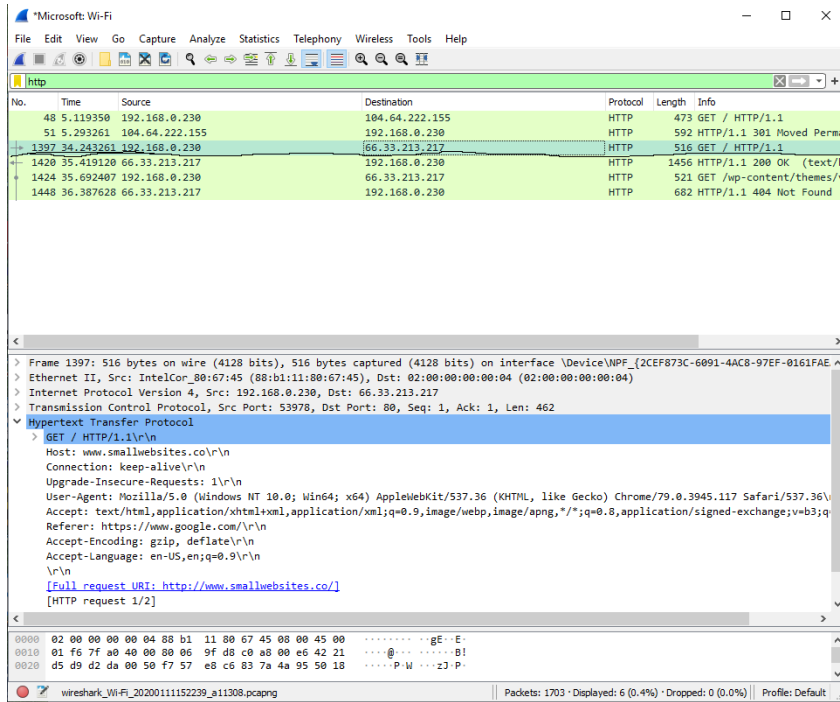


COMPSCI 4C03 Assignment 1

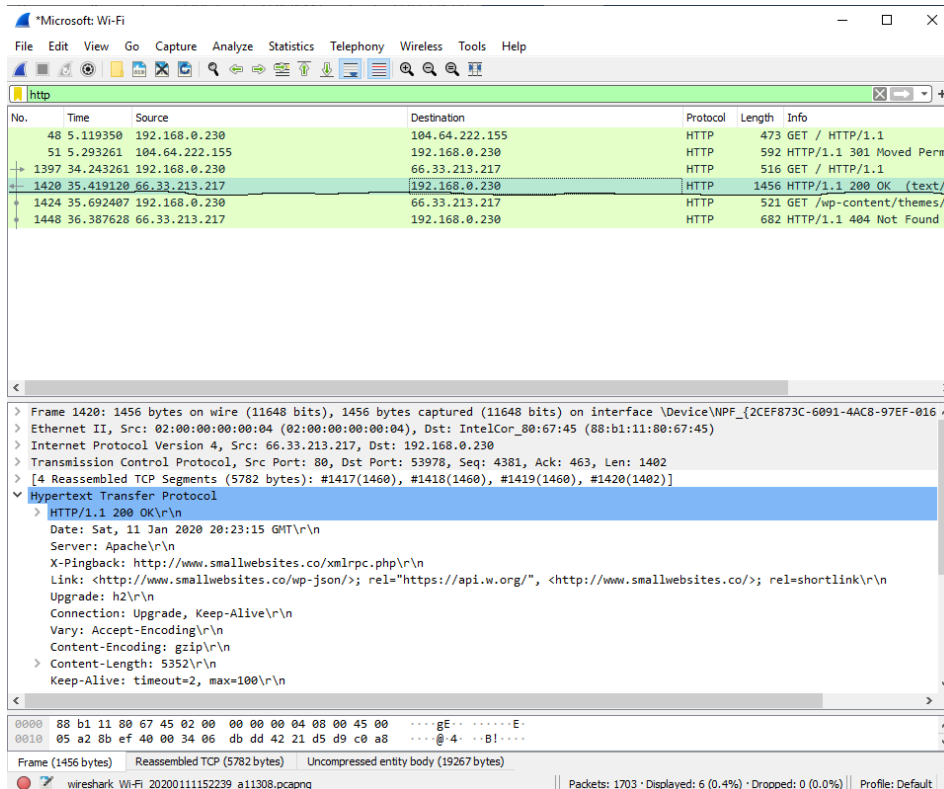
Question 1: Capturing an HTTP Message

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 python's re library is as follows:

```
Import re
```

```
text = GET_message_text
```

```
# returns [Full request URI: URL GOES HERE]
```

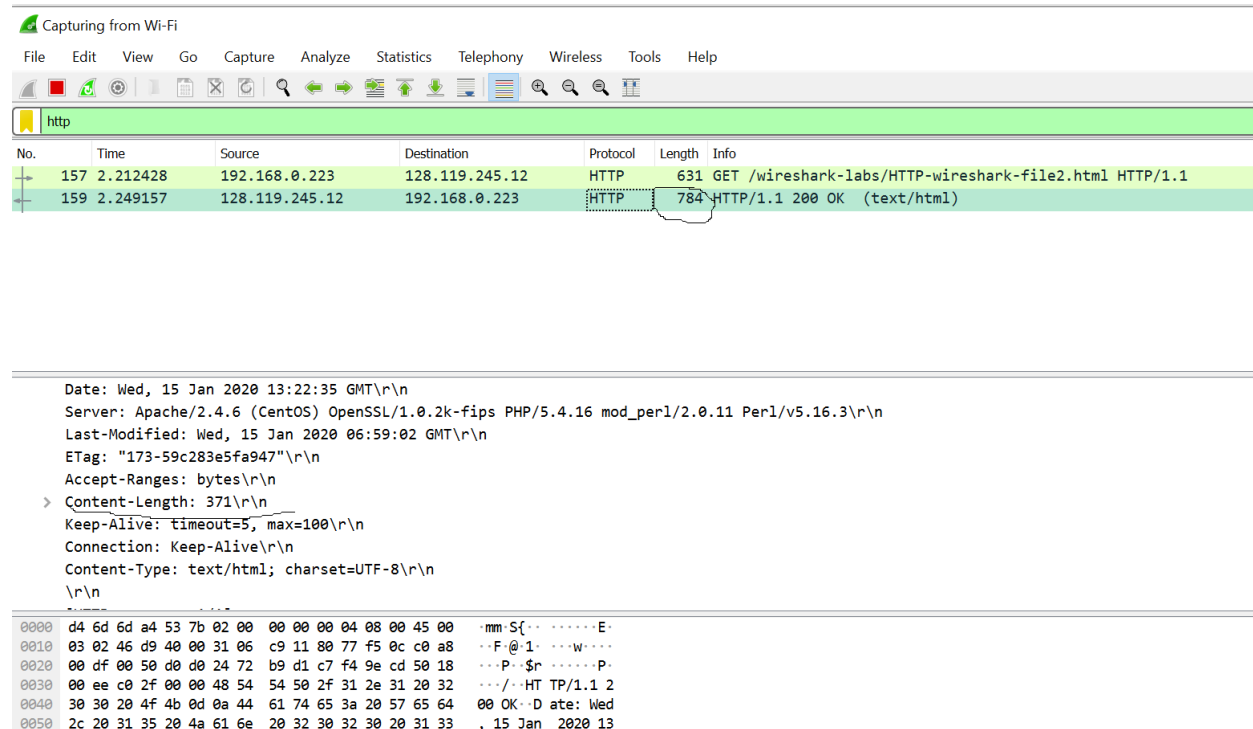
```
encased_URL = re.search("[Full request URI: .*]", text).group(0)
```

```
# 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?



Wireshark packet capture showing an HTTP GET request and response. The packet list shows a GET request (631 bytes) and a 200 OK response (784 bytes). The packet details for the response show a Content-Length of 371. The packet bytes show the raw HTTP data.

No.	Time	Source	Destination	Protocol	Length	Info
157	2.212428	192.168.0.223	128.119.245.12	HTTP	631	GET /wireshark-labs/HTTP-wireshark-file2.html HTTP/1.1
159	2.249157	128.119.245.12	192.168.0.223	HTTP	784	HTTP/1.1 200 OK (text/html)

```

Date: Wed, 15 Jan 2020 13:22:35 GMT\r\n
Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/5.4.16 mod_perl/2.0.11 Perl/v5.16.3\r\n
Last-Modified: Wed, 15 Jan 2020 06:59:02 GMT\r\n
ETag: "173-59c283e5fa947"\r\n
Accept-Ranges: bytes\r\n
> Content-Length: 371\r\n
Keep-Alive: timeout=5, max=100\r\n
Connection: Keep-Alive\r\n
Content-Type: text/html; charset=UTF-8\r\n
\r\n
0000  d4 6d 6d a4 53 7b 02 00 00 00 00 04 08 00 45 00  .mm.S{.. ....E.
0010  03 02 46 d9 40 00 31 06 c9 11 80 77 f5 0c c0 a8  .F.@.1. ...w...
0020  00 df 00 50 d0 d0 24 72 b9 d1 c7 f4 9e cd 50 18  .P.$P. ....P.
0030  00 ee c0 2f 00 00 48 54 54 50 2f 31 2e 31 20 32  .../.HT TP/1.1 2
0040  30 30 20 4f 4b 0d 0a 44 61 74 65 3a 20 57 65 64  00 OK..D ate: Wed
0050  2c 20 31 35 20 4a 61 6e 20 32 30 32 30 20 31 33  , 15 Jan 2020 13

```

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

No.	Time	Source	Destination	Protocol	Length	Info
27	1.416912	192.168.0.230	128.119.245.12	HTTP	545	GET /wireshark-labs/HTTP-wireshark-file2.html HTTP/1.1
29	1.491643	128.119.245.12	192.168.0.230	HTTP	784	HTTP/1.1 200 OK (text/html)
70	4.364649	192.168.0.230	128.119.245.12	HTTP	631	GET /wireshark-labs/HTTP-wireshark-file2.html HTTP/1.1
73	4.408597	128.119.245.12	192.168.0.230	HTTP	293	HTTP/1.1 304 Not Modified

```
> Transmission Control Protocol, Src Port: 55500, Dst Port: 80, Seq: 492, Ack: 731, Len: 577
  Hypertext Transfer Protocol
    GET /wireshark-labs/HTTP-wireshark-file2.html HTTP/1.1\r\n
    Host: gaia.cs.umass.edu\r\n
    Connection: keep-alive\r\n
    Cache-Control: max-age=0\r\n
    Upgrade-Insecure-Requests: 1\r\n
    User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/79.0.3945.117 Safari/537.36\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9\r\n
    Accept-Encoding: gzip, deflate\r\n
    Accept-Language: en-US,en;q=0.9\r\n
    If-None-Match: "173-59bd7c6f6ebf2"\r\n
    If-Modified-Since: Sat, 11 Jan 2020 06:59:01 GMT\r\n
    \r\n
    [Full request URI: http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file2.html]
    [HTTP request 2/2]
    [Prev request in frame: 27]
    [Response in frame: 73]
```

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.

Microsoft Wi-Fi

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-F>

No.	Time	Source	Destination	Protocol	Length	Info
34	5.165327	2607:f8b0:400b:800::2004	2001:1970:51a2:9f00:f150:4c56:8fd:eba5	TCP	74	443 → 55498 [ACK] Seq=606 Ack=349 Win=668 Len=0
35	5.197758	192.168.0.230	74.125.124.188	TCP	55	53837 → 5228 [ACK] Seq=1 Ack=1 Win=252 Len=1
36	5.240288	74.125.124.188	192.168.0.230	TCP	66	5228 → 53837 [ACK] Seq=1 Ack=2 Win=255 Len=0 SLE=1
37	5.627236	192.168.0.230	128.119.245.12	TCP	54	55594 → 80 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
38	5.627295	192.168.0.230	128.119.245.12	TCP	54	55595 → 80 [FIN, ACK] Seq=1 Ack=1 Win=256 Len=0
39	5.627349	192.168.0.230	128.119.245.12	TCP	54	55595 → 80 [RST, ACK] Seq=2 Ack=1 Win=0 Len=0
40	5.627962	192.168.0.230	128.119.245.12	TCP	66	55596 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS
41	5.682611	128.119.245.12	192.168.0.230	TCP	54	80 → 55595 [ACK] Seq=1 Ack=2 Win=237 Len=0
42	5.682611	128.119.245.12	192.168.0.230	TCP	66	80 → 55596 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0
43	5.682743	192.168.0.230	128.119.245.12	TCP	54	55596 → 80 [ACK] Seq=1 Ack=1 Win=65536 Len=0
44	5.683874	192.168.0.230	128.119.245.12	HTTP	632	GET /wireshark-labs/HTTP-wireshark-file3.html HTTP
45	5.728963	128.119.245.12	192.168.0.230	TCP	54	80 → 55596 [ACK] Seq=1 Ack=579 Win=30464 Len=0
46	5.728964	128.119.245.12	192.168.0.230	HTTP	295	HTTP/1.1 304 Not Modified
47	5.769497	192.168.0.230	128.119.245.12	TCP	54	55596 → 80 [ACK] Seq=579 Ack=242 Win=65280 Len=0
48	5.835486	192.168.0.230	3.215.41.219	TLv1.2	1258	Application Data
49	5.889113	3.215.41.219	192.168.0.230	TLv1.2	213	Application Data

> Frame 44: 632 bytes on wire (5056 bits), 632 bytes captured (5056 bits) on interface \Device\NPF_{2CEF873C-6091-4AC8-97EF-0161FAEA239A}, id 0

> Ethernet II, Src: IntelCor_80:67:45 (88:b1:11:80:67:45), Dst: 02:00:00:00:00:04 (02:00:00:00:00:04)

> Internet Protocol Version 4, Src: 192.168.0.230, Dst: 128.119.245.12

> Transmission Control Protocol, Src Port: 55596, Dst Port: 80, Seq: 1, Ack: 1, Len: 578

> Hypertext Transfer Protocol

> GET /wireshark-labs/HTTP-wireshark-file3.html HTTP/1.1\r\n

Host: gaia.cs.umass.edu\r\n

Connection: keep-alive\r\n

Cache-Control: max-age=0\r\n

Upgrade-Insecure-Requests: 1\r\n

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/79.0.3945.117 Safari/537.36\r\n

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9\r\n

Accept-Encoding: gzip, deflate\r\n

Accept-Language: en-US,en;q=0.9\r\n

If-None-Match: "1194-59bd7c6f619ea"\r\n

If-Modified-Since: Sat, 11 Jan 2020 06:59:01 GMT\r\n

\r\n

[Full request URI: http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file3.html]

0020 f5 0c d9 2c 00 50 6f 8b fb 2b 44 a6 62 79 50 10 .,Po..+D-byP.

0030 01 00 56 95 00 00 47 45 54 20 2f 77 69 72 65 73 ..V...GE T /wires

0040 68 61 72 6b 2d 6c 61 62 73 2f 48 54 54 50 2d 77 hark-lab s/HTTP-w

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.