National University of Sciences and Technology (NUST)
School of Electrical Engineering and Computer Science

Department of Electrical Engineering

Faculty Member:___ Umar Farooq ___ **Dated: ____30/03/2022___**

Semester:____6th ____ **Section: _____D____**

EE-357 Computer and Communication Networks

Experiment - 7

Introduction to Wireshark – HTTP (Hypertext Transfer Protocol)

		PLO5/ CLO3		PLO5/ CLO3	PLO5/ CLO3	PLO5/ CLO3
Name	**Reg. No**	**Viva / Quiz / Lab Performance** **5 Marks**	**Analysis of data in Lab Report** **5 Marks**	**Modern Tool Usage** **5 Marks**	**Ethics and Safety** **5 Marks**	**Individual and Team Work** **5 Marks**
Myesha Khalil	**305093**					
Noor Ansar	**284825**					

National University of Sciences and Technology (NUST)
School of Electrical Engineering and Computer Science

EXPERIMENT NO 7

Introduction to Wireshark
HTTP (Hypertext Transfer Protocol)

Objective of this lab:

In this lab, we'll explore several aspects of the HTTP protocol: the basic GET/response interaction, and HTTP message formats.

Instructions:

- Read carefully before starting the lab.
- These exercises are to be done individually.
- You are supposed to provide the answers to the in-line questions in this document and upload the completed document to your course's LMS site.
- **For all questions, you must not only answer the question, but also supply all necessary information regarding how you arrived at the answer (e.g., use screenshots/ accompanying text, etc.) Use red font color to distinguish your replies from the rest of the text.**
- Avoid plagiarism by copying from the Internet or from your peers. You may refer to source/ text but you must paraphrase the original work.
- You can visit following links for detailed demo video: (Recommended)

  https://www.youtube.com/watch?v=jL4uJfCzBA4&ab_channel=NurulHuda

  https://www.youtube.com/watch?v=OkCF1dCd5c0&ab_channel=Alfietto92

  https://www.youtube.com/watch?v=iYM2zFP3Zn0&ab_channel=TraversyMedia

Background:

The world's web browsers, servers and related web applications all talk to each other through HTTP, the Hypertext Transfer Protocol. Before proceeding to the experiments, it is recommended that you read introductions to some general terms used in this lab, to avoid any confusion.

1. What is a web page?

A Web page (also called a document) consists of objects. An object is a simple file -- such as a HTML file, a JPEG image, a GIF image, a Java applet, an audio clip, etc. -- that is addressable by a single URL. Most Web pages consist of a base HTML file and several referenced objects. For example, if a Web page contains HTML text and five JPEG images, then the Web page has six objects: the base HTML file plus the five images. The base HTML file references the other objects in the page with the objects' URLs. Each URL has two components: the host name of the server that houses the object and the object's path name. For example, the URL www.someSchool.edu/someDepartment/picture.gif has www.someSchool.edu for a host name and /someDepartment/picture.gif for a path name.

2. What is a web browser?

A browser is a user agent for the Web; it displays to the user the requested Web page and provides numerous navigational and configuration features. Web browsers also implement the client side of HTTP. Thus, in the context of the Web, we will interchangeably use the words "browser" and "client". Popular Web browsers include Google Chrome, Netscape Communicator, Apple Safari and Microsoft Explorer.

3. What is a web server?

A Web server hosts Web objects, each addressable by a URL. Web servers also implement the server side of HTTP. Popular Web servers include Apache, Microsoft Internet Information Server, and the Netscape Enterprise Server.

4. Introduction to HTTP:

The Hypertext Transfer Protocol (HTTP), the Web's application-layer protocol, is at the heart of the Web. HTTP is implemented in two programs: a client program and server program. The client program and server programs, executing on different end systems, talk to each other by exchanging HTTP messages. HTTP defines the structure of these messages and how the client and server exchange the messages. HTTP defines how Web clients (i.e., browsers) request Web pages from servers (i.e., Web servers) and how servers transfer Web pages to clients. When a user requests a Web page (e.g., clicks on a hyperlink), the browser sends HTTP request messages for the objects in the page to the server. The server receives the requests and responds with HTTP response messages that contain the objects.

Steps for performing this lab:

For all the experiments we will use Wireshark packet analyzer.

Exercise 01: The Basic HTTP GET/response interaction

Aim of this exercise: We will now learn about what packets are exchanged during a HTTP conversation---we will learn about the HTTP GET message that is sent from the HTTP client to the HTTP server and the HTTP message that is sent as response to this message.

Follow the steps below to complete this exercise and to provide answers to the questions below

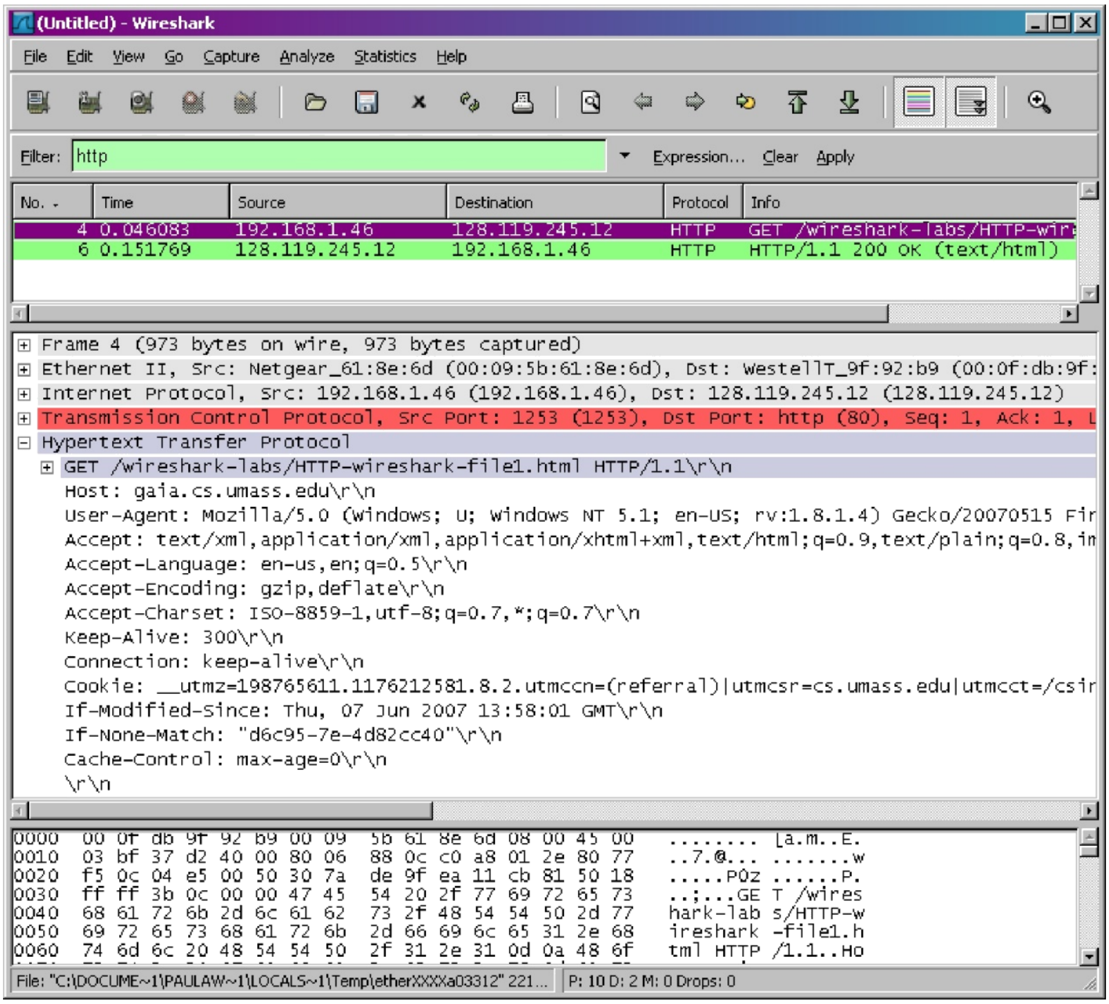
- Start up your web browser.
- Start up the Wireshark packet sniffer (but don't yet begin packet capture). Enter "http" (just the letters, not the quotation marks) in the display-filter-specification window, so that only captured HTTP messages will be displayed later in the packet-listing window. (We're only interested in the HTTP protocol here, and don't want to see the clutter of all captured packets).
- Begin Wireshark packet capture.
- Enter the following to your browser.http://gaia.cs.umass.edu/wireshark-labs/HTTP-

wireshark-file1.html our browser should display the very simple, one-line HTML file.

- Stop Wireshark packet capture.

The example in Figure 1 shows in the packet-listing window that two HTTP messages were captured: the GET message (from your browser to the gaia.cs.umass.edu web server) and the response message from the server to your browser. The packet-contents window shows details of the selected message (in this case the HTTP GET message, which is highlighted in the packet- listing window). Recall that since the HTTP message was carried inside a TCP segment, which was carried inside an IP datagram, which was carried within an Ethernet frame, Wireshark displays the Frame, Ethernet, IP, and TCP packet information as well.

Figure 1: Wireshark display after http://gaia.cs.umass.edu/wireshark-labs/ HTTP-



Filter: http

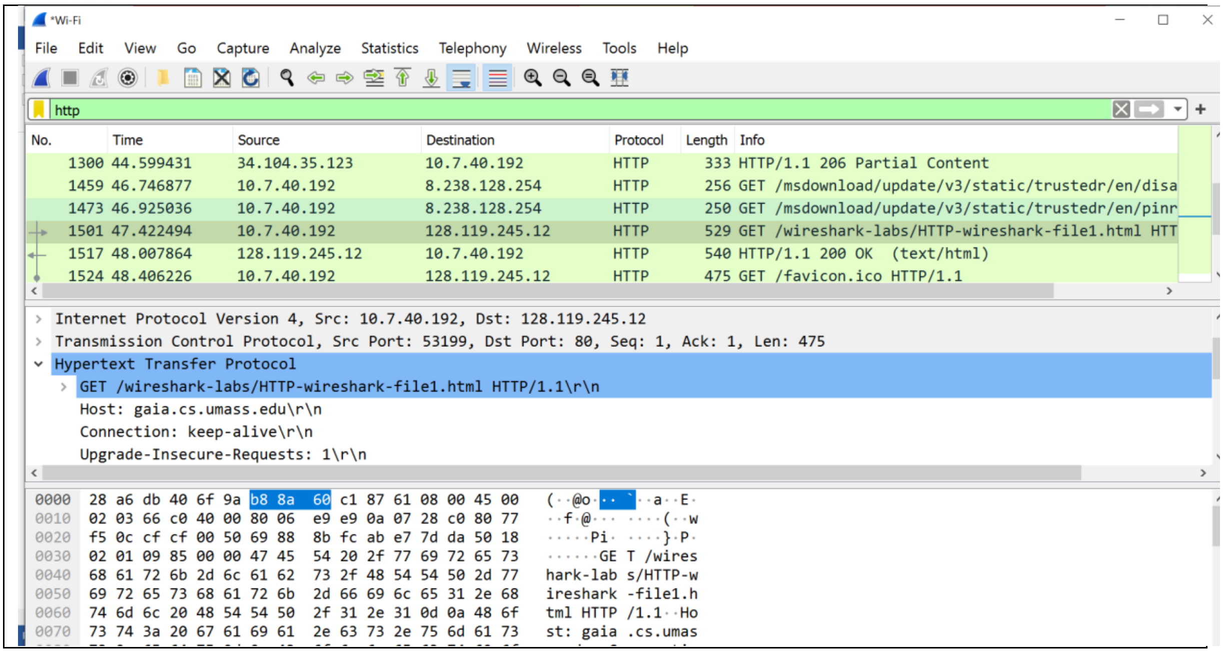
No.	Time	Source	Destination	Protocol	Info
4	0.046083	192.168.1.46	128.119.245.12	HTTP	GET /wireshark-labs/HTTP-wir
6	0.151769	128.119.245.12	192.168.1.46	HTTP	HTTP/1.1 200 OK (text/html)

```
⊞ Frame 4 (973 bytes on wire, 973 bytes captured)
⊞ Ethernet II, Src: Netgear_61:8e:6d (00:09:5b:61:8e:6d), Dst: WestellT_9f:92:b9 (00:0f:db:9f:
⊞ Internet Protocol, Src: 192.168.1.46 (192.168.1.46), Dst: 128.119.245.12 (128.119.245.12)
⊞ Transmission Control Protocol, Src Port: 1253 (1253), Dst Port: http (80), Seq: 1, Ack: 1, L
⊟ Hypertext Transfer Protocol
  ⊞ GET /wireshark-labs/HTTP-wireshark-file1.html HTTP/1.1\r\n
     Host: gaia.cs.umass.edu\r\n
     User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.8.1.4) Gecko/20070515 Fir
     Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,im
     Accept-Language: en-us,en;q=0.5\r\n
     Accept-Encoding: gzip,deflate\r\n
     Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7\r\n
     Keep-Alive: 300\r\n
     Connection: keep-alive\r\n
     Cookie: __utmz=198765611.1176212581.8.2.utmccn=(referral)|utmcsr=cs.umass.edu|utmcct=/csir
     If-Modified-Since: Thu, 07 Jun 2007 13:58:01 GMT\r\n
     If-None-Match: "d6c95-7e-4d82cc40"\r\n
     Cache-Control: max-age=0\r\n
     \r\n
```

```
0000  00 0f db 9f 92 b9 00 09  5b 61 8e 6d 08 00 45 00   ........ [a.m..E.
0010  03 bf 37 d2 40 00 80 06  88 0c c0 a8 01 2e 80 77   ..7.@... .......w
0020  f5 0c 04 e5 00 50 30 7a  de 9f ea 11 cb 81 50 18   .....P0z ......P.
0030  ff ff 3b 0c 00 00 47 45  54 20 2f 77 69 72 65 73   ..;...GE T /wires
0040  68 61 72 6b 2d 6c 61 62  73 2f 48 54 54 50 2d 77   hark-lab s/HTTP-w
0050  69 72 65 73 68 61 72 6b  2d 66 69 6c 65 31 2e 68   ireshark -file1.h
0060  74 6d 6c 20 48 54 54 50  2f 31 2e 31 0d 0a 48 6f   tml HTTP /1.1..Ho
```

File: "C:\DOCUME~1\PAULAW~1\LOCALS~1\Temp\etherXXXXa03312" 221... | P: 10 D: 2 M: 0 Drops: 0

wireshark-file1.html has been retrieved by your browser



*Wi-Fi
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

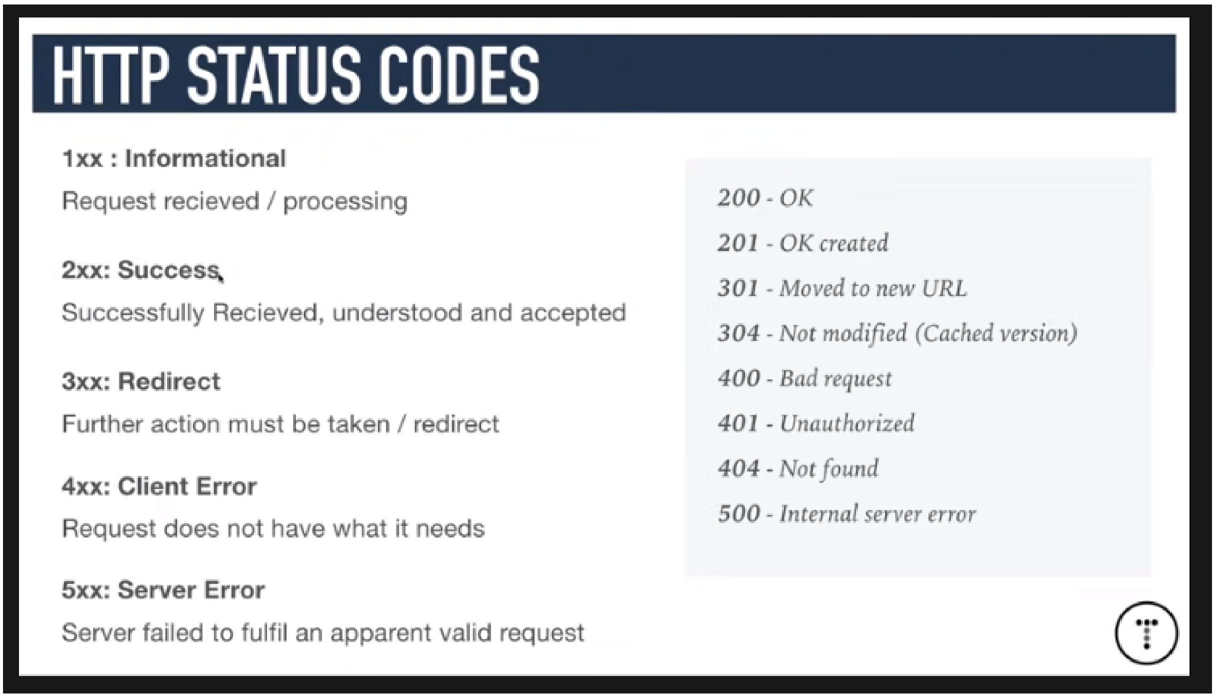
http

```
No.   Time      Source          Destination     Protocol Length Info
 1300 44.599431 34.104.35.123   10.7.40.192     HTTP      333 HTTP/1.1 206 Partial Content
 1459 46.746877 10.7.40.192     8.238.128.254   HTTP      256 GET /msdownload/update/v3/static/trustedr/en/disa
 1473 46.925036 10.7.40.192     8.238.128.254   HTTP      250 GET /msdownload/update/v3/static/trustedr/en/pinr
 1501 47.422494 10.7.40.192     128.119.245.12  HTTP      529 GET /wireshark-labs/HTTP-wireshark-file1.html HTT
 1517 48.007864 128.119.245.12  10.7.40.192     HTTP      540 HTTP/1.1 200 OK  (text/html)
 1524 48.406226 10.7.40.192     128.119.245.12  HTTP      475 GET /favicon.ico HTTP/1.1
```

```
> Internet Protocol Version 4, Src: 10.7.40.192, Dst: 128.119.245.12
> Transmission Control Protocol, Src Port: 53199, Dst Port: 80, Seq: 1, Ack: 1, Len: 475
v Hypertext Transfer Protocol
  > GET /wireshark-labs/HTTP-wireshark-file1.html HTTP/1.1\r\n
    Host: gaia.cs.umass.edu\r\n
    Connection: keep-alive\r\n
    Upgrade-Insecure-Requests: 1\r\n
```

```
0000  28 a6 db 40 6f 9a b8 8a  60 c1 87 61 08 00 45 00   (··@o··`··a··E·
0010  02 03 66 c0 40 00 80 06  e9 e9 0a 07 28 c0 80 77   ··f·@··· ····(··w
0020  f5 0c cf cf 00 50 69 88  8b fc ab e7 7d da 50 18   ·····Pi· ····}·P·
0030  02 01 09 85 00 00 47 45  54 20 2f 77 69 72 65 73   ······GE T /wires
0040  68 61 72 6b 2d 6c 61 62  73 2f 48 54 54 50 2d 77   hark-lab s/HTTP-w
0050  69 72 65 73 68 61 72 6b  2d 66 69 6c 65 31 2e 68   ireshark -file1.h
0060  74 6d 6c 20 48 54 54 50  2f 31 2e 31 0d 0a 48 6f   tml HTTP /1.1··Ho
0070  73 74 3a 20 67 61 69 61  2e 63 73 2e 75 6d 61 73   st: gaia .cs.umas
```

Some common HTTP Status codes are listed below:



HTTP STATUS CODES

1xx : Informational
Request recieved / processing

2xx: Success
Successfully Recieved, understood and accepted

3xx: Redirect
Further action must be taken / redirect

4xx: Client Error
Request does not have what it needs

5xx: Server Error
Server failed to fulfil an apparent valid request

200 - OK
201 - OK created
301 - Moved to new URL
304 - Not modified (Cached version)
400 - Bad request
401 - Unauthorized
404 - Not found
500 - Internal server error

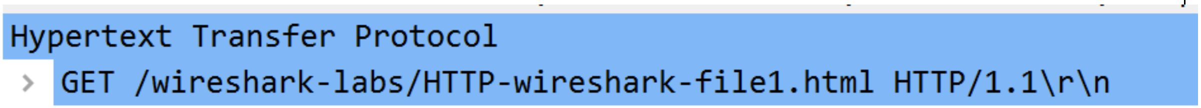
You can read about all the response codes for http response status codes in the following link:

https://developer.mozilla.org/en-US/docs/Web/HTTP/Status

By looking at the information in the HTTP GET and response messages that you have captured, answer the following questions:

1.1 Which version of HTTP is the browser running 1.0 or 1.1? Which HTTP version is the server running?

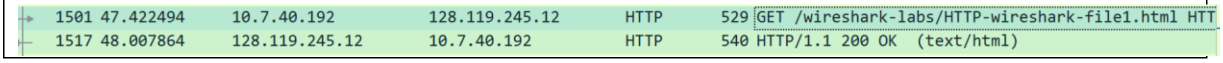
HTTP version 1.1



```
Hypertext Transfer Protocol
 > GET /wireshark-labs/HTTP-wireshark-file1.html HTTP/1.1\r\n
```

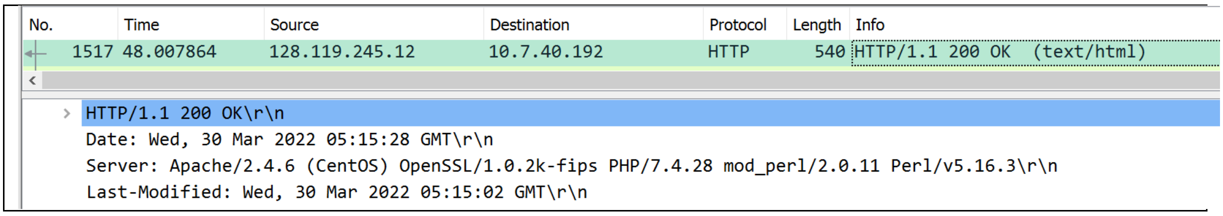
1.2 What is the status code returned from the server to your browser?

200 OK



```
1501 47.422494   10.7.40.192      128.119.245.12   HTTP   529 GET /wireshark-labs/HTTP-wireshark-file1.html HTT
1517 48.007864   128.119.245.12   10.7.40.192      HTTP   540 HTTP/1.1 200 OK  (text/html)
```

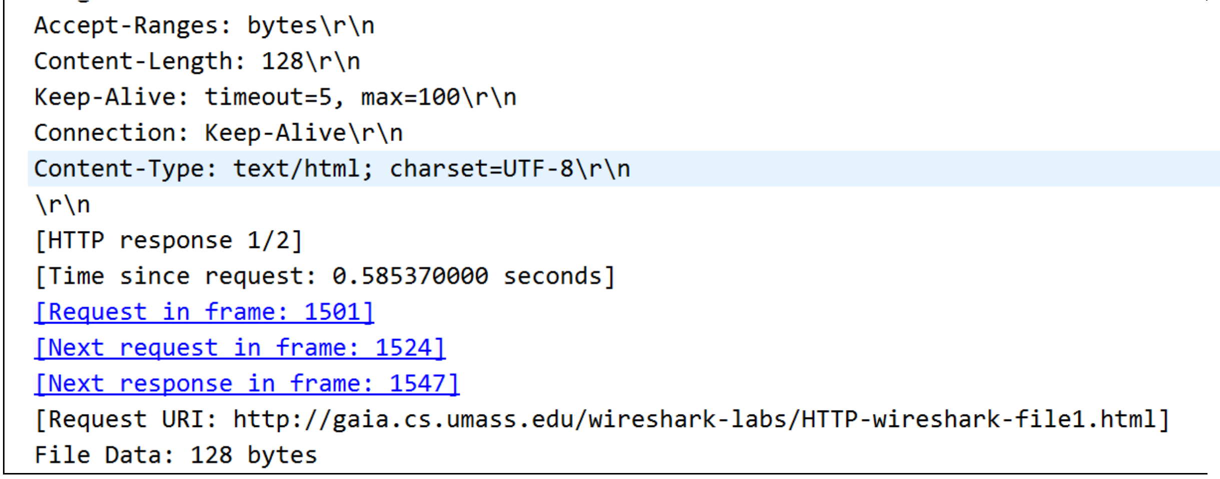
1.3 When the HTML file that you are retrieving was last modified at the server?



```
No.   Time        Source           Destination    Protocol  Length  Info
 1517 48.007864   128.119.245.12   10.7.40.192    HTTP         540  HTTP/1.1 200 OK  (text/html)

  > HTTP/1.1 200 OK\r\n
    Date: Wed, 30 Mar 2022 05:15:28 GMT\r\n
    Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/7.4.28 mod_perl/2.0.11 Perl/v5.16.3\r\n
    Last-Modified: Wed, 30 Mar 2022 05:15:02 GMT\r\n
```

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

128 bytes



```
Accept-Ranges: bytes\r\n
Content-Length: 128\r\n
Keep-Alive: timeout=5, max=100\r\n
Connection: Keep-Alive\r\n
Content-Type: text/html; charset=UTF-8\r\n
\r\n
[HTTP response 1/2]
[Time since request: 0.585370000 seconds]
[Request in frame: 1501]
[Next request in frame: 1524]
[Next response in frame: 1547]
[Request URI: http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file1.html]
File Data: 128 bytes
```

Exercise 02: The HTTP CONDITIONAL GET/response interaction

Aim of this exercise: We will now learn about a variant of the HTTP GET request message that we've seen earlier. We will note how the HTTP CONDITIONAL GET request and the reply to such a request differs from a simple HTTP GET request. Before performing the steps below, make sure your browser's cache is empty. (To do this under Firefox, select Tools->Clear Recent History and check the Cache box, or for Internet Explorer, select Tools->Internet Options->Delete File; these actions will remove cached files from your browser's cache.)

The following indicate the steps for this experiment:

- Start up your web browser, and make sure your browser's cache is cleared, as discussed above.
- Start up the Wireshark packet sniffer
- Enter the following URL into your browser
  http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file2.html

  Your browser should display a very simple five-line HTML file.

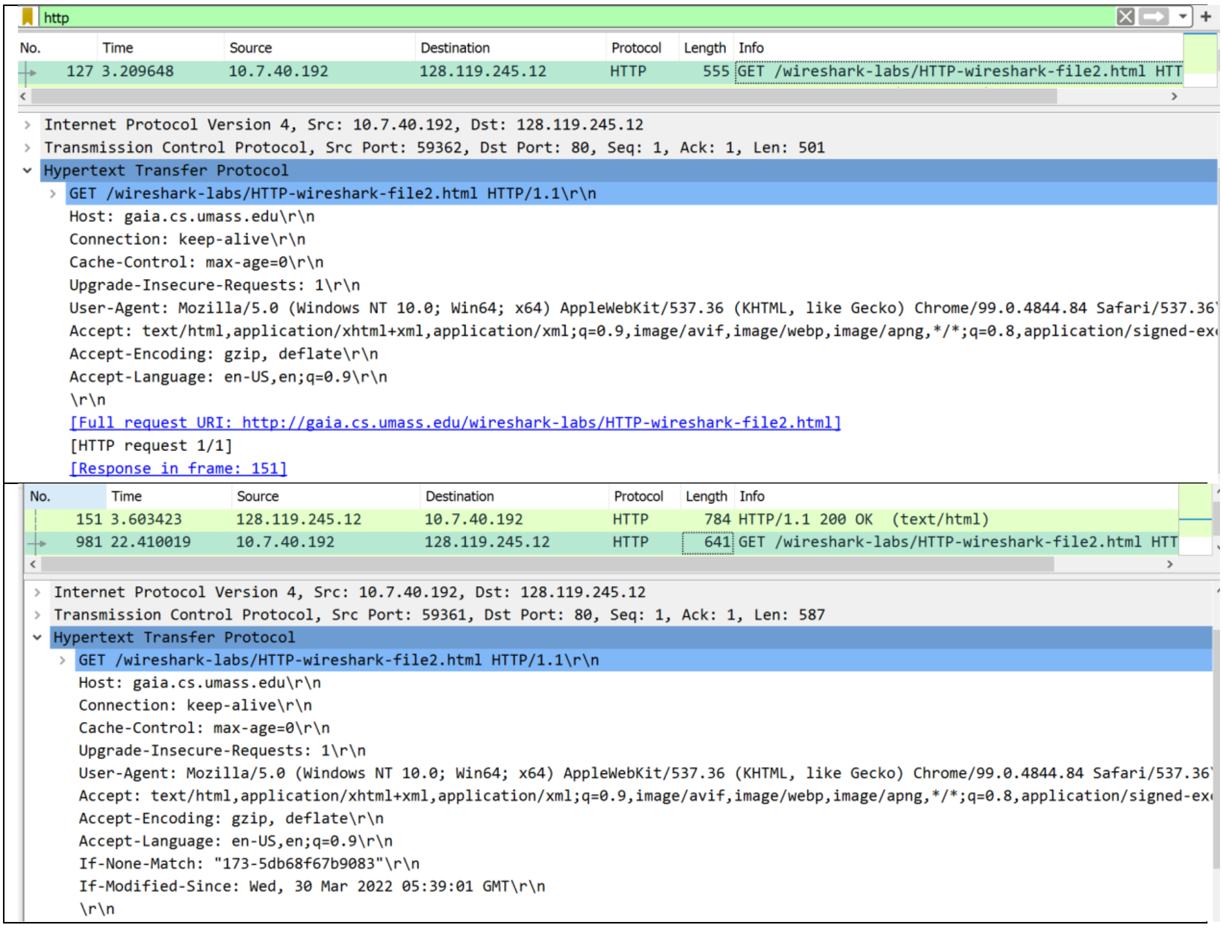
- Quickly enter the same URL into your browser again (or simply select the refresh button on your browser)

- Stop Wireshark packet capture, and enter “http” in the display-filter-specification window, so that only captured HTTP messages will be displayed later in the packet-listing window.

- Filter out all the non-HTTP packets and focus on the HTTP header information in the packet-header detail window.

- By looking at the information in the HTTP GET and response messages, answer the following questions:

2.1 Inspect the contents of the first and 2nd HTTP GET requests from the browser to the server. Do you see “IF-MODIFIED-SINCE” and “IF-NONE-MATCH” lines in these HTTP GET message? Why?



```
http

No.   Time      Source        Destination     Protocol Length Info
  127 3.209648  10.7.40.192   128.119.245.12  HTTP     555    GET /wireshark-labs/HTTP-wireshark-file2.html HTT

> Internet Protocol Version 4, Src: 10.7.40.192, Dst: 128.119.245.12
> Transmission Control Protocol, Src Port: 59362, Dst Port: 80, Seq: 1, Ack: 1, Len: 501
v 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
    Cache-Control: max-age=0\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/99.0.4844.84 Safari/537.36
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-ex
    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: 151]
```

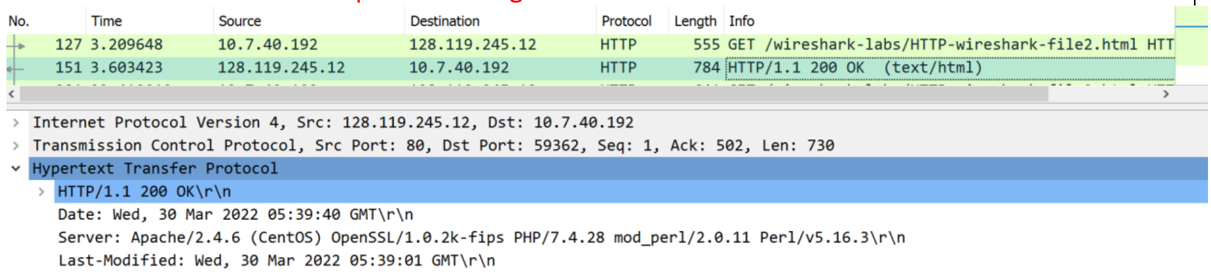
```
No.   Time       Source          Destination     Protocol Length Info
  151 3.603423   128.119.245.12  10.7.40.192     HTTP     784    HTTP/1.1 200 OK  (text/html)
  981 22.410019  10.7.40.192     128.119.245.12  HTTP     641    GET /wireshark-labs/HTTP-wireshark-file2.html HTT

> Internet Protocol Version 4, Src: 10.7.40.192, Dst: 128.119.245.12
> Transmission Control Protocol, Src Port: 59361, Dst Port: 80, Seq: 1, Ack: 1, Len: 587
v 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
    Cache-Control: max-age=0\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/99.0.4844.84 Safari/537.36
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-ex
    Accept-Encoding: gzip, deflate\r\n
    Accept-Language: en-US,en;q=0.9\r\n
    If-None-Match: "173-5db68f67b9083"\r\n
    If-Modified-Since: Wed, 30 Mar 2022 05:39:01 GMT\r\n
    \r\n
```

2.2 What is the difference in first and second response received? What is the last modified time in

the first response message?

Difference: for the first HTTP GET request, the message shown is 'last modified since' while the 2nd HTTP GET shows 'if modified since' because for the second request, wireshark packet capture searches the cache to see if the initial html page was modified or not.

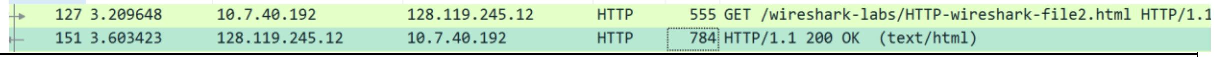
Last modified time in first response message



```
No.     Time        Source           Destination      Protocol  Length  Info
    127 3.209648    10.7.40.192      128.119.245.12   HTTP        555   GET /wireshark-labs/HTTP-wireshark-file2.html HTT
    151 3.603423    128.119.245.12   10.7.40.192      HTTP        784   HTTP/1.1 200 OK  (text/html)

> Internet Protocol Version 4, Src: 128.119.245.12, Dst: 10.7.40.192
> Transmission Control Protocol, Src Port: 80, Dst Port: 59362, Seq: 1, Ack: 502, Len: 730
v Hypertext Transfer Protocol
  > HTTP/1.1 200 OK\r\n
    Date: Wed, 30 Mar 2022 05:39:40 GMT\r\n
    Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/7.4.28 mod_perl/2.0.11 Perl/v5.16.3\r\n
    Last-Modified: Wed, 30 Mar 2022 05:39:01 GMT\r\n
```

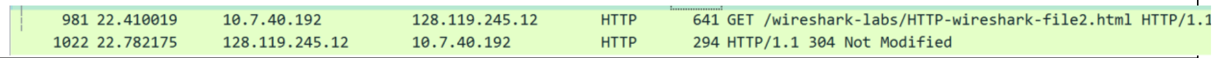
2.3 What is the HTTP status code and phrase returned from the server in response to the first and second HTTP GET? Did the server explicitly return the contents of the file? Explain.

First



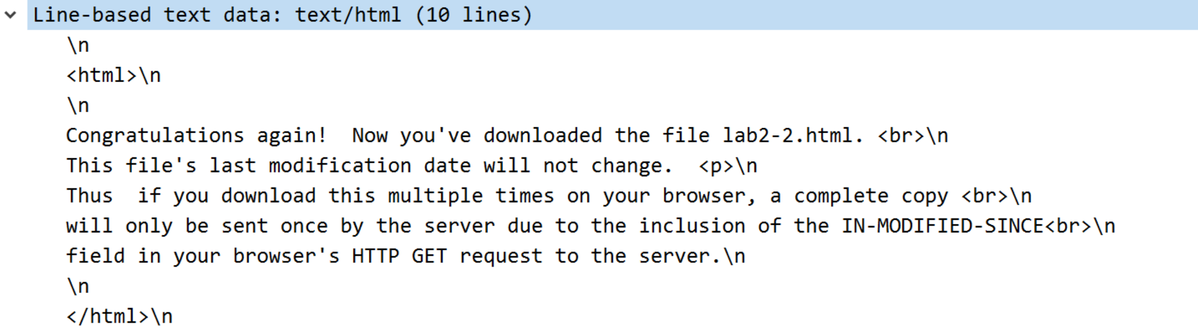
```
    127 3.209648    10.7.40.192      128.119.245.12   HTTP        555   GET /wireshark-labs/HTTP-wireshark-file2.html HTTP/1.1
    151 3.603423    128.119.245.12   10.7.40.192      HTTP        784   HTTP/1.1 200 OK  (text/html)
```

Second



```
    981 22.410019   10.7.40.192      128.119.245.12   HTTP        641   GET /wireshark-labs/HTTP-wireshark-file2.html HTTP/1.1
   1022 22.782175   128.119.245.12   10.7.40.192      HTTP        294   HTTP/1.1 304 Not Modified
```

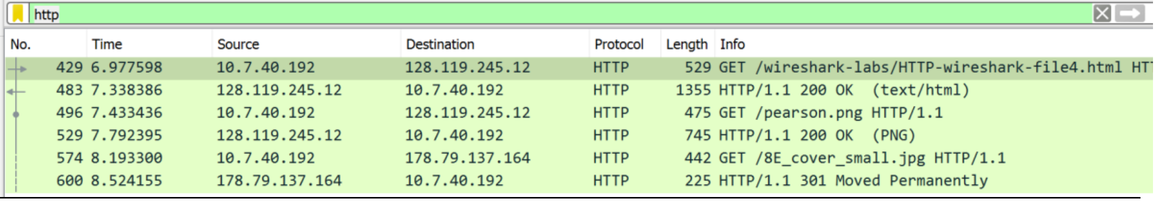
The server did explicitly return the contents of the file for the first HTTP GET



```
v 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
```

2.4 Empty your browser cache again and open the webpage http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file4.html and capture the GET and OK response messages. How many total objects does the server return?

3 objects: HTML file, png file, jpg file



```
http

No.     Time        Source           Destination      Protocol  Length  Info
    429 6.977598    10.7.40.192      128.119.245.12   HTTP        529   GET /wireshark-labs/HTTP-wireshark-file4.html HT
    483 7.338386    128.119.245.12   10.7.40.192      HTTP       1355   HTTP/1.1 200 OK  (text/html)
    496 7.433436    10.7.40.192      128.119.245.12   HTTP        475   GET /pearson.png HTTP/1.1
    529 7.792395    128.119.245.12   10.7.40.192      HTTP        745   HTTP/1.1 200 OK  (PNG)
    574 8.193300    10.7.40.192      178.79.137.164   HTTP        442   GET /8E_cover_small.jpg HTTP/1.1
    600 8.524155    178.79.137.164   10.7.40.192      HTTP        225   HTTP/1.1 301 Moved Permanently
```

2.5 Write a short summary for the interaction in 2.4?

Upon entering the URL into the browser, a short HTML file with two images reinforced in it opens up. The images themselves are not contained in the base HTML, instead the URLs for the images are contained in the downloaded HTML file. The browser then retrieves these from the indicated websites, downloading them serially, as indicated by the time stamps.