Patrick Austin

CPE 400 Homework 1

9/24/2017

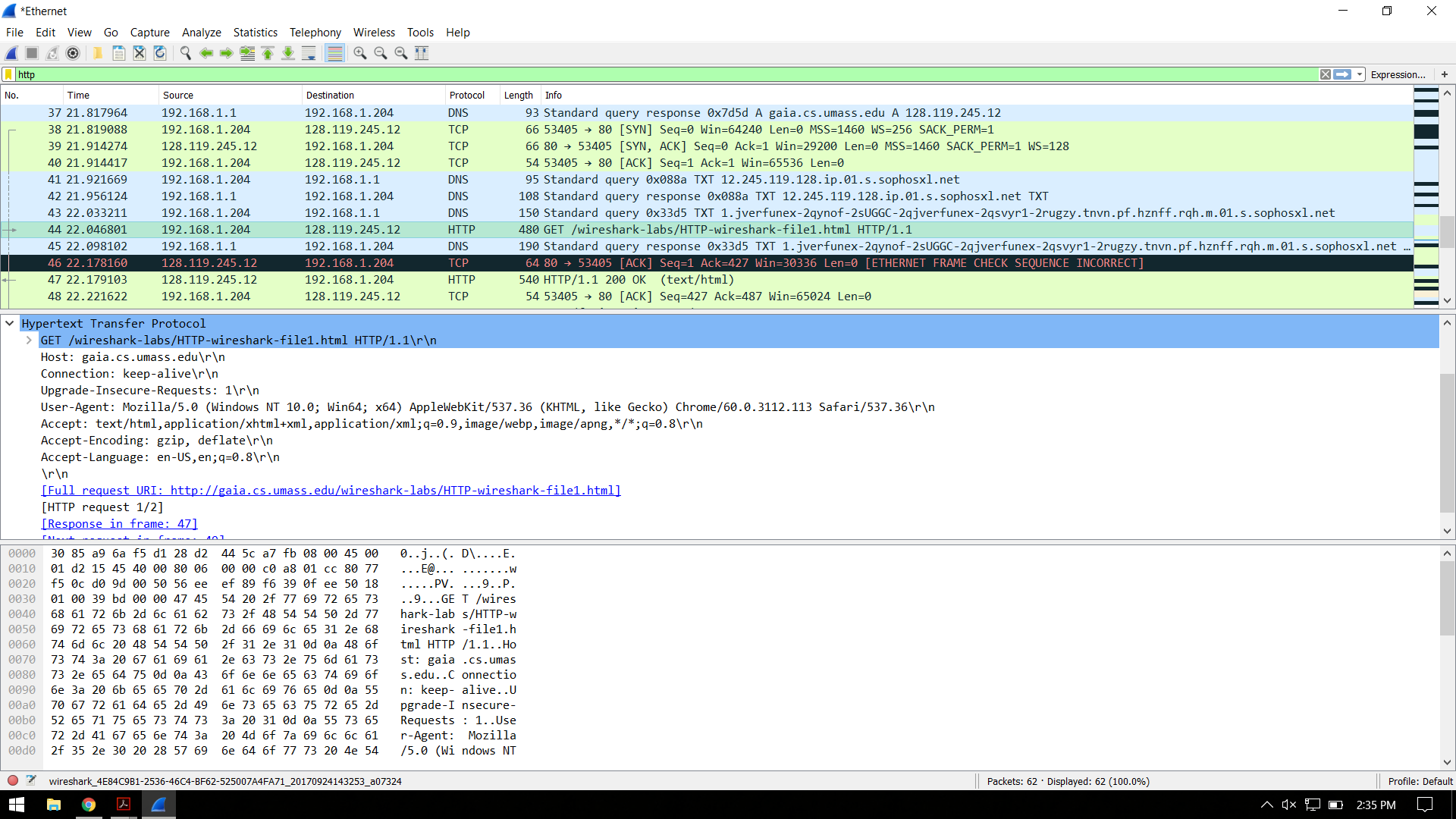
1.

a. Link A-B transmits 1000 bytes, or 8000 bits, over a link which transfers 1 Megabit/second. Therefore the delay is 8000 bits / 1 x 106 bits/second, for a delay of .008 seconds, or 8 milliseconds.

Likewise, link B-C transmits 8000 bits over a link which transfers 2 Megabits/second. Therefore the delay is 8000 bits / 2 x 106 bits/second, for a delay of .004 seconds, or 4 milliseconds.

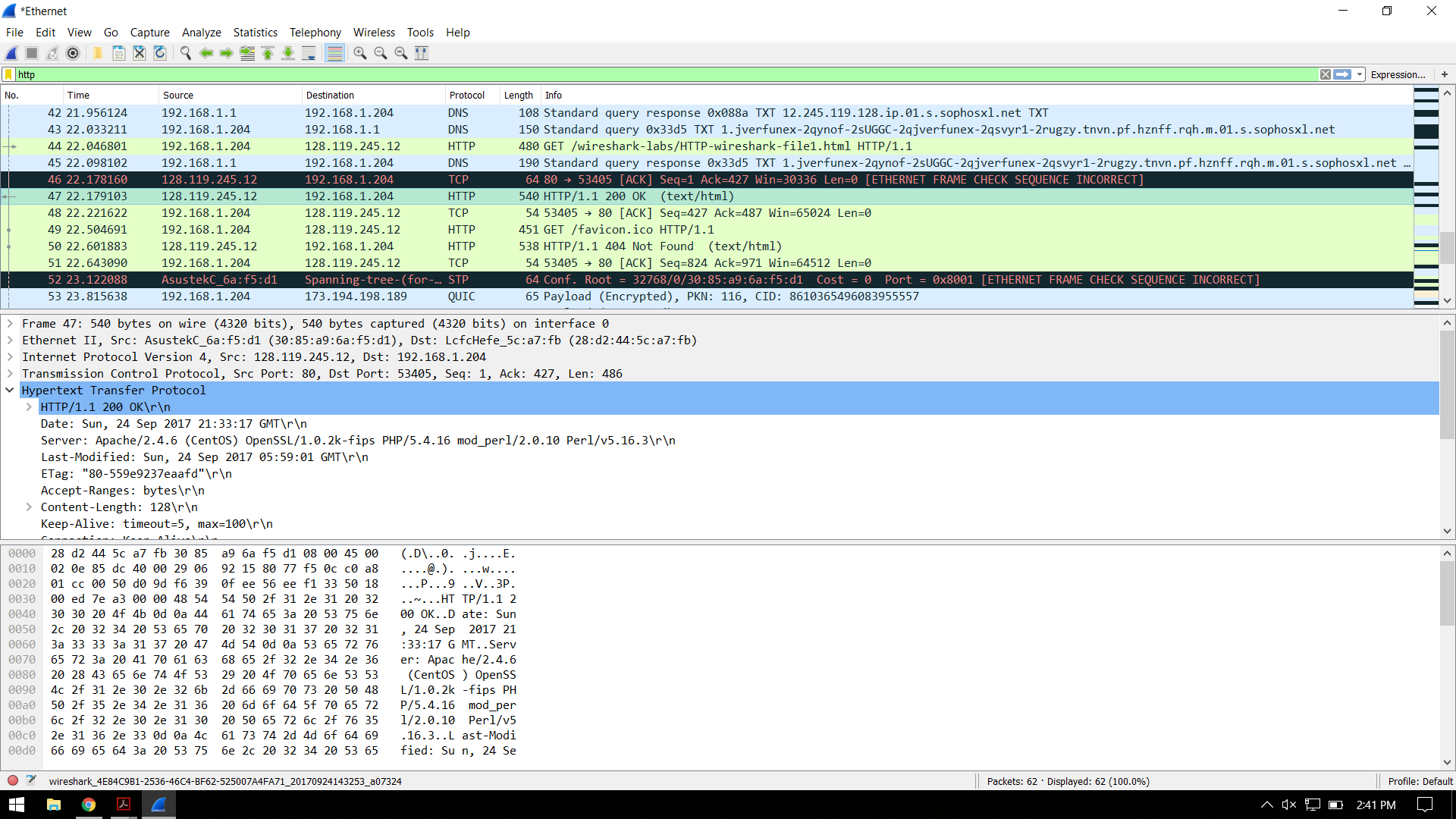
b. The total delay, then, is 8 milliseconds of transmission delay from A-B, 4 milliseconds of queueing and processing delay at B, and 4 milliseconds of transmission delay from B-C. The total delay, then, is .016 seconds, or 16 milliseconds.

2.



a. My browser is running HTTP 1.1.

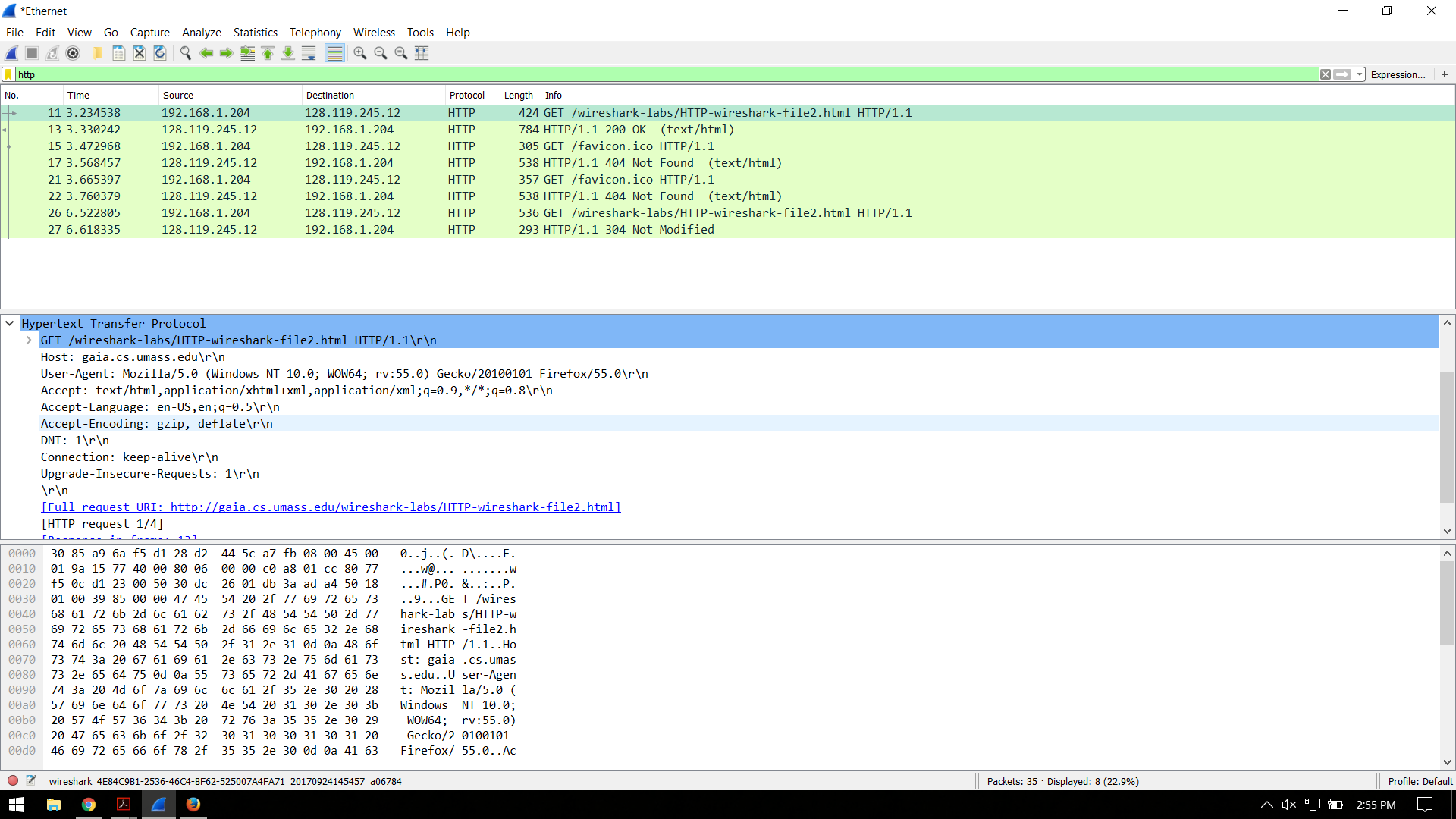
b. The IP address of my computer is 192.168.1.204 and the address of the gaia.cs.umass.edu server is 128.119.245.12.

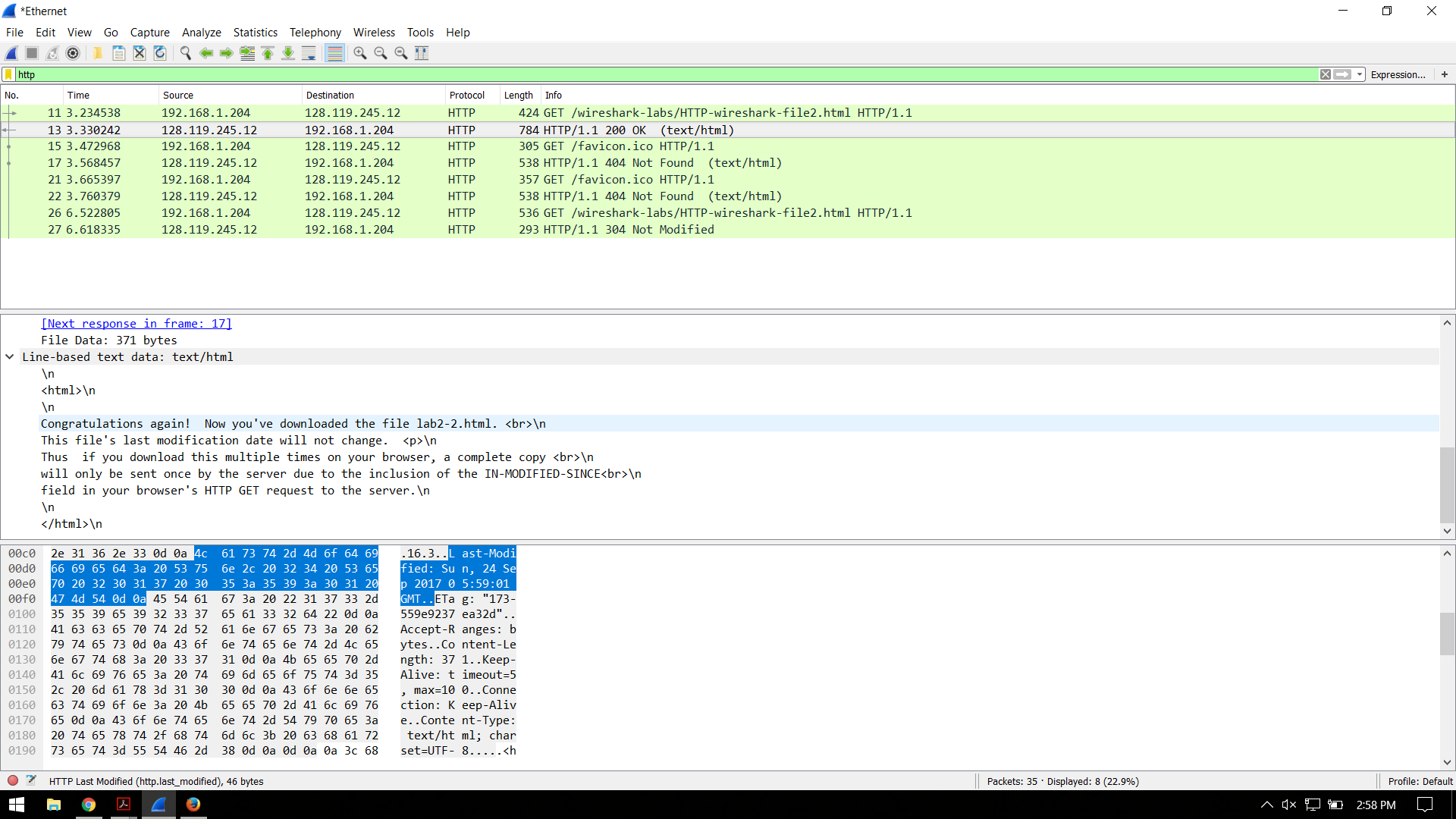


c. The server returned status code 200, denoting an OK, to my browser.

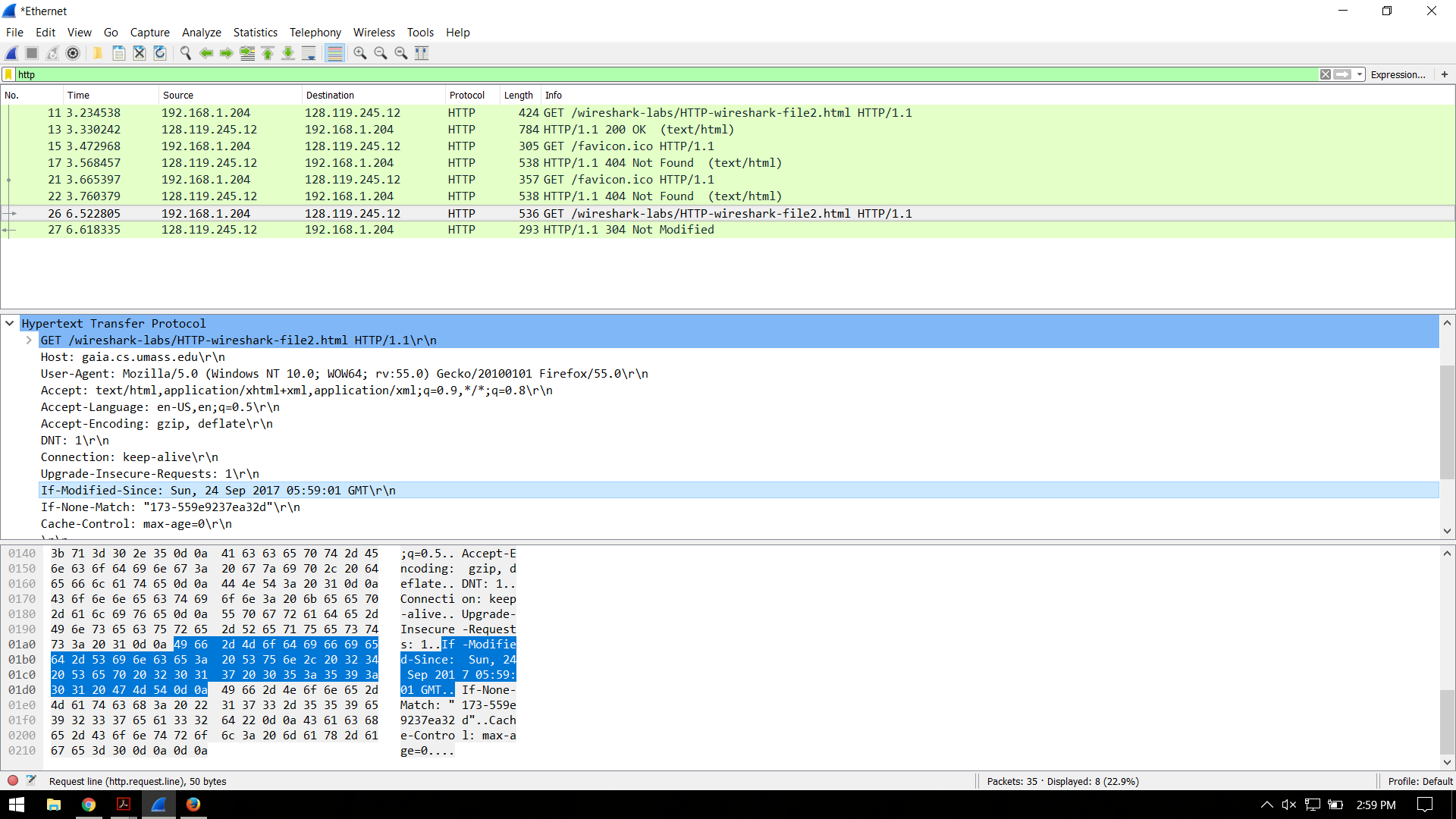
d. The HTML file was last modified by the server on September 24, 2017.

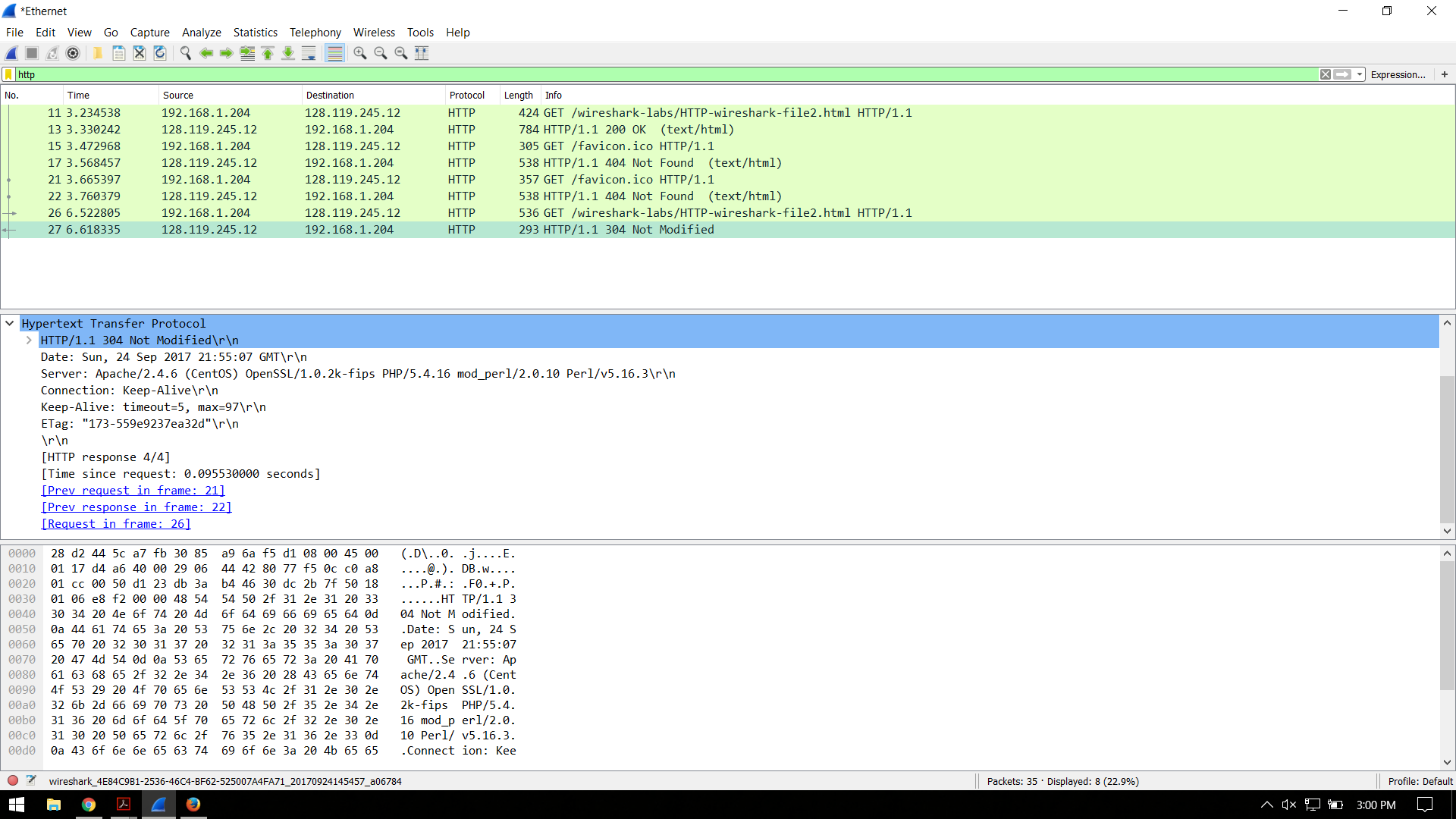
3.





a. There is no IF-MODIFIED-SINCE line in the GET, as shown in the first screenshot above. The server did explicitly return the requested contents of the file, as seen in the second screenshot.

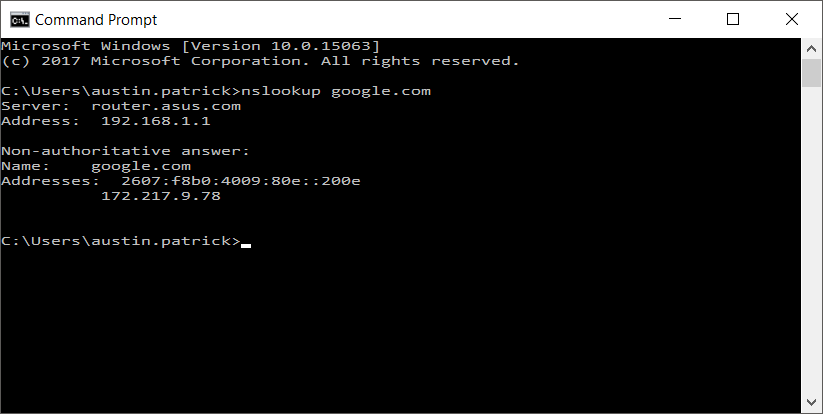




b. The second GET request does have an IF-MODIFIED-SINCE line, as shown in the first screenshot. A timestamp follows on this line, denoting that there is no need to return the file unless it has been modified since the given timestamp; in other words, the version available at that timestamp has been saved in the cache and won’t be redownloaded unless necessary.

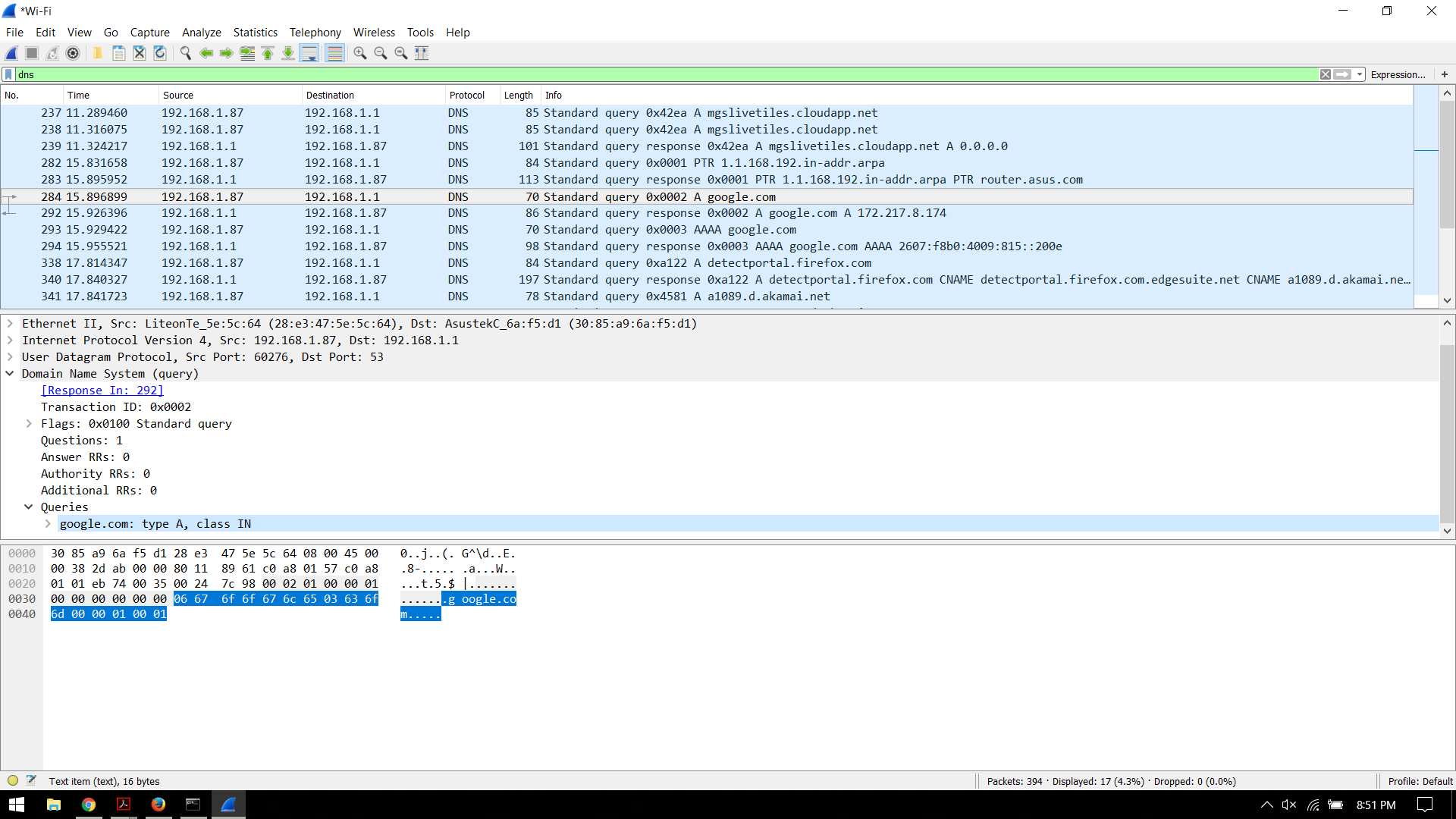
As shown in the second screenshot, the server returns a 304 status code, with the phrase ‘not modified’. This means that the file on the server has not been modified since the timestamp given in the get, so the cached version is the latest and no download of an updated version is necessary.

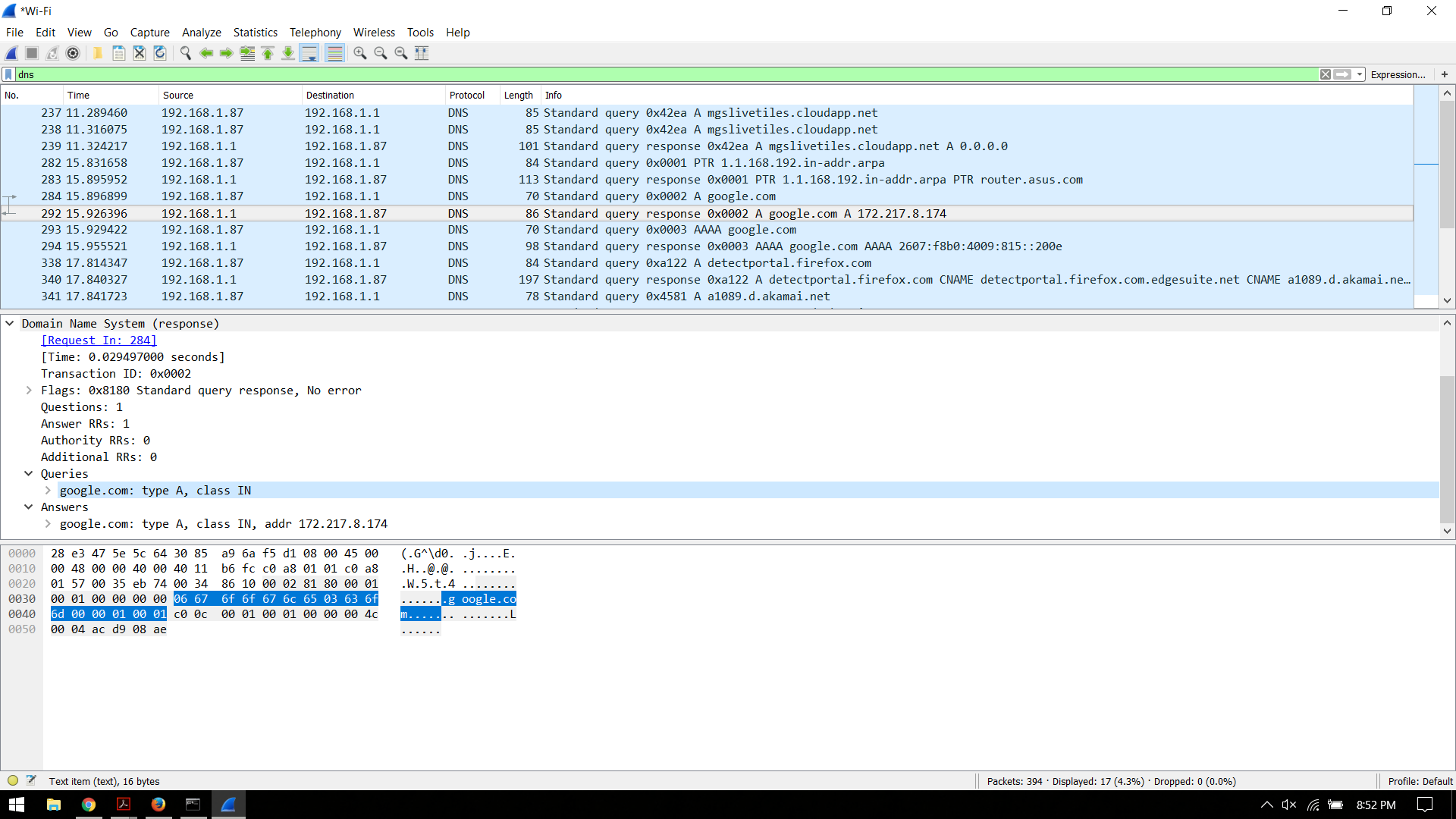
4.



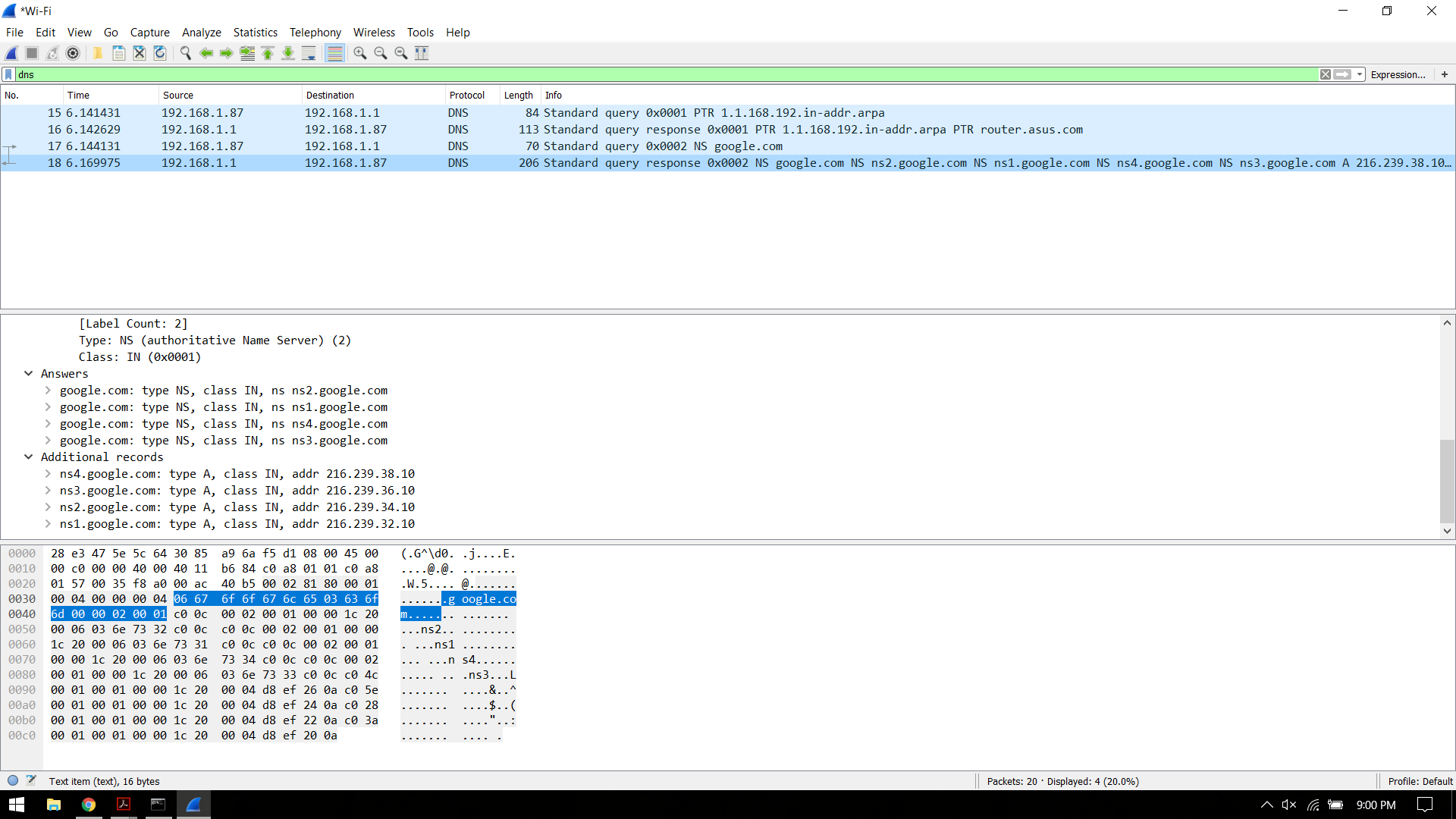
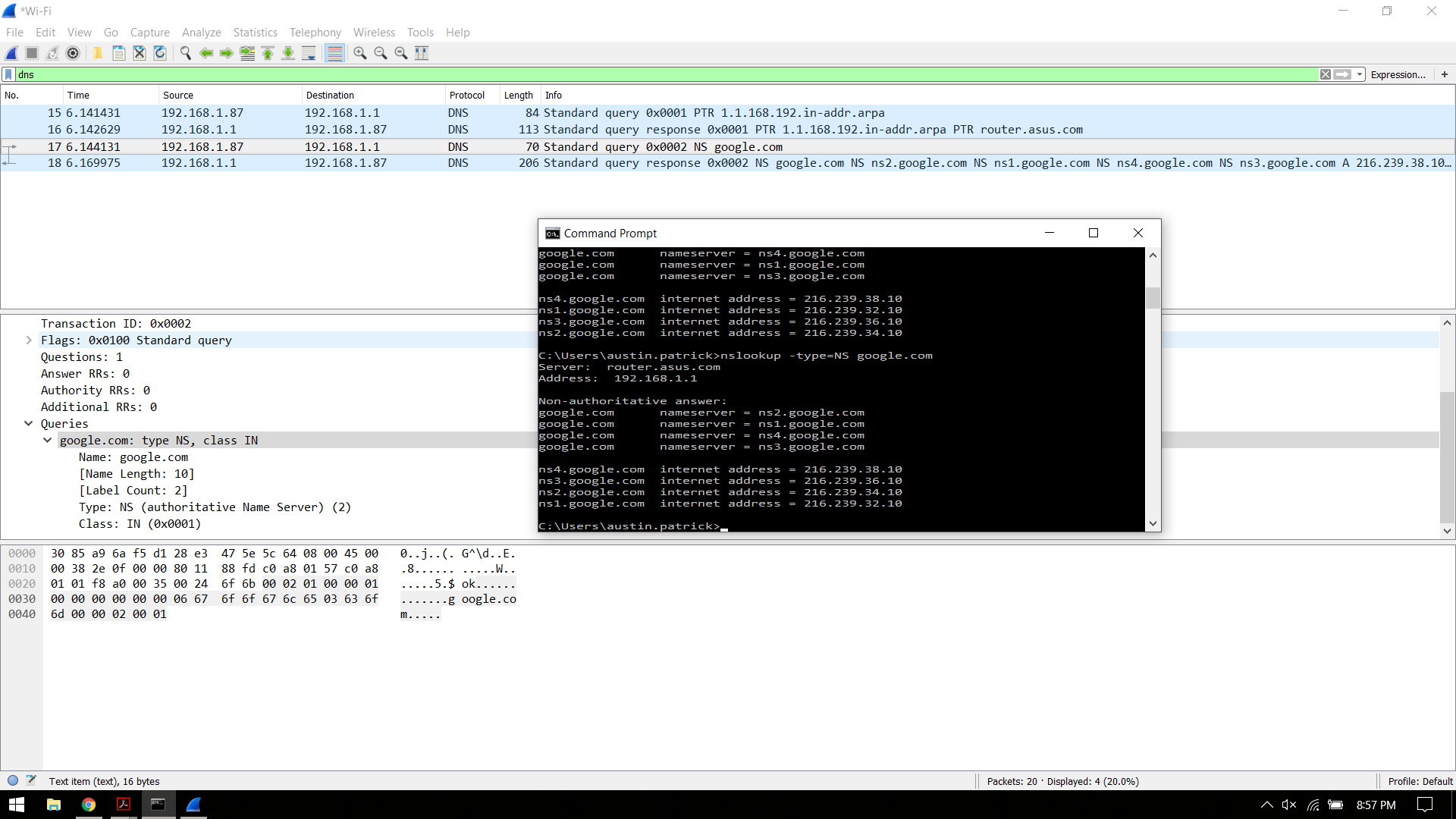
a. The command is nslookup google.com, and the resulting address is 172.217.9.78.

b. The server that returned this answer is located at 192.168.1.1 (my router). The answer, as shown above, is non-authoritative.



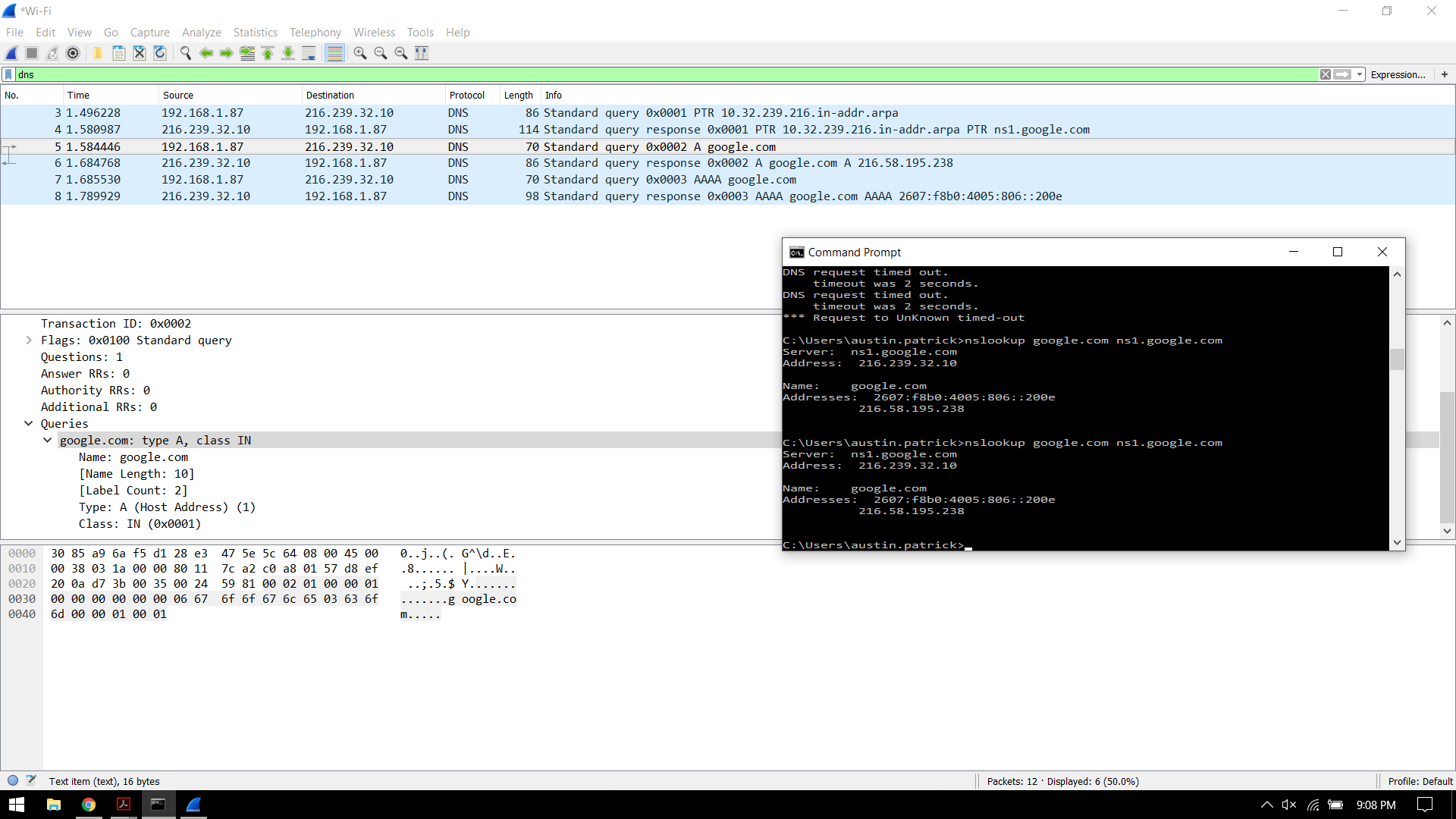


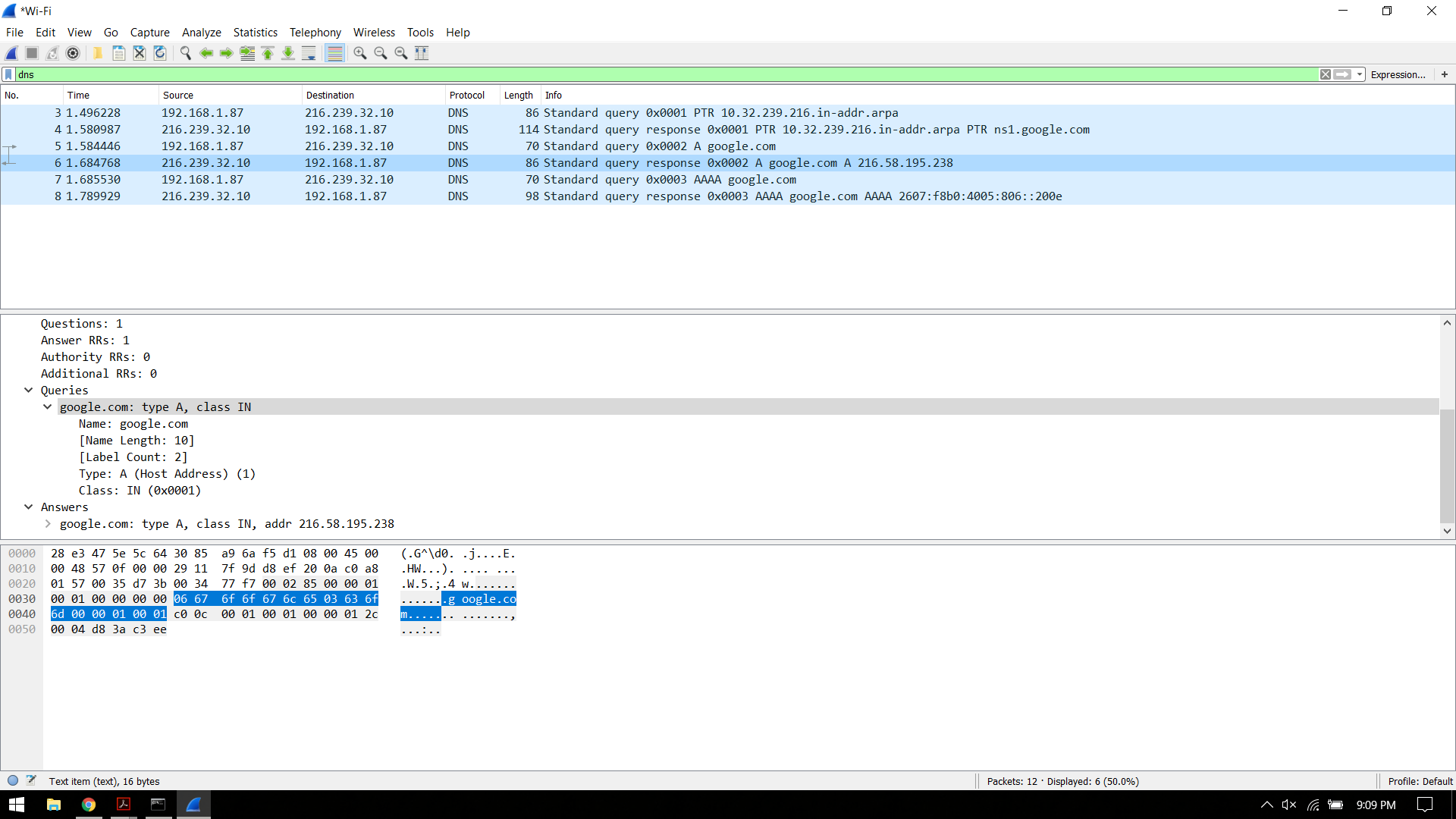
c. Running Wireshark in the background of the nslookup google.com request shows the process of my computer making a type A DNS query for google.com (first screenshot) and receiving an answer with the address from my router’s DNS cache (second screenshot). This also happens for the IPv6 address, using a request and response of type AAAA.



d. The command nslookup -type=NS google.com provides the authoritative DN servers for google.com, as shown in the first screenshot above. Wireshark shows the DNS request, also shown in the first screenshot, with type NS, as expected.

The DNS server’s response, shown in the second screenshot, shows the requested information. The response contains the four google nameservers and their IP addresses. Since this lookup still from my router’s cache, they are still considered non-authoritative answers.





e. The command is nslookup google.com ns1.google.com, as shown in the first screenshot. Analysis via wireshark shows two request/response pairs: in the requests, my computer asks for the IPv4 and IPv6 addresses of google.com from ns1.google.com, and in the responses these are provided by ns1.google.com back to my computer. Wireshark shows that the IPv4 request is A type, and the IPv6 request is AAAA type.

That is, my computer at 192.168.1.87 asks ns1.google.com at 216.239.32.10 for google.com’s address; ns1.google.com provides these in IPv4 and IPv6 formats: the addresses for google.com are 216.58.195.238 and 2607:f8b0:4005:806::200e, respectively.