CS 39006: Networks Lab

Assignment 1: Use Wireshark for Analyzing Network Packet Traces

Date: 11th January, 2018

Objective:

The objective of this assignment is to understand the Wireshark tool and how you can analyse network packet traces. You have to use Wireshark for answering the questions.

Submission Instructions:

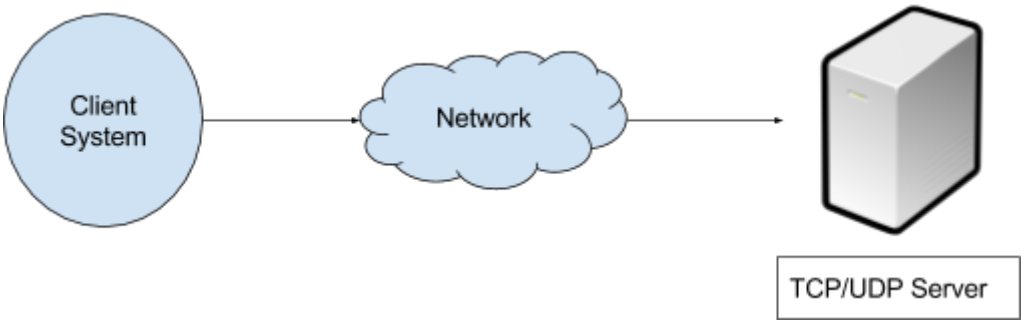
You need to prepare a report that will contain the followings.

1. **Steps** followed in executing the experiments.
2. **Observations** from the experiments.
3. Intuitive **justification** behind the observations

You need to submit the report in a single compressed (tar.gz) file. Rename the compressed file as `Assignment_1_Roll1_Roll2.tar.gz`, where `Roll1` and `Roll2` are the roll numbers of the two members in the group. Submit the compressed file through Moodle by the submission deadline. The submission deadline is: **January 18, 2018 02:00 PM.** Please note that this is a strict deadline and no extension will be granted.

Please note that your submission will be awarded zero marks without further consideration, if it is find to be copied. In such cases, all the submissions will be treated equally, without any discrimination to figure out who has copied from whom.

Assignment Statement:



The client system (representing your system) is connected through the network to a TCP and UDP server. The TCP server for this system is the standard Python *HTTP Server* (*IP: 10.5.20.128, Default Port:8000, Files: 5 Images - pic1.jpg to pic5.jpg*) and for UDP an *iperf* server (*IP: 10.5.20.128, Default Port: 5001*) is running in the host machine.

The necessary commands are as follows:

1) UDP Client (with 28 Kbps) for UDP performance measurement using *iperf*:
   `iperf -c 10.5.20.128 -u -b 28000`
   This command will send UDP packets to the iperf server running at 10.5.20.128. You can specify the time for which you want to send UDP packets. Please check iperf documentation.
2) TCP Client: **`wget --no-proxy`** http://10.5.20.128:8000/pic1.jpg
   This command uses the HTTP protocol to access the image fime pic1.jpg from the web server running at 10.5.20.128. You can change the URI to access pic1.jpg to pic5.jpg.

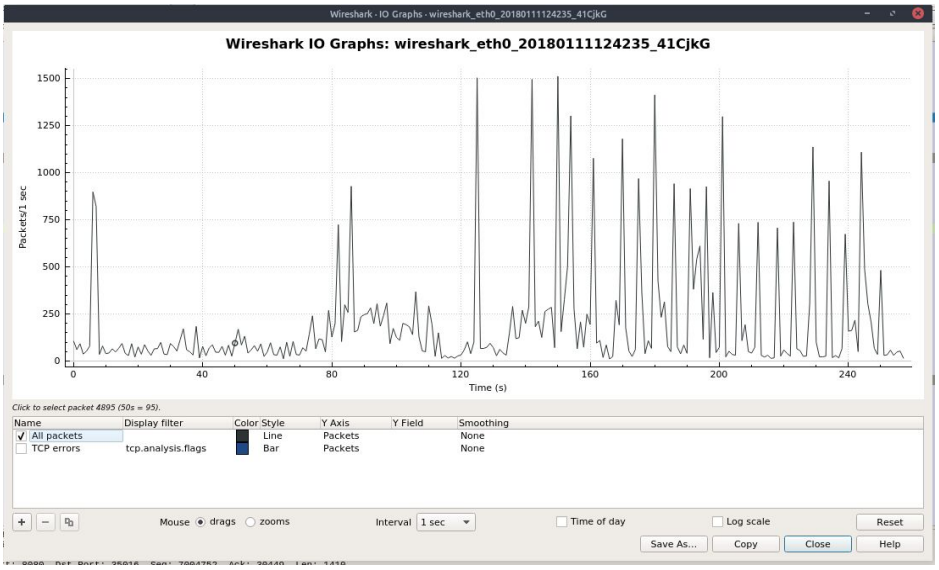
Capture the packet traces using *Wireshark* in your client system and answer the following.

(1) List the different protocols that you observe in the packet trace, at application, transport and network layer for each of the UDP and TCP test cases.

(2) Analyse the packet trace using Wireshark and compute the followings,
   (a) How many TCP packets are transferred for each cases while accessing the files pic1.jpg to pic5.jpg? Are all the packets of same size? What are the different packet size you observe for each of the file access?

   (b) For the test case with UDP, are all the UDP packets of same size? If not, what are the different UDP packet sizes you observe?

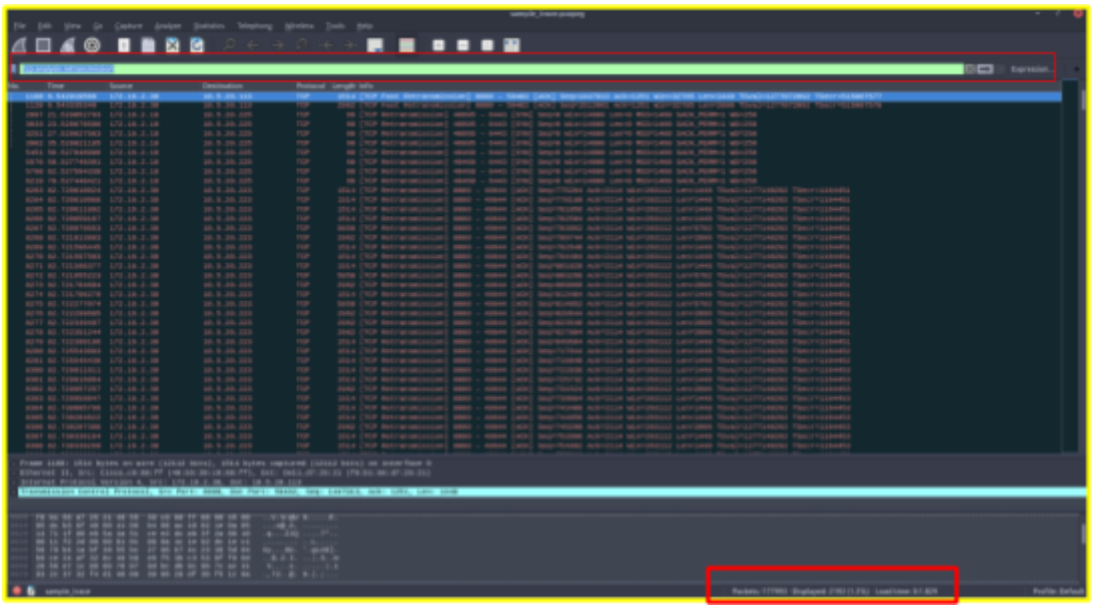
   (c) Observe the TCP and the UDP throughput using Wireshark (Menu->Statistics->IO Graphs), as shown in the following figure.



(d) Compute the UDP throughput (amount of UDP data received per second) for following cases of UDP traffic generation rates (bandwidth)

(i) 64 Kbps
(ii) 128 Kbps
(iii) 256 Kbps
(iv) 512 Kbps
(v) 1024 Kbps
(vi) 2048 Kbps

(3) Analyze the number of TCP packets retransmitted (Use: *tcp.analysis.retransmission*) from Wireshark, as shown in figure below.



(4) Plot the following

1. UDP throughput with respect to the UDP bandwidth
2. Number of UDP packets transmitted with respect to UDP bandwidth

What are your observations from these plots?

NB: You may use the open source utility gnuplot (http://www.gnuplot.info/) to plot the graph.