

Lab Sheet 1

Analyzing http using Wireshark

Name: Vinayak V Thayil

Roll No:AM.EN.U4CSE21161

1. Open the Packet sniffer [Wireshark] Application and Capture YOUR NETWORK Interface

| | | | | |
|---------------|------------------------|-------------------|--------|---|
| 3649 3.900694 | fe80::e113:7ce4:f07... | ff02::1:ff00:1 | ICMPv6 | 86 Neighbor Solicitation for fe80::1 from 2c:3b:70:fa:c1:95 |
| 3650 3.900750 | fe80::8e86:6482:622... | ff02::1:ff22:b525 | ICMPv6 | 86 Multicast Listener Report |
| 3651 3.901536 | 10.110.68.24 | 224.0.0.251 | MDNS | 555 Standard query response 0x0000 TXT cache flush PTR _mi-connect_udp.local PTR ("nm":"Mi 101","as":"[8194]","ip":"24")_mi-connect... |
| 3652 3.902965 | fe80::1c92:ffe8:b4e... | ff02::fb | MDNS | 1018 Standard query response 0x0000 TXT PTR HEWAN's MacBook Air._airplay_tcp.local PTR 1085884EE77B@HEWAN's MacBook Air._raop_tcp.lo... |
| 3653 3.903185 | fe80::b92d:3261:ecb... | ff02::1:ff00:1 | ICMPv6 | 86 Neighbor Solicitation for fe80::1 from 90:e8:68:13:06:9f |
| 3654 3.903394 | fe80::a375:5902:21e... | ff02::1:ff00:1 | ICMPv6 | 86 Neighbor Solicitation for fe80::1 from ac:19:8e:ee:81:71 |
| 3655 3.904746 | 10.113.21.140 | 224.0.0.251 | MDNS | 998 Standard query response 0x0000 TXT PTR HEWAN's MacBook Air._airplay_tcp.local PTR 1085884EE77B@HEWAN's MacBook Air._raop_tcp.lo... |
| 3656 3.905695 | 0.0.0.0 | 255.255.255.255 | DHCP | 590 DHCP Discover - Transaction ID 0x3d25a522 |
| 3657 3.905760 | fe80::bbf7:bea2:539... | ff02::1:ff00:1 | ICMPv6 | 86 Neighbor Solicitation for fe80::1 from e0:d0:45:5c:46:ee |
| 3658 3.906088 | fe80::fafb:125b:901... | ff02::1:2 | DHCPv6 | 120 Information-request XID: 0x39a58e CID: 000100012c98742b2c3b705874f1 |
| 3659 3.906122 | GxIntern_22:a4:db | Broadcast | ARP | 56 Who has 192.168.0.250? Tell 192.168.1.1 (duplicate use of 192.168.1.1 detected!) |
| 3660 3.906565 | 10.113.18.67 | 10.113.255.255 | NDNS | 92 Name query NB DESKTOP-8U9K51c1c |
| 3661 3.906681 | fe80::e918:f68f:7a7... | ff02::1:ff00:1 | ICMPv6 | 86 Neighbor Solicitation for fe80::1 from 9c:da:3e:84:a1:25 |
| 3662 3.906716 | GxIntern_1f:e2:7f | Broadcast | ARP | 56 Who has 192.5.41.41? Tell 192.168.1.1 (duplicate use of 192.168.1.1 detected!) |
| 3663 3.907049 | fe80::1081:74f4:d22... | ff02::2:ff81:fa78 | ICMPv6 | 86 Multicast Listener Report |
| 3664 3.907226 | IntelCor_d2:aa:2a | Broadcast | ARP | 56 Who has 10.110.54.224? Tell 10.113.11.109 |
| 3665 3.907431 | RealmeCh_9a:d1:4f | Broadcast | ARP | 56 Who has 10.110.4.93? Tell 10.110.42.23 |
| 3666 3.907480 | RealmeCh_9a:d1:4f | Broadcast | ARP | 56 Who has 10.110.20.92? Tell 10.110.42.23 |
| 3667 3.907505 | fe80::3c0f:492f:d9a... | ff02::1:ff00:1 | ICMPv6 | 86 Neighbor Solicitation for fe80::1 from d4:54:8b:40:96:0b |
| 3668 3.908871 | 10.110.57.48 | 224.0.0.251 | MDNS | 715 Standard query 0x0000 PTR _companion-link_tcp.local, "QM" question PTR _rdlink_tcp.local, "QM" question PTR _hap_tcp.local, "Q... |
| 3669 3.909060 | fe80::d96b:1d0a:5d0... | ff02::1:ff00:1 | ICMPv6 | 86 Neighbor Solicitation for fe80::1 from 48:e7:da:0d:07:ad |
| 3670 3.909194 | fe80::e6df:e6d4:a89... | ff02::1:ff00:1 | ICMPv6 | 86 Neighbor Solicitation for fe80::1 from 84:7b:57:af:b4:0f |

| | |
|---|--|
| Frame 1: 179 bytes on wire (1432 bits), 179 bytes captured (1432 bits) on Ethernet II, Src: Azurelax_65:e2:a9 (00:e9:3a:65:e2:a9), Dst: IPv6mcast | 0000 01 00 5e 7f ff fa 00 e9 3a 65 e2 a9 00 00 45 00 ..^.....:e....E- |
| Internet Protocol Version 4, Src: 10.113.18.66, Dst: 239.255.255.250 | 0010 00 a5 fc bc 00 00 04 11 ac de 0a 71 12 42 ef ff:q.B.. |
| User Datagram Protocol, Src Port: 57617, Dst Port: 1900 | 0020 ff fa e1 11 07 6c 00 91 17 83 4d 2d 53 45 41 52:M-SEAR |
| Simple Service Discovery Protocol | 0030 43 48 20 2a 20 48 54 54 50 2f 31 2e 31 0d 0a 48 CH * HTT P/1:1 -H |
| | 0040 6f 73 74 3a 20 32 33 39 2e 32 35 35 2e 32 35 35 ost: 239 .255.255 |
| | 0050 2e 32 35 30 3a 31 39 30 30 0d 0a 53 54 3a 20 75 .250:190 0: ST: u |
| | 0060 72 6e 3a 73 63 68 65 6d 61 73 2d 75 70 6e 70 2d rn:schem as-upnp- |
| | 0070 6f 72 67 3a 64 65 76 69 63 65 3a 49 6e 74 65 72 org:devi ce:Inter |
| | 0080 6e 65 74 47 61 74 65 77 61 79 44 65 76 69 63 65 netGatew ayDevice |
| | 0090 3a 31 0d 0a 4d 61 6e 3a 20 22 73 73 64 70 3a 64 :1:Man: "ssdp:d |
| | 00a0 69 73 63 6f 76 65 72 2d 0d 0a 4d 58 3a 20 33 0d iscover" --MX: 3- |
| | 00b0 0a 0d 0a |

2. Do this activity capture frames?

a. Request for a web page from amrita.edu and search for some keywords on the webpage

Do-It-Yourself Web Authoring - a beginner's HTML tutorial



A random photo... (The Hudson River at 125th Street about 2002)

[Frank da Cruz](#)

Updated in 2019 and 2021 for HTML5 and "fluidity".

This page shows how to create Web pages by hand, the original way. Although today most Web pages are created by "Web authoring systems" that are designed to shield you from technical details, the fact is that HTML (the "programming" language of the Web) is not that difficult, as you can see if you follow this tutorial. To get an idea of what is possible with this technique, see these 100% hand-made websites:

- [The New Deal in New York City 1933-1943](#)
- [The History of Computing at Columbia University 1890-2005](#)
- [The Washington DC Nation Mall in World War II](#)
- [Arlington, Virginia, 1956-61: The Hall's Hill Segregation Wall](#)
- [Frankfurt, Germany, 1959-61](#)

CONTENTS

1. [Creating a Web Page](#)
2. [HTML Syntax](#)
3. [Special Characters](#)
4. [Converting Plain Text to HTML](#)
5. [Effects](#)
6. [Lists](#)

b. Open any URL where you can post comments) in your browser and post a comment in it.

Wi-Fi
File Edit View Go Capture Analyze Statistics Telephony Wireless Help

HTTP

| No. | Time | Source | Destination | Protocol | Length | Info |
|------|-----------|---|--|----------|--------|--|
| 425 | 5.838214 | 192.168.188.155 | 77.234.45.64 | HTTP | 618 | POST /R/A4UKIGWzjh1ZwEYMTQ5hZQYzNHmZk3YTRJhVzJNZ5YNGE0eGEAQJGkwbIgh-KggIBBDxKpAp5oICAMQsY6DQeQcAgCPzm26kBM5eIBBDxKpApAR1ACj1 |
| 505 | 7.269488 | 77.234.45.64 | 192.168.188.155 | HTTP | 641 | HTTP/1.1 200 OK |
| 507 | 7.473183 | 192.168.188.155 | 77.234.45.64 | HTTP | 372 | GET /R/A4UKIGWzjh1ZwEYMTQ5hZQYzNHmZk3YTRJhVzJNZ5YNGE0eGEAQJGkwbIgh-KggIBBDxKpAp5oICAMQsY6DQeQcAgCPzm26kBM5eIBBDxKpApAR1ACj1 |
| 582 | 8.590695 | 77.234.45.64 | 192.168.188.155 | HTTP | 210 | HTTP/1.1 200 OK |
| 584 | 8.717987 | 192.168.188.155 | 77.234.45.64 | HTTP | 372 | GET /R/A4UKIGWzjh1ZwEYMTQ5hZQYzNHmZk3YTRJhVzJNZ5YNGE0eGEAQJGkwbIgh-KggIBBDxKpAp5oICAMQsY6DQeQcAgCPzm26kBM5eIBBDxKpApAR1ACj1 |
| 665 | 10.340586 | 77.234.45.64 | 192.168.188.155 | HTTP | 70 | HTTP/1.1 200 OK |
| 682 | 11.435178 | 2409:4073::4d0b:35f::2a01:111:2003::52 | 2401:111:2003::52 | HTTP | 229 | GET /connecttest.txt HTTP/1.1 |
| 683 | 11.435179 | 192.168.188.155 | 13.107.4.52 | HTTP | 208 | GET /connecttest.txt HTTP/1.1 |
| 687 | 11.528378 | 2a01:111:2003::52 | 2409:4073::4d0b:35f::2a01:111:2003::52 | HTTP | 613 | HTTP/1.1 200 OK (text/plain) |
| 693 | 11.528720 | 13.107.4.52 | 192.168.188.155 | HTTP | 593 | HTTP/1.1 200 OK (text/plain) |
| 701 | 11.785696 | 192.168.188.155 | 77.234.45.64 | HTTP | 372 | GET /R/A4UKIGWzjh1ZwEYMTQ5hZQYzNHmZk3YTRJhVzJNZ5YNGE0eGEAQJGkwbIgh-KggIBBDxKpAp5oICAMQsY6DQeQcAgCPzm26kBM5eIBBDxKpApAR1ACj1 |
| 729 | 12.388061 | 77.234.45.64 | 192.168.188.155 | HTTP | 981 | HTTP/1.1 200 OK |
| 731 | 12.482786 | 192.168.188.155 | 77.234.45.64 | HTTP | 372 | GET /R/A4UKIGWzjh1ZwEYMTQ5hZQYzNHmZk3YTRJhVzJNZ5YNGE0eGEAQJGkwbIgh-KggIBBDxKpAp5oICAMQsY6DQeQcAgCPzm26kBM5eIBBDxKpApAR1ACj1 |
| 745 | 18.231662 | 2409:4073::4d0b:35f::2404:6800:4007:829:: | 2404:6800:4007:829:: | HTTP | 281 | GET /gsr1/gsr1.crl HTTP/1.1 |
| 748 | 18.466347 | 2404:6800:4007:829:: | 2409:4073::4d0b:35f:: | HTTP | 297 | HTTP/1.1 304 Not Modified |
| 760 | 18.811594 | 192.168.188.155 | 152.195.38.76 | HTTP | 312 | GET /digicert/assuredigrootca.crl HTTP/1.1 |
| 762 | 19.044298 | 152.195.38.76 | 192.168.188.155 | HTTP | 339 | HTTP/1.1 304 Not Modified |
| 990 | 24.503403 | 192.168.188.155 | 44.228.249.3 | HTTP | 768 | POST /connect.php HTTP/1.1 (application/x-www-form-urlencoded) |
| 1021 | 25.085711 | 44.228.249.3 | 192.168.188.155 | HTTP | 673 | HTTP/1.1 200 OK (text/html) |
| 1105 | 27.950606 | 77.234.45.64 | 192.168.188.155 | HTTP | 982 | HTTP/1.1 200 OK |
| 1110 | 27.966409 | 192.168.188.155 | 77.234.45.64 | HTTP | 372 | GET /R/A4UKIGWzjh1ZwEYMTQ5hZQYzNHmZk3YTRJhVzJNZ5YNGE0eGEAQJGkwbIgh-KggIBBDxKpAp5oICAMQsY6DQeQcAgCPzm26kBM5eIBBDxKpApAR1ACj1 |

> Frame 425: 618 bytes on wire (4944 bits), 618 bytes captured (4944 bits) on interface \Device\NPF{6F07...}

Ethernet II, Src: CloudNet77:98:05 (38:45:7a:77:98:05), Dst: 42:aa:7a:50:9b:42 (4a:7a:50:9b:42:aa:7a:50:9b:42)

Internet Protocol Version 4, Src: 192.168.188.155, Dst: 77.234.45.64

Transmission Control Protocol, Src Port: 60902, Dst Port: 80, Seq: 1371, Ack: 1, Len: 564

[2 Reassembly TCP Segments (1370, 4425/564)]

Hypertext Transfer Protocol

Data (1593 bytes)

```

0000  42 aa 7a 50 9b 38 d5 7a 77 98 05 08 00 45 00  B3: zp-8: zw...E
0001  02 5c 3b c4 48 00 80 6c 69 c0 a8 bc 9d 4d 4b  4b: g...i...H
0002  2d 48 ea 79 08 01 5b 04 08 7d 58 29 a2 50 18  4b: y...f...XJ) P
0003  01 00 9b 94 0a 00 fe 13 28 46 15 bc 70 fc 11  0000: XXXF...
0004  9c b1 6d 61 95 40 0f 70 98 2a 8c 14 5a 1f 77  0000: ==a CBP...LZ w
0005  4a 1c 97 28 64 e2 24 9c cf 66 d3 88 75 d3 a4  0000: (d $...f hu
0006  ed 75 ac 28 e9 74 6e d4 24 27 69 58 29 4d 00  0000: u ( t n $...KD
0007  fb e2 fe 25 b8 9f 96 6e 0a e0 d4 13 b8 a3 00  0000: ~+g...g
0008  5c b2 fc d4 5d 47 8d 87 d4 1d 38 b7 d1 54 95  0000: \...jg...-8.T
0009  81 74 11 79 da 34 35 63 a9 80 8c 4b 39 32 3a  0000: t...y3...-K.r:
000a  16 dc af ba 66 c7 90 8d 73 b1 a4 12 9b 11 41  0000: 1...f...s...A
000b  c0 60 43 2b 15 a8 09 93 02 7f 00 57 ca 59 dc  0000: C...f...H.V.R
000c  33 17 42 eb f7 00 c6 e8 1f 01 35 25 23 3e 9b  0000: 3...f...-5 #.5
000d 
```

3. Based on the above activities, explain the working of the HTTP protocol [HTTP GET and POST message] briefly with typed answers and answer highlighted screenshots for the above activity from Wireshark.

Working:

- The browser sends an HTTP GET request to the web server hosting the URL you entered. This request asks the server to send back the HTML code for the webpage.
- The web server responds to the GET request by sending back the HTML code for the webpage.
- The browser renders the HTML code and displays the webpage. Depending on the webpage design, there may be a form or text box where you can enter your comment.
- When you enter your comment and click the "submit" button, the browser sends an HTTP POST request to the web server. This request contains the data you entered in the form.
- The web server receives the POST request and processes the data. Depending on how the web application is designed, the data may be stored in a database or sent via email to the website owner.

- The web server sends an HTTP response back to the browser. This response may be a confirmation message, a redirect to a different webpage, or an error message if there was a problem processing the request.

a. Analyze the sender and destination IP address, Port address, and Physical address (Proxy Server)

```

v Internet Protocol Version 4, Src: 10.113.11.158, Dst: 128.59.105.24
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0xb8 (DSCP: EF PHB, ECN: Not-ECT)
    Total Length: 597
    Identification: 0xe557 (58711)
  > 010. .... = Flags: 0x2, Don't fragment
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 128
    Protocol: TCP (6)
    Header Checksum: 0x1331 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 10.113.11.158
    Destination Address: 128.59.105.24
v Transmission Control Protocol, Src Port: 57691, Dst Port: 80, Seq: 1, A
  Source Port: 57691
  Destination Port: 80
  [Stream index: 14]
  [Conversation completeness: Incomplete (28)]
  [TCP Segment Len: 557]
  Sequence Number: 1 (relative sequence number)
  Sequence Number (raw): 2530421891
  [Next Sequence Number: 558 (relative sequence number)]
  Acknowledgment Number: 1 (relative ack number)
  Acknowledgment number (raw): 1737654594
  0101 .... = Header Length: 20 bytes (5)

```

b. Analyze the Host machine and webpage name in the file server.

```

> Ethernet II, Src: AzureWav_fa:26:b1 (ec:2e:98:fa:26:b1), Dst: Fortine
> Internet Protocol Version 4, Src: 10.113.11.158, Dst: 128.59.105.24
> Transmission Control Protocol, Src Port: 57691, Dst Port: 80, Seq: 1,
v Hypertext Transfer Protocol
  > GET /~fdc/sample.html HTTP/1.1\r\n
    Host: www.columbia.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/
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,imag
    Referer: https://www.google.com/\r\n
    Accept-Encoding: gzip, deflate\r\n
    Accept-Language: en-US,en;q=0.9\r\n
    If-Modified-Since: Fri, 17 Sep 2021 19:26:14 GMT\r\n
    \r\n
    [Full request URI: http://www.columbia.edu/~fdc/sample.html]
    [HTTP request 1/13]
    [Response in frame: 306]
    [Next request in frame: 868]

```

c. Check the acknowledgment frame for successful request and the unsuccessful request.

```

Internet Protocol Version 4, Src: 10.11.11.100, Dst: 128.39.129.24
Transmission Control Protocol, Src Port: 57691, Dst Port: 80, Seq: 1,
  Source Port: 57691
  Destination Port: 80
  [Stream index: 14]
  [Conversation completeness: Incomplete (28)]
  [TCP Segment Len: 557]
  Sequence Number: 1    (relative sequence number)
  Sequence Number (raw): 2530421891
  [Next Sequence Number: 558    (relative sequence number)]
  Acknowledgment Number: 1    (relative ack number)
  Acknowledgment number (raw): 1737654594
  0101 .... = Header Length: 20 bytes (5)

```

e. Find the following from frames received for the above activity.

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

Browser and server are running in HTTP version 1.1

```

Hypertext Transfer Protocol
  GET /cu/computinghistory/ibm-manuals-03-160.jpg HTTP/1.1\r\n
    [Expert Info (Chat/Sequence): GET /cu/computinghistory/ibm-manu
      [GET /cu/computinghistory/ibm-manuals-03-160.jpg HTTP/1.1\r\n
      [Severity level: Chat]
      [Group: Sequence]
      Request Method: GET
      Request URI: /cu/computinghistory/ibm-manuals-03-160.jpg
      Request Version: HTTP/1.1
      Host: www.columbia.edu\r\n
      Connection: keep-alive\r\n

```

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

Accepted Languages : en-GB, en-US

```

Hypertext Transfer Protocol
  GET /cu/computinghistory/ibm-manuals-03-160.jpg HTTP/1.1\r\n
    Host: www.columbia.edu\r\n
    Connection: keep-alive\r\n
    User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/53
    Accept: image/avif,image/webp,image/apng,image/svg+xml,image/*,*/*;q
    Referer: http://www.columbia.edu/cu/computinghistory/\r\n
    Accept-Encoding: gzip, deflate\r\n
    Accept-Language: en-US,en;q=0.9\r\n
    Cookie: BIGipServer~CUIT~www.columbia.edu-80-pool=!eNSD/xqucDRCTaGxk
\r\n
    [Full request URI: http://www.columbia.edu/cu/computinghistory/ibm-m
    [HTTP request 1/7]
    [Response in frame: 1407]
    [Next request in frame: 1409]

```

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

Status code returned from the server to browser is 200.

```
Hypertext Transfer Protocol
  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
```

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

```
Hypertext Transfer Protocol
  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
    Date: Sat, 25 Nov 2023 07:11:37 GMT\r\n
    Server: Apache\r\n
    Last-Modified: Sun, 27 Feb 2022 11:59:50 GMT\r\n
    Accept-Ranges: bytes\r\n
    Vary: Accept-Encoding,User-Agent\r\n
    Content-Encoding: gzip\r\n
  > Content-Length: 2479\r\n
    Keep-Alive: timeout=15, max=95\r\n
    Connection: Keep-Alive\r\n
    Content-Type: text/css\r\n
    Set-Cookie: BIGipServer~CUIT~www.columbia.edu-80-pool=!mCVNMThVp\r\n
    [HTTP response 4/13]
```

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

Capture-Length = 107 bytes

```
Frame 1365: 107 bytes on wire (856 bits), 107 bytes captured (856 bit
  Section number: 1
  > Interface id: 0 (\Device\NPF_{9E7F194A-99E6-4917-8027-8738F92DD67B}
    Encapsulation type: Ethernet (1)
    Arrival Time: Nov 25, 2023 12:41:37.478880000 India Standard Time
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1700896297.478880000 seconds
    [Time delta from previous captured frame: 0.000000000 seconds]
    [Time delta from previous displayed frame: 0.394641000 seconds]
    [Time since reference or first frame: 39.291314000 seconds]
    Frame Number: 1365
    Frame Length: 107 bytes (856 bits)
    Capture Length: 107 bytes (856 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ethertype:ip:tcp:http:data-text-lines]
    [Coloring Rule Name: HTTP]
    [Coloring Rule String: http || tcp.port == 80 || http2]
```

vi. 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.

Ans. To view the TCP header in the packet content window, select a packet that contains TCP data, expand the "Transmission Control Protocol" section, and look for the "TCP Header" field. You can then view the raw data of the TCP header and any other headers that may be present within the packet.

vii. Can you tell whether your browser downloaded the two images serially, or whether they were downloaded from the two websites in parallel? Explain.

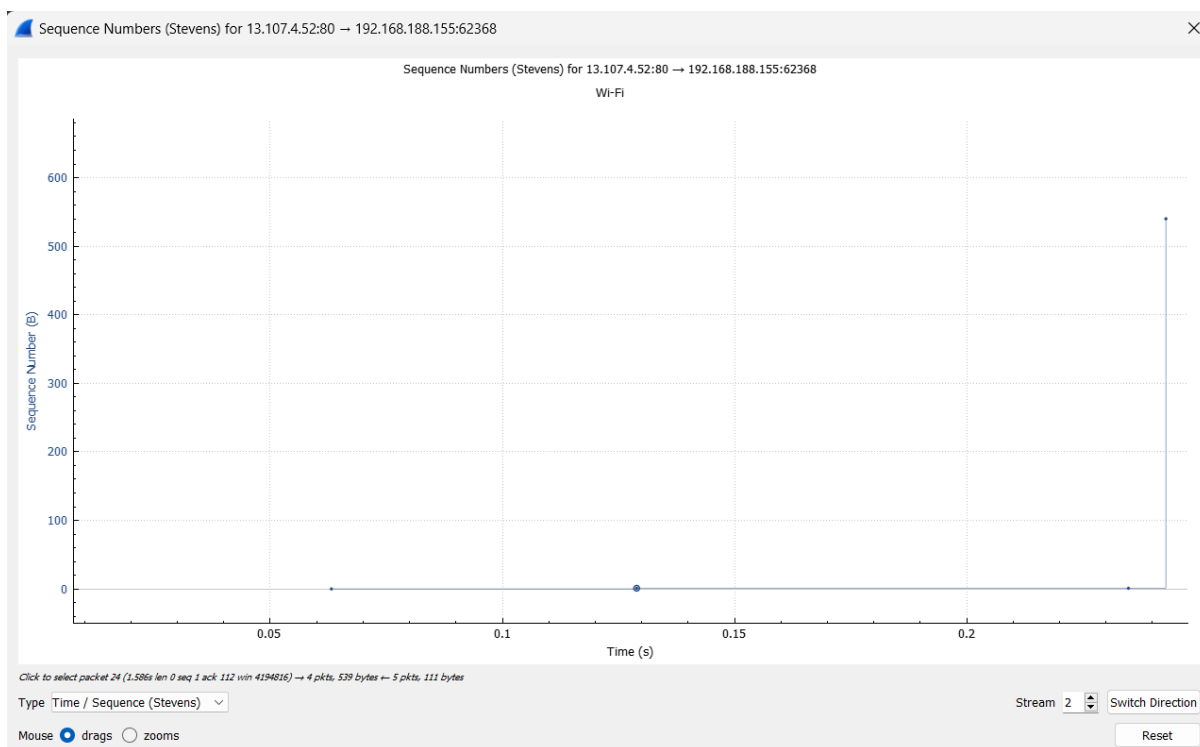
Ans. If the packets for both images are interspersed in the packet-listing window and have similar timestamps, it is likely that they were downloaded in parallel. This would suggest that the browser made separate HTTP requests for each image, and the responses were received and downloaded simultaneously. On the other hand, if the packets for one image are all downloaded before the packets for the second image, it is likely that the images were downloaded serially. This would suggest that the browser made a request for one image, received and downloaded the response, and then made a separate request for the second image.

To confirm this, you can also look at the sequence numbers of the TCP packets. If the sequence numbers of the packets for both images are similar, it suggests that they were downloaded in parallel, whereas if the sequence numbers for one image are significantly higher than the other, it suggests that the images were downloaded serially.

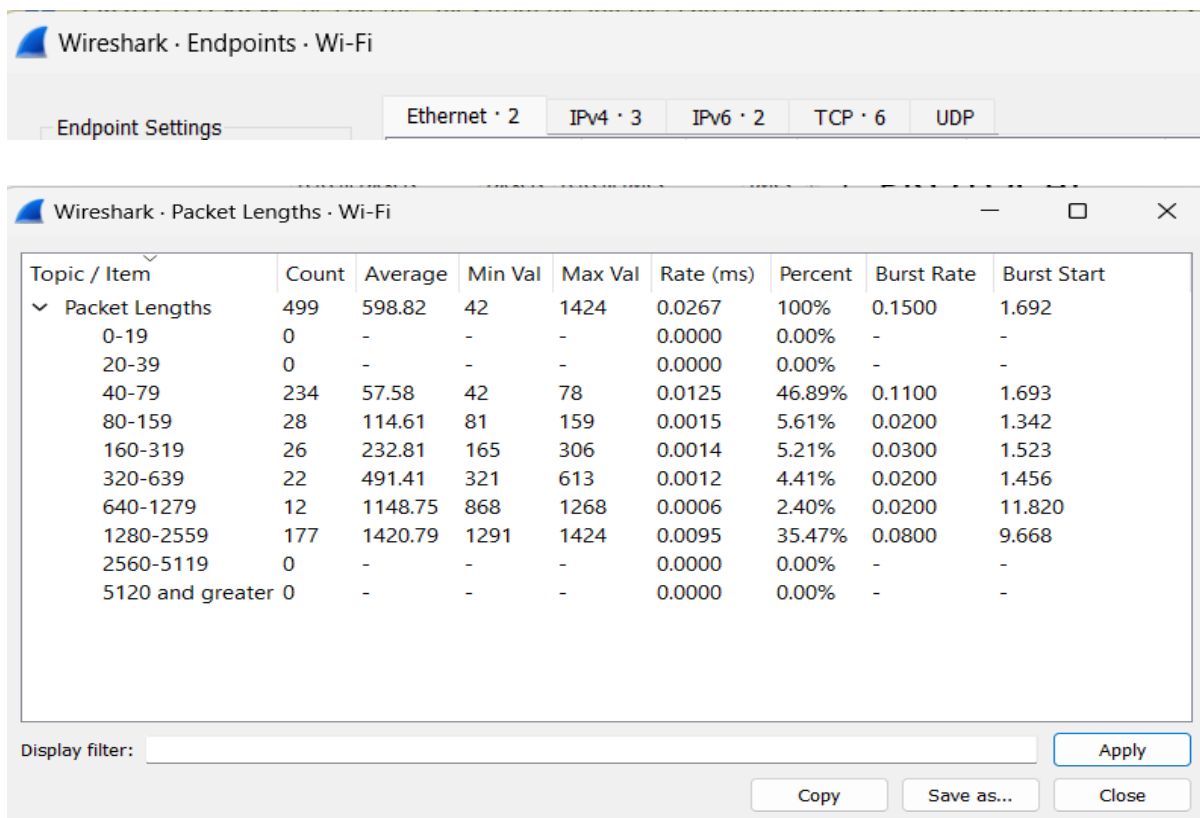
viii. What is the server's response (status code and phrase) in response to the initial HTTP GET message from your browser?

```
Hypertext Transfer Protocol
  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
```

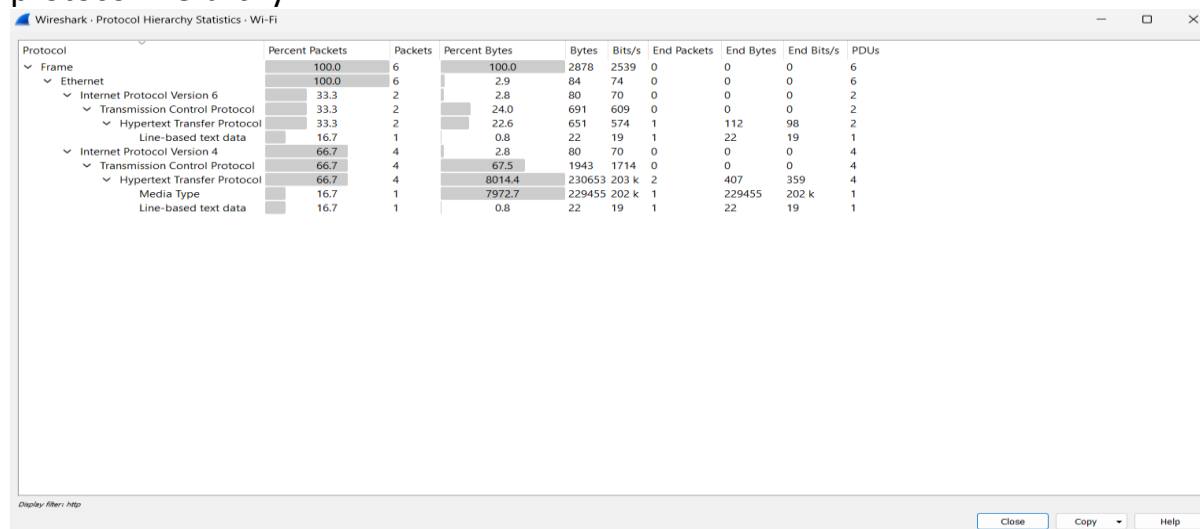
4. Open AUMS and download a file. Stop capture. Using statistics tools find the time taken for downloading the file.



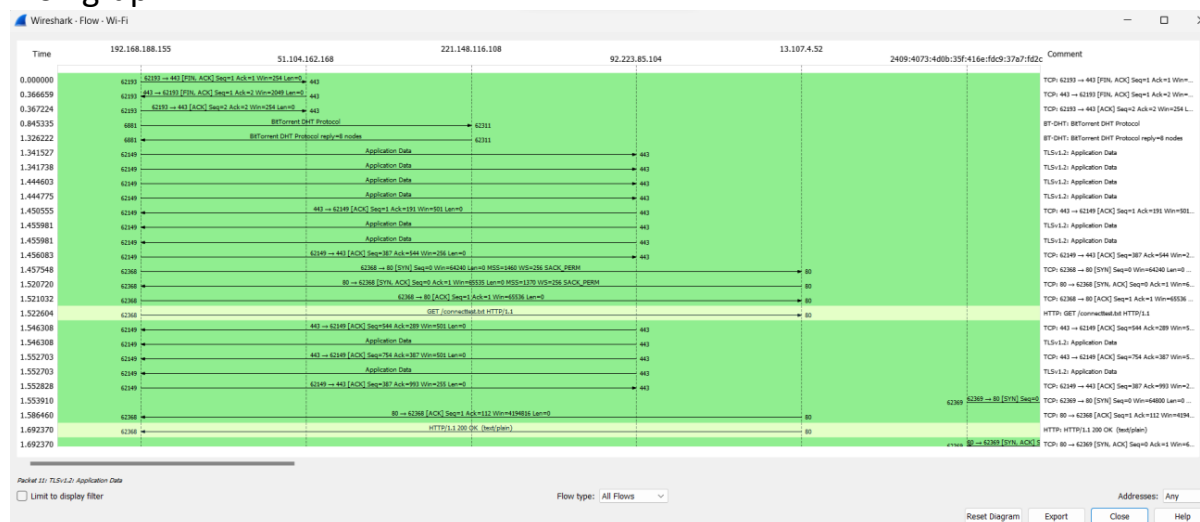
5. Open any SharePoint page and download a video file. Find the number of routing endpoints visited and the switches visited. Find the packet length transmitted and received, give the protocol hierarchy and the Flowgraph.



protocol hierarchy



Flowgraph.



6. Capture the frames for the following commands in the command prompt.
 print "GET /HTTP/1.0\r\n" | nc ac.amrita.ac.in 80

Start Up your web browser, and make sure your browser's cache is cleared. Do this activity and capture frames.

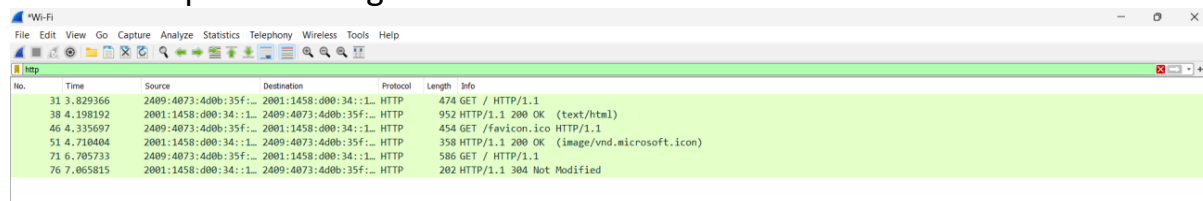
| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|---------------|----------------|----------------|----------|--------|-------------------------|
| 245 | 287.411997039 | 10.0.2.15 | 185.125.190.48 | HTTP | 141 | GET / HTTP/1.1 |
| 247 | 287.869821340 | 185.125.190.48 | 10.0.2.15 | HTTP | 243 | HTTP/1.1 204 No Content |
| 267 | 587.436593275 | 10.0.2.15 | 91.189.91.49 | HTTP | 141 | GET / HTTP/1.1 |
| 269 | 587.932213218 | 91.189.91.49 | 10.0.2.15 | HTTP | 243 | HTTP/1.1 204 No Content |
| 297 | 887.431082277 | 10.0.2.15 | 185.125.190.18 | HTTP | 141 | GET / HTTP/1.1 |
| 299 | 887.807218501 | 185.125.190.18 | 10.0.2.15 | HTTP | 243 | HTTP/1.1 204 No Content |

7. Open your web browser, and make sure your browser's cache is cleared. Do this activity and capture frames.

a. Enter the following URL into your browser "http://ac.amrita.ac.in"

b. Quickly enter the same URL into your browser again (or simply select the refresh button on your browser)

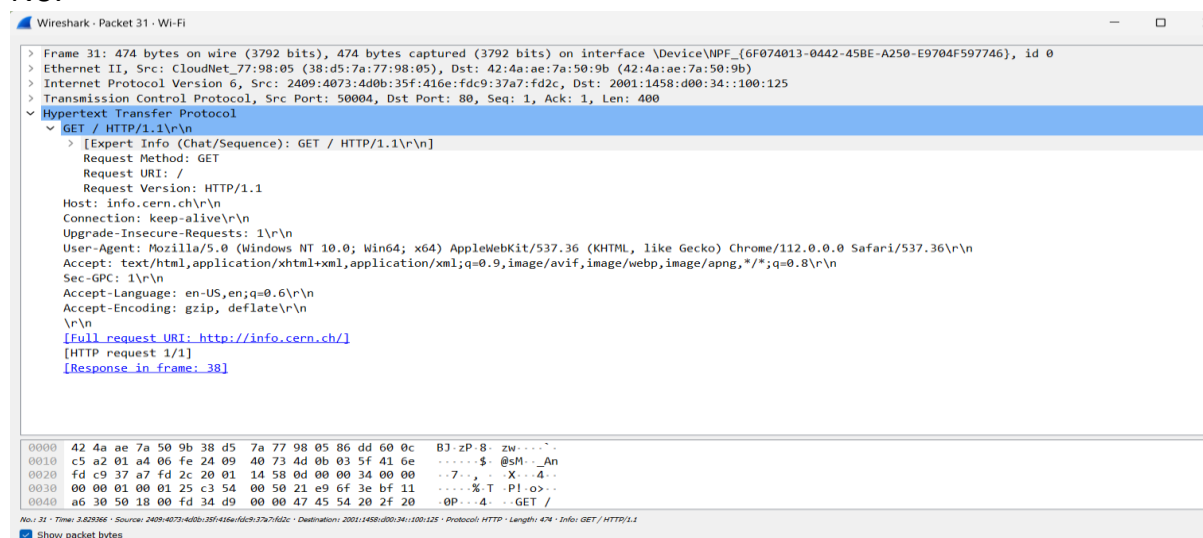
c. Stop Wireshark packet capture and enter “HTTP” in the display-filter-specification window, so that only captured HTTP messages will be displayed later in the packet listing window.



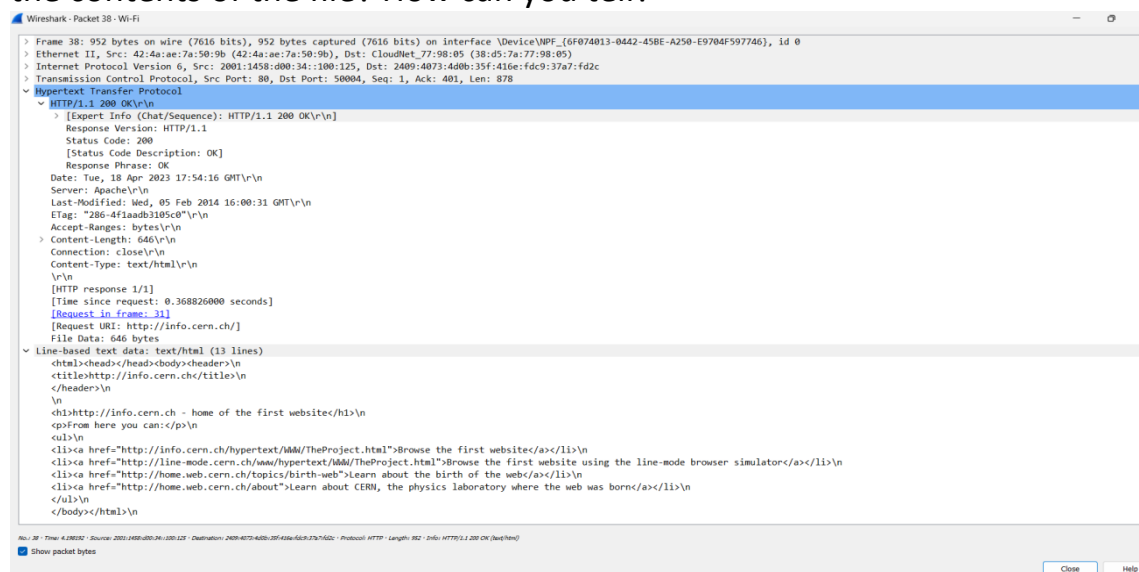
d. Answer the following questions:

i. 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 request?

No.

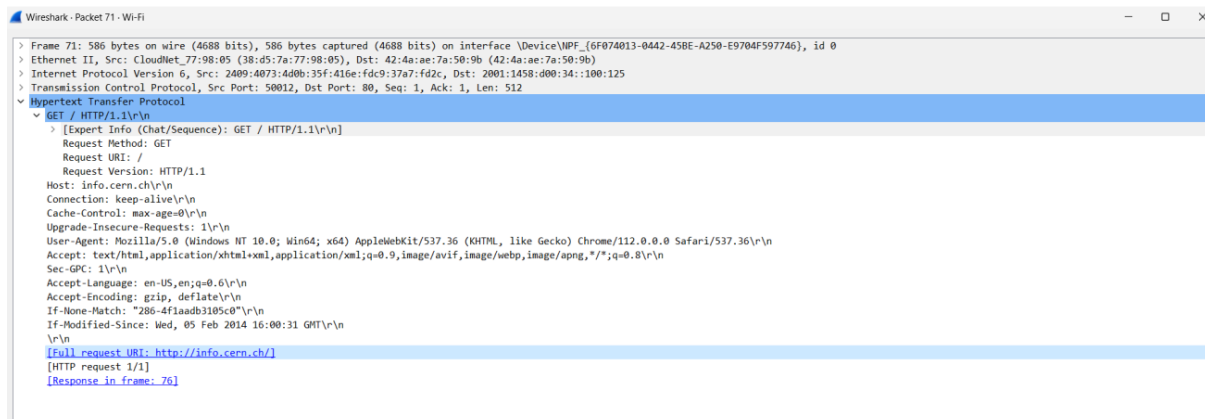


ii. Inspect the contents of the server response. Did the server explicitly return the contents of the file? How can you tell?

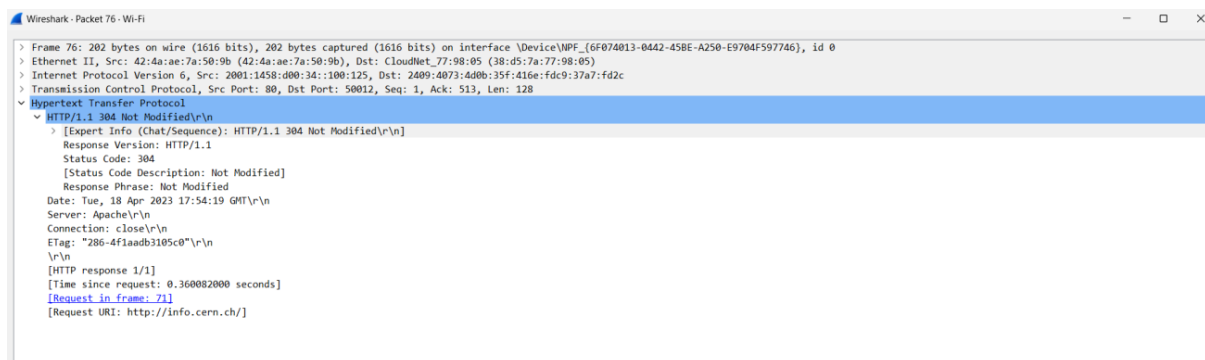


iii. Now inspect the contents of the second HTTP GET request from your browser to the server. Do you see an “IF-MODIFIED-SINCE:” line in the HTTP GET? If so, what information follows the “IF-MODIFIED-SINCE:” header?

YES.



iv. 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.



v. How many HTTP GET request messages were sent by your browser?

The number of HTTP GET request messages sent by the browser depends on the number of files requested. Here it is 3 HTTP GET request messages

vi. How many data-containing TCP segments were needed to carry the single HTTP response? What is the size for each of the segments?

The number of data-containing TCP segments needed to carry the single HTTP response depends on the size of the file and the maximum segment size (MSS)

of the TCP connection. The size of each segment will be less than or equal to the MSS.

vii. What is the status code and phrase associated with the response to the HTTP GET request?

The status code and phrase associated with the response to the HTTP GET request will be either HTTP 200 "OK" or HTTP 304 "Not Modified", depending on whether the file has been modified since the last time the browser requested it.

```
> Frame 38: 952 bytes on wire (7616 bits), 952 bytes captured (7616 bits) on interface \Device\NPF_{6F074013-0442-45BE-A250-E9704F597746}, id 0
> Ethernet II, Src: 42:4a:ae:7a:50:9b (42:4a:ae:7a:50:9b), Dst: CloudNet_77:98:05 (38:d5:7a:77:98:05)
> Internet Protocol Version 6, Src: 2001:1458:d00:34::100:125, Dst: 2409:4073:4d0b:35f:416e:fdc9:37a7:fd2c
> Transmission Control Protocol, Src Port: 80, Dst Port: 50004, Seq: 1, Ack: 401, Len: 878
< Hypertext Transfer Protocol
  < 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

> Frame 76: 202 bytes on wire (1616 bits), 202 bytes captured (1616 bits) on interface \Device\NPF_{6F074013-0442-45BE-A250-E9704F597746}, id 0
> Ethernet II, Src: 42:4a:ae:7a:50:9b (42:4a:ae:7a:50:9b), Dst: CloudNet_77:98:05 (38:d5:7a:77:98:05)
> Internet Protocol Version 6, Src: 2001:1458:d00:34::100:125, Dst: 2409:4073:4d0b:35f:416e:fdc9:37a7:fd2c
> Transmission Control Protocol, Src Port: 80, Dst Port: 50012, Seq: 1, Ack: 513, Len: 128
< Hypertext Transfer Protocol
  < HTTP/1.1 304 Not Modified\r\n
    > [Expert Info (Chat/Sequence): HTTP/1.1 304 Not Modified\r\n]
      Response Version: HTTP/1.1
      Status Code: 304
      [Status Code Description: Not Modified]
      Response Phrase: Not Modified
```