Lab Exercise 2: HTTP & Socket Programming

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?

- `Status Code: 200`
- `Response Phrase: 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?

- `Last-Modified: Tue, 23 Sep 2003 05:29:00 GMT`
- The response's date header: `Date: Tue, 23 Sep 2003 05:29:50 GMT`
- `Last-Modified` is used to judge if resources have changed by checking `Last-Modified` and `If-Modified-Since` is the same. `DATE header` stores the date of this request.

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

- The connection is persistent
- Because the connection is `Keep-alive`

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

`File Data: 73 bytes`, 73 bytes of content are being returned to the browser

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

```
<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, there is not `"IF-MODIFIED-SINCE"` line in the HTTP GET

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

Yes. In HyperText Transfer Protocol, there is `Last-Modified: Tue, 23 Sep 2003 05:35:00 GMT\r\n`

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.
- `If-Modified-Since: Tue, 23 Sep 2003 05:35:00 GMT\r\n`
- `If-None-Match: "1bfef-173-8f4ae900"\r\n`

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.

- `Status Code: 304` and `Response Phrase: Not Modified`
- No. Because status code is 304 and it means not modified, therefore, when client send a request, server will tell the client to use cache.

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?

- 2nd response etag: `ETag: "1bfef-173-8f4ae900"\r\n`
- When client sends request to server, `If-None-Match` header has last returned Etag value from last response by server. When server second receives the client request, if it find the request header has `If-None-Match`, server will calculate the Etag, then, if the two Etag match, server assume no resource change and return `Status Code: 304` to client and server use cache.
- 1st response etag: `ETag: "1bfef-173-8f4ae900"\r\n`
- It doesn't changed as two Etag value is same.

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

Output

```
$ java PingClient 127.0.0.1 1998
ping to 127.0.0.1, seq = 3331, rtt = 331 ms
ping to 127.0.0.1, seq = 3332, rtt = 231 ms
ping to 127.0.0.1, seq = 3333, time out
ping to 127.0.0.1, seq = 3334, rtt = 174 ms
ping to 127.0.0.1, seq = 3335, rtt = 216 ms
ping to 127.0.0.1, seq = 3336, time out
ping to 127.0.0.1, seq = 3337, time out
ping to 127.0.0.1, seq = 3338, rtt = 92 ms
ping to 127.0.0.1, seq = 3339, rtt = 210 ms
ping to 127.0.0.1, seq = 3340, time out
ping to 127.0.0.1, seq = 3341, time out
ping to 127.0.0.1, seq = 3342, rtt = 263 ms
ping to 127.0.0.1, seq = 3343, rtt = 201 ms
ping to 127.0.0.1, seq = 3344, rtt = 232 ms
ping to 127.0.0.1, seq = 3345, rtt = 130 ms
Minimum rtt :92 ms
Maximum rtt :331 ms
Average rtt :208 ms
```

```
$ java PingServer 1998
Received from 127.0.0.1: PING seq = 3331 1601864060498
   Reply sent.
Received from 127.0.0.1: PING seq = 3332 1601864060879
   Reply sent.
Received from 127.0.0.1: PING seq = 3333 1601864061110
   Reply not sent.
Received from 127.0.0.1: PING seq = 3334 1601864061767
   Reply sent.
Received from 127.0.0.1: PING seq = 3335 1601864061941
   Reply sent.
Received from 127.0.0.1: PING seq = 3336 1601864062158
   Reply not sent.
Received from 127.0.0.1: PING seq = 3337 1601864062930
   Reply not sent.
Received from 127.0.0.1: PING seq = 3338 1601864063630
   Reply sent.
Received from 127.0.0.1: PING seq = 3339 1601864063722
   Reply sent.
Received from 127.0.0.1: PING seq = 3340 1601864063932
   Reply not sent.
Received from 127.0.0.1: PING seq = 3341 1601864064625
   Reply not sent.
Received from 127.0.0.1: PING seq = 3342 1601864065403
   Reply sent.
Received from 127.0.0.1: PING seq = 3343 1601864065666
   Reply sent.
Received from 127.0.0.1: PING seq = 3344 1601864065867
   Reply sent.
```

```
Received from 127.0.0.1: PING seq = 3345 1601864066099
   Reply sent.
```