COMP3331

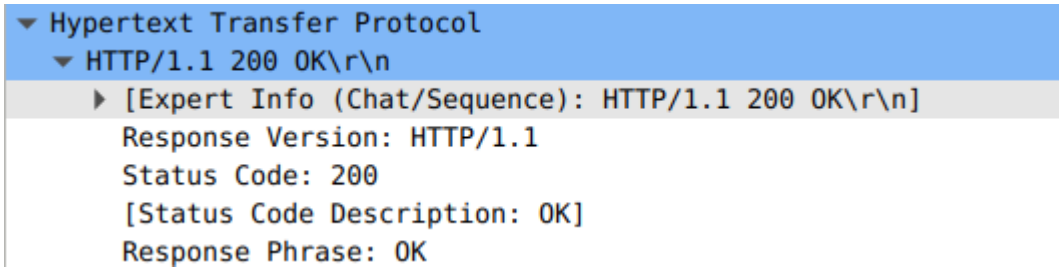
Lab 2

Dean So (z5204873)

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

1. *What is the status code and phrase returned from the server to the client browser?*

Status Code 200 and Response Phrase OK.



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?*

```
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
```

Date is a header for the time, day and date when the HTTP response was generated and Last-Modified is a validator header for the time at which the resource at the server was last modified. Thus, the HTML file that the browser is retrieving was last modified at the server on Tue, 23 Sep 2003 05:29:00 GMT.

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

```
Keep-Alive: timeout=10, max=100\r\n
Connection: Keep-Alive\r\n
```

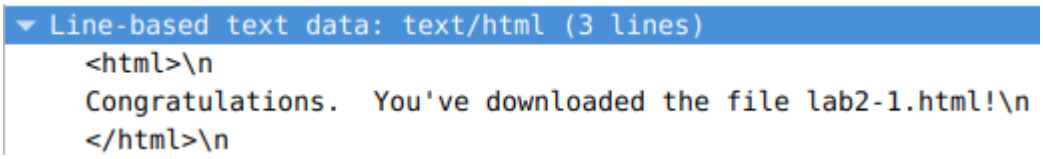
The connection is indeed a persistent connection type as it says “Keep-alive” (which is consistent with the connection type that HTTP/1.1 uses).

4. *How many bytes of content are being returned to the browser?*

```
Accept-Ranges: bytes\r\n
Content-Length: 73\r\n
```

73 bytes of content are being returned to the browser.

5. *What is the data contained inside the HTTP response packet?*



The data contained inside the HTTP response packet is the above text.

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

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?*

The first HTTP GET request does not have an "IF-MODIFIED-SINCE" line in the HTTP GET (because it is the first time you have accessed the website).

2. *Does the response indicate the last time that the requested file was modified?*

```
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
```

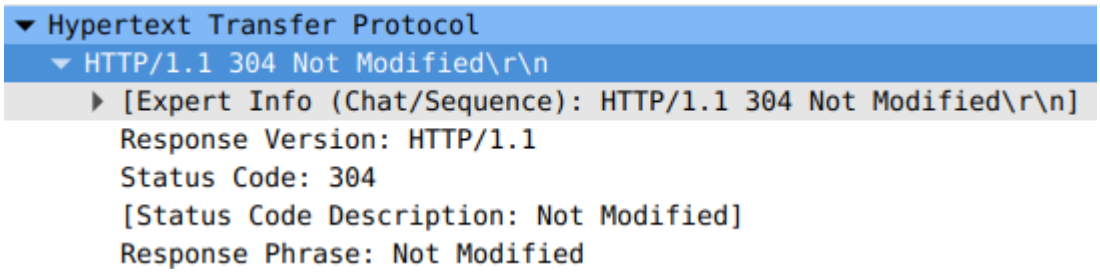
The response indicates the last time the requested file was modified was Tue, 23 Sep 2003 05:35:00 GMT.

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?*

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

The 2nd HTTP GET request from the browser does indeed contain both mentioned lines in the HTTP GET. The If-Modified-Since header contains the date, time and day the last time the content being requested has been modified – and thus the server does not need to send the content again. The If-None-Match header contains the content's Etag value.

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.*



Then HTTP status code is 304 and the phrase returned is “Not Modified”. The server did not explicitly return the contents of the file as the requested content has not been modified and thus does not require updating.

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?*

```
ETag: "1bfef-173-8f4ae900"\r\n
```

The Etag field value is as above. It is used as an identifier for the version of the resource. The Etag field value has not changed since the 1st response message was received. If the browser’s locally cached version of the resource matches the Etag of the web server’s version, the webserver will not send the resource again and will respond with “304 Not Modified”.

Exercise 5: Ping Client

Sample output of the my PingClient.py

```
ping to 127.0.0.1, seq = 3331, rtt = 71.0ms
ping to 127.0.0.1, seq = 3332, rtt = 193.0ms
ping to 127.0.0.1, seq = 3333, rtt = 188.0ms
ping to 127.0.0.1, seq = 3334, rtt = 179.0ms
ping to 127.0.0.1, seq = 3336, time out
ping to 127.0.0.1, seq = 3336, rtt = 165.0ms
ping to 127.0.0.1, seq = 3337, rtt = 75.0ms
ping to 127.0.0.1, seq = 3338, rtt = 28.0ms
ping to 127.0.0.1, seq = 3339, rtt = 106.0ms
ping to 127.0.0.1, seq = 3340, rtt = 149.0ms
ping to 127.0.0.1, seq = 3341, rtt = 59.0ms
ping to 127.0.0.1, seq = 3342, rtt = 174.0ms
ping to 127.0.0.1, seq = 3343, rtt = 49.0ms
ping to 127.0.0.1, seq = 3345, time out
ping to 127.0.0.1, seq = 3346, time out
min/max/avg rtt 28.0ms 193.0ms 120.0ms
```