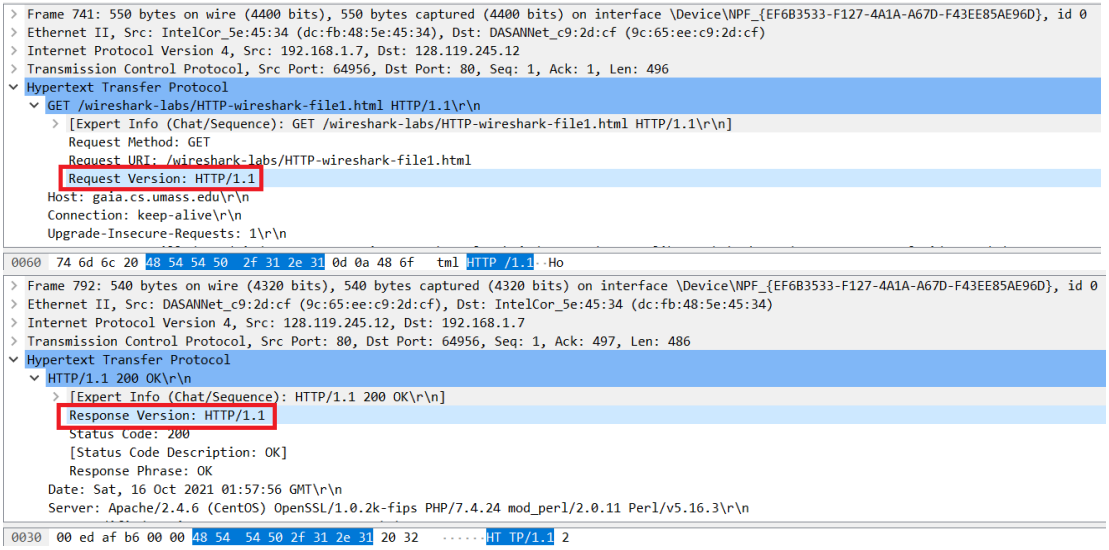
COMPUTER NETWORK
Lab 2a

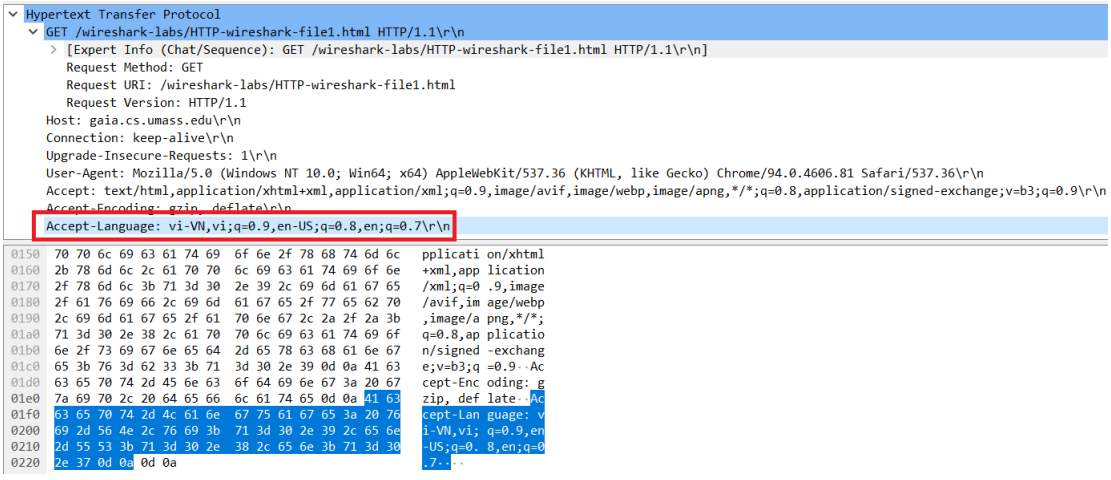
Student name: Nguyễn Minh Tâm
ID: 1952968

I/ The Basic HTTP GET/response interaction

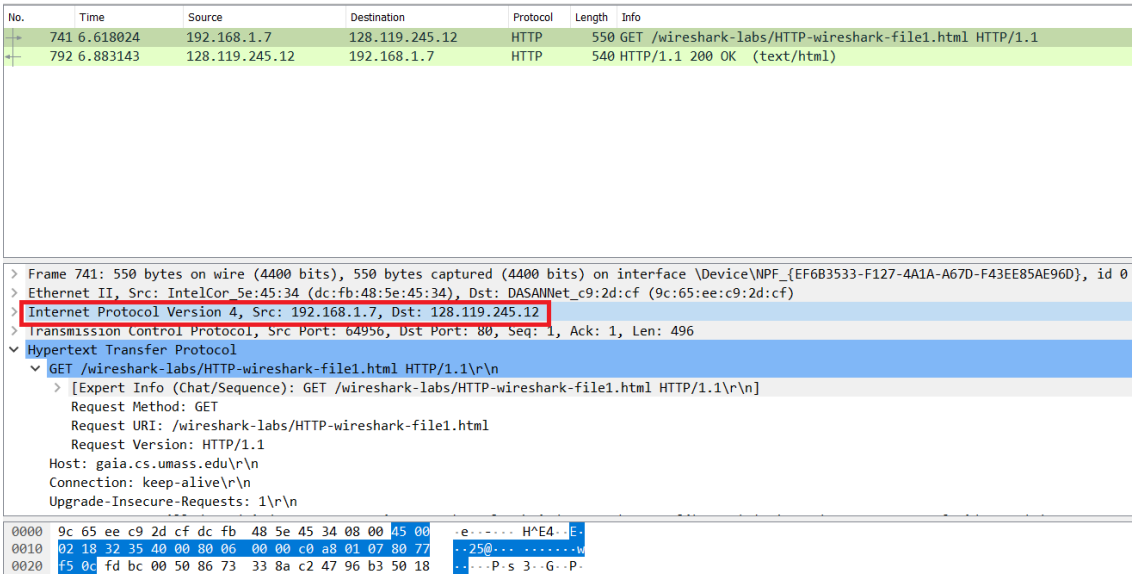
1. Is your browser running HTTP version 1.0 or 1.1? What version of HTTP is the server running?
   Ans: Both of them are version 1.1



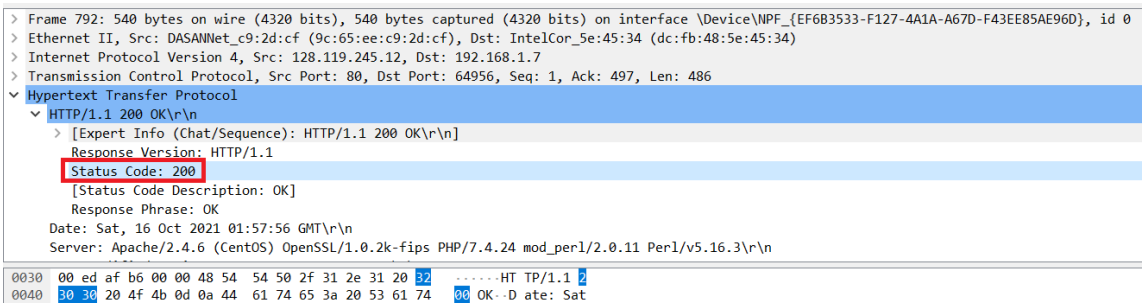
2. What languages (if any) does your browser indicate that it can accept to the server?
   Ans: vi-VN and en-US



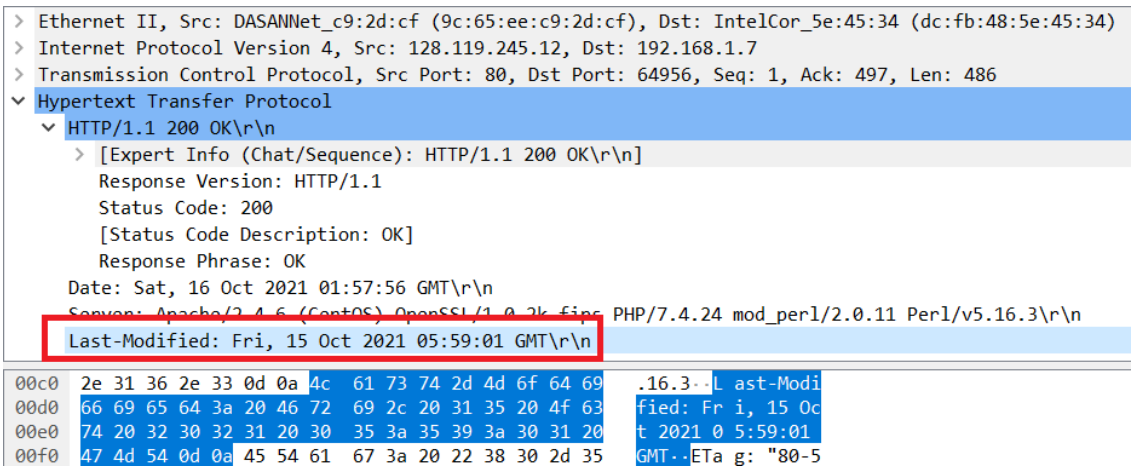
3. What is the IP address of your computer? Of the gaia.cs.umass.edu server?
   My computer: 192.168.1.7
   gaia.cs.umass.edu: 128.119.245.12



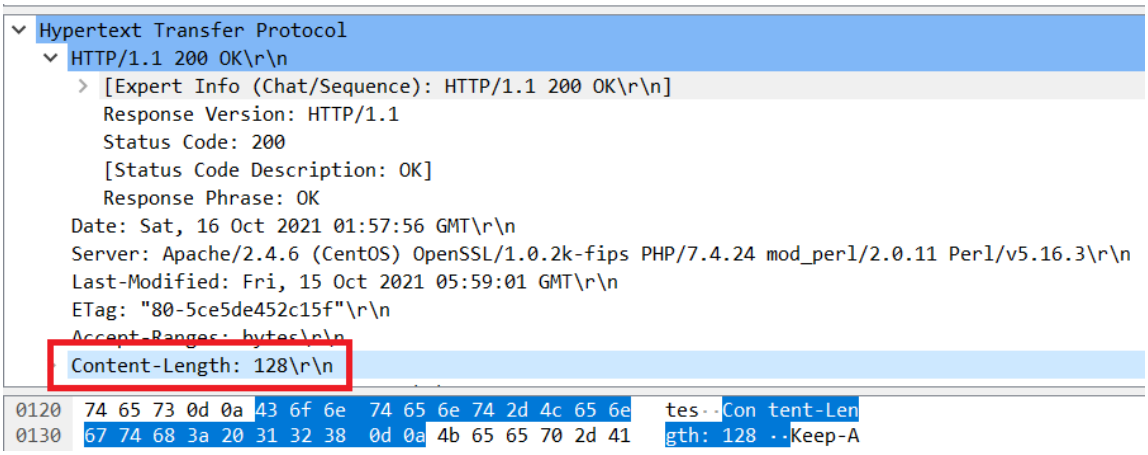
4. What is the status code returned from the server to your browser?
   Ans: 200



5. When was the HTML file that you are retrieving last modified at the server?
   Ans: Fri, 15 Oct 2021 05:59:01 GMT



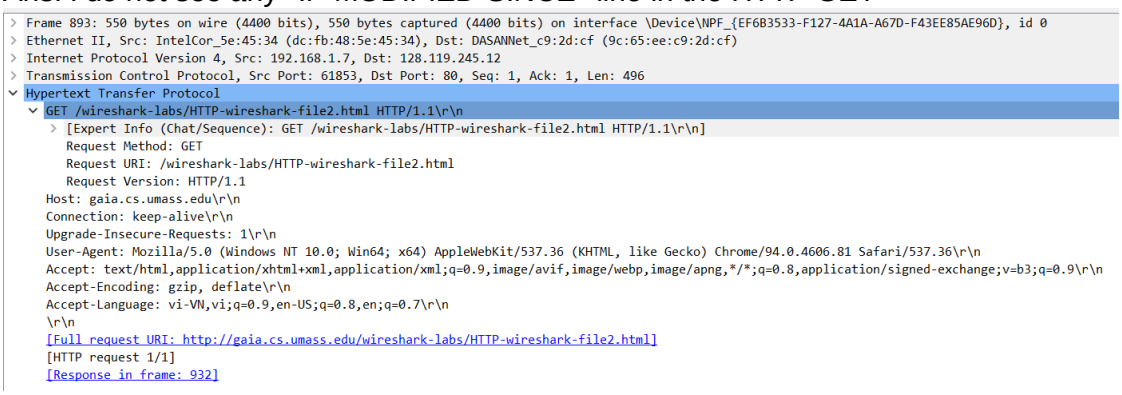
6. How many bytes of content are being returned to your browser?
   Ans: 128 bytes



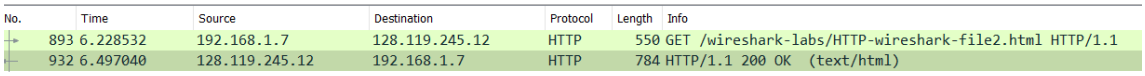
7. By inspecting the raw data in the packet content window, do you see any headers within the data that are not displayed in the packet-listing window? If so, name one.
   Ans: I do not see any headers within the data that are not displayed in the packet-listing window

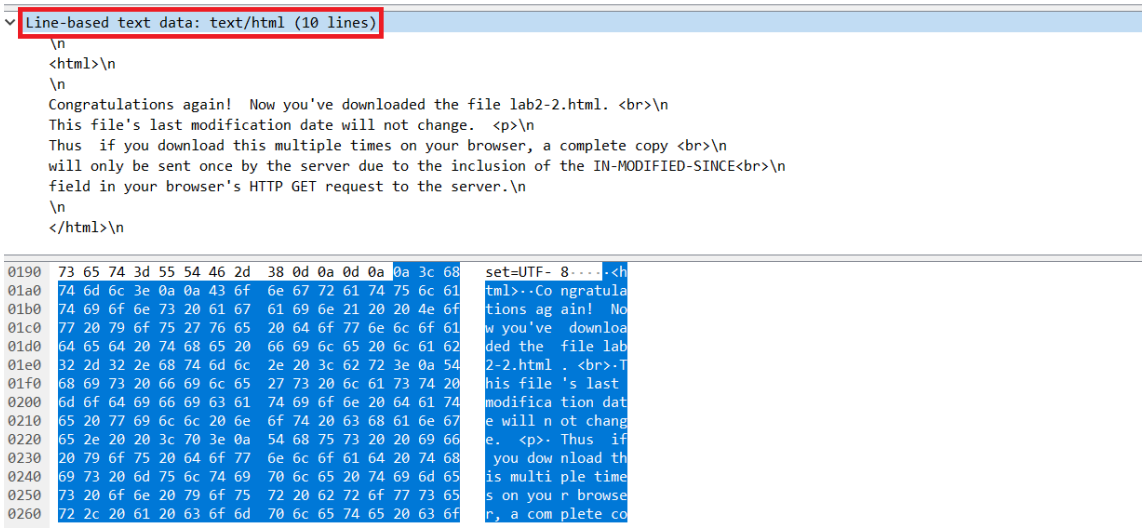
II/ The HTTP CONDITIONAL GET/response interaction

8. 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?
   Ans: I do not see any “IF-MODIFIED-SINCE” line in the HTTP GET

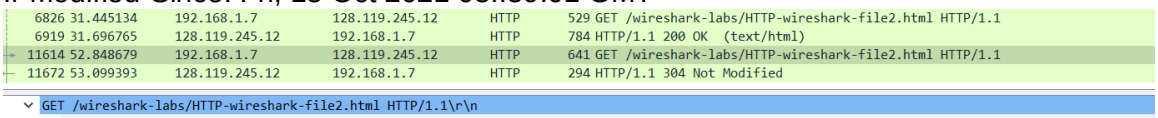


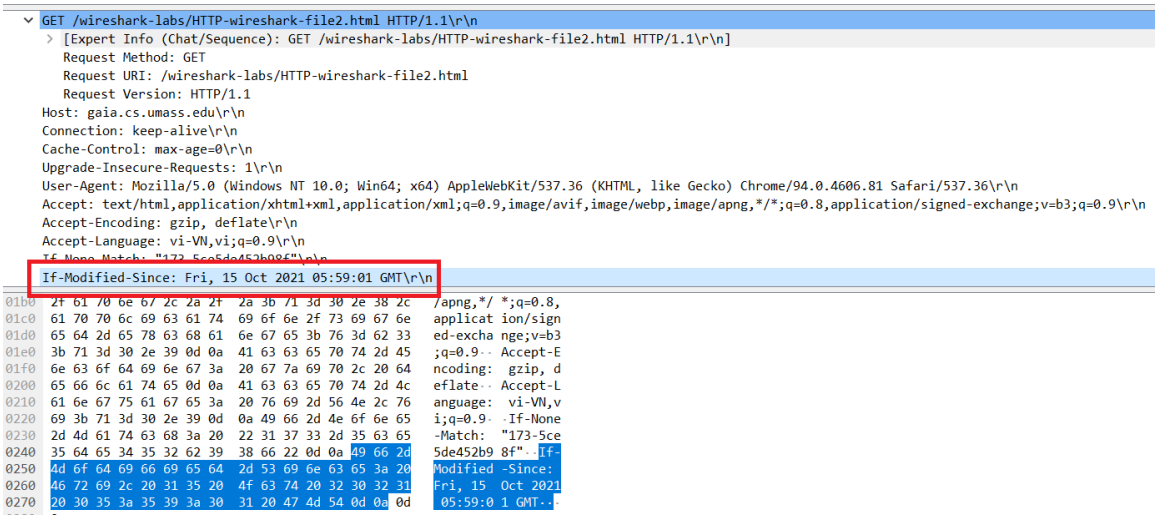
9. Inspect the contents of the server response. Did the server explicitly return the contents of the file? How can you tell?
   Ans: Yes, the server explicitly returned the contents of the file by “Line-based text data” section





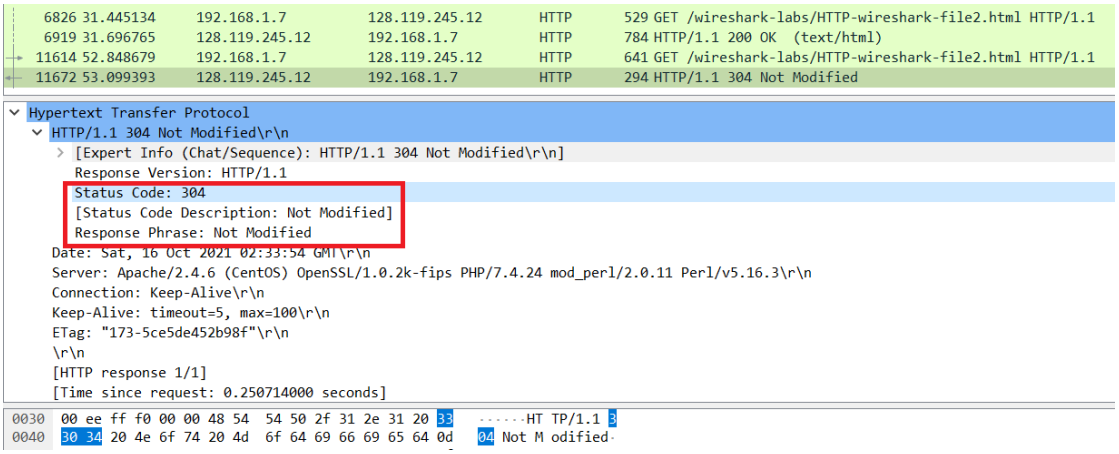
10. Now inspect the contents of the second HTTP GET request from your browser to the server. Do you see an “IF-MODIFIED-SINCE:” line in the HTTP GET? If so, what information follows the “IF-MODIFIED-SINCE:” header?
    Ans: Yes, I see an “IF-MODIFIED-SINCE:” line the HTTP GET.
    If-Modified-Since: Fri, 15 Oct 2021 05::59:01 GMT





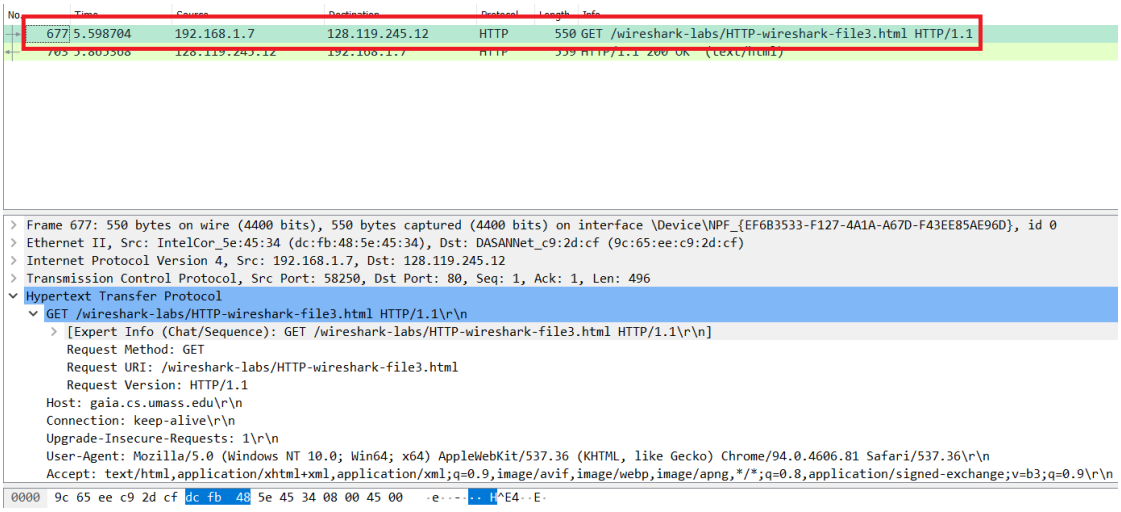
11. What is the HTTP status code and phrase returned from the server in response to this second HTTP GET? Did the server explicitly return the contents of the file? Explain.

Ans:
HTTP status code is 304.
Response phrase: Not Modified.
The server did not explicitly return the contents of the file because there is no change in the file.

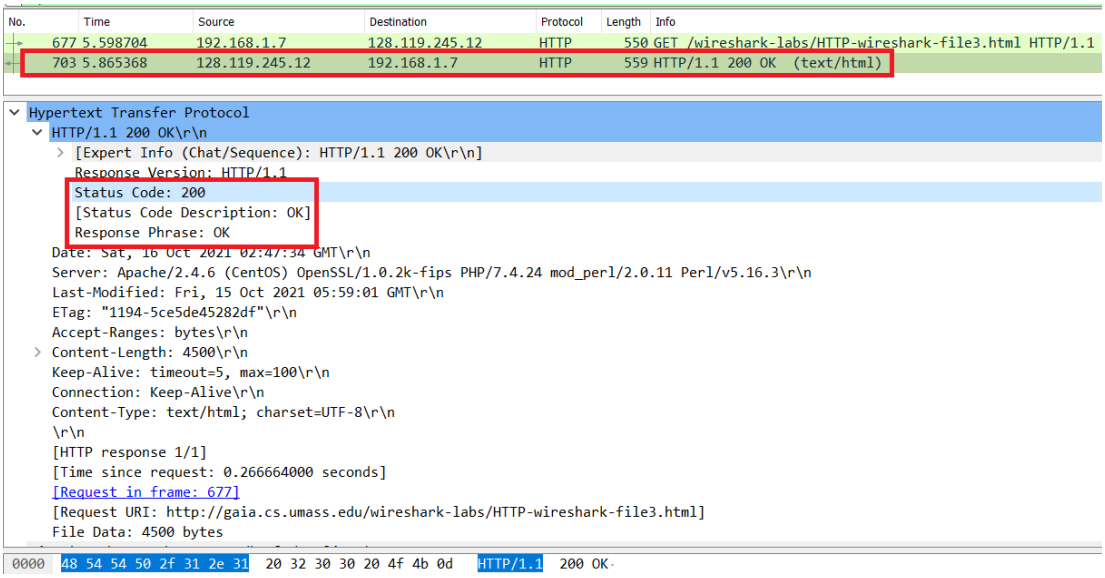


III/ Retrieving Long Documents

12. How many HTTP GET request messages did your browser send? Which packet number in the trace contains the GET message for the Bill or Rights?
    Ans: Only 1 HTTP GET request message was sent. The packet number in the trace contains the GET message for the Bill or Rights is 677

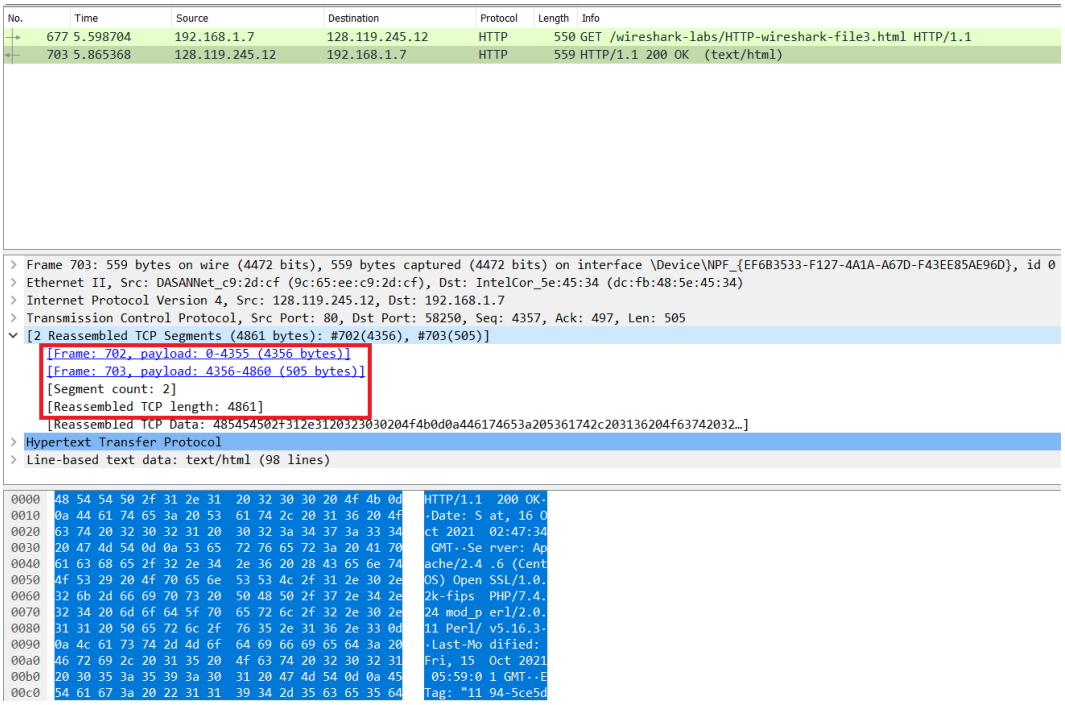


13. Which packet number in the trace contains the status code and phrase associated with the response to the HTTP GET request?
    Ans: The packet number in the trace contains the status code and phrase associated with the response to the HTTP GET request: 703



14. What is the status code and phrase in the response?
    Ans: Using the figure above.
    Status Code: 200 and Response Phrase: OK.

15. How many data-containing TCP segments were needed to carry the single HTTP response and the text of the Bill of Rights?
    Ans: 2 data-containing TCP segments were needed



IV/ HTML Documents with Embedded Objects

16. How many HTTP GET request messages did your browser send? To which Internet addresses were these GET requests sent?
    Ans: There were 3 HTTP GET request messages sent.
    The intial page “HTTP-wideshark-file4.html” and the figure “pearson.png” were sent to the IP address: 128.119.245.12.
    The figure “8E_cover_small.jpg” was sent to the IP address: 178.79.137.164.

No.	Time	Source	Destination	Protocol	Length	Info
519	3.777938	192.168.1.7	128.119.245.12	HTTP	550	GET /wireshark-labs/HTTP-wireshark-file4.html HTTP/1.1
574	4.042686	128.119.245.12	192.168.1.7	HTTP	1355	HTTP/1.1 200 OK (text/html)
586	4.090306	192.168.1.7	128.119.245.12	HTTP	496	GET /pearson.png HTTP/1.1
635	4.365195	128.119.245.12	192.168.1.7	HTTP	761	HTTP/1.1 200 OK (PNG)
684	4.596337	192.168.1.7	178.79.137.164	HTTP	463	GET /8E_cover_small.jpg HTTP/1.1
708	4.801853	178.79.137.164	192.168.1.7	HTTP	225	HTTP/1.1 301 Moved Permanently

17. Can you tell whether your browser downloaded the two images serially, or whether they were downloaded from the two websites in parallel? Explain.
    Ans: By checking the TCP port, I can see that the 2 images were received over 2 different TCP port connections which are 54087 and 64414. Therefore, they were downloaded serially.

No.	Time	Source	Destination	Protocol	Length	Info
519	3.777938	192.168.1.7	128.119.245.12	HTTP	550	GET /wireshark-labs/HTTP-wireshark-file4.html HTTP/1.1
574	4.042686	128.119.245.12	192.168.1.7	HTTP	1355	HTTP/1.1 200 OK (text/html)
586	4.090306	192.168.1.7	128.119.245.12	HTTP	496	GET /pearson.png HTTP/1.1
635	4.365195	128.119.245.12	192.168.1.7	HTTP	761	HTTP/1.1 200 OK (PNG)
684	4.596337	192.168.1.7	178.79.137.164	HTTP	463	GET /8E_cover_small.jpg HTTP/1.1
708	4.801853	178.79.137.164	192.168.1.7	HTTP	225	HTTP/1.1 301 Moved Permanently

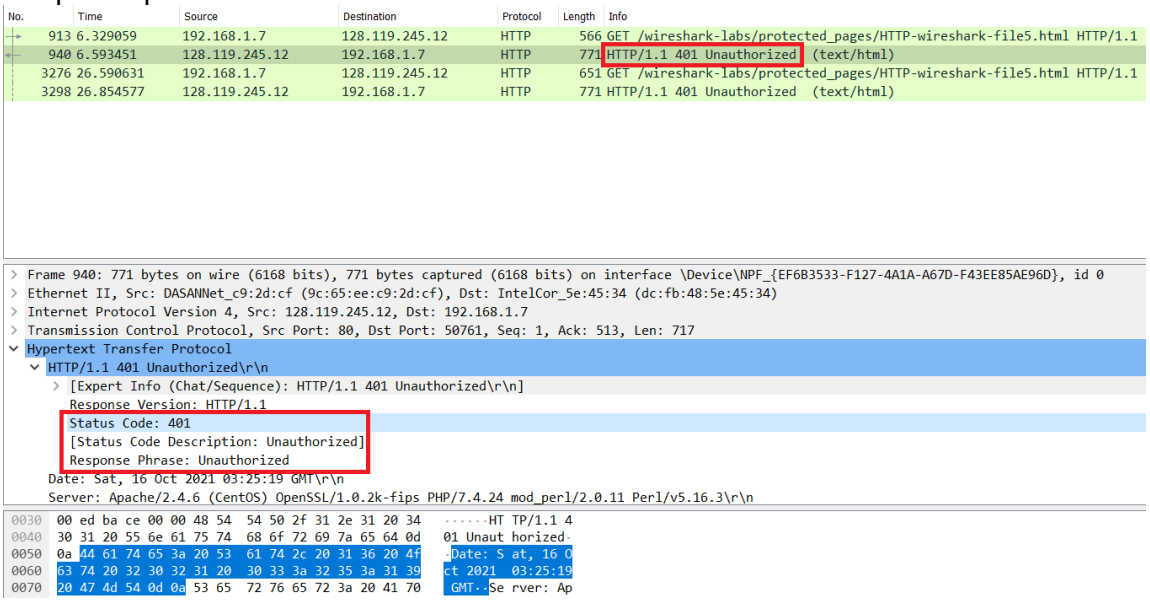
```
> Frame 586: 496 bytes on wire (3968 bits), 496 bytes captured (3968 bits) on interface \Device\NPF_{EF6B3533-F127-4A1A-A67D-F43EE85AE96D}, id 0
> Ethernet II, Src: IntelCor_5e:45:34 (dc:fb:48:5e:45:34), Dst: DASANNet_c9:2d:cf (9c:65:ee:c9:2d:cf)
> Internet Protocol Version 4, Src: 192.168.1.7, Dst: 128.119.245.12
v Transmission Control Protocol, Src Port: 54087, Dst Port: 80, Seq: 497, Ack: 1302, Len: 442
    Source Port: 54087
    Destination Port: 80
    [Stream index: 3]
    [TCP Segment Len: 442]
    Sequence Number: 497    (relative sequence number)
    Sequence Number (raw): 2487044650
    [Next Sequence Number: 939    (relative sequence number)]
    Acknowledgment Number: 1302    (relative ack number)
    Acknowledgment number (raw): 1245562003
0020  f5 0c d3 47 00 50 94 3d  4a 2a 4a 3d c4 93 50 18   ···G·P·= J*J=··P·
```

No.	Time	Source	Destination	Protocol	Length	Info
519	3.777938	192.168.1.7	128.119.245.12	HTTP	550	GET /wireshark-labs/HTTP-wireshark-file4.html HTTP/1.1
574	4.042686	128.119.245.12	192.168.1.7	HTTP	1355	HTTP/1.1 200 OK (text/html)
586	4.090306	192.168.1.7	128.119.245.12	HTTP	496	GET /pearson.png HTTP/1.1
635	4.365195	128.119.245.12	192.168.1.7	HTTP	761	HTTP/1.1 200 OK (PNG)
684	4.596337	192.168.1.7	178.79.137.164	HTTP	463	GET /8E_cover_small.jpg HTTP/1.1
708	4.801853	178.79.137.164	192.168.1.7	HTTP	225	HTTP/1.1 301 Moved Permanently

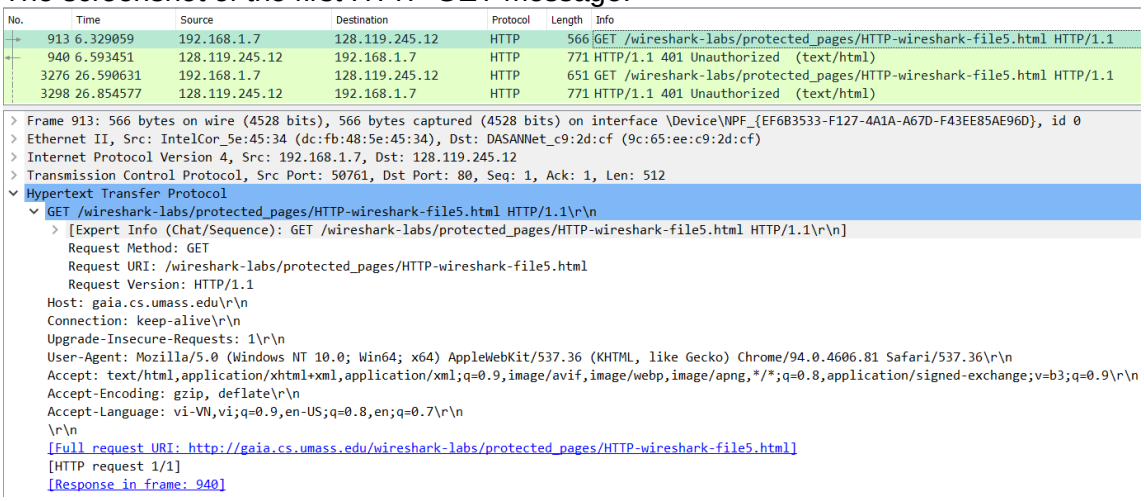
```
> Frame 684: 463 bytes on wire (3704 bits), 463 bytes captured (3704 bits) on interface \Device\NPF_{EF6B3533-F127-4A1A-A67D-F43EE85AE96D}, id 0
> Ethernet II, Src: IntelCor_5e:45:34 (dc:fb:48:5e:45:34), Dst: DASANNet_c9:2d:cf (9c:65:ee:c9:2d:cf)
> Internet Protocol Version 4, Src: 192.168.1.7, Dst: 178.79.137.164
v Transmission Control Protocol, Src Port: 64414, Dst Port: 80, Seq: 1, Ack: 1, Len: 409
    Source Port: 64414
    Destination Port: 80
    [Stream index: 8]
    [TCP Segment Len: 409]
    Sequence Number: 1    (relative sequence number)
    Sequence Number (raw): 3705028877
    [Next Sequence Number: 410    (relative sequence number)]
    Acknowledgment Number: 1    (relative ack number)
    Acknowledgment number (raw): 2675106424
0020  89 a4 fb 9e 00 50 dc d6  41 0d 9f 72 e2 78 50 18   ·····P·· A··r·xP·
```

V/ HTTP Authentication

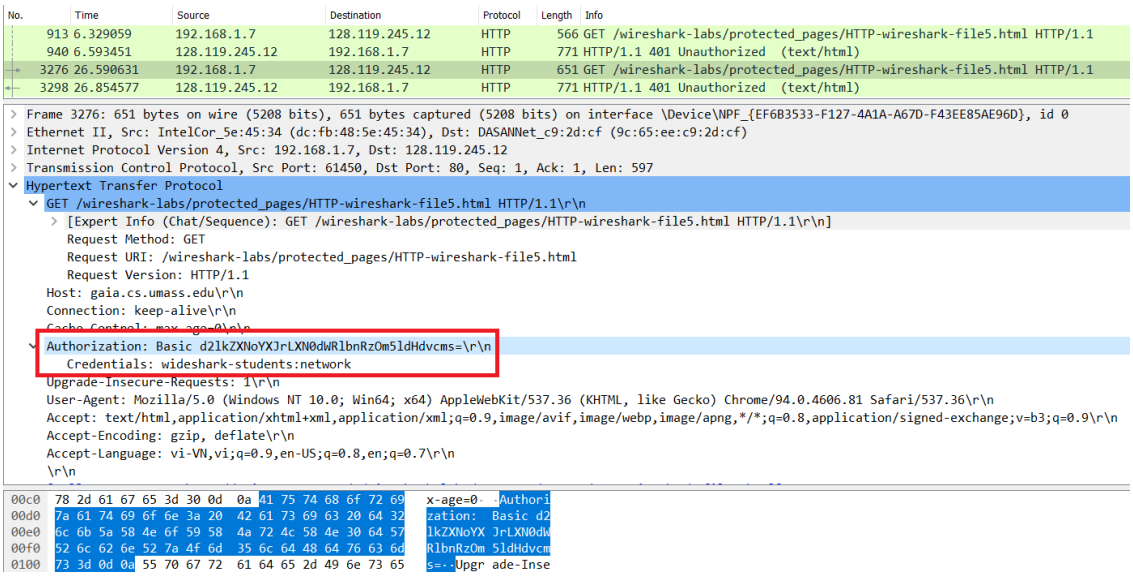
18. What is the server's response (status code and phrase) in repsonse to the intial HTTP GET message from your browser?
    Ans: We get a response: 'HTTP/1.1 401 Unauthorized'.
    Status code: 401.
    Response phrase: Unauthorized.



19. When your browser sends the HTTP GET message for the second time, what new field is included in the HTTP GET message?
    Ans:
    The screenshot of the first HTTP GET message:



    The screenshot of the second HTTP GET message:



Comparing these two HTTP GET messages, it is easy to find that the second HTTP GET message contains the ‘Authorization’ field.
The username (wideshark-students) and password (network) that I entered are encoded in the string of characters (d2lyZXNoYXJrLXN0dWRlbnRzOm5ldHdvcms=) following the “Authorization: Basic” header in the client’s HTTP GET message.