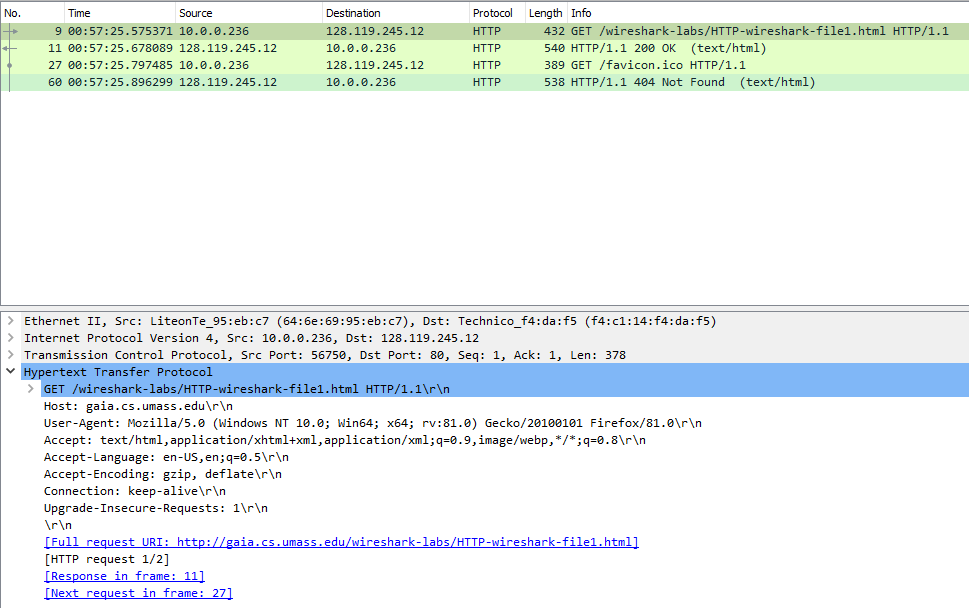
Anthony Chavez

Professor Sun

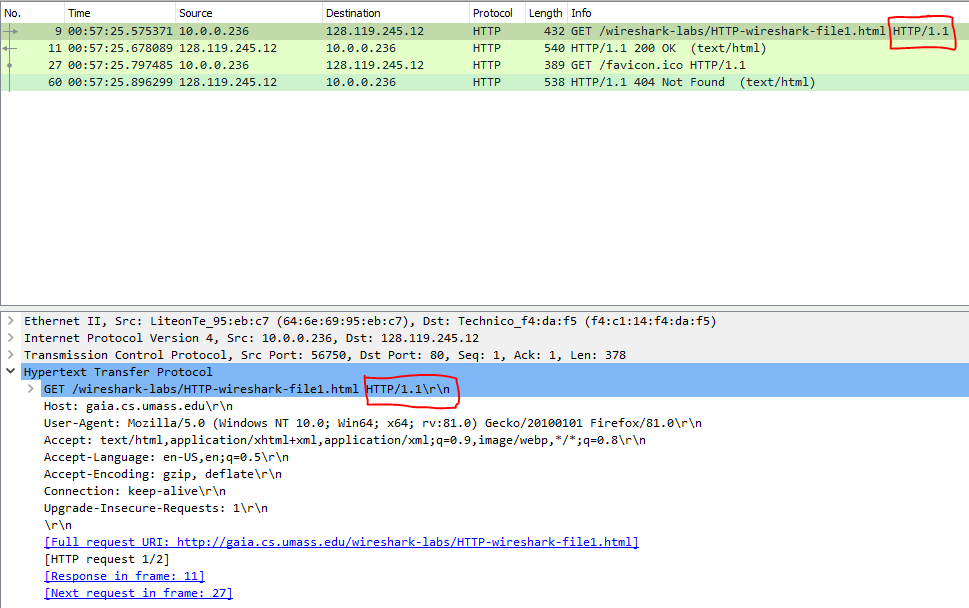
Lab 2 – Wireshark – HTTP Lab

The Basic HTTP Get/response interaction

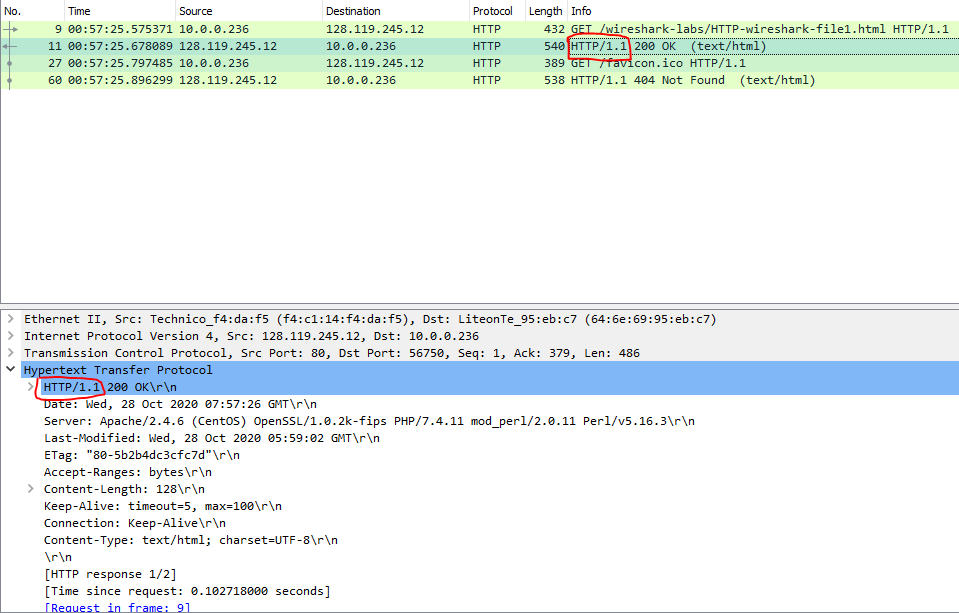


Here are the http packets I captured following the provided instructions. Please ignore packets 27 and 60 as they do not pertain to the scope of this experiment.

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

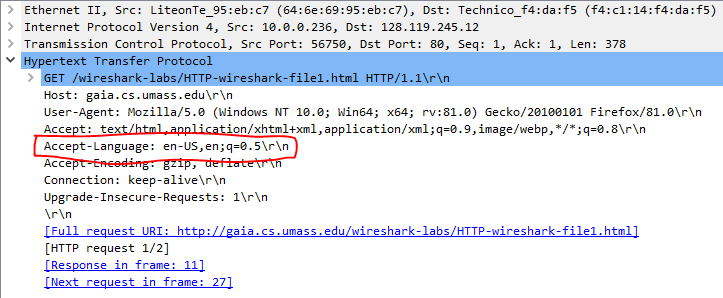


* 1. As marked by the red rectangles in the image above, my browser is running HTTP version 1.1



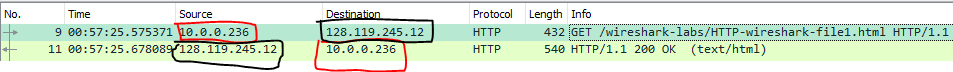
* 1. As marked by the red rectangles in the image above, the server is running HTTP 1.1

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



* 1. en-US, indicated by the red rectangle in the image above.

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



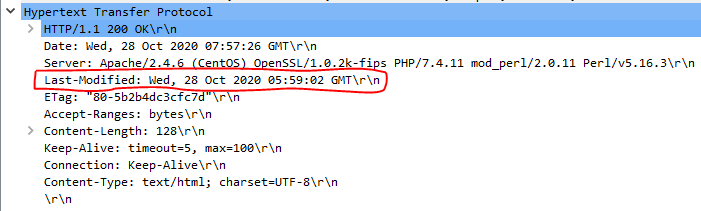
* 1. My computer is IP address is 10.0.0.236 (red rectangles) and the server IP address is 128.119.245.12 (black rectangles).

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



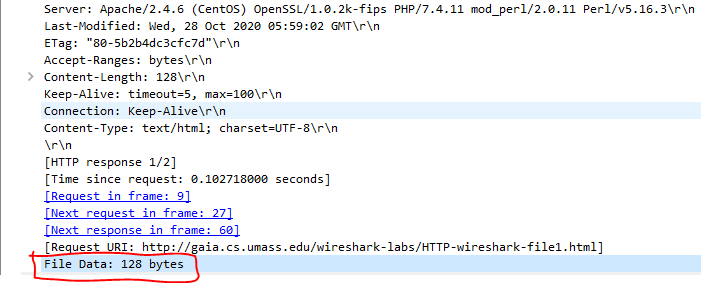
* 1. 200 OK is the status code returned from the server to my browser.

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



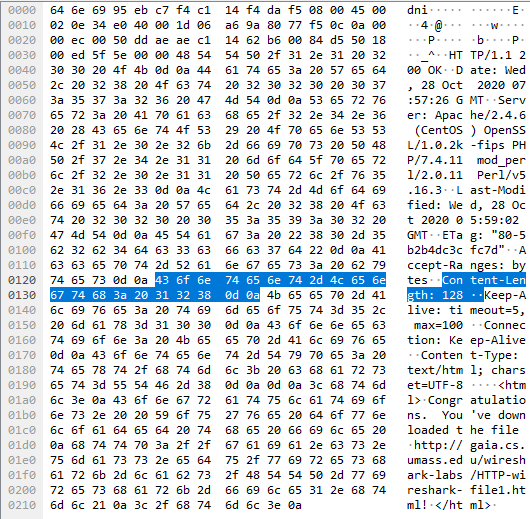
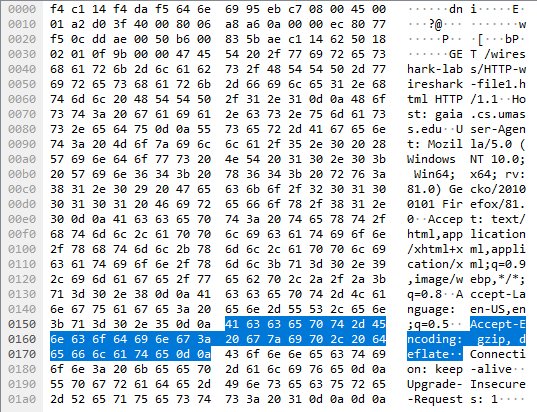
* 1. The HTML file was last modified on Wednesday, 28 Oct 2020 05:59:02 GMT. See red rectangle in the image above.

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



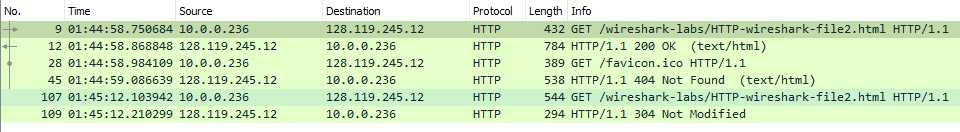
* 1. 128 bytes of content are being returned to my browser as shown in the above image.

1. 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.



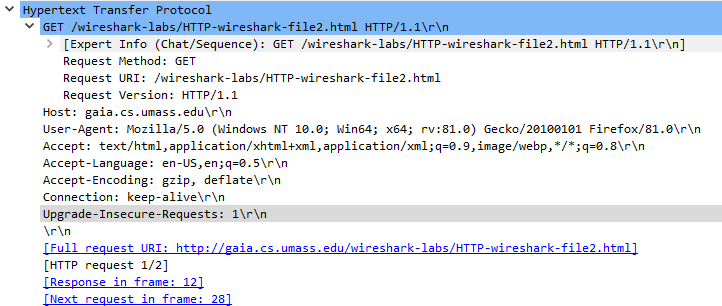
* 1. No additional headers within the data that are not displayed in the packet-listing window. Raw data of GET packet on the left and raw data of 200 OK packet on the right.

The HTTP Conditional GET/response interaction



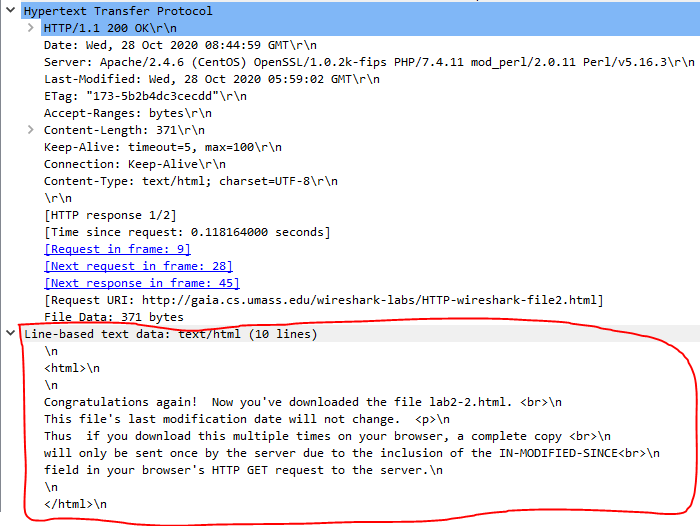
Here are the http packets I captured following the provided instructions. Please ignore packets 28 and 45 as they do not pertain to the scope of this experiment.

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?



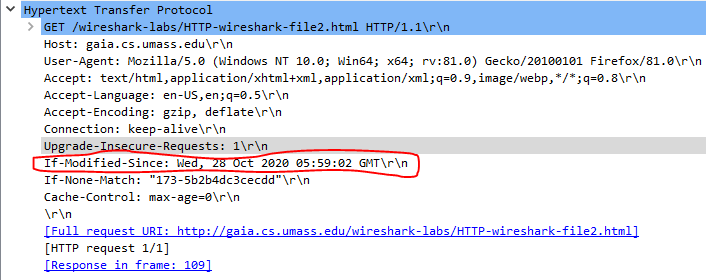
* 1. As can be seen in the above image, the first HTTP GET request from the browser to the server does not contain an “IF-MODIFIED-Since:” line in the HTTP GET.

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



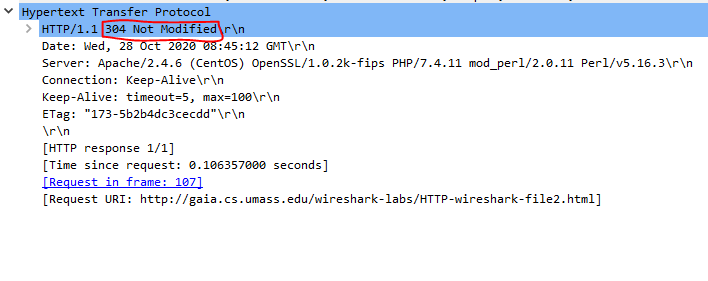
* 1. The server explicitly returned the contents of the file. Refer to the Line-based text data segment indicated by the red rectangle in the above image.

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?



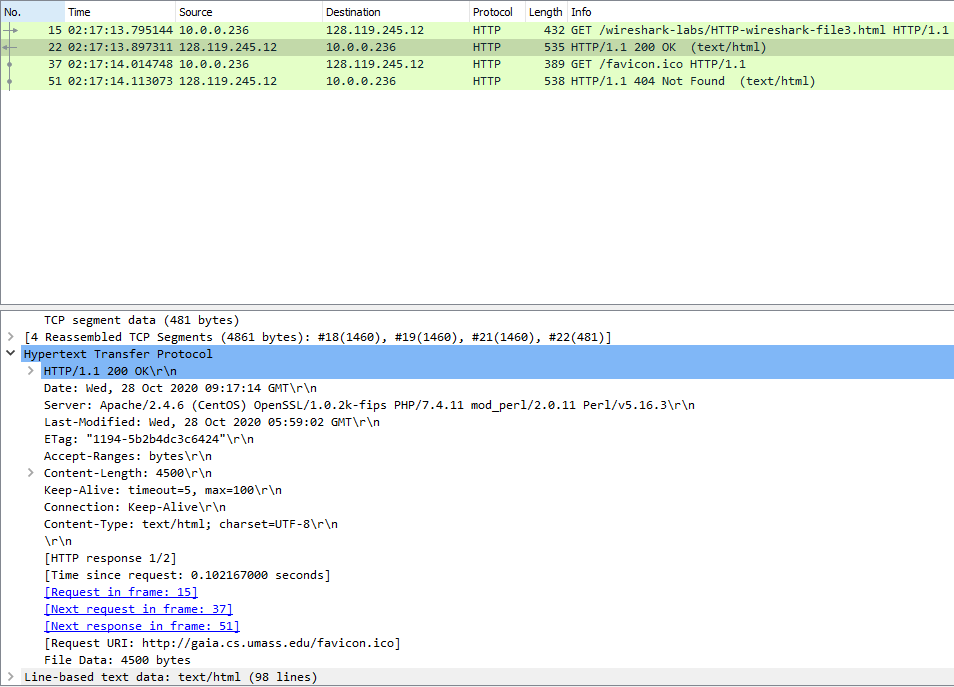
* 1. As we can see in the above image, the second HTTP GET request from the browser to the server does contain an “IF-MODIFIED-SINCE:” line in the HTTP GET. The information that follows the “IF-MODIFIED-SINCE:” header is the time condition of when to update the cached object.

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.



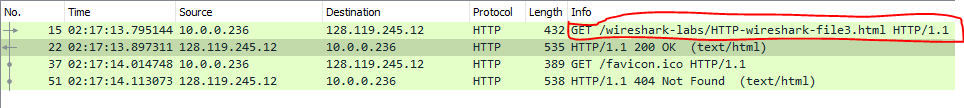
* 1. The HTTP status code and phrase returned from the server in response to this second HTTP GET is “304 Not Modified”. Since the “If-Modified-Since:” time condition was not met, the cached object is considered up to date. We avoided meeting this condition by quickly refreshing the page. Therefore, the server did not explicitly return the contents of the file for the second HTTP GET.

Retrieving Long Documents



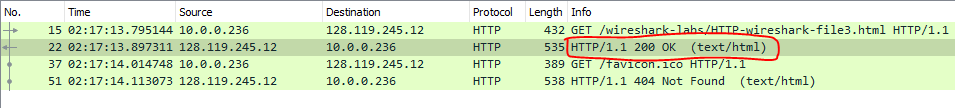
Here are the http packets I captured following the provided instructions. Please ignore packets 37 and 51 as they do not pertain to the scope of this experiment.

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



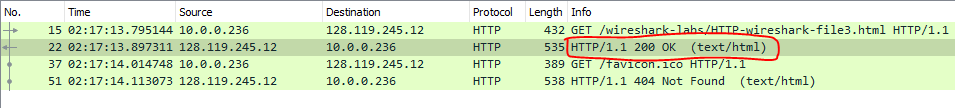
* 1. My browser sent only 1 HTTP GET request message. The packet number in the trace that contains the GET message for the Bill of Rights is 15.

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



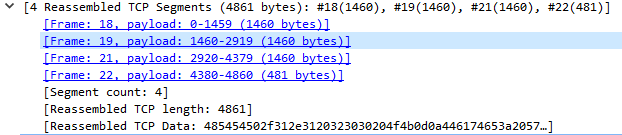
* 1. Packet number 22 contains the status code and phrase associated with the response to the HTTP GET request.

1. What is the status code and phrase in the response?



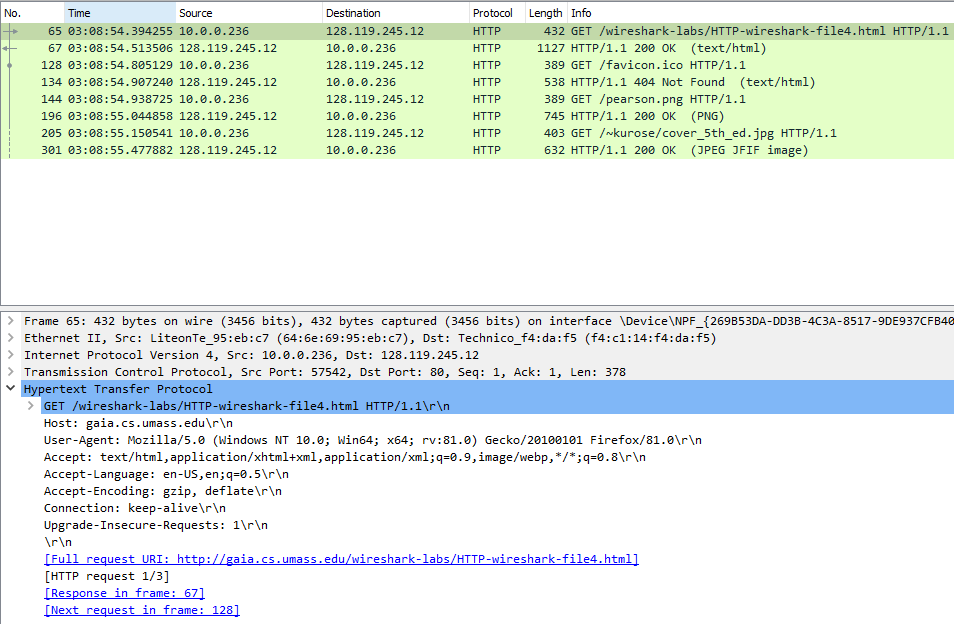
* 1. The status code and phrase in the response is 200 and OK.

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



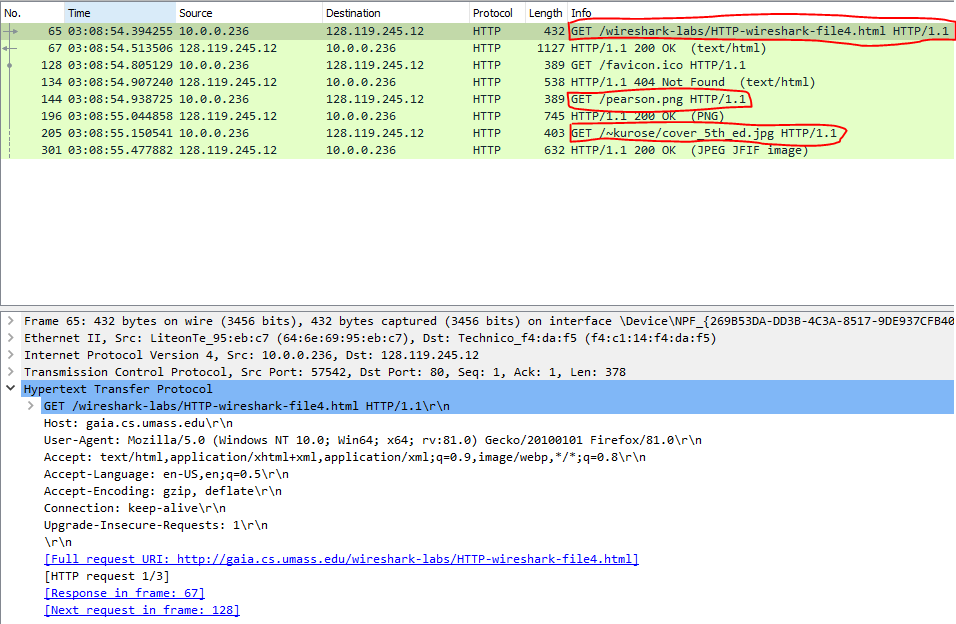
* 1. 4 data-containing TCP segments were needed to carry the single HTTP response and the text of the Bill of Rights.

HTML Documents with Embedded Objects



Here are the http packets I captured following the provided instructions. Please ignore packets 128 and 134 as they do not pertain to the scope of this experiment.

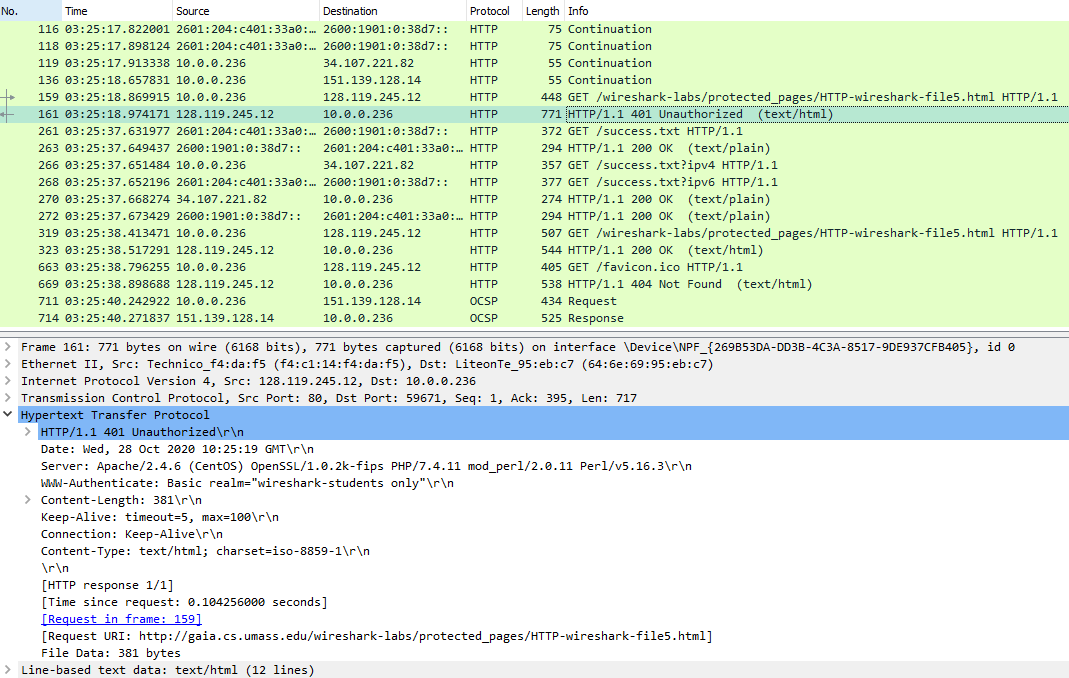
1. How many HTTP GET request messages did your browser send? To which Internet addresses were these GET requests sent?



* 1. A total of 3 HTTP GET request messages were sent by my browser.
     1. Wireshark-labs/HTTP-wireshark-file4.html
     2. Pearson.png
     3. ~kurose/cover\_5th\_ed.jpg

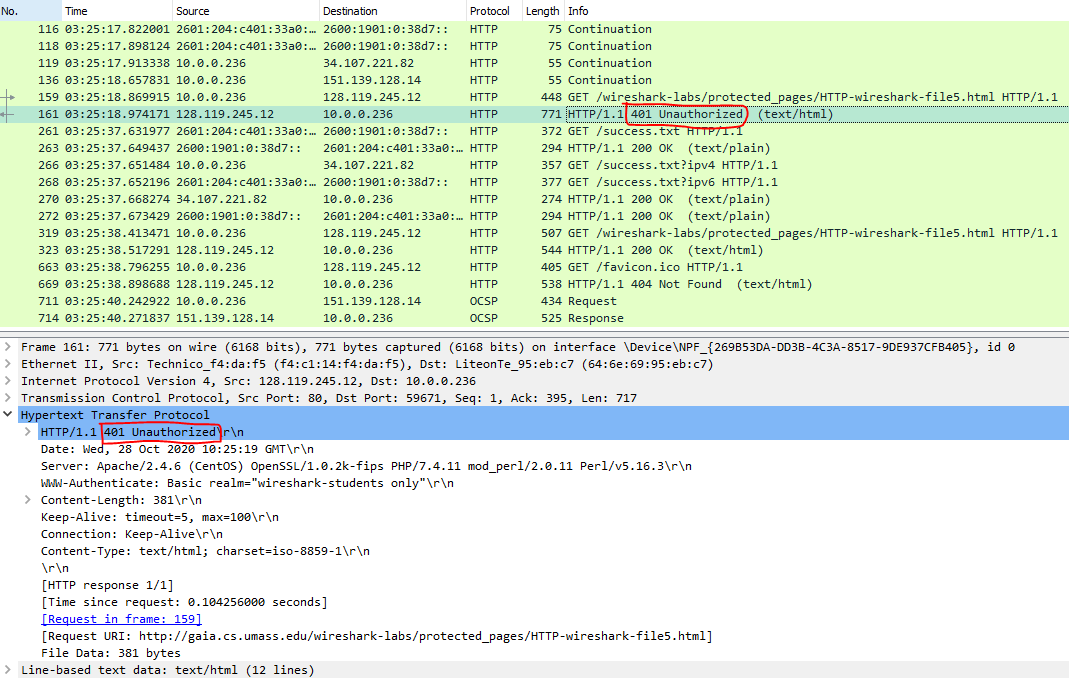
1. Can you tell whether your browser downloaded the two images serially, or whether they were downloaded from the two web sites in parallel? Explain.
   1. The browser downloaded the two images serially. The images could not have been downloaded from the two web sites in parallel as shown in the packet listing window. The browser sent an HTTP GET request packet for the pearson.png first and after receiving the image, only then sent another HTTP GET request packet for the second image.

HTTP Authentication



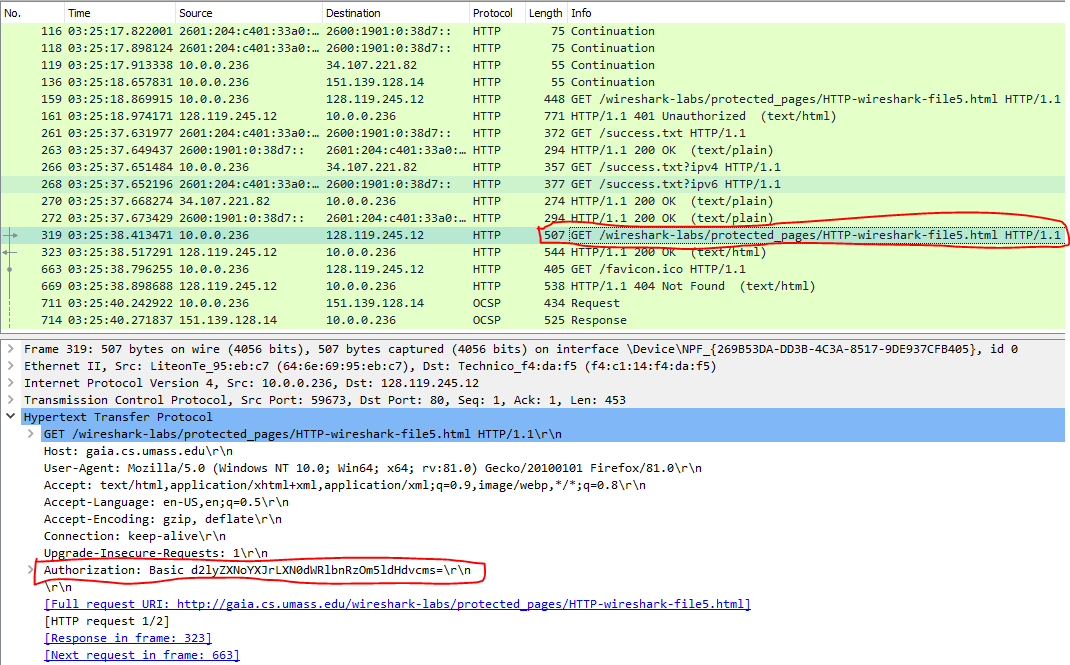
Here are the http packets I captured following the provided instructions. Please ignore packets 663 and 669 as they do not pertain to the scope of this experiment.

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



* 1. The server’s response in response to the initial HTTP GET message from my browser is “401 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?



* 1. When the browser sends the HTTP GET message for the second time, the new field included in the HTTP GET message is “Authorization: Basic”