

Queens College

Internet and Web Technology (CSCI 355)

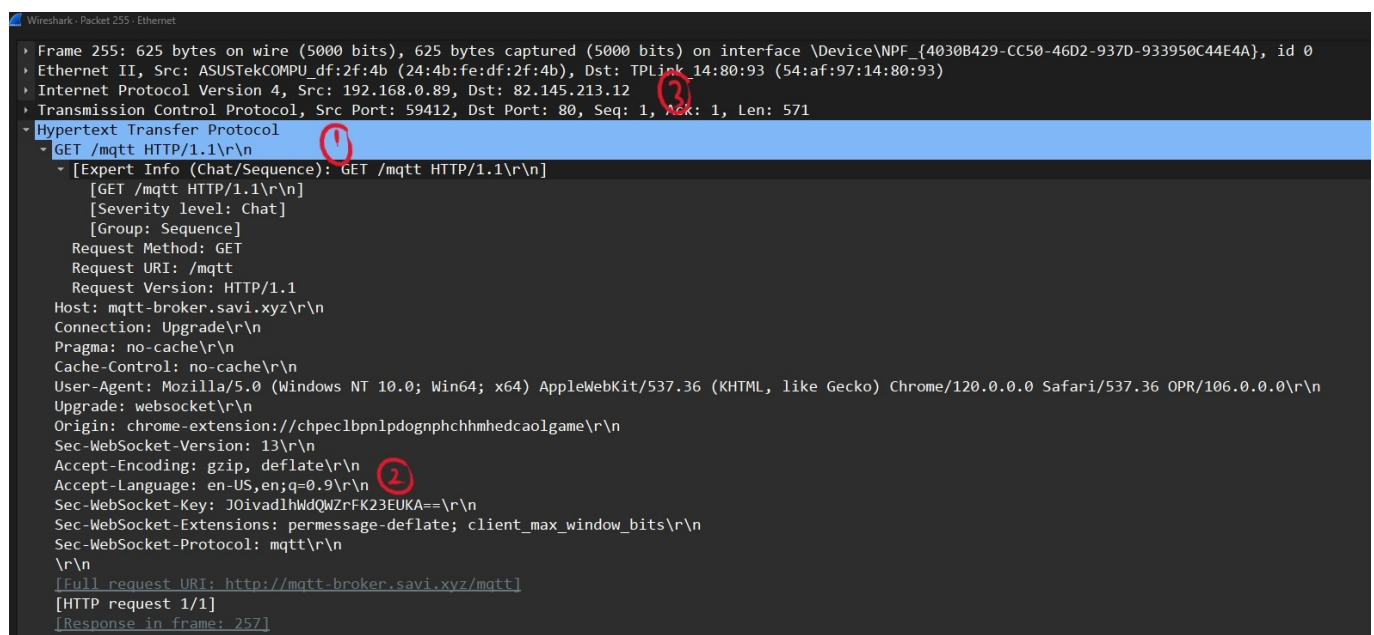
Winter 2024

Assignment 02 - Wireshark

Essmer Sanchez

Collaboration: None

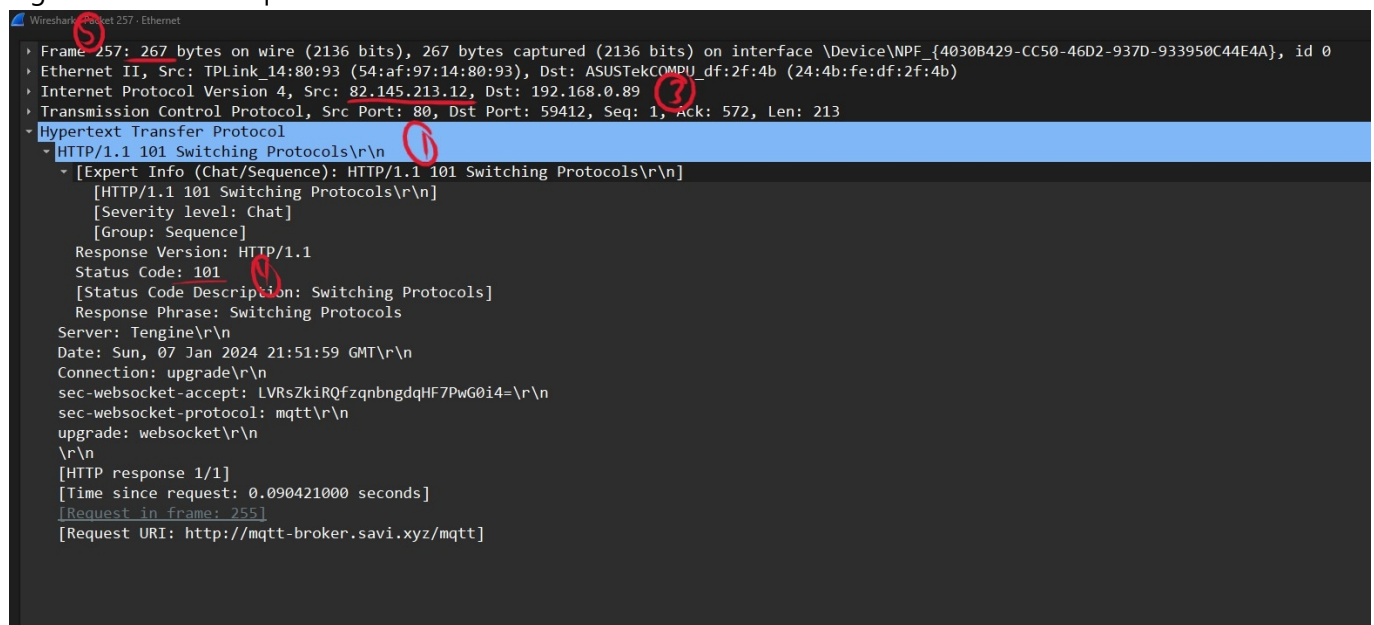
1. The Basic HTTP GET/response interaction.



```

Wireshark - Packet 255 - Ethernet
  Frame 255: 625 bytes on wire (5000 bits), 625 bytes captured (5000 bits) on interface \Device\NPF_{4030B429-CC50-46D2-937D-933950C44E4A}, id 0
  Ethernet II, Src: ASUSTekCOMPU_df:2f:4b (24:4b:fe:df:2f:4b), Dst: TPLink_14:80:93 (54:af:97:14:80:93)
  Internet Protocol Version 4, Src: 192.168.0.89, Dst: 82.145.213.12
  Transmission Control Protocol, Src Port: 59412, Dst Port: 80, Seq: 1, Ack: 1, Len: 571
  Hypertext Transfer Protocol
    GET /mqtt HTTP/1.1\r\n
    [Expert Info (Chat/Sequence): GET /mqtt HTTP/1.1\r\n]
    [GET /mqtt HTTP/1.1\r\n]
    [Severity level: Chat]
    [Group: Sequence]
    Request Method: GET
    Request URI: /mqtt
    Request Version: HTTP/1.1
    Host: mqtt-broker.savi.xyz\r\n
    Connection: Upgrade\r\n
    Pragma: no-cache\r\n
    Cache-Control: no-cache\r\n
    User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.0.0 Safari/537.36 OPR/106.0.0.0\r\n
    Upgrade: websocket\r\n
    Origin: chrome-extension://chpeclbnpdpdognphchhmedcaolgame\r\n
    Sec-WebSocket-Version: 13\r\n
    Accept-Encoding: gzip, deflate\r\n
    Accept-Language: en-US,en;q=0.9\r\n
    Sec-WebSocket-Key: JOivadhWdQWzrFK23EUKA==\r\n
    Sec-WebSocket-Extensions: permmessage-deflate; client_max_window_bits\r\n
    Sec-WebSocket-Protocol: mqtt\r\n
    \r\n
    [Full request URI: http://mqtt-broker.savi.xyz/mqtt]
    [HTTP request 1/1]
    [Response in frame: 257]
  
```

Figure 1.1: Client Request



```

Wireshark - Packet 257 - Ethernet
  Frame 257: 267 bytes on wire (2136 bits), 267 bytes captured (2136 bits) on interface \Device\NPF_{4030B429-CC50-46D2-937D-933950C44E4A}, id 0
  Ethernet II, Src: TPLink_14:80:93 (54:af:97:14:80:93), Dst: ASUSTekCOMPU_df:2f:4b (24:4b:fe:df:2f:4b)
  Internet Protocol Version 4, Src: 82.145.213.12, Dst: 192.168.0.89
  Transmission Control Protocol, Src Port: 80, Dst Port: 59412, Seq: 1, Ack: 572, Len: 213
  Hypertext Transfer Protocol
    HTTP/1.1 101 Switching Protocols\r\n
    [Expert Info (Chat/Sequence): HTTP/1.1 101 Switching Protocols\r\n]
    [HTTP/1.1 101 Switching Protocols\r\n]
    [Severity level: Chat]
    [Group: Sequence]
    Response Version: HTTP/1.1
    Status Code: 101
    [Status Code Description: Switching Protocols]
    Response Phrase: Switching Protocols
    Server: Tengine\r\n
    Date: Sun, 07 Jan 2024 21:51:59 GMT\r\n
    Connection: upgrade\r\n
    sec-websocket-accept: LVRsZkiRQfzqnbngdqHF7PwG0i4=\r\n
    sec-websocket-protocol: mqtt\r\n
    upgrade: websocket\r\n
    \r\n
    [HTTP response 1/1]
    [Time since request: 0.090421000 seconds]
    [Request in frame: 255]
    [Request URI: http://mqtt-broker.savi.xyz/mqtt]
  
```

Figure 1.2: Server Response

1. Is your browser running HTTP version 1.0 or 1.1? What version of HTTP is the server running?

- Figure 1.1 #1: My browser is running HTTP version 1.1.

Request Version: HTTP/1.1

- Figure 1.2 #1: The server is running HTTP version 1.1.

Response Version: HTTP/1.1

2. What languages (if any) does your browser indicate that it can accept to the server?

- Figure 1.1 #2: My browser indicates that it can accept American English (en-US) and general English (en). With a preference of en-US over en.

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

3. What is the IP address of your computer? Of the gaia.cs.umass.edu server?

- Figure 1.1 #3: The IP address of my computer is:

192.168.0.89

- Figure 1.2 #3: The IP address of the server is:

82.145.213.12

4. What is the status code returned from the server to your browser?

- Figure 1.2 #4: The status code returned is:

Status Code: 101

5. When was the HTML file that you are retrieving last modified at the server?

- Wireshark did **NOT** provide me with a "Last Modified" value.

6. How many bytes of content are being returned to your browser?

- The number of bytes of content is **NOT** displayed.
- Figure 1.2 #5: A total length of the package is displayed: 267 bytes.

7. By inspecting the raw data in the packet content window, do you see any headers within the data that are not displayed in the packet-listing window? If so, name one.

- There are **NO** distinguishing headers that can be determined to not be displayed in the packet-listing window.

2. The HTTP CONDITIONAL GET/response interaction.

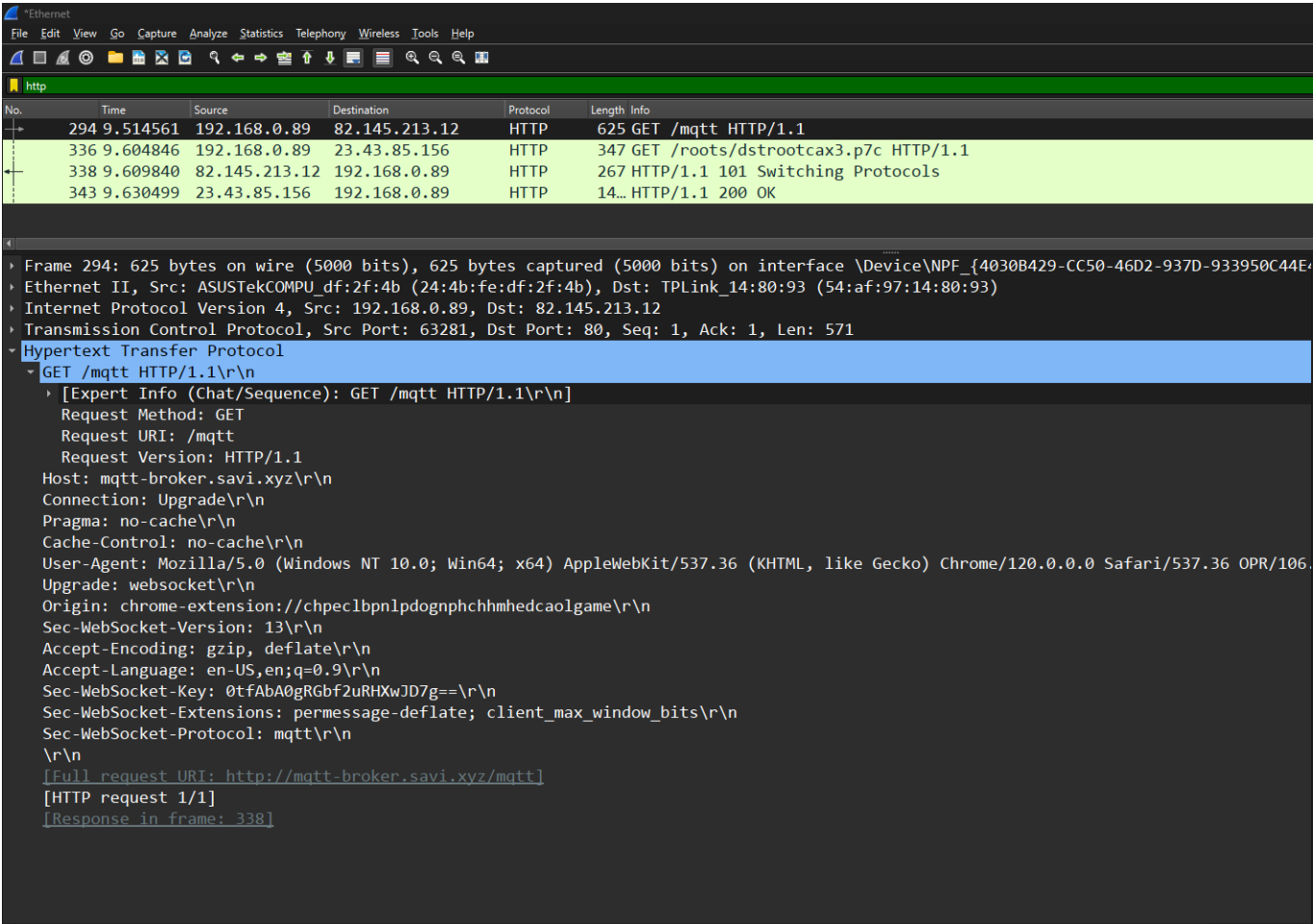


Figure 2.1: Client GET Request 1.

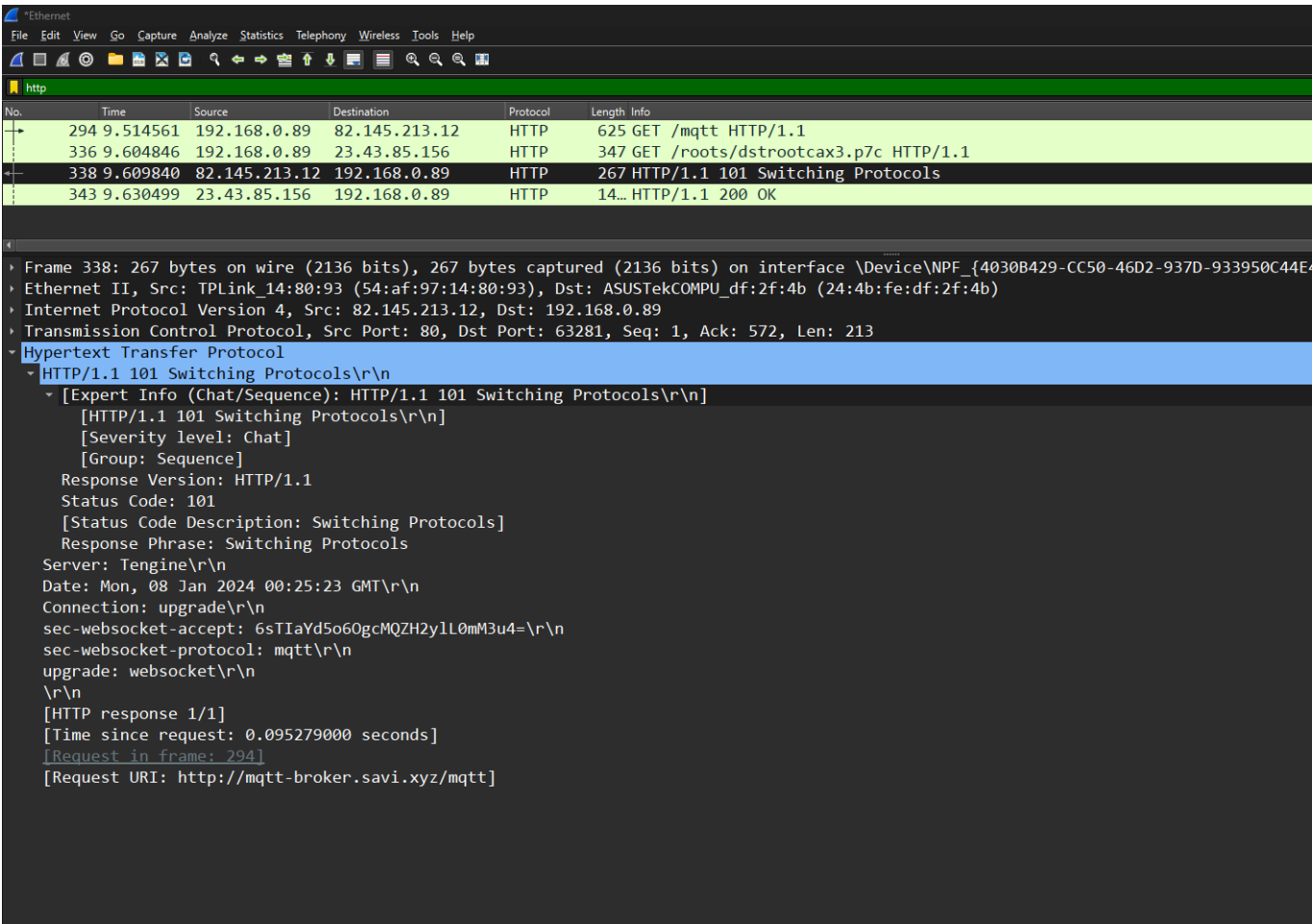


Figure 2.2: Server Resonse 1.

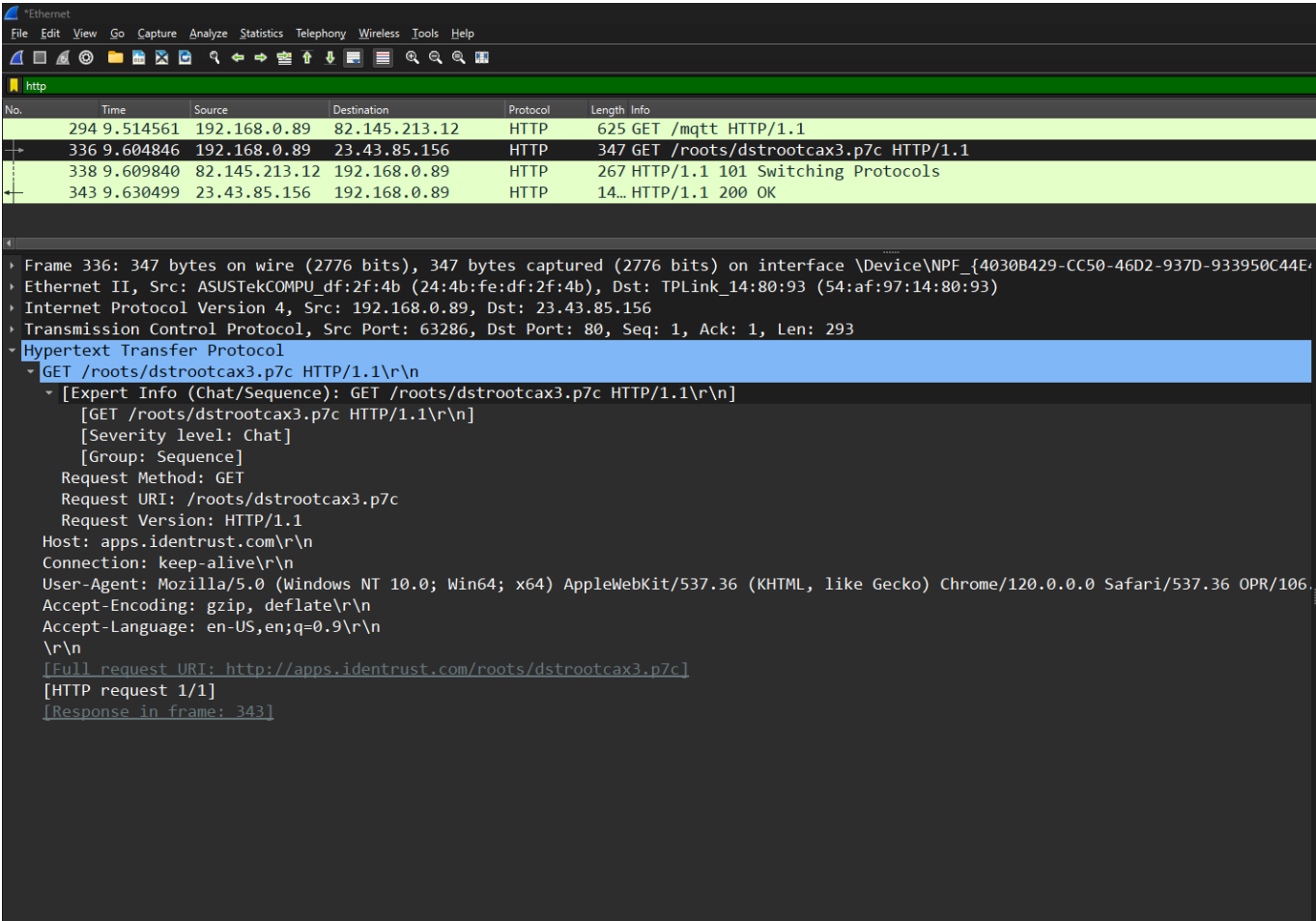


Figure 2.3: Client GET Request 2.

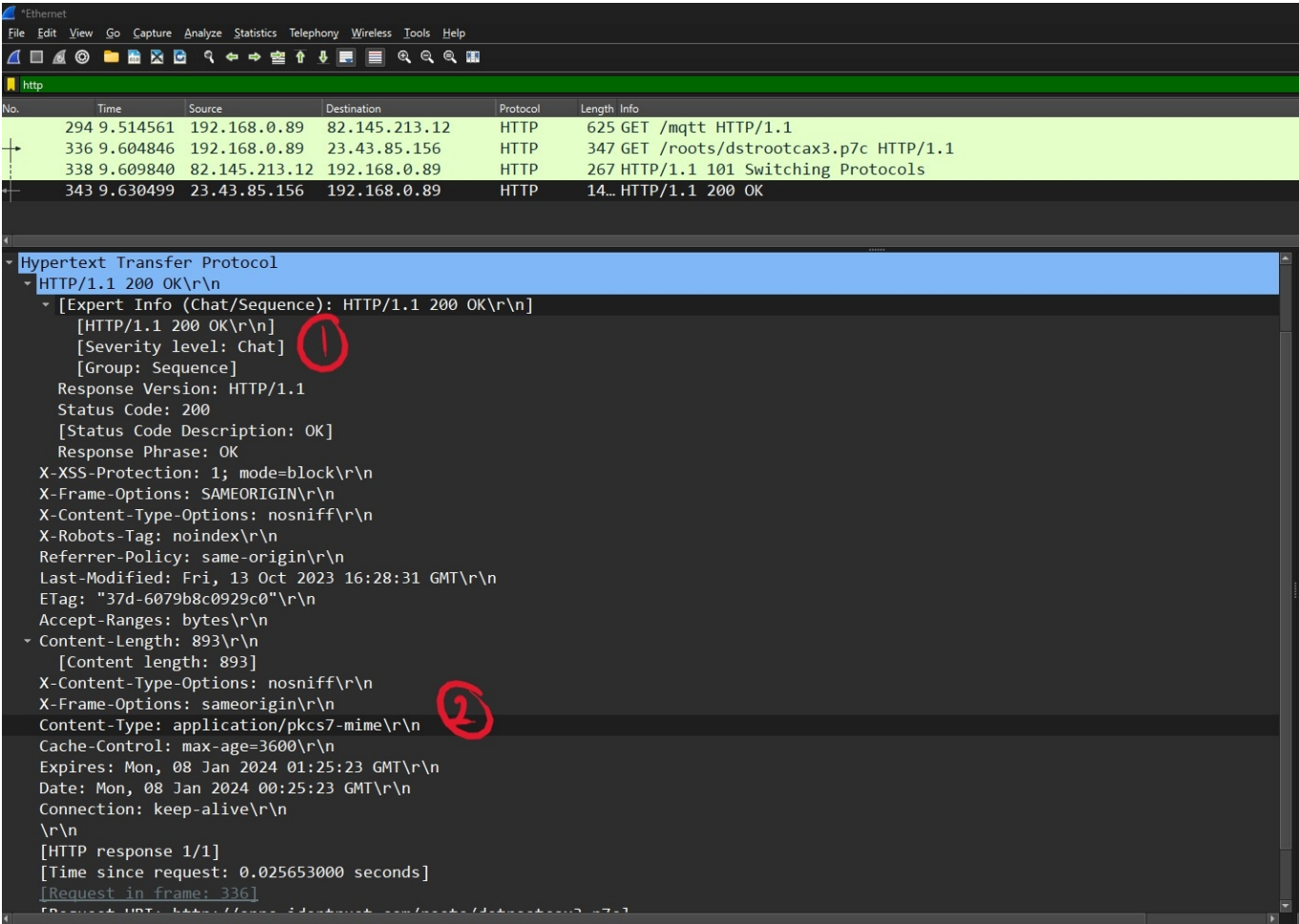


Figure 2.4: Server Resonse 2.

8. Inspect the contents of the first HTTP GET request from your browser to the server. Do you see an "IF-MODIFIED-SINCE" line in the HTTP GET?
 - Figure 2.1: There is no **"IF-MODIFIED-SINCE"** line in the HTTP GET.
9. Inspect the contents of the server response. Did the server explicitly return the contents of the file? How can you tell?
 - Figure 2.2: I **can not** tell if the server explicitly returned the contents.
10. Now inspect the contents of the second HTTP GET request from your browser to the server. Do you see an "IF-MODIFIED-SINCE:" line in the HTTP GET? If so, what information follows the "IF-MODIFIED-SINCE:" header?
 - Figure 2.3: The second HTTP GET request **does not** display an **"IF-MODIFIED-SINCE:"** line.
11. What is the HTTP status code and phrase returned from the server in response to this second HTTP GET? Did the server explicitly return the contents of the file? Explain.
 - Figure 2.4 #1: Status code: **200**.
 - Figure 2.4 #1: Phrase Returned: **OK**.
 - Figure 2.4 #2: It seems that the server did not explicitly return the contents of the file.

Content Type: `application/pkcs7-mime\r\n` This indicates that the returning data is encrypted.

3. Retrieving Long Documents.

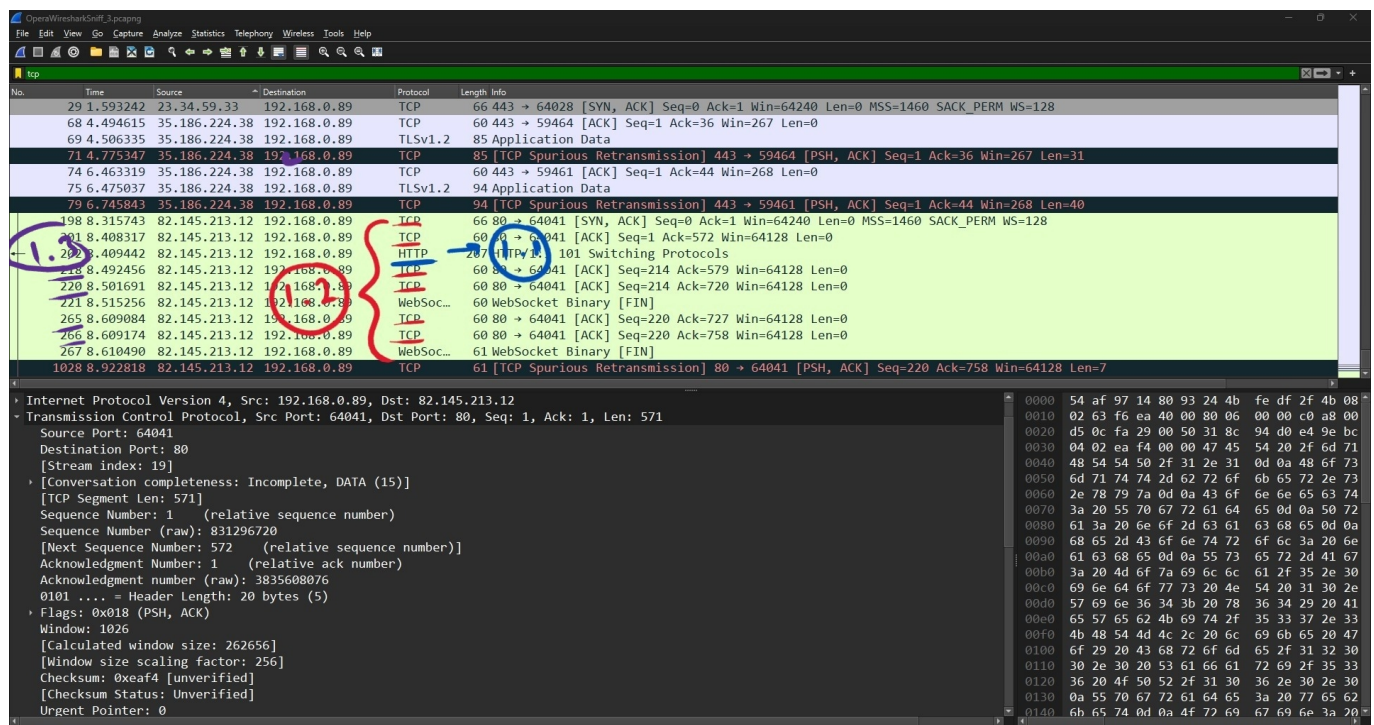


Figure 3.1: Client and Server.

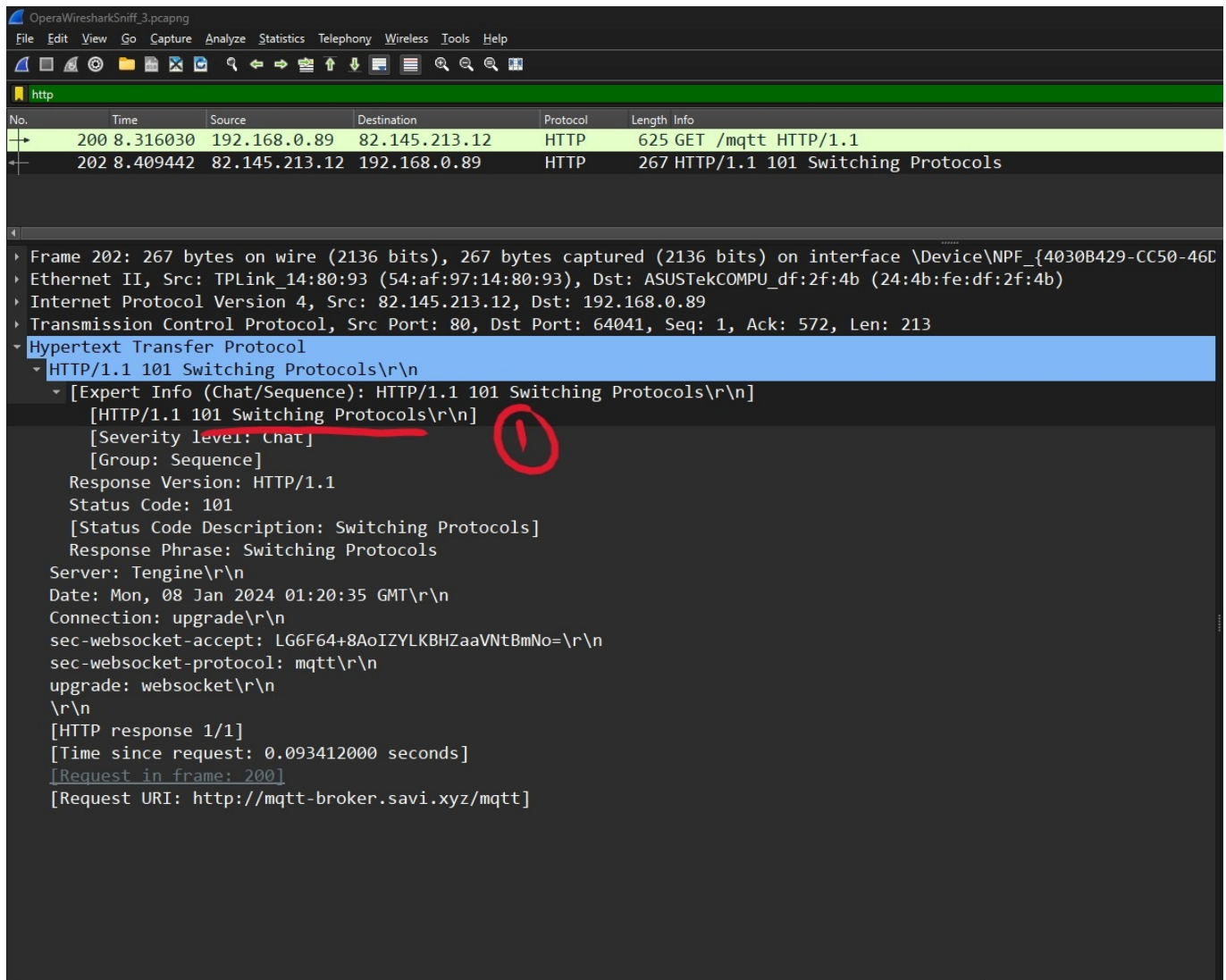


Figure 3.2: Server Response.

- How many HTTP GET request messages did your browser send? Which packet number in the trace contains the GET message for the Bill of Rights?
 - Figure 3.1 #1.1: The browser sent only one HTTP GET request.
 - Figure 3.1 #1.2: There are six (6) different packets that contain the GET message for the Bill of Rights.
- Which packet number in the trace contains the status code and phrase associated with the response to the HTTP GET request?
 - Unable to determine the packet number.
- What is the status code and phrase in the response?
 - Figure 3.2 #1: **Status Code: 101**
 - Figure 3.2 #1: **Response Phrase: Switching Protocols**
- How many data-containing TCP segments were needed to carry the single HTTP response and the text of the Bill of Rights?
 - Figure 3.2 #1.3: Four (4) TCP segments were needed: 218, 220, 265, 266.

4. HTML Documents with Embedded Objects.

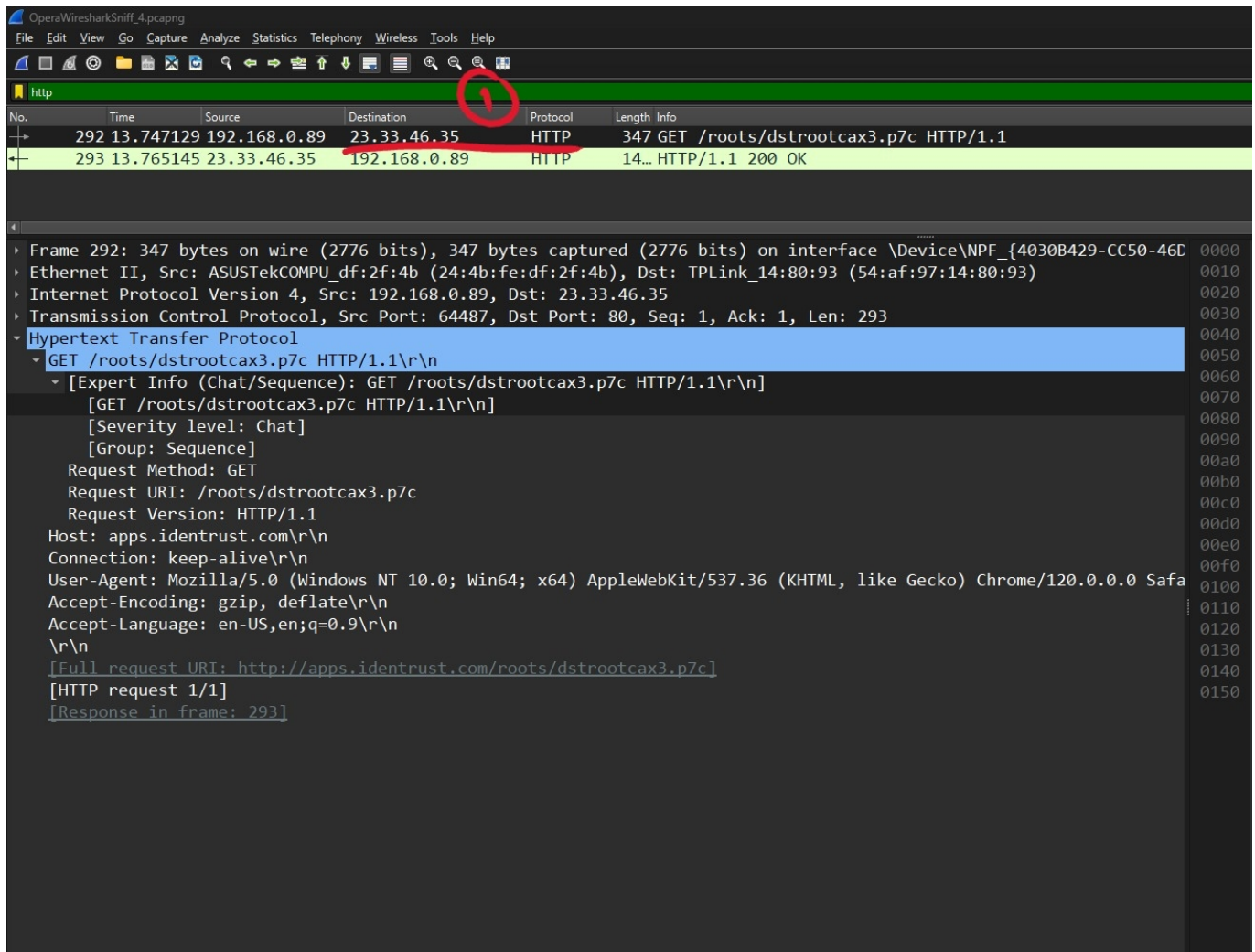


Figure 4.1: Client and Server.

16. How many HTTP GET request messages did your browser send? To which Internet addresses were these GET requests sent?
 - Figure 4.1 #1: Wireshark displays only one (1) GET request. Maybe due to the browser (Opera Crypto) it is only displaying one but since there are multiple images, it should probably be three (3) GET requests. One for making the request and the other two for obtaining the images.
 - Figure 4.1 #1: Only able to determine one Internet address: 23.33.46.35.
17. Can you tell whether your browser downloaded the two images serially, or whether they were downloaded from the two web sites in parallel? Explain.
 - **Unable** to tell.

5. HTTP Authentication

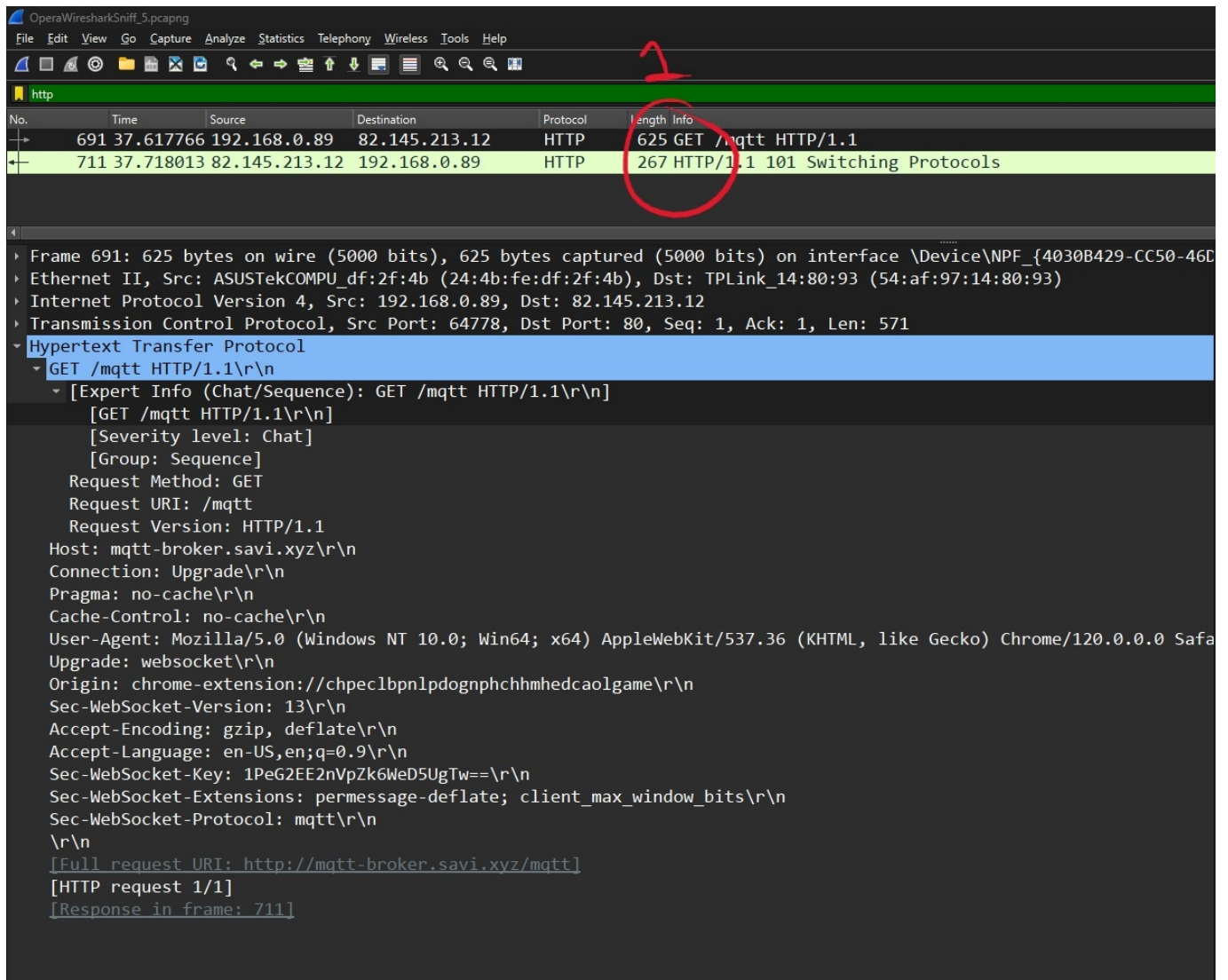


Figure 5.1: Client.

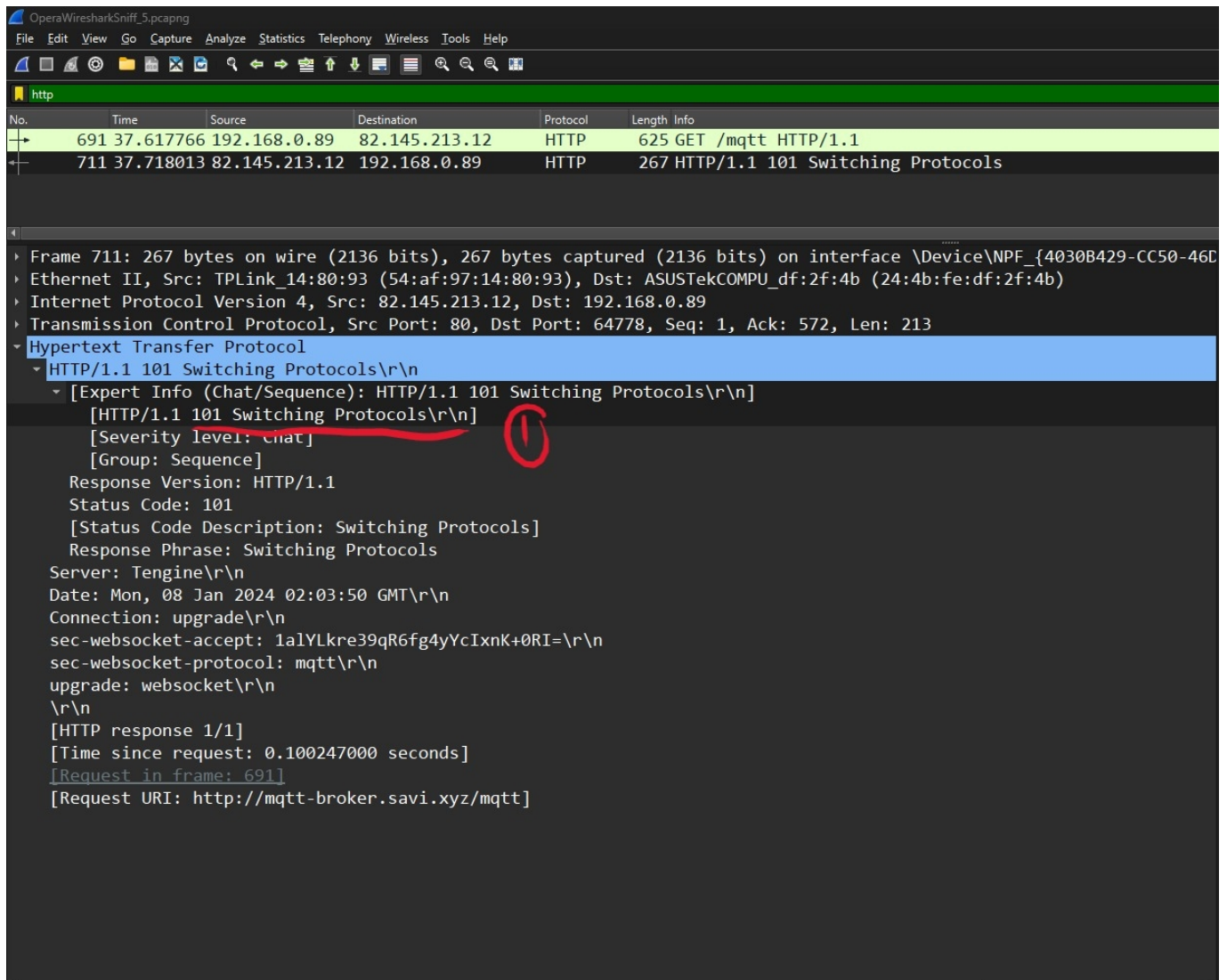


Figure 5.2: Server.

- What is the server's response (status code and phrase) in response to the initial HTTP GET message from your browser?
 - Figure 5.2: Status Code: 101
 - Figure 5.2: Respond Phrase: Switching Protocols
- When your browser's sends the HTTP GET message for the second time, what new field is included in the HTTP GET message?
 - Figure 5.1: Only able to see one (1) GET message.