1. [2 marks] Is your browser running HTTP version 1.0, 1.1, 2 or 3? What version of HTTP is the server running?

Answer 1

As seen in Fig 1, the browser is running HTTP 1.1 and in Fig 2, the server is also running HTTP 1.1.

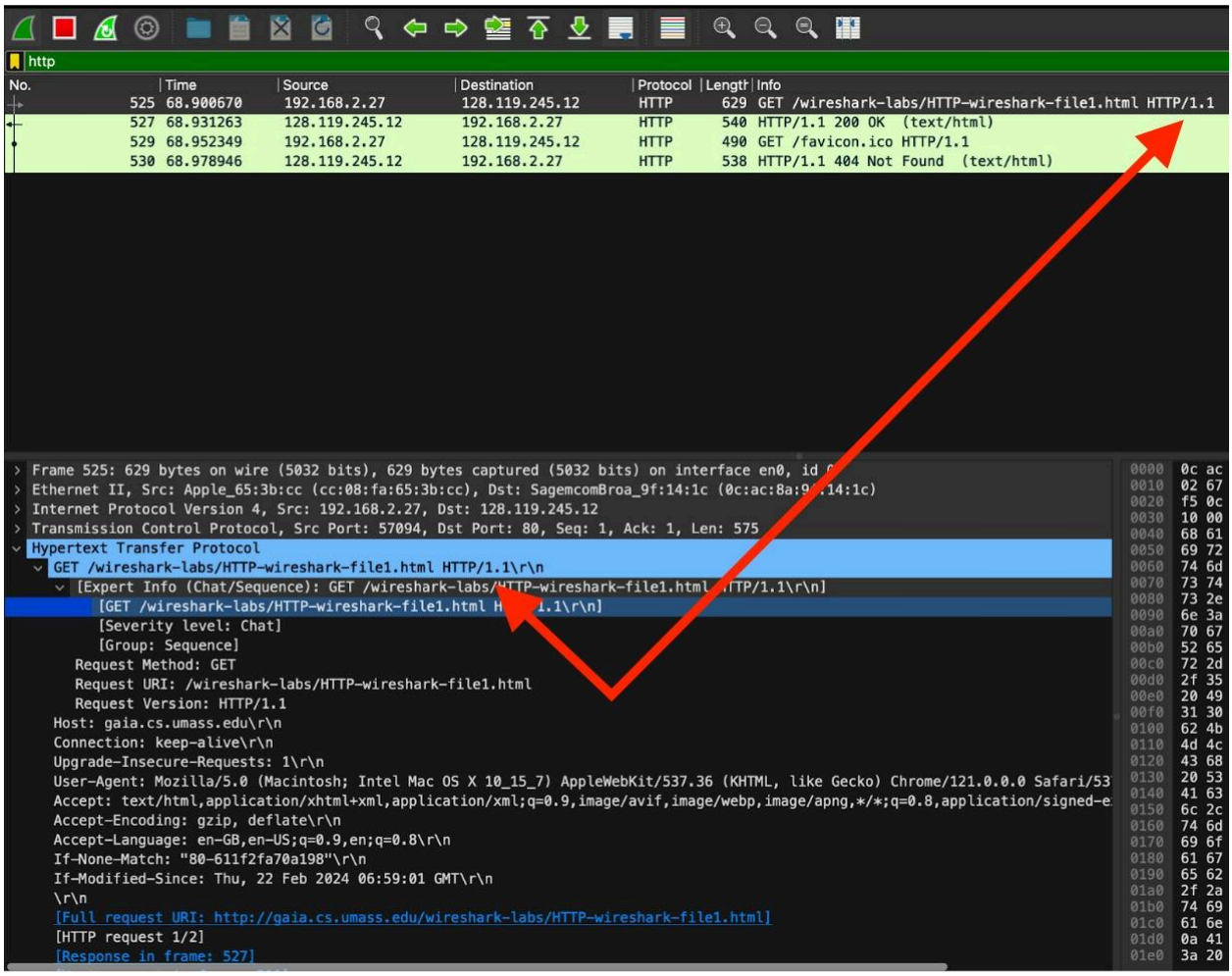

Fig 1

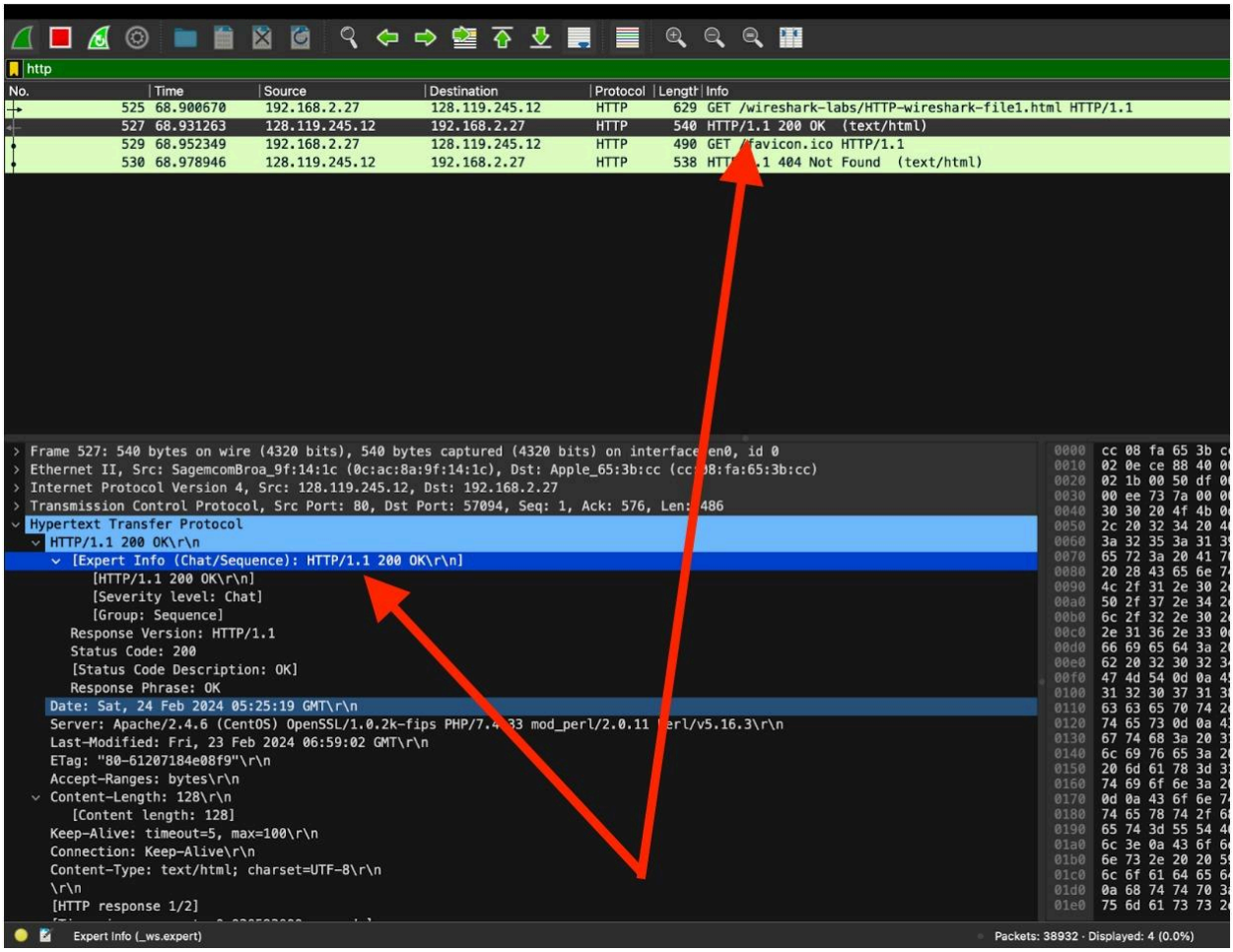

Fig 2

2. [2 marks] What languages (if any) does your browser indicate that it can accept to the server?

Answer 2

As seen in HyperText Transfer Protocol section of Fig 3, Accept-Language: en-Gb, en-US

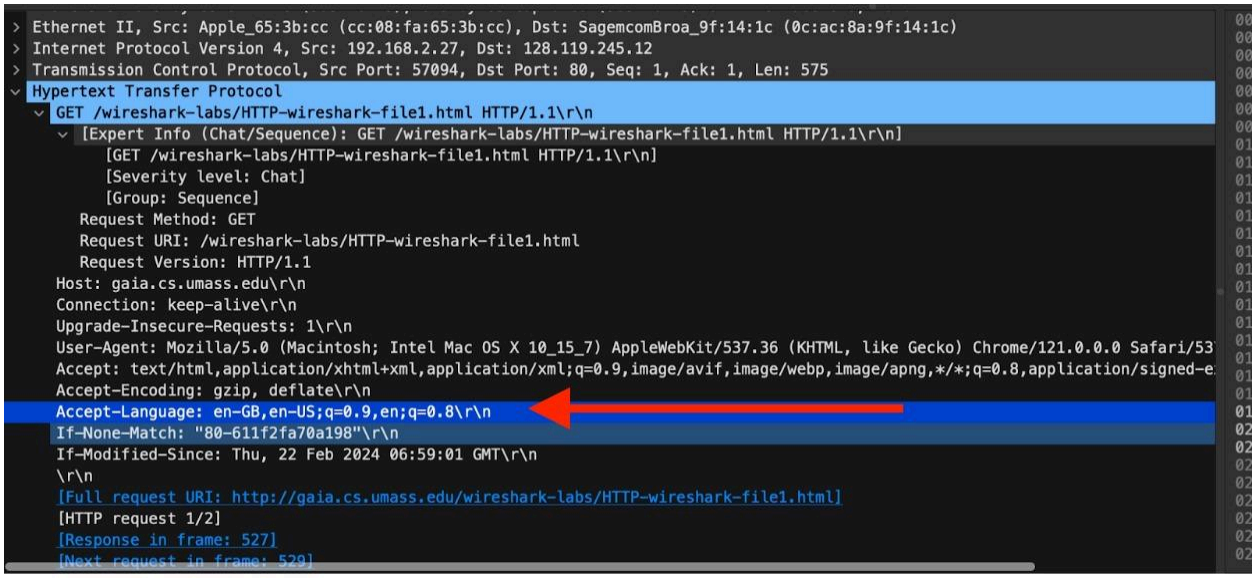

Fig 3

3. [2 marks] What was the round-trip time for the request (i.e. time between sending the request and capturing the response)?

Answer 3

According to fig 4, Round Trip Time for the request is :- 0.0305 seconds (Noted from time since request field)

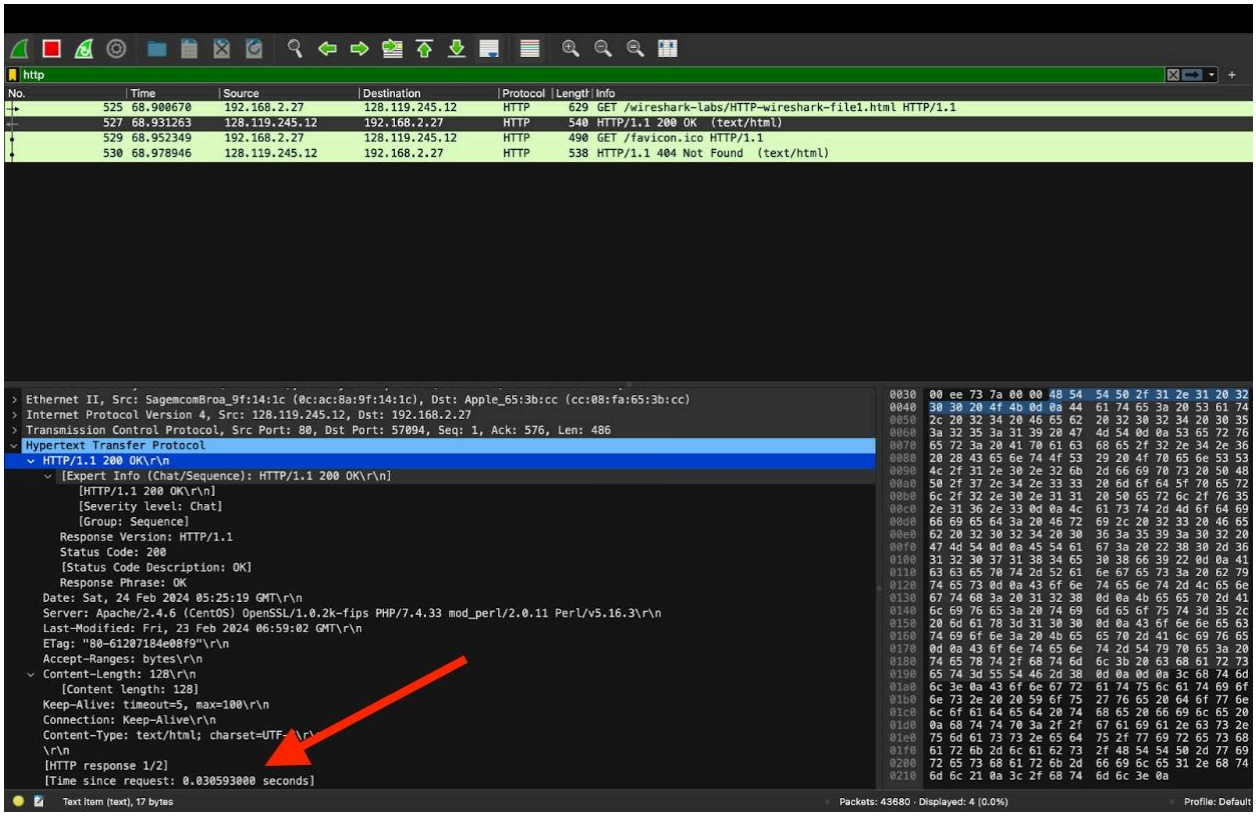

Fig 4

4. [2 marks] What status code is returned to your browser from the server?

Answer 4

As it can be seen by red arrow in HyperText Transfer Protocol section of fig 5 , Status Code is HTTP/1.1 200 OK (text/html)

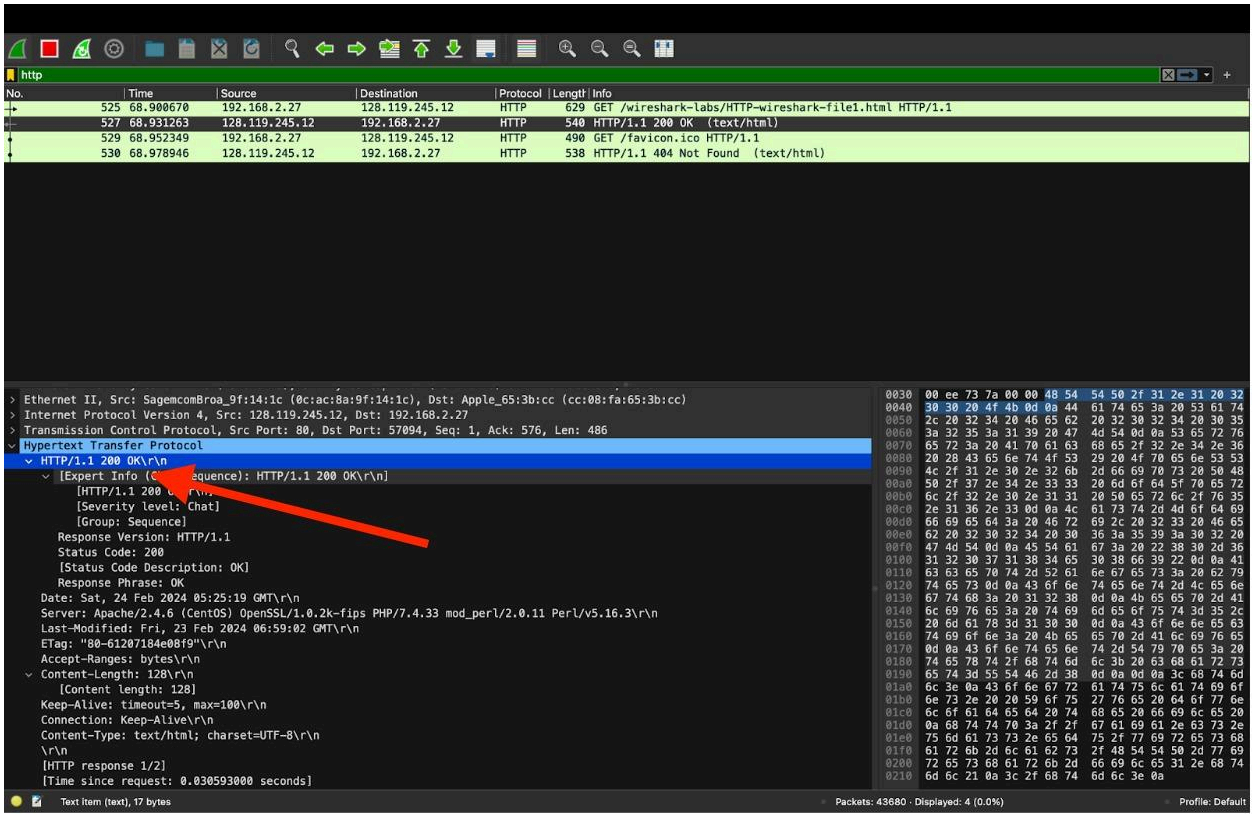

Fig 5

5. [2 marks] When was the HTML file you are retrieving last modified at the server?

Answer 5

Last-Modified: Fri, 23 Feb 2024 06:59:02 GMT, as it can be seen in below fig 6 by red arrow.

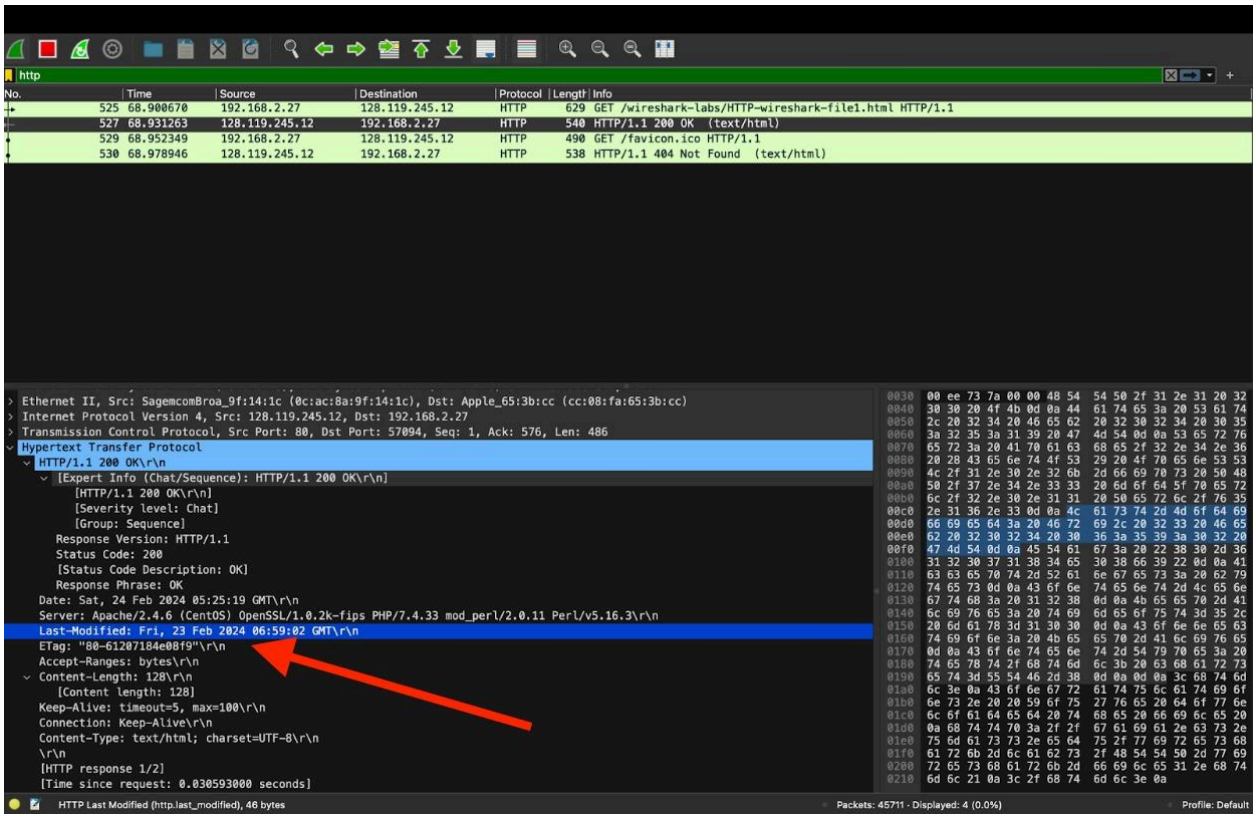

Fig 6

6. [2 marks] How many bytes of content are being returned to your browser?

Answer 6

Content Length :- 128 as it can be seen in below fig 7 by red arrow.

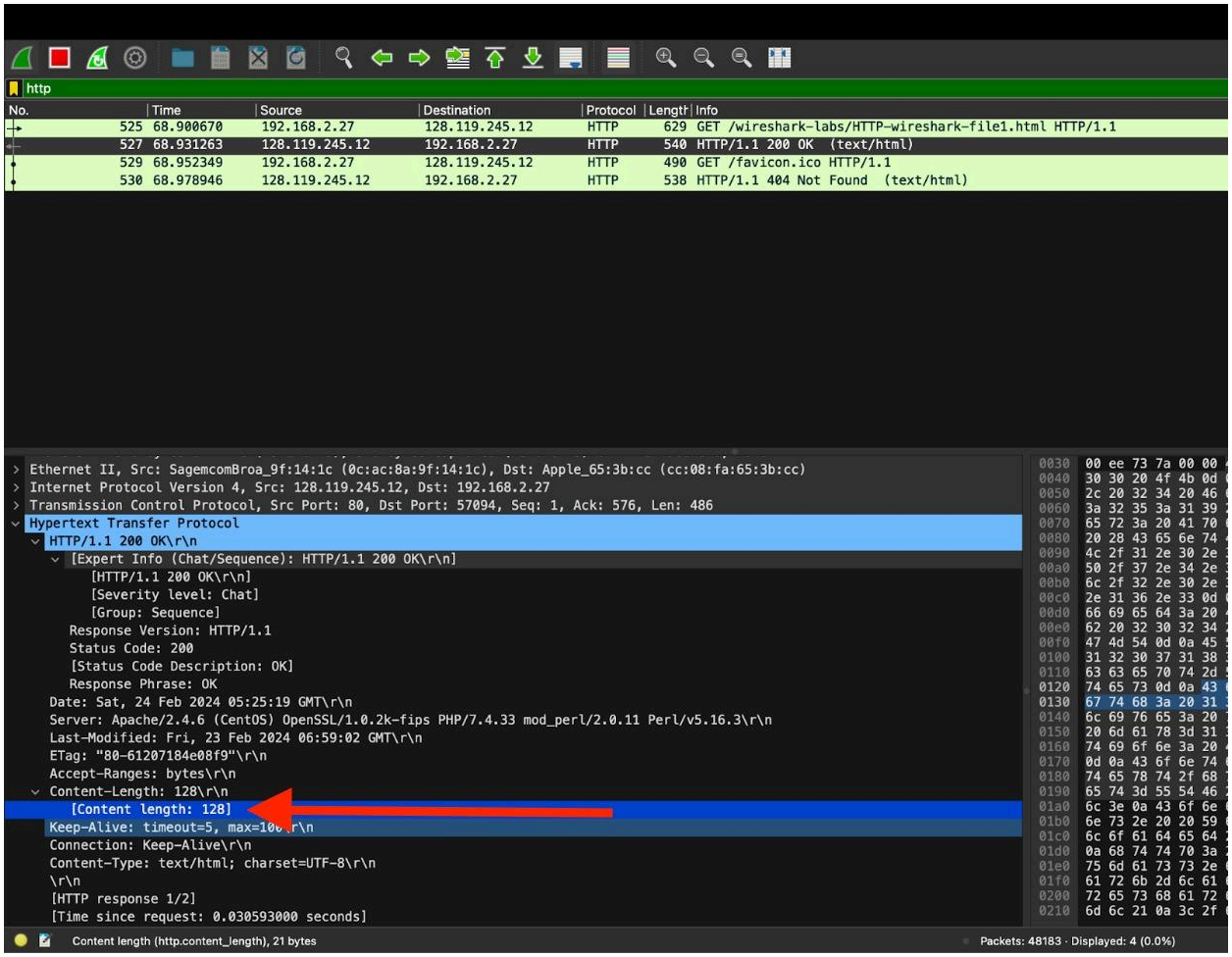

Fig 7

7. [2 marks] By inspecting the raw data in the packet content window, do you see any headers within the data that are not displayed in the packet-listing window? If so, name one.

Answer 7

No, all of the headers can be found in the raw data as it can be seen in below figures 8 and 9.

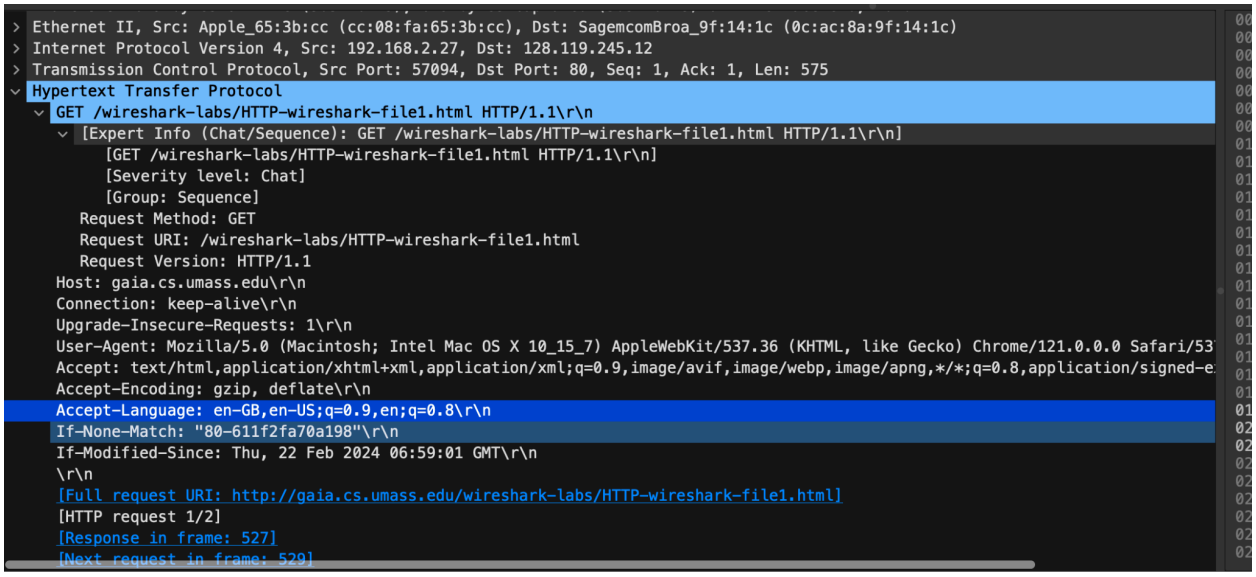
Ethernet II, Src: Apple_65:3b:cc (cc:08:fa:65:3b:cc), Dst: SagemcomBroa_9f:14:1c (0c:ac:8a:9f:14:1c)
Internet Protocol Version 4, Src: 192.168.2.27, Dst: 128.119.245.12
Transmission Control Protocol, Src Port: 57094, Dst Port: 80, Seq: 1, Ack: 1, Len: 575
Hypertext Transfer Protocol
GET /wireshark-labs/HTTP-wireshark-file1.html HTTP/1.1\r\n
[Expert Info (Chat/Sequence): GET /wireshark-labs/HTTP-wireshark-file1.html HTTP/1.1\r\n]
[GET /wireshark-labs/HTTP-wireshark-file1.html HTTP/1.1\r\n]
[Severity level: Chat]
[Group: Sequence]
Request Method: GET
Request URI: /wireshark-labs/HTTP-wireshark-file1.html
Request Version: HTTP/1.1
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/121.0.0.0 Safari/53
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-e
Accept-Encoding: gzip, deflate\r\n
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8\r\n
If-None-Match: "80-611f2fa70a198"\r\n
If-Modified-Since: Thu, 22 Feb 2024 06:59:01 GMT\r\n
\r\n
[Full request URI: http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file1.html]
[HTTP request 1/2]
[Response in frame: 527]
[Next request in frame: 529]

Fig 8

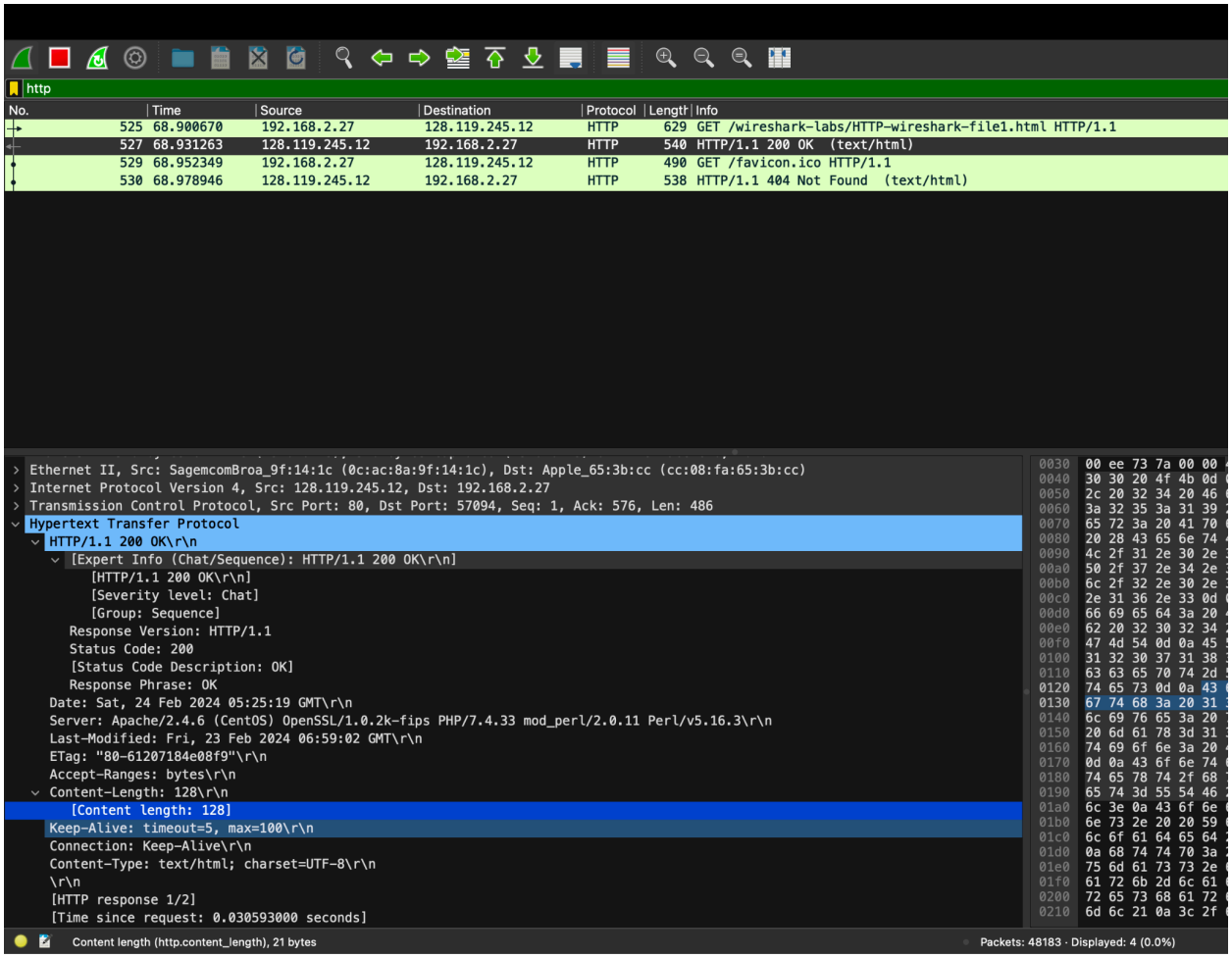
http
No. Time Source Destination Protocol Length Info
525 68.900670 192.168.2.27 128.119.245.12 HTTP 629 GET /wireshark-labs/HTTP-wireshark-file1.html HTTP/1.1
527 68.931263 128.119.245.12 192.168.2.27 HTTP 540 HTTP/1.1 200 OK (text/html)
529 68.952349 192.168.2.27 128.119.245.12 HTTP 490 GET /favicon.ico HTTP/1.1
530 68.978946 128.119.245.12 192.168.2.27 HTTP 538 HTTP/1.1 404 Not Found (text/html)
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: 57094, Seq: 1, Ack: 576, Len: 486
Hypertext Transfer Protocol
HTTP/1.1 200 OK\r\n
[Expert Info (Chat/Sequence): HTTP/1.1 200 OK\r\n]
[HTTP/1.1 200 OK\r\n]
[Severity level: Chat]
[Group: Sequence]
Response Version: HTTP/1.1
Status Code: 200
[Status Code Description: OK]
Response Phrase: OK
Date: Sat, 24 Feb 2024 05:25:19 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: Fri, 23 Feb 2024 06:59:02 GMT\r\n
ETag: "80-61207184e08f9"\r\n
Accept-Ranges: bytes\r\n
Content-Length: 128\r\n
[Content length: 128]
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: 0.030593000 seconds]
Content length (http.content_length), 21 bytes
Packets: 48183 · Displayed: 4 (0.0%)

Fig 9

8. [2 marks] Inspect the contents of the first HTTP GET request from your browser to the server.
Do you see an “IF-MODIFIED-SINCE” line in the HTTP GET?

Answer 8

No, there is no “IF-MODIFIED-SINCE” line in the first HTTP GET as it can be seen in fig 10. But it is there in the second HTTP GET as it can be seen in fig 11.

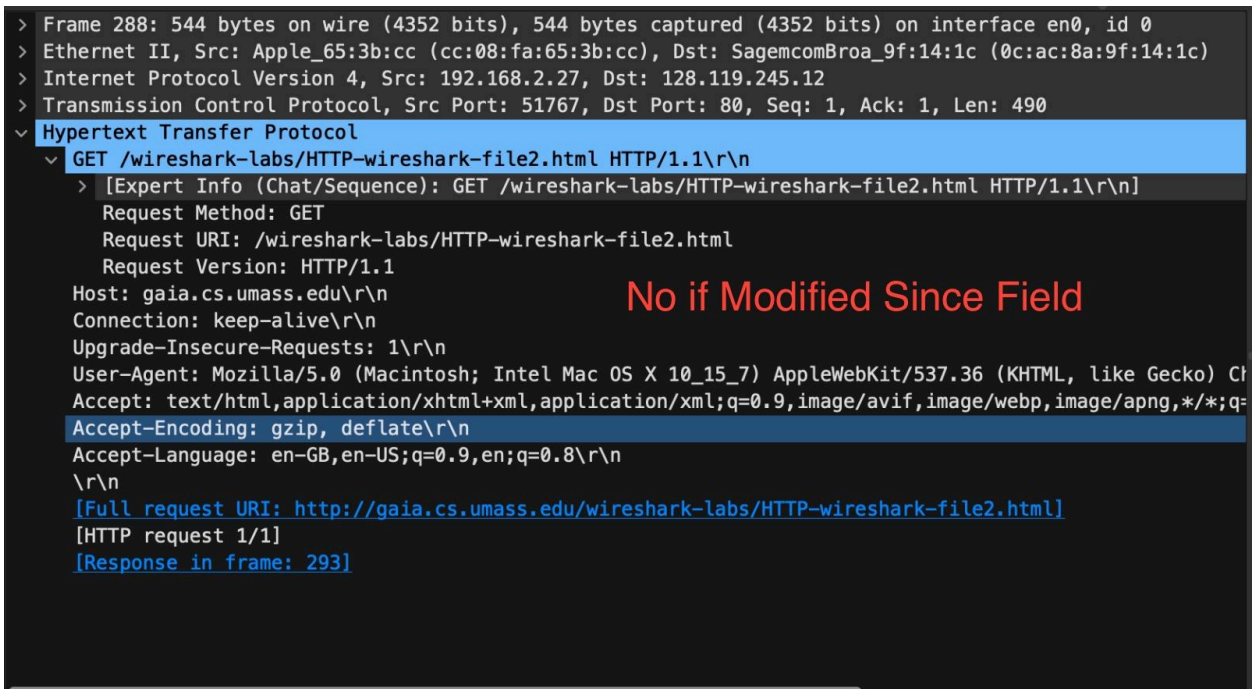

Fig 10

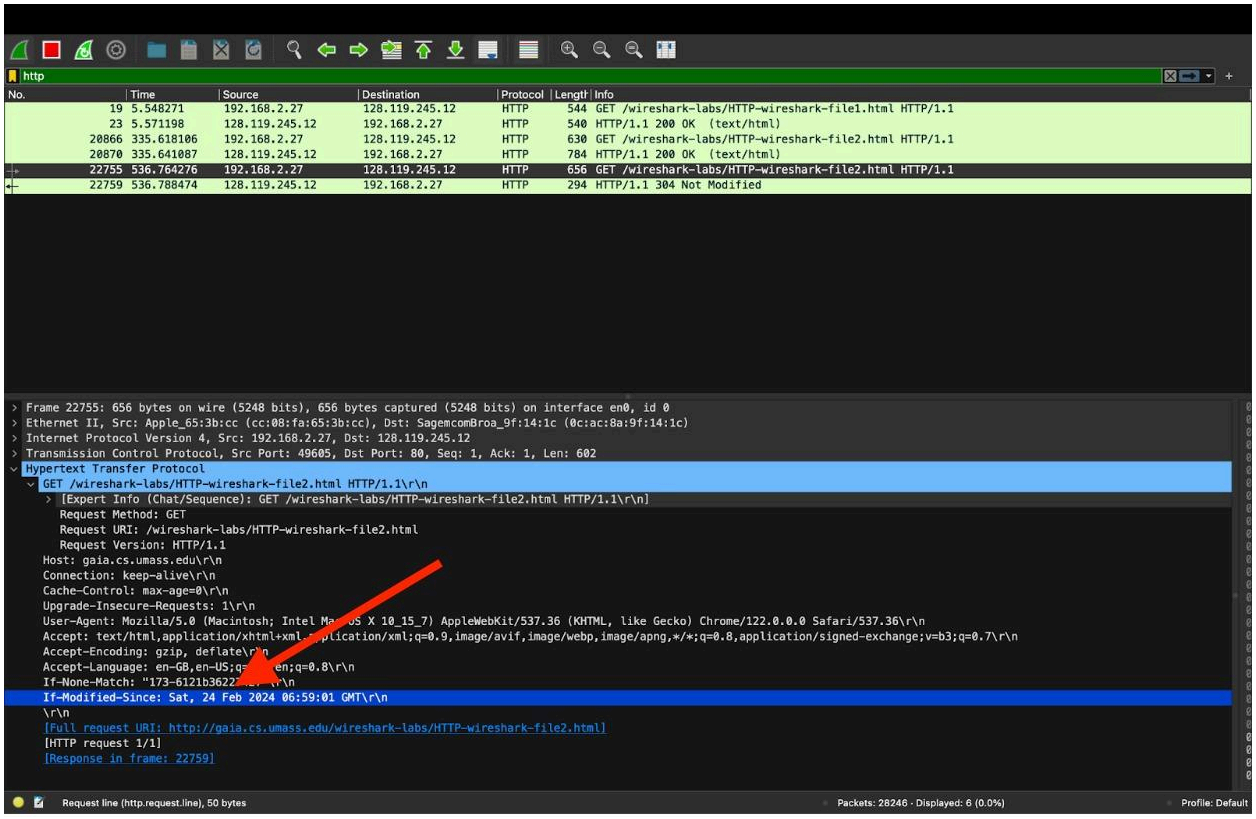

Fig 11

9. [4 marks] Inspect the contents of the server response. Did the server explicitly return the contents of the file? How can you tell?

Answer 9

The server explicitly returns the contents of the file in the first HTTP Get response, We can see the contents in the Line-based text data field in fig 12.

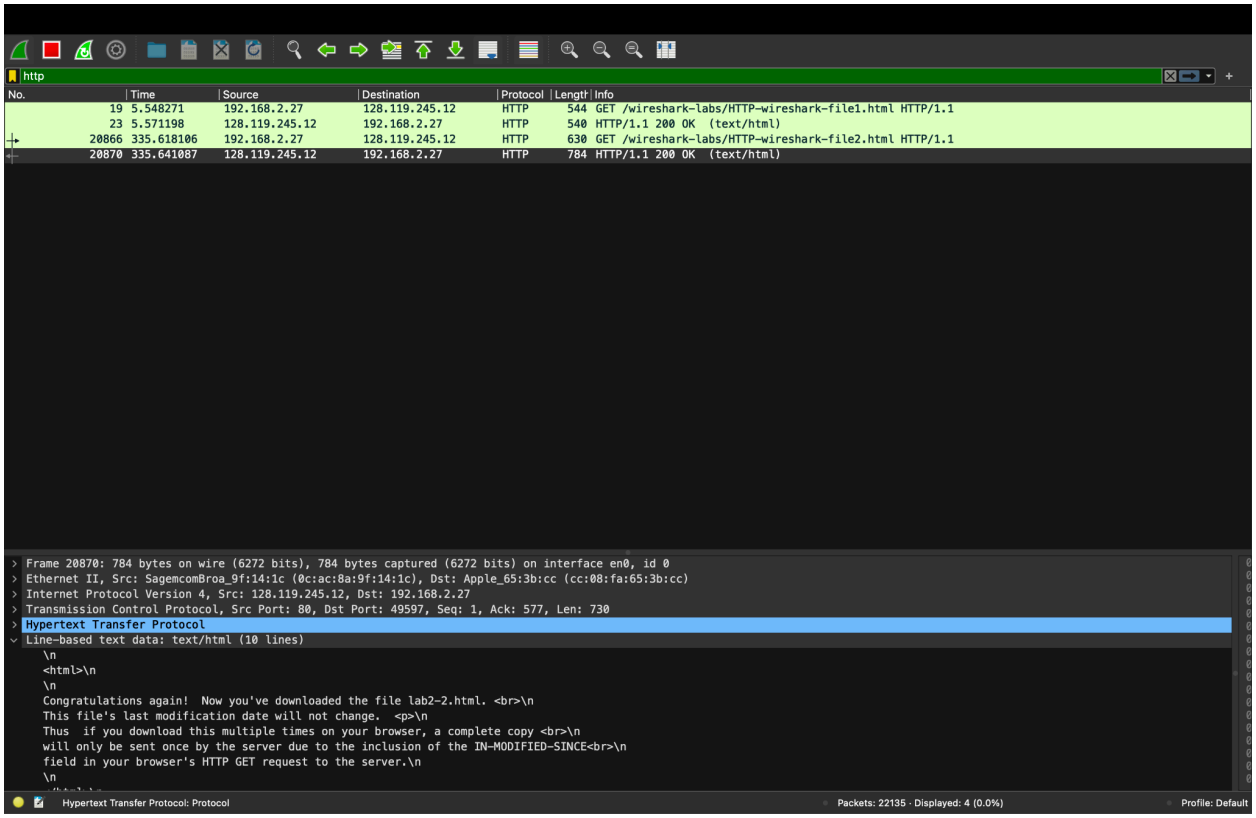

Fig 12

But according to Fig 13, the server did not explicitly return the contents of the file in the second HTTP Get response since the file had not been modified.

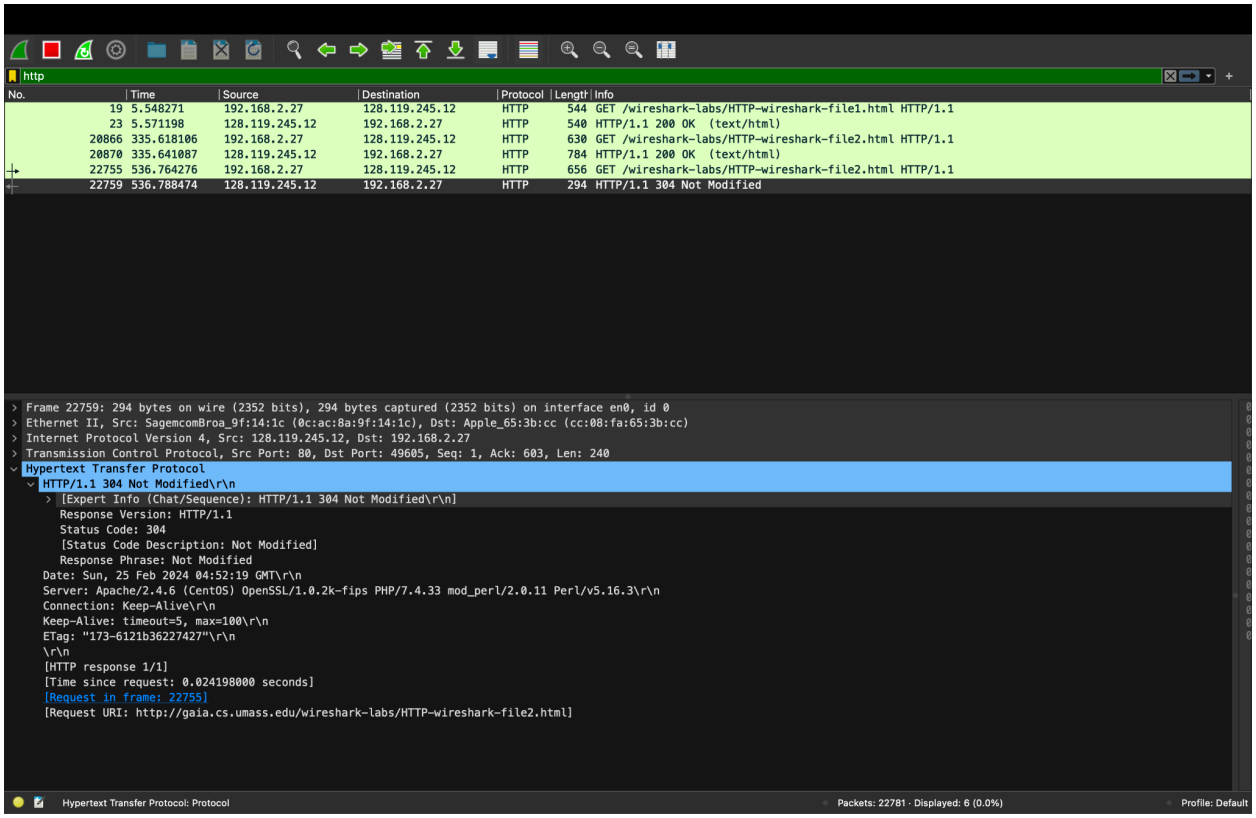

Fig 13

10. [2 marks] Now inspect the contents of the second HTTP GET request from your browser to the server. Do you see an “IF-MODIFIED-SINCE:” line in the HTTP GET?3 If so, what information follows the “IF-MODIFIED-SINCE:” header?

Answer 10

Yes. The information followed is: **Sat, 24 Feb 2024 06:59:01 GMT\r\n** as it can be seen in Fig 14. which is the date of the last previous get request.

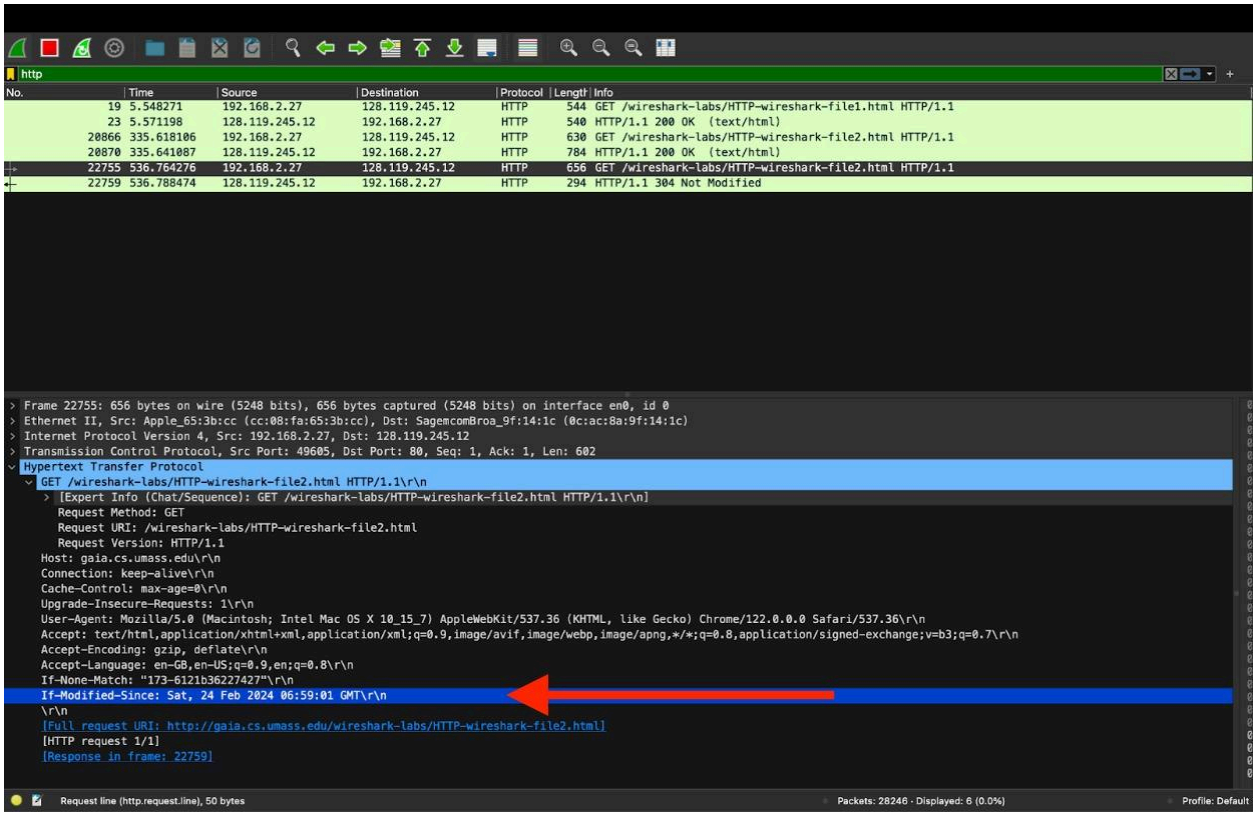

Fig 14

11. [3 marks] What is the HTTP status code and phrase returned from the server in response to this second HTTP GET? Did the server explicitly return the contents of the file? Explain.

Answer 11

The status code and phrase returned from the server in the second HTTP GET is **HTTP/1.1 304 Not Modified**.

The server didn't return the contents of the file since the browser loaded it from its cache. It can be also seen in Fig 15 that there is no contents of the file returned from the server.

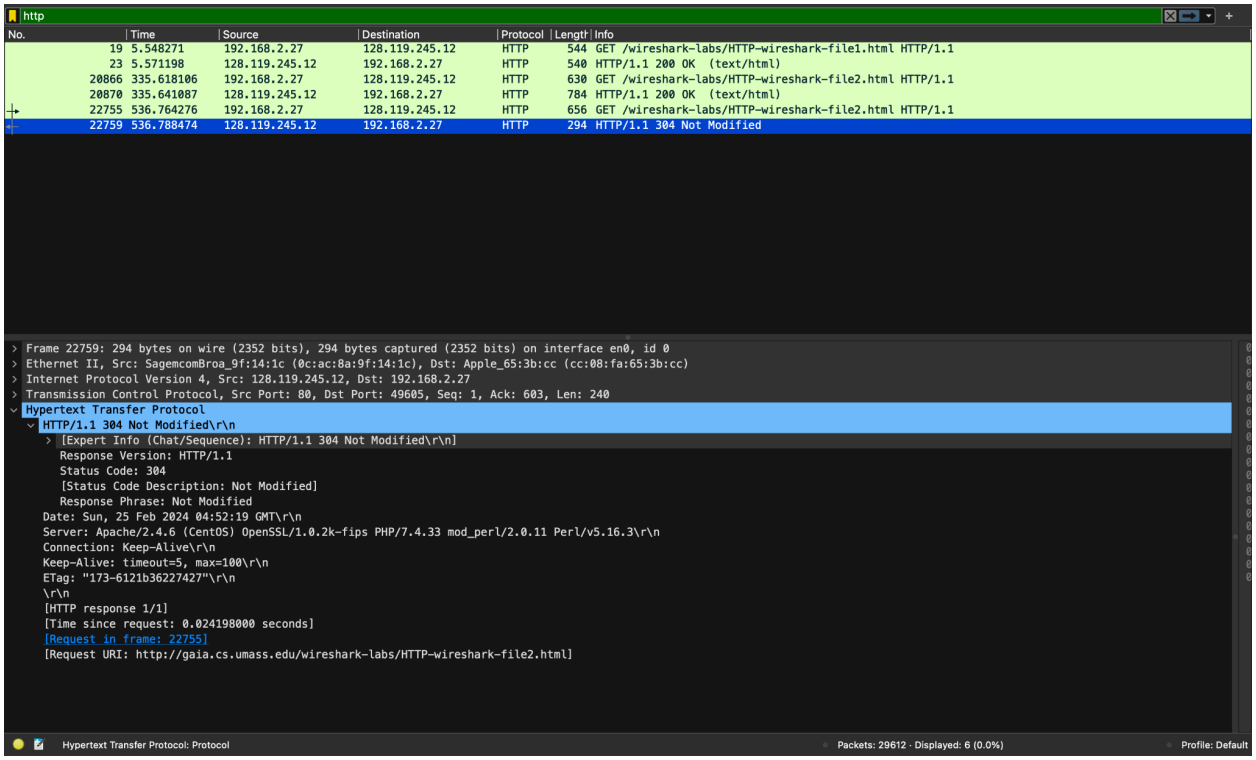

Fig 15

Server returned the contents in the first Get request (as seen in Fig 16) but not in the second request.

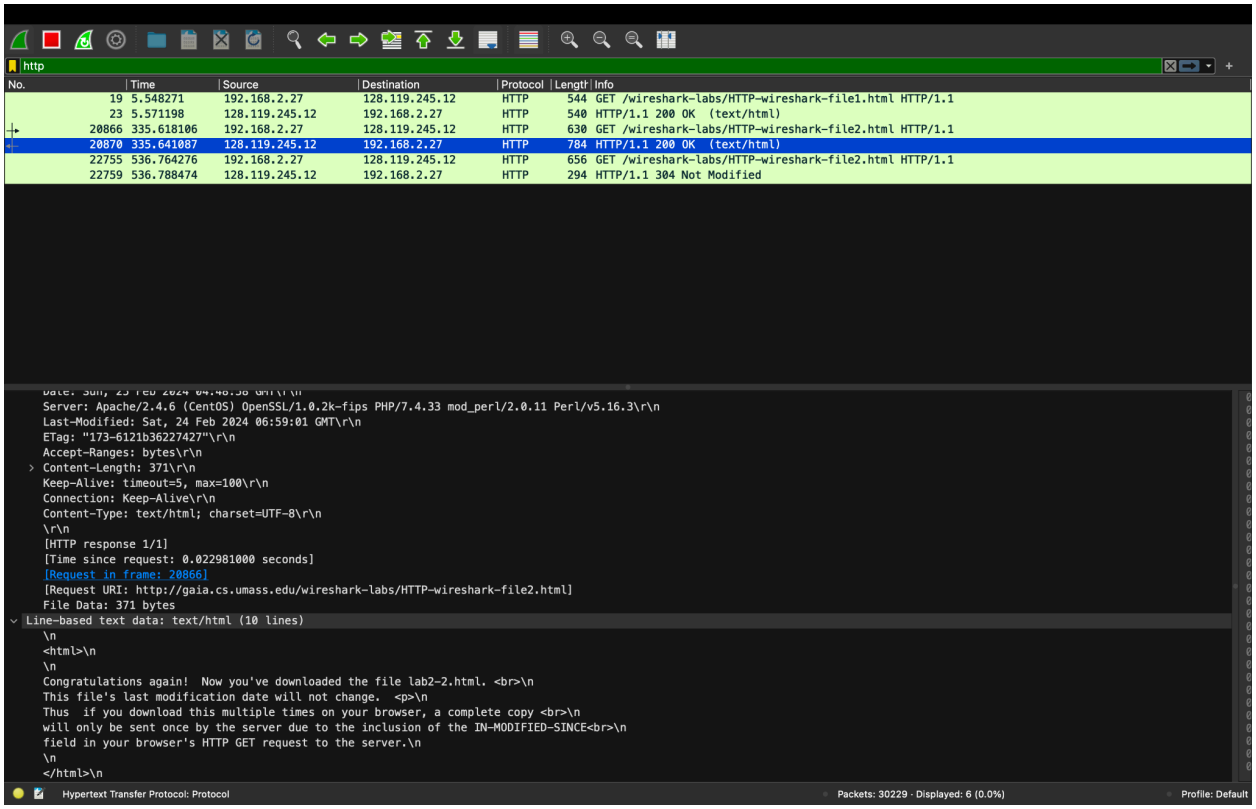

Fig 16

12. [2 marks] How many HTTP GET request messages did your browser send?

Answer 12

There was 1 HTTP GET request message sent by my browser as it can be seen in Fig 17.

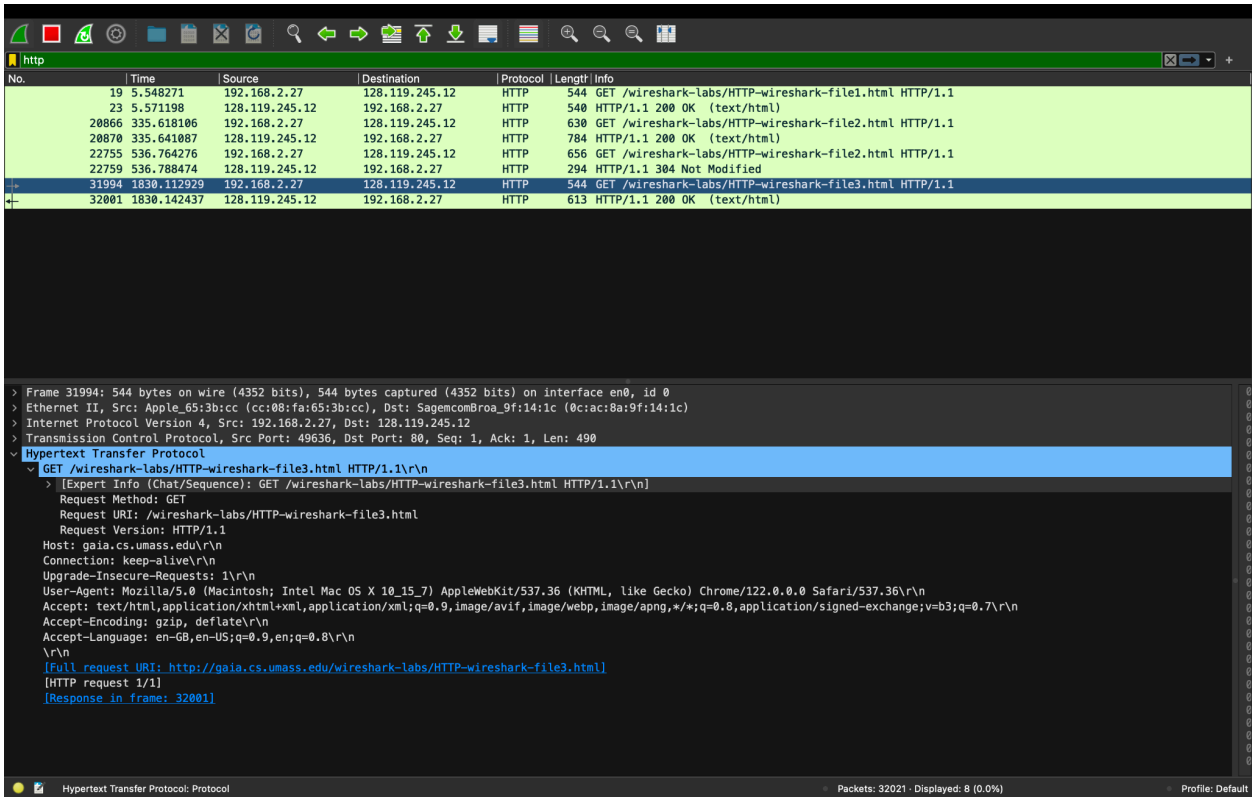

Fig 17

13. [2 marks] Which packet number in the trace contains the status code and phrase associated with the response to the HTTP GET request?

Answer 13

Packet Number(frame 32001) , as it can be seen in Fig 18 that packet 32001 contains HTTP/1.1 200 OK response.

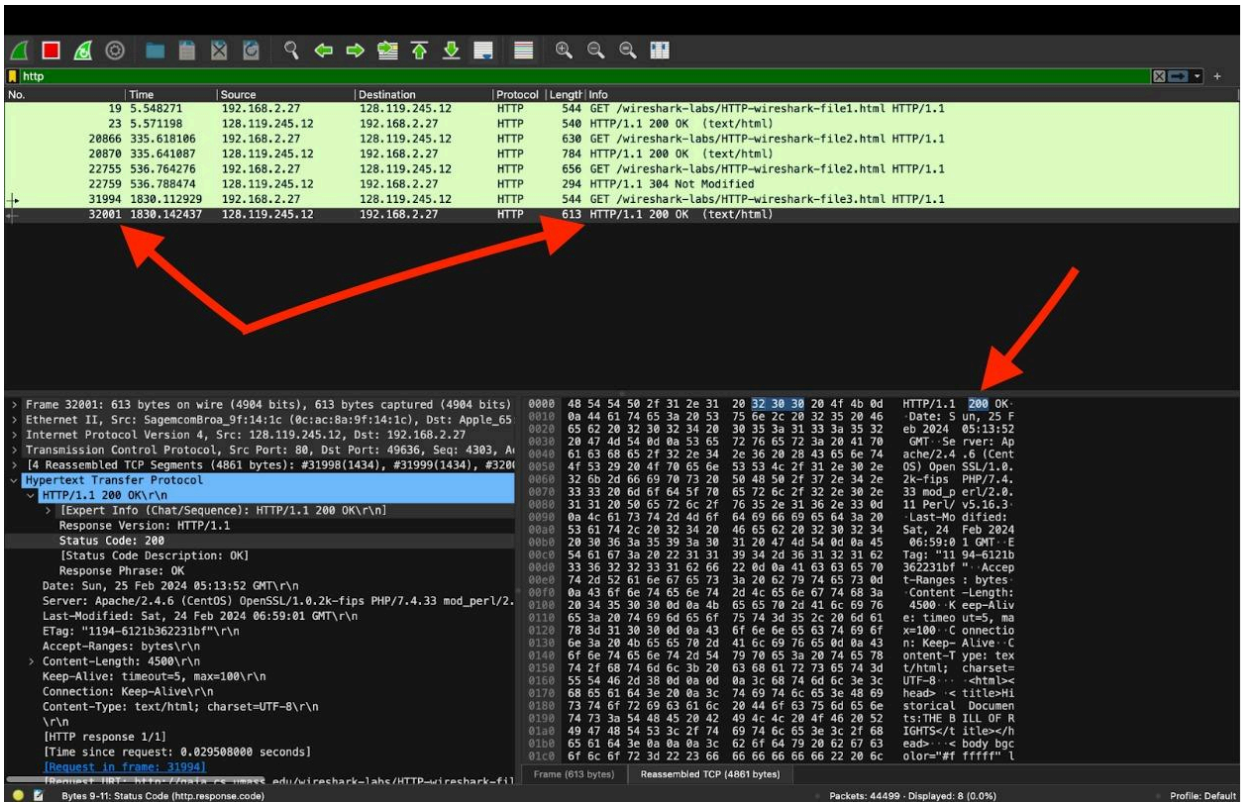

Fig 18

14. [2 marks] What is the status code and phrase in the response?

Answer 14 **200 OK**, As it can be seen in Fig 19 that HTTP status code is 200 OK.

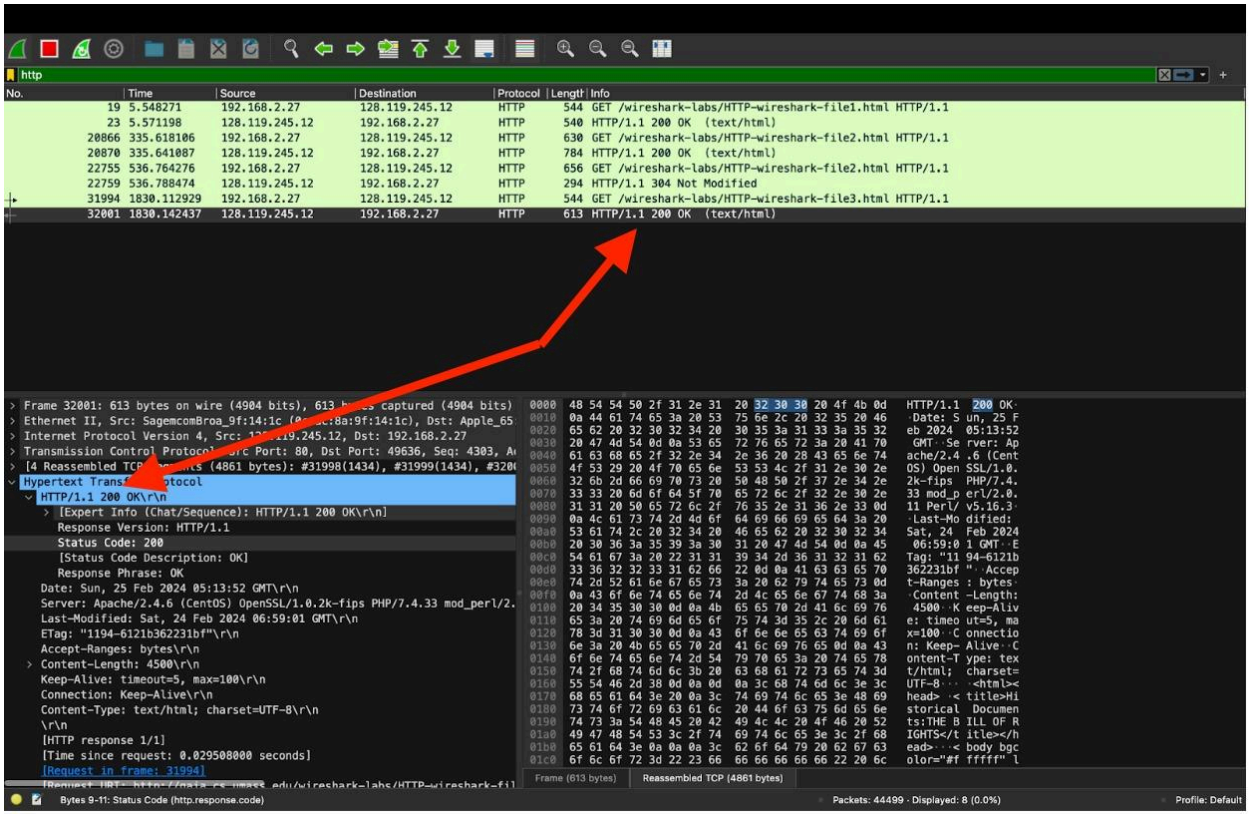

Fig 19

15. [3 marks] How many data-containing TCP segments were needed to carry the single HTTP response?

Answer 15

According to the fig 20 below, 3 TCP segments (31998,31999, 32000) were needed to carry the single HTTP response.

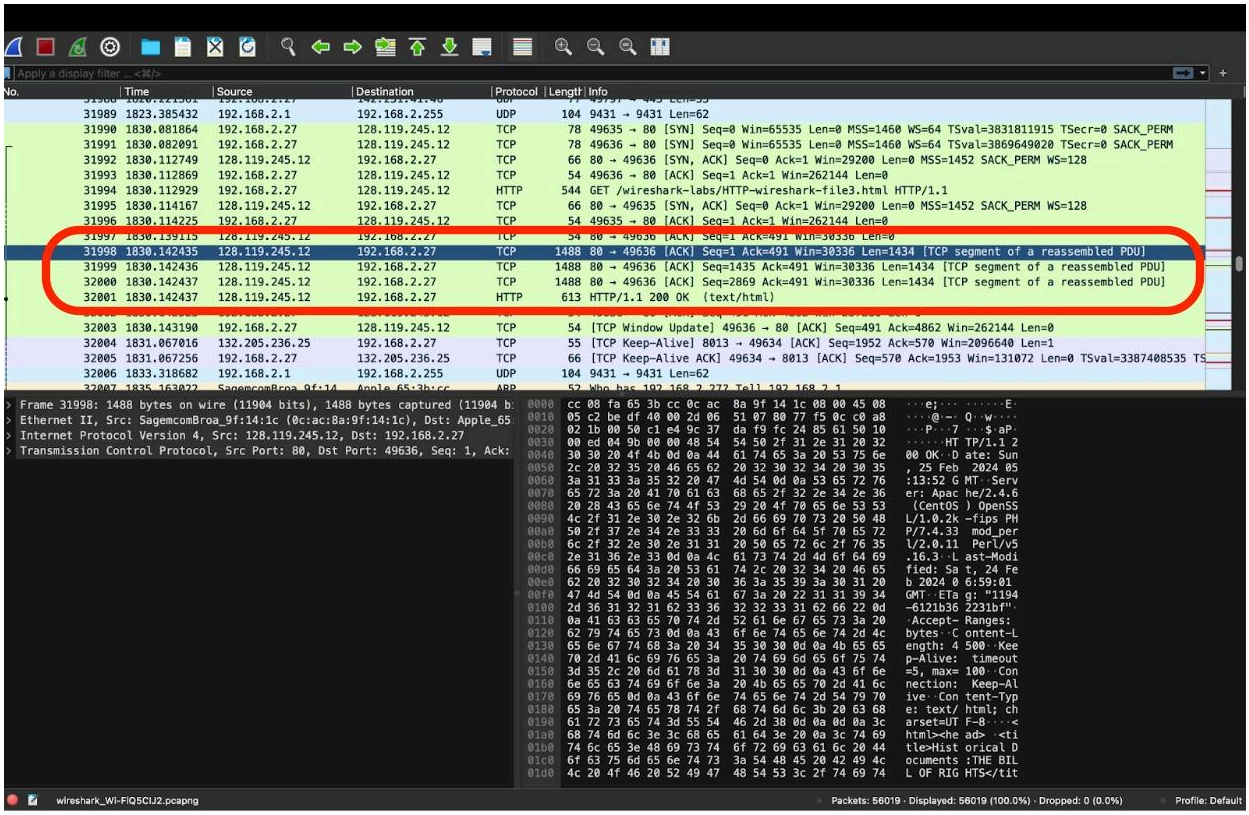

Fig 20

16. [6 marks] How many bytes of overhead were generated in TCP to transport the response? What percentage is the TCP overhead of the entire TCP + HTTP + Data transmission? (Yes, we deliberately ignore IP and Ethernet for this question). Make sure to explain what you think is Overhead.

Answer 16

TCP Overhead is 20 bytes (as it can be seen in Fig 21)
Total TCP Header Size= 20 * 3= 60 bytes
Total Http Header size= 361 bytes

HTTP Content length + HTTP Header + TCP Header is 4500 bytes + 361 bytes + 20*3 bytes= 4921 bytes(as it can be seen in Fig 21 and Fig 22)

Percentage Overhead = TCP Overhead/(TCP Length + HTTP Content length + TCP Header) * 100

Percentage Overhead = 60/4921 * 100

Percentage Overhead = 1.21 Percent

In the case of TCP, the overhead includes the TCP header, which contains control information necessary for reliable communication between the sender and receiver and managing the TCP connection.

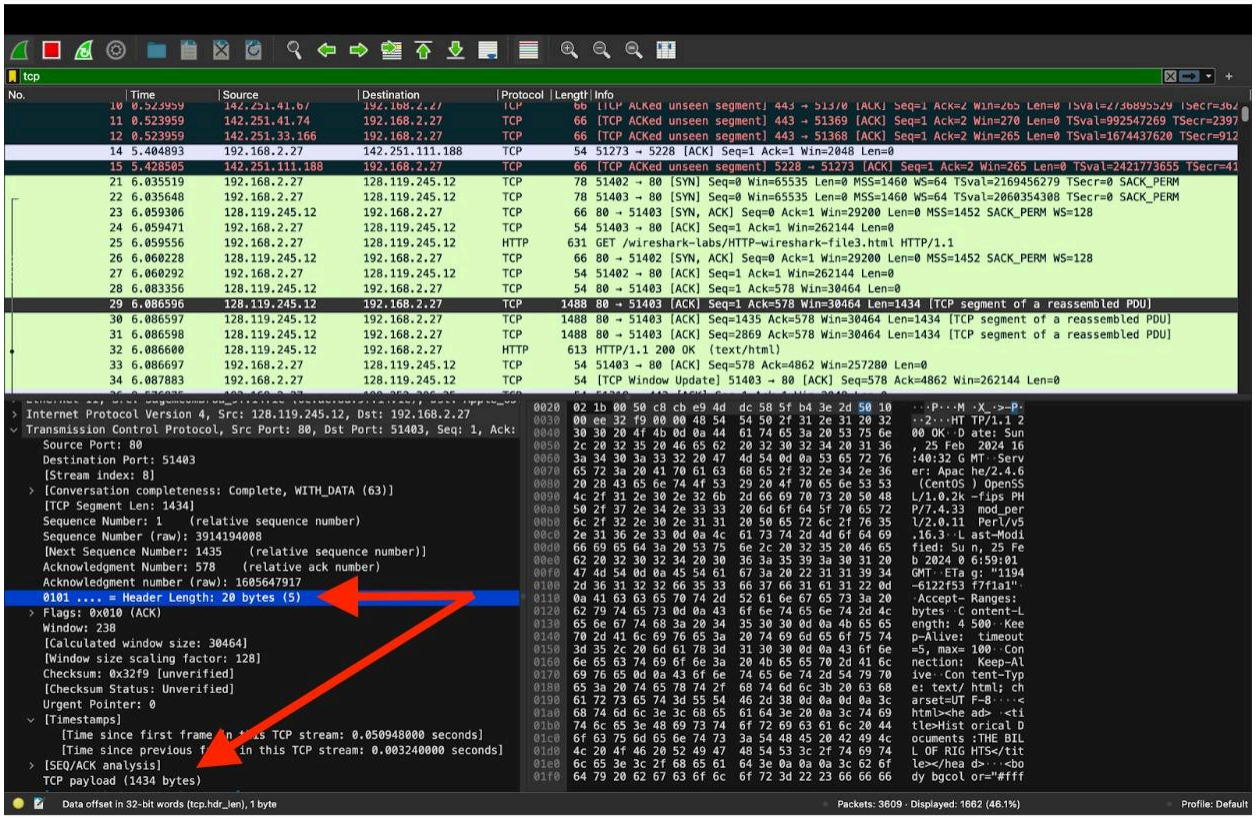

Fig 21

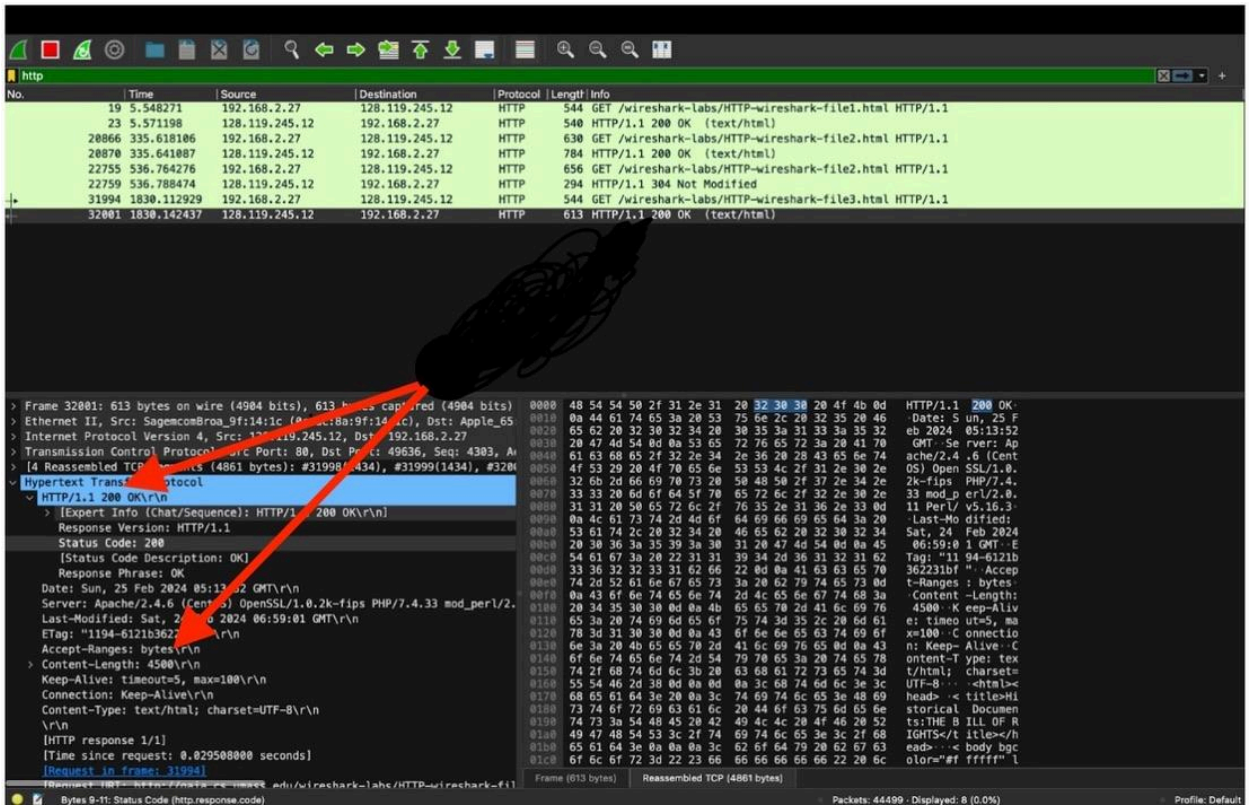

Fig 22

17. [5 marks] How many HTTP GET request messages did your browser send? To which Internet addresses were these GET requests sent?

Answer 17

According to Fig 23, the browser sent 3 HTTP GET request messages. Packet 41 was sent to 128.119.245.12, packet 49 was sent to 128.119.245.12, and packet 80 was sent to 178.79.137.164

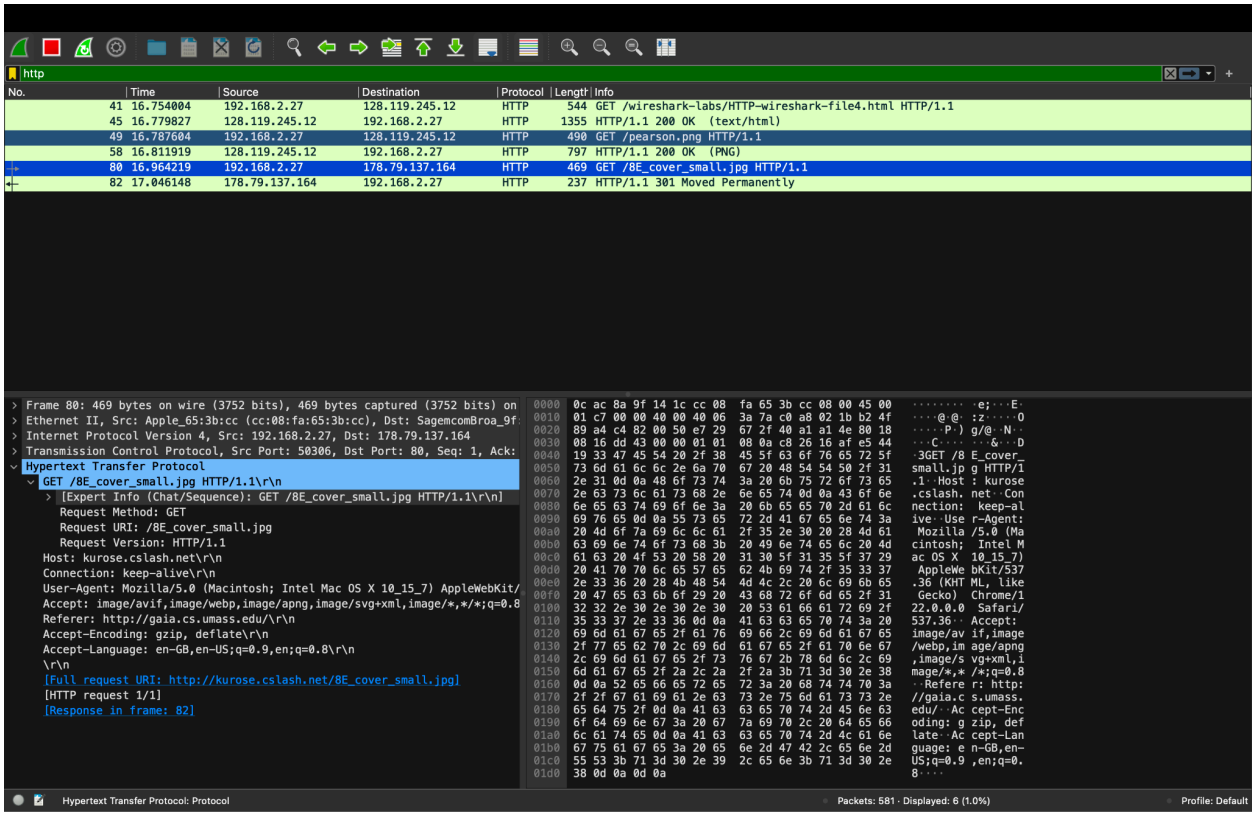

Fig 23

18. [5 marks] Can you tell whether your browser downloaded the two images serially or downloaded them from the two websites in parallel? Explain.

Answer 18

The browser downloaded the two images serially. As we can see in the fig 24, the time stamp shows that the **png** image was downloaded after the 2nd **GET** and the **jpg** image was downloaded after the 3rd **GET** request.

Also By checking the TCP ports, we can confirm if our files were downloaded serially or in parallel. In this case the 2 images were transmitted over 2 TCP connections therefore they were downloaded serially. (as seen in fig 24 and fig 25).

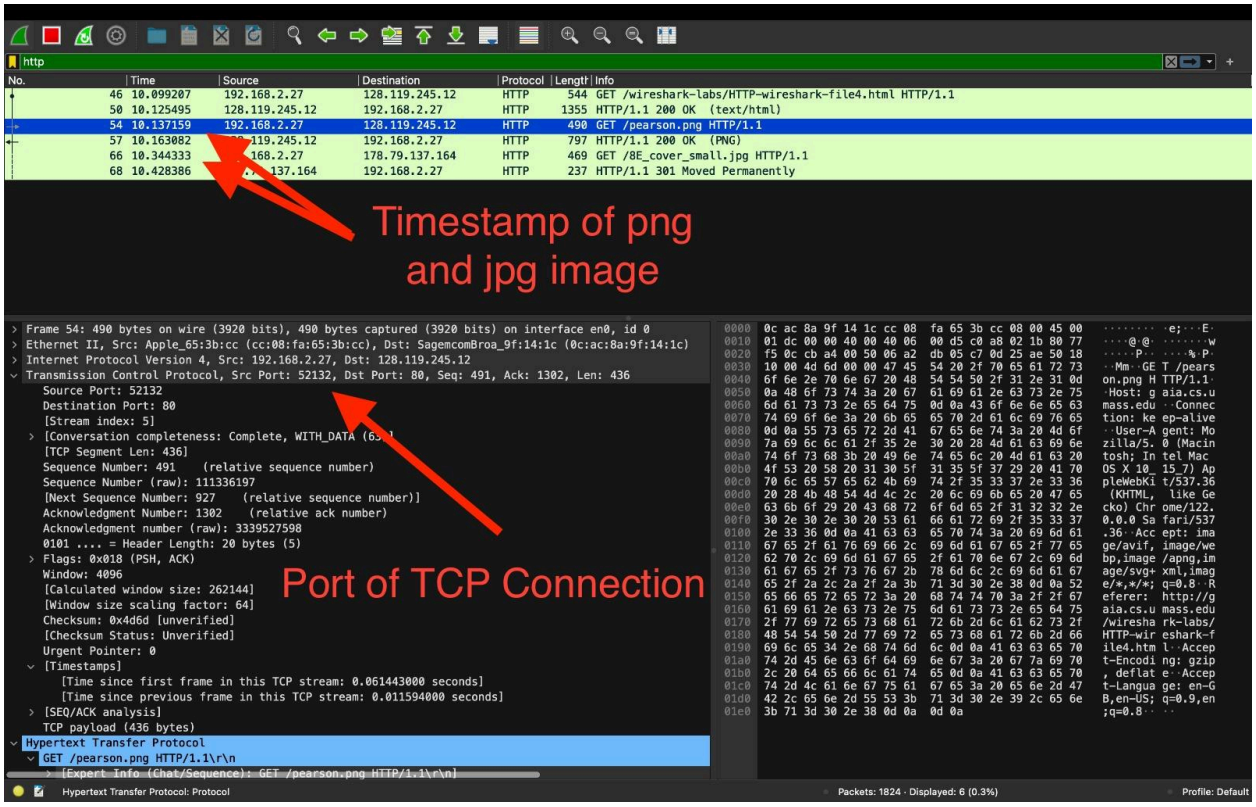
http
No. Time Source Destination Protocol Length Info
46 10.099207 192.168.2.27 128.119.245.12 HTTP 544 GET /wireshark-labs/HTTP-wireshark-file4.html HTTP/1.1
50 10.125495 128.119.245.12 192.168.2.27 HTTP 1355 HTTP/1.1 200 OK (text/html)
54 10.137159 192.168.2.27 128.119.245.12 HTTP 490 GET /pearson.png HTTP/1.1
57 10.163082 128.119.245.12 192.168.2.27 HTTP 797 HTTP/1.1 200 OK (PNG)
66 10.344333 168.2.27 178.79.137.164 HTTP 469 GET /8E_cover_small.jpg HTTP/1.1
68 10.428386 137.164 192.168.2.27 HTTP 237 HTTP/1.1 301 Moved Permanently
Timestamp of png and jpg image
Frame 54: 490 bytes on wire (3920 bits), 490 bytes captured (3920 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: 52132, Dst Port: 80, Seq: 491, Ack: 1302, Len: 436
Source Port: 52132
Destination Port: 80
[Stream index: 5]
[TCP Segment Len: 436]
Sequence Number: 491 (relative sequence number)
Sequence Number (raw): 111336197
[Next Sequence Number: 927 (relative sequence number)]
Acknowledgment Number: 1302 (relative ack number)
Acknowledgment number (raw): 3339527598
0101 .... = Header Length: 20 bytes (5)
Flags: 0x018 (PSH, ACK)
Window: 4096
[Calculated window size: 262144]
[Window size scaling factor: 64]
Checksum: 0x4d6d [unverified]
[Checksum Status: Unverified]
Urgent Pointer: 0
[Timestamps]
[Time since first frame in this TCP stream: 0.061443000 seconds]
[Time since previous frame in this TCP stream: 0.011594000 seconds]
[SEQ/ACK analysis]
TCP payload (436 bytes)
Hypertext Transfer Protocol
GET /pearson.png HTTP/1.1\r\n
Port of TCP Connection
Hypertext Transfer Protocol: Protocol
Packets: 1824 · Displayed: 6 (0.3%)
Profile: Default

Fig 24

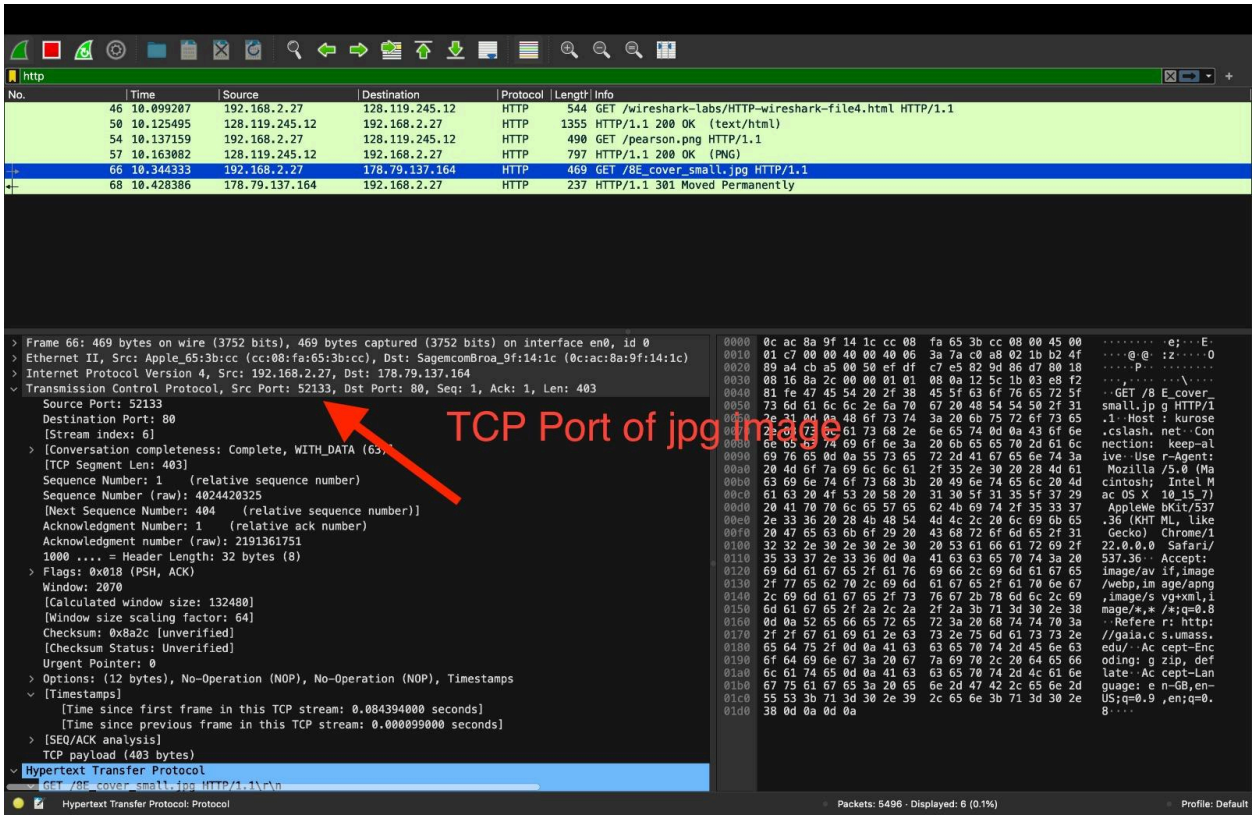
http
No. Time Source Destination Protocol Length Info
46 10.099207 192.168.2.27 128.119.245.12 HTTP 544 GET /wireshark-labs/HTTP-wireshark-file4.html HTTP/1.1
50 10.125495 128.119.245.12 192.168.2.27 HTTP 1355 HTTP/1.1 200 OK (text/html)
54 10.137159 192.168.2.27 128.119.245.12 HTTP 490 GET /pearson.png HTTP/1.1
57 10.163082 128.119.245.12 192.168.2.27 HTTP 797 HTTP/1.1 200 OK (PNG)
66 10.344333 192.168.2.27 178.79.137.164 HTTP 469 GET /8E_cover_small.jpg HTTP/1.1
68 10.428386 178.79.137.164 192.168.2.27 HTTP 237 HTTP/1.1 301 Moved Permanently
Frame 66: 469 bytes on wire (3752 bits), 469 bytes captured (3752 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: 178.79.137.164
Transmission Control Protocol, Src Port: 52133, Dst Port: 80, Seq: 1, Ack: 1, Len: 403
Source Port: 52133
Destination Port: 80
[Stream index: 6]
[TCP Segment Len: 403]
Sequence Number: 1 (relative sequence number)
Sequence Number (raw): 4024420325
[Next Sequence Number: 404 (relative sequence number)]
Acknowledgment Number: 1 (relative ack number)
Acknowledgment number (raw): 2191361751
1000 .... = Header Length: 32 bytes (8)
Flags: 0x018 (PSH, ACK)
Window: 2070
[Calculated window size: 132480]
[Window size scaling factor: 64]
Checksum: 0x8a2c [unverified]
[Checksum Status: Unverified]
Urgent Pointer: 0
Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
[Timestamps]
[Time since first frame in this TCP stream: 0.084394000 seconds]
[Time since previous frame in this TCP stream: 0.000099000 seconds]
[SEQ/ACK analysis]
TCP payload (403 bytes)
Hypertext Transfer Protocol
GET /8E_cover_small.jpg HTTP/1.1\r\n
TCP Port of jpg image
Hypertext Transfer Protocol: Protocol
Packets: 5496 · Displayed: 6 (0.1%)
Profile: Default

Fig 25