, we'll henceforth refer to link-layer addresses as MAC addresses. For most LANs (including Ethernet and 802.11 wireless LANs), the MAC address is 6 bytes long, giving 248 possible MAC addresses. As shown in Figure 5.16, these 6-byte addresses are typically expressed in hexadecimal notation, with each byte of the address expressed as a pair of hexadecimal numbers. Although MAC addresses were designed to be permanent, it is now possible to change an adapter's MAC address via software. For the rest of this section, however, we'll assume that an adapter's MAC address is fixed.

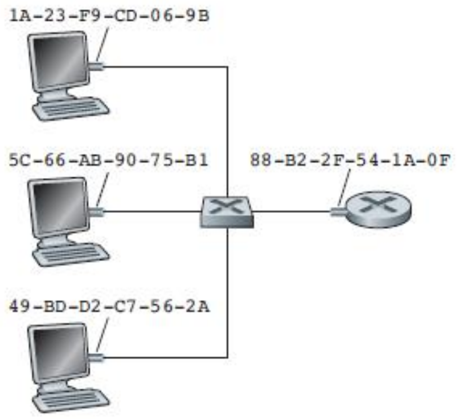


Figure 5.16 ♦ Each interface connected to a LAN has a unique MAC address

One interesting property of MAC addresses is that no two adapters have the same address. This might seem surprising given that adapters are manufactured in many countries by many companies. How does a company manufacturing adapters in Taiwan make sure that it is using different addresses from a company manufacturing adapters in Belgium? The answer is that the IEEE manages the MAC address space. In particular, when a company wants to manufacture adapters, it purchases a chunk of the address space consisting of 224 addresses for a nominal fee. IEEE allocates the chunk of 224 addresses by fixing the first 24 bits of a MAC address and letting the company create unique combinations of the last 24 bits for each adapter.

An adapter's MAC address has a flat structure (as opposed to a hierarchical structure) and doesn't change no matter where the adapter goes. A laptop with an Ethernet interface always has the same MAC address, no matter where the computer goes. A smartphone with an 802.11 interface always has the same MAC address, no matter where the smartphone goes. Recall that, in contrast, IP addresses have a hierarchical structure (that is, a network part and a host part), and a host's IP addresses needs to be changed when the host moves, i.e, changes the network to which it is attached. An adapter's MAC address is analogous to a person's social security number, which also has a flat addressing structure and which doesn't change no matter where the person goes. An IP address is analogous to a person's postal address, which is hierarchical and which must be changed whenever a person moves. Just as a person may find it useful to have both a postal address and a social security number, it is useful for a host and router interfaces to have both a network-layer address and a MAC address.

When an adapter wants to send a frame to some destination adapter, the sending adapter inserts the destination adapter's MAC address into the frame and then sends the frame into the LAN. As we will soon see, a switch occassionally broadcasts an incoming frame onto all of its interfaces. We'll see in Chapter 6 that 802.11 also broadcasts frames. Thus, an adapter may receive a frame that isn't addressed to it. Thus, when an adapter receives a frame, it will check to see whether the destination MAC address in the frame matches its own MAC address. If there is a match, the adapter extracts the enclosed datagram and passes the datagram up the protocol stack. If there isn't a match, the adapter discards the frame, without passing the network-layer datagram up. Thus, the destination only will be interrupted when the frame is received.

However, sometimes a sending adapter *does* want all the other adapters on the LAN to receive and *process* the frame it is about to send. In this case, the sending adapter inserts a special MAC **broadcast address** into the destination address field of the frame. For LANs that use 6-byte addresses (such as Ethernet and 802.11), the broadcast address is a string of 48 consecutive 1s (that is, FF-FF-FF-FF-FFFF in hexadecimal notation).

Address Resolution Protocol (ARP)

Because there are both network-layer addresses (for example, Internet IP addresses) and link-layer addresses (that is, MAC addresses), there is a need to translate between them. For the Internet, this is the job of the **Address Resolution Protocol (ARP)** [RFC 826].

To understand the need for a protocol such as ARP, consider the network shown in Figure 5.17. In this simple example, each host and router has a single IP address and single MAC address. As usual, IP addresses are shown in dotted-decimal notation and MAC addresses are shown in hexadecimal notation. For the purposes of this discussion, we will assume in this section that the switch broadcasts all frames; that is, whenever a switch receives a frame on one interface, it forwards the frame on all of its other interfaces. In the next section, we will provide a more accurate explanation of how switches operate.

Now suppose that the host with IP address 222.222.222.220 wants to send an IP datagram to host 222.222.222.222. In this example, both the source and destination are in the same subnet, in the addressing sense of Section 4.4.2. To send a datagram, the source must give its adapter not only the IP datagram but also the MAC address for destination 222.222.222.222. The sending adapter will then construct a link-layer frame containing the destination's MAC address and send the frame into the LAN.

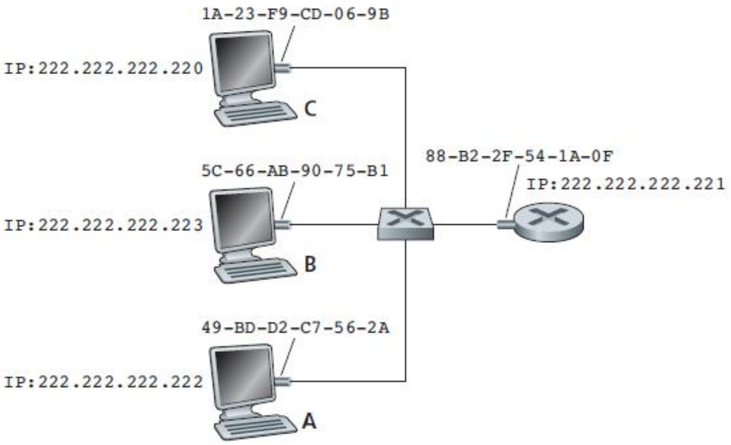


Figure 5.17 ♦ Each interface on a LAN has an IP address and a MAC address

Now suppose that host 222.222.222.220 wants to send a datagram that is IPaddressed to another host or router on that subnet. The sending host needs to obtain the MAC address of the destination given the IP address. This task is easy if the sender's ARP table has an entry for the destination node. But what if the ARP table doesn't currently have an entry for the destination? In particular, suppose 222.222.222.220 wants to send a datagram to 222.222.222.222. In this case, the sender uses the ARP protocol to resolve the address. First, the sender constructs a special packet called an **ARP packet**. An ARP packet has several fields, including the sending and receiving IP and MAC addresses. Both ARP query and response packets have the same format. The purpose of the ARP query packet is to query all the other hosts and routers on the subnet to determine the MAC address corresponding to the IP address that is being resolved.

Returning to our example, 222.222.222.220 passes an ARP query packet to the adapter along with an indication that the adapter should send the packet to the MAC broadcast address, namely, FF-FF-FF-FF-FF-FF. The adapter encapsulates the ARP packet in a link-layer frame, uses the broadcast address for the frame's destination address, and transmits the frame into the subnet. Recalling our social security number/postal address analogy, an ARP query is equivalent to a person shouting out in a crowded room of cubicles in some company (say, AnyCorp): "What is the social security number of the person whose postal address is Cubicle 13, Room 112, AnyCorp, Palo Alto, California?" The frame containing the ARP query is received by all the other adapters on the subnet, and (because of the broadcast address) each adapter passes the ARP packet within the frame up to its ARP module. Each of these ARP modules checks to see if its IP address matches the destination IP address in the ARP packet. The one with a match sends back to the querying host a response ARP packet with the desired mapping. The querying host 222.222.222.220 can then update its ARP table and send its IP datagram, encapsulated in a link-layer frame whose destination MAC is that of the host or router responding to the earlier ARP query.

There are a couple of interesting things to note about the ARP protocol. First, the query ARP message is sent within a broadcast frame, whereas the response ARP message is sent within a standard frame. Before reading on you should think about why this is so. Second, ARP is plug-and-play; that is, an ARP table gets built automatically—it doesn't have to be configured by a system administrator. And if a host becomes disconnected from the subnet, its entry is eventually deleted from the other ARP tables in the subnet.

Students often wonder if ARP is a link-layer protocol or a network-layer protocol. As we've seen, an ARP packet is encapsulated within a link-layer frame and thus lies architecturally above the link layer. However, an ARP packet has fields containing link-layer addresses and thus is arguably a link-layer protocol, but it also contains network-layer addresses and thus is also arguably a networklayer protocol. In the end, ARP is probably best considered a protocol that straddles the boundary between the link and network layers—not fitting neatly into the simple layered protocol stack we studied in Chapter 1. Such are the complexities of real-world protocols!

5.4.2 Ethernet

Ethernet has pretty much taken over the wired LAN market. In the 1980s and the early 1990s, Ethernet faced many challenges from other LAN technologies, including token ring, FDDI, and ATM. Some of these other technologies succeeded in capturing a part of the LAN market for a few years. But since its invention in the mid-1970s, Ethernet has continued to evolve and grow and has held on to its dominant position. Today, Ethernet is by far the most prevalent wired LAN technology, and it is likely to remain so for the foreseeable future. One might say that Ethernet has been to local area networking what the Internet has been to global networking.

There are many reasons for Ethernet's success. First, Ethernet was the first widely deployed high-speed LAN. Because it was deployed early, network administrators became intimately familiar with Ethernet—its wonders and its quirks— and were reluctant to switch over to other LAN technologies when they came on the scene. Second, token ring, FDDI, and ATM were more complex and expensive than Ethernet, which further discouraged network administrators from switching over. Third, the most compelling reason to switch to another LAN technology (such as FDDI or ATM) was usually the higher data rate of the new technology; however, Ethernet always fought back, producing versions that operated at equal data rates or higher. Switched Ethernet was also introduced in the early 1990s, which further increased its effective data rates. Finally, because Ethernet has been so popular, Ethernet hardware (in particular, adapters and switches) has become a commodity and is remarkably cheap.

The original Ethernet LAN was invented in the mid-1970s by Bob Metcalfe and David Boggs. The original Ethernet LAN used a coaxial bus to interconnect the nodes. Bus topologies for Ethernet actually persisted throughout the 1980s and into the mid-1990s. Ethernet with a bus topology is a broadcast LAN—all transmitted frames travel to and are processed by *all* adapters connected to the bus. Recall that we covered Ethernet's CSMA/CD multiple access protocol with binary exponential backoff in Section 5.3.2.

By the late 1990s, most companies and universities had replaced their LANs with Ethernet installations using a hub-based star topology. In such an installation the hosts (and routers) are directly connected to a hub with twisted-pair copper wire. A **hub** is a physical-layer device that acts on individual bits rather than frames. When a bit, representing a zero or a one, arrives from one interface, the hub simply re-creates the bit, boosts its energy strength, and transmits the bit onto all the other interfaces. Thus, Ethernet with a hub-based star topology is also a broadcast LAN—whenever a hub receives a bit from one of its interfaces, it sends a copy out on all of its other interfaces. In particular, if a hub receives frames from two different interfaces at the same time, a collision occurs and the nodes that created the frames must retransmit.

In the early 2000s Ethernet experienced yet another major evolutionary change. Ethernet installations continued to use a star topology, but the hub at the center was replaced with a **switch**. We'll be examining switched Ethernet in depth later in this chapter. For now, we only mention that a switch is not only "collision-less" but is also a bona-fide store-and-forward packet switch; but unlike routers, which operate up through layer 3, a switch operates only up through layer 2.

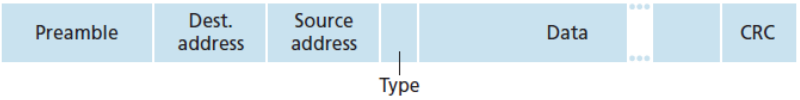


Figure 5.20 ♦ Ethernet frame structure

EXPERIMENT STEPS

Using repeater HUB

1. In the first part, you have to create a LAN using repeater HUBs as shown in Figure-1

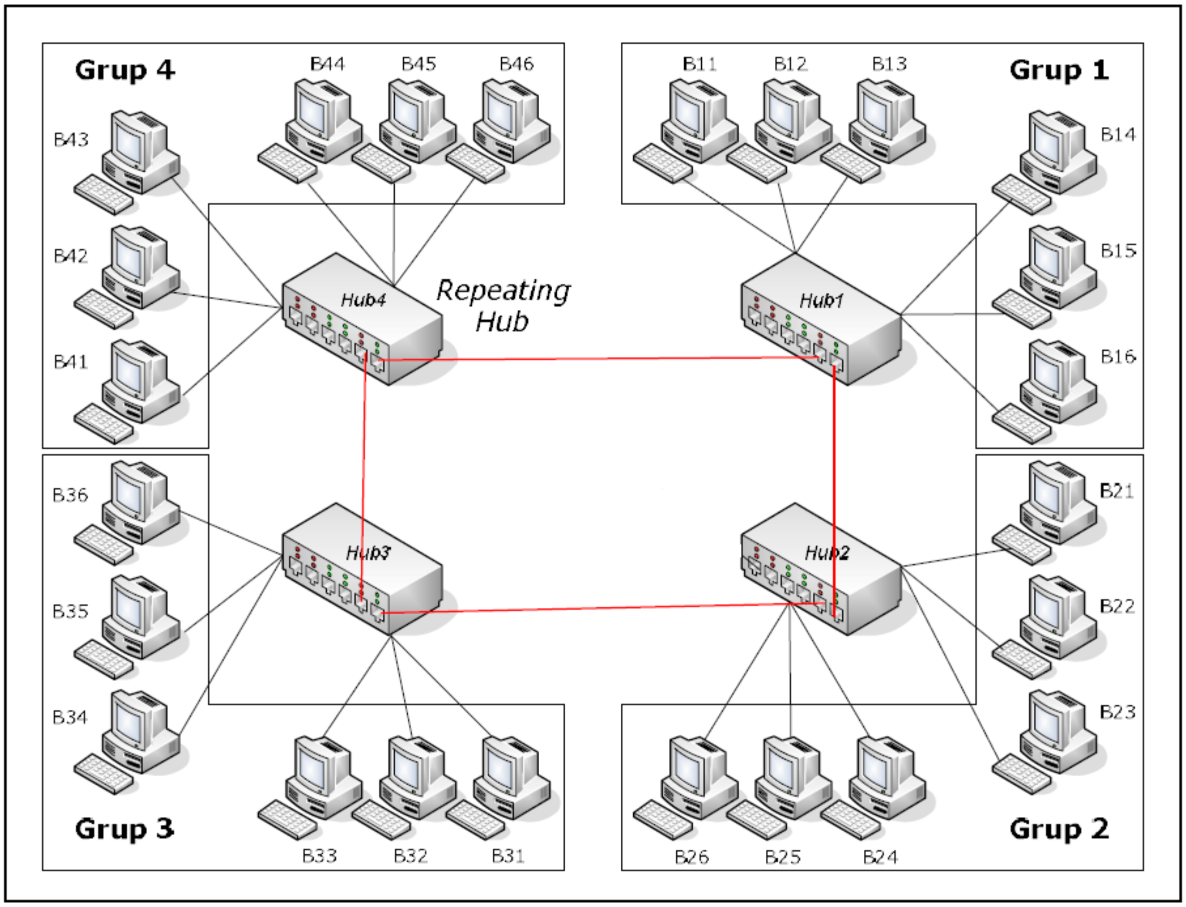
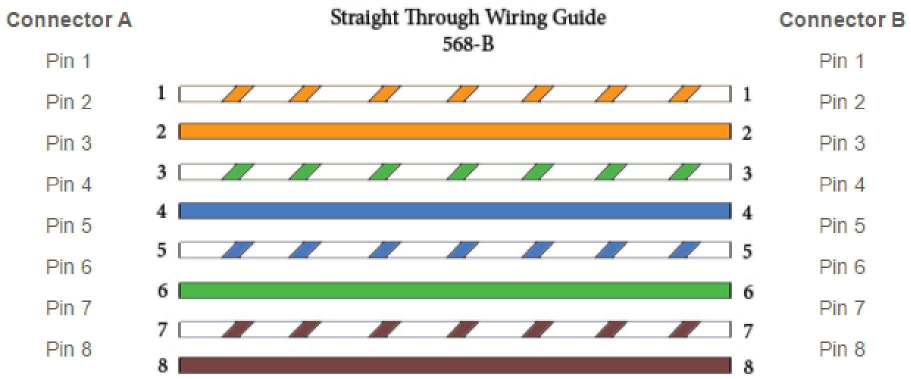


Figure 1 - Network using HUBs

2. You should prepare straight-through and cross-over network cables, to be able to connect related devices. Cable preparation tools will be explained by your lab instructor.

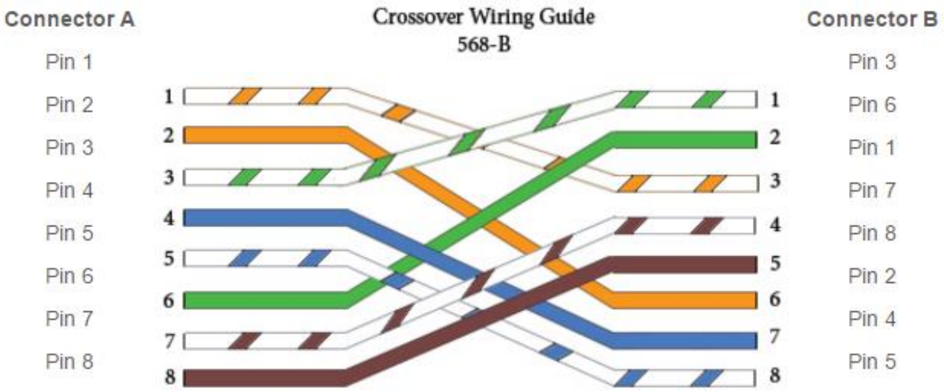
Straight-Through Wired Cables

Straight-Through refers to cables that have the pin assignments on each end of the cable. In other words Pin 1 connector A goes to Pin 1 on connector B, Pin 2 to Pin 2 ect. Straight-Through wired cables are most commonly used to connect a host to client. When we talk about cat5e patch cables, the Straight-Through wired cat5e patch cable is used to connect computers, printers and other network client devices to the router switch or hub (the host device in this instance).



Crossover Wired Cables

Crossover wired cables (commonly called crossover cables) are very much like Straight-Through cables with the exception that TX and RX lines are crossed (they are at opposite positions on either end of the cable. Using the 568-B standard as an example below you will see that Pin 1 on connector A goes to Pin 3 on connector B. Pin 2 on connector A goes to Pin 6 on connector B etc. Crossover cables are most commonly used to connect two hosts directly. Examples would be connecting a computer directly to another computer, connecting a switch directly to another switch, or connecting a router to a router. *Note: While in the past when connecting two host devices directly a crossover cable was required. Now days most devices have auto sensing technology that detects the cable and device and crosses pairs when needed.*



3. Assign IP addresses to your computers *eth0* adapter as described in the Table-1

Group name	IP address	Subnet mask
Grup1	10.100.10.1 - 10.100.10.6	255.255.0.0
Grup2	10.100.20.1 - 10.100.20.6	255.255.0.0
Grup3	10.100.30.1 - 10.100.30.6	255.255.0.0
Grup4	10.100.40.1 - 10.100.40.6	255.255.0.0

4. Make sure that your Ethernet interface of your computer is active, and your cable is plugged to your group's HUB device. Green light on your HUB's port indicates that there is a physical connnection established between the hub and the end device.
5. Check that you can ping other computers in the local network (using *ping <IP address>* command).
6. Make sure that there is a connection between HUB devices
7. Run Wireshark program and start capturing your active interface (*eth0*). Before capturing, all computers have to stop pinging (Ctrl+C)
8. Only one computer (decided by lab instructor, lets call A) will start to ping a destination computer (lets say B), and all others will observe the captured packets on Wireshark.
9. After observing ping packets, computer A will start an FTP connection to the FTP server on B. In the meantime all other computers will observe the connection and login process. (Lab instructor will determine the computer B and configure FTP user settings)
10. You should mention **all steps you have done** (with printscreens) and **discuss about your observations**.

Using Switch

11. Change the HUB devices with Switches and create same topology described in Figure-1
12. Make sure that your Ethernet interface of your computer is active, and your cable is plugged to your group's Switch device. Green light on your Switch's port indicates that there is a physical connnection established between the hub and the end device.
13. Check that you can ping other computers in the local network (using *ping <IP address>* command).
14. Make sure that there is a connection between Switch devices
15. Run Wireshark program and start capturing your active interface (*eth0*). Before capturing, all computers have to stop pinging (Ctrl+C)
16. Only one computer (decided by lab instructor, lets call A) will start to ping a destination computer (lets say B), and all others will observe the captured packets on Wireshark.
17. After observing ping packets, computer A will start an FTP connection to the FTP server on B. In the meantime all other computers will observe the connection and login process. (Lab instructor will determine the computer B and configure FTP user settings)
18. You should mention **all steps you have done** (with printscreens) and **discuss about your observations**.
19. Discuss the differences between HUB and Switch as you see in the experiments. Reach a conclusion about their working logic.
20. Describe the protocol other than ICMP that you saw in this scenario which was mentioned in introduction part. And explain why you observe it and associate with the frame sending process in Local Area Network.

REFERENCES

- https://en.wikipedia.org/wiki/Local_area_network
- Computer Networks: A top-down approach, Kurose and Ross, 6th Edition, Addison-Wesley
- https://www.computercablestore.com/straight-through-crossover-and-rollover-wiring