Department of Electrical Engineering, UET Lahore

EE432 Computer Networks Lab

Course Instructor: Dr. Naveed Nawaz	**Dated: 09/09/2024**
Session: Fall 2024	**Semester: 7th**

LAB 2 Exploring HTTP protocol and HTTP request and response messages

Name	Roll. No.	Report Marks (10)	Viva Marks (5)	Total Marks (15)
Ayesha Ahmad	2021-EE-052			

Signature: _____

Contents

List of figures

Exploring HTTP protocol and HTTP request/response messages

2.1. Objectives

At the end of this lab, students will have achieved the following goals:

- Explore several aspects of the HTTP protocol
- Observe the basic GET/response interaction, HTTP message formats

2.2. Instructions

1. Read manual carefully before starting lab.
2. All exercises are individual exercises.
3. You are supposed to provide the answers to the questions listed at the end of this manual in text. Paste screenshots/images in the textboxes where required. You will be required to submit your complete manual on Google classroom.
4. Avoid plagiarism by copying from the Internet or from your peers. You may refer to source/text but you must paraphrase the original work. Your submitted work should be written by yourself.
5. You must aim to complete the lab half an hour before the lab time ends.
6. At the end of the lab, a viva will be conducted to evaluate your understanding.

2.3. Background

Having introduced the Wireshark packet analyzer in the introductory lab, we're now ready to use Wireshark to investigate protocols in operation, like HTTP, which is a common language of the modern global Internet. 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, read introductions to some general terms used in this lab, to avoid any confusion.

2.3.1. Web page

A Web page (also called a document) consists of objects. An object is a simple file – such as an 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.

2.3.2. 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 and Microsoft Explorer.

2.3.3. 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.

2.4. Lab procedure

For all the experiments, we will use Wireshark packet analyzer that we used in the lab 1.

2.4.1. The basic HTTP GET/response interaction

2.4.1.1. Aim of this exercise

We will now learn about what packets are exchanged during an 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.

2.4.1.2. Procedure

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, as described in lab 1 (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).
- Wait a bit more than one minute (we'll see why shortly), and then begin Wireshark packet capture.
- Enter the following to your browser http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file1.html.Your browser should display the very simple, one-line HTML file.
- Stop Wireshark packet capture.

The example in Figure 2.1 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.

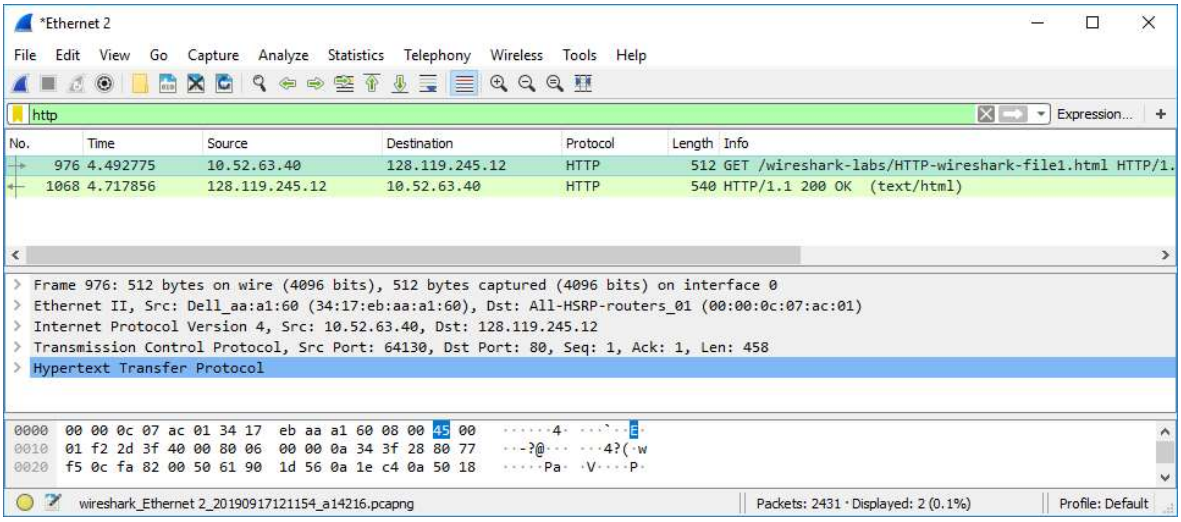


Figure 2.1: Wireshark display after http://gaia.cs.umass.edu/wireshark-labs/ HTTP-wireshark-file1.html has been retrieved by your browser

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

2.4.1.3. Questions

1. Which version of HTTP is the browser running 1.0 or 1.1? Which HTTP version is the server running? *Paste screenshots and accompanying text to answer this question.*

[3 marks]

The browser is running HTTP version 1.1

No.	Time	Source	Destination	Protocol	Length	Info
1068	10.205325	10.5.88.181	128.119.245.12	HTTP	540	GET /wireshark-labs/HTTP-wireshark-file1.html HTTP/1.1

The server is running HTTP version 1.1

1359	11.799923	128.119.245.12	10.5.88.181	HTTP	540	HTTP/1.1 200 OK (text/html)

2. What languages (if any) does the browser indicate that it can accept to the server? *Paste screenshot (containing referenced item) and accompanying text to answer this question.*

[2 marks]

```
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
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWe
Accept: text/html,application/xhtml+xml,application/xml;q=0.9
Accept-Encoding: gzip, deflate\r\n
Accept-Language: en-US,en;q=0.9\r\n
\r\n
```

The browser indicates that it accepts the US english with priority level 1 and english with a priority level of 0.9.

```
Accept-Language: en-US,en;q=0.9\r\n
```

3. What is the IP address of your computer and of the gaia.cs.umass.edu server? *Describe how you determined these IP addresses.*

[2 marks]

The source of the request message (GET) is our computer and the destination is the gaia.cs.umass.edu.server. Thus source of GET is the IP address of the computer and the destination of the GET message is the server.
IP address of computer: 10.5.88.181 IP address of gaia server: 128.119.245.12

No.	Time	Source	Destination	Protocol	Length	Info
1068	10.205325	10.5.88.181	128.119.245.12	HTTP	540	GET /wireshark-labs/HTTP-wireshark-file1.html HTTP/1.1
1359	11.799923	128.119.245.12	10.5.88.181	HTTP	540	HTTP/1.1 200 OK (text/html)

4. What is the status code returned from the server to your browser? *Paste screenshot (containing referenced item) and accompanying text to answer this question.*

[2 marks]

The status code returned from the server to the browser is **200** with description OK.

1359	11.799923	128.119.245.12	10.5.88.181	HTTP	540	HTTP/1.1 200 OK (text/html)

5. When was the HTML file that you are retrieving last modified at the server? *Describe how you determined this.*

[2 marks]

The Last-Modified at the server of the retrieved HTML file is Tuesday, 10 September 2024 05:59:02 GMT

```
Last-Modified: Tue, 10 Sep 2024 05:59:02 GMT\r\n
```

```
HTTP/1.1 200 OK\r\n
Date: Wed, 11 Sep 2024 03:58:22 GMT\r\n
Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PH
Last-Modified: Tue, 10 Sep 2024 05:59:02 GMT\r\n
ETag: "80-621bd91ffa6c2"\r\n
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
```

6. How many bytes of content are being returned to your browser? *Describe how you determined this*

[2 marks]

The Content-Length is given as 128, which implies that **128 bytes** are returned to the browser.

```
Content-Length: 128\r\n
```

```
HTTP/1.1 200 OK\r\n
Date: Wed, 11 Sep 2024 03:58:22 GMT\r\n
Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PH
Last-Modified: Tue, 10 Sep 2024 05:59:02 GMT\r\n
ETag: "80-621bd91ffa6c2"\r\n
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
```

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. *Paste screenshot (containing referenced item) and accompanying text to answer this question.*

[2 marks]

No, all the headers within the raw-data are displayed in the packet-listing window.
This was confirmed by clicking on the raw-data bytes to confirm the corresponding headers.

```
0040  30 30 20 4f 4b 0d 0a 44  61 74 65 3a 20 54 75 65   00 OK··D ate: Tue
0050  2c 20 31 37 20 53 65 70  20 32 30 32 34 20 31 36   , 17 Sep  2024 16
0060  3a 30 39 3a 34 36 20 47  4d 54 0d 0a 53 65 72 76   :09:46 G MT··Serv
0070  65 72 3a 20 41 70 61 63  68 65 2f 32 2e 34 2e 36   er: Apac he/2.4.6
0080  20 28 43 65 6e 74 4f 53  29 20 4f 70 65 6e 53 53    (CentOS ) OpenSS
0090  4c 2f 31 2e 30 2e 32 6b  2d 66 69 70 73 20 50 48   L/1.0.2k -fips PH
00a0  50 2f 37 2e 34 2e 33 33  20 6d 6f 64 5f 70 65 72   P/7.4.33  mod_per
00b0  6c 2f 32 2e 30 2e 31 31  20 50 65 72 6c 2f 76 35   l/2.0.11  Perl/v5
00c0  2e 31 36 2e 33 0d 0a 4c  61 73 74 2d 4d 6f 64 69   .16.3··L ast-Modi
00d0  66 69 65 64 3a 20 54 75  65 2c 20 31 37 20 53 65   fied: Tu e, 17 Se
00e0  70 20 32 30 32 34 20 30  35 3a 35 39 3a 30 31 20   p 2024 0 5:59:01
```

2.4.2. The HTTP CONDITIONAL GET/response interaction

2.4.2.1. 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 (which we talked about in exercise **Error! Reference source not found.**).

2.4.2.2. Procedure

The following are the steps for this exercise:

1. Start up your web browser, and make sure your browser's cache is cleared, as discussed above.
2. Start up the Wireshark packet sniffer.
3. 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.
4. Quickly enter the same URL into your browser again (or simply select the refresh button on your browser).
5. 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.
6. Filter out all the non-HTTP packets and focus on the HTTP header information in the packet-header detail window.
7. By looking at the information in the HTTP GET and response messages (the first two messages), answer the following questions.

2.4.2.3. Questions

1. Inspect the contents of the first HTTP GET request from the browser to the server. Do you see an "IF-MODIFIED-SINCE" line in the HTTP GET? *Paste screenshot (containing referenced item) and accompanying text to answer this question.*

[2 marks]

There is NO IF-MODIFIED-SINCE line in the 1st HTTP GET line.

```
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/53
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/
Accept-Encoding: gzip, deflate\r\n
Accept-Language: en-US,en;q=0.9\r\n
\r\n
```

2. Inspect the contents of the server response. Did the server explicitly return the contents of the file? How can you tell? *Paste screenshot (containing referenced item) and accompanying text to answer this question.*

[3 marks]

The server explicitly returned the contents of the file, as the Content-Length is 371 bytes with type text/html. The content is showed in the Line-based text data window.

```
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
```

```
HTTP/1.1 200 OK\r\n
Date: Tue, 17 Sep 2024 16:09:46 GMT\r\n
Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips P
Last-Modified: Tue, 17 Sep 2024 05:59:01 GMT\r\n
ETag: "173-6224a62da531c"\r\n
Accept-Ranges: bytes\r\n
Content-Length: 371\r\n
   [Content length: 371]
Keep-Alive: timeout=5, max=100\r\n
Connection: Keep-Alive\r\n
Content-Type: text/html; charset=UTF-8\r\n
\r\n
```

3. Does the response indicate the last time that the requested file was modified? *Paste screenshot (containing referenced item) and accompanying text to answer this question.*

[2 marks]

The response indicates that the file was Last-Modified at Tuesday, 17 September 2024 05:59:01 GMT

```
Last-Modified: Tue, 17 Sep 2024 05:59:01 GMT\r\n
```

```
HTTP/1.1 200 OK\r\n
Date: Tue, 17 Sep 2024 16:09:46 GMT\r\n
Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips P
Last-Modified: Tue, 17 Sep 2024 05:59:01 GMT\r\n
ETag: "173-6224a62da531c"\r\n
Accept-Ranges: bytes\r\n
Content-Length: 371\r\n
   [Content length: 371]
Keep-Alive: timeout=5, max=100\r\n
Connection: Keep-Alive\r\n
Content-Type: text/html; charset=UTF-8\r\n
\r\n
```

4. Now inspect the contents of the second HTTP GET request from the browser to the server. Do you see an "IF-MODIFIED-SINCE:" line in the HTTP GET? If so, what information is contained in the "IF-MODIFIED-SINCE:" header? *Paste screenshot (containing referenced item) and accompanying text to answer this question.*

[2 marks]

There is an IF-MODIFIED-SINCE line in the 2nd HTTP GET line. The If-Modified-Since line contains the same time as the Last-Modified time from the previous response of the server.

```
If-Modified-Since: Tue, 17 Sep 2024 05:59:01 GMT\r\n
```

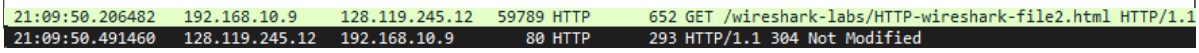
```
GET /wireshark-labs/HTTP-wireshark-file2.html HTTP/1.1\r\n
Host: gaia.cs.umass.edu\r\n
Connection: keep-alive\r\n
Cache-Control: max-age=0\r\n
Upgrade-Insecure-Requests: 1\r\n
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWe
Accept: text/html,application/xhtml+xml,application/xml;q=0.9
Accept-Encoding: gzip, deflate\r\n
Accept-Language: en-US,en;q=0.9\r\n
If-None-Match: "173-6224a62da531c"\r\n
If-Modified-Since: Tue, 17 Sep 2024 05:59:01 GMT\r\n
\r\n
```

5. 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. *Paste screenshot (containing referenced item) and accompanying text to answer this question.*

[3 marks]

The HTTP status code and phrase returned to the 2nd HTTP GET was "304 Not Modified". The server did not explicitly return the contents of the file as there is NO Content-Length defined in the header."

```
HTTP/1.1 304 Not Modified\r\n
Date: Tue, 17 Sep 2024 16:09:49 GMT\r\n
Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips
Connection: Keep-Alive\r\n
Keep-Alive: timeout=5, max=98\r\n
ETag: "173-6224a62da531c"\r\n
\r\n
```



Assessment Rubrics for Computer Networks Lab 2

Student Name: _____ Roll Number: _____

Method:

Lab report evaluation and instructor observation during lab sessions.

Outcomes Assessed:

a. Ability to condut experiments as well as to analyze and interpret data
b. Ability to adhere to safety and disciplinary rules
c. Ability to use the techniques, skills and modern engineering tools necessary for engineering practice

Performance	Exceeds expectation (5-4)	Meets expectation (3-2)	Does not meet expectation (1)	Marks
Realization of experiment (a)	Downloads and installs required software and sets up the system according to the experiment requirements	Needs guidance to set up the system according to the experiment requirements	Incapable of selecting relevant software to the experiment and unable to setup the system with required software tools	
Conducting experiment (a, c)	Carries out each procedural step in a satisfactory manner and studies outputs of the software application rigorously	Needs assistance or guidance to proceed through experiment steps, studies outputs with minor errors in interpretation	Unable to carry out procedural steps and make any useful observations of outputs	
Laboratory safety and disciplinary rules (b)	Observes lab safety rules; adheres to the lab disciplinary guidelines aptly	Observes safety rules and disciplinary guidelines with minor deviations	Disregards lab safety and disciplinary rules	
Data collection (c)	Completes data collection from the experiment setup by following procedural steps, ensures that the data is entered in the lab manual according to the specified instructions	Completes data collection with minor error and enters data in lab manual with slight deviation from guidelines	Fails at collecting data by giving proper inputs and observing output states of experiment setup, unable to fill the lab manual properly	
Data analysis (a, c)	Analyzes the data obtained from experiment thoroughly and accurately verifies it with theoretical understanding, accounts for any discrepancy in data from theory with sound explanation	Analyzes data with minor error and correlates it with theoretical values reasonably. Attempts to account for any discrepancy in data from theory	Unable to establish the relationship between practical and theoretical values and lacks the theoretical understanding to explain any discrepancy in data	