



INDIAN INSTITUTE OF TECHNOLOGY PALAKKAD

Department of Computer Science and Engineering

CS4010 Computer Networks

August-December 2024

IIT PALAKKAD

Assignment – 1

Name: Neeraj Krishna N

Roll no: 112101033

Part 1

Q1

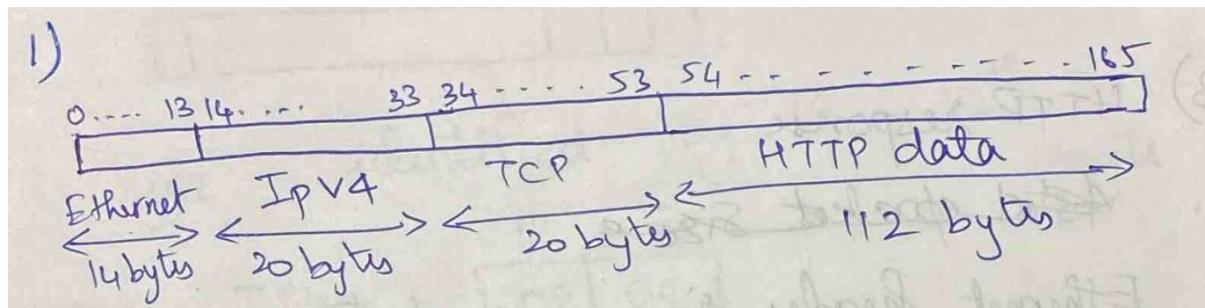


Figure 1: Packet

Q2

For packet-7, the total packet size is 1484 bytes.

Ethernet Header Size = 14 bytes

IPv4 header size = 20 bytes

TCP header size = 20 bytes

$$\begin{aligned}\text{Overhead} &= \frac{\text{Header size}}{\text{Total packet size}} \\ &= \frac{14 + 20 + 20}{1484} \\ &= 3.64\%\end{aligned}$$

Q3

HTTP response

Ethernet Header size = 14 bytes

IPv4 header size = 20 bytes

TCP header size = 20 bytes

$$\begin{aligned}
 \text{HTTP headers + messages(Useful data size)} &= \text{HTTP file data size + html data size} \\
 &= 736 + 14201 \text{ bytes} \\
 &= 14937 \text{ bytes} \\
 \text{Overhead} &= \frac{\text{Total header size}}{\text{Total Packet size}} \\
 &= \frac{14 + 20 + 20}{14937 + 14 + 20 + 20} \\
 &= \frac{54}{14991} \\
 &\approx 0.36\%
 \end{aligned}$$

Q4

Ethernet header size is 14 bytes (from 0th to 13th byte in the packet)

Type is identified by the 12th and the 13th byte. In this case the 12th and 13th byte together has the value 0x0800 which indicates that the type is IPv4. Hence 12th and 13th byte of the Ethernet header field is the demultiplexing key indicating that the next layer is IP, and the value used in this field is 0x0800 to indicate IP.

Q5

IP header size is 20 bytes. In the packet the position of IP header is from 14th to 33rd byte. 9th byte of IP header (equivalently 23rd byte in the packet (indexing starting from 0th byte)), is the demultiplexing key indicating that the next layer is TCP and the value used in this field to indicate TCP is 0x06.

Q6

TCP doesn't include a separate demultiplexing key because here the port number which is 80 acts like the demultiplexing key. If the port number is 80, then it means that HTTP is used for handling the data. TCP port is a unique number assigned to different applications

Part 2

Q1

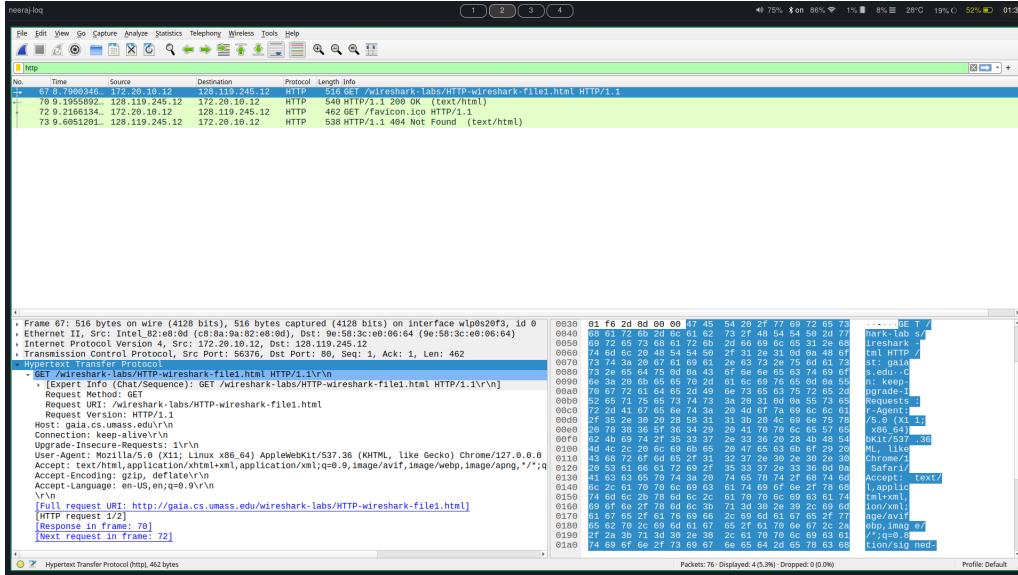


Figure 2: Request Version

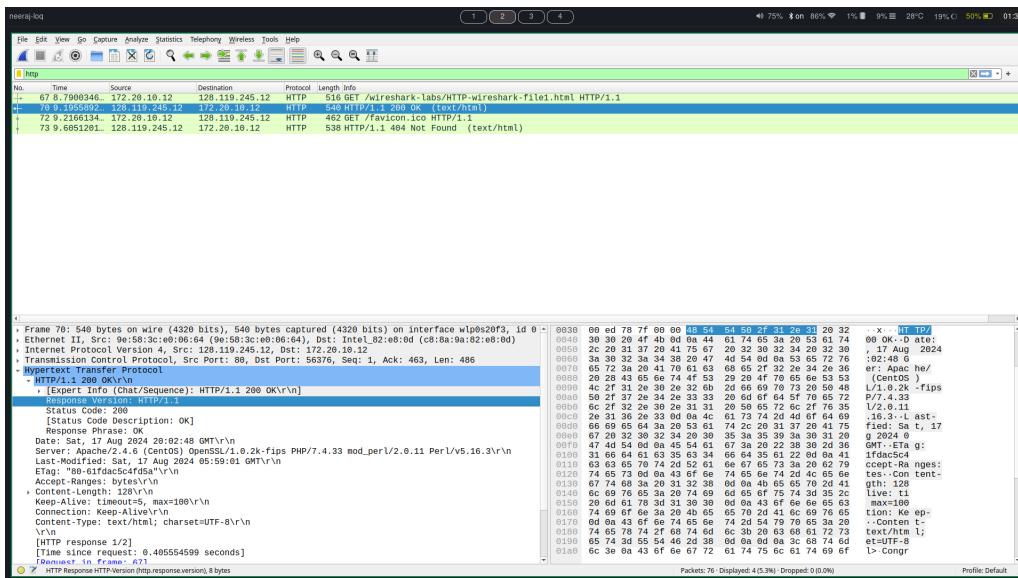


Figure 3: Response Version

Browser version can be checked at the Request version in the captured packet and similarly, server version can be checked at the Response version in the packet captured.

My browser is running HTTP version 1.1 and the server is also running HTTP version 1.1

Q2

In the figure 2, we can see that there is a field named "Accept Languages: en-US; en;q=0.9;\r\n" Browser indicates that it can accept "en-US" (English of US)

Q3

IP address of my computer can be seen from the "src" field of IPv4 header in figure 2, which is 172.20.10.12 and similarly IP address of "gaia.cs.umass.edu.server" can be seen from the "src" field of IPv4 header in figure 3 which is 128.119.245.12

Q4

The status code can be seen from the packet listing, which is shown to be "200 OK"

Q5

"Last modified" field can be checked in the HTTP header in the response packet of the server shown in figure 3 and the value of "Last Modified" is "Sat, 17 Aug 2024 05:59:01 GMT"

Q6

Content size is 128 bytes as can be seen in the bottom status bar in the figure 4

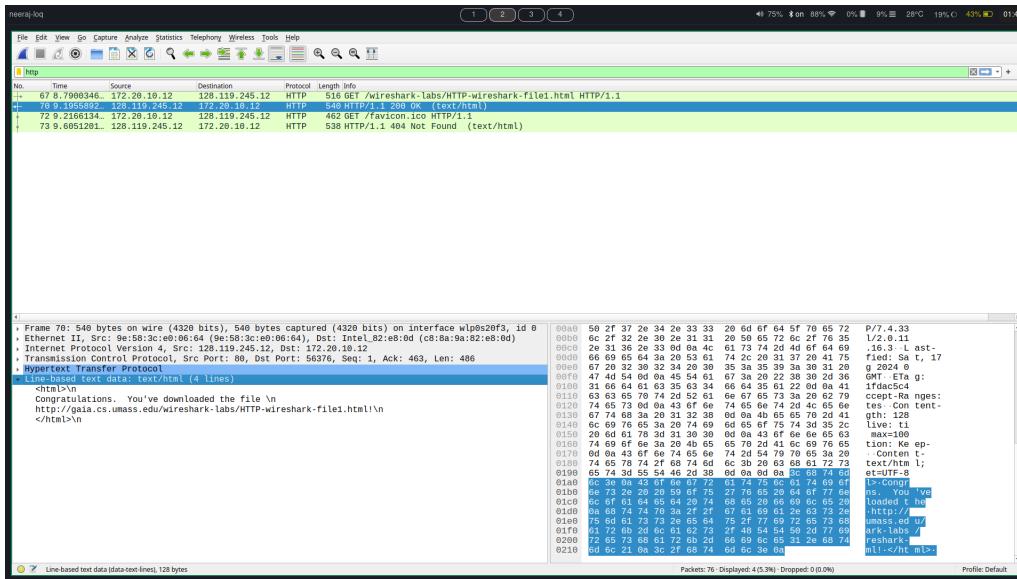


Figure 4: Content Size

Q7

No, all headers are being shown in the packet listing

Q8

No, there is no field "If-Modified-Since" in the first HTTP Get request as shown in figure 5

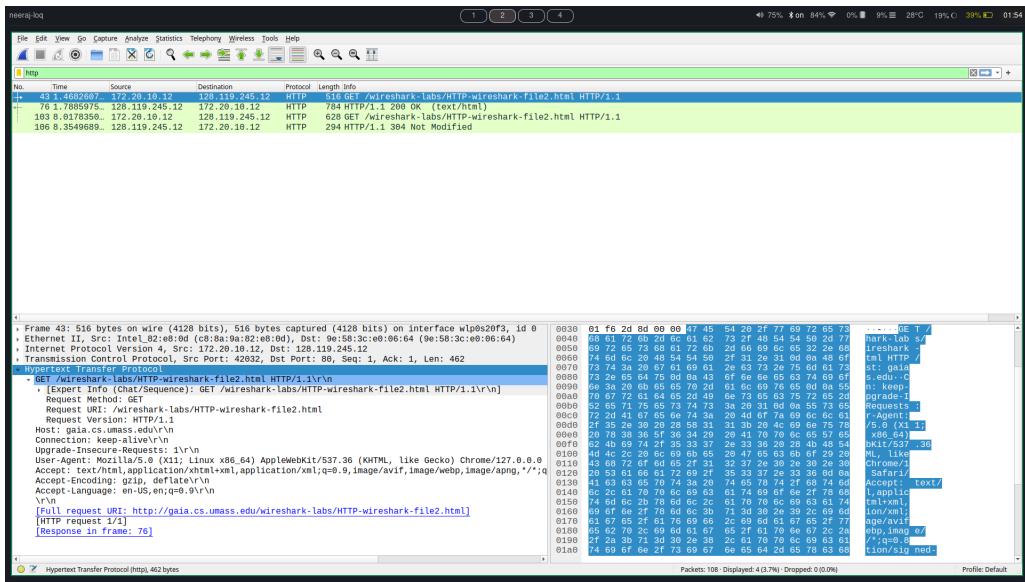


Figure 5: No field named "If Modified" in the first GET request

Q9

Yes, the server explicitly returned the contents of the file. This can be seen from the response of the server which contains the HTML data (the section "Line Based text data")

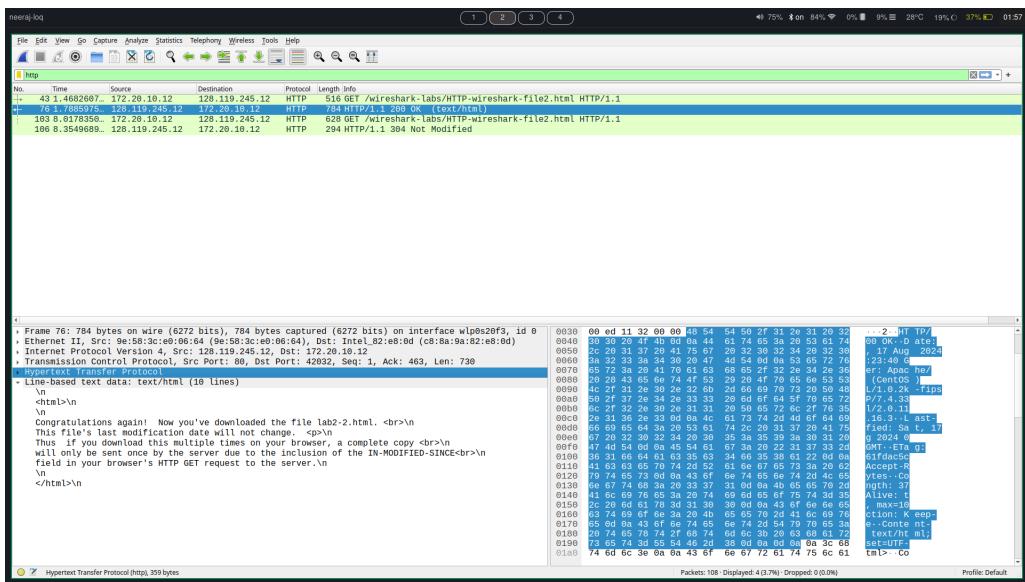


Figure 6: Server explicitly returned the contents

Q10

Yes, now IF-Modified-Since field is being shown. The information shown is "If-Modified-Since: Sat, 17 Aug 2024 05:59:01 GMT"

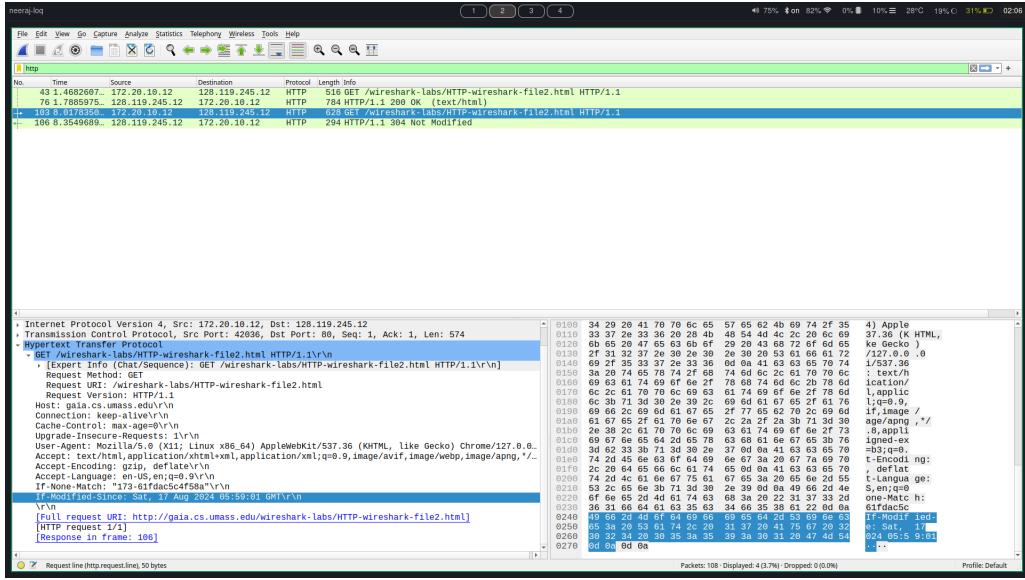


Figure 7: If-Modified-Since field present in the HTTP header

Q11

The status code and phrase is "304 Not Modified", which can be seen from the last response from the server in the figure 7. The server, this time did not explicitly return the contents of the file because, the browser had cached the contents of the file from the response of the previous GET request and since the file has not been modified since then (which can be identified from the "If-Modified=Since" field in the GET request) browser renders the contents of the file from the cache

Q12

The packet whose packet number is 143 (whose timestamp is marked as REF) is the first GET request) in the figure 8. Only 1 HTTP GET request was sent by the browser which can be seen from packet number 143

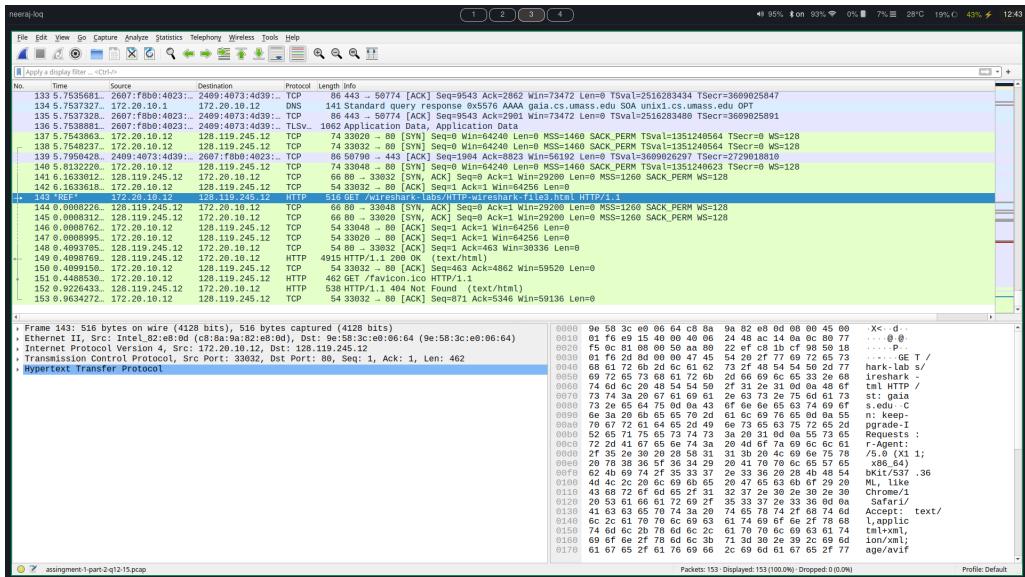


Figure 8: Packet trace for retrieving long document

Q13

Packet number 149 in the trace contains the status code and phrase associated with the response of the HTTP GET request (ref. figure 8)

Q14

Status code and phrase is "200 OK" as can be seen from packet number 149, from the figure 8

Q15

In total, 6 data containing TCP segments were needed to carry the single HTTP response and the text of Bill of Rights.

HTTP response is the packet number 149 in the figure 8.

The text containing the Bill of Rights is carried by 5 TCP segments, as can be seen from the packets 144 to 148 in the figure 8

Q16

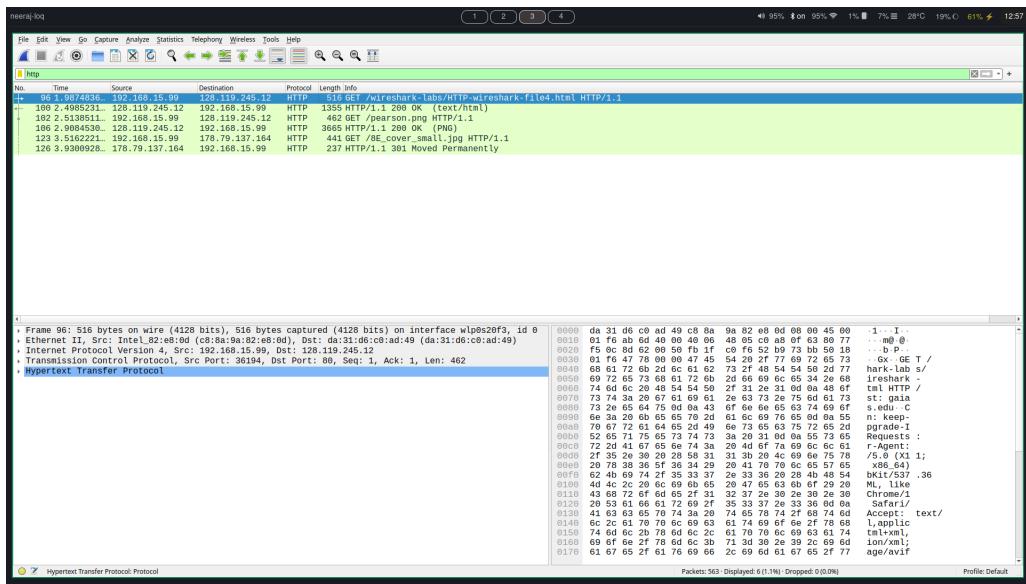


Figure 9: Packet trace for HTML documents with embedded images

1. 3 HTTP GET requests were sent in total as can be seen from the packet trace in the figure 9 (Packet numbers 96, 102, 123)
2. (a) Packet number 96 is the first GET request to get the html document. Destination IP address is 128.119.245.12
(b) Packet number 102 is the second GET request to get the first embedded image ("pearson.png") and the destination IP address is 128.119.145.12
(c) Packet number 123 is the third GET request to get the second embedded image ("8E_cover_small.png") and the destination IP address is 178.79.137.164

Q17

They were downloaded in serial because from the packet trace (in figure 9), we can see that after the GET request to the first image (packet number 102), the GET request for the second image (packet number 123) was sent only after the response for the first image (packet number 106).

Q18

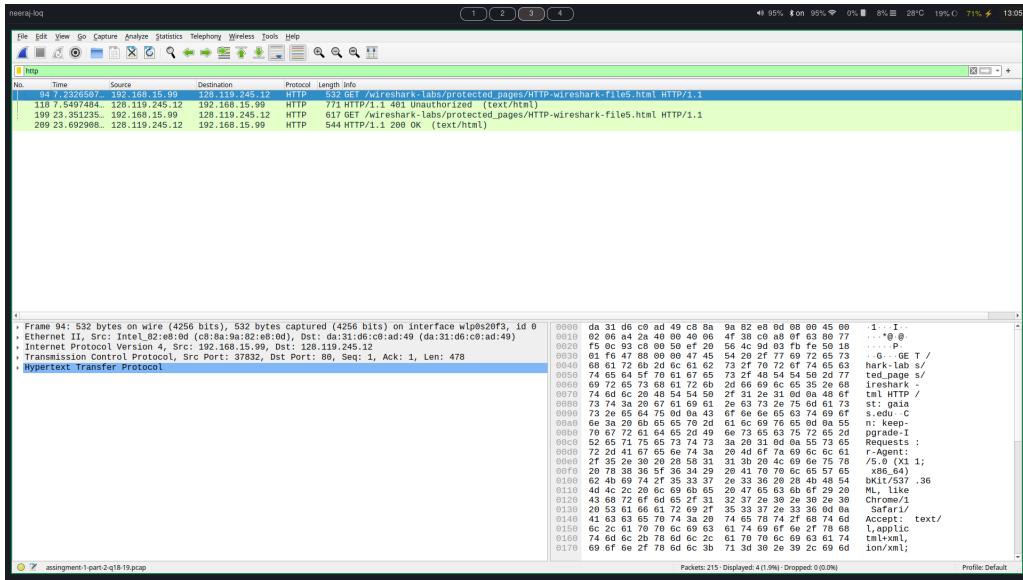


Figure 10: Packet trace for HTTP Authentication

Initial GET request is packet number 94 in the packet trace shown in the figure 10, and initial response is packet number 118 in the figure 10. The status code and phrase is "401 Unauthorized" as shown in the packet listing window.

Q19

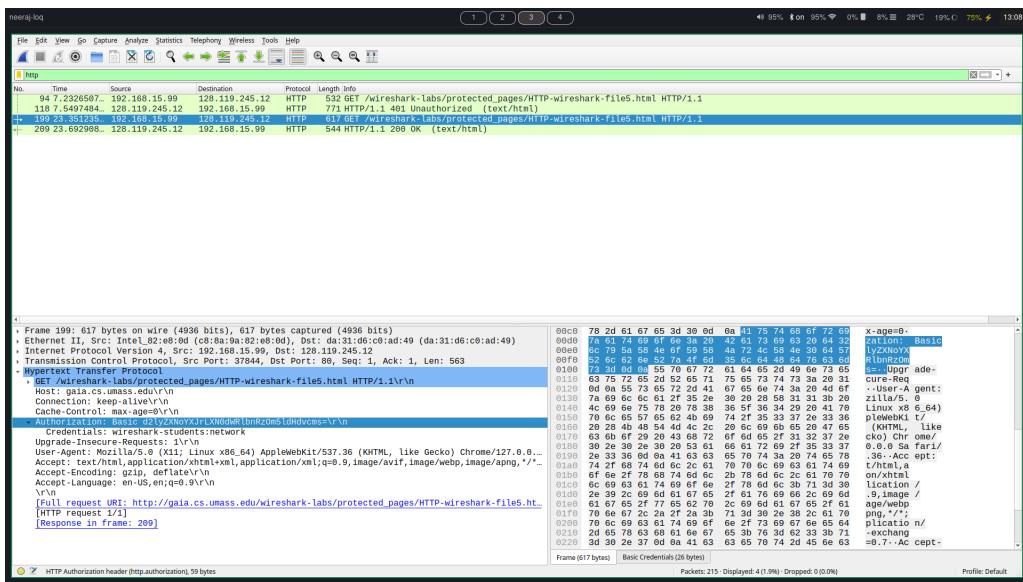


Figure 11: 2nd GET request for HTTP authentication

A new field named "Authorization" is added. as can be seen from the selected part in the figure 11.

```
Authorization: Basic d2lyZXNoYXJrLXN0dWR1bnRzOm5ldHdvcms=\r\n
Credentials: wireshark-students:network
```