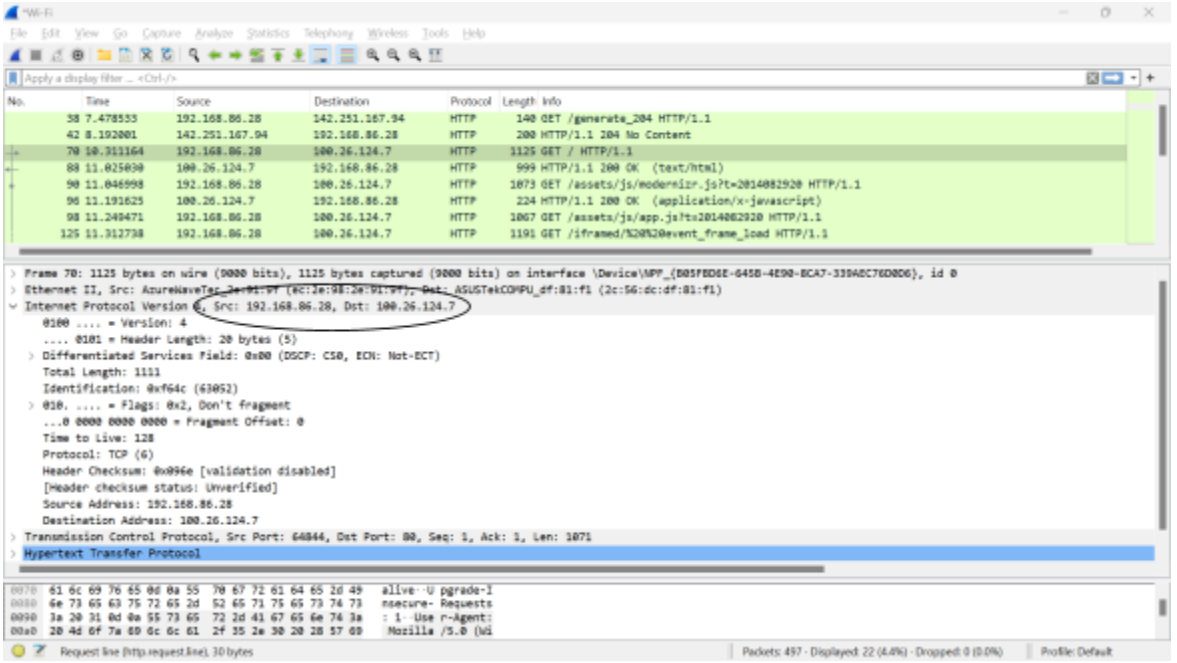
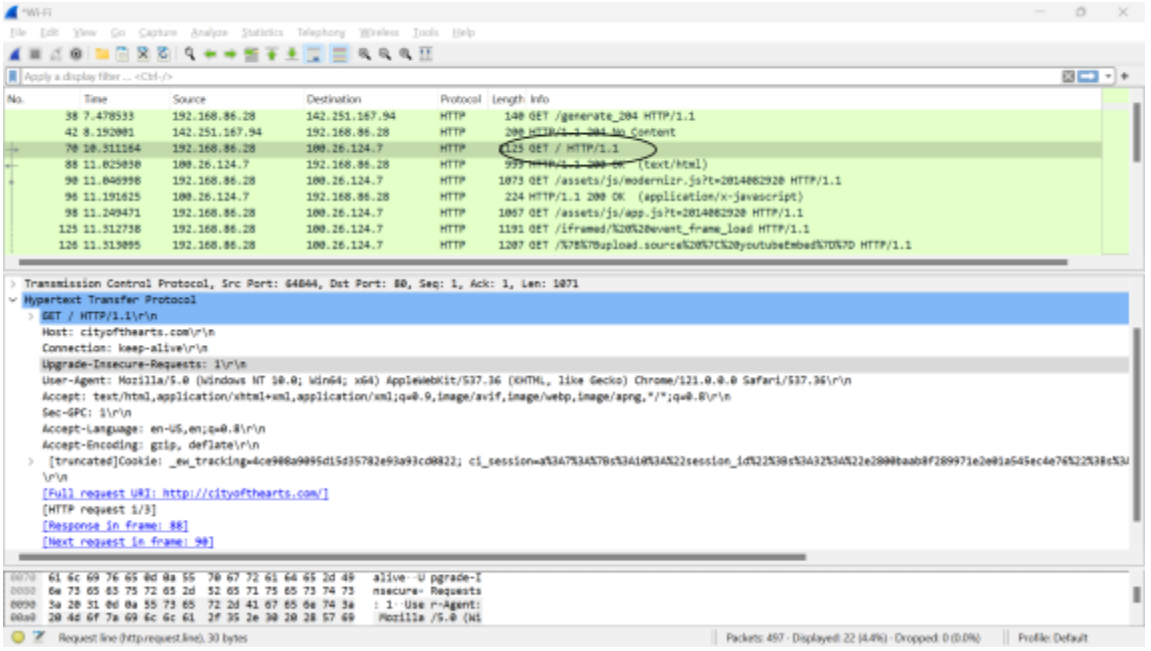
Section 1: Overview of Wireshark

1. Referring to the HTTP Get Request for http://cityofthearts.com/ answer the following questions:

   a. What was your IP address? What was the IP address of http://cityofthearts.com/ ?
      i. My IP address is 192.168.86.28
      ii. The IP address of cityofthearts.com is 100.26.124.7
      iii. Located:

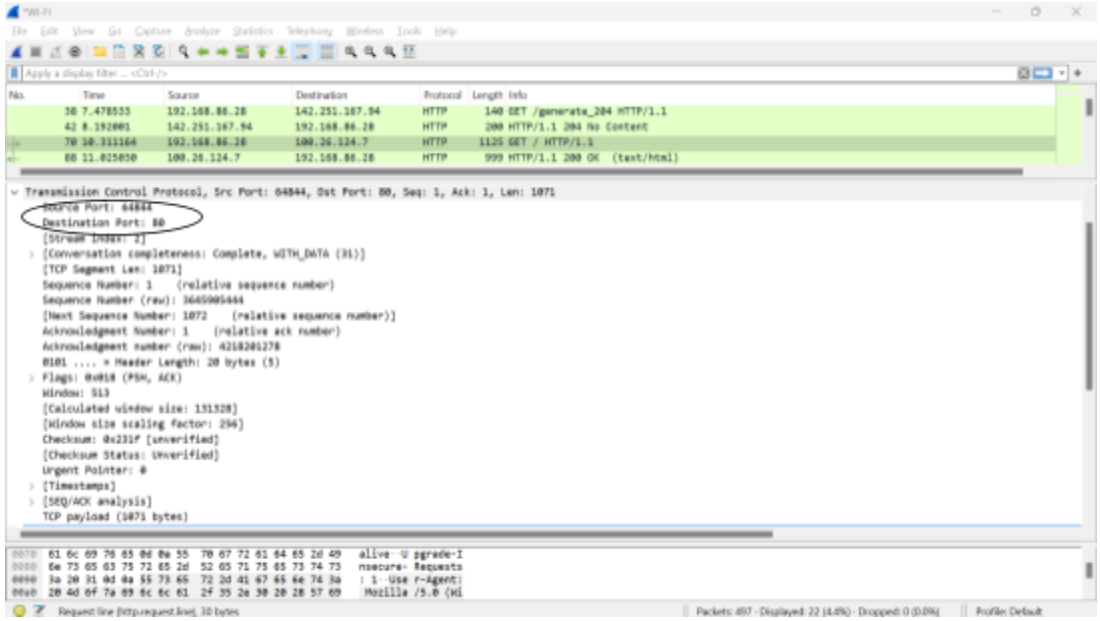


   b. What version of HTTP is your computer using?
      i. The HTTP version is 1.1
      ii. Located at:

1. 

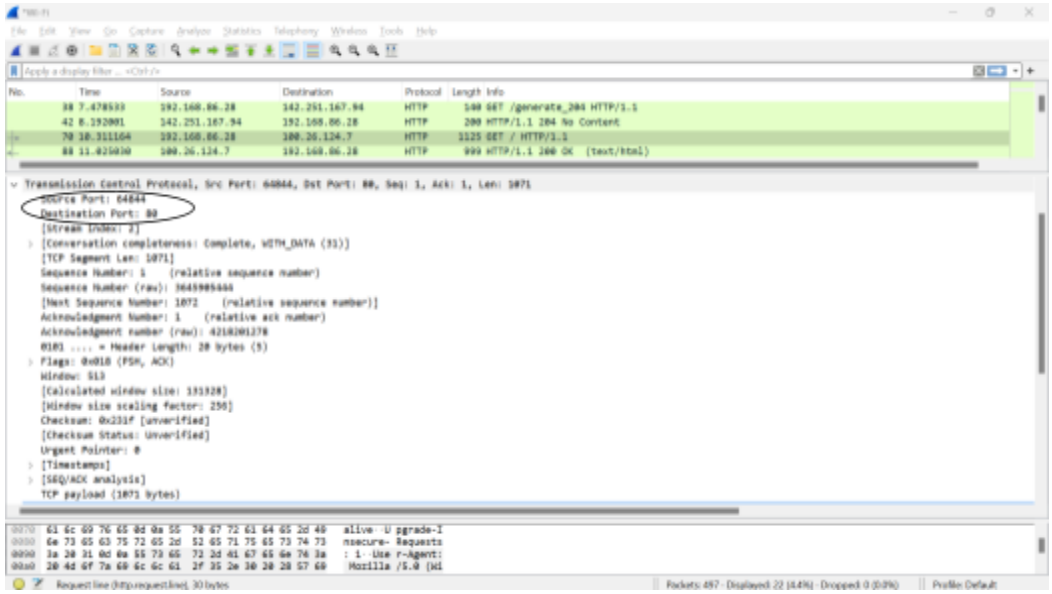
c. What is your computer's port number for this HTTP Get Request?

i. The port number for this HTTP request is 64844

1. 

d. What is the server's port number for http://cityofthearts.com/ ?

i. The server's port number is Port 80.

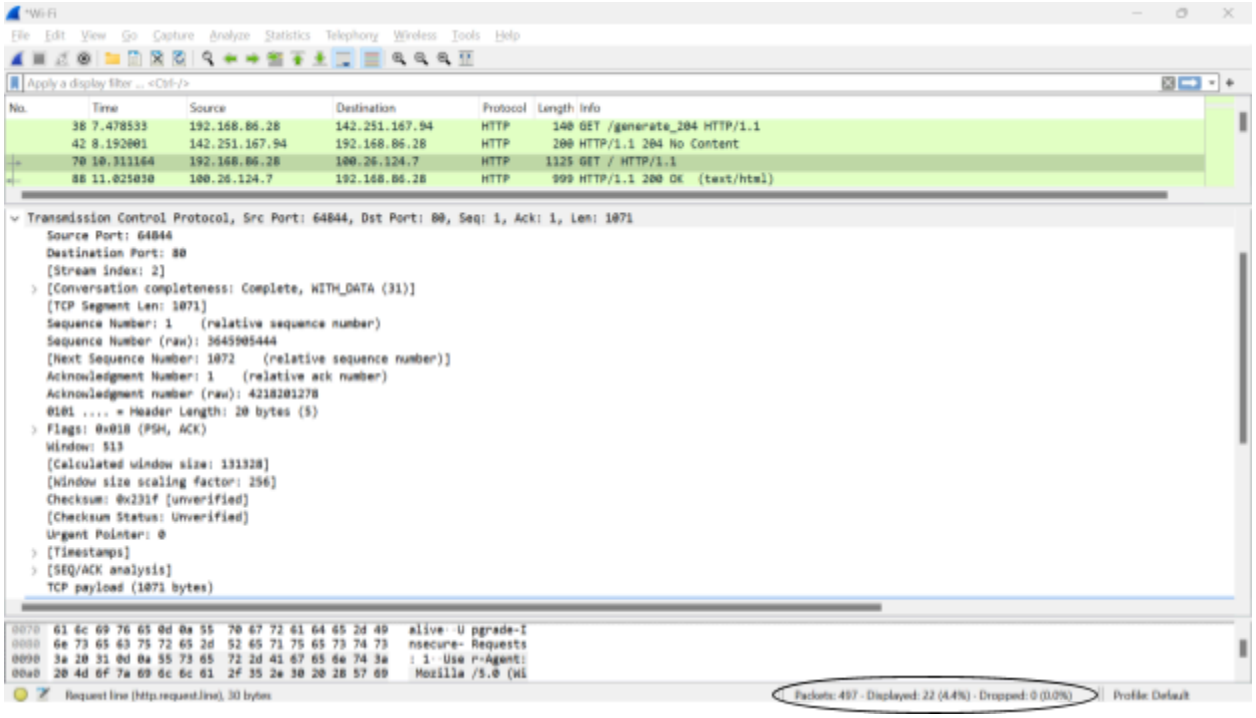
1. 

e. What does Keep-Alive mean?

i. Keep-Alive in the HTTP get request means to keep the connection alive between the host device and the webserver

2. How many packets did you capture (look at the bottom of your Wireshark window)?

497 packets



3. Did your computer communicate with other IP addresses? If so, list some of them (list up to 5 other IP addresses). What do you think the other addresses are for?

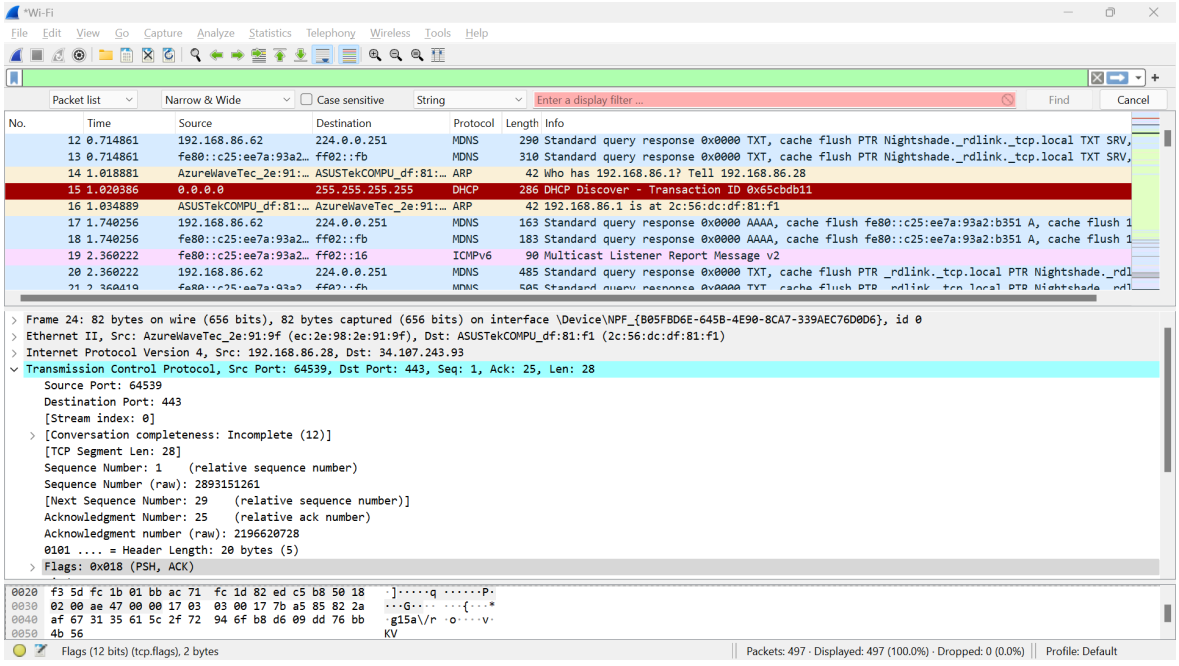
1. 142.251.167.94
2. 34.107.243.93
3. 224.0.0.251
4. 142.251.16.139

The highlighted yellow ip address is labeled as an MDNS protocol which is a multicast DNS so I guess it would help with distributing DNS requests among a small network.

34.107.243.93 was sent as a retransmission message to port 443 so it was sent to an HTTPS port.

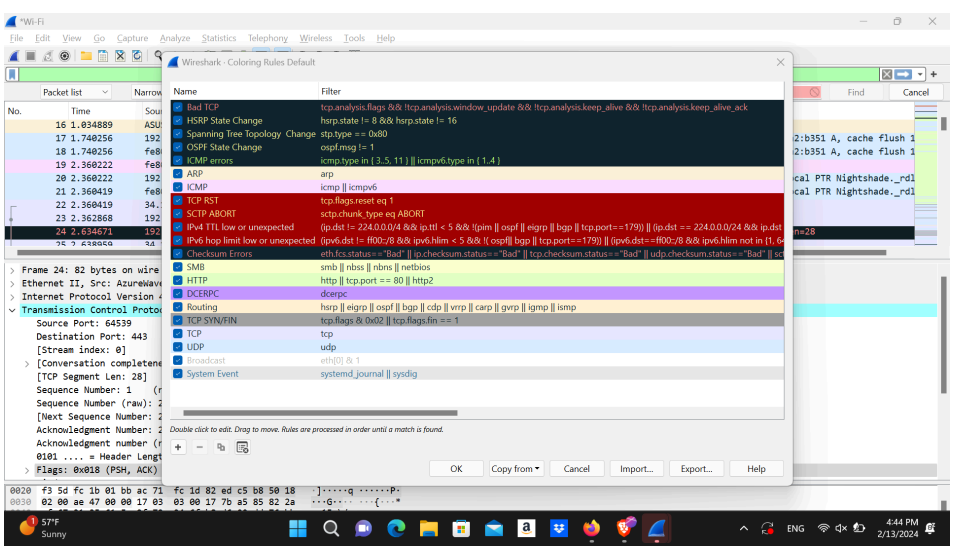
4. Refer to the top pane: List some (at least 4) of the protocol types you see.

1. TCP
2. MDNS
3. ARP
4. ICMPv6

5. 

5. Refer to the Coloring Rules (Go to View => Coloring Rules) . What do the different colors mean (from what you see in your output):

A. What color is used for TCP?
   a. Grayish Blue
B. What color is used for HTTP?
   a. Light Green
C. What color is used for UDP?
   a. Light blue
D. What color is used for ARP?
   a. Yellow
E. What color is used for TCP RST?
   a. Dark red with yellow text

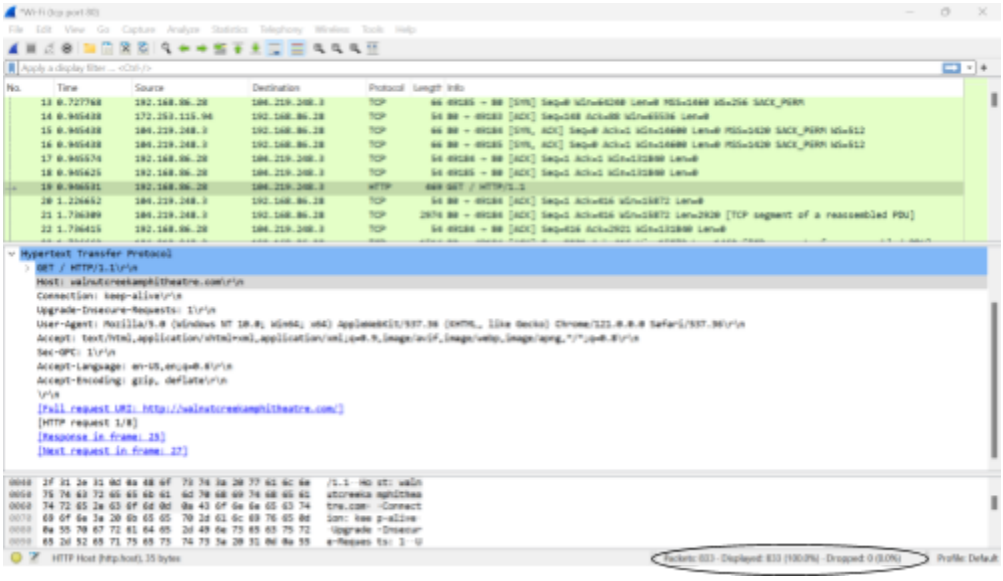


Section 2: TCP/DNS Filtering

1. Answer these questions based on the exercise you just completed.

   a. How many total packets did you capture when filtering for HTTP = 80?

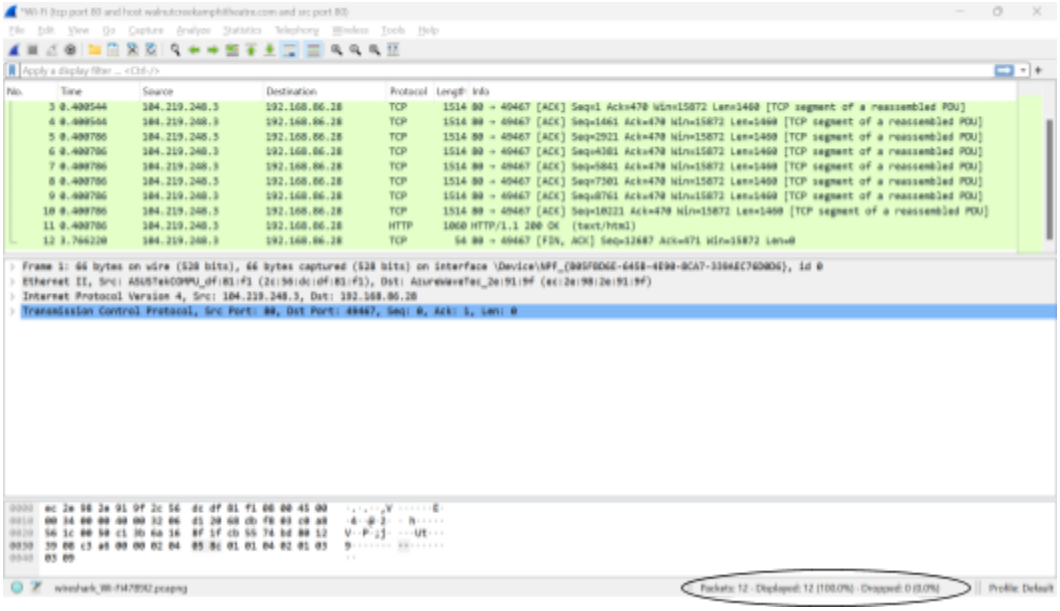
      1. 833 total packets when filtering for TCP = 80



   a.

   b. How many did you capture when filtering for traffic between you and walnutcreekamphitheatre.com with source port 80?

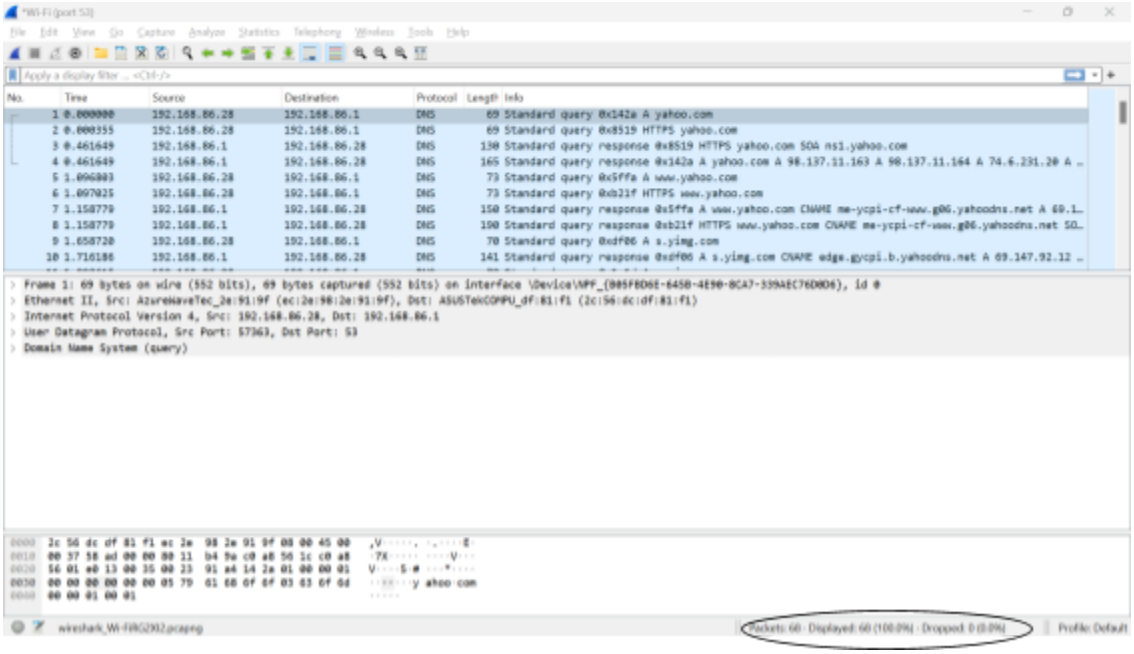
      i. 12 packets when filtering between me and walnutcreekamphitheatre.com



      1.

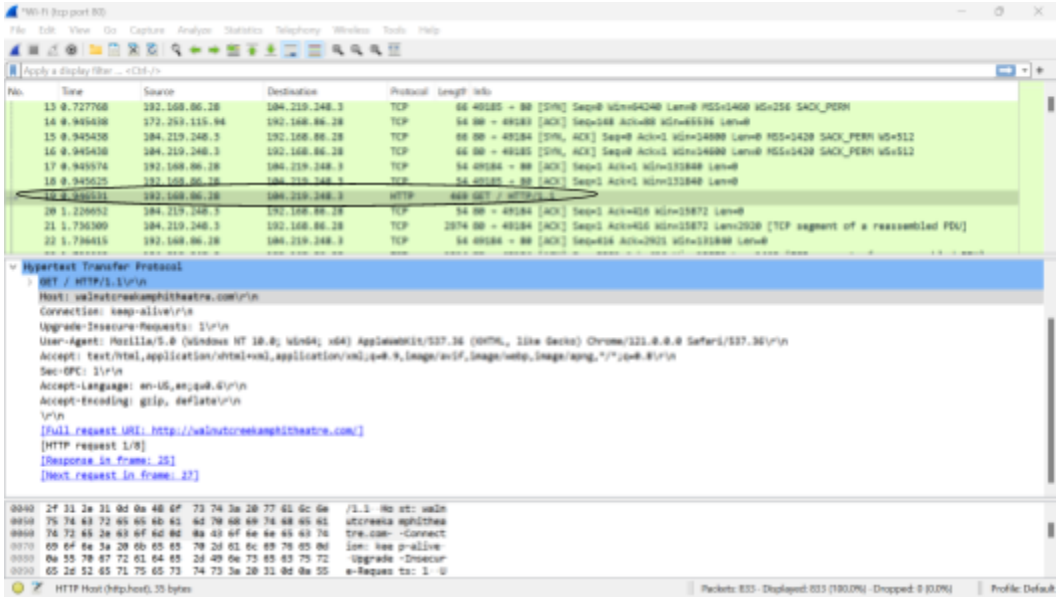
   c. How many did you capture when filtering for DNS?

      i. 68 packets when filtering for DNS

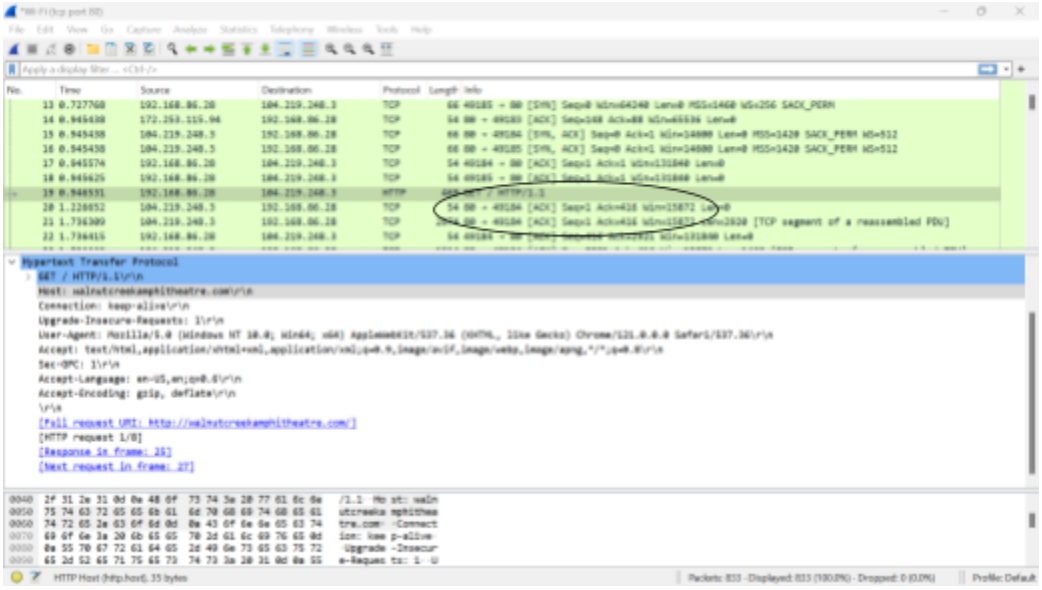
1. 

2. When you first captured packets while filtering for HTTP=80,

   a. What row (number) contained the initial GET request for the website?
      i. Row 19 contained the initial get request

1. 

   b. What port was your computer using on the initial GET request?
      i. The port number my computer was using on the initial get request was 49184

1. 

3. What was your IP address for this exercise?

My IP address was 192.168.86.28

4. Why did your computer send so many HTTP packets ?

My computer sent so many HTTP packets because there were lots of elements on the webpage that my computer had to retrieve. In web development terms there were many elements in the webpage that included but are not limited to, images, banners, paragraphs, headers, links, and scripts. My computer had to retrieve all those elements so that's why there were so many http requests.

5. Why did your computer send so many DNS packets?

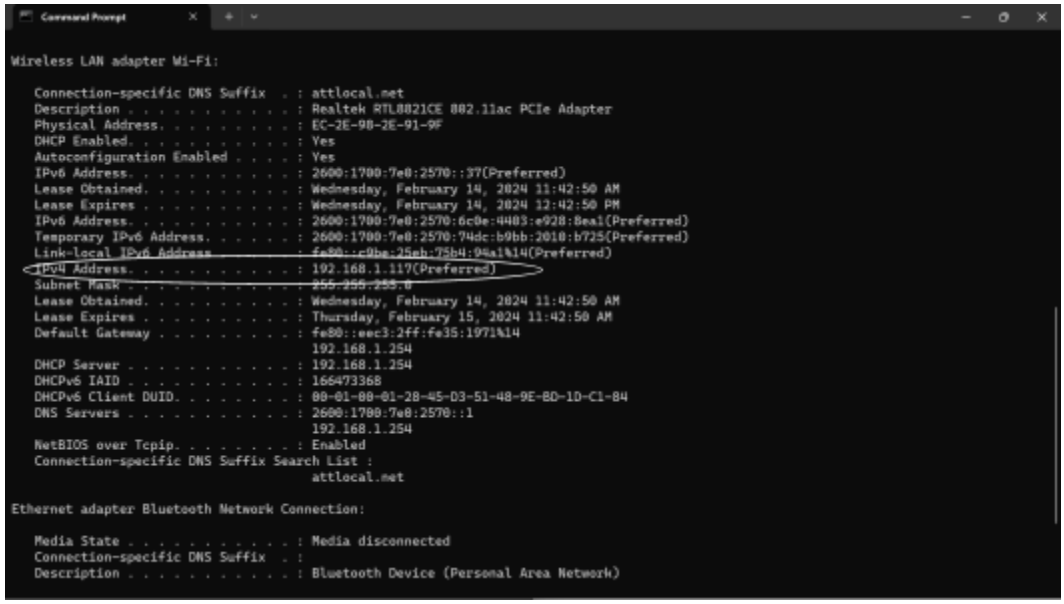
My computer sent so many DNS packets because specifically on yahoo.com, not all the elements could have originated from the yahoo.com webserver. One element would be ads, so my computer would have to send a separate dns request to another web server in order to resolve the showing of the ads on yahoo.com because yahoo.com might be using an external third party service to display these ads or other elements of their webpage.

6. In your own words, explain what you did in this section (don't forget you did 3 different things).

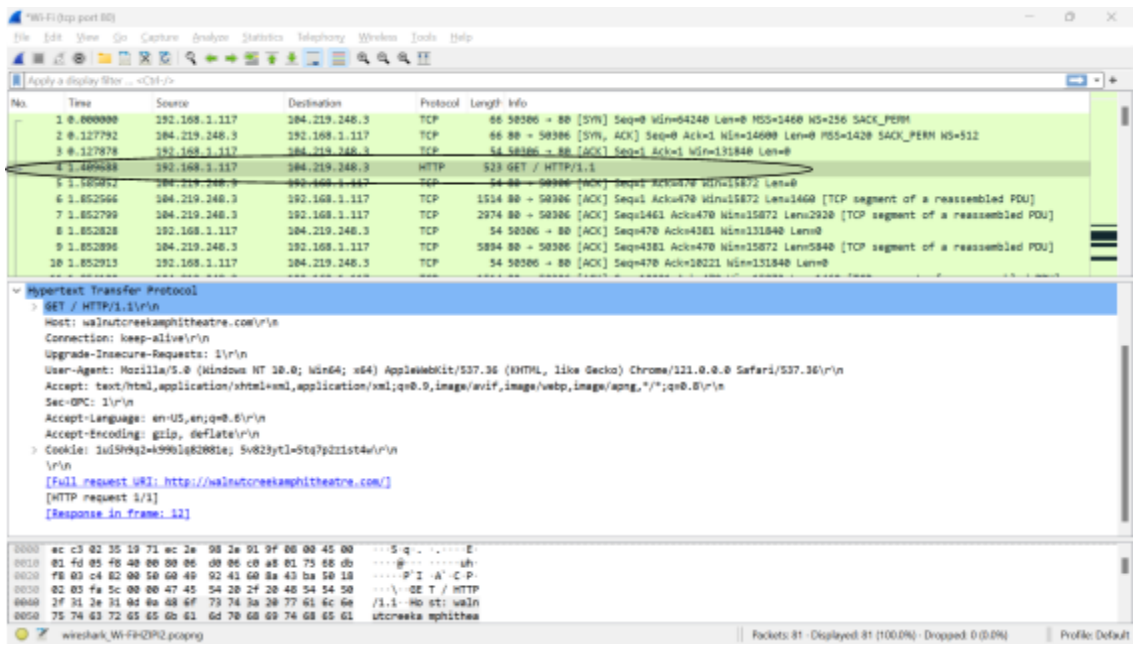
In this section I filtered my packet traffic specifically to three different types of packets. That would be, port 80 traffic, Domain Name System (DNS) traffic and port 80 traffic specific to my computer and a specific website. At first, I filtered my packet traffic to only those that interacted with port 80 or the HTTP port of web servers. Second, I filtered my web traffic to the port 80 or http port specific to walnutcreekamphitheatre.com. Lastly, i filtered my packet traffic to only those that involved port 53 or DNS packets.

Section 3: Dive into TCP Protocol

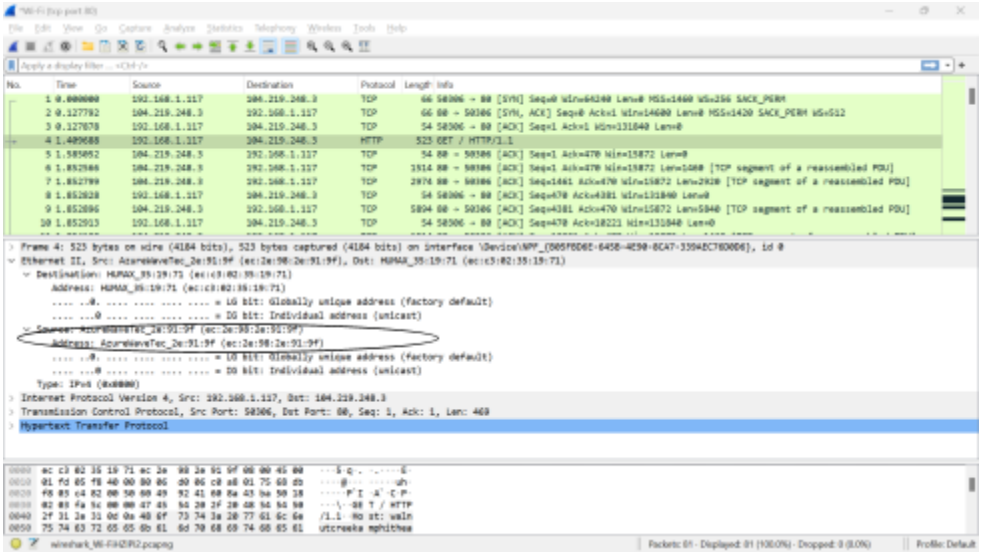
1. What was your IP address during this part?
    a. 192.168.1.117

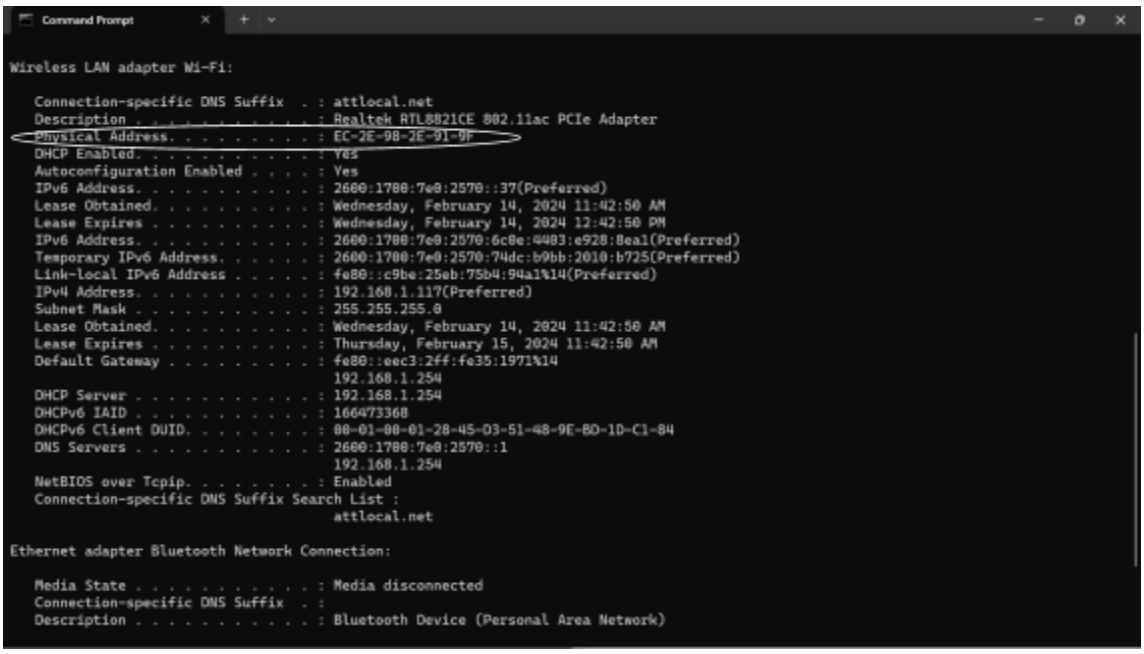
b. 

```
Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . : attlocal.net
   Description . . . . . . . . . . . : Realtek RTL8821CE 802.11ac PCIe Adapter
   Physical Address. . . . . . . . . : EC-2E-98-2E-91-9F
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes
   IPv6 Address. . . . . . . . . . . : 2600:1700:7e0:2570::37(Preferred)
   Lease Obtained. . . . . . . . . . : Wednesday, February 14, 2024 11:42:50 AM
   Lease Expires . . . . . . . . . . : Wednesday, February 14, 2024 12:42:50 PM
   IPv6 Address. . . . . . . . . . . : 2600:1700:7e0:2570:6c0e:4403:e928:8ea1(Preferred)
   Temporary IPv6 Address. . . . . . : 2600:1700:7e0:2570:74dc:b9bb:2010:b725(Preferred)
   Link-local IPv6 Address . . . . . : fe80::c9be:25eb:75b4:94a1%14(Preferred)
   IPv4 Address. . . . . . . . . . . : 192.168.1.117(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Lease Obtained. . . . . . . . . . : Wednesday, February 14, 2024 11:42:50 AM
   Lease Expires . . . . . . . . . . : Thursday, February 15, 2024 11:42:50 AM
   Default Gateway . . . . . . . . . : fe80::eec3:2ff:fe35:1971%14
                                       192.168.1.254
   DHCP Server . . . . . . . . . . . : 192.168.1.254
   DHCPv6 IAID . . . . . . . . . . . : 166473368
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-2B-45-D3-51-48-9E-BD-1D-C1-84
   DNS Servers . . . . . . . . . . . : 2600:1700:7e0:2570::1
                                       192.168.1.254
   NetBIOS over Tcpip. . . . . . . . : Enabled
   Connection-specific DNS Suffix Search List :
                                       attlocal.net

Ethernet adapter Bluetooth Network Connection:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Bluetooth Device (Personal Area Network)
```

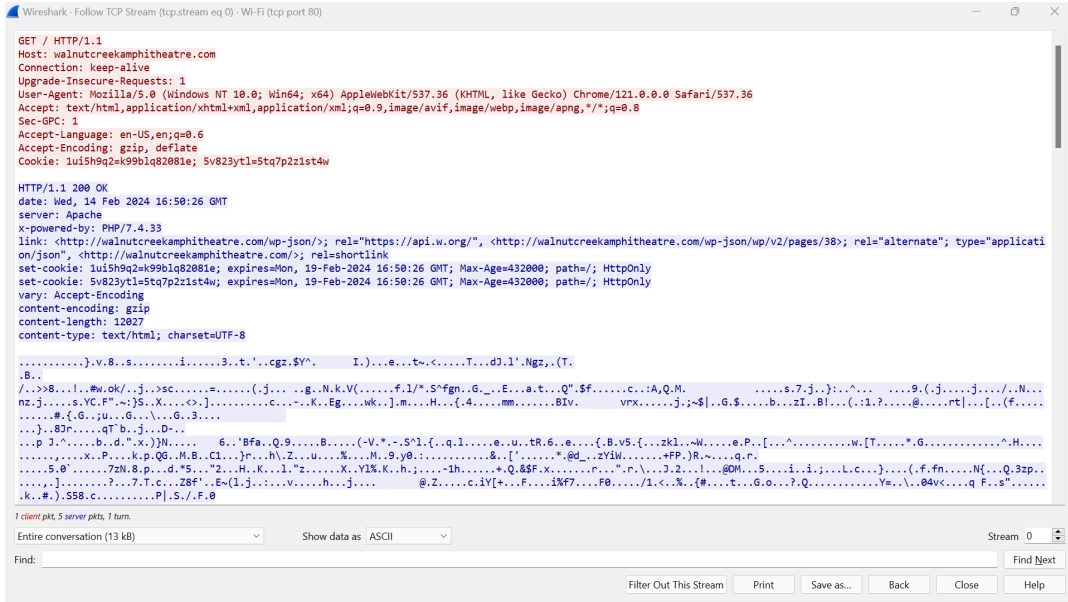
c. 

2. What was your Physical address for this part?
   a. ec:2e:98:2e:91:9f

b. 

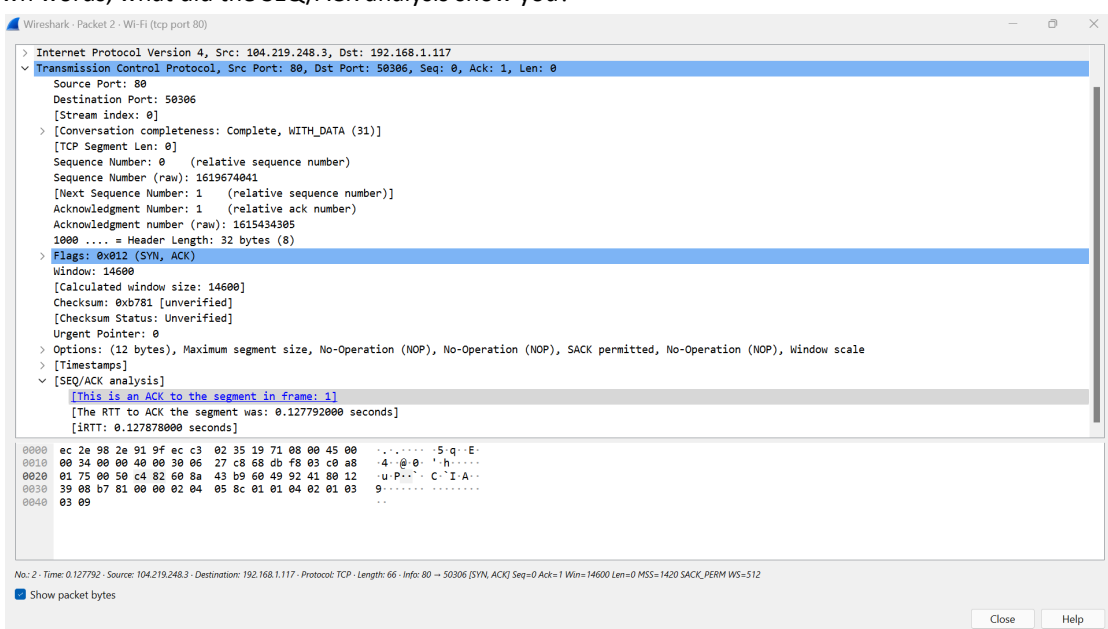
c. 

3. Use your Wireshark output and find the IP address for walnutcreekamphitheatre.com . What is the IP address?
    a. e
    b. The ip address is 104.219.248.3
4. In your own words, what did Wireshark show you when you used the Follow TCP Stream feature (steps 1-23)? (be specific) eeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee

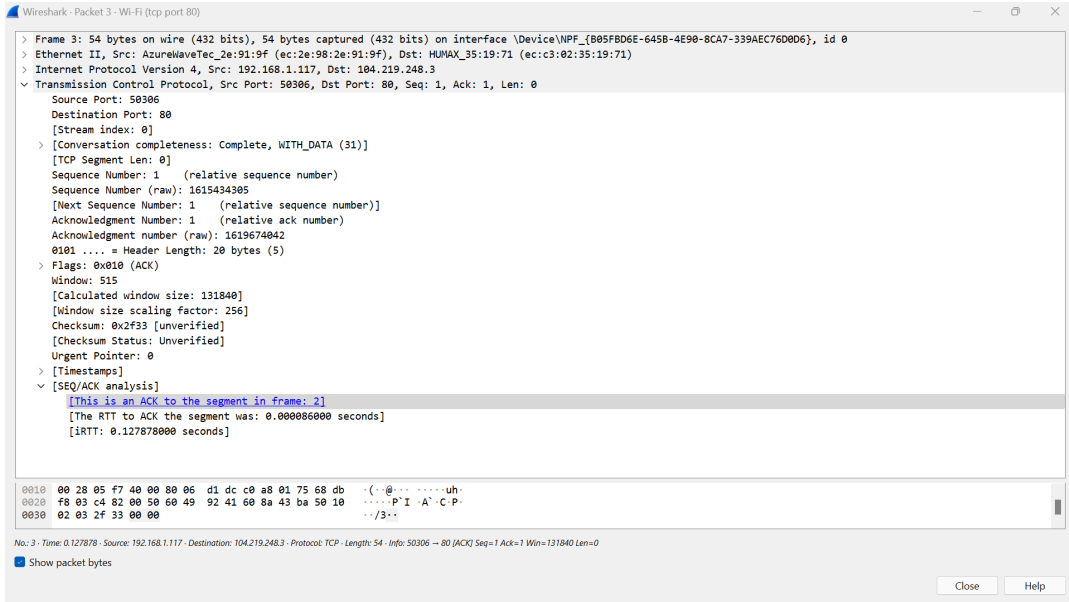
a. 

b. The wireshark TCP stream showed me a summarized view of the interaction between my device the web server of walnutcreekamphitheatre. The red indicates the communications originating from my device to walnutcreekamphitheatre and the blue indicates all communications vice versa. The information displayed looks to be encrypted but walnutcreekamphitheatre sent my device information about the webpage including, where it was hosted: apache, the domain name and the <link> tag in the HTML, a cookie that shows when it expires, and the character set being used on the website, lastly the length of the content is included as well.

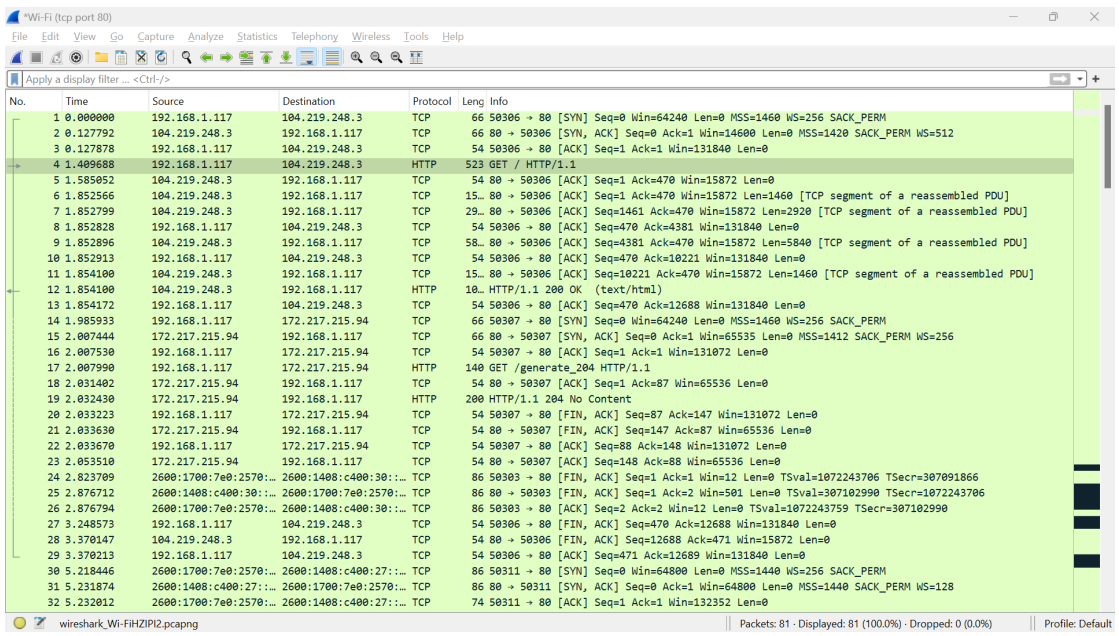
5. In your own words, what did the SEQ/ACK analysis show you?

a. 

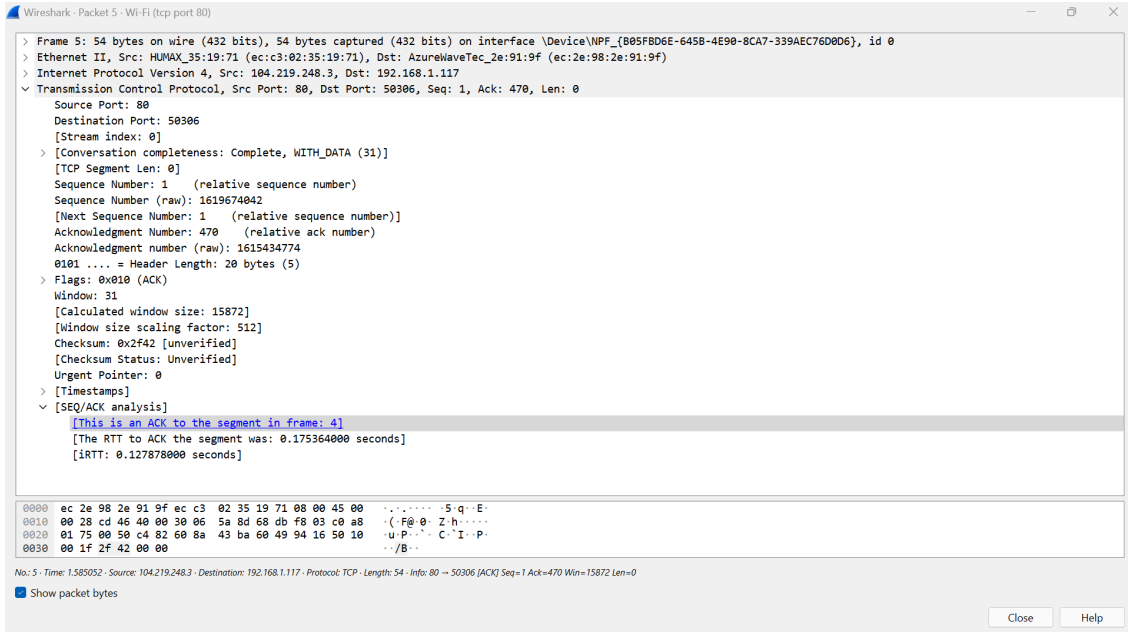
b.


Wireshark · Packet 3 · Wi-Fi (tcp port 80)
> Frame 3: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface \Device\NPF_{B05FBD6E-645B-4E90-8CA7-339AEC76D0D6}, id 0
> Ethernet II, Src: AzureWaveTec_2e:91:9f (ec:2e:98:2e:91:9f), Dst: HUMAX_35:19:71 (ec:c3:02:35:19:71)
> Internet Protocol Version 4, Src: 192.168.1.117, Dst: 104.219.248.3
v Transmission Control Protocol, Src Port: 50306, Dst Port: 80, Seq: 1, Ack: 1, Len: 0
Source Port: 50306
Destination Port: 80
[Stream index: 0]
> [Conversation completeness: Complete, WITH_DATA (31)]
[TCP Segment Len: 0]
Sequence Number: 1 (relative sequence number)
Sequence Number (raw): 1615434305
[Next Sequence Number: 1 (relative sequence number)]
Acknowledgment Number: 1 (relative ack number)
Acknowledgment number (raw): 1619674042
0101 .... = Header Length: 20 bytes (5)
> Flags: 0x010 (ACK)
Window: 515
[Calculated window size: 131840]
[Window size scaling factor: 256]
Checksum: 0x2f33 [unverified]
[Checksum Status: Unverified]
Urgent Pointer: 0
> [Timestamps]
v [SEQ/ACK analysis]
[This is an ACK to the segment in frame: 2]
[The RTT to ACK the segment was: 0.000086000 seconds]
[iRTT: 0.127878000 seconds]
0010 00 28 05 f7 40 00 80 06 d1 dc c0 a8 01 75 68 db
0020 f8 03 c4 82 00 50 60 49 92 41 60 8a 43 ba 50 10
0030 02 03 2f 33 00 00
No.: 3 · Time: 0.127878 · Source: 192.168.1.117 · Destination: 104.219.248.3 · Protocol: TCP · Length: 54 · Info: 50306 → 80 [ACK] Seq=1 Ack=1 Win=131840 Len=0
Show packet bytes
Close
Help

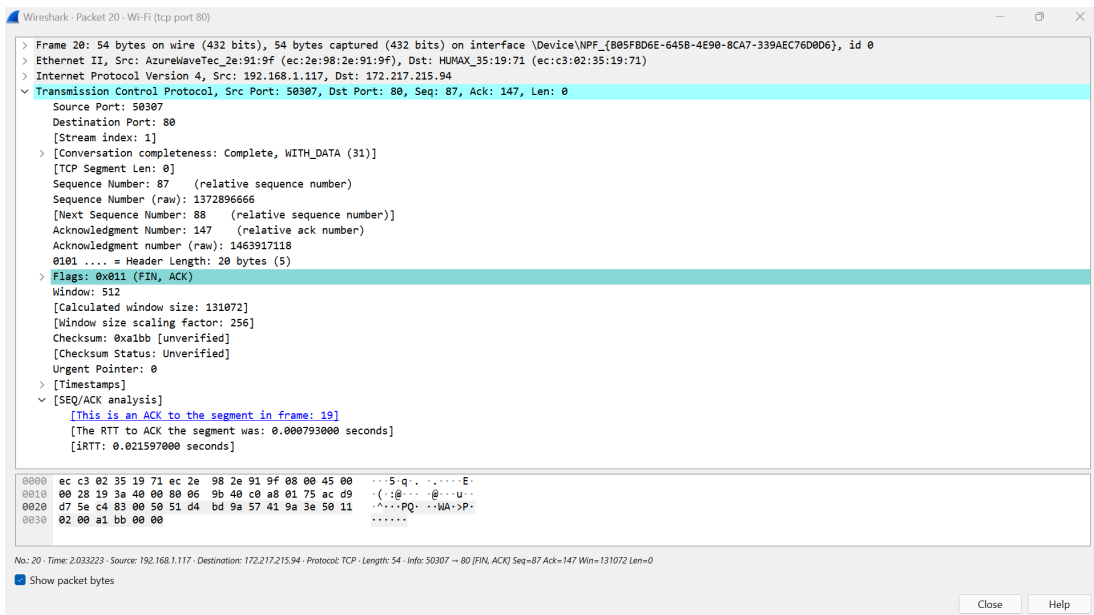
c.


*Wi-Fi (tcp port 80)
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
No. Time Source Destination Protocol Leng Info
1 0.000000 192.168.1.117 104.219.248.3 TCP 66 50306 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
2 0.127792 104.219.248.3 192.168.1.117 TCP 66 80 → 50306 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1420 SACK_PERM WS=512
3 0.127878 192.168.1.117 104.219.248.3 TCP 54 50306 → 80 [ACK] Seq=1 Ack=1 Win=131840 Len=0
4 1.409688 192.168.1.117 104.219.248.3 HTTP 523 GET / HTTP/1.1
5 1.585052 104.219.248.3 192.168.1.117 TCP 54 80 → 50306 [ACK] Seq=1 Ack=470 Win=15872 Len=0
6 1.852566 104.219.248.3 192.168.1.117 TCP 15… 80 → 50306 [ACK] Seq=1 Ack=470 Win=15872 Len=1460 [TCP segment of a reassembled PDU]
7 1.852799 104.219.248.3 192.168.1.117 TCP 29… 80 → 50306 [ACK] Seq=1461 Ack=470 Win=15872 Len=2920 [TCP segment of a reassembled PDU]
8 1.852828 192.168.1.117 104.219.248.3 TCP 54 50306 → 80 [ACK] Seq=470 Ack=4381 Win=131840 Len=0
9 1.852896 104.219.248.3 192.168.1.117 TCP 58… 80 → 50306 [ACK] Seq=4381 Ack=470 Win=15872 Len=5840 [TCP segment of a reassembled PDU]
10 1.852913 192.168.1.117 104.219.248.3 TCP 54 50306 → 80 [ACK] Seq=470 Ack=10221 Win=131840 Len=0
11 1.854100 104.219.248.3 192.168.1.117 TCP 15… 80 → 50306 [ACK] Seq=10221 Ack=470 Win=15872 Len=1460 [TCP segment of a reassembled PDU]
12 1.854100 104.219.248.3 192.168.1.117 HTTP 10… HTTP/1.1 200 OK (text/html)
13 1.854172 192.168.1.117 104.219.248.3 TCP 54 50306 → 80 [ACK] Seq=470 Ack=12688 Win=131840 Len=0
14 1.985933 192.168.1.117 172.217.215.94 TCP 66 50307 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
15 2.007444 172.217.215.94 192.168.1.117 TCP 66 80 → 50307 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1412 SACK_PERM WS=256
16 2.007530 192.168.1.117 172.217.215.94 TCP 54 50307 → 80 [ACK] Seq=1 Ack=1 Win=131072 Len=0
17 2.007990 192.168.1.117 172.217.215.94 HTTP 140 GET /generate_204 HTTP/1.1
18 2.031402 172.217.215.94 192.168.1.117 TCP 54 80 → 50307 [ACK] Seq=1 Ack=87 Win=65536 Len=0
19 2.032430 172.217.215.94 192.168.1.117 HTTP 200 HTTP/1.1 204 No Content
20 2.033223 192.168.1.117 172.217.215.94 TCP 54 50307 → 80 [FIN, ACK] Seq=87 Ack=147 Win=131072 Len=0
21 2.033630 172.217.215.94 192.168.1.117 TCP 54 80 → 50307 [FIN, ACK] Seq=147 Ack=87 Win=65536 Len=0
22 2.033670 192.168.1.117 172.217.215.94 TCP 54 50307 → 80 [ACK] Seq=88 Ack=148 Win=131072 Len=0
23 2.053510 172.217.215.94 192.168.1.117 TCP 54 80 → 50307 [ACK] Seq=148 Ack=88 Win=65536 Len=0
24 2.823709 2600:1700:7e0:2570:… 2600:1408:c400:30::… TCP 86 50303 → 80 [FIN, ACK] Seq=1 Ack=1 Win=12 Len=0 TSval=1072243706 TSecr=307091866
25 2.876712 2600:1408:c400:30::… 2600:1700:7e0:2570:… TCP 86 80 → 50303 [FIN, ACK] Seq=1 Ack=2 Win=501 Len=0 TSval=307102990 TSecr=1072243706
26 2.876794 2600:1700:7e0:2570:… 2600:1408:c400:30::… TCP 86 50303 → 80 [ACK] Seq=2 Ack=2 Win=12 Len=0 TSval=1072243759 TSecr=307102990
27 3.248573 192.168.1.117 104.219.248.3 TCP 54 50306 → 80 [FIN, ACK] Seq=470 Ack=12688 Win=131840 Len=0
28 3.370147 104.219.248.3 192.168.1.117 TCP 54 80 → 50306 [FIN, ACK] Seq=12688 Ack=471 Win=15872 Len=0
29 3.370213 192.168.1.117 104.219.248.3 TCP 54 50306 → 80 [ACK] Seq=471 Ack=12689 Win=131840 Len=0
30 5.218446 2600:1700:7e0:2570:… 2600:1408:c400:27::… TCP 86 50311 → 80 [SYN] Seq=0 Win=64800 Len=0 MSS=1440 WS=256 SACK_PERM
31 5.231874 2600:1408:c400:27::… 2600:1700:7e0:2570:… TCP 86 80 → 50311 [SYN, ACK] Seq=0 Ack=1 Win=64800 Len=0 MSS=1440 SACK_PERM WS=128
32 5.232012 2600:1700:7e0:2570:… 2600:1408:c400:27::… TCP 74 50311 → 80 [ACK] Seq=1 Ack=1 Win=132352 Len=0
wireshark_Wi-FiHZIPI2.pcapng
Packets: 81 · Displayed: 81 (100.0%) · Dropped: 0 (0.0%)
Profile: Default

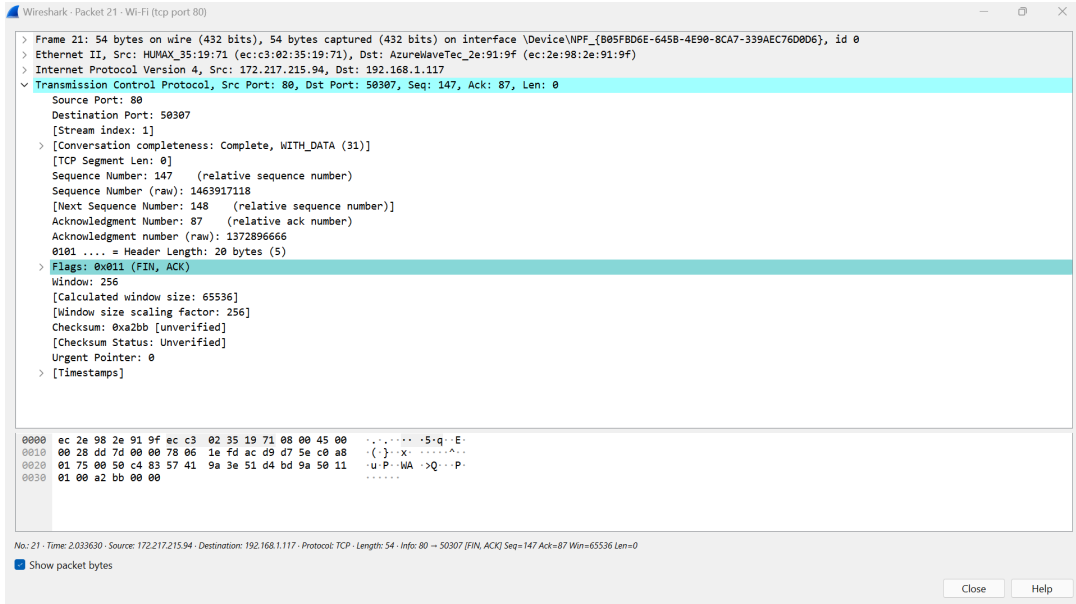
d.


Wireshark · Packet 5 · Wi-Fi (tcp port 80)
Frame 5: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface \Device\NPF_{B05FBD6E-645B-4E90-8CA7-339AEC76D0D6}, id 0
Ethernet II, Src: HUMAX_35:19:71 (ec:c3:02:35:19:71), Dst: AzureWaveTec_2e:91:9f (ec:2e:98:2e:91:9f)
Internet Protocol Version 4, Src: 104.219.248.3, Dst: 192.168.1.117
Transmission Control Protocol, Src Port: 80, Dst Port: 50306, Seq: 1, Ack: 470, Len: 0
Source Port: 80
Destination Port: 50306
[Stream index: 0]
[Conversation completeness: Complete, WITH_DATA (31)]
[TCP Segment Len: 0]
Sequence Number: 1 (relative sequence number)
Sequence Number (raw): 1619674042
[Next Sequence Number: 1 (relative sequence number)]
Acknowledgment Number: 470 (relative ack number)
Acknowledgment number (raw): 1615434774
0101 .... = Header Length: 20 bytes (5)
Flags: 0x010 (ACK)
Window: 31
[Calculated window size: 15872]
[Window size scaling factor: 512]
Checksum: 0x2f42 [unverified]
[Checksum Status: Unverified]
Urgent Pointer: 0
[Timestamps]
[SEQ/ACK analysis]
[This is an ACK to the segment in frame: 4]
[The RTT to ACK the segment was: 0.175364000 seconds]
[iRTT: 0.127878000 seconds]
0000 ec 2e 98 2e 91 9f ec c3 02 35 19 71 08 00 45 00
0010 00 28 cd 46 40 00 30 06 5a 8d 68 db f8 03 c0 a8
0020 01 75 00 50 c4 82 60 8a 43 ba 60 49 94 16 50 10
0030 00 1f 2f 42 00 00
No.: 5 · Time: 1.585052 · Source: 104.219.248.3 · Destination: 192.168.1.117 · Protocol: TCP · Length: 54 · Info: 80 → 50306 [ACK] Seq=1 Ack=470 Win=15872 Len=0
Show packet bytes
Close
Help

e.


Wireshark · Packet 20 · Wi-Fi (tcp port 80)
Frame 20: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface \Device\NPF_{B05FBD6E-645B-4E90-8CA7-339AEC76D0D6}, id 0
Ethernet II, Src: AzureWaveTec_2e:91:9f (ec:2e:98:2e:91:9f), Dst: HUMAX_35:19:71 (ec:c3:02:35:19:71)
Internet Protocol Version 4, Src: 192.168.1.117, Dst: 172.217.215.94
Transmission Control Protocol, Src Port: 50307, Dst Port: 80, Seq: 87, Ack: 147, Len: 0
Source Port: 50307
Destination Port: 80
[Stream index: 1]
[Conversation completeness: Complete, WITH_DATA (31)]
[TCP Segment Len: 0]
Sequence Number: 87 (relative sequence number)
Sequence Number (raw): 1372896666
[Next Sequence Number: 88 (relative sequence number)]
Acknowledgment Number: 147 (relative ack number)
Acknowledgment number (raw): 1463917118
0101 .... = Header Length: 20 bytes (5)
Flags: 0x011 (FIN, ACK)
Window: 512
[Calculated window size: 131072]
[Window size scaling factor: 256]
Checksum: 0xa1bb [unverified]
[Checksum Status: Unverified]
Urgent Pointer: 0
[Timestamps]
[SEQ/ACK analysis]
[This is an ACK to the segment in frame: 19]
[The RTT to ACK the segment was: 0.000793000 seconds]
[iRTT: 0.021597000 seconds]
0000 ec c3 02 35 19 71 ec 2e 98 2e 91 9f 08 00 45 00
0010 00 28 19 3a 40 00 80 06 9b 40 c0 a8 01 75 ac d9
0020 d7 5e c4 83 00 50 51 d4 bd 9a 57 41 9a 3e 50 11
0030 02 00 a1 bb 00 00
No.: 20 · Time: 2.033223 · Source: 192.168.1.117 · Destination: 172.217.215.94 · Protocol: TCP · Length: 54 · Info: 50307 → 80 [FIN, ACK] Seq=87 Ack=147 Win=131072 Len=0
Show packet bytes
Close
Help

f.

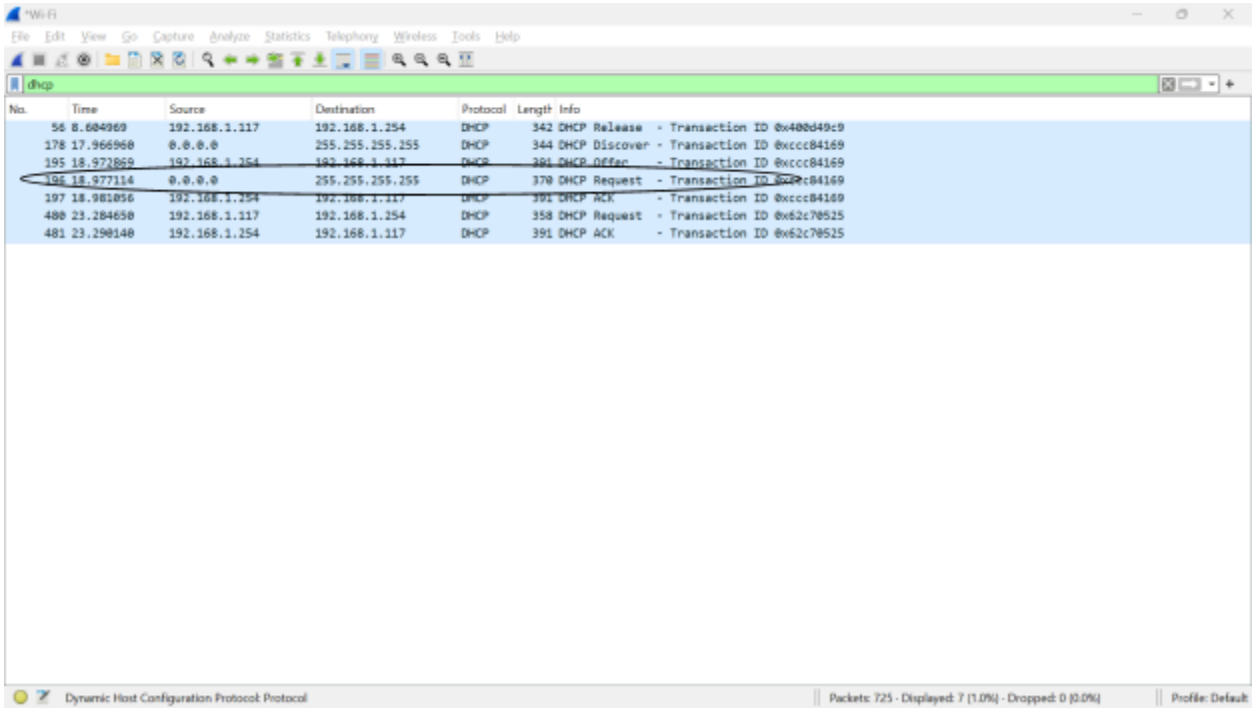


i. The SEQ/ACK analysis showed me how TCP protocol is reliable and the process of how TCP engages in opening and closing a connection between two clients.
ii. The first two screenshots shows the last steps of the three way open, where my computer sends a SYN to the web server, the web servers send back a SYN/ACK (screenshot A) and my computer sends an ACK acknowledging that I receive the request to open the connection (screenshot B).
iii. Screenshots C and D depict how my computer then sends the HTTP get request to the web server requesting the information of the webpage and the web server acknowledges that it received the HTTP get request.
iv. Screenshots E and F depict parts of the 4 way close. The web server initially has no more information to send over (frame 19). My device then sends a FIN/ACK meaning that my device acknowledges theres no more information and thus wants to close the connection (Screenshot E). Screenshot F then shows the FIN/ACK sent from the web server though it is important to note that there was no SEQ/ACK analysis on that frame.

Section 4: DHCP

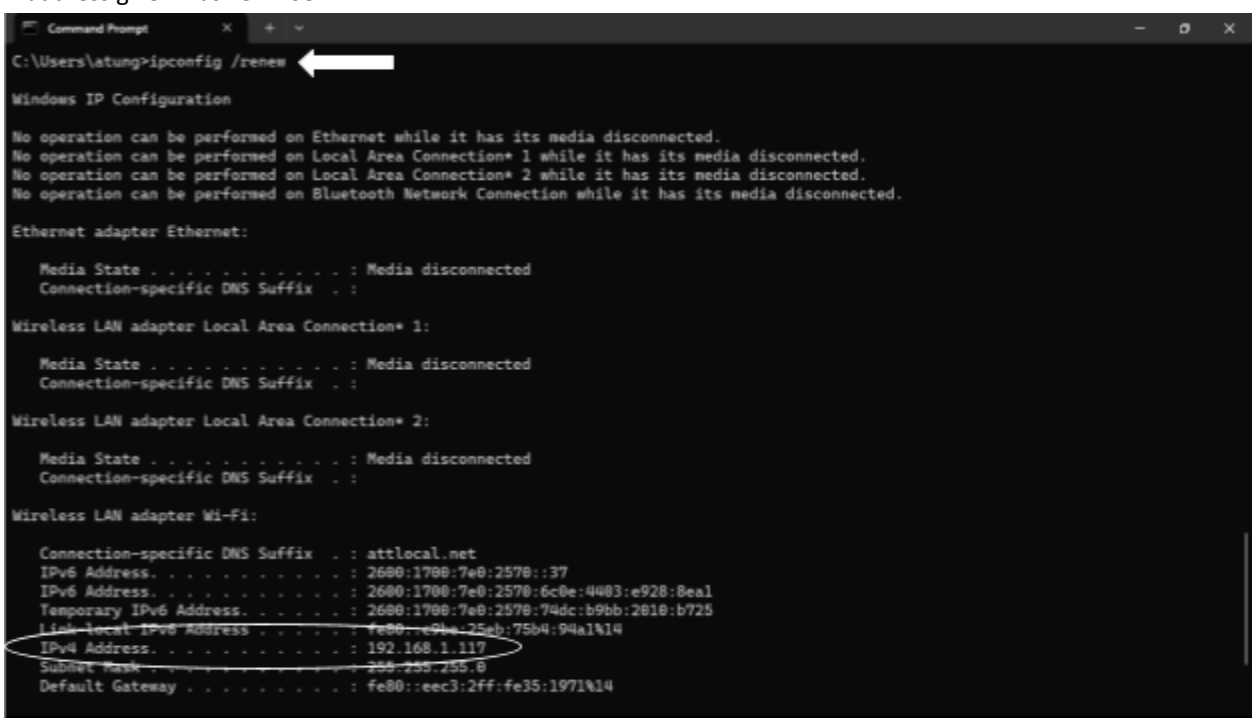
1. What frame number carried the DHCP request?

Frame 196



2. What IP address was given to you when you renewed your address?

IP address given was 192.168.1.117



3. Why do you see the IP address 0.0.0.0 ?
   1. I release my IP address that was assigned when I first connected to the network
      a. As a result I don't have an IP address so that why frame 178 says the source address is 0.0.0.0

2. My computer then requests an IP address but since it still doesn't have IP address assigned the source IP is still 0.0.0.0
    a. In the wireshark window it shows this transaction request after the offer in frame 196 but frame 195 is where the DHCP service offers me an IP of 192.168.1.117
3.