## Assignment 2 Task 01

## Intro to Experimenting with Packet Trace and Misc

*You are given a trace file* [*trace0.pcap*](https://www.dropbox.com/s/hpc8b6n15cxtl9j/trace0.pcap?dl=0)*. The trace has been taken by running* **tcpdump** *on a host with IP address 192.168.0.106. Open this trace in wireshark and first answer the following questions.*

1. The trace has all sorts of packets but is overwhelmingly made of a trace of a connection related to one particular activity or command. What is that?
2. What is the IP address, port number of the two ends of this connection? How many packets do you see of this connection (hint: filter properly and see bottom right of wireshark window). Let us call this Connection X. The following questions are about this Connection X.
   1. At which IP address and port number is the “server” running? What is the application layer protocol in connection X?
   2. At which IP address and port number is the client running? What possible terminal command at the client could generate such a trace? Write the whole command. **(0.5+0.5+1)**
   3. Can you see the client hostname in the trace? If yes what is it? The server hostname? If yes, what is it? **(0.5+0.5)**
3. Look at packets 5 and 6. What question is Packet 5 asking?**(0.5)** What answer did it get in Packet 6? **(0.5)**
4. Look at packets 7,8,9,10. Describe what they are doing **(0.5)** and which is the client,**(0.5)** which is the server**(0.5)**, and what is this kind of server called**(0.5)**.
5. For packet number 14 (which is of Connection X), answer the following questions.
   1. The headers of which layers can you see clearly? Write 3 header field names and their values for each of the layers that you can see in this packet. E.g. For each layer, you can answer in the following format:

**1/5 per header field x 15 = 3**

Layer name:

Field Name1 = value1

Field Name2 = value2

Field Name3 = value3

* 1. The packet number 14 is going from which host (IP addr) to which host (IP addr)? **(0.5+0.5)**
  2. The packet number 14 is going from which network interface (MAC address) to which network interface (MAC addr)? **(0.5+0.5)**

1. Using information in packet numbers 5,6 and 14 (and answers to above questions) can you infer whether there is a direct link between the two end-hosts of Connection X. Justify your answer. Do not use any reasoning other than what is evident in these packets. **(2 for the answer given below)**
   1. If there is no direct link, what is the IP address and MAC address of the next hop from the client? **(1)**

*The following questions need you to look at timestamps and do some calculations. You are advised to use a spreadsheet (e.g. Libreoffice calc) for all these otherwise the calculations will take time, and will have to be repeated. Also, it might be a good idea to filter the trace on the server IP address.*

1. Find packet pairs in the setup and teardown of Connection X that represent roundtrips from the client to server to client. Find 3 such packet pairs. Write the packet number pairs.

**(0.5+0.5+0.5)**

* 1. Calculate the RTT (roundtrip) you are getting from each of these 3 (write which packet pair gives which time). Write the min, max and average.

**(0.5+0.5+0.5)**

Now look at packets from the server to client in packet number range 18 to 32.

1. How many packets came **to** the client **from** the server in this range?

**(0.5)**

1. What is the interarrival time of packets coming **to** the client **from** the server? Write all the times, the average, min and max.

**(2)**

Now observe packets 32 onwards

1. Packet numbers 32 and 34 are sequential arrivals to the client. What is their interarrival time?
   1. What inference can you make from this and the interarrival time of packets 18 to 32? Specifically, do you think the server is using stop and wait protocol to send its data? (You may assume that RTT from server-client-server will have a similar value to client-server-client RTT). If not, what might be the window size (in units of packets). Justify your inference.
   2. How much data in bytes do you think the server sent without waiting for an acknowledgement? (Hint: for this you can use the TCP sequence number seen in the packets. TCP numbers its packet sequence numbers in units of bytes. E.g. The first data packet has sequence number 1. Now, if Packet P\_k has Sequence number S\_k (bytes) and size L bytes, the next packet P(k+1) will have Sequence number S\_k +L bytes. **(1.5 with all calculations shown)**
2. Find series of packets with the same arrival pattern as packets 18 to 34. Find three more such series. Using TCP sequence numbers, figure out how much data the sender is sending (essentially its window size in bytes) without waiting for an ack. Fill in the following table:

Packet series start packet number| series end packet number | Window size in bytes

**(1.5+1.5+1.5) (0.5 per column entry)**

What inference can you draw from this table about the window size (in bytes) the sender is using?

1. What is the “raw” throughput achieved in this connection? (Raw throughput can be calculated e.g. by bytes sent in packets 18-32 and time in which they were received). **(1)**
2. What is the latency from the connection setup request by the client (“SYN”), to getting the first packet of the file? **(1)**
3. What is the latency from the connection setup request by the client (“SYN”), to getting the last packet of the file? **(1)**
4. What is the effective throughput of the whole connection? **(1)**

[Data received / (Receiving time of Last packet of file - connection setup request time)]

1. What do you think is dominating the latency? (slow data rate of some bottleneck link? Or large Round Trip Time) **(3 with marking scheme as below)**