Lab Assignment 1

1. [5 marks] List up to five different protocols that appear in the protocol column in the unfiltered packet-listing window as shown when requesting http://gaia.cs.umass.edu/wireshark-labs/INTRO-wireshark-file1.html. As I don't have control over the data flowing over your network At the time of your lab, I don't know exactly how many and what protocols those will be. I do expect that you have a bunch (if less than 2, please look harder). Just list out those that you see, but don't bother to list more than 5.

Answer 1

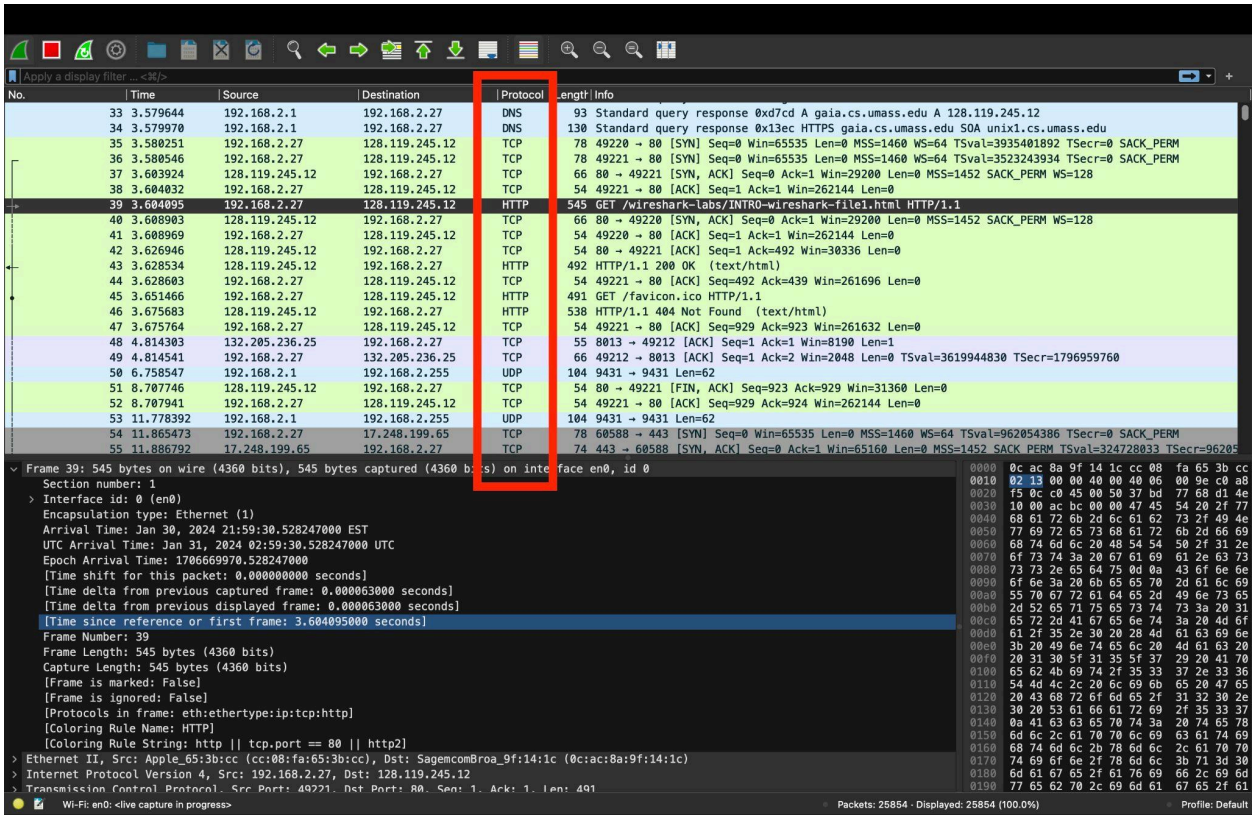

Fig 1

Different protocols that appear in the protocol column in the unfiltered packet-listing window are highlighted in red in Fig 1 above when requesting http://gaia.cs.umass.edu/wireshark-labs/INTRO-wireshark-file1.html.

Protocols are:-

1. **DNS**
2. **TCP**
3. **UDP**
4. **HTTP**

2. [10 marks] How long did it take from when the HTTP GET message was sent until the HTTP OK reply was received? (By default, the value of the Time column in the packet-listing window is the amount of time, in seconds, since Wireshark tracing began. (If you want to display the Time field in time-of-day format, select the Wireshark View pull down menu, then select Time Display Format, then select Time-of-day.)

Answer 2

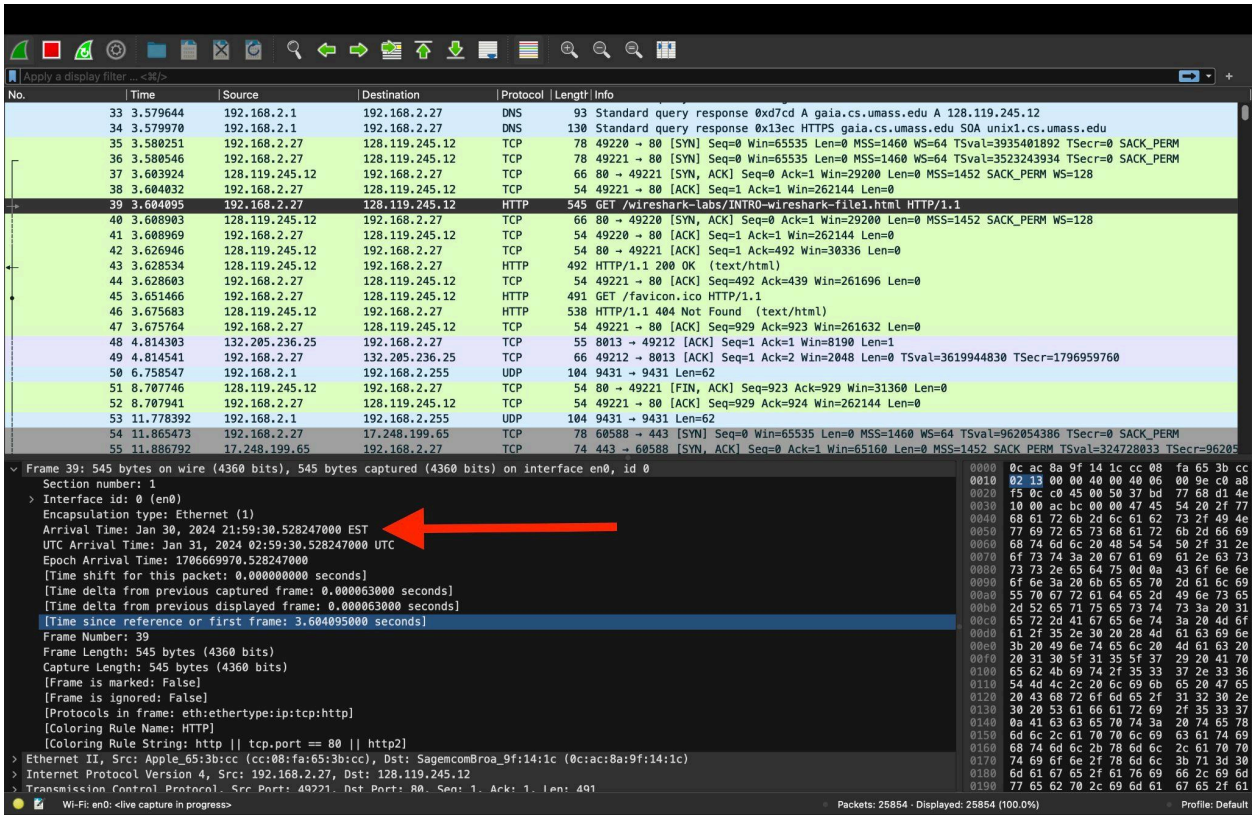

Fig 2

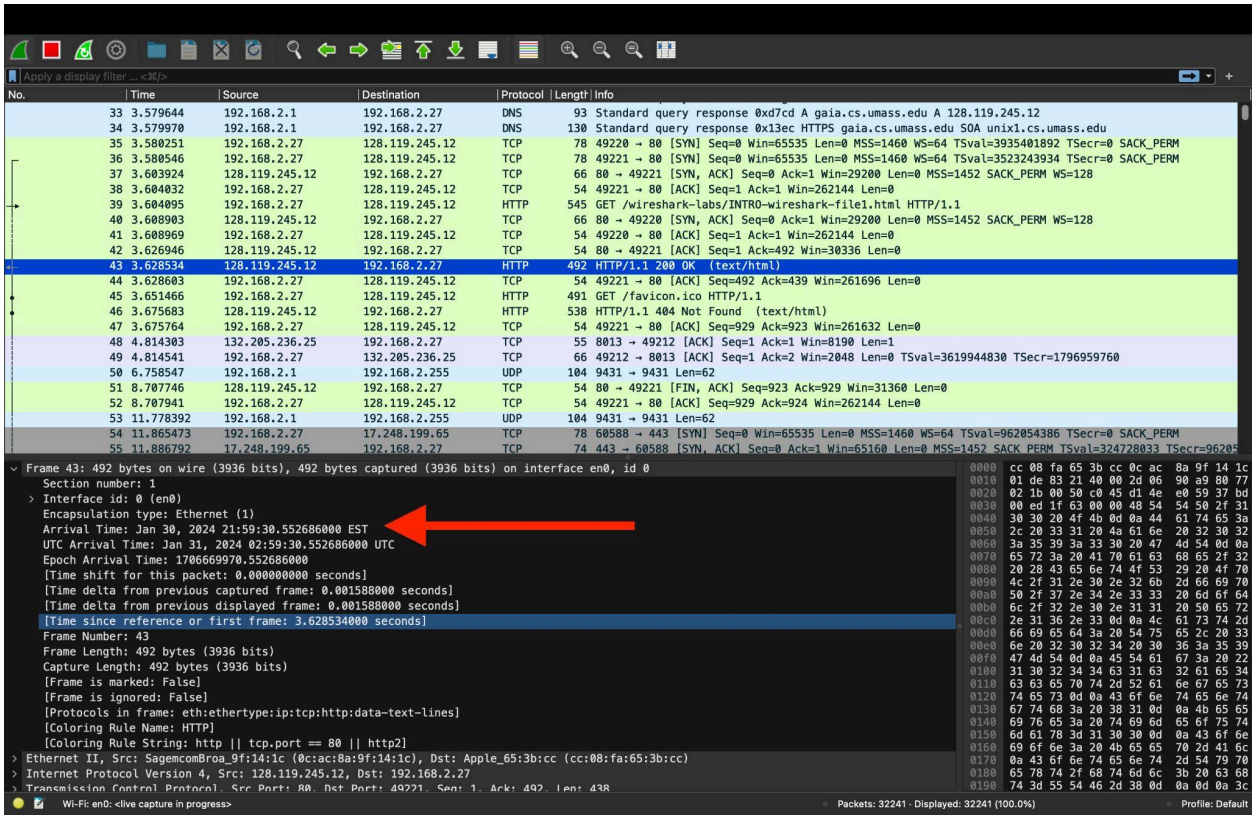

Fig 3

Looking at the frame section 39 of Fig 2 above of the GET request (highlighted in black), we see that the time, the packet arrived is 21:59:30.528247000 (highlighted with red arrow)

The same section 43 of Fig 3 above for the HTTP OK (highlighted in blue) shows an arrival time of 21:59:30.552686000 (highlighted with red arrow)

As hours, minutes of both the packets are the same so will compare the seconds of both the packets.

Difference of these 2 times = 30.552686000 - 30.528247000= 0.024439000 seconds

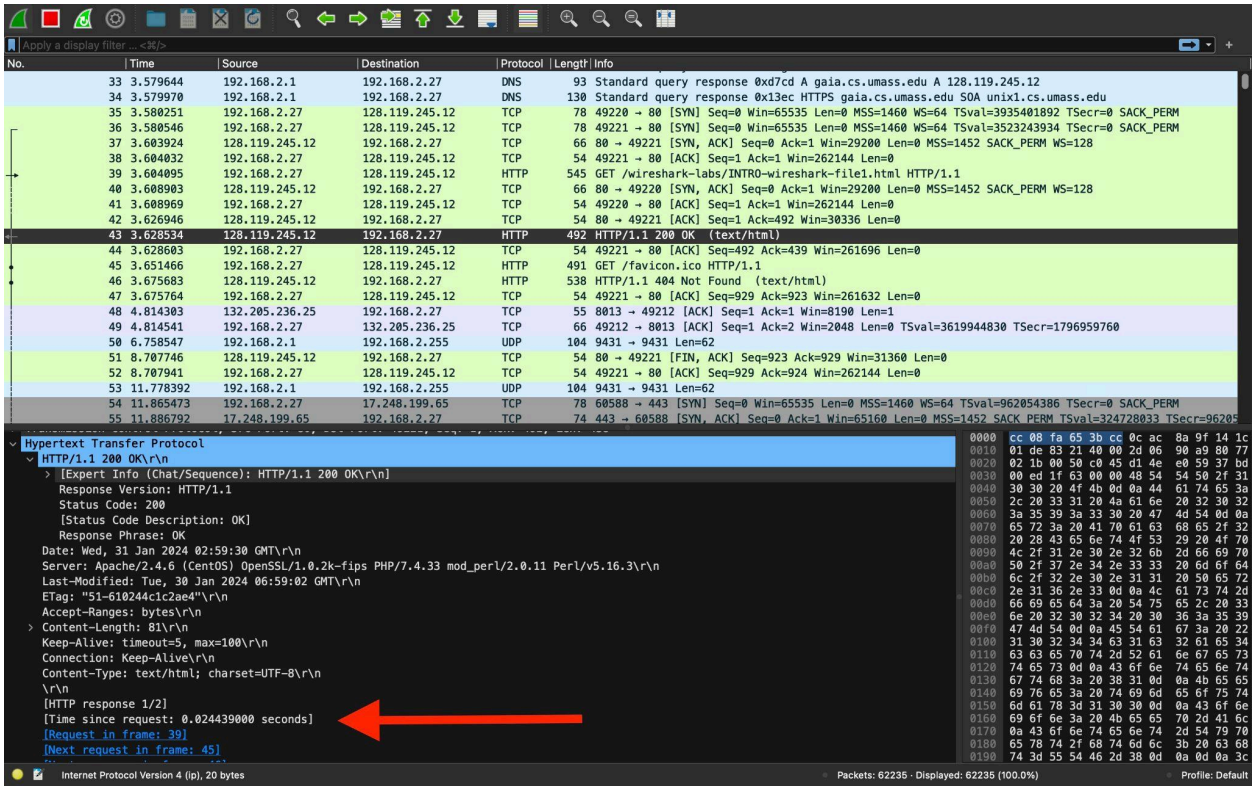

Fig 4

As we can see, the time since request in Hyper Transfer Protocol information for OK response is also the same as what we calculated above i.e. 0.024439000 seconds (highlighted in Fig 4 above with red arrow).

3. [5 marks] What is the Internet address of the gaia.cs.umass.edu? What is the Internet address of your computer (This might be a private address if you are behind a NAT device. No worries, we'll learn about that later) or (if you are using the trace file) the computer that sent the HTTP GET message? Include a screenshot and describe where you got the data to answer this question.

Answer 3

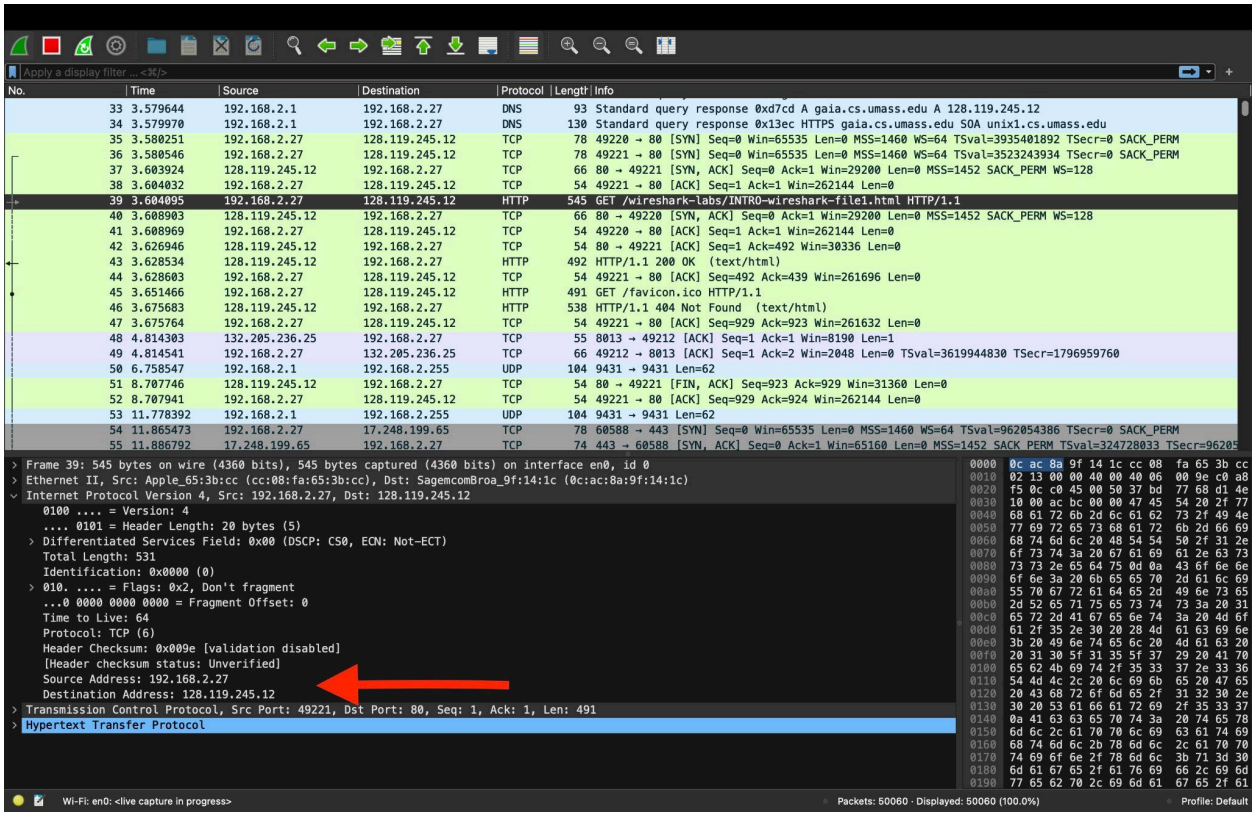

Fig 5

If we look at the IP section of the GET request in Fig 5(highlighted in black), the source and destination are shown (highlighted with red arrow)

The source is the local machine's address and the destination is the web server's public address.

My (local machine's) address = 192.168.2.27

IP address of gaia.cs.umass.edu = 128.19.245.12

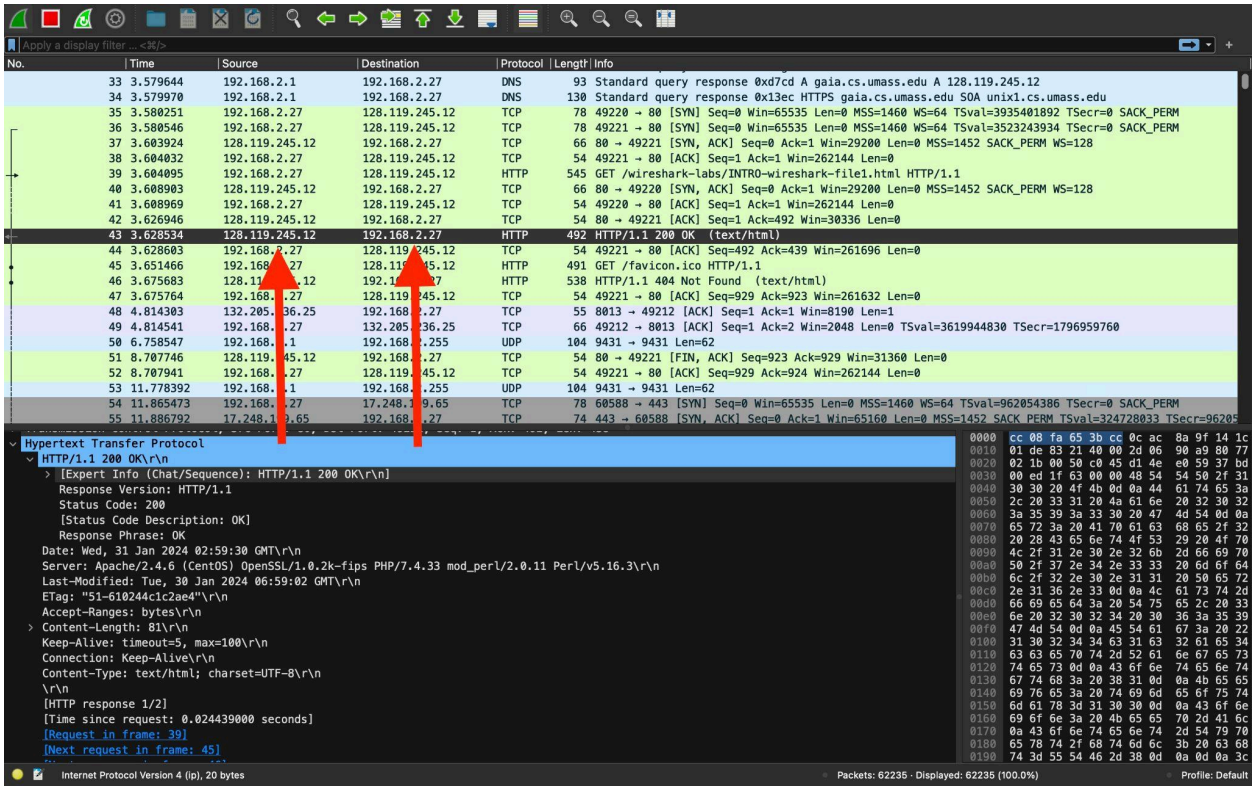

Fig 6

And If we look at the IP section of the OK response in Fig 6 (highlighted in black) also , the source and destination are shown.

The source is the web server's public address here and the destination is the local machine's address which is opposite to the Get message of the above packet (highlighted with red arrow).

4. [5 marks] Expand the information on the Transmission Control Protocol (TCP) for this packet (from question 3) in the Wireshark "Details of selected packet" window (see Figure 3 in the lab writeup) so you can see the fields in the TCP segment carrying the HTTP message. What is the destination port number (the number following "Dest Port:" for the TCP segment containing the HTTP request) to which this HTTP request is being sent? To answer this question, you'll need to select the TCP packet containing the HTTP GET request (hint: this is packet number 286). The purpose of this question is to familiarize you with using Wireshark's "Details of selected packet window"; see Figure 3. To do this, click on Packet 286 (your screen should look similar to Figure 3). To answer this question, then look in the "Details of selected packet" window toggle the triangle for HTTP (your screen should then look similar to Figure 5).

Answer 4

As we expand Transmission Control protocol information for frame 39 (as shown in Fig 7 below) , we can see that :-

Destination Port : 80 (highlighted with red arrow in Fig 7)

Information of Transmission Control Protocol (TCP) :

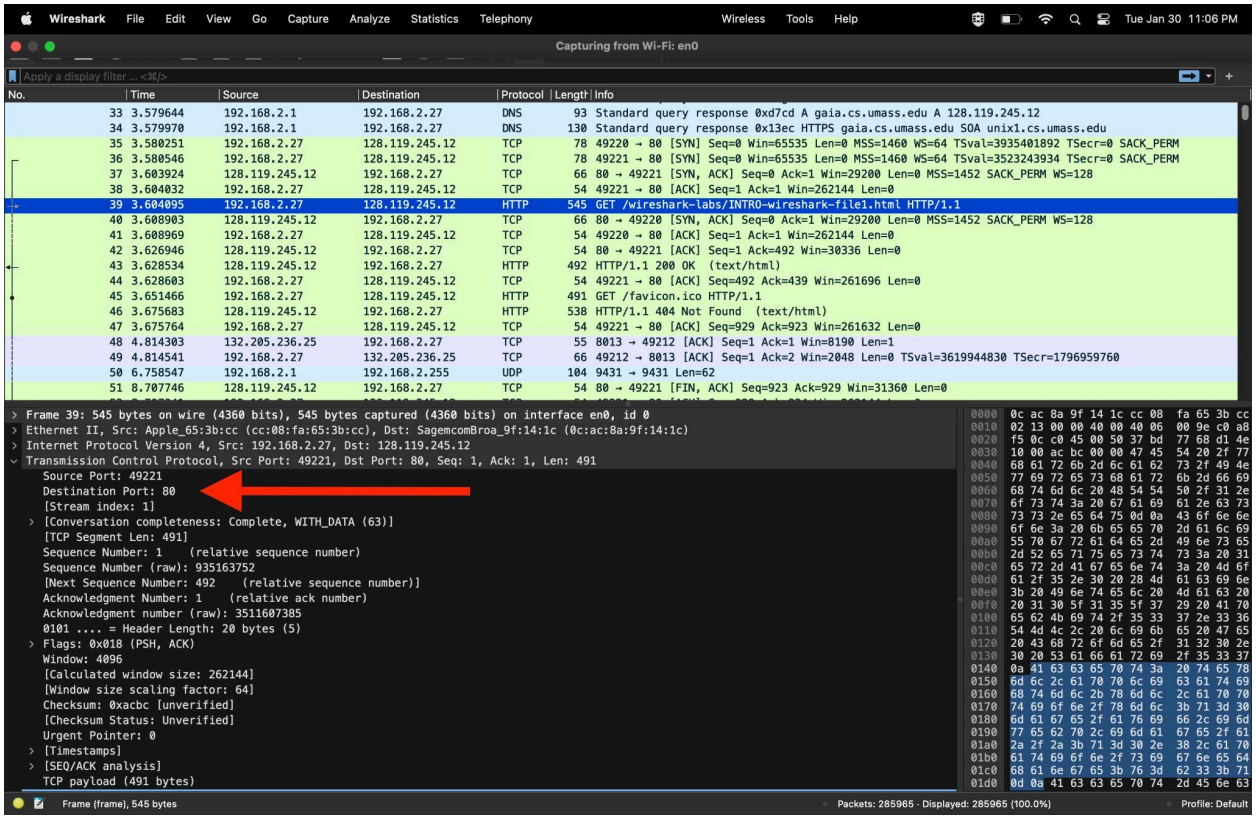

Fig 7

5. [20 marks] How many packets did you capture (total of all protocols, not just HTTP)? Now, use display filters to determine how many packets contain your ip address (hint: Use ip.addr instead of the clumsy ip.src or ip.dst format). What is this filter you used? Now, reverse the filter to determine how many packets don't contain your ip address. See any problems here? If not, you've already figured out the point of this question, so explain how you did so. If so, how can this problem be fixed? What are the appropriate display filters to use? How does Wireshark warn you of such a problem? (This is an important detail to remember about Wireshark. Please ensure you've discussed the problem well enough so that the marker can ensure you explore it thoroughly. If your numbers show you don't have a problem, then figure out how you might reverse the filter in such a way as to cause a problem.)

Answer 5

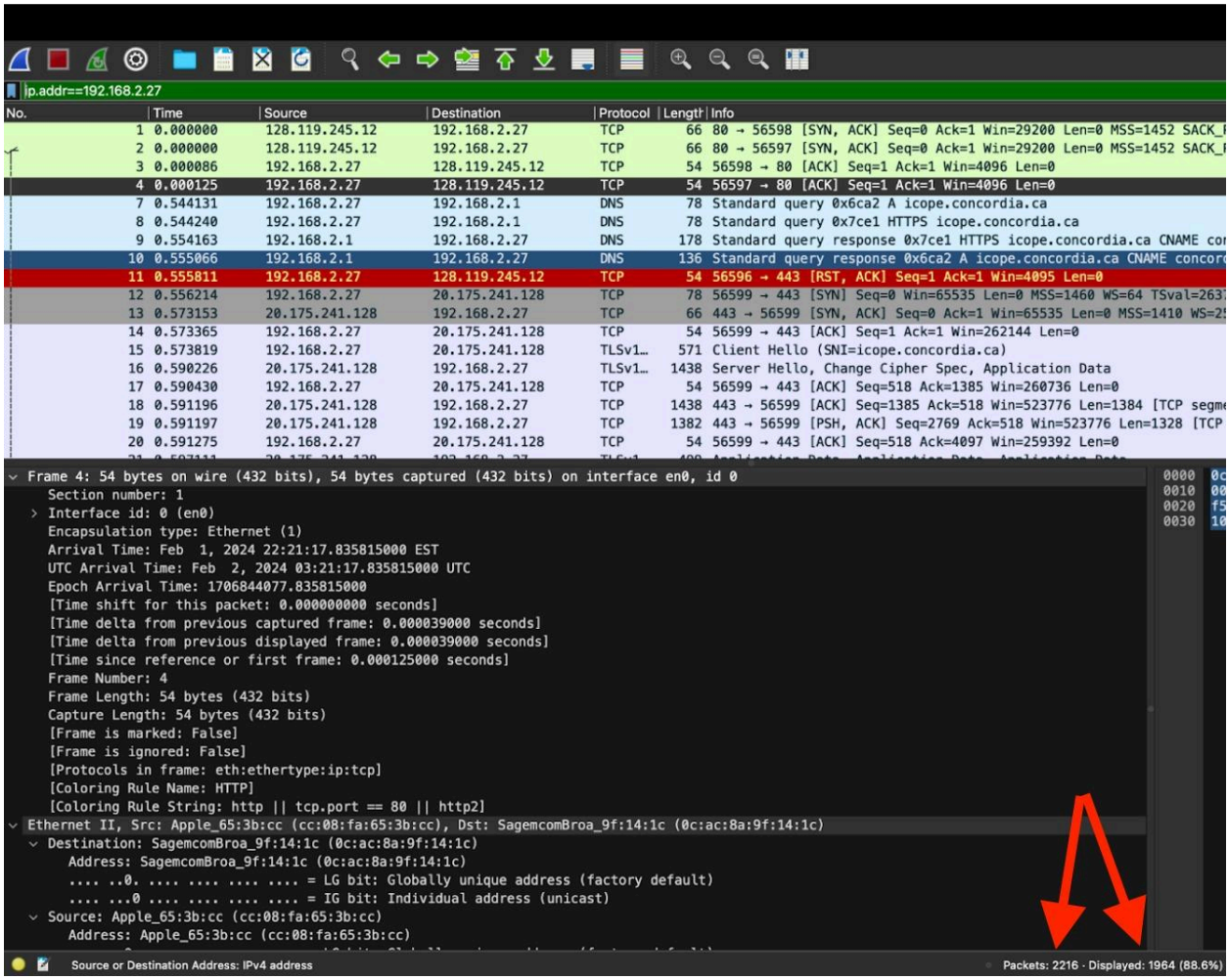

Fig 8

Total Packets captured(total of all protocols, not just HTTP)= 2216 (As it can be seen in the bottom status bar of the above Fig 8 having "packets" label and highlighted with red arrow)

Total packets containing my ip address i.e. 192.168.2.27 are :- 1964 (As it can be seen in the bottom status bar of the above Fig 8 having "Displayed" label and highlighted with red arrow)

Filter used for it is:- **ip.addr == 192.168.2.27**
(Ip address has been retrieved from command (ifconfig | grep "inet " | grep -v 127.0.0.1) by running on the terminal.)

Ideally number of packets that don't contain my ip address i.e. 192.168.2.27 should be :-
Total Packets Displayed - Total Packets containing my ip address = 2216 - 1964 = 252

Filter that causes problem :-

To filter out any traffic to or from 192.168.2.27 We might try the following filter :

ip.addr != 192.168.2.27

which is equivalent to

ip.src != 192.168.2.27 or ip.dst != 192.168.2.27

This translates to "pass all traffic except for traffic with a source address of 192.168.2.27 and a destination address of 192.168.2.27", which isn't what we wanted. That's why the total number of packets displayed are 222(as shown in Fig 9 below, highlighted with red arrow) and not 252 which we wanted.

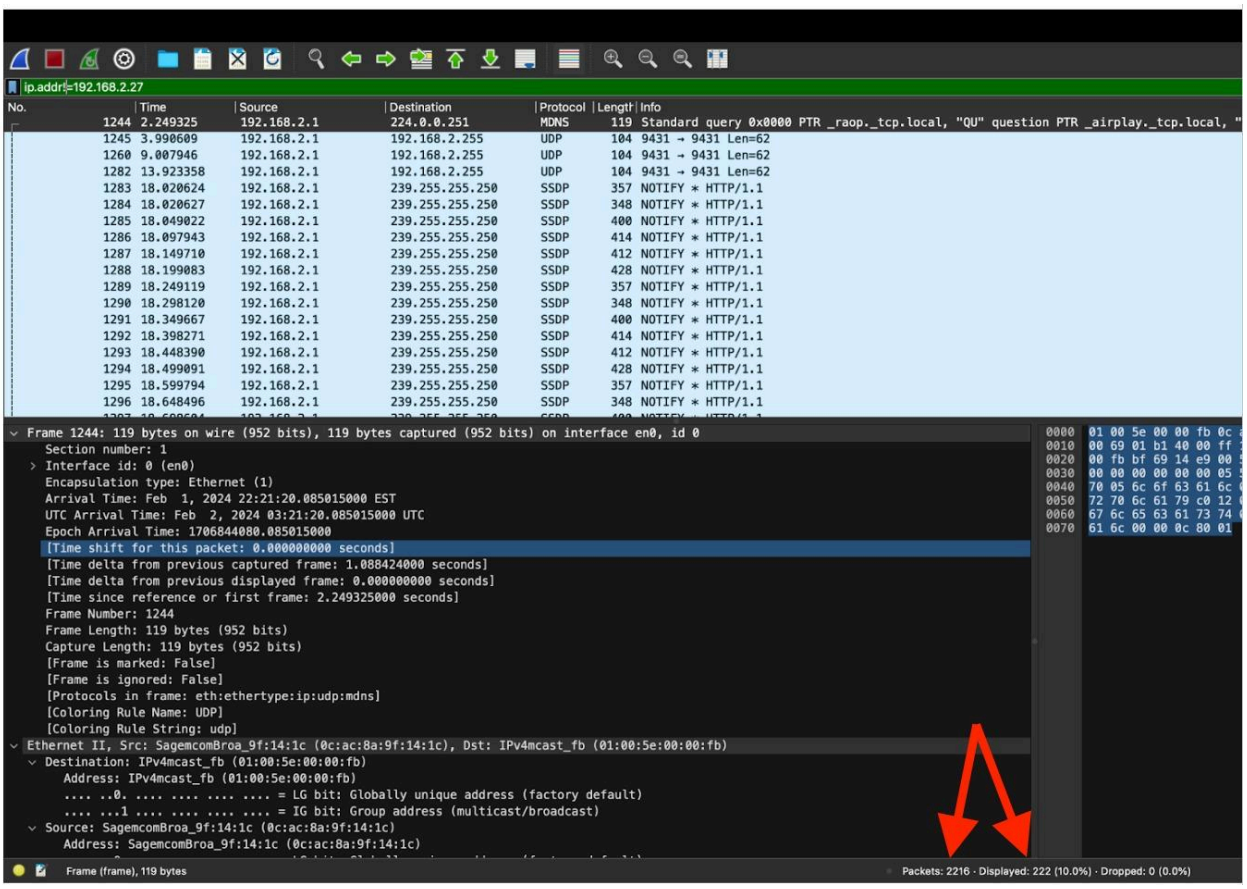

Fig 9

Appropriate display filters and fixing the problem:

Instead we need to negate the expression, like so:

Filter used for it :- **! (ip.addr == 192.168.2.27)**

which is equivalent to

! (ip.src == 192.168.2.27 or ip.dst == 192.168.2.27)

This translates to "pass any traffic except with a source address of 192.168.2.27 or a destination address of 192.168.2.27", which is what we wanted. Now, we get a total number of packets 252 (highlighted in Fig 10 below with a red arrow at the bottom status bar).

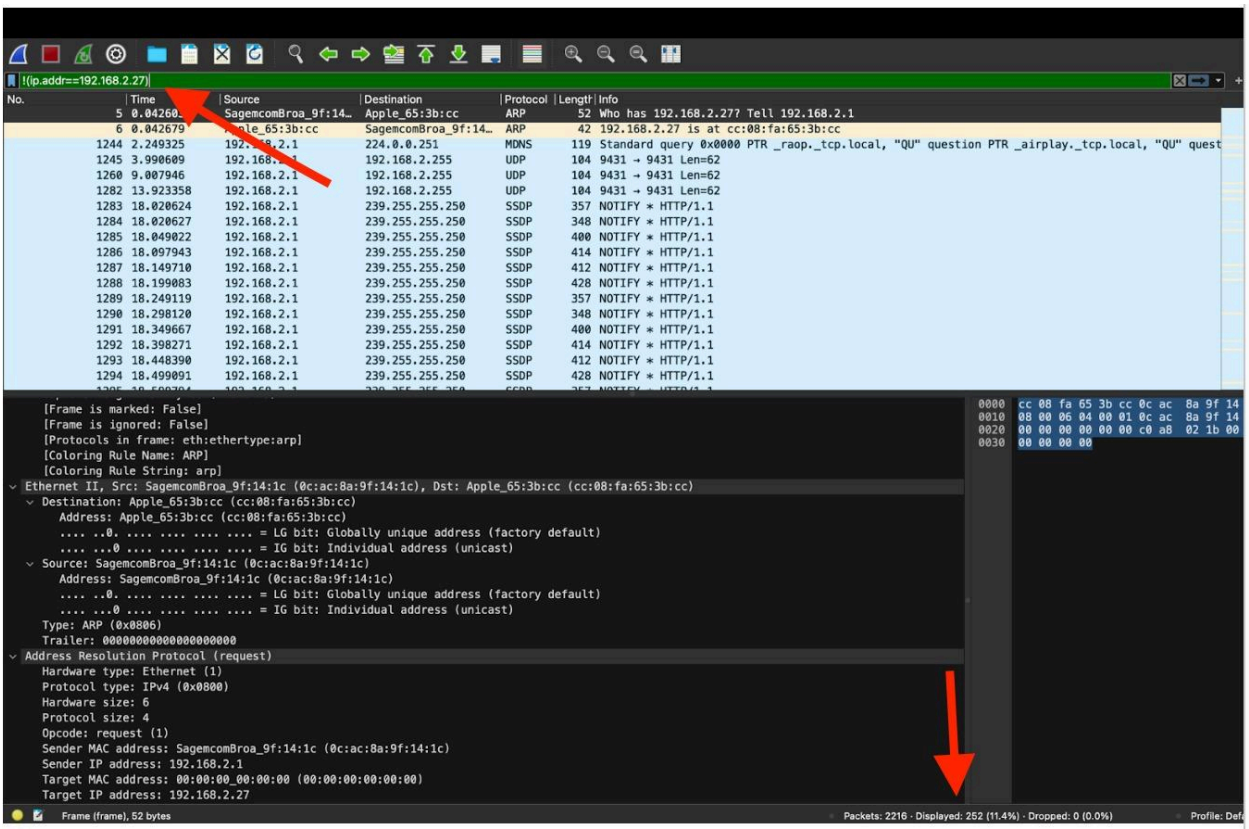

Fig 10

Wireshark warning:

Wireshark does not provide explicit warnings after version 4.0 for potential filter issues. It is important for the user to now understand the filtering logic and ensure that it accurately captures the desired traffic.

6. [5 marks] Print the two HTTP messages (GET and OK) referred to in question 2 above.

Answer 6

For Printing, Select the Http Get and expand the Hypertext protocol details of selected packets and click on file and then print, Http message will be printed as visible in below Fig 11.

HTTP GET Message

/var/folders/rn/tq8mn2ss4ms_v77dr3zw4fsc0000gn/T/wireshark_Wi-FiKPI5H2.pcapng 184725 total packets, 184725 shown

```
No.     Time           Source                Destination           Protocol Length Info
     39 3.604095       192.168.2.27          128.119.245.12        HTTP     545    GET /wireshark-
labs/INTRO-wireshark-file1.html HTTP/1.1
Frame 39: 545 bytes on wire (4360 bits), 545 bytes captured (4360 bits) on interface en0, id 0
Ethernet II, Src: Apple_65:3b:cc (cc:08:fa:65:3b:cc), Dst: SagemcomBroa_9f:14:1c (0c:ac:8a:9f:
14:1c)
Internet Protocol Version 4, Src: 192.168.2.27, Dst: 128.119.245.12
Transmission Control Protocol, Src Port: 49221, Dst Port: 80, Seq: 1, Ack: 1, Len: 491
Hypertext Transfer Protocol
    GET /wireshark-labs/INTRO-wireshark-file1.html HTTP/1.1\r\n
    Host: gaia.cs.umass.edu\r\n
    Connection: keep-alive\r\n
    Upgrade-Insecure-Requests: 1\r\n
    User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/120.0.0.0 Safari/537.36\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/
apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7\r\n
    Accept-Encoding: gzip, deflate\r\n
    Accept-Language: en-GB,en-US;q=0.9,en;q=0.8\r\n
    \r\n
    [Full request URI: http://gaia.cs.umass.edu/wireshark-labs/INTRO-wireshark-file1.html]
    [HTTP request 1/2]
    [Response in frame: 43]
    [Next request in frame: 45]
```

Fig 11

HTTP OK Message

For Printing, Select the Http OK and expand the Hypertext protocol details of selected packets and click on file and then print, Http message will be printed as visible in below Fig 12.

/var/folders/rn/tq8mn2ss4ms_v77dr3zw4fsc0000gn/T/wireshark_Wi-FiKPI5H2.pcapng 184949 total packets, 184949 shown

```
No.     Time           Source                Destination           Protocol Length Info
     43 3.628534       128.119.245.12        192.168.2.27          HTTP     492    HTTP/1.1 200 OK
(text/html)
Frame 43: 492 bytes on wire (3936 bits), 492 bytes captured (3936 bits) on interface en0, id 0
Ethernet II, Src: SagemcomBroa_9f:14:1c (0c:ac:8a:9f:14:1c), Dst: Apple_65:3b:cc (cc:08:fa:
65:3b:cc)
Internet Protocol Version 4, Src: 128.119.245.12, Dst: 192.168.2.27
Transmission Control Protocol, Src Port: 80, Dst Port: 49221, Seq: 1, Ack: 492, Len: 438
Hypertext Transfer Protocol
    HTTP/1.1 200 OK\r\n
    Date: Wed, 31 Jan 2024 02:59:30 GMT\r\n
    Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/7.4.33 mod_perl/2.0.11 Perl/v5.16.3\r\n
    Last-Modified: Tue, 30 Jan 2024 06:59:02 GMT\r\n
    ETag: "51-610244c1c2ae4"\r\n
    Accept-Ranges: bytes\r\n
    Content-Length: 81\r\n
    Keep-Alive: timeout=5, max=100\r\n
    Connection: Keep-Alive\r\n
    Content-Type: text/html; charset=UTF-8\r\n
    \r\n
    [HTTP response 1/2]
    [Time since request: 0.024439000 seconds]
    [Request in frame: 39]
    [Next request in frame: 45]
    [Next response in frame: 46]
    [Request URI: http://gaia.cs.umass.edu/wireshark-labs/INTRO-wireshark-file1.html]
    File Data: 81 bytes
Line-based text data: text/html (3 lines)
```

Fig 12