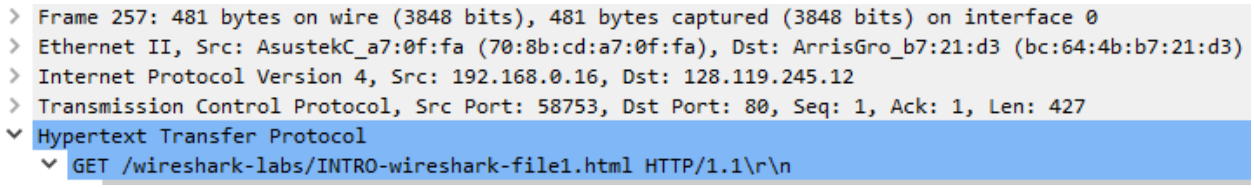
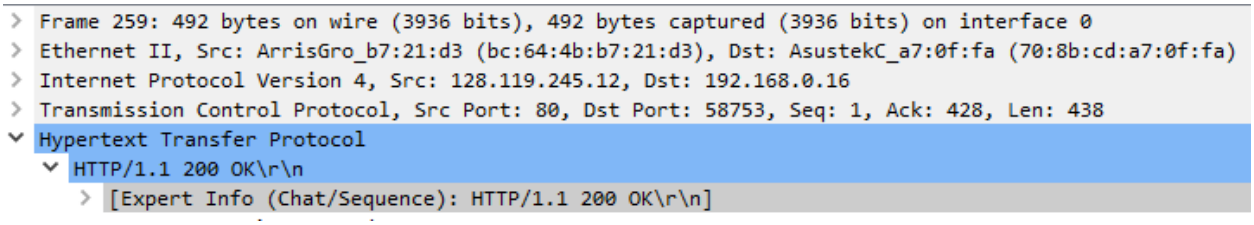
Erin Alltop
CS372 – Spring 2018
Lab 2

1. My browser is running HTTP version 1.1 as seen in the screenshot below:



This is the first part of the GET request. On the last line you can see it says “HTTP/1.1” indicating the HTTP version the browser is running.

The version of HTTP the server is running is also 1.1 as see in the screenshot below:



This is the first part of the OK message. On the last line you can see it says “HTTP/1.1” indicating the HTTP version the server is running.

2. My browser indicates that it can accept the en-US language to the server, as shown below:

```
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8\r\n
Accept-Encoding: gzip, deflate\r\n
Accept-Language: en-US,en;q=0.9\r\n
```

Note the last line indicating en-US as an accepted language.

3. The IP address of my computer is 192.168.0.16 and the IP address of the gaia.cs.umass.edu server is 128.119.245.12. Shown in the screenshot below.

```
Internet Protocol Version 4, Src: 192.168.0.16, Dst: 128.119.245.12
```

This is taken from the GET message. Source is my computer and the Destination is the server.

```
Internet Protocol Version 4, Src: 128.119.245.12, Dst: 192.168.0.16
```

This is taken from the OK message. Source is the server, and the Destination is my computer.

4. The status code returned to my computer is 200, or “OK”.

```
Request Version: HTTP/1.1
Status Code: 200
[Status Code Description: OK]
Response Phrase: OK
```

Status Code taken from the OK message from the server.

5. The retrieved HTML file was last modified Sunday, April 29, 2018 at 5:59:01 GMT.

```
Last-Modified: Sun, 29 Apr 2018 05:59:01 GMT\r\n
```

Screenshot taken from the OK message indicating last modified time.

6. 81 byes were returned to my computer from the server:

```
File Data: 81 bytes
```

Screenshot taken from the OK message indicating size of file.

7. No, the headers within the data appear to match what is listed in the packet-listing window exactly.

8. No, I cannot see any IF-MODIFIED-SINCE line in the HTTP GET of the first request.

```
Hypertext Transfer Protocol
  GET /wireshark-labs/HTTP-wireshark-file2.html HTTP/1.1\r\n
    [Expert Info (Chat/Sequence): GET /wireshark-labs/HTTP-wireshark-file2.html HTTP/1.1\r\n]
    Request Method: GET
    Request URI: /wireshark-labs/HTTP-wireshark-file2.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 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/65.0.3325.181 Safari/537.36\r\n
  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8\r\n
  Accept-Encoding: gzip, deflate\r\n
  Accept-Language: en-US,en;q=0.9\r\n
  \r\n
  [Full request URI: http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file2.html]
  [HTTP request 1/1]
  [Response in frame: 24]
```

No IF-MODIFIED-SINCE line to be found!

9. Yes, the server explicitly returned the contents of the file. It is shown on the last portion of the HTTP OK message, seen here:

```
Line-based text data: text/html
    \n
    <html>\n
    \n
    Congratulations again!  Now you've downloaded the file lab2-2.html. <br>\n
    This file's last modification date will not change.  <p>\n
    Thus  if you download this multiple times on your browser, a complete copy <br>\n
    will only be sent once by the server due to the inclusion of the IN-MODIFIED-SINCE<br>\n
    field in your browser's HTTP GET request to the server.\n
    \n
    </html>\n
```

Last portion of the first OK message from the server in the HTTP OK.

10. Yes, the second GET request has the IF-MODIFIED-SINCE line. It appears to show the last modified date of the first request sent.

```
If-Modified-Since: Sun, 29 Apr 2018 05:59:01 GMT\r\n
\r\n
[Full request URI: http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file2.html]
[HTTP request 1/2]
[Response in frame: 106]
```

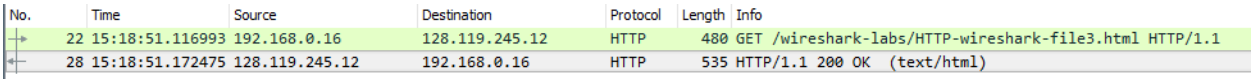
IF-MODIFIED-SINCE message in the second GET request.

11. The second HTTP status code returned from the server is 304, or “Not Modified.” There is not a section that explicitly returns the contents of the file. This makes sense because there is no need to resend a packet with the data if nothing has changed. This step of requesting if anything has been modified will free up time and bandwidth by not resending unnecessary data.

```
Hypertext Transfer Protocol
  HTTP/1.1 304 Not Modified\r\n
    [Expert Info (Chat/Sequence): HTTP/1.1 304 Not Modified\r\n]
        [HTTP/1.1 304 Not Modified\r\n]
        [Severity level: Chat]
        [Group: Sequence]
      Request Version: HTTP/1.1
      Status Code: 304
      [Status Code Description: Not Modified]
      Response Phrase: Not Modified
  Date: Sun, 29 Apr 2018 18:49:58 GMT\r\n
  Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/5.4.16 mod_perl/2.0.10 Perl/v5.16.3\r\n
  Connection: Keep-Alive\r\n
  Keep-Alive: timeout=5, max=100\r\n
  ETag: "173-56af66f0d0abc"\r\n
  \r\n
  [HTTP response 1/2]
  [Time since request: 0.054729000 seconds]
  [Request in frame: 104]
  [Next response in frame: 109]
```

The second response from the server indicates a status of “Not Modified” and does not return the explicit contents of the file as it is not necessary if nothing has been modified.

12. My browser sent one GET message to the server. Packet number 22 in the trace contains the GET message for the Bill of Rights.

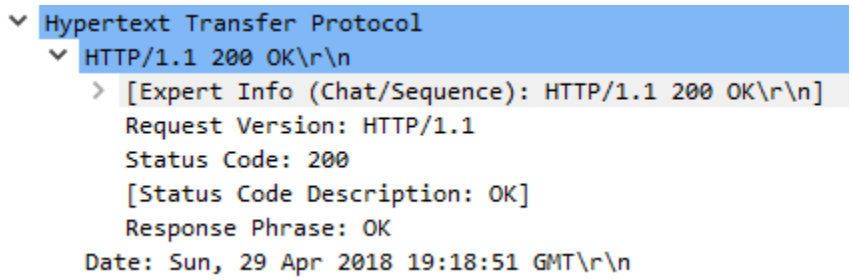


No.	Time	Source	Destination	Protocol	Length	Info
22	15:18:51.116993	192.168.0.16	128.119.245.12	HTTP	480	GET /wireshark-labs/HTTP-wireshark-file3.html HTTP/1.1
28	15:18:51.172475	128.119.245.12	192.168.0.16	HTTP	535	HTTP/1.1 200 OK (text/html)

The trace statements. GET is packet 22.

13. Packet number 28 in the trace contains the status code and phrase associated with the response to the HTTP GET request (shown in the screenshot above).

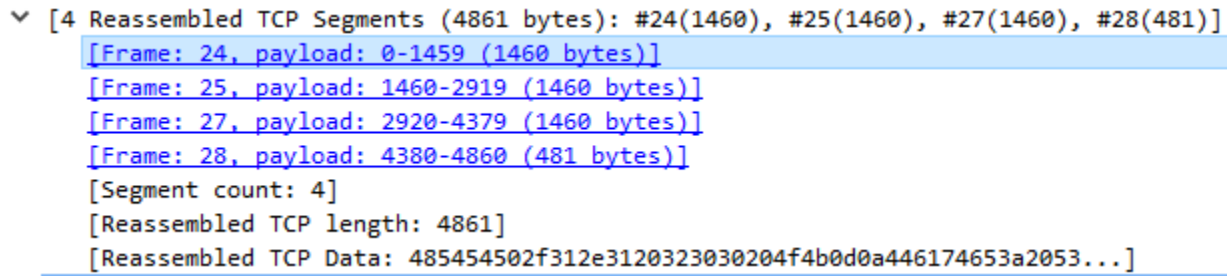
14. The status code in the response was 200 and the response phrase was “OK.”



```
Hypertext Transfer Protocol
  HTTP/1.1 200 OK\r\n
    [Expert Info (Chat/Sequence): HTTP/1.1 200 OK\r\n]
    Request Version: HTTP/1.1
    Status Code: 200
    [Status Code Description: OK]
    Response Phrase: OK
  Date: Sun, 29 Apr 2018 19:18:51 GMT\r\n
```

The server response with the status code and response phrase.

15. Four data-containing TCP segments were needed to carry the single HTTP response and the text of the Bill of Rights.



```
[4 Reassembled TCP Segments (4861 bytes): #24(1460), #25(1460), #27(1460), #28(481)]
  [Frame: 24, payload: 0-1459 (1460 bytes)]
  [Frame: 25, payload: 1460-2919 (1460 bytes)]
  [Frame: 27, payload: 2920-4379 (1460 bytes)]
  [Frame: 28, payload: 4380-4860 (481 bytes)]
  [Segment count: 4]
  [Reassembled TCP length: 4861]
  [Reassembled TCP Data: 485454502f312e3120323030204f4b0d0a446174653a2053...]
```

The above screenshot shows the four TCP segments that carry the single HTTP response and text of the Bill of Rights.

16. My browser sent four HTTP Get requests to:

/wireshark-labs/HTTP-wireshark-file4.html
/pearson.png
/~kurose/cover_5th_ed.jpg
/~kurose/cover_5th_ed.jpg

No.	Time	Source	Destination	Protocol	Length	Info
14	15:31:26.383823	192.168.0.16	128.119.245.12	HTTP	480	GET /wireshark-labs/HTTP-wireshark-file4.html HTTP/1.1
16	15:31:26.435932	128.119.245.12	192.168.0.16	HTTP	1127	HTTP/1.1 200 OK (text/html)
17	15:31:26.450074	192.168.0.16	128.119.245.12	HTTP	451	GET /pearson.png HTTP/1.1
21	15:31:26.498789	128.119.245.12	192.168.0.16	HTTP	745	HTTP/1.1 200 OK (PNG)
25	15:31:26.543863	192.168.0.16	128.119.240.90	HTTP	465	GET /~kurose/cover_5th_ed.jpg HTTP/1.1
28	15:31:26.593434	128.119.240.90	192.168.0.16	HTTP	510	HTTP/1.1 302 Found (text/html)
36	15:31:26.649225	192.168.0.16	128.119.240.90	HTTP	465	GET /~kurose/cover_5th_ed.jpg HTTP/1.1
124	15:31:26.883672	128.119.240.90	192.168.0.16	HTTP	526	HTTP/1.1 200 OK (JPEG JFIF image)

Packet trace list indicating the four GET requested (and four server responses).

17. These appear to have been downloaded serially. As seen in the screenshot above, the GET requests for the images were made in packets 17 and 25 respectively. The OK message from the server with the first image was returned in packet 21 which means that the browser requested and downloaded the first image before requesting the second image in packet 25 and downloading it in packet 124.

18. The server's response to the initial HTTP GET message from my browser was a status code of 401 and response phrase of "Unauthorized" as seen below.

```
Hypertext Transfer Protocol
  HTTP/1.1 401 Unauthorized\r\n
    [Expert Info (Chat/Sequence): HTTP/1.1 401 Unauthorized\r\n]
    Request Version: HTTP/1.1
    Status Code: 401
    [Status Code Description: Unauthorized]
    Response Phrase: Unauthorized
  Date: Sun, 29 Apr 2018 19:40:58 GMT\r\n
```

Status code and response phrase from initial HTTP GET request before password is entered.

19. In the second HTTP GET request, the new field is "Authorization" with the credentials that were entered for access.

```
  Authorization: Basic d2lyZXNoYXJrLXN0dWRlbnRzOm5ldHdvcms=\r\n
      Credentials: wireshark-students:network
```

The second GET request has an Authorization field with the given credientials.