CSCI 466: Networks

Lecture 6: Wireshark

Reese Pearsall

Fall 2023

*All images are stolen from the internet

Announcements

Wireshark Lab 1 is released, due on **September 20th**

Cookies can be stolen using a variety of different attacks (XSS, Session Hijack, etc). Cookies are generally encrypted

Download Speed vs Upload Speed

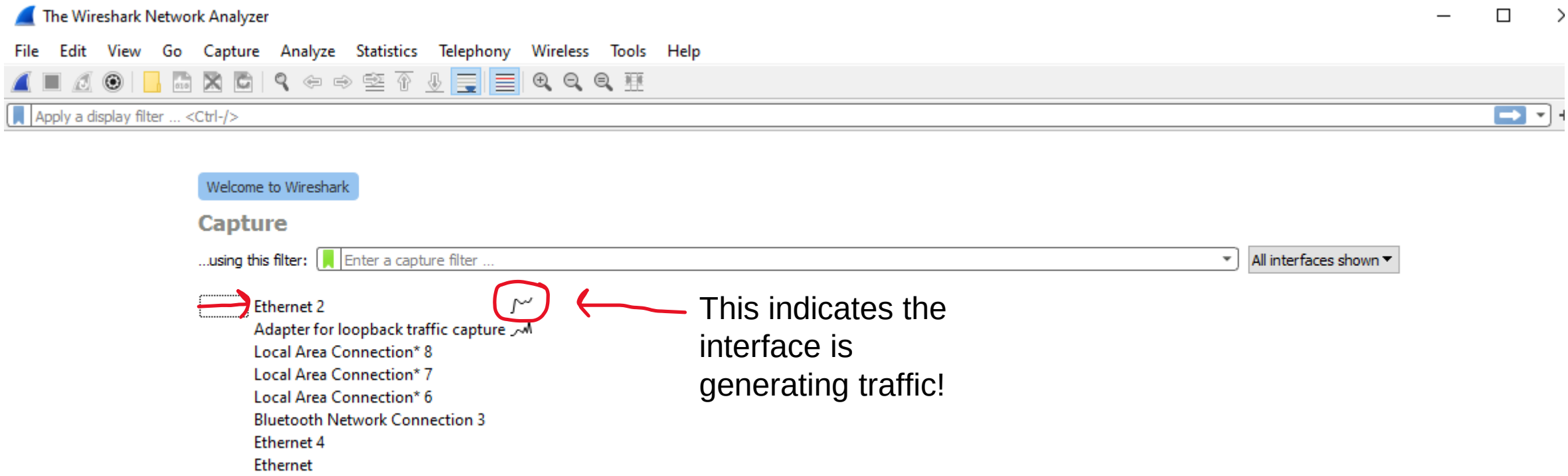
Wireshark

Wireshark is a free and open-source network packet analyzer

It captures packets that are leaving and arriving to your machine and provides details about them

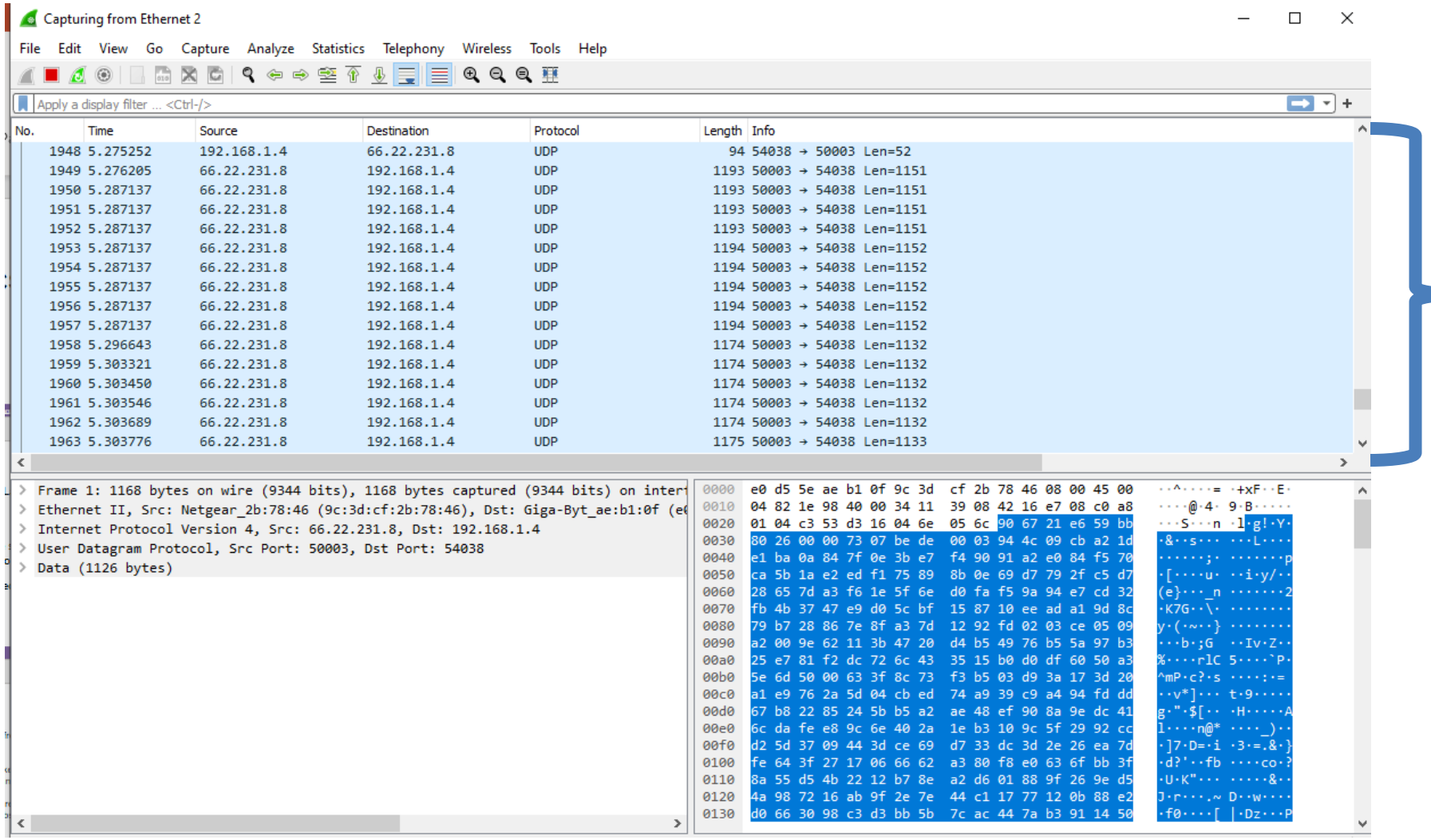
You will use Wireshark to analyze real life network traffic for the labs

When you boot up Wireshark, you will need to select an **interface** to capture on



This indicates the interface is generating traffic!

In general, you will select **ethernet** (wired connections) or **Wi-fi** (wireless connections)



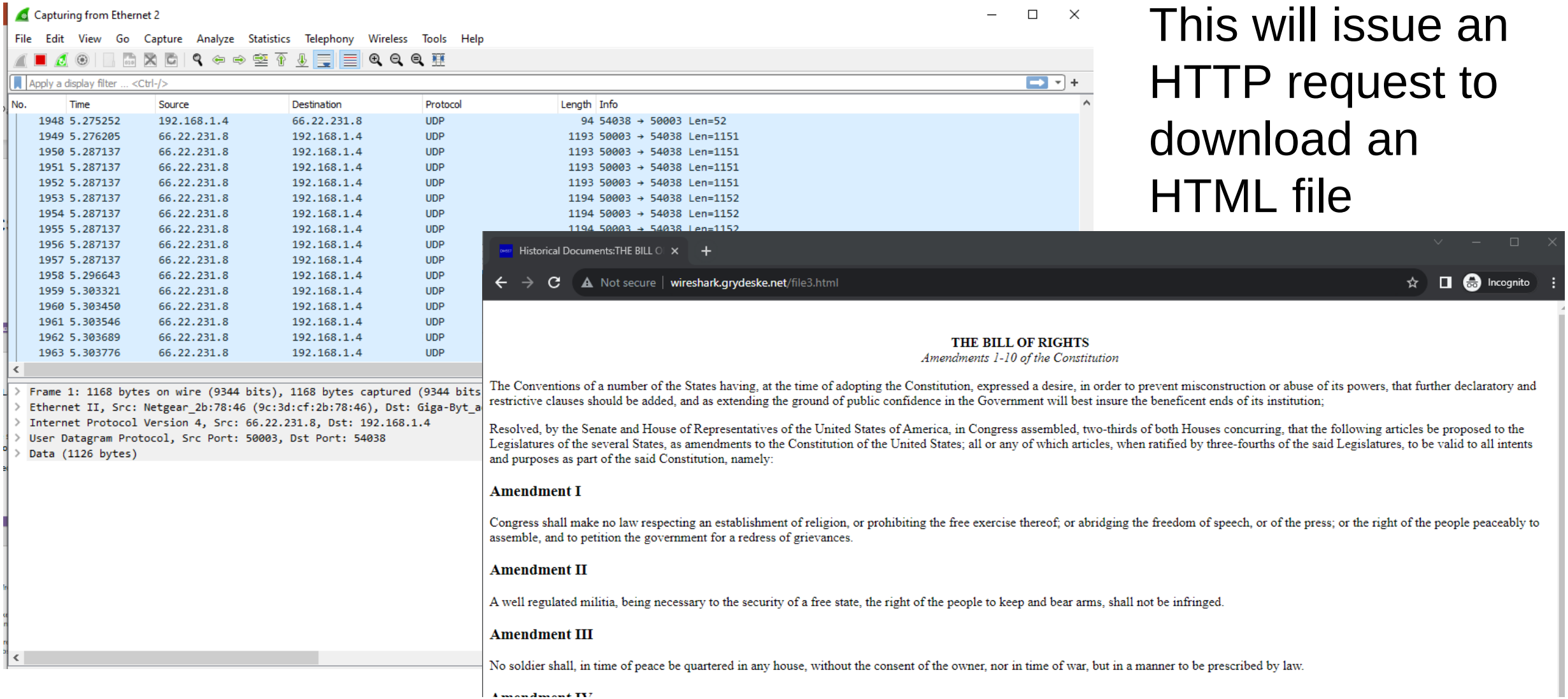
Each row represents a single packet

Once an interface is selected, it will begin capturing packets!

Let’s start generating some HTTP traffic!

We can do this via browsing the web, or through one of our other tools (curl, postman, etc)

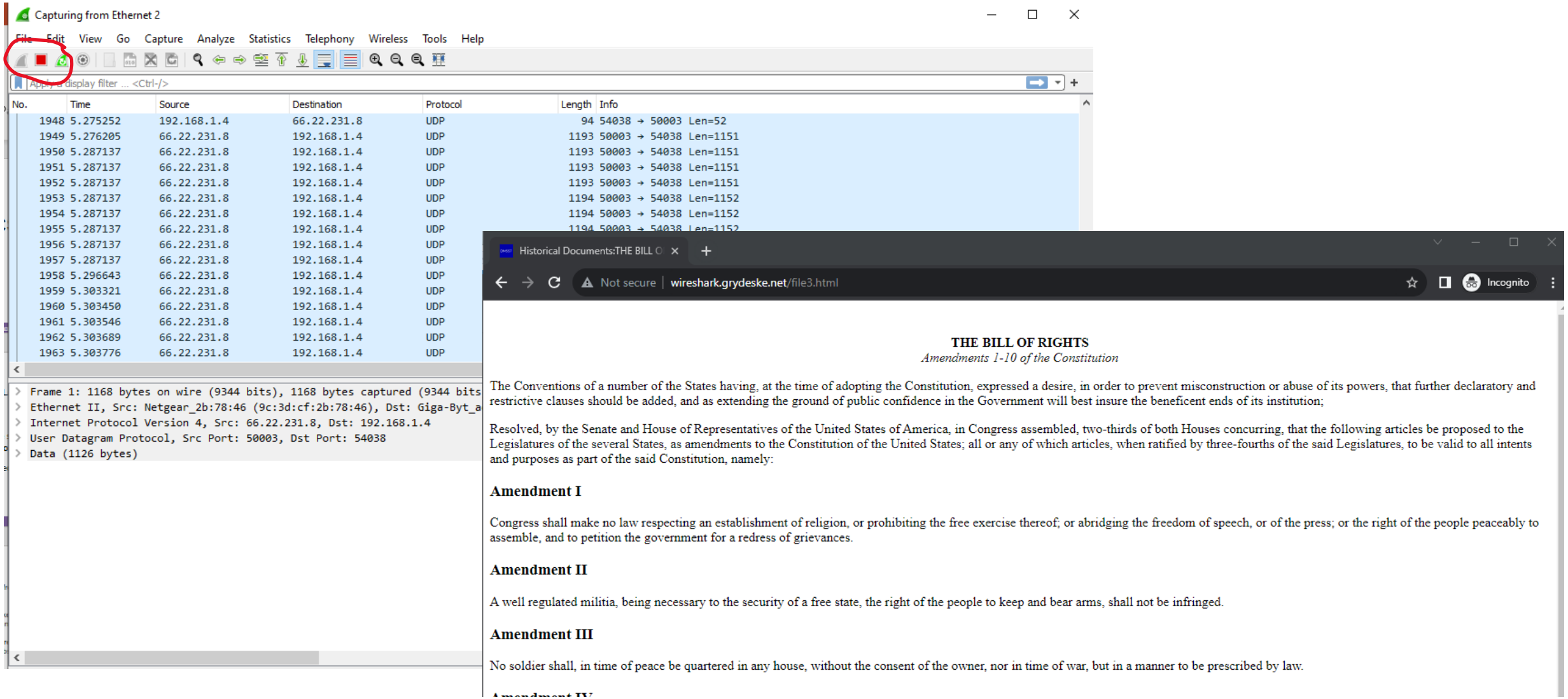
If you use a web browser, I recommend using an incognito/private window



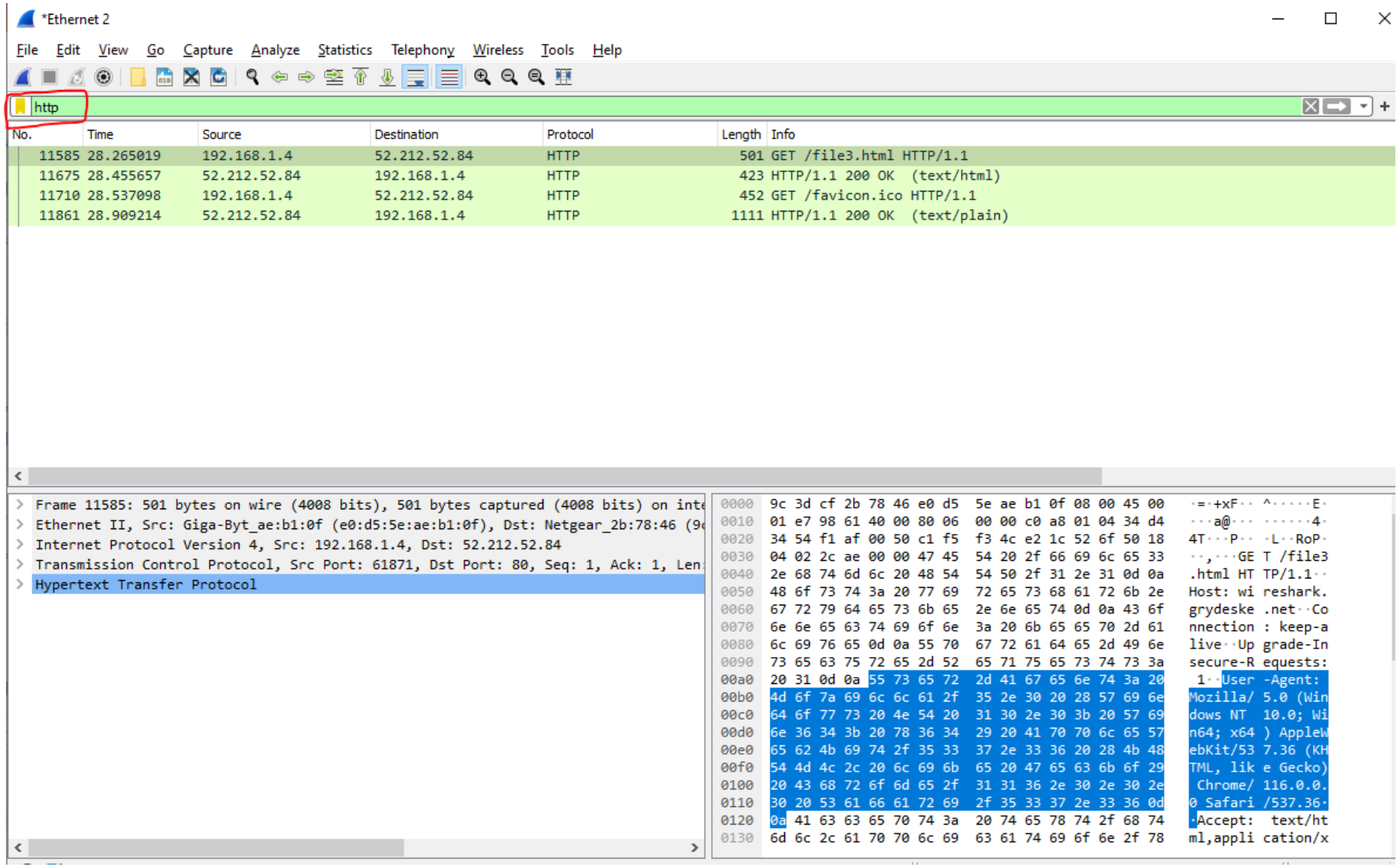
This will issue an HTTP request to download an HTML file

http://wireshark.grydeske.net/file3.html

(must use HTTP, not HTTPS)



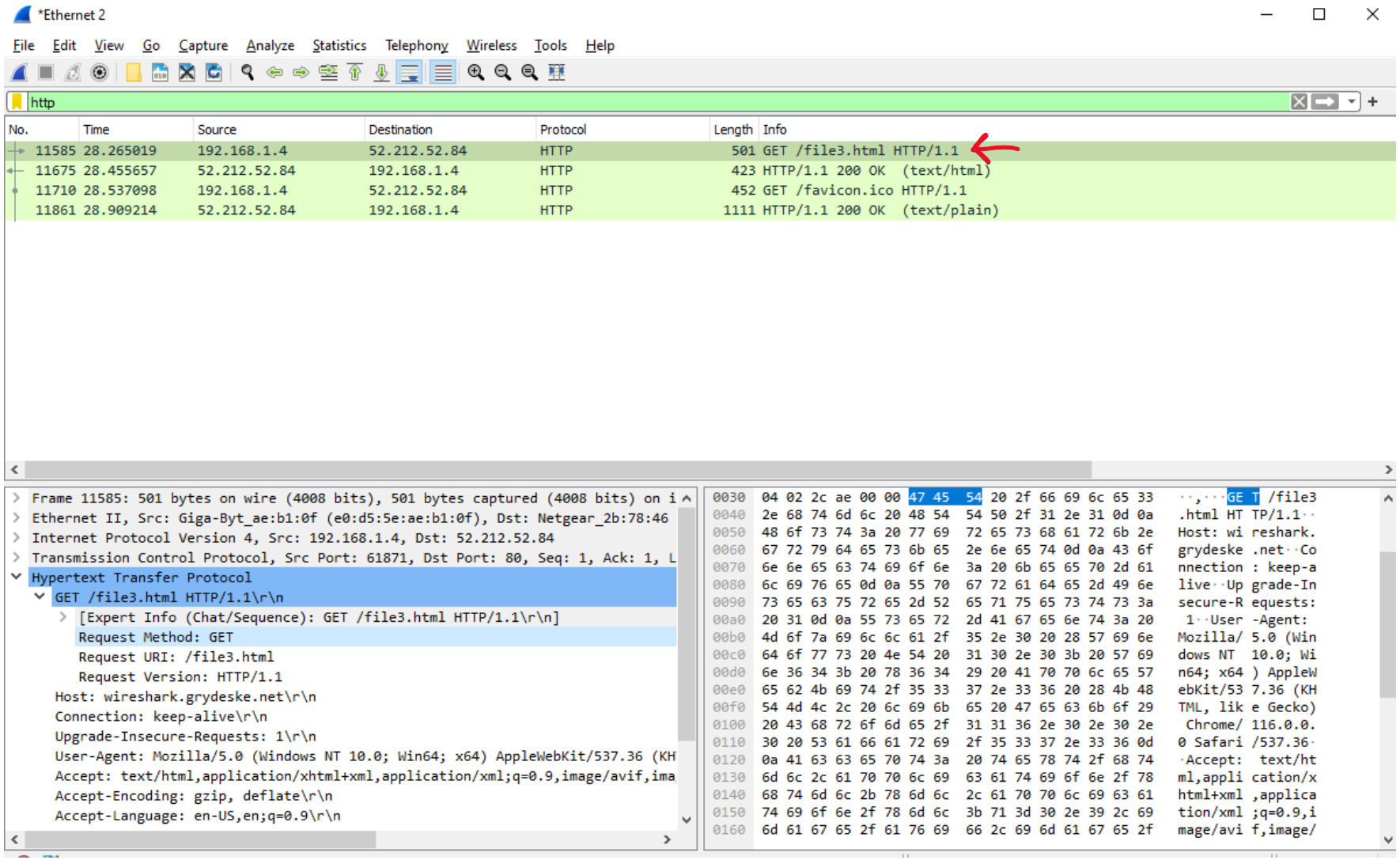
Once the page is loaded, press the red square button to stop collecting data



We can add a **display filter** to filter out any non-http traffic

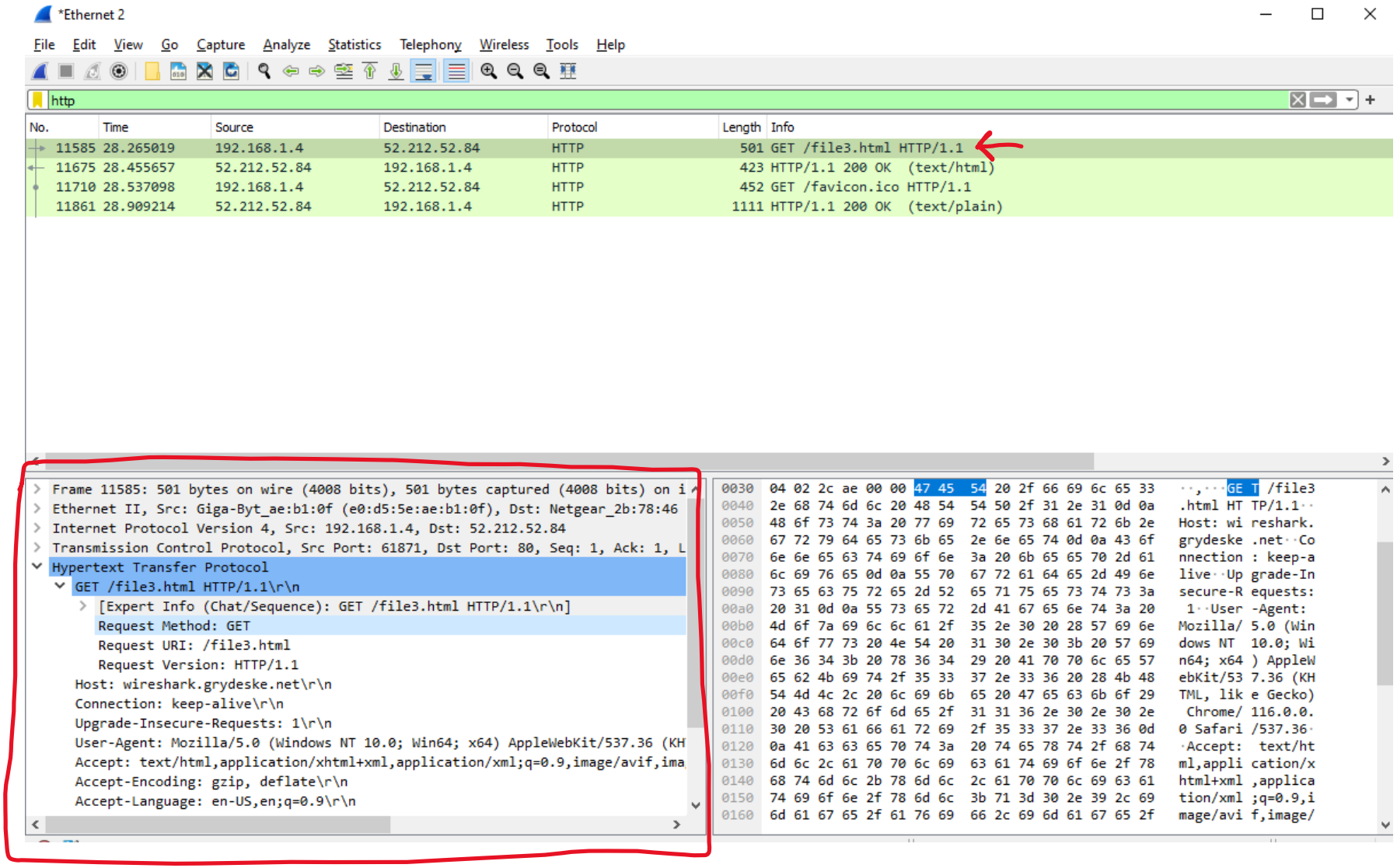
Add the filter `http`

This will show only HTTP packets



Click on the first packet

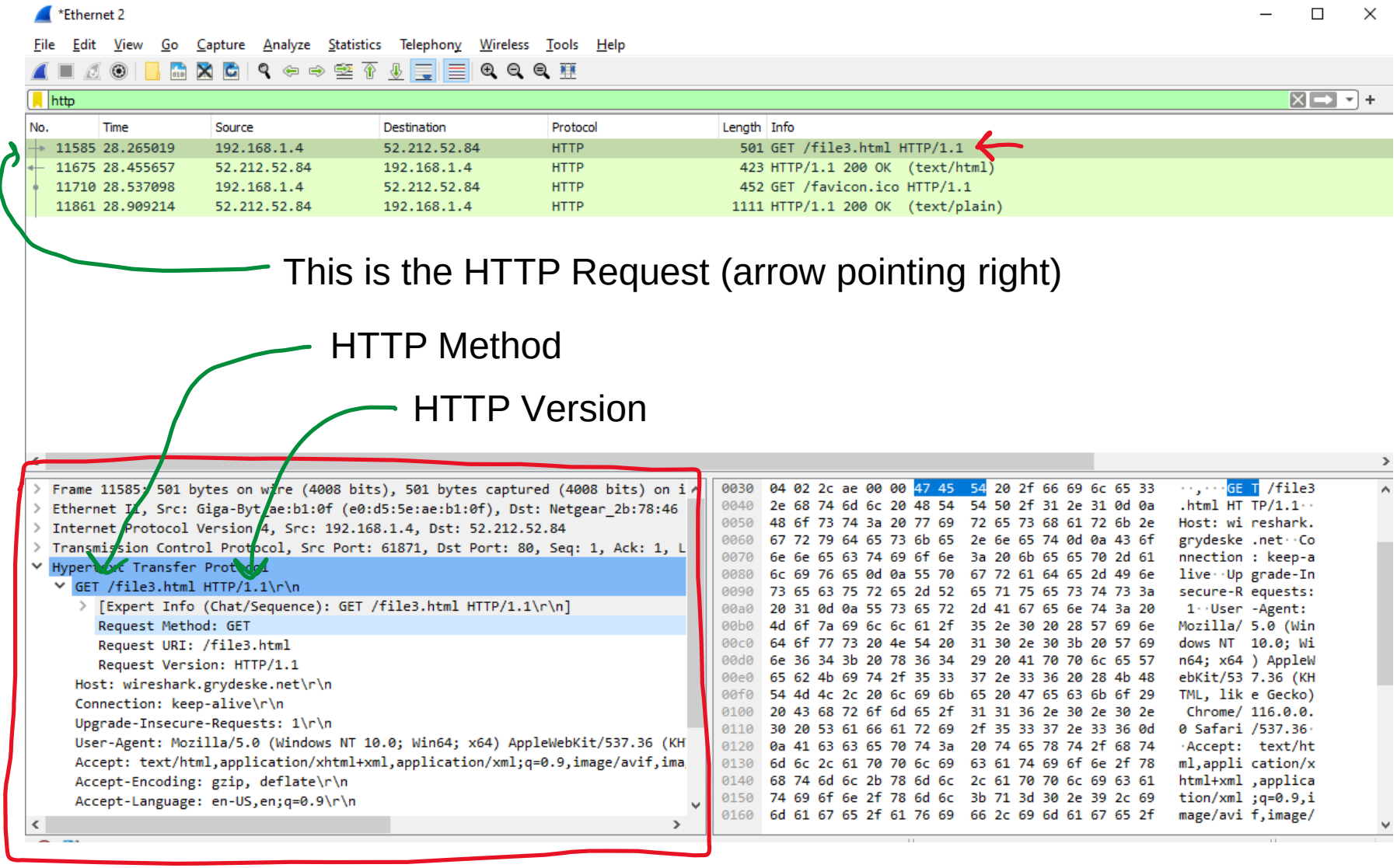
Click on the first packet



This pane shows detailed information about the entire packet (at all layers of the OSI)

We can click the “Hypertext Transfer Protocol” dropdown to view contents of the HTTP Request/Response

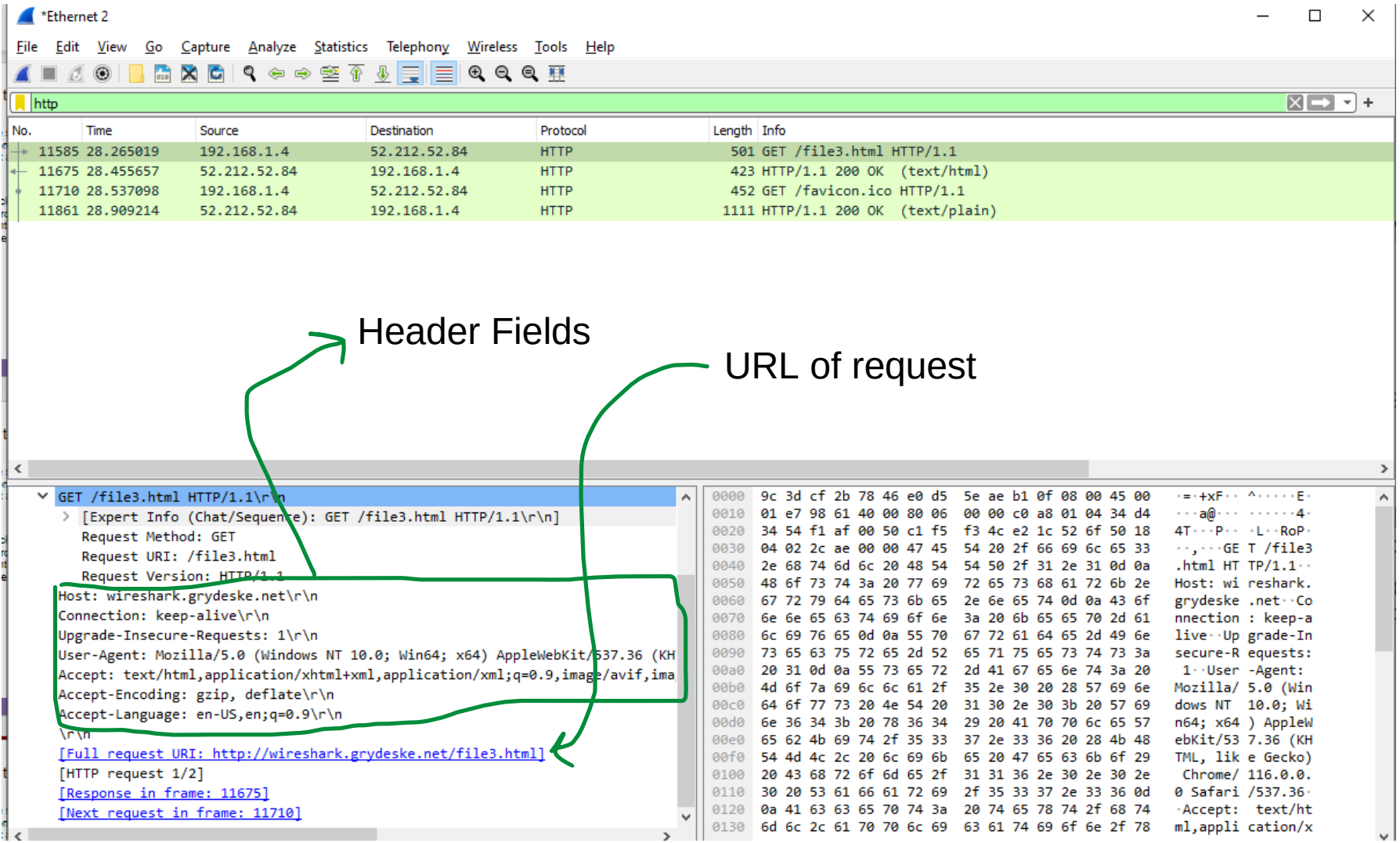
Click on the first packet



This pane shows detailed information about the entire packet (at all layers of the OSI)

We can click the “Hypertext Transfer Protocol” dropdown to view contents of the HTTP Request/Response

Click on the first packet



This pane shows detailed information about the entire packet (at all layers of the OSI)

We can click the “Hypertext Transfer Protocol” dropdown to view contents of the HTTP Request/Response

*Ethernet 2

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

http

No.	Time	Source	Destination	Protocol	Length	Info
11585	28.265019	192.168.1.4	52.212.52.84	HTTP	501	GET /file3.html HTTP/1.1
11675	28.455657	52.212.52.84	192.168.1.4	HTTP	423	HTTP/1.1 200 OK (text/html)
11710	28.537098	192.168.1.4	52.212.52.84	HTTP	452	GET /favicon.ico HTTP/1.1
11861	28.909214	52.212.52.84	192.168.1.4	HTTP	1111	HTTP/1.1 200 OK (text/plain)

The second packet is the HTTP Response (arrow pointing left)


```
> Frame 11675: 423 bytes on wire (3384 bits), 423 bytes captured (3384 bits) on i
> Ethernet II, Src: Netgear_2b:78:46 (9c:3d:cf:2b:78:46), Dst: Giga-Byt_ae:b1:0f
> Internet Protocol Version 4, Src: 52.212.52.84, Dst: 192.168.1.4
> Transmission Control Protocol, Src Port: 80, Dst Port: 61871, Seq: 4381, Ack: 4
> [4 Reassembled TCP Segments (4749 bytes): #11672(1460), #11673(1460), #11674(14
v Hypertext Transfer Protocol
  v HTTP/1.1 200 OK\r\n
    > [Expert Info (Chat/Sequence): HTTP/1.1 200 OK\r\n]
      Response Version: HTTP/1.1
      Status Code: 200
      [Status Code Description: OK]
      Response Phrase: OK
    Connection: keep-alive\r\n
    Server: nginx\r\n
    Date: Fri, 08 Sep 2023 05:19:10 GMT\r\n
    Content-Type: text/html\r\n
    > Content-Length: 4500\r\n
```

```
0000  e0 d5 5e ae b1 0f 9c 3d  cf 2b 78 46 08 00 45 00   ··^····= ·+xF··E·
0010  01 99 4e a3 40 00 29 06  d6 e7 34 d4 34 54 c0 a8   ··N·@·)· ··4·4T··
0020  01 04 00 50 f1 af e2 1c  63 8b c1 f5 f5 0b 50 18   ···P···· c·····P·
0030  00 46 3d a1 00 00 68 65  20 43 6f 6e 73 74 69 74   ·F=···he  Constit
0040  75 74 69 6f 6e 2c 20 6f  66 20 63 65 72 74 61 69   ution, o f certai
0050  6e 20 72 69 67 68 74 73  2c 20 73 68 61 6c 6c 0a   n rights , shall·
0060  6e 6f 74 20 62 65 20 63  6f 6e 73 74 72 75 65 64   not be c onstrued
0070  20 74 6f 20 64 65 6e 79  20 6f 72 20 64 69 73 70    to deny  or disp
0080  61 72 61 67 65 20 6f 74  68 65 72 73 20 72 65 74   arage ot hers ret
0090  61 69 6e 65 64 20 62 79  20 74 68 65 20 70 65 6f   ained by  the peo
00a0  70 6c 65 2e 0a 0a 3c 2f  70 3e 3c 70 3e 3c 61 20   ple.··</ p><p><a
00b0  6e 61 6d 65 3d 22 31 30  22 3e 3c 73 74 72 6f 6e   name="10 "><stron
00c0  67 3e 3c 68 33 3e 41 6d  65 6e 64 6d 65 6e 74 20   g><h3>Am endment
00d0  58 3c 2f 68 33 3e 3c 2f  73 74 72 6f 6e 67 3e 3c   X</h3></ strong><
00e0  2f 61 3e 0a 0a 3c 70 3e  3c 2f 70 3e 0a 3c 70 3e   /a>··<p> </p>·<p>
00f0  54 68 65 20 70 6f 77 65  72 73 20 6e 6f 74 20 64   The powe rs not d
0100  65 6c 65 67 61 74 65 64  20 74 6f 20 74 68 65 20   elegated  to the
0110  55 6e 69 74 65 64 20 53  74 61 74 65 73 20 62 79   United S tates by
0120  20 74 68 65 20 43 6f 6e  73 74 69 74 75 74 69 6f    the Con stitutio
```

*Ethernet 2

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

http

No.	Time	Source	Destination	Protocol	Length	Info
11585	28.265019	192.168.1.4	52.212.52.84	HTTP	501	GET /file3.html HTTP/1.1
11675	28.455657	52.212.52.84	192.168.1.4	HTTP	423	HTTP/1.1 200 OK (text/html)
11710	28.537098	192.168.1.4	52.212.52.84	HTTP	452	GET /favicon.ico HTTP/1.1
11861	28.909214	52.212.52.84	192.168.1.4	HTTP	1111	HTTP/1.1 200 OK (text/plain)

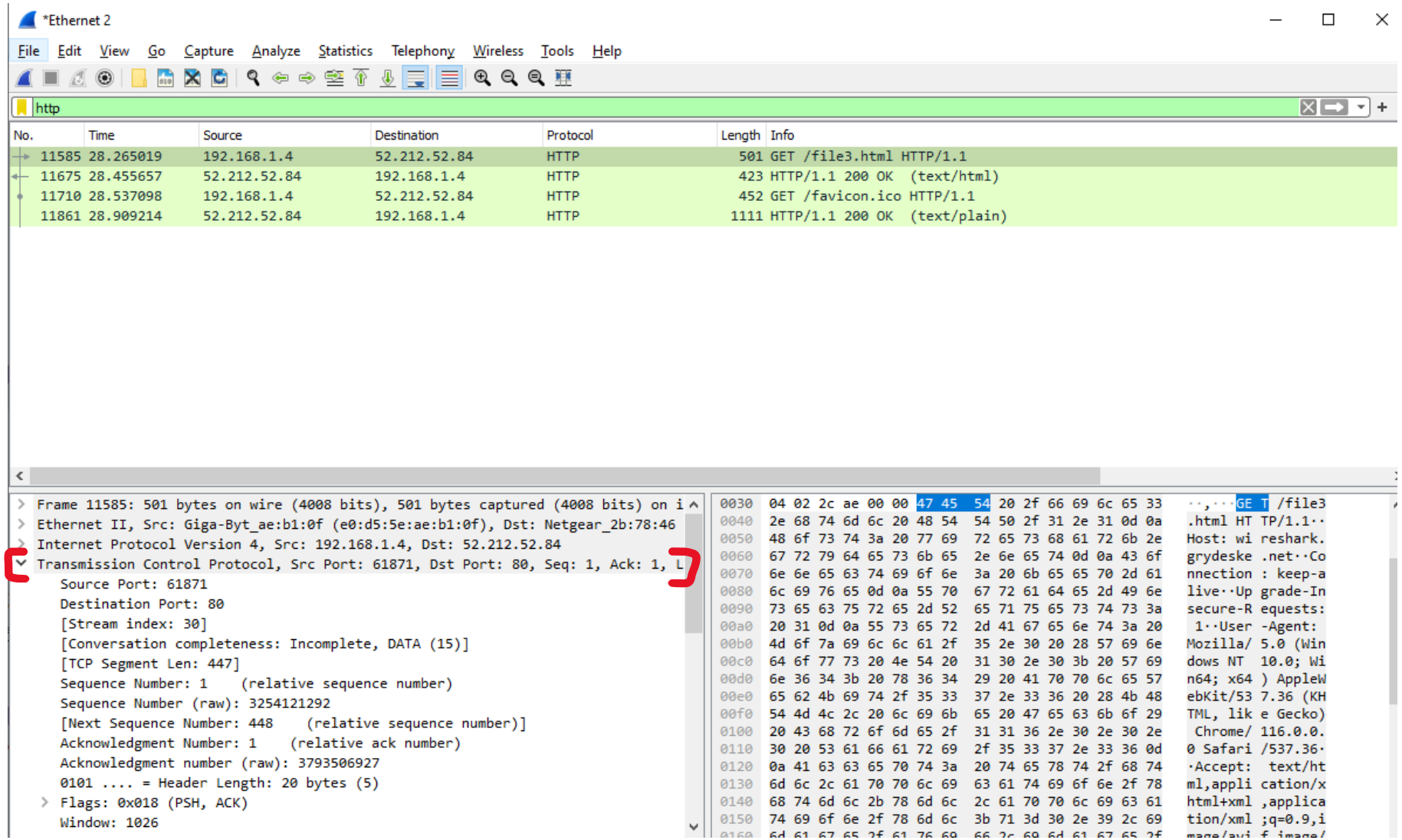
Response Status Code

HTTP Version of Response

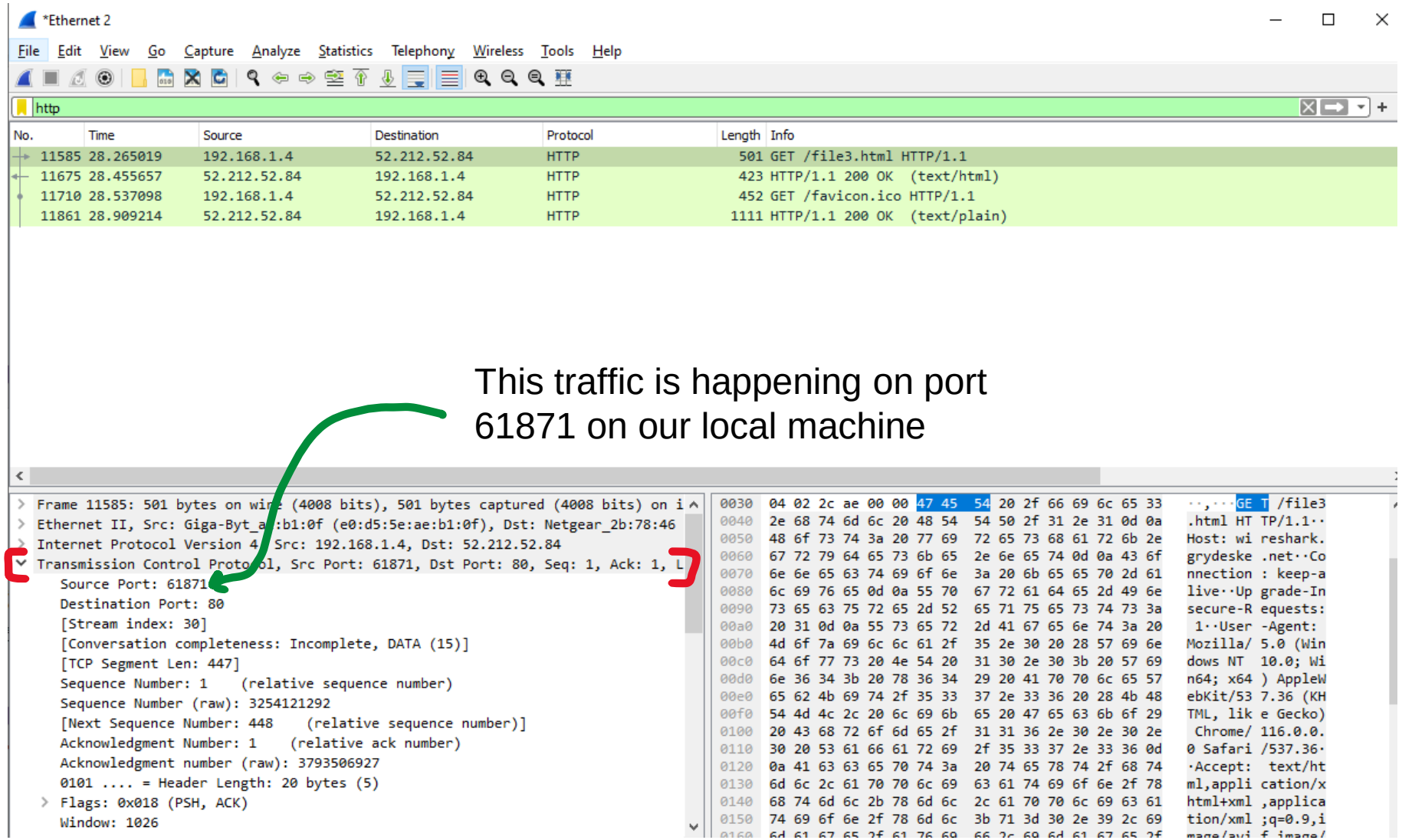
```
> Frame 11675: 423 bytes on wire (3384 bits), 423 bytes captured (3384 bits) on i
> Ethernet II, Src: Netgear_2b:78:46 (9c:3d:cf:2b:78:46), Dst: Giga-Byt_ae:b1:0f
> Internet Protocol Version 4, Src: 52.212.52.84, Dst: 192.168.1.4
> Transmission Control Protocol, Src Port: 80, Dst Port: 61871, Seq: 4381, Ack: 4
> [4 Reassembled TCP Segments (4749 bytes): #11672(1460), #11673(1460), #11674(14
v Hypertext Transfer Protocol
  v HTTP/1.1 200 OK\r\n
    > [Expert Info (Chat/Sequence): HTTP/1.1 200 OK\r\n]
      Response Version: HTTP/1.1
      Status Code: 200
      [Status Code Description: OK]
      Response Phrase: OK
    Connection: keep-alive\r\n
    Server: nginx\r\n
    Date: Fri, 08 Sep 2023 05:19:10 GMT\r\n
    Content-Type: text/html\r\n
    > Content-Length: 4500\r\n
```

```
0000  e0 d5 5e ae b1 0f 9c 3d  cf 2b 78 46 08 00 45 00   ··^····= ·+xF··E·
0010  01 99 4e a3 40 00 29 06  d6 e7 34 d4 34 54 c0 a8   ··N·@·)· ··4·4T··
0020  01 04 00 50 f1 af e2 1c  63 8b c1 f5 f5 0b 50 18   ···P···· c·····P·
0030  00 46 3d a1 00 00 68 65  20 43 6f 6e 73 74 69 74   ·F=···he  Constit
0040  75 74 69 6f 6e 2c 20 6f  66 20 63 65 72 74 61 69   ution, o f certai
0050  6e 20 72 69 67 68 74 73  2c 20 73 68 61 6c 6c 0a   n rights , shall·
0060  6e 6f 74 20 62 65 20 63  6f 6e 73 74 72 75 65 64   not be c onstrued
0070  20 74 6f 20 64 65 6e 79  20 6f 72 20 64 69 73 70    to deny  or disp
0080  61 72 61 67 65 20 6f 74  68 65 72 73 20 72 65 74   arage ot hers ret
0090  61 69 6e 65 64 20 62 79  20 74 68 65 20 70 65 6f   ained by  the peo
00a0  70 6c 65 2e 0a 0a 3c 2f  70 3e 3c 70 3e 3c 61 20   ple.··</ p><p><a
00b0  6e 61 6d 65 3d 22 31 30  22 3e 3c 73 74 72 6f 6e   name="10 "><stron
00c0  67 3e 3c 68 33 3e 41 6d  65 6e 64 6d 65 6e 74 20   g><h3>Am endment
00d0  58 3c 2f 68 33 3e 3c 2f  73 74 72 6f 6e 67 3e 3c   X</h3></ strong><
00e0  2f 61 3e 0a 0a 3c 70 3e  3c 2f 70 3e 0a 3c 70 3e   /a>··<p> </p>·<p>
00f0  54 68 65 20 70 6f 77 65  72 73 20 6e 6f 74 20 64   The powe rs not d
0100  65 6c 65 67 61 74 65 64  20 74 6f 20 74 68 65 20   elegated  to the
0110  55 6e 69 74 65 64 20 53  74 61 74 65 73 20 62 79   United S tates by
0120  20 74 68 65 20 43 6f 6e  73 74 69 74 75 74 69 6f    the Con stitutio
```

This pane will try to decode the contents of the HTTP request/HTTP response (HTML)

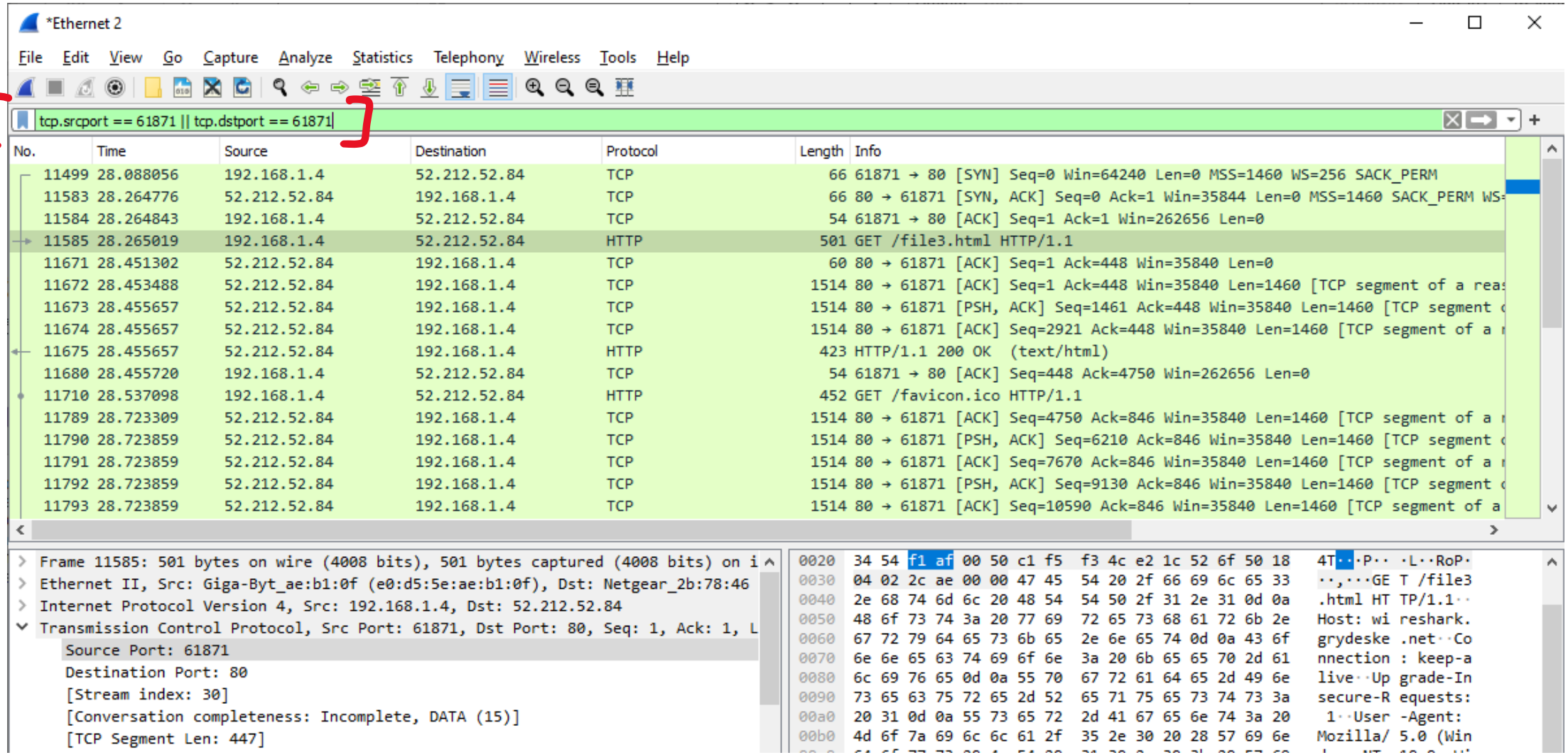


By clicking on "Transmission Control Protocol" we can see information about the transport layer



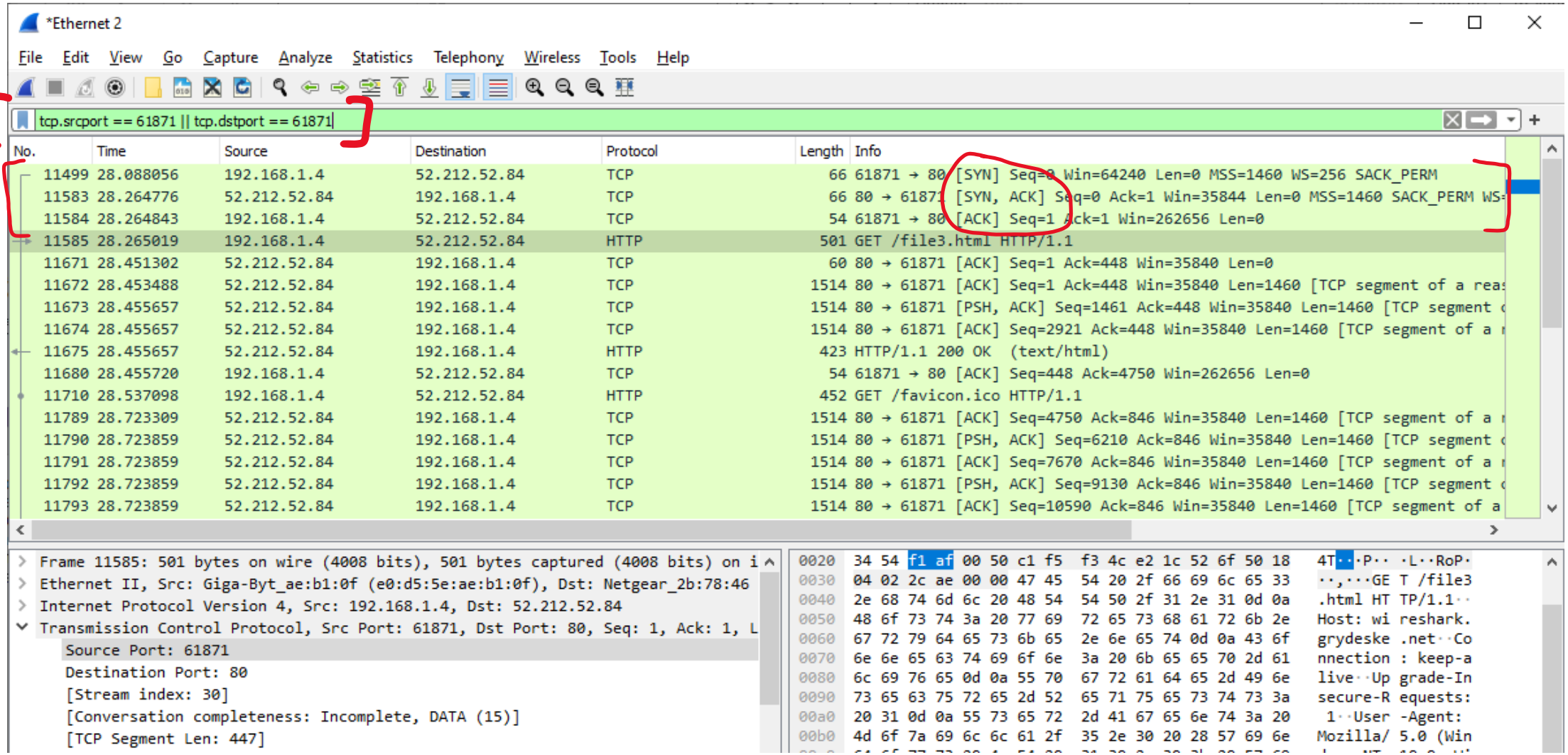
This traffic is happening on port 61871 on our local machine

By clicking on "Transmission Control Protocol" we can see information about the transport layer



Let's change our filter to `tcp.srcport == 61871 || tcp.dstport == 61871`

This will only show traffic where the source port or destination port is 61871



The first three packets is us establishing a TCP connection before we send our fist HTTP request!