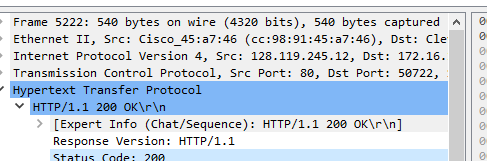
1. Is your browser running HTTP version 1.0 or 1.1? What version of HTTP is the server running?

http 1.1



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

en-US



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

Computer ip address:172.16.10.40

gaia.cs.umass.edu ip address: 128.119.245.12



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

Status code: 200

5. When was the HTML file that you are retrieving last modified at the server?



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

128

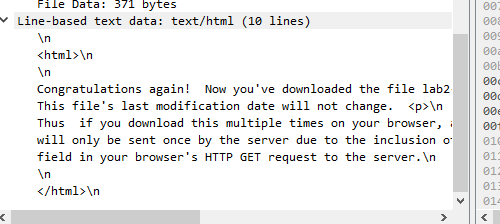
**Task 2 (The HTTP Conditional GET/response interaction)**

1. 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?

NO

1. Inspect the contents of the server response. Did the server explicitly return the contents of the file?  How can you tell?

YES IT DID



1. 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?

YES



1. 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.

CODE

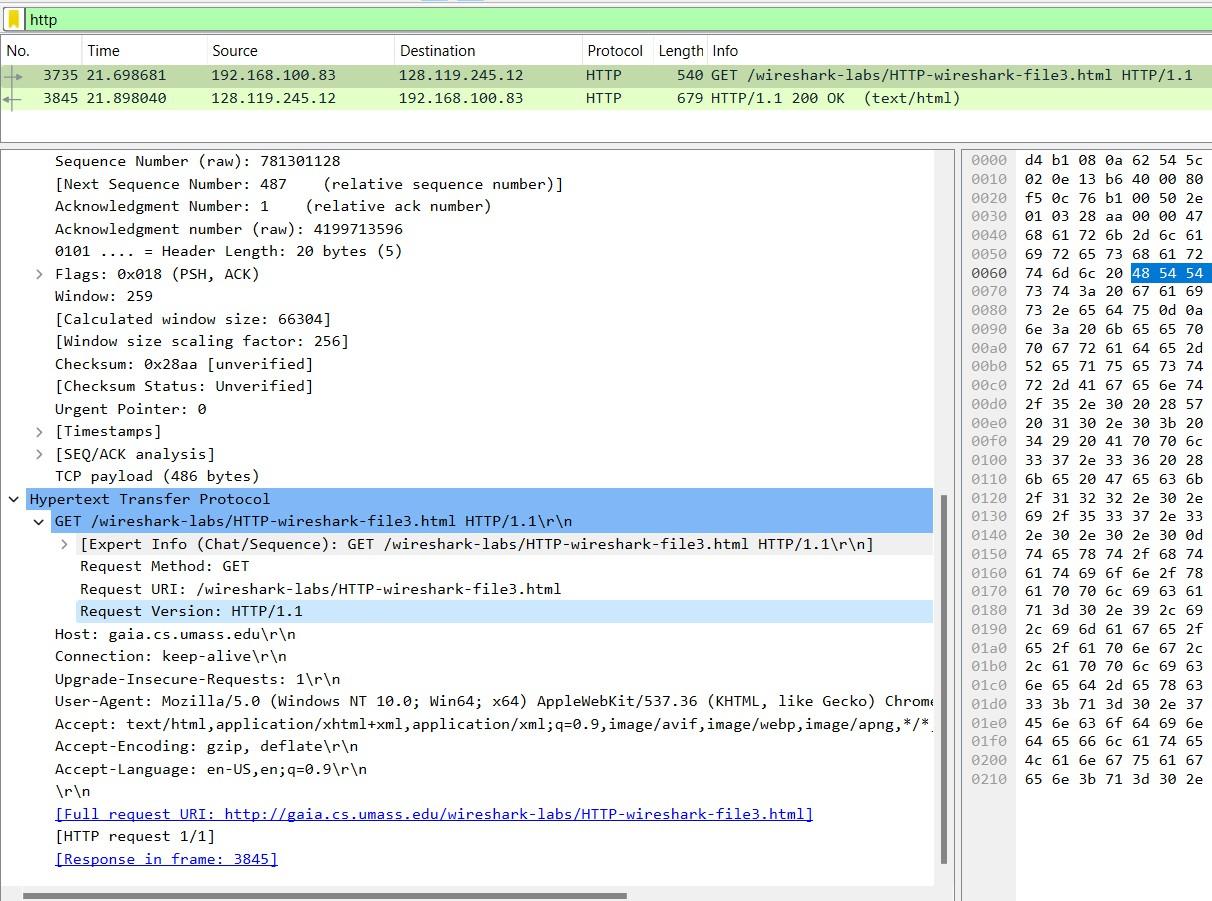
304 NOT MODIFIED



1. Retrieving Long Documents

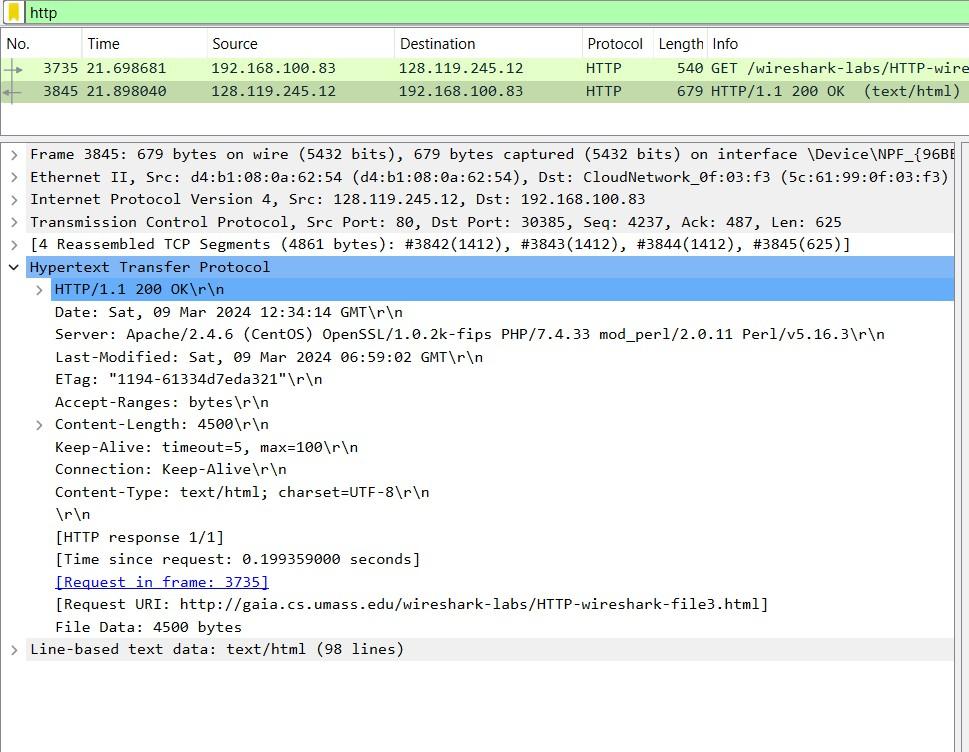
How many HTTP GET request messages did your browser send? Which packet number in the trace contains the GET message for the Bill or Rights?

One GET request message was sent. Packet number: 3735



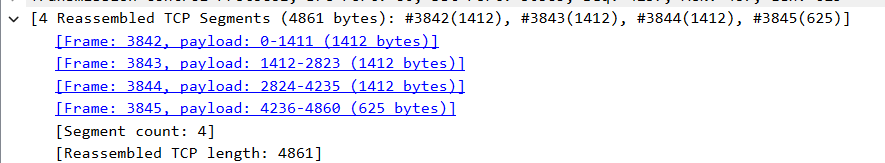
1. Which packet number in the trace contains the status code and phrase associated with the response to the HTTP GET request?

Packet number 3845



1. What is the status code and phrase in the response? status code: 200, phrase: OK
2. How many data-containing TCP segments were needed to carry the single HTTP response and the text of the Bill of Rights?

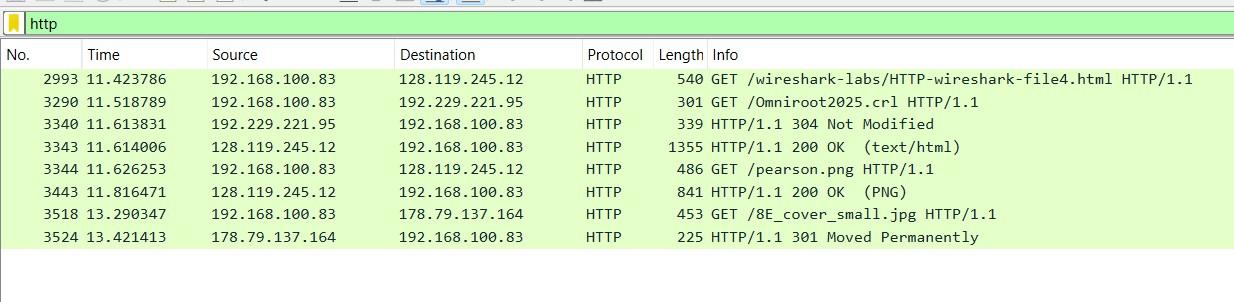
4 TCP segments were required.



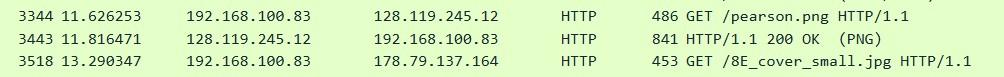
1. HTML Documents with Embedded Objects

How many HTTP GET request messages did your browser send? To which Internet addresses were these GET requests sent?

Four GET request messages were sent.

Internet addresses can be seen under “Destination” in the screenshot below with each GET request.

1. Can you tell whether your browser downloaded the two images serially, or whether they were downloaded from the two websites in parallel? Explain.



As seen in the screenshot above, the second image was retrieved after approximately 2 seconds of the first image retrieval. Hence, the images were retrieved serially.

1. HTTP Authentication

What is the server’s response (status code and phrase) in response to the initial HTTP GET message from your browser?

Response code: 401, phrase: Unauthorized.

1. When your browser’s sends the HTTP GET message for the second time, what new field is included in the HTTP GET message?

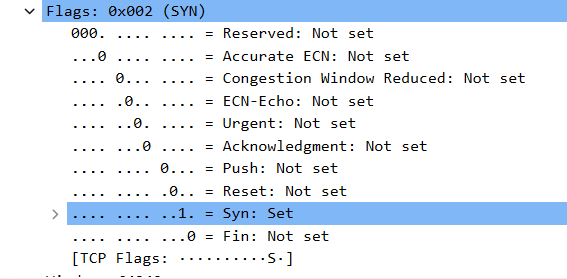


My credentials are sent the second time.

1. A First Look at The Captured Trace

What is the sequence number of the TCP SYN segment that is used to initiate the TCP connection between the client computer and gaia.cs.umass.edu? What is it in the segment that identifies the segment as a SYN segment?

Sequence number: 291. They SYN flag is set in the packet header.



1. What is the sequence number of the SYNACK segment sent by gaia.cs.umass.edu to the client computer in reply to the SYN? What is the value of the Acknowledgement field in the SYNACK segment? How did gaia.cs.umass.edu determine that value? What is it in

the segment that identifies the segment as a SYNACK segment?

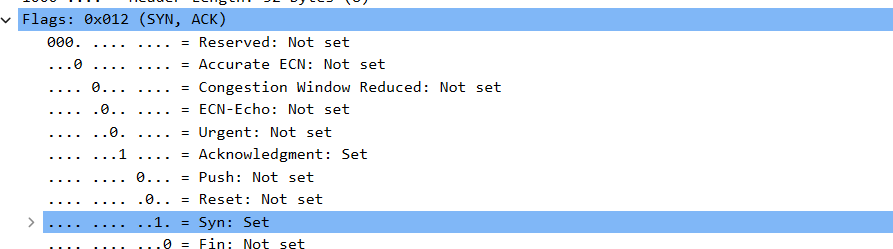
Sequence number: 291.



This acknowledgement number was identified by the initial sequence number of the GET request message which was:



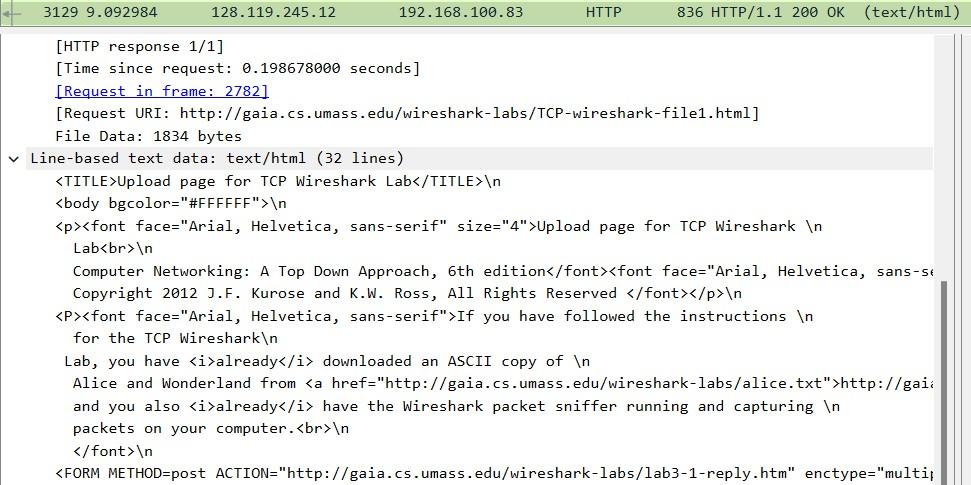
The syn and ACK flags are set which identifies the packet as SYNACK



1. What is the sequence number of the TCP segment containing the HTTP POST

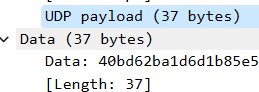
command? Note that in order to find the POST command, you’ll need to dig into the packet content field at the bottom of the Wireshark window, looking for a segment with a “POST” within its DATA field.

Sequence number: 3129.



UDP Packets

1. By consulting the displayed information in Wireshark’s packet content field for this packet, determine the length (in bytes) of each of the UDP header fields.



1. The value in the Length field is the length of what? (You can consult the text for this answer). Verify your claim with your captured UDP packet.

It is the length of the data being sent or received (in bytes).

1. What is the maximum number of bytes that can be included in a UDP payload? (Hint: the answer to this question can be determined by your answers above).

65527 bytes

1. What is the largest possible source port number? (Hint: see the hint provided in above question).

UDP port numbers can be between 0 and 65,535.

1. What is the protocol number for UDP? Give your answer in both hexadecimal and decimal notation. To answer this question, you’ll need to look into the Protocol field of the IP datagram containing this UDP segment?



1. Examine a pair of UDP packets in which your host sends the first UDP packet and the second UDP packet is a reply to this first UDP packet. (Hint: for a second packet to be sent in response to a first packet, the sender of the first packet should be the destination of the second packet). Describe the relationship between the port numbers in the two packets.

Relationship between the ports, First packet:



Second packet:

