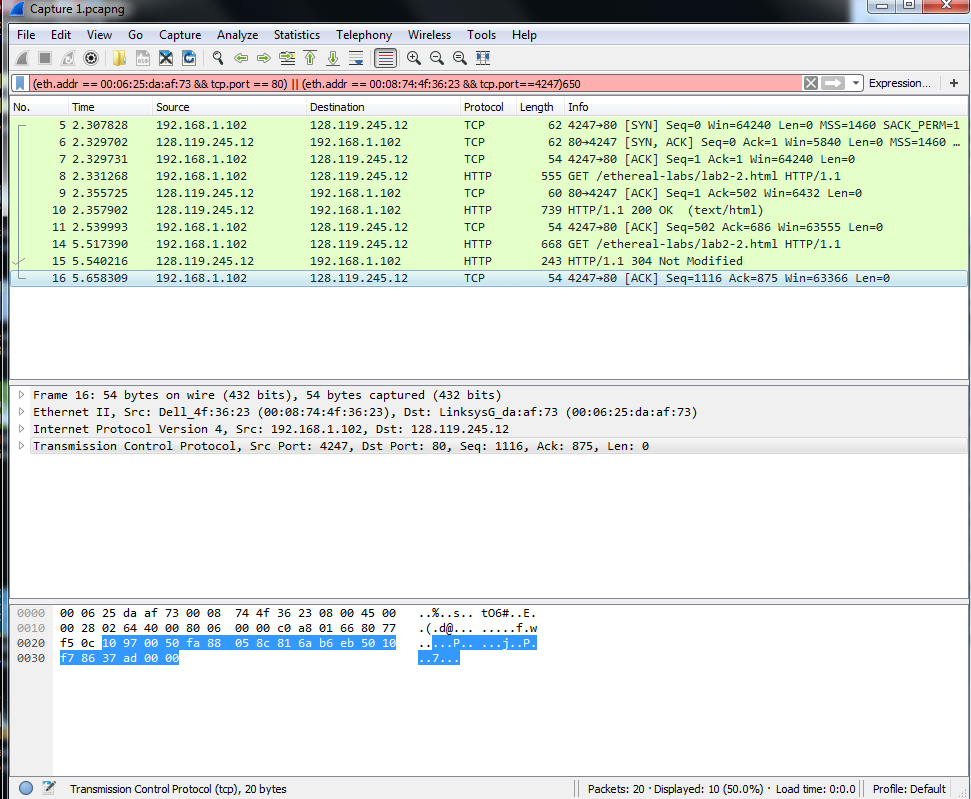
# **Capture 1:**



Q1:

(eth.addr == 00:06:25:da:af:73 && tcp.port == 80) || (eth.addr == 00:08:74:4f:36:23 && tcp.port==4247)

Q2:

Before:

In Packet 8, we are sending a GET request for a file if its not modified since a specific date. In response the server says that the file is not modified, hence it doesn’t send it.

After:

Now after we have replaced the file on the server and the Last Modified Date is not 9 Nov 2018, the same GET request will now fetch the file. Here the Get request is for the file having 250 bytes.

Previously packet 9 content length is 189. Now an additional of 250 bytes will be added which will make it 189+250 = 439. The current sequence number is 686 so in next ACK in Packet 10 the ACK number will be 686+439 = 1125. While the sequence Number will be still 1116 since Client just sent the same request in both cases.

# **Capture 2:**

Q3

(ip.src == 192.168.1.2 && tcp.dstport == 21) || (ip.src == 195.89.6.167 && (tcp.srcport == 20 || tcp.srcport == 21))

Q4:

PORT 192,168,1,2,63,215\r\n

Request command: PORT

Active IP address: 178.128.12.54

Active port: 16979

Q5:

Destination IP = 178.128.12.54

Destination Port = 16979

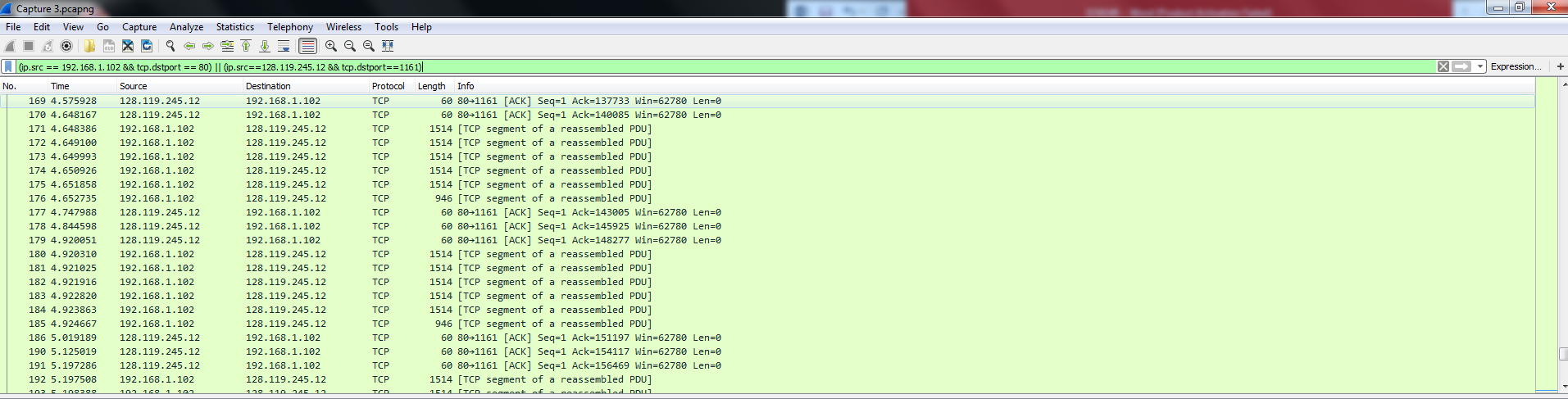
Q6:

eth.addr == 60:67:20:55:7b:ac && udp.dstport == 53

Q7:

DNS Packets are being sent using UDP, hence there’s no ACK or SEQ number

# **CAPTURE 3**



Q8:

(ip.src == 192.168.1.102 && tcp.dstport == 80) || (ip.src==128.119.245.12 && tcp.dstport==1161)

Q9:

The ACK number in every packet from client 192.168.1.102 to server is 1 which means the client is telling the server to send the data starting from byte 1 which means the client has currently received a packet with body of 0 bytes from server.

Throughout the communication, it’s the client that is sending the file and not the server, that’s why server always will get an ACK of 1 (since its not sending any file data)

Q10:

The first ACK sent from client to server was for the three way handshake successful completion from client side to server. After that the client is continuously sending file data to the server while the server is just receiving the data and not sending any file data in return as a response. Now at the end client is again sending a file in a POST request to server, when the server gives the response page the length of that page is 730. Hence in the last ACK from client to server, the client has an ACK number 731 (intimating the server that kindly send the next data if any with the byte number starting from 731).