

## COMP9331-21T2-Lab2

Haojin Guo

(z5216214)

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

The image shows a Wireshark capture of an HTTP response. The top pane displays a list of packets, with packet 12 selected. The middle pane shows the details of the selected packet, and the bottom pane shows the raw hex data.

No.	Time	Source	Destination	Protocol	Length	Info
10	4.694850	192.168.1.102	128.119.245.12	HTTP	555	GET /ethereal-labs/lab2-1.html HTTP/1.1
12	4.718993	128.119.245.12	192.168.1.102	HTTP	439	HTTP/1.1 200 OK (text/html)
13	4.724332	192.168.1.102	128.119.245.12	HTTP	541	GET /favicon.ico HTTP/1.1
14	4.750366	128.119.245.12	192.168.1.102	HTTP	1395	HTTP/1.1 404 Not Found (text/html)

Frame 12: 439 bytes on wire (3512 bits), 439 bytes captured (3512 bits)  
> Ethernet II, Src: LinksysG\_da:af:73 (00:06:25:da:af:73), Dst: Dell\_4f:36:23 (00:08:74:4f:36:23)  
> Internet Protocol Version 4, Src: 128.119.245.12, Dst: 192.168.1.102  
> Transmission Control Protocol, Src Port: 80, Dst Port: 4127, Seq: 1, Ack: 502, Len: 385  
▼ Hypertext Transfer Protocol  
    > HTTP/1.1 200 OK\r\n  
        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  
        ETag: "1bfed-49-79d5bf00"\r\n  
        Accept-Ranges: bytes\r\n  
    > 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  
        \r\n  
    [HTTP response 1/2]  
    [Time since request: 0.024143000 seconds]  
    [\[Request in frame: 10\]](#)  
    [\[Next request in frame: 13\]](#)  
    [\[Next response in frame: 14\]](#)  
    [Request URI: http://gaia.cs.umass.edu/favicon.ico]  
    File Data: 73 bytes  
▼ Line-based text data: text/html (3 lines)  
    <html>\r\n

0000 00 08 74 4f 36 23 00 06 25 da af 73 08 00 45 00 ..t06#...%..s..E..  
0010 01 a9 b6 fa 40 00 37 06 53 c2 80 77 f5 0c c0 a8 ...@-7 S-w...  
0020 01 66 00 50 10 1f 6b a6 54 92 f5 32 66 a7 50 18 ..f.P..k..T..2f.P..  
0030 19 20 7a 1c 00 00 48 54 54 50 2f 31 2e 31 20 32 ..z...HT TP/1.1 2

#### Question 1.

- Status Code: 200
- Returned Phrase: OK

#### Question 2.

- Last modified at the server:  
Last-Modified: Tue, 23 Sep 2003 05:29:00 GMT
- The response contains a DATE header.  
Date: Tue, 23 Sep 2003 05:29:50 GMT
- Date represents that the time when the Web was created.  
However, the last-Modified shows the last modification time of the page.

#### Question 3.

- The connection established is persistent between the browser and the sever.
- The reason is showed by,  
Connection: Keep-Alive

Also, HTTP 1.1 can show the connection is persistent.

#### Question 4.

There are 73 bytes of content are being returned to the browser, according to the fact that Content-Length: 73\r\n .

#### Question 5.

```
> Transmission Control Protocol, Src Port: 80, Dst Port: 4127, Seq: 1, Ack: 502, Len: 385
  > Hypertext Transfer Protocol
    > HTTP/1.1 200 OK\r\n
      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
      ETag: "1bfed-49-79d5bf00"\r\n
      Accept-Ranges: bytes\r\n
    > 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
      \r\n
      [HTTP response 1/2]
      [Time since request: 0.024143000 seconds]
      [Request in frame: 10]
      [Next request in frame: 13]
      [Next response in frame: 14]
      [Request URI: http://gaia.cs.umass.edu/favicon.ico]
      File Data: 73 bytes
  > Line-based text data: text/html (3 lines)
    <html>\n
    Congratulations. You've downloaded the file lab2-1.html!\n
    </html>\n
```

By the part of “Line-based text data: text/html (3 lines)”, the response packet is “Congratulations. You've downloaded the file lab2-1.html!”

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

The image displays a Wireshark capture of an HTTP conditional GET interaction. The packet list at the top shows three frames:

No.	Time	Source	Destination	Protocol	Length	Info
8	2.331268	192.168.1.102	128.119.245.12	HTTP	555	GET /ethereal-labs/lab2-2.html HTTP/1.1
10	2.357902	128.119.245.12	192.168.1.102	HTTP	739	HTTP/1.1 200 OK (text/html)
14	5.517390	192.168.1.102	128.119.245.12	HTTP	668	GET /ethereal-labs/lab2-2.html HTTP/1.1
15	5.540216	128.119.245.12	192.168.1.102	HTTP	243	HTTP/1.1 304 Not Modified

The packet details for frame 10 (HTTP/1.1 200 OK) are expanded, showing the following headers and fields:

- 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
- Accept-Ranges: bytes\r\n
- Content-Length: 371\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

The packet bytes show the HTML content of the response, which includes a congratulatory message and a note about the file's last modification date.

### Question 1.

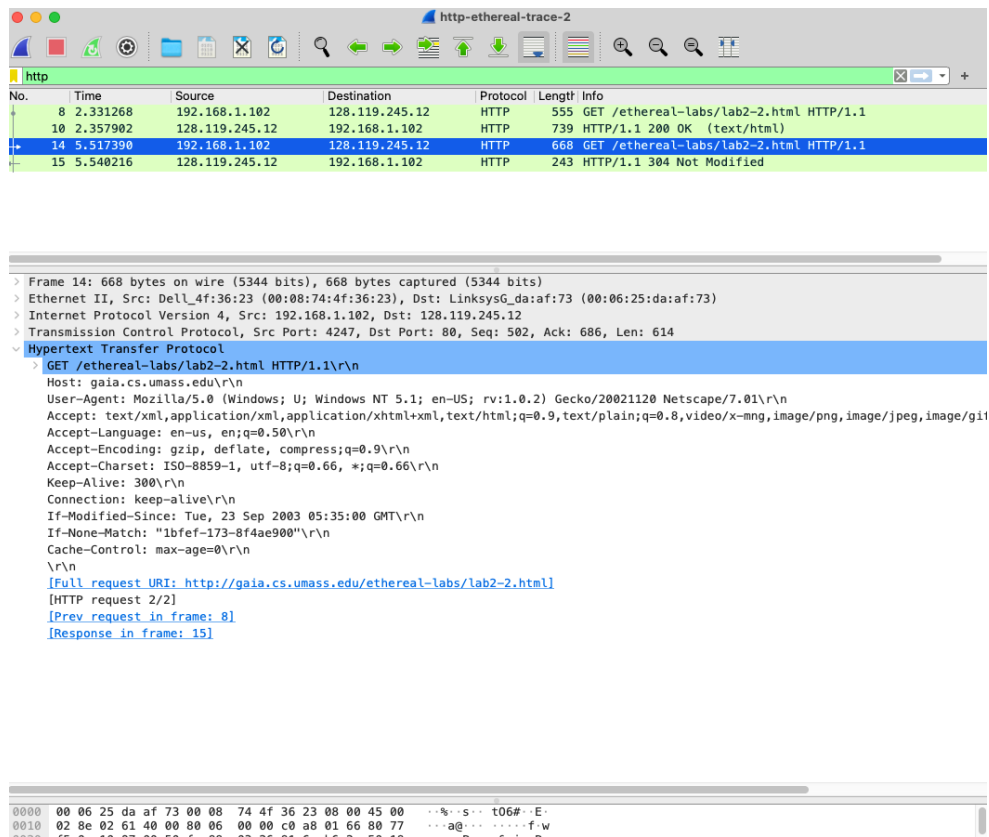
No, the header “IF-MODIFIED-SINCE” is not present in the first HTTP GET request. This is because this Header exists after the Cache and is used for cache detection together with Etag.

### Question 2.

Yes. The response does indicate that the file was last modified on Tue, 23 Sep 2003 05:35:00 GMT.

### Question 3.

The detail of second HTTP GET request is as follows.

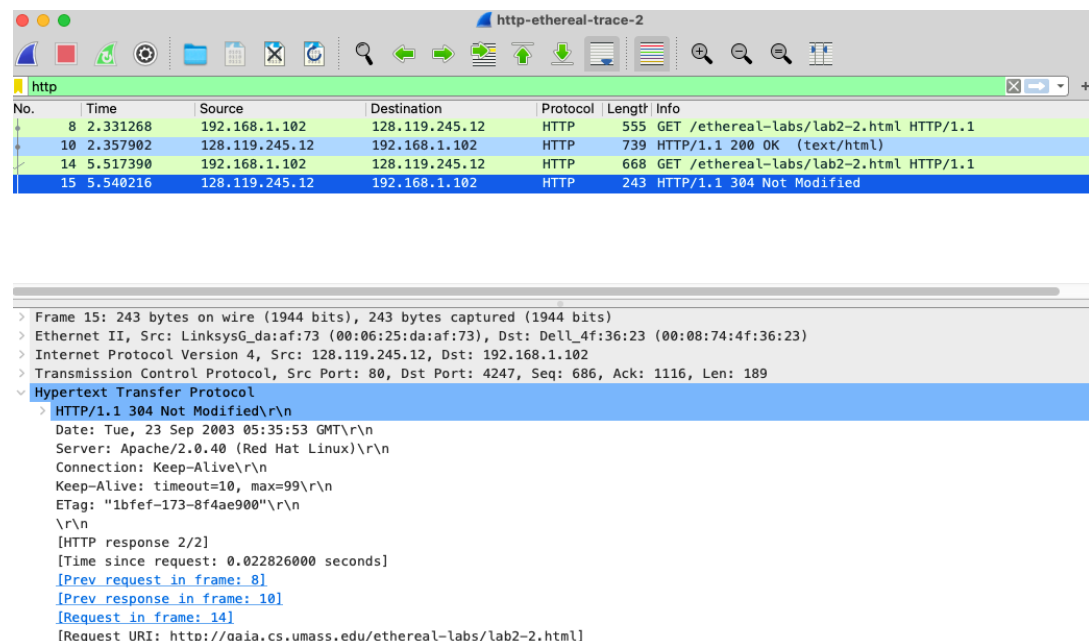


The second GET request does contain the part of "If-Modified-Since" and "If-None-Match" .

a) If-Modified-Since: Tue, 23 Sep 2003 05:35:00 GMT

b) If-None-Match: "1bfef-173-8f4ae900"

#### Question 4.



- a) Status code: 304
- b) Phrase returned from the server: Not Modified
- c) No, the server does not respond back with the requested file. Because, the server has not modified the page, and the browser can simply show the locally cached version of this file.

#### Question 5.

- a) The entity tag (Tag) value is "1bfef-173-8f4ae900".
- b) ETag is used in conjunction with "If-None-Match" header field.

#### Exercise 5: Ping Client.

Server message:

```
[^C(base) MacBook-Pro-Hankin:lab2 guohaojin$ javac PingServer.java
(base) MacBook-Pro-Hankin:lab2 guohaojin$ java PingServer 3000
Received from 127.0.0.1: PING3331 -- 2021-06-21 21:36:58.375165
  Reply sent.
Received from 127.0.0.1: PING3332 -- 2021-06-21 21:36:58.569830
  Reply sent.
Received from 127.0.0.1: PING3333 -- 2021-06-21 21:36:58.654749
  Reply sent.
Received from 127.0.0.1: PING3334 -- 2021-06-21 21:36:58.677074
  Reply sent.
Received from 127.0.0.1: PING3335 -- 2021-06-21 21:36:58.682076
  Reply sent.
Received from 127.0.0.1: PING3336 -- 2021-06-21 21:36:58.716723
  Reply sent.
Received from 127.0.0.1: PING3337 -- 2021-06-21 21:36:58.797757
  Reply sent.
Received from 127.0.0.1: PING3338 -- 2021-06-21 21:36:58.837863
  Reply sent.
Received from 127.0.0.1: PING3339 -- 2021-06-21 21:36:58.881129
  Reply sent.
Received from 127.0.0.1: PING3340 -- 2021-06-21 21:36:58.975586
  Reply not sent.
Received from 127.0.0.1: PING3341 -- 2021-06-21 21:36:59.577908
  Reply sent.
Received from 127.0.0.1: PING3342 -- 2021-06-21 21:36:59.732048
  Reply not sent.
Received from 127.0.0.1: PING3343 -- 2021-06-21 21:37:00.333340
  Reply sent.
Received from 127.0.0.1: PING3344 -- 2021-06-21 21:37:00.491342
  Reply sent.
Received from 127.0.0.1: PING3345 -- 2021-06-21 21:37:00.514920
  Reply sent.
```

Client message:

```
(base) MacBook-Pro-Hankin:lab2 guohaojin$ python3 PingClient.py 127.0.0.1 3000
Ping to 127.0.0.1, seq = 3331, rtt = 194 ms
Ping to 127.0.0.1, seq = 3332, rtt = 85 ms
Ping to 127.0.0.1, seq = 3333, rtt = 22 ms
Ping to 127.0.0.1, seq = 3334, rtt = 5 ms
Ping to 127.0.0.1, seq = 3335, rtt = 35 ms
Ping to 127.0.0.1, seq = 3336, rtt = 81 ms
Ping to 127.0.0.1, seq = 3337, rtt = 40 ms
Ping to 127.0.0.1, seq = 3338, rtt = 43 ms
Ping to 127.0.0.1, seq = 3339, rtt = 94 ms
Ping to 127.0.0.1, seq = 3340, rtt = time out
Ping to 127.0.0.1, seq = 3341, rtt = 154 ms
Ping to 127.0.0.1, seq = 3342, rtt = time out
Ping to 127.0.0.1, seq = 3343, rtt = 158 ms
Ping to 127.0.0.1, seq = 3344, rtt = 23 ms
Ping to 127.0.0.1, seq = 3345, rtt = 32 ms

In 15 packets, there are 13 packets received.
The minimum RTT is 5 ms
The maximum RTT is 194 ms
The average RTT is 74 ms
(base) MacBook-Pro-Hankin:lab2 guohaojin$
```