COMP9331 LAB 2 Report

Z5219960 Heng-Chuan Lin

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?

   Ans:

   Status: 200 & 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?

   Ans:

   HTML file that the browser is retrieving last modified at the server was
   Last-Modified: Tue, 23 Sep 2003 05:29:00 GMT\r\n
   The DATE header is
   Date: Tue, 23 Sep 2003 05:29:50 GMT\r\n

   Last-Modified means the time when the object was created or modified.
   Date means the time when server responded.

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

   Ans:

   It's persistent connection.
   According to the header, the Connection fields indicate that is Keep-Alive.
   Keep-Alive is default setting in HTTP/1.1

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

   Ans: 73 bytes according to the File Data row in the record.

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

   Ans: it's a HTML file.

<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

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?

   Ans:

   No.

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

   Ans:

   Yes, it’s
   Last-Modified: Tue, 23 Sep 2003 05:35:00 GMT\r\n

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?

   Ans:

   Yes, both of them are observed.
   If-Modified-Since: Tue, 23 Sep 2003 05:35:00 GMT\r\n
   If-None-Match: "1bfef-173-8f4ae900"\r\n
   If-Modified-Since means after this date, if the last modified time of this object didn’t change, send the respond without contain.
   If-None-Match means if the identification (ETag) of the object didn’t change, then send the response without contain.

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.

   Ans:

   Status: 304 & Phrase: Not Modified.
   since the both last-modified timestamp and Etag the same, the response was sent without contents of file.

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?

Ans:

Both Etag of files in the 2nd and 1st are
1bfef-173-8f4ae900.
The Etag in HTTP response header is an ID for specific version of an object. Etag improves efficiency of web caching i.e. web servers could save bandwidth by sending a response without content if the object is the same.

Reference: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/ETag

Exercise 5: Ping Client

```
HCLLLLLL:lab2 hcl$ java PingServer 1044
Received from 127.0.0.1: PING 0 2019-10-07 15:33:47.608663
   Reply not sent.
Received from 127.0.0.1: PING 1 2019-10-07 15:33:48.611263
   Reply sent.
Received from 127.0.0.1: PING 2 2019-10-07 15:33:48.662925
   Reply not sent.
Received from 127.0.0.1: PING 3 2019-10-07 15:33:49.664359
   Reply sent.
Received from 127.0.0.1: PING 4 2019-10-07 15:33:49.696464
   Reply sent.
Received from 127.0.0.1: PING 5 2019-10-07 15:33:49.752445
   Reply not sent.
Received from 127.0.0.1: PING 6 2019-10-07 15:33:50.755915
   Reply sent.
Received from 127.0.0.1: PING 7 2019-10-07 15:33:50.953679
   Reply sent.
Received from 127.0.0.1: PING 8 2019-10-07 15:33:51.067841
   Reply sent.
Received from 127.0.0.1: PING 9 2019-10-07 15:33:51.227446
   Reply sent.
```

Server side

```
HCLLLLLL:lab2 hcl$ python PingClient.py 127.0.0.1 1044
ping to 127.0.0.1, seq = 0, rtt = timeout
ping to 127.0.0.1, seq = 1, rtt = 51 ms
ping to 127.0.0.1, seq = 2, rtt = timeout
ping to 127.0.0.1, seq = 3, rtt = 32 ms
ping to 127.0.0.1, seq = 4, rtt = 55 ms
ping to 127.0.0.1, seq = 5, rtt = timeout
ping to 127.0.0.1, seq = 6, rtt = 197 ms
ping to 127.0.0.1, seq = 7, rtt = 114 ms
ping to 127.0.0.1, seq = 8, rtt = 159 ms
ping to 127.0.0.1, seq = 9, rtt = 21 ms
min_rtt: 21 ms, max_rtt: 197 ms, avg_rtt: 89 ms
```

Client side