COE768: Lab 1
Network Layer Architecture and Encapsulation

1. Virtual Machine (VM)

This lab will be conducted under the VM (virtual machine) environment. To start the VM, you first log into the linux workstation with your own login ID and password. Once you are in, you can start the VM by choosing "Comlab" from the Applications pull down. There are two pull downs in "Comlab". They are:

1. Comlab Virtual Linux
2. Clean Virtual Disk

The functions of these pull downs are explained as follows:

Comlab Virtual Linux - This item is to startup the VM. The VM will go through the standard linux booting process. Once it is up, you can use administrator's commands as a superuser with login ID "**root**" (password: **letrootin**).

Clean Virtual Disk – Selection of this pull down will **remove any modifications** that you have done to the VM when you were working on it. You usually select this pull down if your kernel module continuously crashes the VM.

Note that any changes on the VM will remain within the VM until either you log out of the linux workstation or click on this function. Also note that once the VM is cleaned all changes are lost so ensure you store all the changes in your home directory before cleaning or logging out from the linux workstation.

2. Lab procedure

a. Start the VM

b. Each VM has three Ethernet interfaces. Each Ethernet interface is connected to one of the ports at the patch panel. The interface and the associate port have the same number assigned to them (The port number is on the top of the lab computer). To allow VMs communicate with each other, connect one of the Ethernet interfaces (use Eth0) to the Hub by connecting the associate port at the patch panel to one of the ports of the Hub.

c. After connecting the VM to the Hub, you need to assign a unique IP address to the interface. Start up a text terminal at the VM and type in the following two command:

su – (type in the superuser password: letrootin).

ifconfig eth0 10.1.1.*n* netmask 255.255.255.0

The value of *n* is the interface number of the workstation where your VM is running. To verify the connection setup, type in the following command at a VM:

ping 10.1.1.*m*

The command causes the VM to send an "echo-request" IP packet to the VM of another computer whose IP address is 10.1.1.*m*, where *m* is the workstation number other than *n*. If steps (b) and (c) above are done properly, the original VM will receive a response from the "10.1.1.*m*" VM.

d. In order to study the concepts of layer architecture, network traffic is first generated and then captured and analyzed. In this lab, you will use ping command to generate the network traffic.

Start Wireshark in VM.

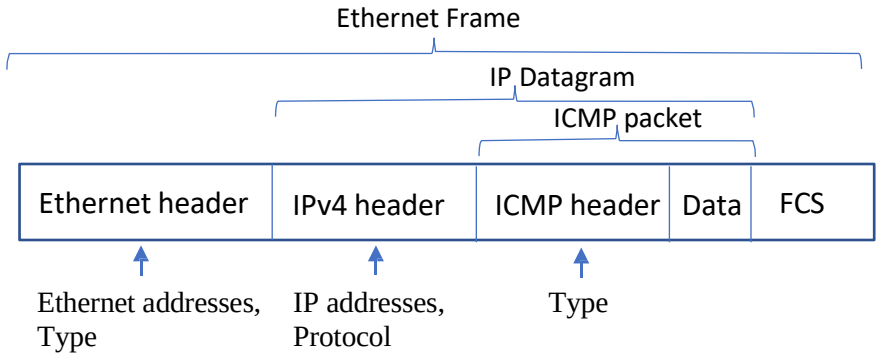
At the Wireshark window, select Eth0 as the interface you want to capture the traffic. Then click on <Capture> and select <start>.

In this section, you will analyze the traffic generated by Ping. From your VM. type

ping 10.1.1.*m*

The command causes your VM to send an "echo-request" IP packet to the VM of another computer whose IP address is 10.1.1.*m*,
This will generate ping traffic between the two machines. Let the ping run for a few seconds, then kill it (<CTRL> + <D>) and stop the capturing in Wireshark.

Now you can analyze the captured packets in Wireshark. Ping packets are generated by the ICMP protocol, which resides above IP protocol. The IP protocol, in turn, resides above the Ethernet Protocol. Consequently, a ping packets has 3 headers as shown in the following figure.



Ethernet header contains source and destination Ethernet Addresses that identify the source and destination machine at the Ethernet layer (layer-2 addresses). The header also contains a "type" field that indicates the type of packet encapsulated by the Ethernet packet (Ethernet packet is usually referred as Ethernet frame), in our case, the IP packet (called IP datagram). IP header carries the IP addresses of the source and destination (layer-3 addresses). The IP header contains many fields. The one you are interested in this lab is the "protocol" field, which indicates the type of packet encapsulated by IP datagram, in our case, the ICMP packet. The ICMP packet also contains many fields. The one you want to focus on is the "Type" field, which indicates if the ICMP packet is the ping request, sent by your VM to the VM of another machine or ping response packet, sent back to your VM.

You can find the various head formats from your textbook or on-line. Based on the content of the captured packets in Wireshark, identify the values of the following fields:

- Source and Destination Ethernet Addresses;
- Ethernet Type field;
- Source and Destination IP addresses;
- IP protocol field;
- ICMP type fields for ping request and response.

e. As another example of capture and network analysis, you will use Echo network application to generate the network traffic. Essentially, you setup an Echo server in one VM and an Echo client in the other VM. After that, you can initiate a communication between the server and client.

The source programs of echo server and echo client (echo_server.c and echo_client.c) can be directly downloaded from D2L. You may also copy them from the following directory in the department file system with the command:

/home/courses/coe768/public_html/socket_progs/Echo

Copy the files plus the Makefile to the VM folder

/home/bob/Desktop

Compile the source program echo_server.c and echo_client.c to generate echo_server and echo_client executable files, respectively, in the VM machines:

make echo_server

make echo_client

Setup an “Echo” server (echo_server) in the VM using the command:

./echo_server port_number

The port_number can be any value between 2^{10} and 2^{16} (e.g. 15000).

Start the “wireshark” application at the VM.

The wireshark application is a process that captures Ethernet frames “seen” by the Ethernet interface. It is a valuable tool for the network traffic analysis.

At the wireshark window, select Eth0 at the interface you want to capture the traffic. After that click **Capture** and select **start**.

In another VM, start an Echo client in the VM using the command:

./echo_client server_IP_address port_number

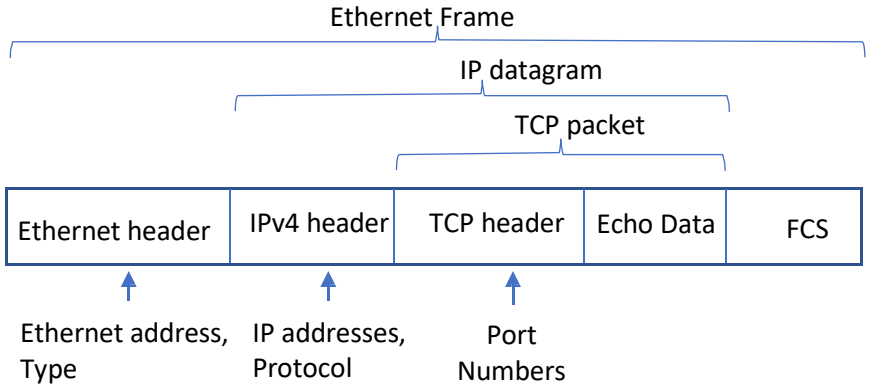
The server_IP_address is the IP address of the VM where the server process is running; the port_number value is the same as the port_number value used by the server (e.g. 15000). The command invokes the client to establish a TCP connection with the server.

At the client side, type a message and hit the CR. The message will be sent to the server through the TCP connection. The server then echoes the message back to the client. Consequently, the message you typed will appear at the terminal twice. (Note: the server just echoes the message back to the client; it will not show the message at its terminal.)

Type control-D to end the echo service

Stop the wireshark capture process.

Echo service operates on top of TCP, consequently, the Echo packet has three headers as shown:



The TCP header has many fields. We will concentrate on the source and destination port numbers. Port number is part of the address of the network application process (the other part is the IP address). Based on this understand, you should be able to deduce the destination port number in the packet sent from the client to the server.

Study the packets captured by the Wireshark and find the values of the following field

- The IP protocol field (which should be different from that in part II, why?);
- The source and destination port numbers for the packet sent by client to the server;
- The source and destination port numbers for the packet sent by the server to the client.

3. Lab questions of part (e)

a. At the application level, only two packets are exchanged between the client and the server. However, you will find that 8 or more TCP packets are actually generated. TCP is a connection-oriented protocol, that is, a logical connection must be established between two communication entities before data transfer (somewhat similar to the telephone connection). With this understanding, find out the functions of those extra TCP packets.

b. From the wireshark window, locate the frame that contains the message sent from the client to the server. The frame is displayed in hexadecimal. Record the content of the frame and indicate the locations of the following parameters:

- The Ethernet addresses of the client and server VMs;
- The Ethernet type;
- The IP addresses of the client and the server;
- The protocol number;
- The source and destination port numbers.

c. Identify the layers the packet had to go through before it was sent out.