CS3000: Homework1

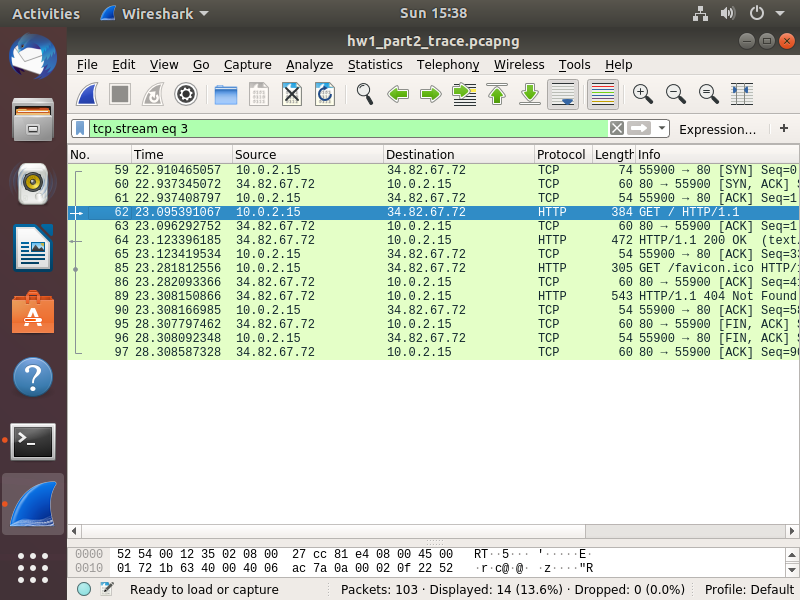
Part 1:

IP Address of client: 10.0.2.15

MAC Address of client: fe80::9dde::4245::3dbb::67ed

IP Address of server: 34.82.67.72

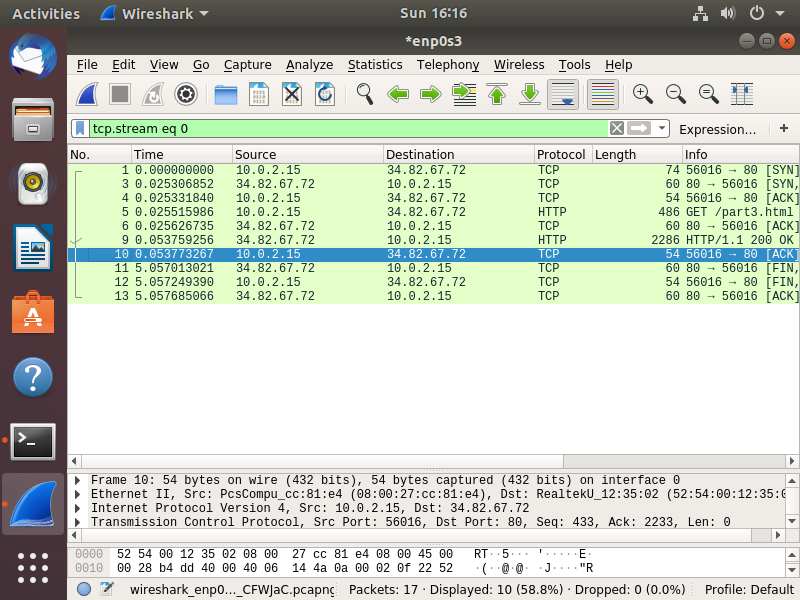
Part 2:



I found the http request on No. 62 and response on No.64.

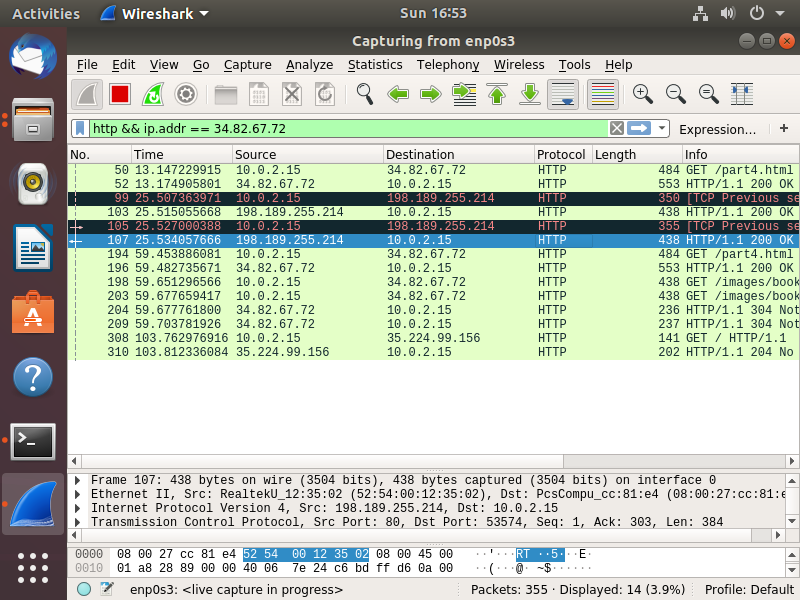
1. The http version of my browser in my client is HTTP/1.1. The server is also running HTTP/1.1
2. Based on the traces of the screenshot above, the client must be 10.0.2.15 because that is the source IP Address of the http get request. The server must be the destination for the http get because that is the IP Address of the server.
3. Status code I believe is HTTP/1.1 404 not found.
4. Server was last modified at Sun, 08 Mar 2020 22:20:28 GMT
5. The size of the html which can be found in the length section for wireshark, 54 bytes.

Part 3:



1. I got one HTTP GET request message, and the trace number for this No. 5.
2. The trace number that contains the status code is No. 9
3. The status code is 200 and the phrase is OK
4. When I ran the wireshark and collected the packets, the total number of TCP segments needed were 8 TCP segments. This is because I have to connect from the client port which is 56016 to the apache web server which is port 80. After sending the get request I had one more TCP segment, which the web server sent the connection port 80 to the client browser. After the http response, there were four more TCP segments.

Part 4:



1.Originally there are 3-4 get request messages, my browser ended up reloading multiple times. But there should be one GET request for the website and two GET requests for the objects. The trace numbers are 194, 198, and 203.

2. I believe that the browser downloaded the two images in parallel, this because the first and second image were both requested before being sent back.

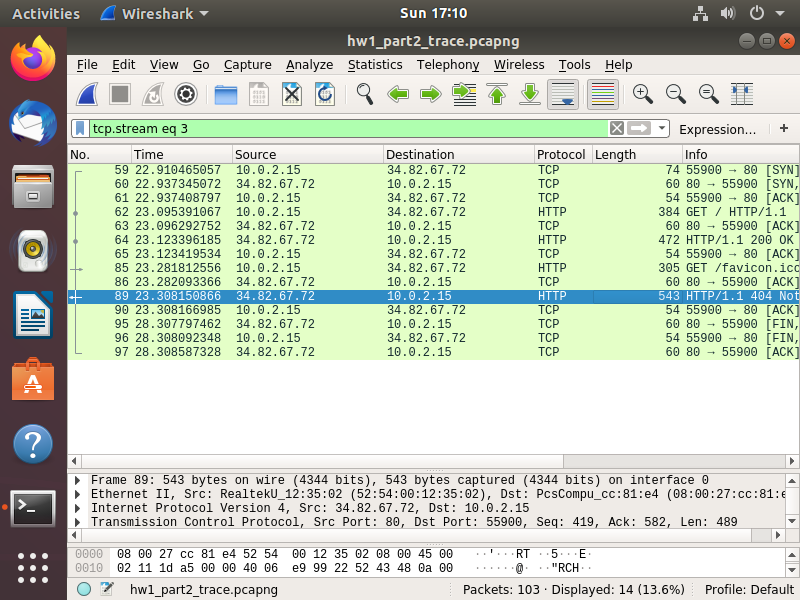
Part 5:

Http header information: GET / HTTP/1.1\r\n

source /destination port number: 55900/80

source /destination ip number: 10.0.2.15/34.82.67.72

source/destination MAC Address: 08:00:27:cc:81:e4/52:54:00:12:35:02



I know that the encapsulation happens on the sender side because the packet length encapsulates on the ethernet connection and increases from 384 length to 472. Decapsulation occurs when the request arrives at the server side this is when the length decreases again when coming back to the source.