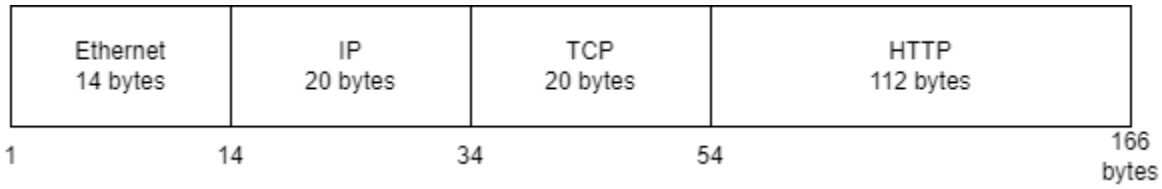
Lab Assignment-1

112001046
Vaibhav B Nagrale

Part 1

1. Hand in your packet drawing.
Ans.



Protocol layer structure of the HTTP GET packet

2. Estimate the download protocol overhead on packet 7 in the given trace.
Ans.
Frame size = 1484 bytes
Ethernet = 14 bytes
IP = 20 bytes
TCP = 20 bytes
Total(Ethernet, IP, TCP) = 14 + 20 + 20 = 54 bytes
Overhead (on packet 7) = (54/1484)*100 = 3.639

3. Estimate the download protocol overhead for the entire HTTP response, as defined above.
Ans.
Frame size = 66 + 60 + 1484 + 1484 + 1282 + 1484 + 1484 + 1290 + 1484 + 1484 + 1290 + 1484 + 1281 + 60 = 15717 bytes
Total (Ethernet, IP, TCP) = (32+20+14) + (20+20+20) + (20+20+14) + (20+20+14) + (20+20+14) + (20+20+14) + (20+20+14) + (20+20+14) + (20+20+14) + (20+20+14) + (20+20+14) + (20+20+14) + (20+20+14) + (20+20+20) = 780 bytes
Overhead = (780/15717)*100 = 4.963

4. Which Ethernet header field is the demultiplexing key indicating that the next higher layer is IP? What value is used in this field to indicate IP?
Ans.
The demultiplexing key for Ethernet is the Type field. It holds 0x800 when the higher layer is IP.

5. Which IP header field is the demultiplexing key indicating that the next higher layer is TCP? What value is used in this field to indicate TCP?
Ans.
The demultiplexing key for IP is the Protocol field. It has value 6 when the higher layer is TCP.

6. Why doesn't TCP's header contain a demultiplexing key? How can TCP know to deliver the data to HTTP on the receiving side?

Ans.

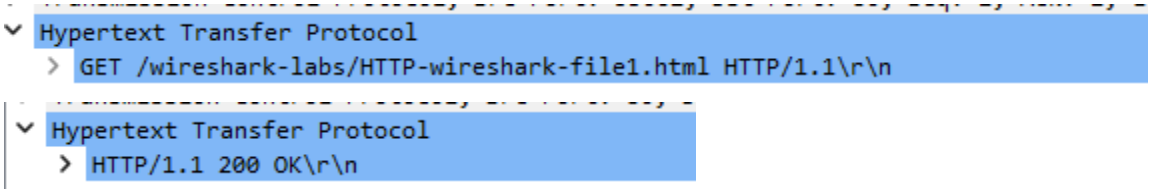
It doesn’t have demultiplexing key because TCP relies on destination port number in the header to identify the receiving application. This port number serves as a means for TCP to accurately route the data on the receiving end.

Part 2

1. Is your browser running HTTP version 1.0, 1.1, or 2? What version of HTTP is the server running?

Ans.

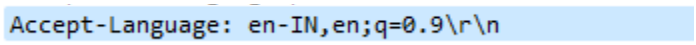
Both of them are version 1.1



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

Ans.

en-IN, en

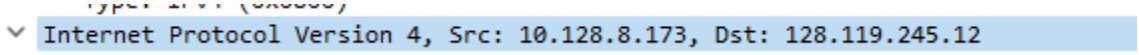


3. What is the IP address of your computer? What is the IP address of the gaia.cs.umass.edu server?

Ans.

Computer IP: 10.128.8.173

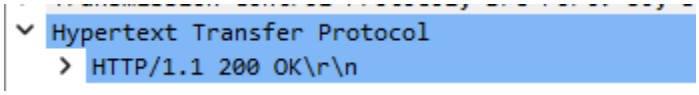
Gaia.cs.umass.edu IP: 128.199.245.12



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

Ans.

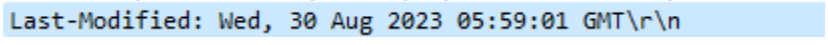
Status code: 200 ok



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

Ans.

Wed, 30 August 2023



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

Ans.

128

```
Content-Length: 128\r\n
    [Content length: 128]
```

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.

No, I don't see any in the HTTP Message below

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.

There is no IF-MODIFIED-SINCE in first GET

```
1233 4.996423    10.128.8.173    128.119.245.12    HTTP    526 GET /wireshark-labs/HTTP-wireshark-file2.html HTTP/1.1

Hypertext Transfer Protocol
  GET /wireshark-labs/HTTP-wireshark-file2.html HTTP/1.1\r\n
  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
  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;
  Accept-Encoding: gzip, deflate\r\n
  Accept-Language: en-IN,en;q=0.9\r\n
  \r\n
  [Full request URI: http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file2.html]
  [HTTP request 1/3]
  [Response in frame: 1267]
  [Next request in frame: 1278]
```

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

Ans.

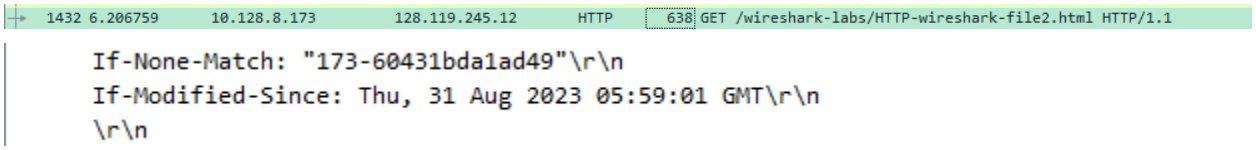
```
1267 5.289242    128.119.245.12    10.128.8.173    HTTP    784 HTTP/1.1 200 OK  (text/html)

Line-based text data: text/html (10 lines)
    \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
```

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.

2nd GET has IF-MODIFIED-SINCE

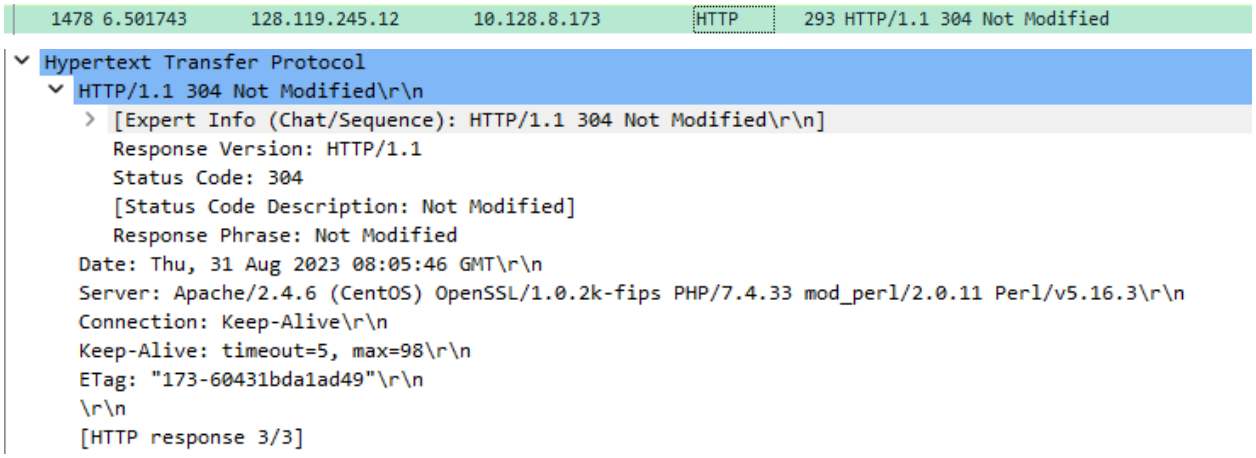


11. What is the HTTP status code and phrase returned from the server in response to this second HTTP GET? Did the server explicitly return the contents of the file? Explain.

Ans.

Status Code: 304

The file has not been modified. So the text of the file is not returned in HTTP message.



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?

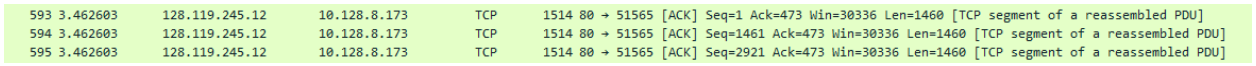
Ans.

HTTP GET request message: 1

In the 537th packet message Bill or RIghts was present.

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

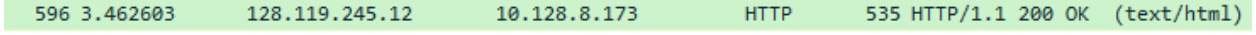
Ans.



Packet 593

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

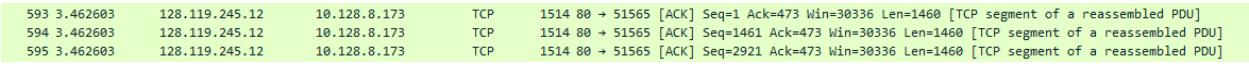
Ans.



200 ok

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

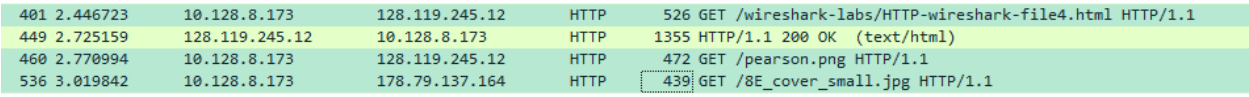
Ans.



Three packets were required to carry single HTTP response, they were 593, 594 and 595.

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

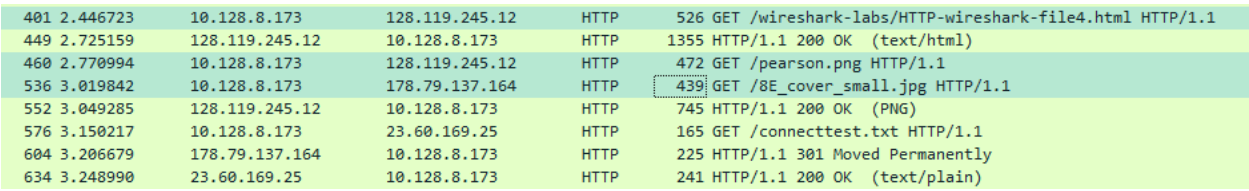
Ans.



3 HTTP GET packets: packet 401 (to get the base file), packet 460 (to get pearson.png) and packet 536 (to get 8E_cover_small.jpg)
Packet 401 was sent to 128.119.245.12, packet 460 to 128.119.245.12, and packet 536 to 178.79.137.164.

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.



The downloads occurred in parallel. The two GET messages for the images are in packets 460 and 536. The 200 OK reply containing the images show up as packets 552 and 634. Thus the request for the second image file (packet 552) was made BEFORE packet 634, the first image file was received.

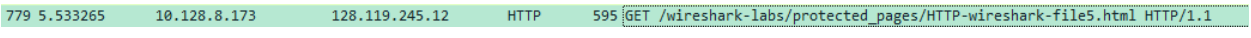
18. What is the server's response (status code and phrase) in response to the initial HTTP GET message from your browser?

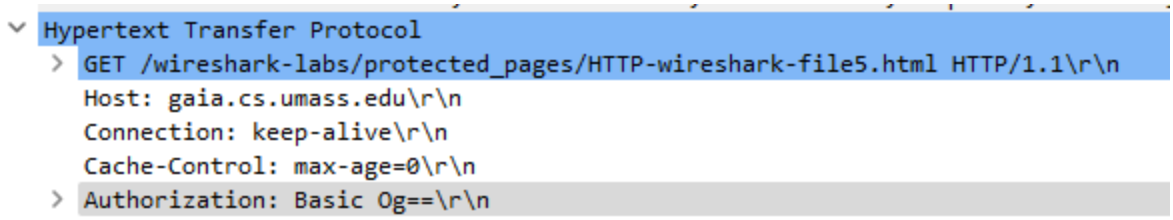
Ans.

Packet 381 in the trace has first GET and packet 431 has the reply. The server response in packet 431 is '401 Unathorized'.

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 HTTP GET includes the Authorization: Basic Og==\r\n