Report-Assignment 1

Aditya Sawant 17CS10060
Soumen Dutta 17CS10057

1. UDP:
   Capture Filter used: src host 10.145.147.239 and dst host 10.5.18.163
   Wireshark had been configured to capture packets with my IP as source and destination as 10.5.18.163 .
   The experiment was performed with Ethernet connection.
   We used the following terminal command to send udp packets to the server.
   iperf3 -c 10.5.18.163 -u -b 28000

   Observation:

   Physical Layer:
   Frame 27: 834 bytes on wire (6672 bits), 834 bytes captured (6672 bits) on interface 0

   Data-link Layer:
   Ethernet II, Src: Azurewav_b6:ee:ab (f0:03:8c:b6:ee:ab), Dst: All-HSRP-routers_4c (00:00:0c:07:ac:4c)

   Network Layer:
   Internet Protocol Version 4, Src: 10.145.147.239, Dst: 10.5.18.163

   Transport Layer:
   User Datagram Protocol, Src Port: 42305, Dst Port: 5201

   Application Layer:

   TCP:
   Capture Filter used: host 10.5.18.163 and host 10.145.147.239
   Wireshark had been configured to capture packets with my IP as one of the host and 10.5.18.163 as the other host.
   The experiment was performed with Ethernet connection.
   We used the following terminal command to download a jpeg image from the server (in the form of TCP packets).
   wget --no-proxy http://10.5.18.163:8000/1.jpg

   Observation:

   Physical Layer:
   Frame 630: 1038 bytes on wire (8304 bits), 1038 bytes captured (8304 bits) on interface 0

   Data-link Layer:
   Ethernet II, Src: Cisco_60:22:bf (c8:9c:1d:60:22:bf), Dst: LcfcHefe_6c:7f:47 (54:e1:ad:6c:7f:47)

   Network Layer:
   Internet Protocol Version 4, Src: 10.5.18.163, Dst: 10.109.51.58

Transport Layer:
Transmission Control Protocol, Src Port: 8000, Dst Port: 43370, Seq: 906466, Ack: 149, Len: 972

Application Layer:
Hypertext Transfer Protocol

2)
a)

We are considering two way transfer of packets.
For file1.jpg- 1152 TCP packets and 2 HTTP packets
For file2.jpg- 2177 TCP packets and 2 HTTP packets
For file3.jpg- 7245 TCP packets and 2 HTTP packets
For file4.jpg- 2483 TCP packets and 2 HTTP packets
For file5.jpg- 3016 TCP packets and 2 HTTP packets

All packets are not of same sizes.

For packets from server to client we observed the following different TCP packet sizes for file1.jpg.(We used the display filter: ip.src==10.5.18.163)

66, 74, 83, 1514, 2962, 4410, 5858, 13098

For packets from client to server we observed the following different TCP packet sizes for file1.jpg(We used the display filter: ip.dst==10.5.18.163)

66, 74

For packets from server to client we observed the following different TCP packet sizes for file2.jpg.(We used the display filter: ip.src==10.5.18.163)

66, 74, 83, 1514, 2962, 4410, 5858, 7306, 8754, 30474

For packets from client to server we observed the following different TCP packet sizes for file2.jpg(We used the display filter: ip.dst==10.5.18.163)

66, 74

For packets from server to client we observed the following different TCP packet sizes for file3.jpg (We used the display filter: ip.src==10.5.18.163)

66, 74, 83, 1191, 1514, 2962, 4410, 7306

For packets from client to server we observed the following different TCP packet sizes for file3.jpg

66, 74

For packets from server to client we observed the following different TCP packet sizes for file4.jpg(We used the display filter: ip.src==10.5.18.163)

66, 74, 83, 1514, 2962,3314, 4410, 5858,7306, 8754,10202, 11650

For packets from client to server we observed the following different TCP packet sizes for file4.jpg

66, 74

For packets from server to client we observed the following different TCP packet sizes for file5.jpg(We used the display filter: ip.src==10.5.18.163)

66, 74, 83, 1514, 2962, 4410, 5858,7306, 8754, 10202

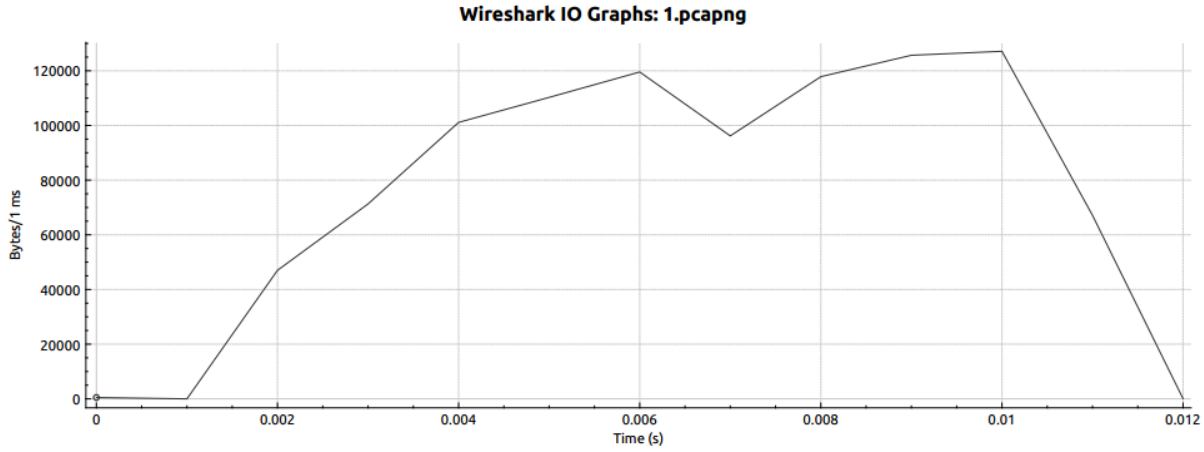
For packets from client to server we observed the following different TCP packet sizes for file5.jpg

66, 74

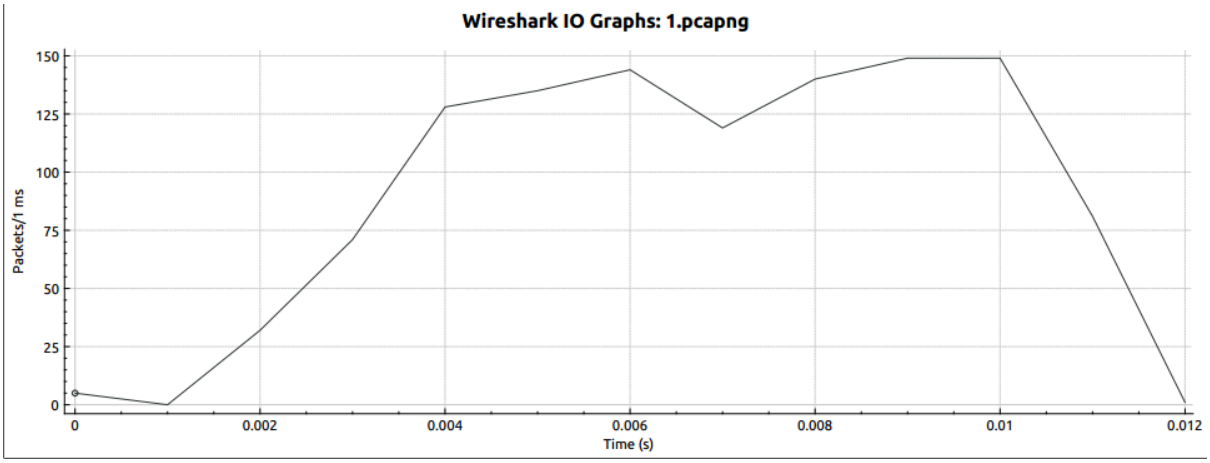
b)

All UDP packets sent from the client to the server are of the same size (834 bytes) except for the first packet which of size 46 bytes. Also one UDP packet of size 60 bytes is sent from the server to the client as response after the 1st packet is received by the server. This is true for all bandwidths used in this experiment.
We used the display filter: udp

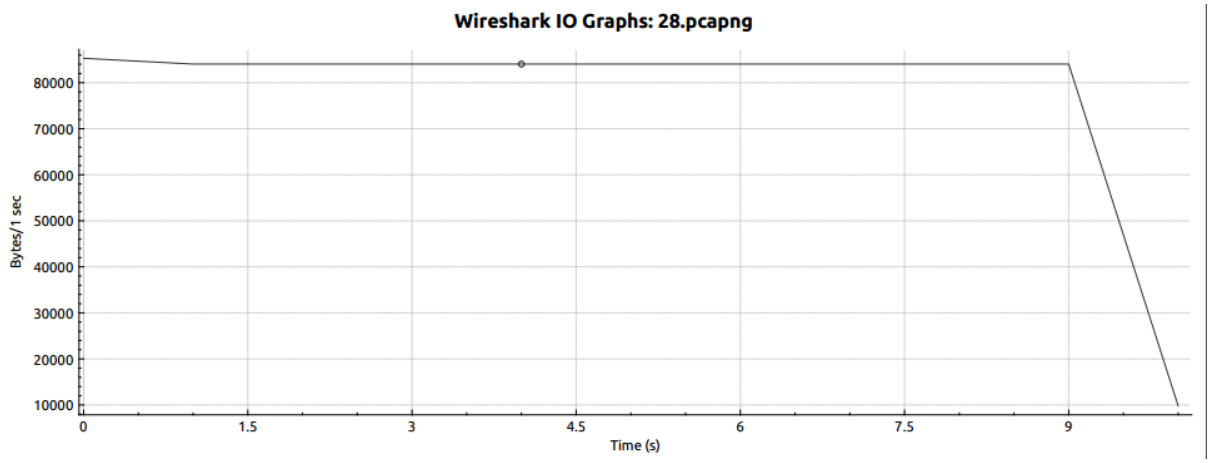
c)



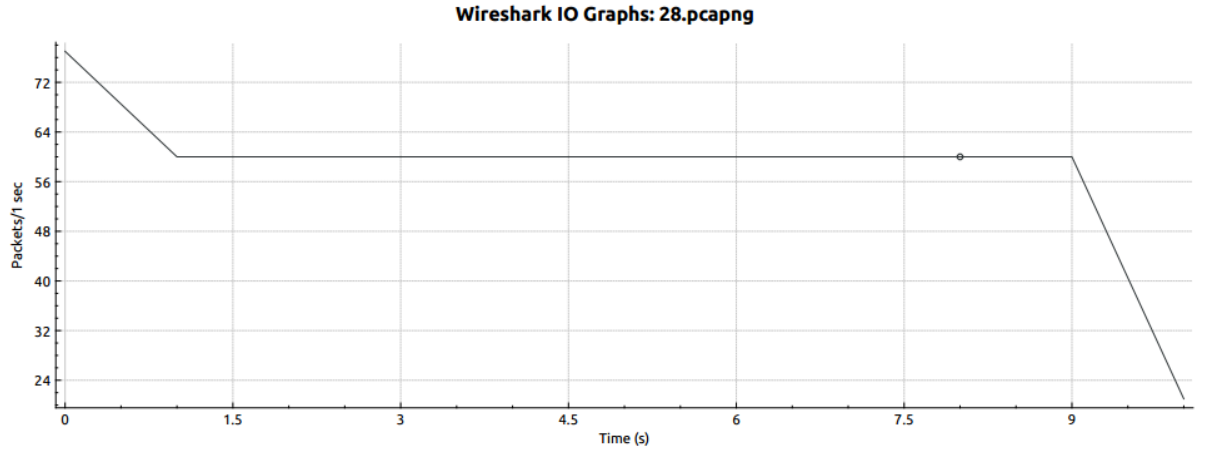
Observing TCP throughput in Bytes/1ms



Observing TCP throughput in Packets/1 ms



Observing UDP throughput in Bytes/1 ms



Observing UDP throughput in Packets/ 1ms

d)

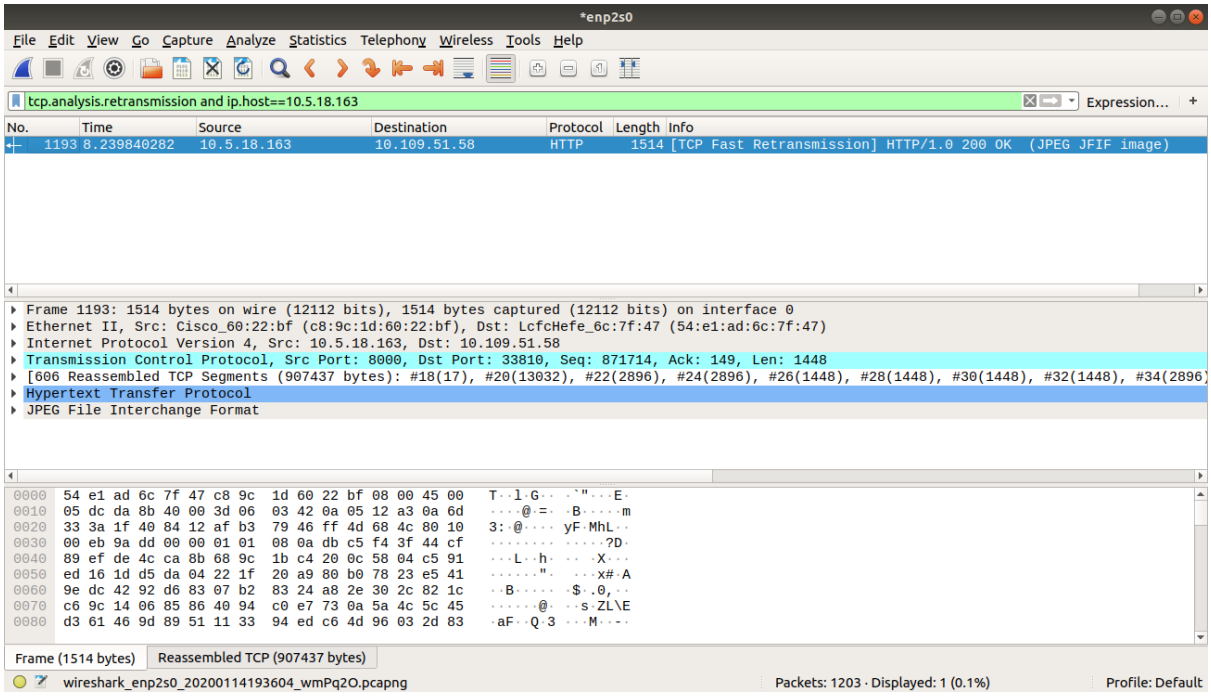
For this part of the question, we are considering all UDP packets sent by the client to the server except the first one as first one is to establish connection. Throughput is calculated as: ((packet size)*(number of packets))/(time for which packets are sent). Here packet size = 834 bytes and time = 10s

For Bandwidth = 28 Kbps, UDP Throughput = 8.4234 kB/s (101 Packets)
For Bandwidth = 64 Kbps, UDP Throughput = 8.4234 kB/s (101 Packets)
For Bandwidth = 128 Kbps, UDP Throughput = 8.4234 kB/s (101 Packets)
For Bandwidth = 256 Kbps, UDP Throughput = 8.4234 kB/s (101 Packets)
For Bandwidth = 512 Kbps, UDP Throughput = 8.4234 kB/s (101 Packets)
For Bandwidth = 1024 Kbps, UDP Throughput = 13.0104 kB/s (156 Packets)
For Bandwidth = 2048 Kbps, UDP Throughput = 25.9374 kB/s (311 Packets)

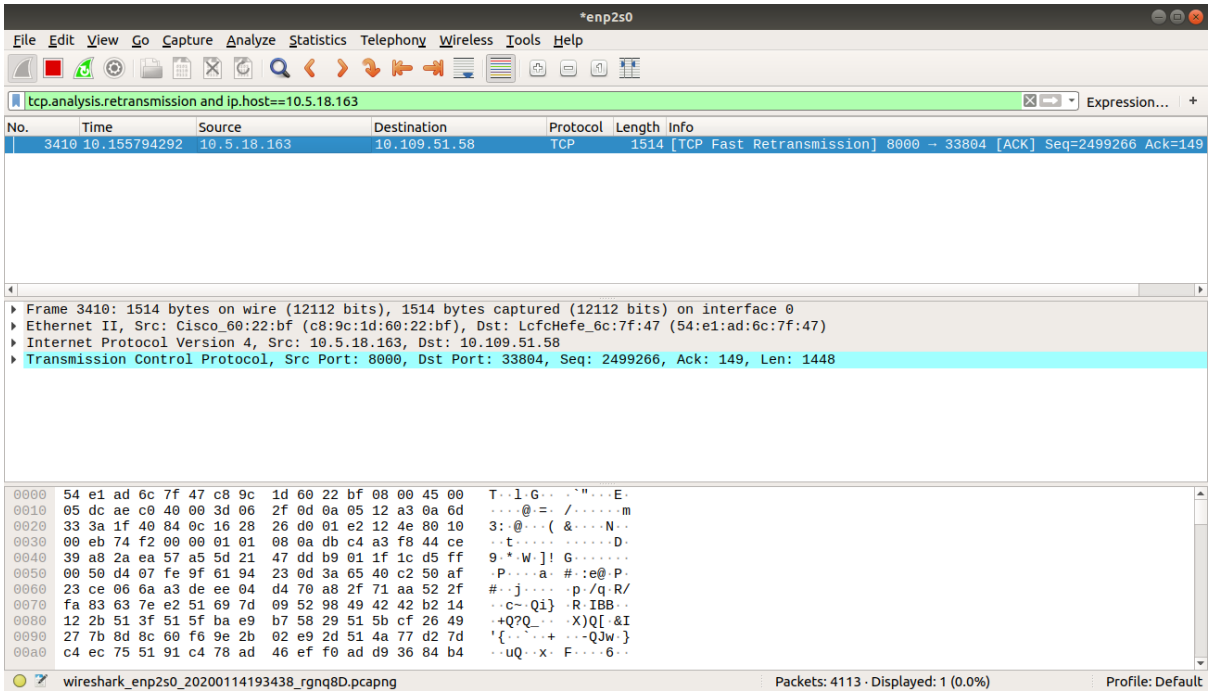
3

Analysing retransmitted packets. Very few packets are retransmitted from the 10.5.18.163 server indicating good connection. The display filter used is tcp.analysis.retransmission and ip.host==10.5.18.163 .

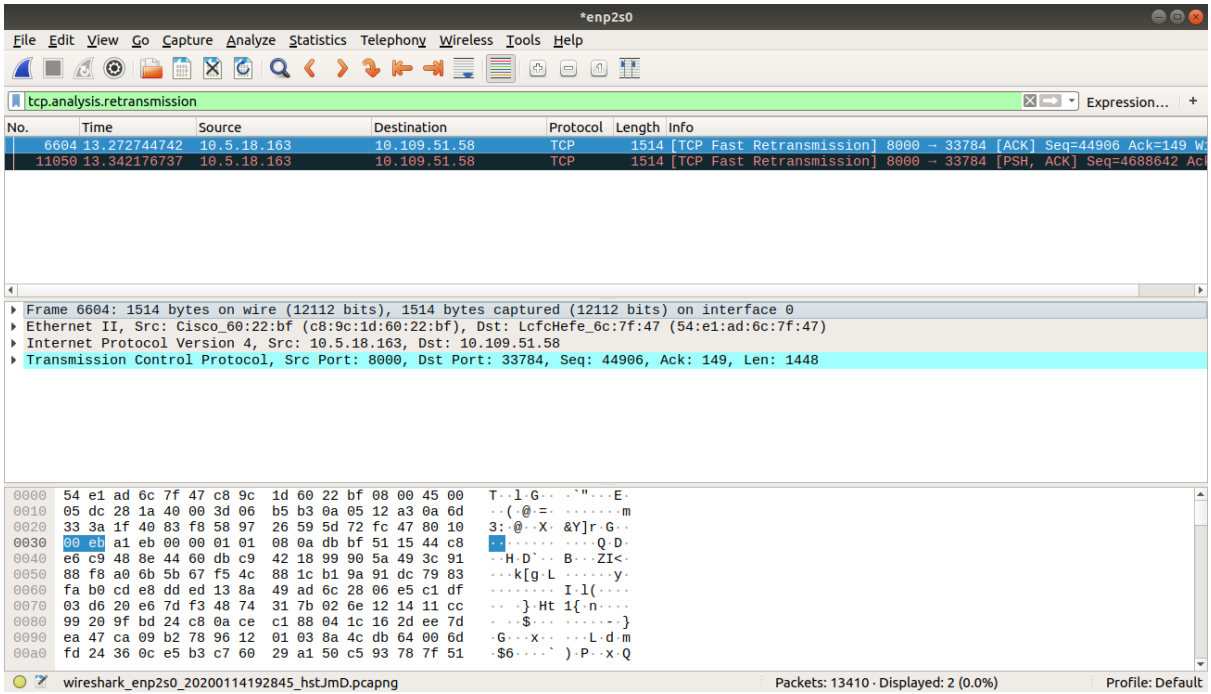
Following image is for 1.jpg



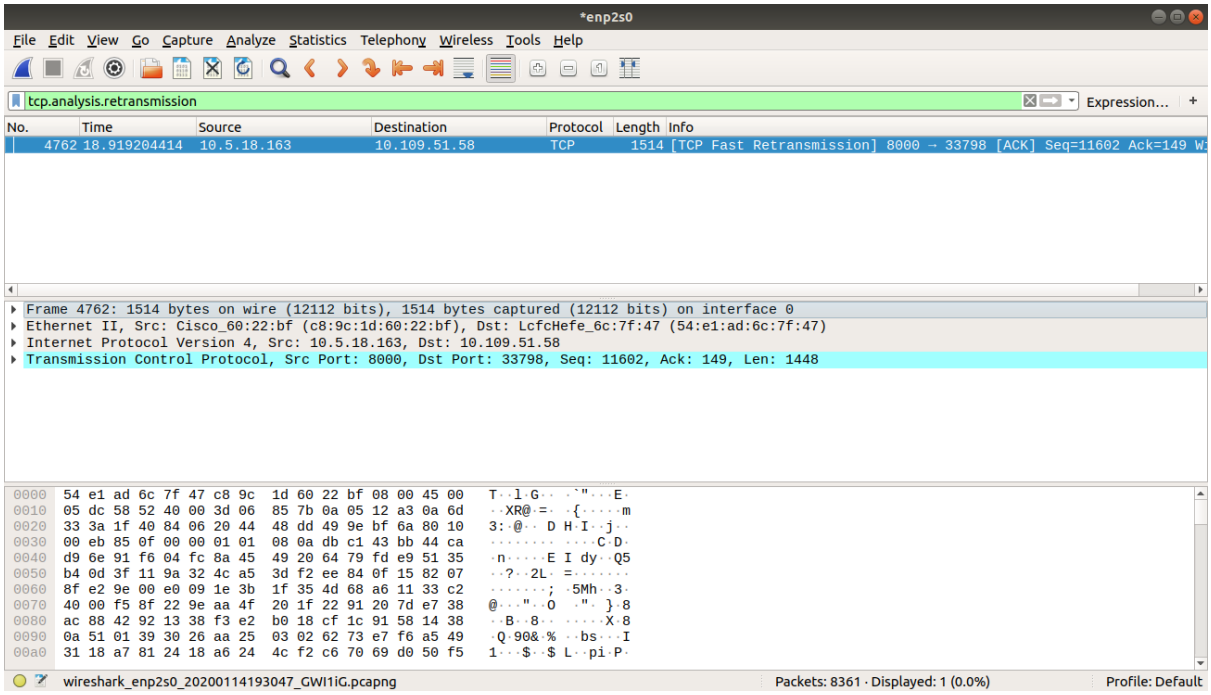
Following image is for 2.jpg



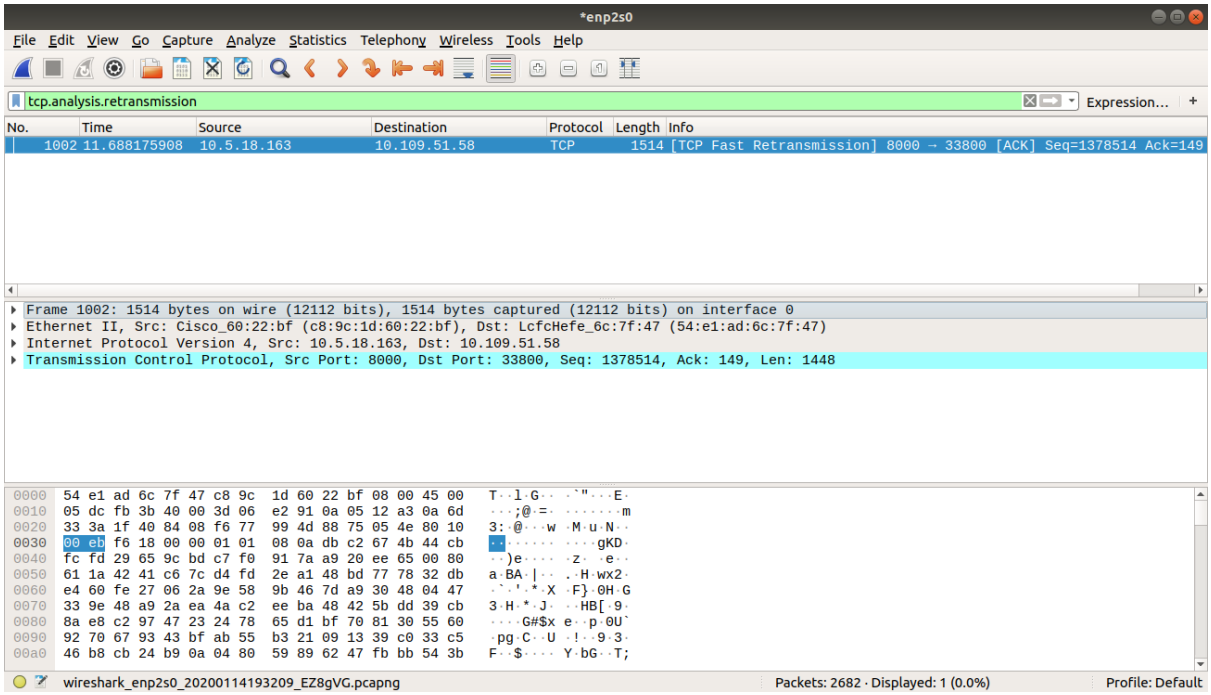
Following image is for 3.jpg



Following image is for 4.jpg

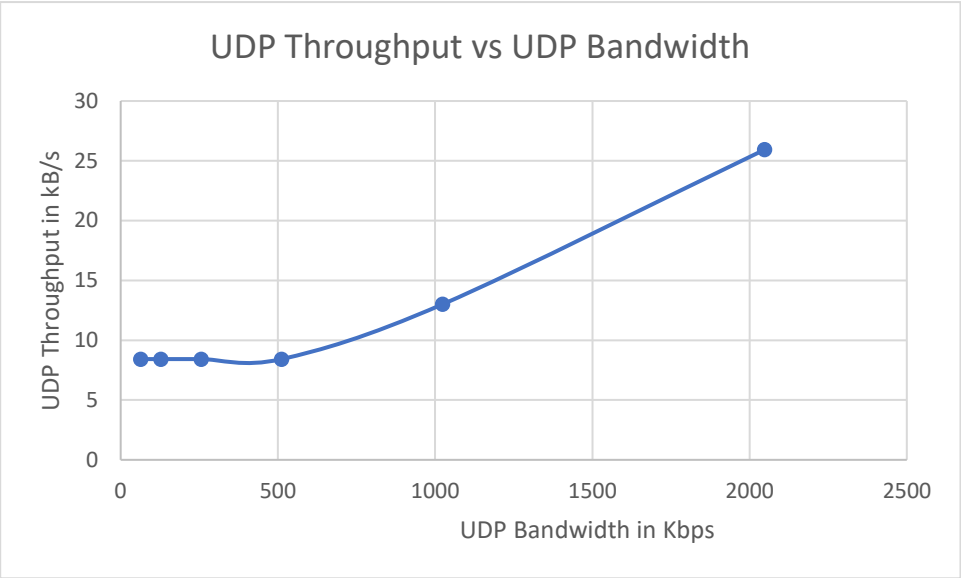


Following image is for 5.jpg



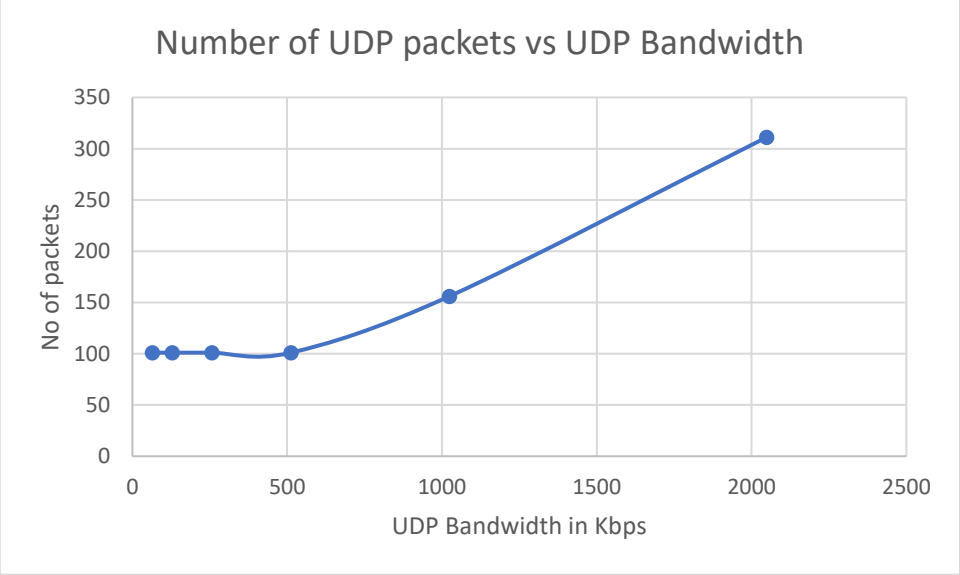
4 a)

Graph depicting UDP Throughput in KB/s vs UDP Bandwidth in Kbps



Initially the UDP throughput remains constant with bandwidth then increases in a straight line.

b) Graph on Number of UDP packets send vs UDP Bandwidth



Initially the no of packets remains constant with bandwidth then increases in a straight line.