ID:z5195715 Name:Junyu Ren

Lab2

Exercise 3: Using Wireshark to understand basic HTTP request/response messages

Question 1: What is the status code and phrase returned from the server to the client browser?

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

Status code is 200 and phrase is OK.

Question 2: When was the HTML file that the browser is retrieving last modified at the server? Does the response also contain a DATE header? How are these two fields different?

```
    Response Version: HTTP/1.1
    Status Code: 200
    [Status Code Description: OK]
    Response Phrase: OK
Date: Tue, 23 Sep 2003 05:29:50 GMT\r\n
Server: Apache/2.0.40 (Red Hat Linux)\r\n
Last-Modified: Tue, 23 Sep 2003 05:29:00 GMT\r\n
```

Last modified at Tue, 23 Sep 2003 05:29:00 GMT.

Yes,the response contains a DATE header

Date is the time when server generates and sends response.

The Last-Modified response HTTP header contains the date and time at which the origin server believes the resource was last modified.

In this case,the server set Last-modified date the same as the date when the response created.

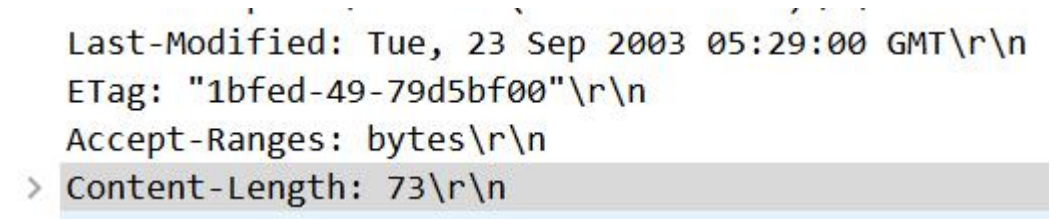
Question 3: Is the connection established between the browser and the server persistent or non-persistent? How can you infer this?

```
> Content-Length: 73\r\n
Keep-Alive: timeout=10, max=100\r\n
Connection: Keep-Alive\r\n
Content-Type: text/html; charset=ISO-8859-1\r\n
```

```
Accept-Language: en-us, en;q=0.50\r\n
Accept-Encoding: gzip, deflate, compress;q=0.9\r\n
Accept-Charset: ISO-8859-1, utf-8;q=0.66, *;q=0.66\r\n
Keep-Alive: 300\r\n
Connection: keep-alive\r\n
```

From the pictures above, we can infer that the connection is persistent both in request and response.

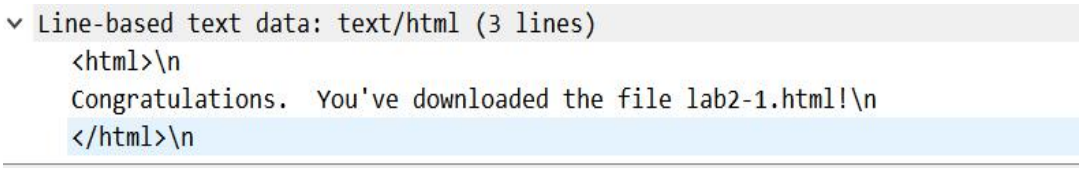
Question 4: How many bytes of content are being returned to the browser?



73 Bytes of content are being returned to the browser.

Question 5: What is the data contained inside the HTTP response packet?

It is a html file containing following information.



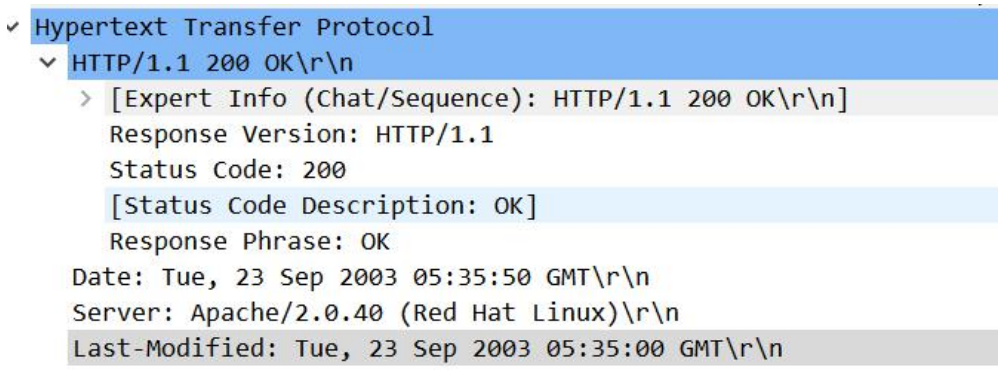
Exercise 4: Using Wireshark to understand the HTTP CONDITIONAL GET/response interaction

Question 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?

No, I can't see this line in the HTTP GET.

Question 2: Does the response indicate the last time that the requested file was modified?

Yes.Last modified date is Tue, 23 Sep 2003 05:35:00 GMT.



Question 3: Now inspect the contents of the second HTTP GET request from the browser to the server. Do you see an "IF-MODIFIED-SINCE:" and "IF-NONE-MATCH" lines in the HTTP GET? If so, what information is contained in these header lines?

```
Keep-Alive: 300\r\n
Connection: keep-alive\r\n
If-Modified-Since: Tue, 23 Sep 2003 05:35:00 GMT\r\n
If-None-Match: "1bfef-173-8f4ae900"\r\n
```

Yes.I see both “IF-MODIFIED-SINCE” and “IF-NONE-MATCH”.

“IF-MODIFIED-SINCE” contains the date and time that last modified(Tuesday, 23 Sep 2003 at 05:35 GMT).”IF-NONE-MATCH” contains "1bfef-173-8f4ae900".

Question 4: 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.

```
Hypertext Transfer Protocol
  HTTP/1.1 304 Not Modified\r\n
  Date: Tue, 23 Sep 2003 05:35:53 GMT\r\n
  Server: Apache/2.0.40 (Red Hat Linux)\r\n
  Connection: Keep-Alive\r\n
  Keep-Alive: timeout=10, max=99\r\n
  ETag: "1bfef-173-8f4ae900"\r\n
```

Status code: 304

Phrase : Not Modified

No.304 Not Modified will be returned to the client when the cached copy of a particular file is up to date with the server. Since the cached version is the same as the server-stored version,the server doesn’t need to return the file again.The browser can access the cached version of file.

Question 5: What is the value of the Etag field in the 2nd response message and how it is used? Has this value changed since the 1st response message was received?

```
Connection: Keep-Alive\r\n
Keep-Alive: timeout=10, max=99\r\n
ETag: "1bfef-173-8f4ae900"\r\n
\r\n
```

```
1f 2e 89 37 00 00 48 54  54 50 2f 31 2e 31 20 33   ·.·7··HT TP/1.1 3
30 34 20 4e 6f 74 20 4d  6f 64 69 66 69 65 64 0d   04 Not M odified·
0a 44 61 74 65 3a 20 54  75 65 2c 20 32 33 20 53   ·Date: T ue, 23 S
65 70 20 32 30 30 33 20  30 35 3a 33 35 3a 35 33   ep 2003  05:35:53
20 47 4d 54 0d 0a 53 65  72 76 65 72 3a 20 41 70    GMT··Se rver: Ap
61 63 68 65 2f 32 2e 30  2e 34 30 20 28 52 65 64   ache/2.0 .40 (Red
20 48 61 74 20 4c 69 6e  75 78 29 0d 0a 43 6f 6e    Hat Lin ux)··Con
6e 65 63 74 69 6f 6e 3a  20 4b 65 65 70 2d 41 6c   nection:  Keep-Al
69 76 65 0d 0a 4b 65 65  70 2d 41 6c 69 76 65 3a   ive··Kee p-Alive:
20 74 69 6d 65 6f 75 74  3d 31 30 2c 20 6d 61 78    timeout =10, max
3d 39 39 0d 0a 45 54 61  67 3a 20 22 31 62 66 65   =99··ETa g: "1bfe
66 2d 31 37 33 2d 38 66  34 61 65 39 30 30 22 0d   f-173-8f 4ae900"·
0a 0d 0a                                           ...
```

Etag value in the 2nd response message is “1bfef-173-8f4ae900”.

In Frame 10, the client receives response from the sever , when in Frame 14, client request the same file again with the same number in If-None-Match . The server checks the response number, Since the etag value does not need to send full response when the contents not changed.

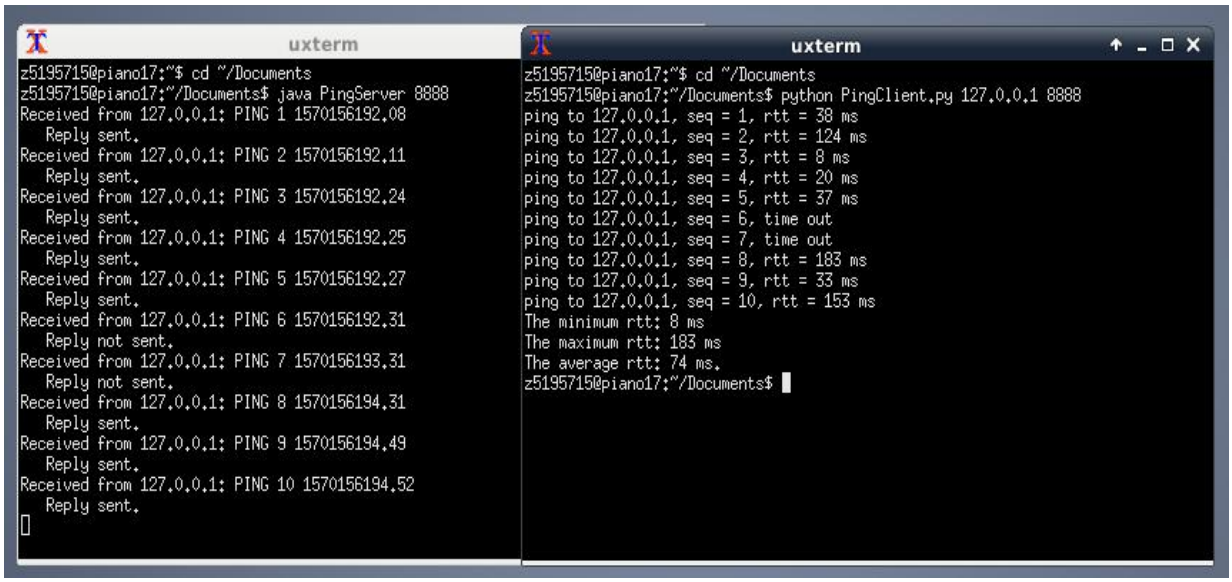
```
Hypertext Transfer Protocol
  HTTP/1.1 200 OK\r\n
    [Expert Info (Chat/Sequence): HTTP/1.1 200 OK\r\n]
    Response Version: HTTP/1.1
    Status Code: 200
    [Status Code Description: OK]
    Response Phrase: OK
  Date: Tue, 23 Sep 2003 05:35:50 GMT\r\n
  Server: Apache/2.0.40 (Red Hat Linux)\r\n
  Last-Modified: Tue, 23 Sep 2003 05:35:00 GMT\r\n
  ETag: "1bfef-173-8f4ae900"\r\n
```

```
Hypertext Transfer Protocol
  HTTP/1.1 304 Not Modified\r\n
    [Expert Info (Chat/Sequence): HTTP/1.1 304 Not Modified\r\n]
    Response Version: HTTP/1.1
    Status Code: 304
    [Status Code Description: Not Modified]
    Response Phrase: Not Modified
  Date: Tue, 23 Sep 2003 05:35:53 GMT\r\n
  Server: Apache/2.0.40 (Red Hat Linux)\r\n
  Connection: Keep-Alive\r\n
  Keep-Alive: timeout=10, max=99\r\n
  ETag: "1bfef-173-8f4ae900"\r\n
```

Compare 2 pictures, Value has not changed since 1st response message was received.

Exercise 5: Ping Client

Sample Output:



As is shown in the picture, 2 packages (6th & 7th) send time out and other 8 packages send successfully.

The max rtt, min rtt and average rtt are calculated and showed in the end of the program.

See PingClient.py for code.