Zachary Waters

Ellen Zegura

1/31/2019

CS 3251

Homework 1:

1.

1.1:

HTTP, ICMPv6, UDP

1.2:

The GET message was sent at 18:01:51:456053 and the RESPONSE message was received at 18:01:51:506744 so it took 0.50691 seconds.

1.3:

The URL of the server is: /wireshark-labs/INTRO-wireshark-file1.html

The URL of my computer is: /1.1

The IP address of the server is: 128.119.245.12

The IP address of my computer is: 10.0.0.171

1.4:

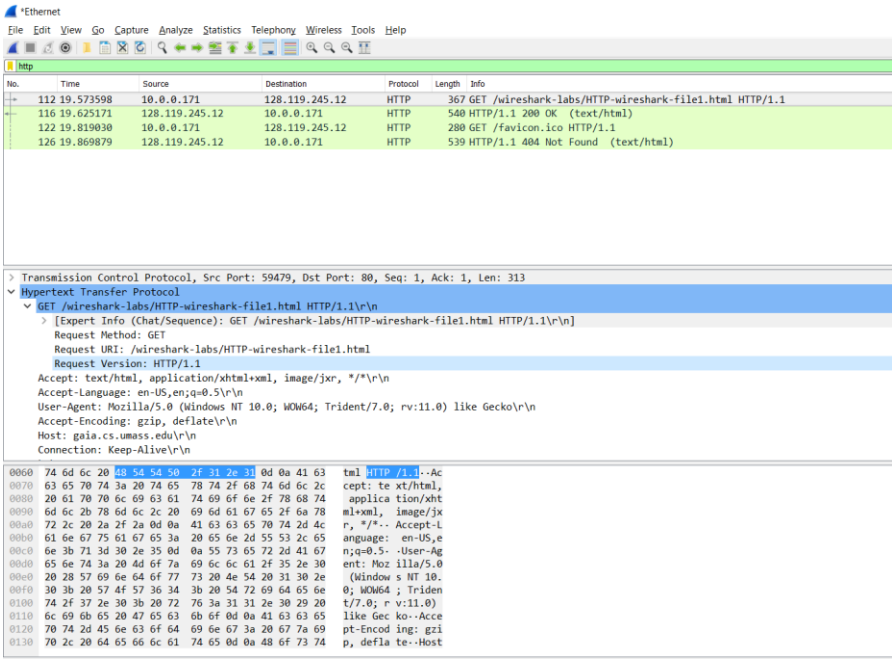
The two print pages for the GET and RESPONSE messages are attached separately in the submission.

2.

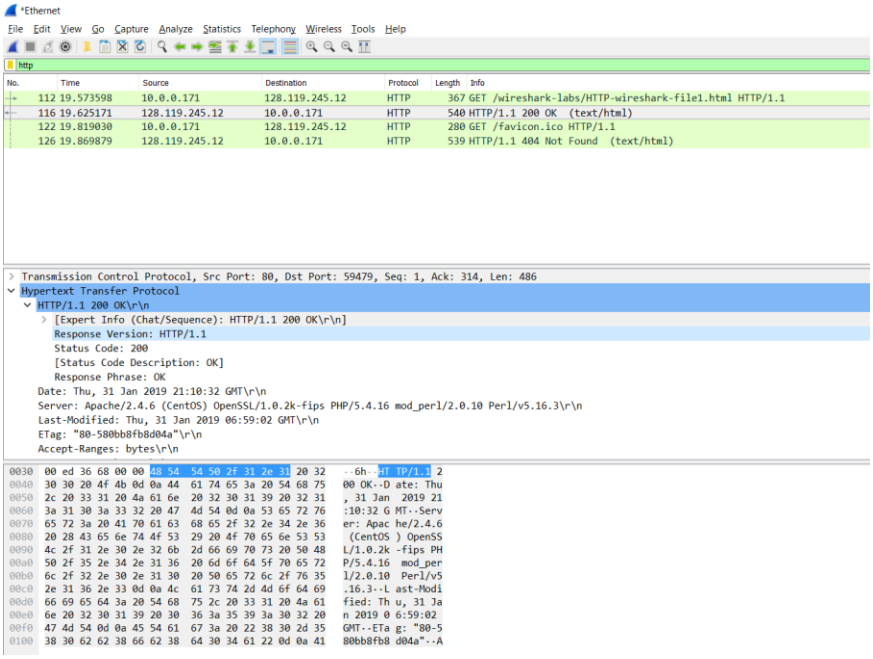
2.1:

My browser is running version 1.1 of HTTP, while the server is running version 1.1

Screen Shot of the GET message with the Highlighted Portion Below:



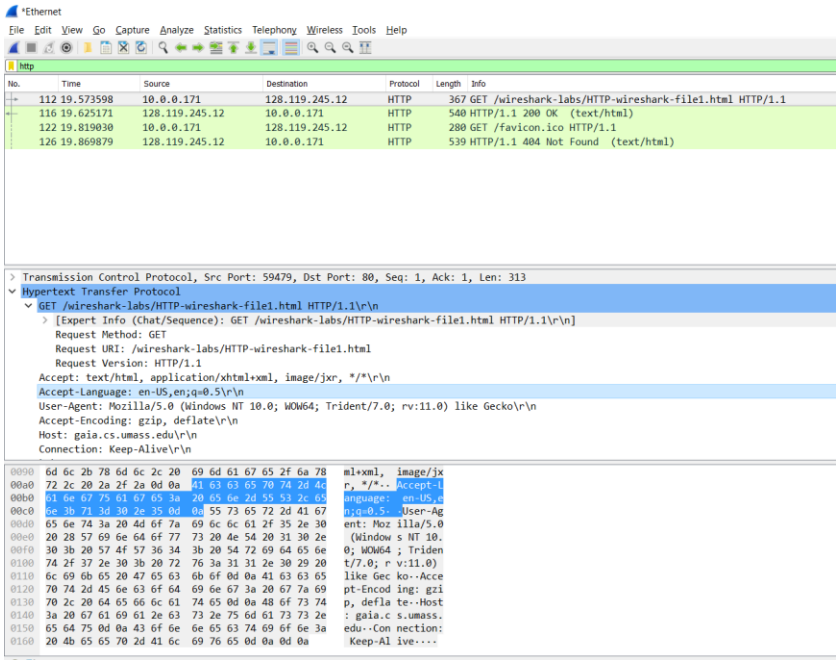
Screen Shot of the RESPONSE message with the Highlighted Portion Below:



2.2:

My browser indicates that it can accept English.

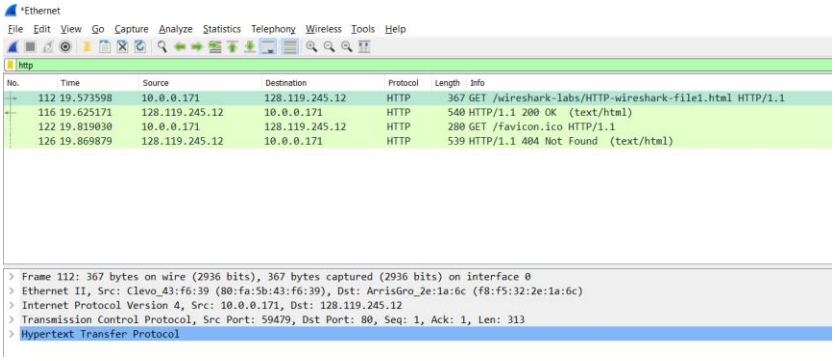
Screen Shot of the GET message with the Highlighted Portion Below:



2.3:

My computer's IP address is 10.0.0.171, while the IP address of the server is 128.119.245.12

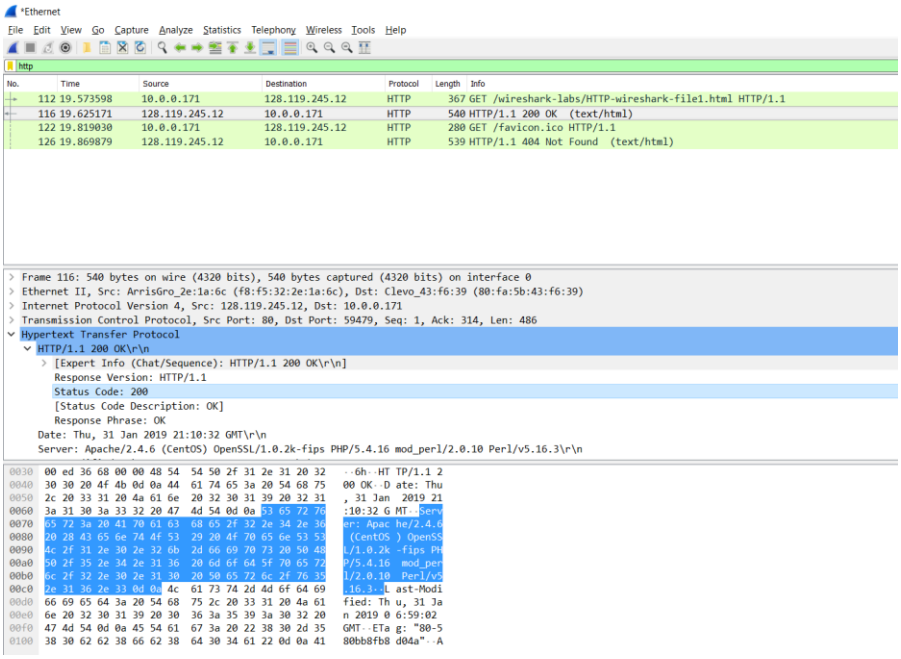
Screen Shot of the GET message with the Highlighted Portion Below:



2.4:

The status code returned from the server is 200.

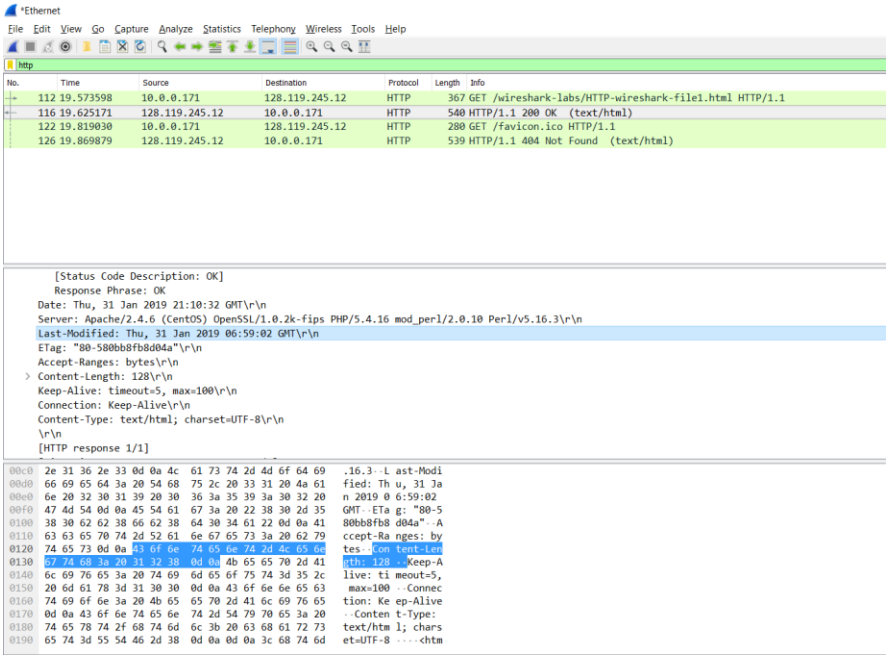
Screen Shot of the RESPONSE message with the Highlighted Portion Below:



2.5:

The retrieved HTML file was last modified Thu, 31 Jan 2019 06:59:02

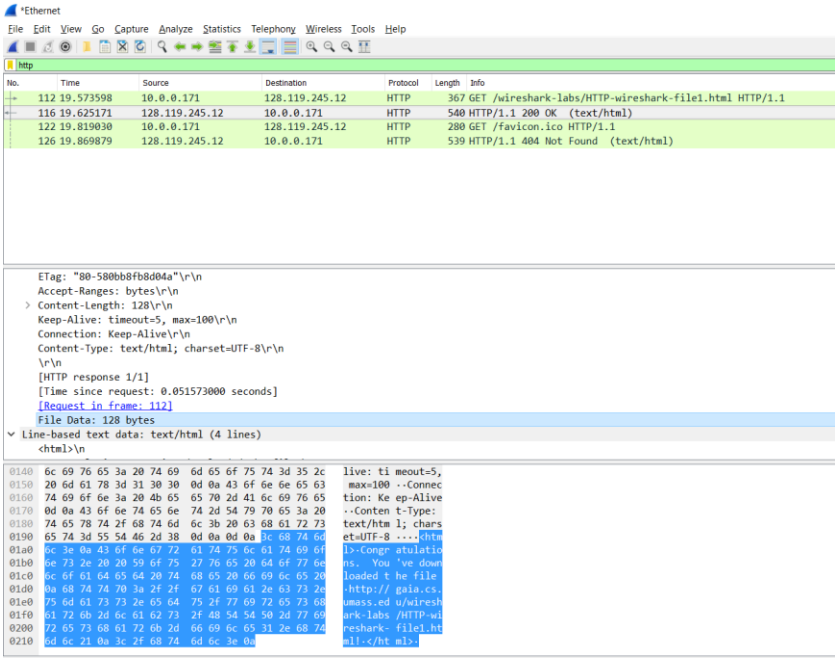
Screen Shot of the RESPONSE message with the Highlighted Portion Below:



2.6:

The retrieved file was 128 bytes.

Screen Shot of the RESPONSE message with the Highlighted Portion Below:



2.7:

I was unable to find anything that was not displayed in the packet-listing window.

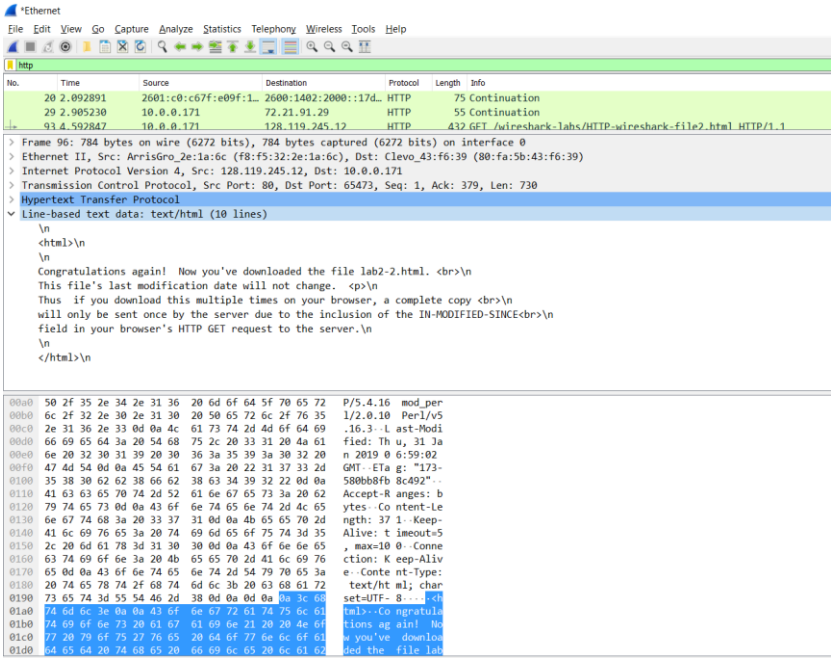
2.8:

I do not see a “IF-MODIFIED-SINCE” line

2.9:

The Server explicitly returned the contents of the file, you can see this by looking at the Line-based text data field.

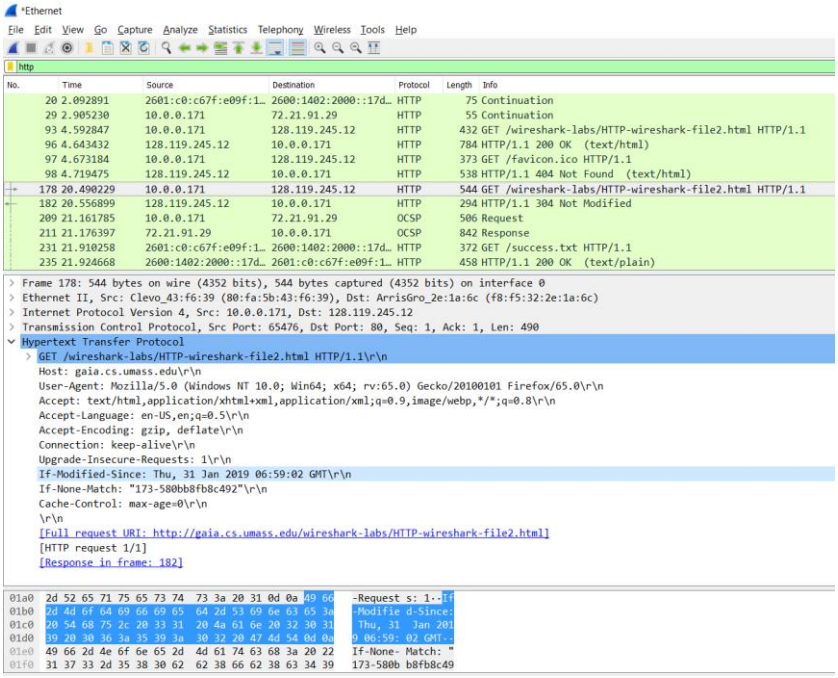
Screen Shot of the RESPONSE message with the Highlighted Portion Below:



2.10:

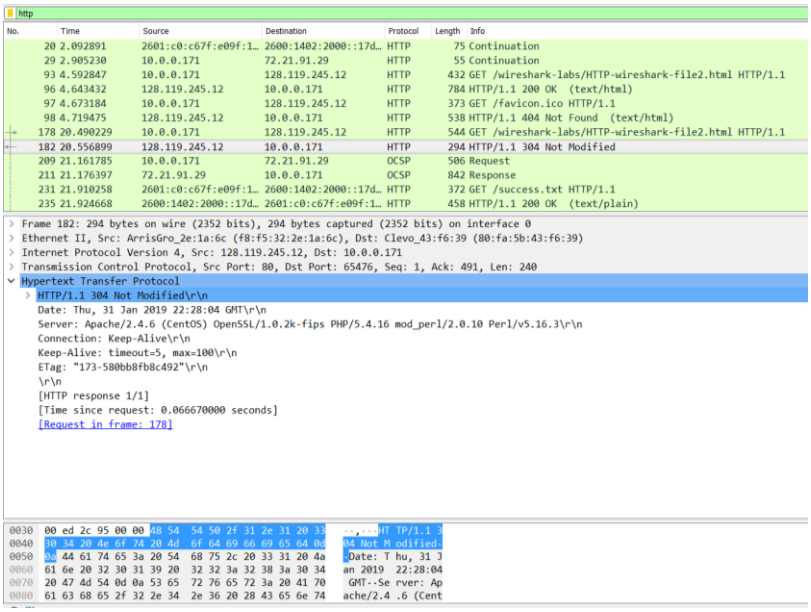
Yes, there is now a “IF-MODIFIED-SINCE” line in the GET request, what follows is “Thu, 31 Jan 2019 06:59:02 GMT/r/n”.

Screen Shot of the GET message with the Highlighted Portion Below:



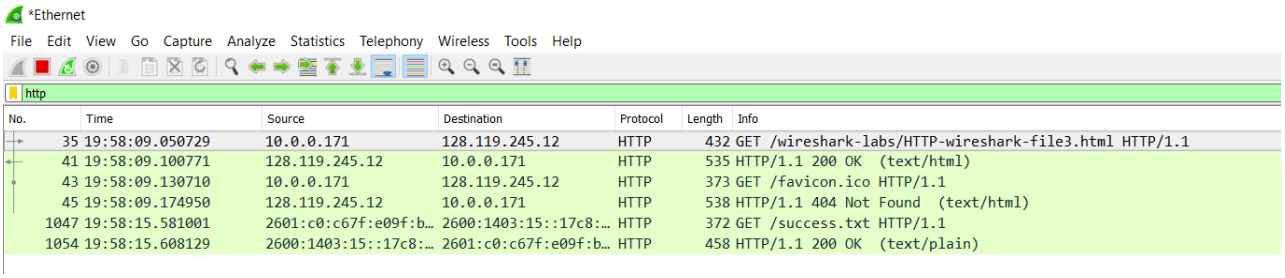
2.11:

The Server Sent a 304-status code with the phrase NOT MODIFIED. And did not return the content of the file.



2.12:

3 GET requests were sent. The Packet with number 35 was the GET message for the bill of rights.

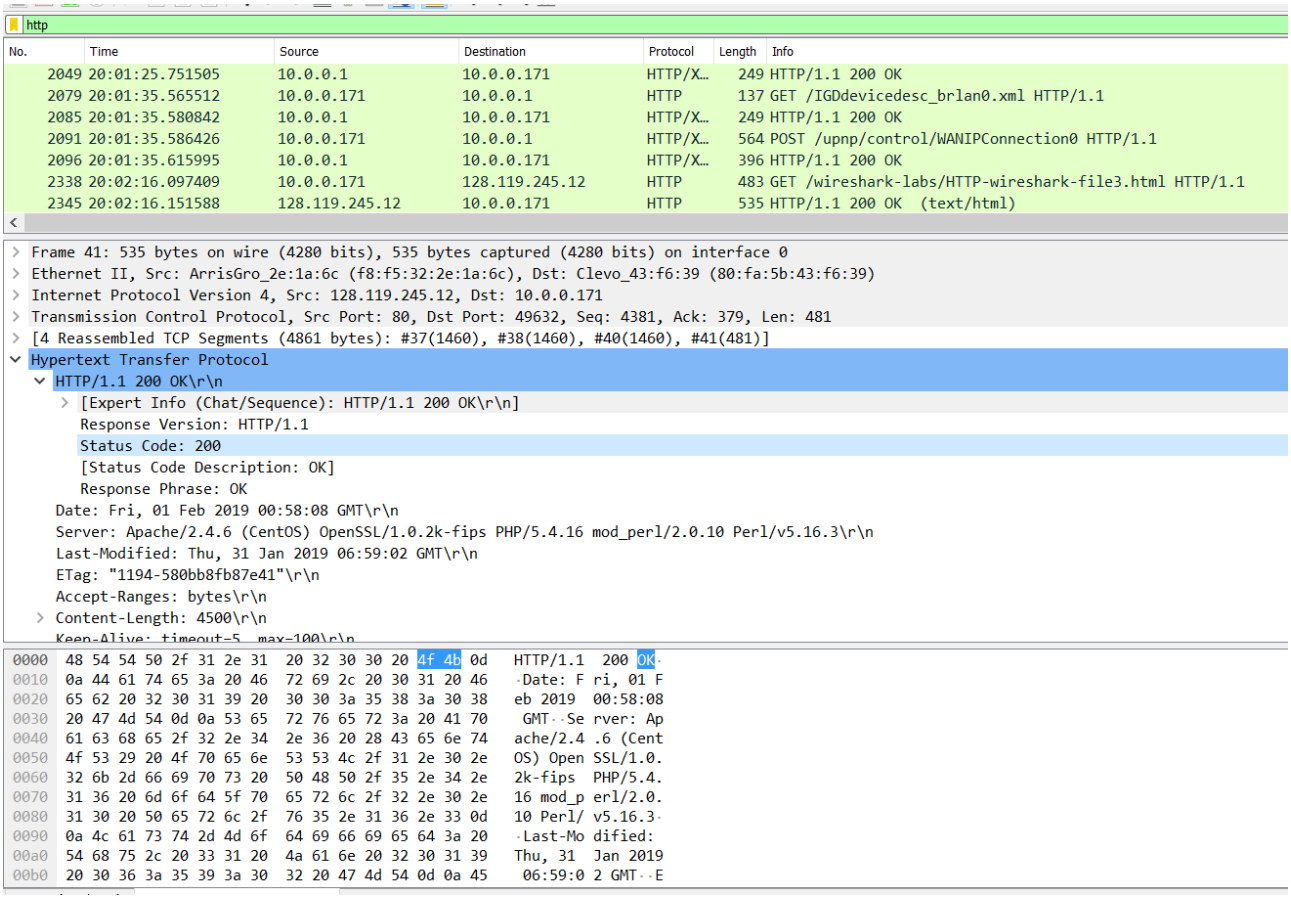


2.13:

The packet with number 41 contains the status code and phrase associated with the response to the GET request.

2.14:

The status code is 200, and the response phrase is “OK”.

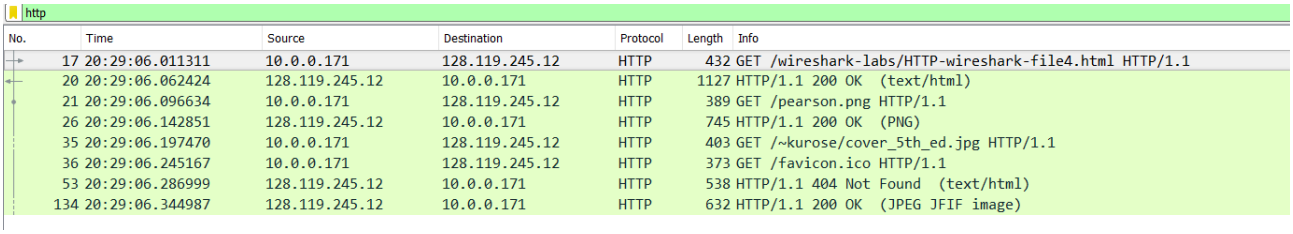


2.15:

The Response message was broken up into 4 TCP segments.

2.16:

4 GET requests were sent.



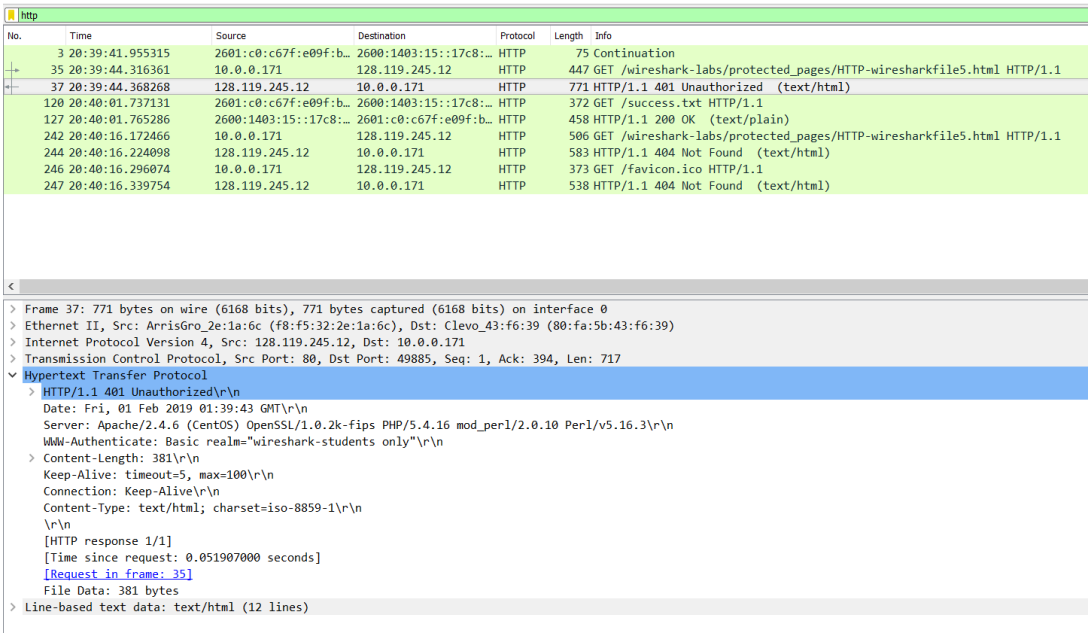
No.	Time	Source	Destination	Protocol	Length	Info
17	20:29:06.011311	10.0.0.171	128.119.245.12	HTTP	432	GET /wireshark-labs/HTTP-wireshark-file4.html HTTP/1.1
20	20:29:06.062424	128.119.245.12	10.0.0.171	HTTP	1127	HTTP/1.1 200 OK (text/html)
21	20:29:06.096634	10.0.0.171	128.119.245.12	HTTP	389	GET /pearson.png HTTP/1.1
26	20:29:06.142851	128.119.245.12	10.0.0.171	HTTP	745	HTTP/1.1 200 OK (PNG)
35	20:29:06.197470	10.0.0.171	128.119.245.12	HTTP	403	GET /~kurose/cover_5th_ed.jpg HTTP/1.1
36	20:29:06.245167	10.0.0.171	128.119.245.12	HTTP	373	GET /favicon.ico HTTP/1.1
53	20:29:06.286999	128.119.245.12	10.0.0.171	HTTP	538	HTTP/1.1 404 Not Found (text/html)
134	20:29:06.344987	128.119.245.12	10.0.0.171	HTTP	632	HTTP/1.1 200 OK (JPEG JFIF image)

2.17:

I believe they were downloaded serially, as the requests and responses for the two images were at different times.

2.18:

The server's RESPONSE message contains status code 401 and phrase "Unauthorized".

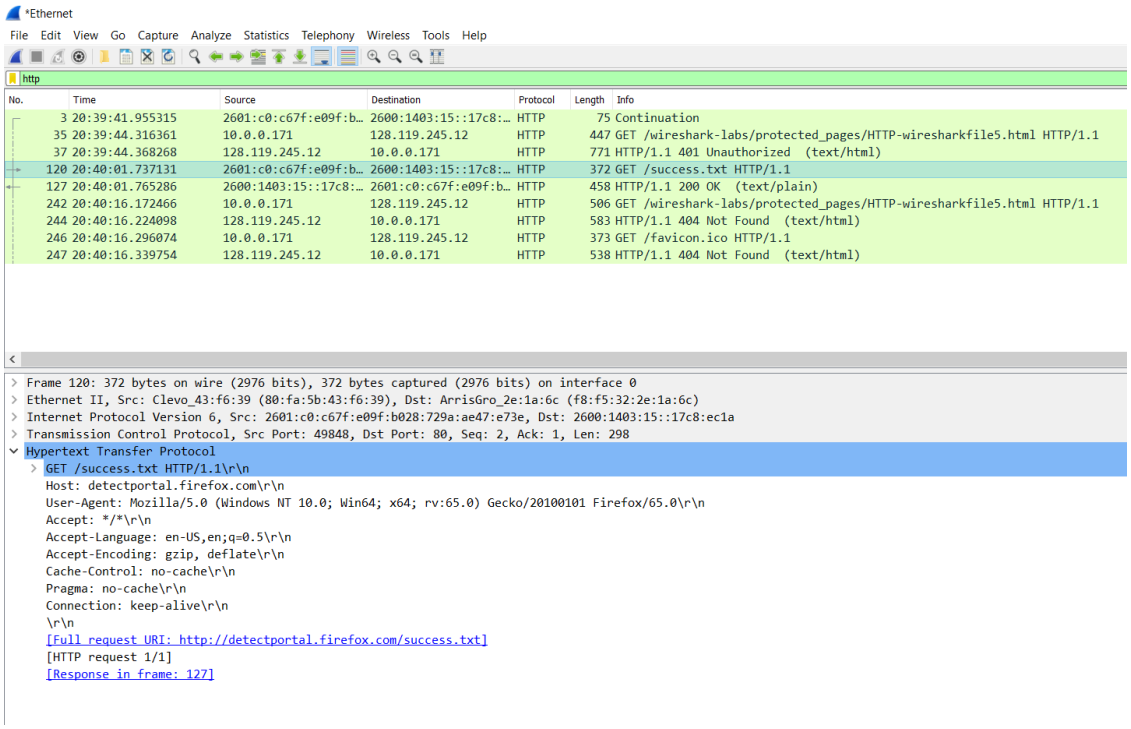


No.	Time	Source	Destination	Protocol	Length	Info
3	20:39:41.955315	2601:c0:c67f:e09f:b…	2600:1403:15::17c8:…	HTTP	75	Continuation
35	20:39:44.316361	10.0.0.171	128.119.245.12	HTTP	447	GET /wireshark-labs/protected_pages/HTTP-wiresharkfile5.html HTTP/1.1
37	20:39:44.368268	128.119.245.12	10.0.0.171	HTTP	771	HTTP/1.1 401 Unauthorized (text/html)
120	20:40:01.737131	2601:c0:c67f:e09f:b…	2600:1403:15::17c8:…	HTTP	372	GET /success.txt HTTP/1.1
127	20:40:01.765286	2600:1403:15::17c8:…	2601:c0:c67f:e09f:b…	HTTP	458	HTTP/1.1 200 OK (text/plain)
242	20:40:16.172466	10.0.0.171	128.119.245.12	HTTP	506	GET /wireshark-labs/protected_pages/HTTP-wiresharkfile5.html HTTP/1.1
244	20:40:16.224098	128.119.245.12	10.0.0.171	HTTP	583	HTTP/1.1 404 Not Found (text/html)
246	20:40:16.296074	10.0.0.171	128.119.245.12	HTTP	373	GET /favicon.ico HTTP/1.1
247	20:40:16.339754	128.119.245.12	10.0.0.171	HTTP	538	HTTP/1.1 404 Not Found (text/html)

```
Frame 37: 771 bytes on wire (6168 bits), 771 bytes captured (6168 bits) on interface 0
Ethernet II, Src: ArrisGro_2e:1a:6c (f8:f5:32:2e:1a:6c), Dst: Clevo_43:f6:39 (80:fa:5b:43:f6:39)
Internet Protocol Version 4, Src: 128.119.245.12, Dst: 10.0.0.171
Transmission Control Protocol, Src Port: 80, Dst Port: 49885, Seq: 1, Ack: 394, Len: 717
Hypertext Transfer Protocol
  HTTP/1.1 401 Unauthorized\r\n
  Date: Fri, 01 Feb 2019 01:39:43 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
  WWW-Authenticate: Basic realm="wireshark-students only"\r\n
  Content-Length: 381\r\n
  Keep-Alive: timeout=5, max=100\r\n
  Connection: Keep-Alive\r\n
  Content-Type: text/html; charset=iso-8859-1\r\n
  \r\n
  [HTTP response 1/1]
  [Time since request: 0.051907000 seconds]
  [Request in frame: 35]
  File Data: 381 bytes
Line-based text data: text/html (12 lines)
```

2.19:

The second GET request has the added phrase "/success.txt" and has the added fields "Cache-Control: no-cache/r/n" and Pragma "no-cache/r/n".



3.

3.1:

The end to end delay is the time it takes for “the whole message” to arrive, in this case it’s the time it takes for the tenth and final car to cross the finish line. We can solve this problem by separating it out into two subproblems. The first subproblem is to figure out how long it would take a car going 100km/hour to travel 150 km. The solution to this is trivial, its 1.5 hours or 90 minutes. The second subproblem is that we need to calculate the time it would take for the final car to get through all three tollbooths. Each tollbooth can only serve a single car and it takes 12 seconds for the car to be served. So, 3 tollbooths multiplied by 12 seconds multiplied by 10 cars gives us the time of 360 seconds (6 minutes) for the final car to complete its final tollbooth. Which means the end to end time for this problem is 6 minutes + 90 minutes = 96 minutes.

3.2:

With 8 cars the only thing that’s changed is the total tollbooth time which we can still solve using 3 tollbooths multiplied by 12 seconds a car multiplied by 8 cars. Which is 288 seconds (4.8 minutes), so our final solution is 4.8 minutes + 90 minutes = 94.8 minutes.

4.

To get a bit from Host A to B it must first be put into a packet, to be put in a packet the bit must wait for all the other bits in the packet to be generated. A packet is 48 bytes, or 384 bits, and host A has a 64-kbps stream. We have 64000 bits a second, or 1 bit per 0.000015625 seconds. So, to fill in all 384 bits of a packet it takes 0.006 seconds. The transmission rate of 2Mbps means we can send 2 million bits per second, so to send our 1 packet of 384 bits it would take 0.000192 seconds. Finally, the propagation delay is 10msec or 0.01 seconds. Our total time is 0.01 + 0.000192 + 0.006 = 0.016192 seconds.

5.

5.1:

To construct an equation for generating the end to end delay of a packet. We first calculate the time it takes for a packet of length L to be transmitted onto all the links in the system. This can be solved by dividing the length of the packet by the transmission rate of each switch and then summing them all together. Next, we need to calculate the delay caused by the packet propagating through each of the links by summing the length of the link divided by the propagation speed of the same link. finally, we need to account for is the total delay caused by the processing through summing the Dproc's for each of the packet switches. Now our final solution is simply the summation of all the above.

End to End Delay = L/R1+L/R2+L/R3 + D1/S1 + D2/S2 + D3/S3 + Dproc1 + Dproc2

5.2:

The packet is 1,500 bytes or 12000 bits. The transmission rate of all 3 links is 2 Mbps or 2 million bits per second. Which is a bit per 0.0000005 seconds. So, to transmit a packet though a link is 0.006 seconds. It goes through 3 links so it will take a total of 0.018 for the transmission time of all links. The propagation delay of all switches is 3msec or 0.003 seconds and because there are 2 switches it will take 0.006 seconds. Now we simply calculate the time of the length of each link divided by its propagation speed. So 5000km/2.5$x10^8 m/s$ + 4000km/2.5$x10^8 m/s$+1000km/2.5$x10^8 m/s$ = 0.02 seconds + 0.016 seconds + 0.004 seconds = 0.04 seconds. Now we simply add it all up. 0.04 + 0.006 +0.018 = 0.064 seconds

6.

6.1:

Assuming a parallel, non-persistent, connection, we would split it into 10 connections each sharing the 150/bits per second bandwidth connection, meaning they have access to 15 bits per second. Now assuming you are trying to download a packet of 100,000 bits long it would take, the number of bits divided by the download speed of a connection plus the transmission time of the getting that. We simply repeat these 3 times for the number of control packets each of size 200 bits that we also need to receive.

$(\frac{200}{150})3 + (\frac{200}{15})3 + (\frac{1000000}{150}) + (\frac{1000000}{15})$ = 7377 seconds

6.2:

With a persistent connection:

$(\frac{200}{150})3+(\frac{1000000}{150}) + 10((\frac{200}{150}) + (\frac{1000000}{15}))$ =7351 seconds

6.3:

The persistent connection is slightly better.