COE768: Lab 2
Network Layer Architecture and Encapsulation

In this lab, we will study the concepts of network layer architecture by inspecting the contents of packets generated by various network applications.

I. Installation of Wireshark

1. Wireshark is the most common network analyzer used nowadays. It is capable to capture packets sent or received on interfaces. You will install Wireshark on one of the VMs (VM1).

2. To install Wireshark, type the following sequence of commands in the terminal:

```
$sudo
$sudo chmod +x /usr/bin/dumpcap
$reboot
```

   After the reboot, Wireshark is ready to be used.

II. Capture and Analyze Ping packets

3. Start Wireshark in VM1.

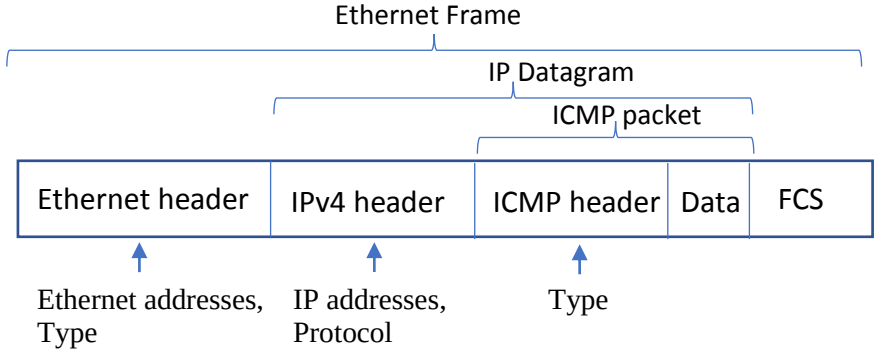
4. At the Wireshark window, select enp0s3 as the interface you want to capture the traffic. Then click on <Capture> and select <start>.

5. In this section, you will analyze the traffic generated by Ping. From VM1. Type

```
$ping IP_address_VM2
```

   This will generate ping traffic between VM1 and VM2. Let the ping run for a few seconds, then kill it (<CTRL> + <C>) and stop the capturing in Wireshark.

6. 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). You already found out the IP addresses of VMs in Lab 1. 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 VM1 to VM2 or ping response packet, sent by VM2 to VM1.

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.

III. Capture and analyze traffic generated at the application layer

In this part of the lab, you will setup an Echo server on VM1. After that, you enable an Echo client on VM2. The Echo client will then initiate a TCP connection with the server. Once the TCP connection was established, the user of the Echo client (that means "you") can send messages to the server and the server echoes back the same messages to the client. The Echo service is implemented at the application layer, thus the programs of the service can be compiled and executed at the user space.

7. The source programs of echo server and echo client in C (echo_server.c and echo_client.c) can be found in the D2L course site under

   Content/Lab Assignments/Socket Programs.

8. To compile the source program echo_server.c and echo_client.c, you need a C compiler. You can install a C compiler, called gcc, on the VM using the following command:

   $sudo apt install gcc

9. To generate echo_server and echo_client executable files on the VM machines, type

$gcc -o echo_server echo_server.c -lnsl

$gcc -o echo_client echo_client.c -lnsl

10. Start the Wireshark capture on VM1.

11. Start up an "Echo" server (echo_server) on VM1 using the command:

    $./echo_server port_number

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

12. On VM2, start an Echo client:

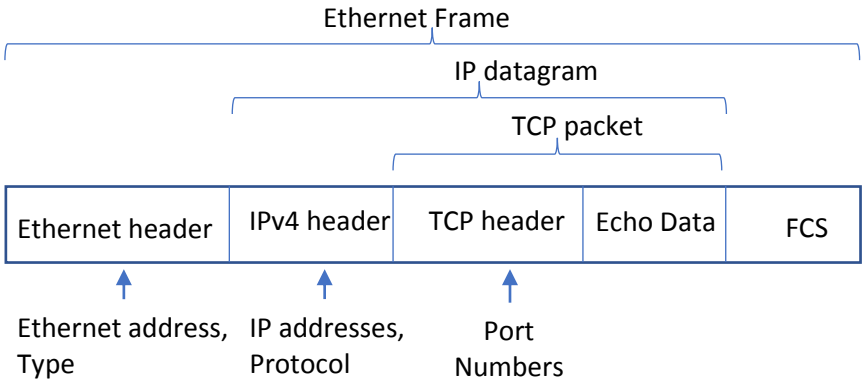
    $./echo_client server_IP_address port_number

    The server_IP_address is the IP address of VM1; the port_number is the same port_number used for starting up the server.

13. At the client side, type a message and hit the <CR>. The message will be sent to the server. The server then echoes the message back to the client. Consequently, the message you typed will appear in the client terminal twice. (Note: the server just echoes the message back to the client; it will not show the message in its own terminal.). Type control-D to end the echo service.

14. Stop the Wireshark capture process.

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



16. The TCP header has many fields. We will concentrate on the source and destination port numbers. As we mentioned in Lab 1, 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.

17. 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.

What you needed to demonstrate to your TA

1. Show your Wireshark captured data in part II. Identify the locations and values of Ethernet Source and Destination addresses, Ethernet Type field, IP protocol field, ICMP type fields for ping request and response.

2. Show your Wireshark captured data in part III. Identify the locations and values of IP protocol field and the source and destination port numbers. (Note: You will see more than 2 TCP packets captured by the Wireshark. The functions of the extra packets will be covered in the lectures and Lab 3).