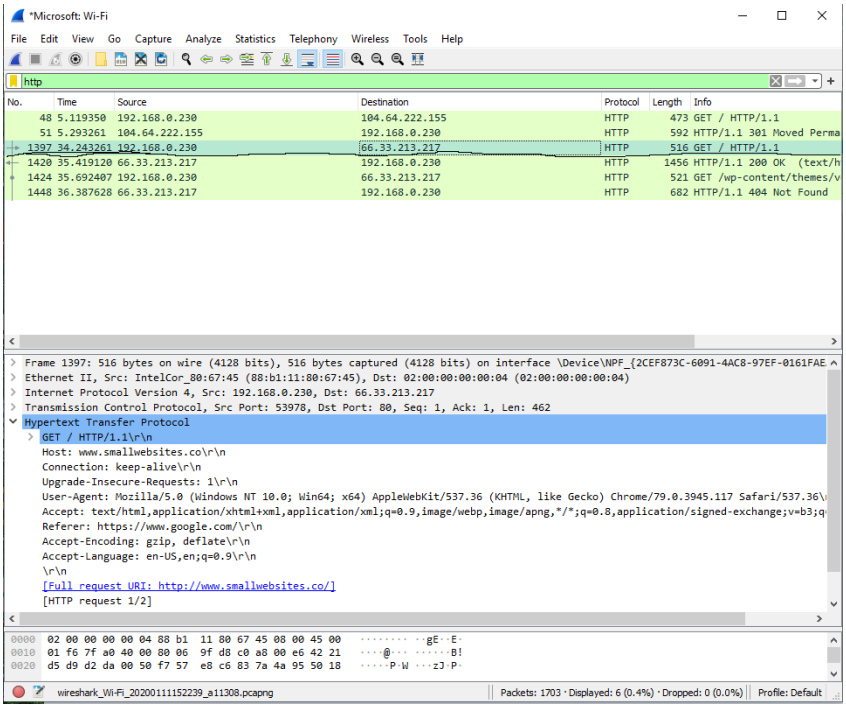
COMPSCI 4C03 Assignment 1

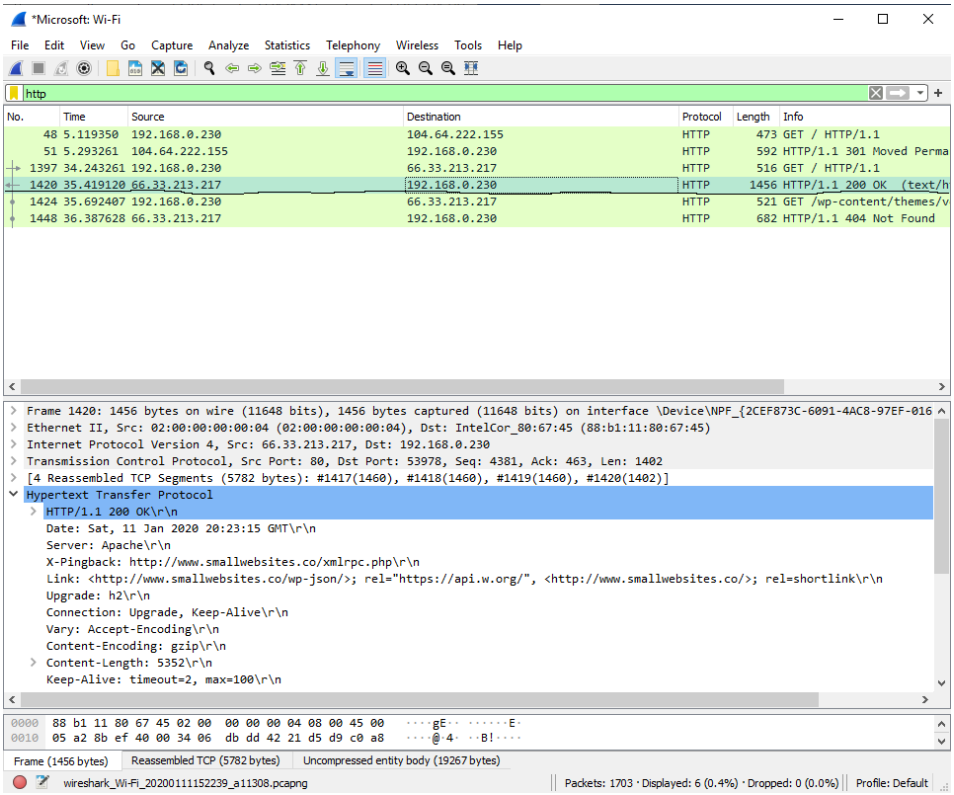
Question 1: Capturing an HTTP Message

1-A: Add the Wireshark screenshot and highlight the two messages in it.

HTTP GET message:



HTTP OK Message:



1-B: Write the complete URL of the webpage referred in GET message. Which fields/lines of GET message can be used to acquire the complete URL?

The complete URL of the webpage is: http://www.smallwebsites.co/ and can be found under the "Hypertext Transfer Protocol section of the message body, under "Full request URI".

1-C: Assume the content of a GET message is provided to you in the form of a string. Write the pseudocode or algorithm to parse this string to extract the complete URL out of this string.

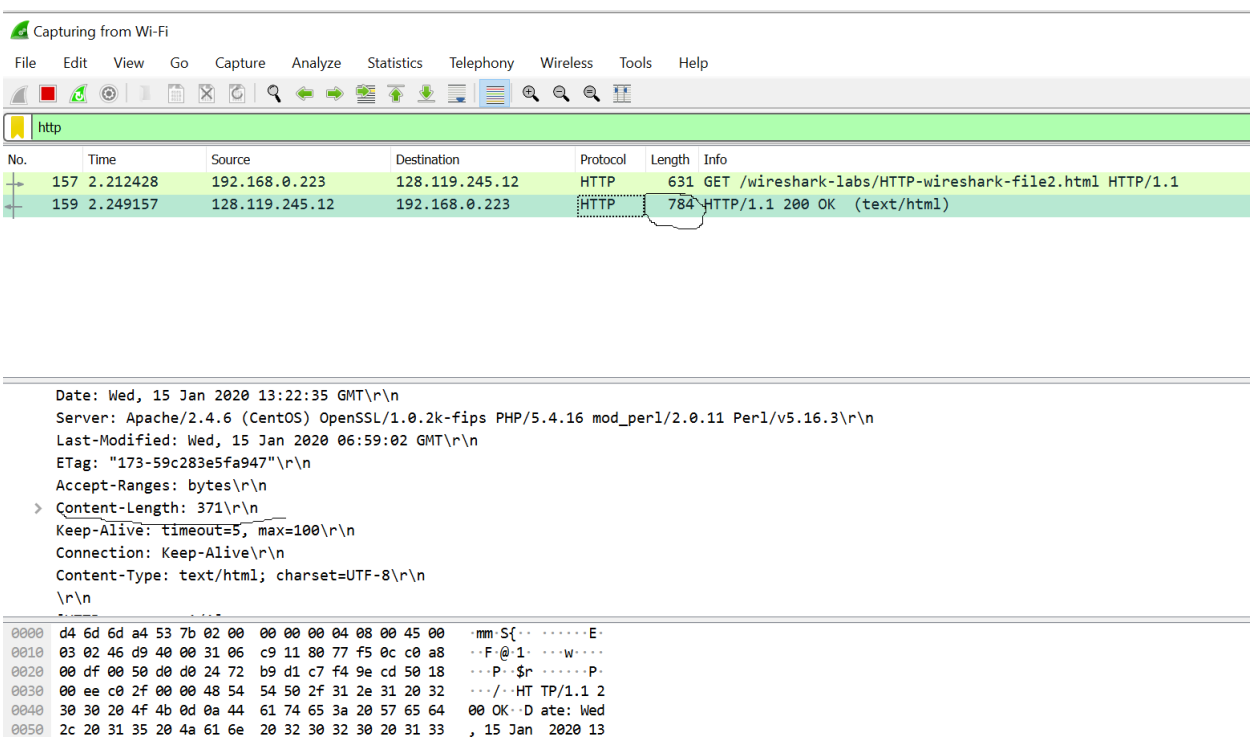
We can easily use regex libraries to parse out the URL, since it is always encapsulated in

[Full request URI: URL GOES HERE] within the message body. The pseudocode using pythons re library is as follows:

```
Import re

text = GET_message_text

# returns [Full request URI: URL GOES HERE]

encased_URL = re.search("\[Full request URI: .*\]", text).group(0)

# will parse out URI from above

URL=encased_URL[19:-1]
```

Question 2: Analyzing HTTP Messages

2-A: For a certain "HTTP OK " message, what does the difference in the values of "Content-Length" and the "Length" column in Wireshark window indicate?

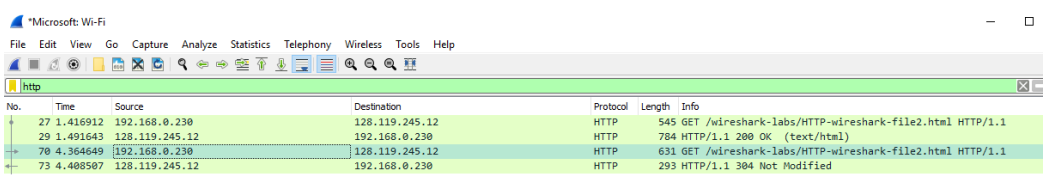


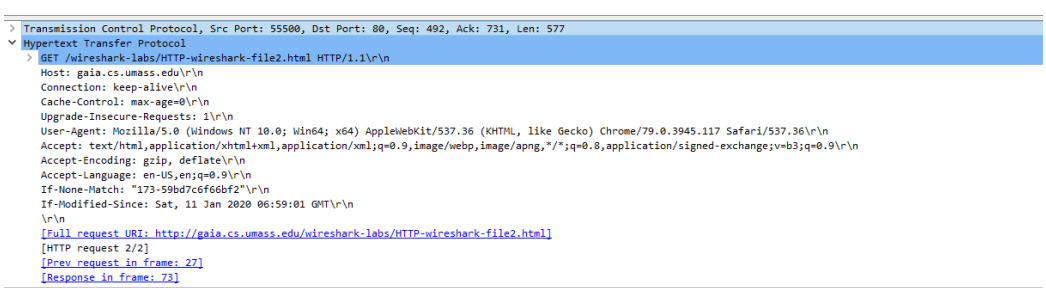
The difference between content-length and length in this http ok message is 784 – 371 = 413. This 413 represents the size of the http header that was part of the get request. One way to determine this is from the label “file data: 371 bytes,” which is found within the Hypertext Transfer Protocol subsection of the message body. This label clearly indicates that the actual body contains 371 bytes, meaning the difference must be the header. Another way to verify is to look at the actual body of the message and look at what individual parts represent.

```
    Keep-Alive: timeout=5, max=100\r\n
    Connection: Keep-Alive\r\n
    Content-Type: text/html; charset=UTF-8\r\n
    \r\n
    [HTTP response 1/1]
    [Time since request: 0.036729000 seconds]
    [Request in frame: 157]
    [Request URI: http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file2.html]
    File Data: 371 bytes
> Line-based text data: text/html (10 lines)
```

2-B: An HTTP GET message was sent with “IF-MODIFIED-SINCE” entry. From the response message, how can we identify if the content is modified since the time mentioned in the GET message?

When a HTTP GET message is sent with “IF-MODIFIED-SINCE,” the date after the header is saved in our browser’s cache. When a following get request is made to the same URL, the browser only downloads a new copy of the HTML file if the date is different than the timestamped date. By looking at the response code sent back in the response message, we can tell if a new copy was downloaded. A response code of 200 (ok) will only be sent if the given source file was modified after the given date, otherwise a 304 response (not-modified) will be sent back

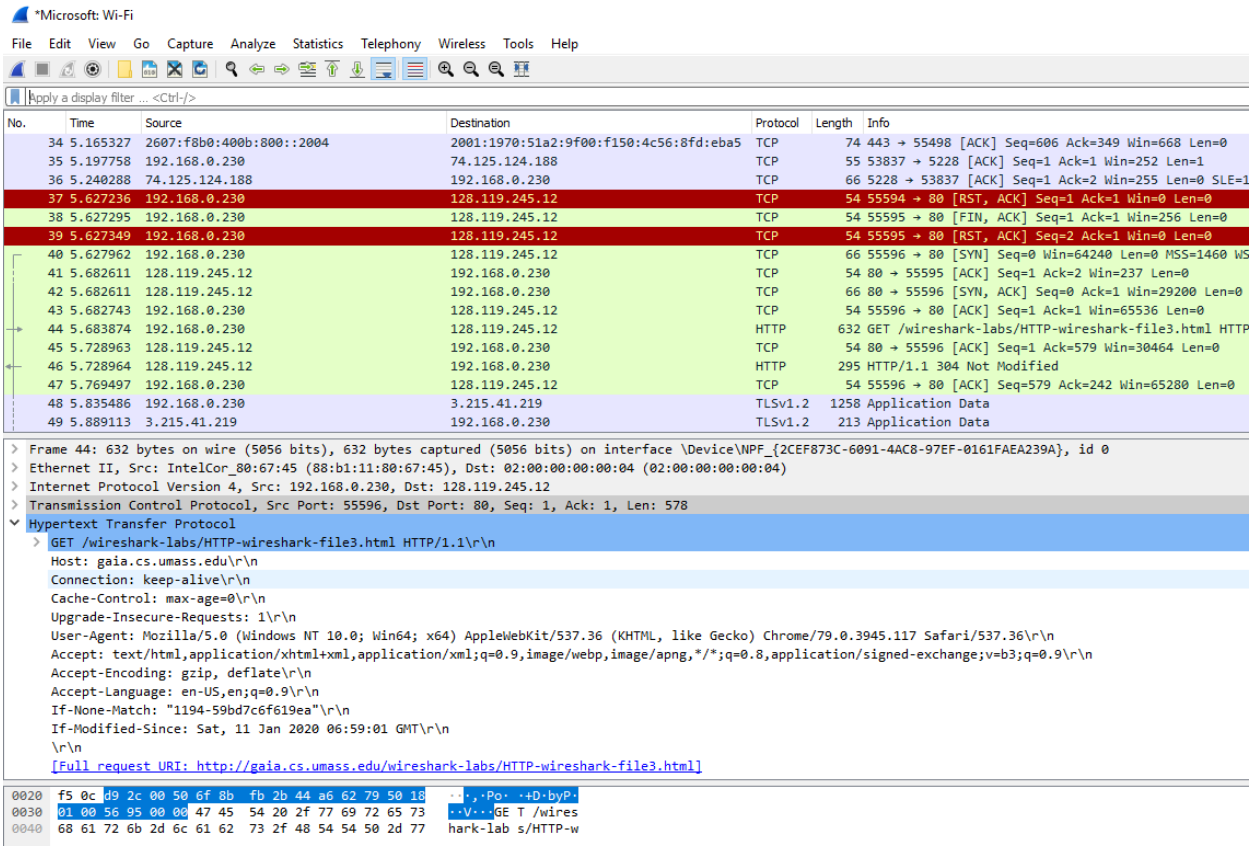




As seen in the above screenshot, I made 2 consecutive GET requests to the textbook website lab2 demonstration URL. The first response was 200, meaning a copy of the file was downloaded in my browser. However, the following response code was 304 (not modified), meaning the file was not modified since the time indicated in the “if-modified-since” header.

2-C: When an HTTP message is contained in multiple TCP segments, does each TCP segment contain the HTTP OK status message?

No, there is only a single HTTP response message, even when the HTTP message is contained in multiple TCP segments.



In the following screenshot, I made a GET request to the sample “large” file download URL in lab2. As you can see the multiple segmented TCPs, there is only a single HTTP response message with a code of 304 (not modified). The reason it’s 304 instead of 200 is because I had already had cached the file during my run-through of the lab.