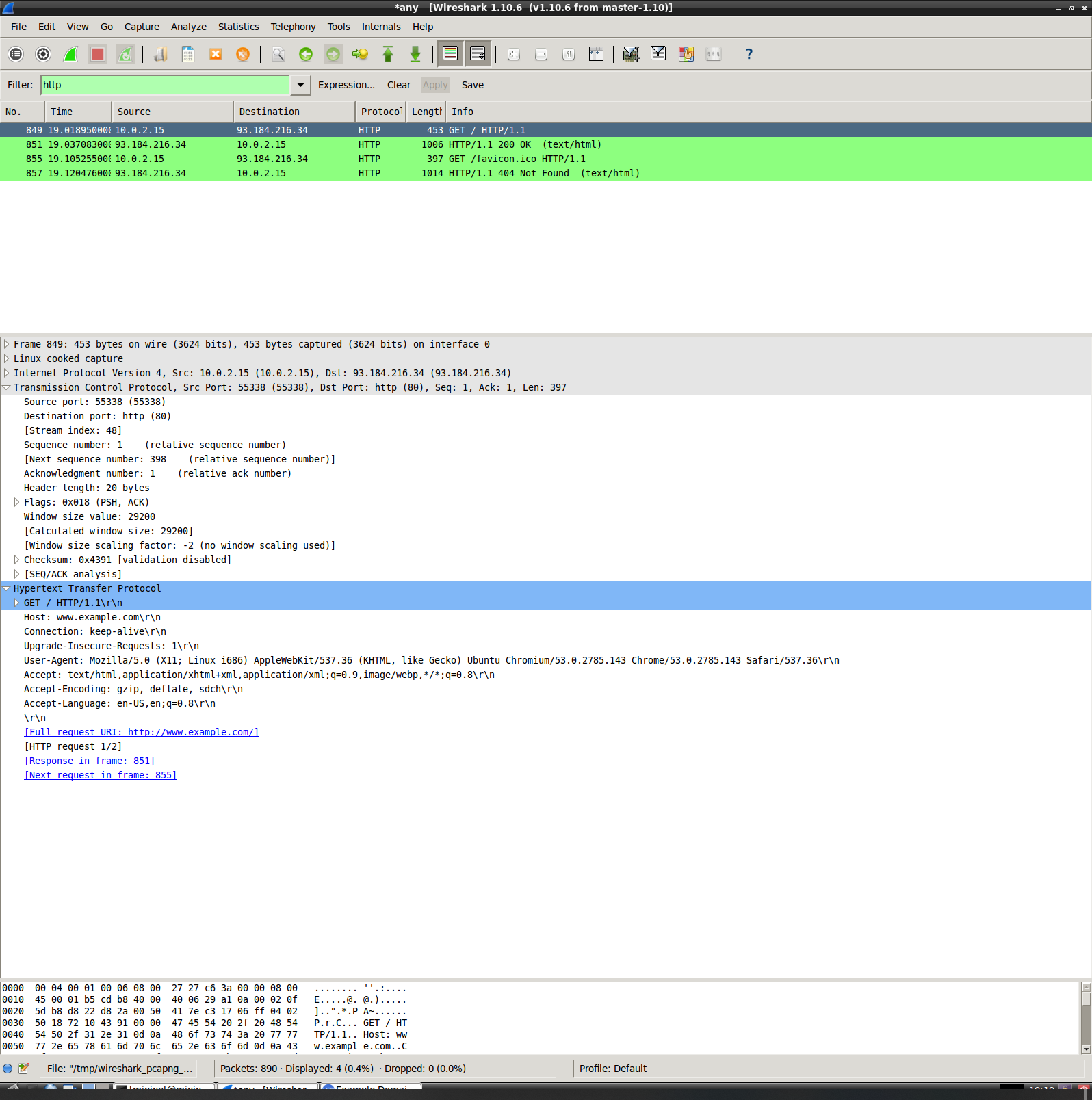
Chris Sclipei

3/10/19

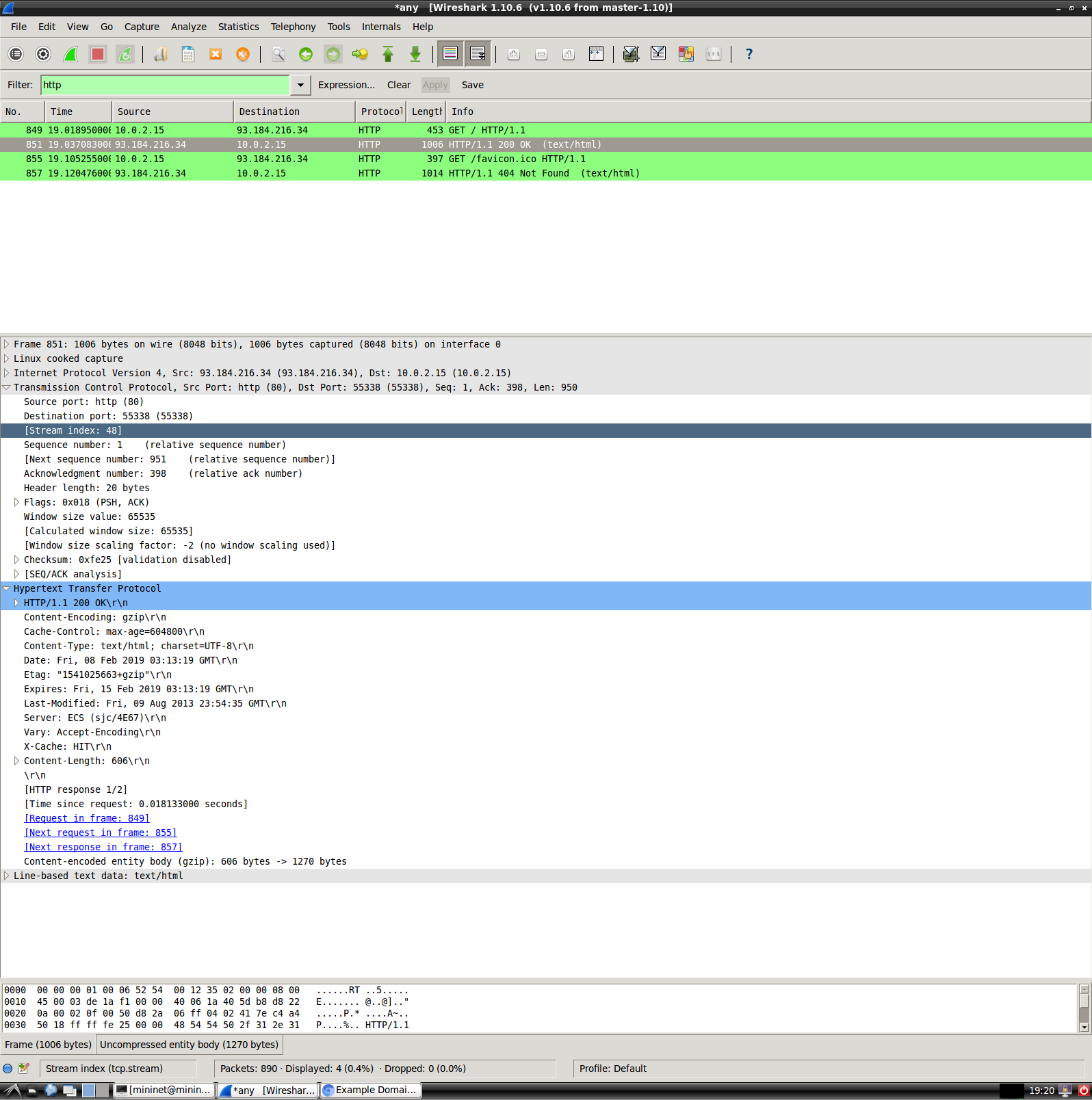
CMPE 150

**Lab 2**

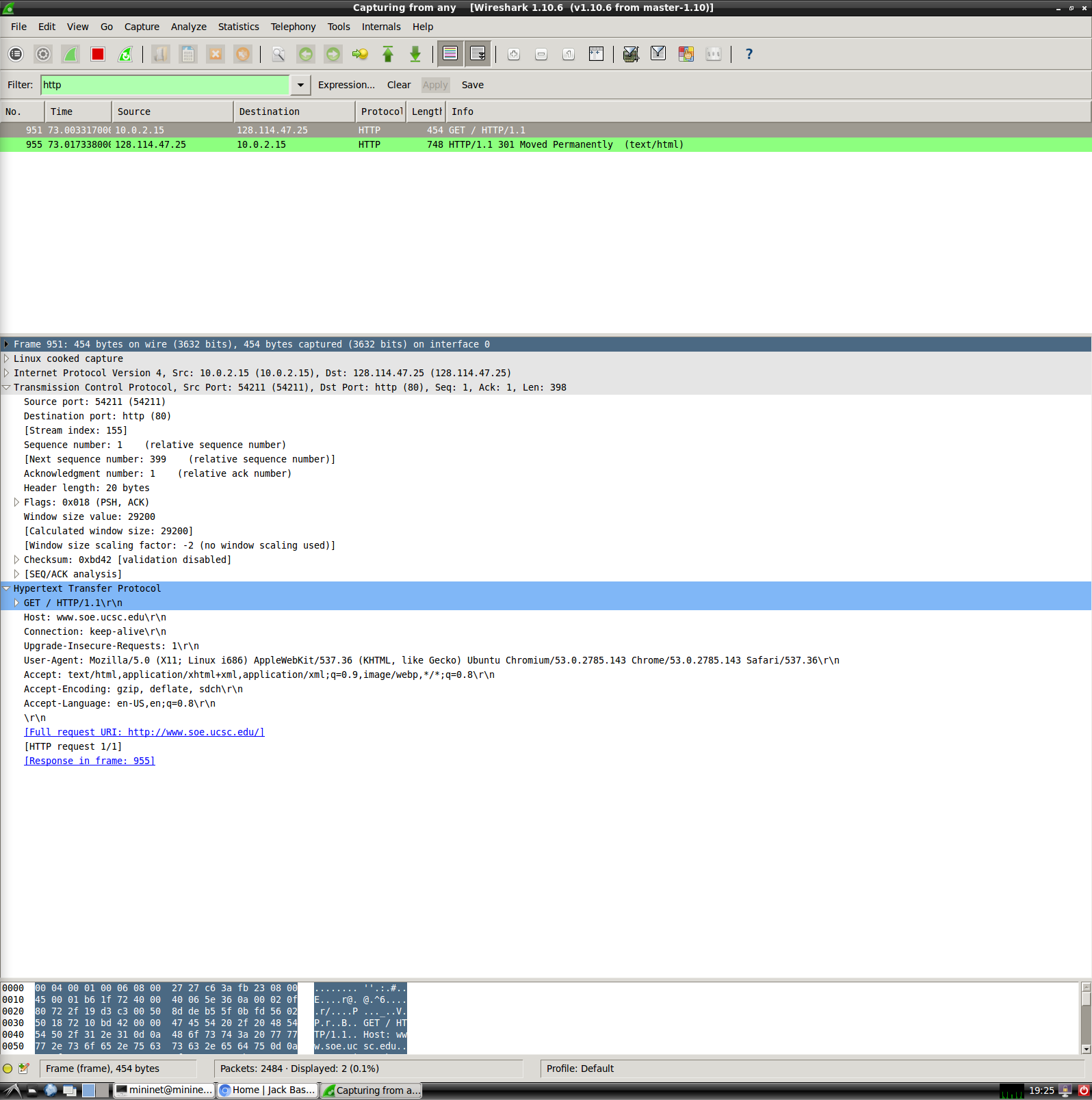
1. The packet that corresponds to the initial HTTP request made by this computer used the method GET to make the request. The URI that my computer requested was <http://www.example.com> which was the same the link typed in. This is because the initial request is made to access the site and since the URL refers to the subset of URIs, it shows it makes a request to the corresponding URI.



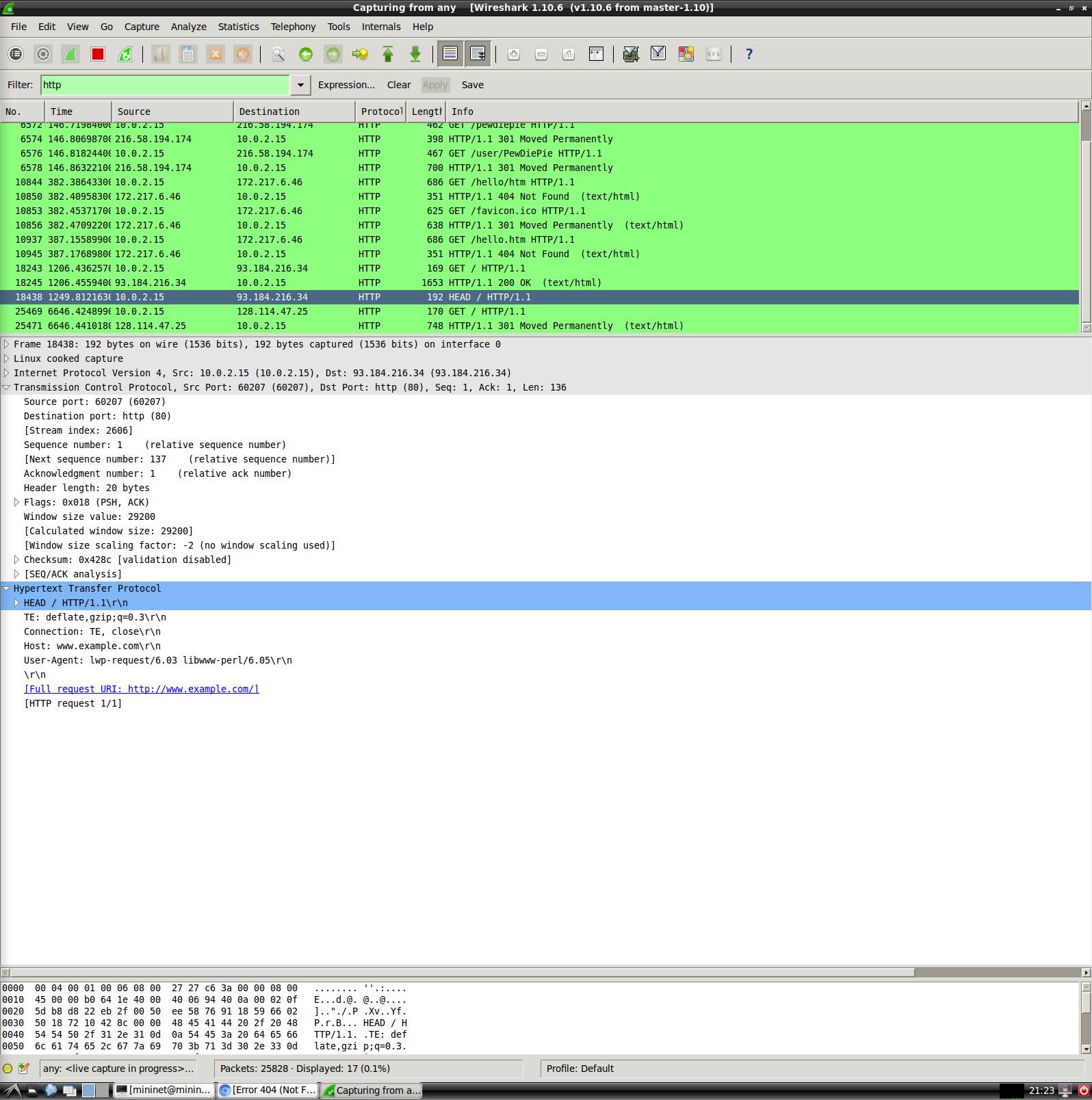
1. You will notice in the picture below that the packet corresponding to the initial HTTP response the server issued in response to my request returned a status code of 200 OK. The content type is labeled as text/html, which makes sense considering that the website is purely text based.

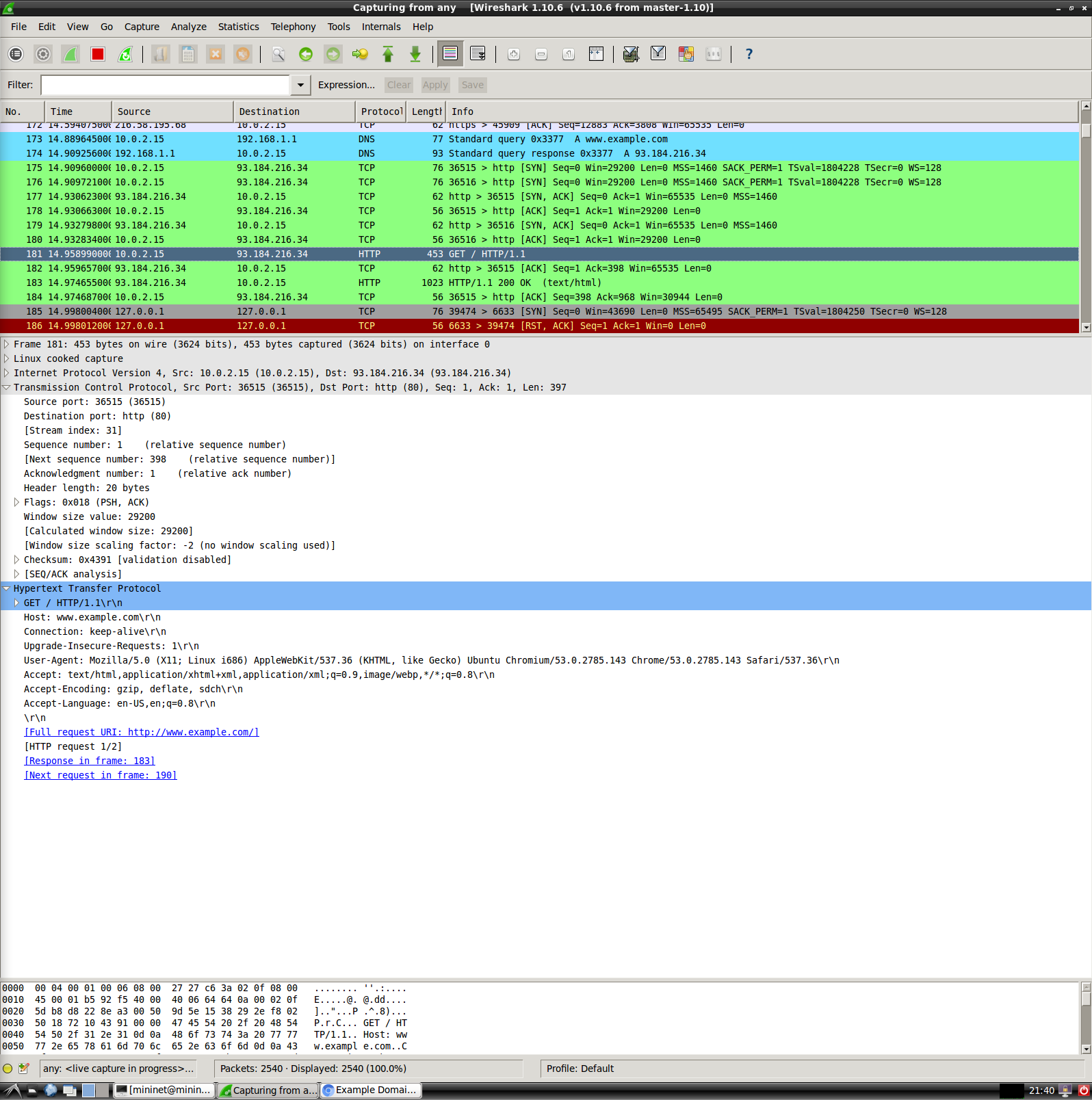
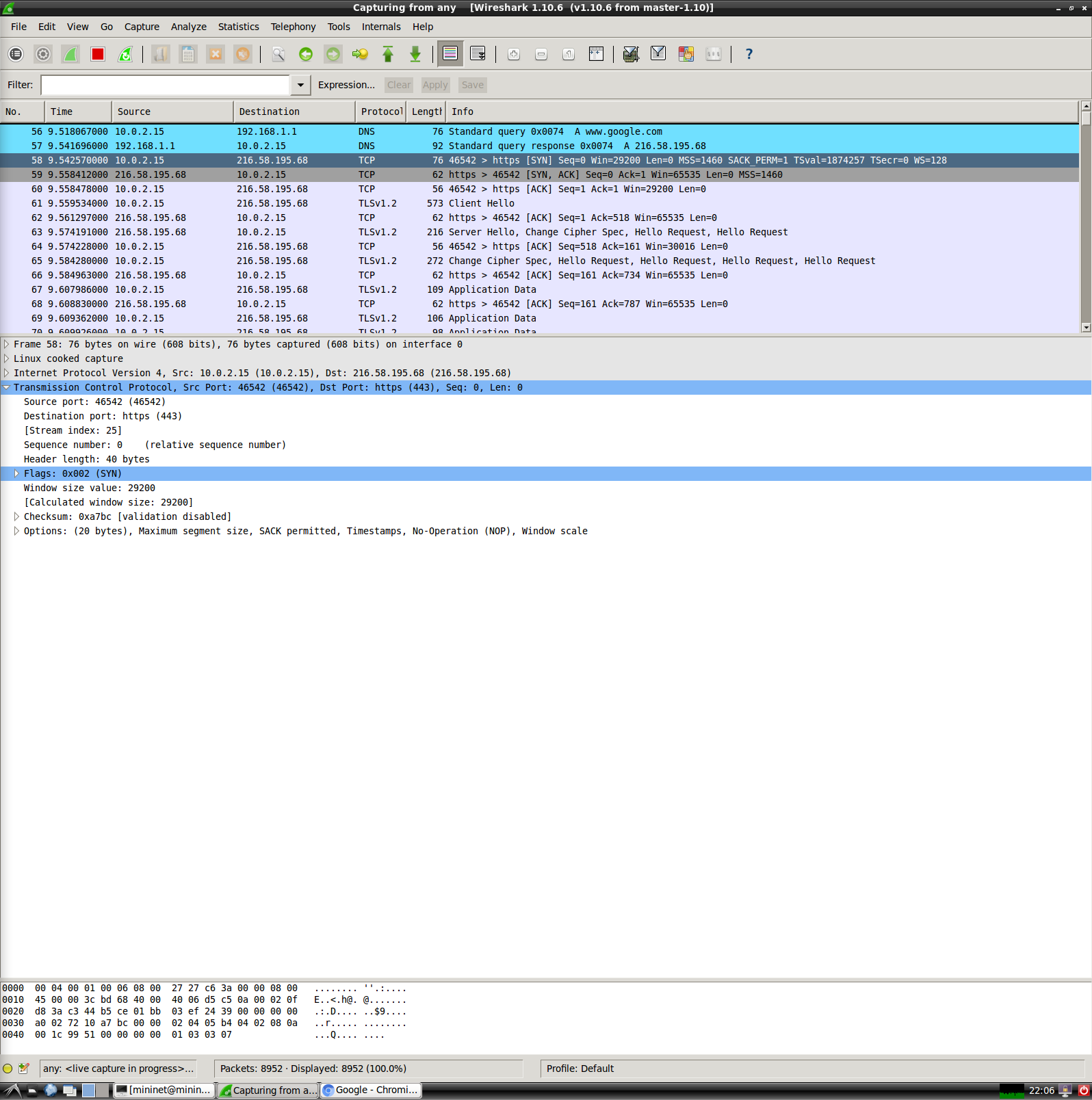
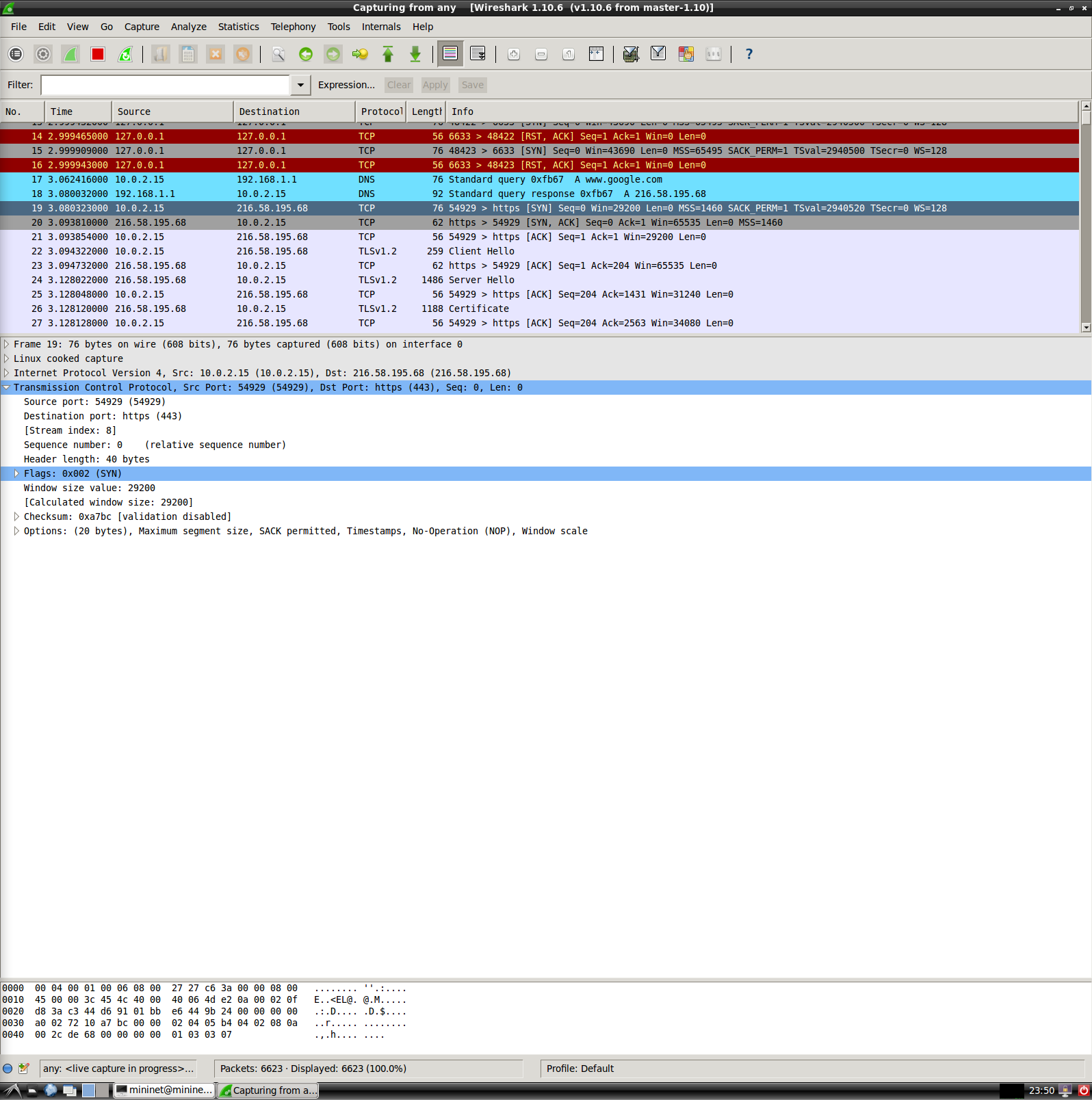


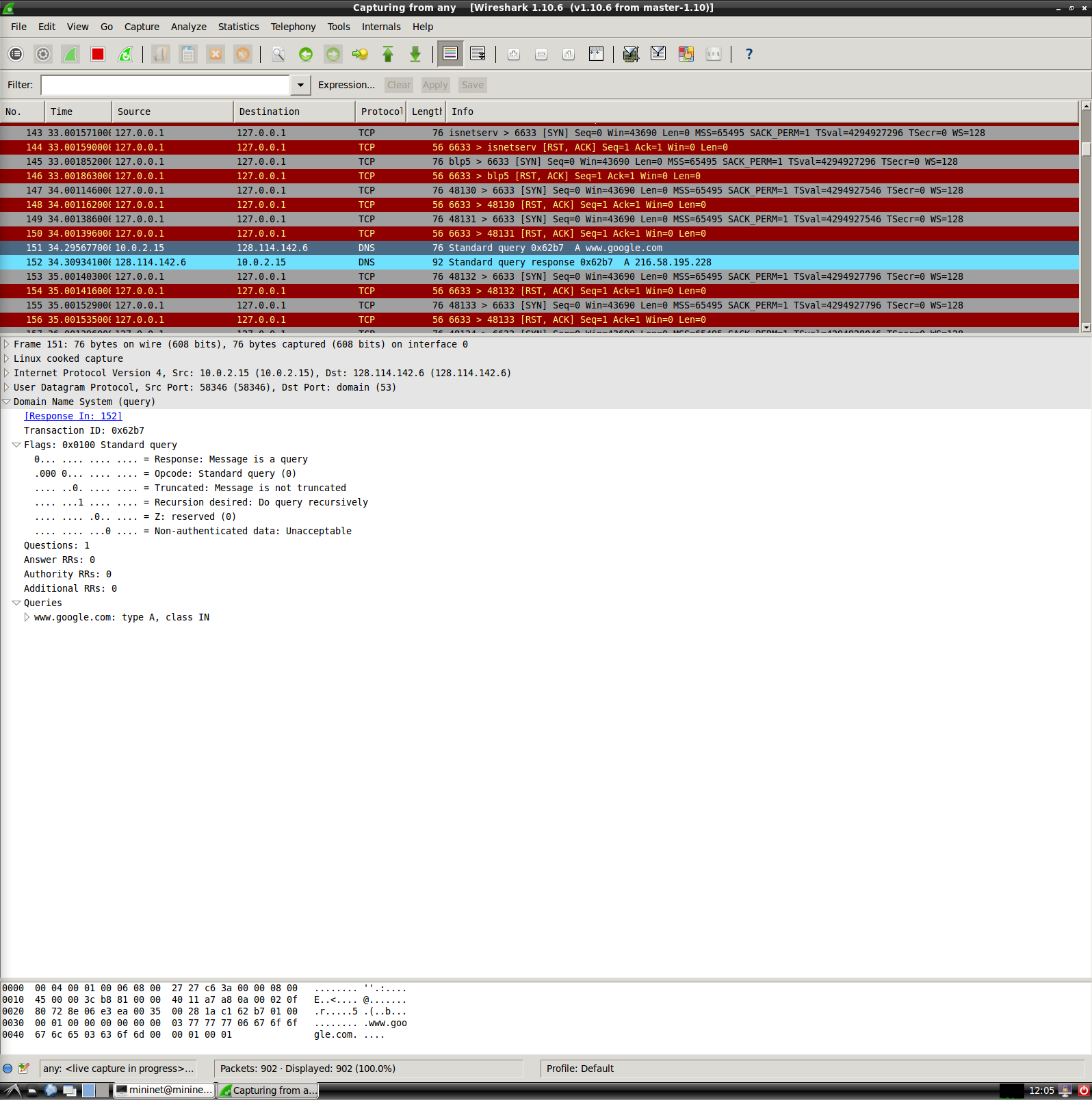
1. The packets corresponding to the initial request made by my computer are identified down below. You will notice that this time the status code returned is different from the one previously, because when you navigate to <http://ww.soe.ucsc.edu> it redirects you to a secure site with https as the protocol. Thus, incurring the 301 status code which means the site was moved permanently.



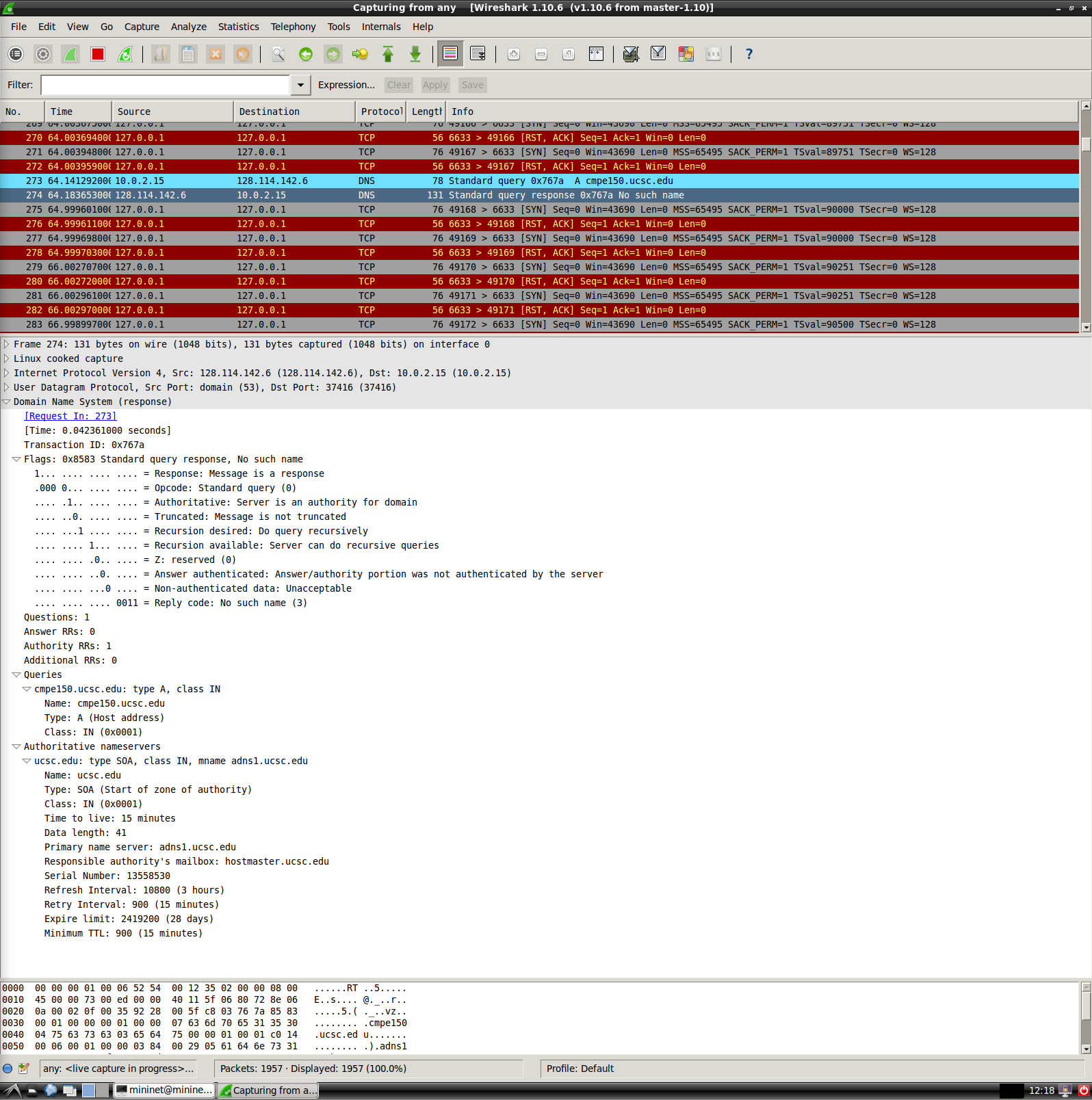
1. The screenshot below shows a packet received through the method HEAD which was used in the Linux terminal to retrieve. This returns the header of the website, which in this case was used for [www.example.com](http://www.example.com).



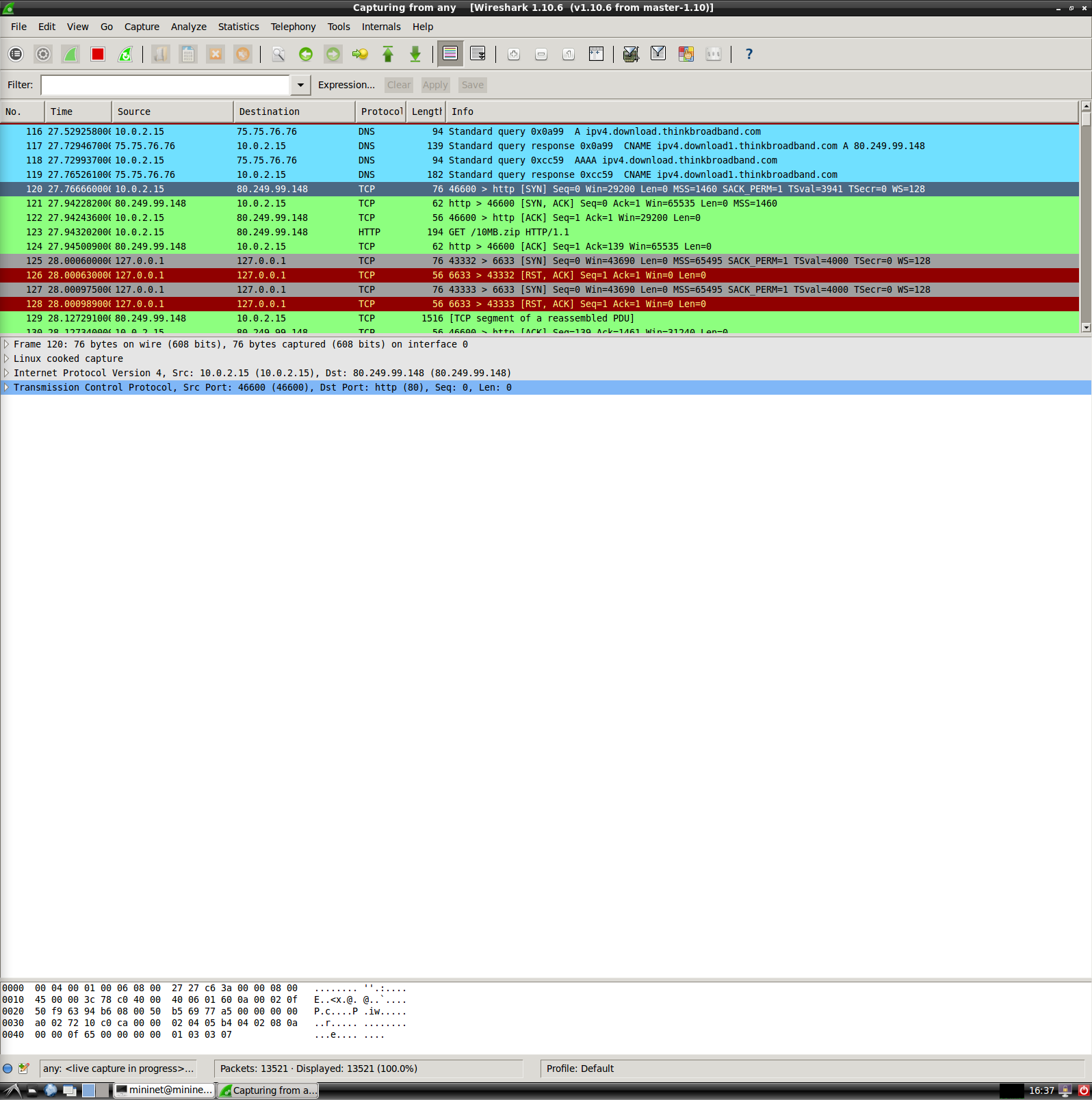
1. You can see the steps taken by my computer before the web page was loaded. First, it established a connection with the server using DNS protocols and then established a 3-way handshake using TCP protocols, where you can see [SYN], [SYN, ACK], and [ACK]. Finally, results in the HTTP protocol which uses the method GET to retrieve the web page. This makes sense because we are given web page site URL, which needs to fetch information from the server before displaying the web page.
2. You can see the steps taken by my computer before the web page was loaded. This time it is different from the previous request because we are given the IP address of the domain, which in this case is Google. You will notice that the first 2 packets refer to the DNS packets trying to authenticate a server connection, which are then followed by TCP protocols to get a request from the server. The first 2 TCP protocols are what is important because this shows the handshake established by the client and the server, which results in an HTTP protocol that retrieves the web page. 
3. The IP address I was given for [www.google.com](http://www.google.com) is 216.58.195.68, which is slightly different from that of the IP address we were given above.
4. My computer did want to complete the request recursively because you can see in the picture below that my computer still uses the DNS protocol to do a query search and if you look at the flags, it says recursion is desired so it does it recursively.



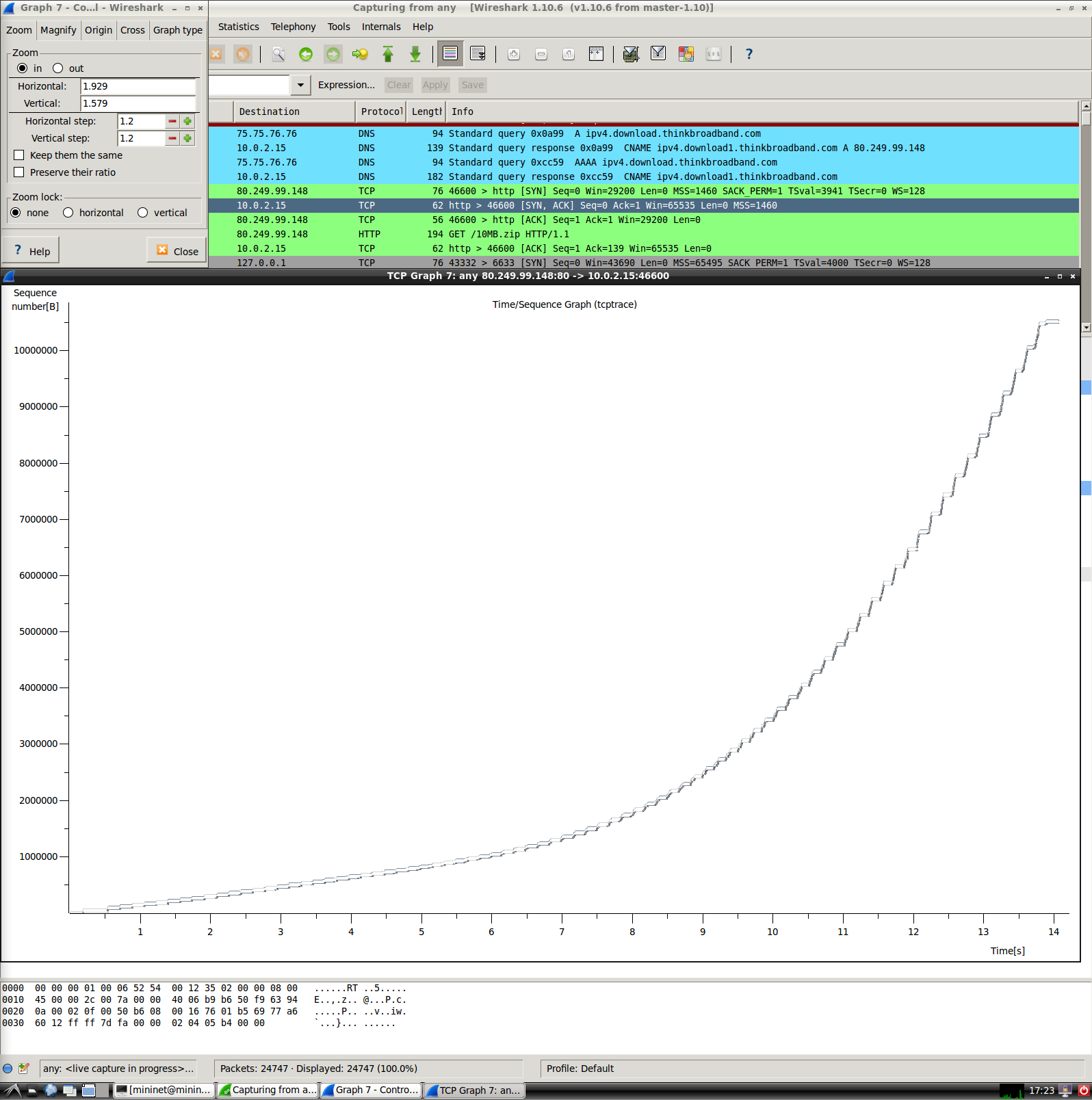
1. There was no IP address given to me for cmpe150.ucsc.edu because it was not authenticated by the server therefore it never resolved by giving an IP address.
2. The authoritative name server for the ucsc.edu is adns1.ucsc.edu if you look at the picture below. You will see it is given in the primary name server.



1. The initial window size that my computer advertised to the server was 29200, whereas the window size that the server advertised was 65535, as shown in the picture below. You will notice that the 3-way handshake was initiated first before the receiving an HTTP code of 200 OK.



1. In the graph depicted below, you can see that this tcptrace graph has the source address as the server’s address and the destination address as my computer’s address. This corresponds to the [SYN, ACK] packet which responds to the computer’s request. Doing a tcptrace on the TCP protocol returns a graph, where it shows the number of packets sent, corresponding to the sequence number, over the total time, which was 14 seconds. You will notice that the sequence number represents the size of the download which was 10Mb, which is the equivalent to 10,000,000 bits. The top line represents the receive window which is the server window, whereas the bottom line are the ACKs. In between those lines is the TCP segment and between the TCP segment and the ACKs are the bytes in flight.



1. The picture below depicts the tcptrace graph of the packets sent and received, where there are sections where the download experiences 0% and 100% loss. 0% loss being circled along with where the slow-start occurs. The parts that flatline correspond to the 100% loss, whereas everything else corresponds to the 0% loss. You can see the slow start close to the beginning of each section when we change packet loss, which starts off slow and increases exponentially.

