Student Name: Luyang Ye
Student ID:5280537

Lab2

Exercise 3: Using Wireshark to understand basic HTTP request/response messages (marked, include in your report)

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

The status code returned from the server to the client browser is 200.

The phrase returned from the server to the client browser 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?

The HTML file that the browser is retrieving last modified at Tue, 23 Sep 2003 05:29:50 GMT.

Yes, the response does also contain a DATE header.

The difference between these two fields is that the value in the DATE header is Tue, 23 Sep 2003 05:29:00 GMT, which is 50 seconds later than the Last-Modifiled. In my opinon, this is because that the HTML file first need to be last modified, and then the DATE header will be update.

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

I think the connection established between the browser and the server is persistent. Because the content in the Connection header is Keep-Alive.

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

I think there are 73 bytes of content being returned to the browser, since the content of the Content-Length header is 73.

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

The data contained inside the HTTP response packet is as following:
<html>\n
Congratulations.	You’ve downloaded the file lab2-1.html!\n
</html>\n

Exercise 4: Using Wireshark to understand the HTTP CONDITIONAL GET/response interaction (marked, include in your report)

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 don’t see an “IF-MODIFIED-SINCE” line in the HTTP GET.

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

Yes, the response indicate the last time the requested file was modified at 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?

Yes. The "IF-MODIFIED-SINCE:" header contained the following information: Tue, 23 Sep 2003 05:35:00 GMT.

The "IF-NONE-MATCH" header contain the following information: "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.

The status code from the server in response to the second HTTP GET is 304. The phrase returned from the server is Not Modified.

I think the server did not explicitly return the contents of the file. Because the response to the second HTTP GET is much shorter than the first one, and I cannot find the Line-based text data in this response. In my opinion, since this is the second HTTP GET, and the phrase is Not Modified, the browser can load the content from its cache, therefore the sever didn't explicitly return the content of the 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 1 st response message was received?

The value of the Etag filed in the 2nd response message is "1bfef-173-8f4ae900". The Etag is used to identify whether the web server need to resend a full response packet. After add the Etag value in the response packet, if the client send request again and the Etag hasn't changed, then the server will not need to resend a full response pakcet.

Here, the value didn't change since the 1st response message was received.

Exercise 5: Ping Client (marked, submit source code as a separate file, include sample output in the report)

The PingClient.java should be executed with the following command:

```
java PingClient.java host port
```

```
e.g: java PingClient.java 127.0.0.1 5001
```

The results will be printed at the terminal.