ASSIGNMENT-1

COMPUTER NETWORKS (Group 7)

By -
DEWANSH SINGH CHANDEL (22110072)
PRANJAL GAUR (22110201)

Github Repo Link

Part 1: Metrics and Plots (40 pts) From the chosen X.pcap file, extract and generate the following metrics for the data as captured by your program when you perform the pcap replay using tools like tcpreplay:

(NOTE: PCAP file used 7.pcap)

1. Find the total amount of data transferred (in bytes), the total number of packets transferred, and the minimum, maximum, and average packet sizes. Also, show the distribution of packet sizes (e.g., by plotting a histogram of packet sizes).

- Total Packets transferred: **246,519**
- Total Data Transferred: **134,996,148 bytes**
- Minimum Packet Size: **42 bytes**
- Maximum Packet Size: **1514 bytes**
- Average Packet Size: **547 bytes**

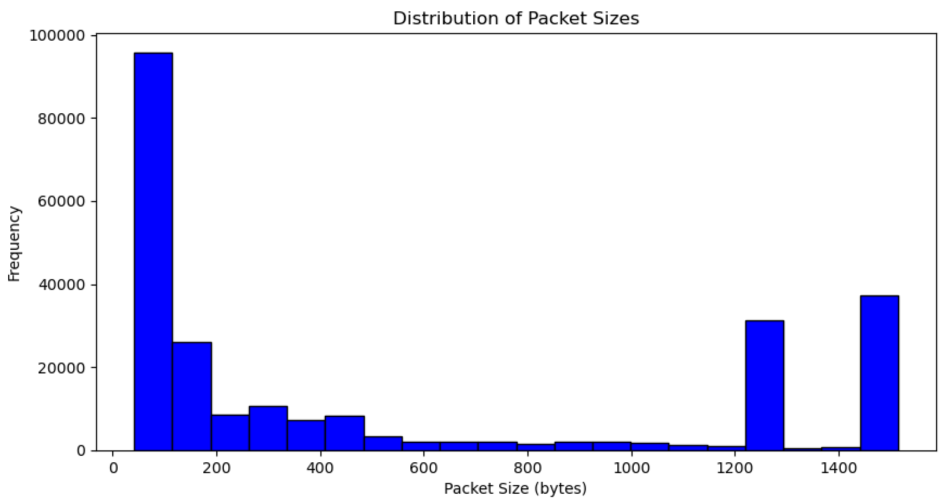
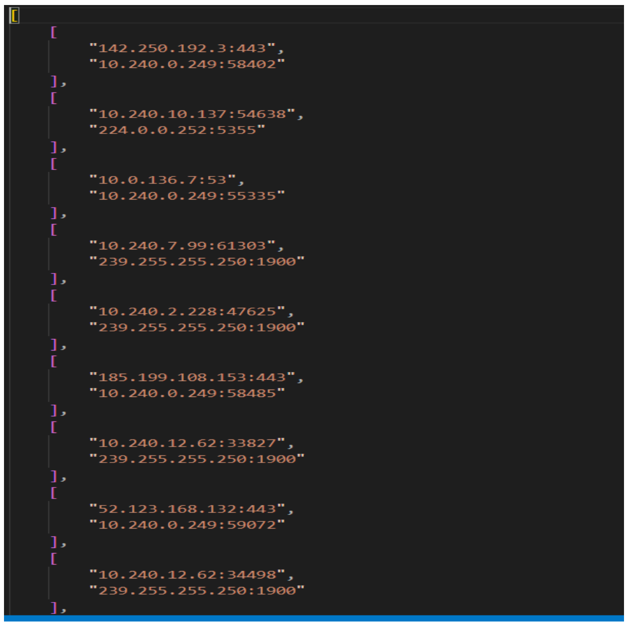


Fig 1. Packet Size Distribution

2. Find unique source-destination pairs (source IP:port and destination IP:port) in the captured data.

===== Unique Source-Destination Pairs =====: 9603

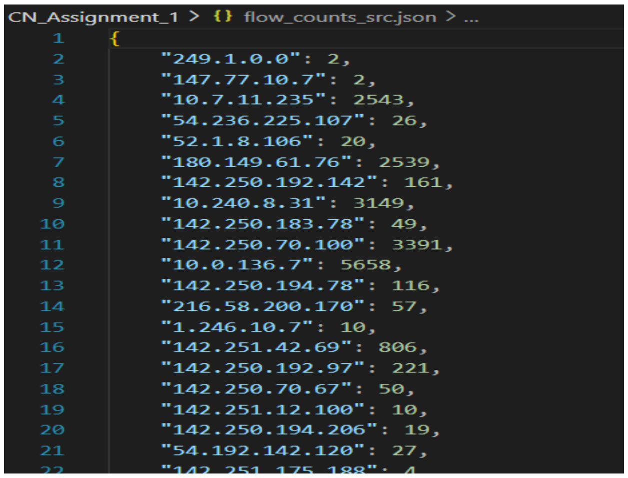


```
[
    [
        "142.250.192.3:443",
        "10.240.0.249:58402"
    ],
    [
        "10.240.10.137:54638",
        "224.0.0.252:5355"
    ],
    [
        "10.0.136.7:53",
        "10.240.0.249:55335"
    ],
    [
        "10.240.7.99:61303",
        "239.255.255.250:1900"
    ],
    [
        "10.240.2.228:47625",
        "239.255.255.250:1900"
    ],
    [
        "185.199.108.153:443",
        "10.240.0.249:58485"
    ],
    [
        "10.240.12.62:33827",
        "239.255.255.250:1900"
    ],
    [
        "52.123.168.132:443",
        "10.240.0.249:59072"
    ],
    [
        "10.240.12.62:34498",
        "239.255.255.250:1900"
    ],
```

Fig2. Snapshot of Unique Souce- Destination pair

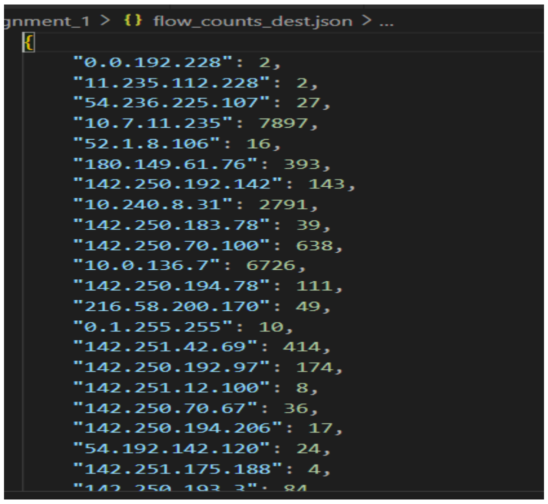
(Complete JSON file is in the GitHub repo named unique_pairs.json)

3. Display a dictionary where the key is the IP address and the value is the total flows for that IP address as the source. Similarly display a dictionary where the key is the IP address and the value is the total flows for that IP address as the destination. Find out which source-destination (source IP:port and destination IP:port) have transferred the most data



```
CN_Assignment_1 > {} flow_counts_src.json > ...
{
    "249.1.0.0": 2,
    "147.77.10.7": 2,
    "10.7.11.235": 2543,
    "54.236.225.107": 26,
    "52.1.8.106": 20,
    "180.149.61.76": 2539,
    "142.250.192.142": 161,
    "10.240.8.31": 3149,
    "142.250.183.78": 49,
    "142.250.70.100": 3391,
    "10.0.136.7": 5658,
    "142.250.194.78": 116,
    "216.58.200.170": 57,
    "1.246.10.7": 10,
    "142.251.42.69": 806,
    "142.250.192.97": 221,
    "142.250.70.67": 50,
    "142.251.12.100": 10,
    "142.250.194.206": 19,
    "54.192.142.120": 27,
    "142.251.175.188": 4
```

Fig 3. Snapshot of Flowcounts for source IP
(Complete JSON file is in GitHub repo named flow_count_scr.json)



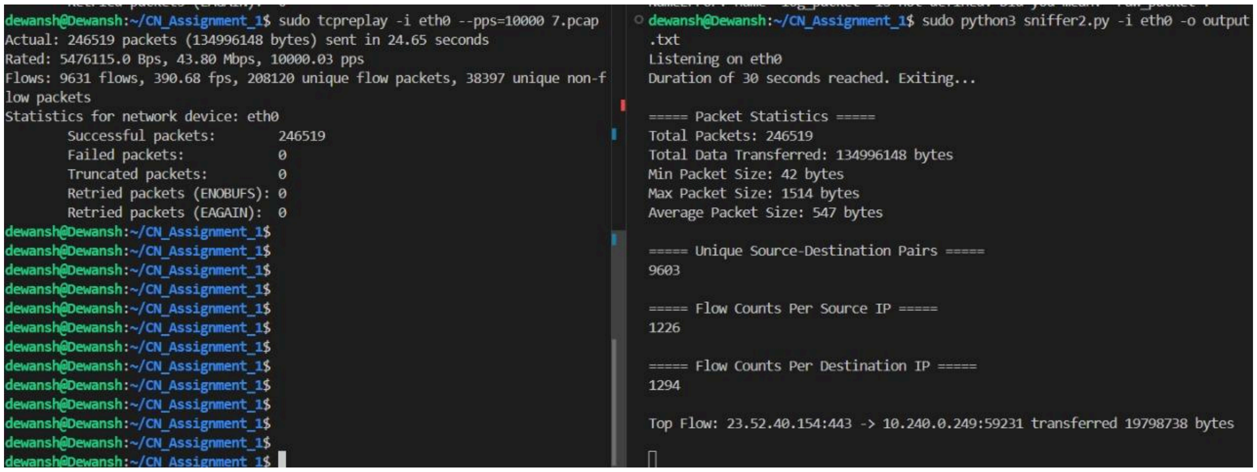
```
gnment_1 > {} flow_counts_dest.json > ...
{
    "0.0.192.228": 2,
    "11.235.112.228": 2,
    "54.236.225.107": 27,
    "10.7.11.235": 7897,
    "52.1.8.106": 16,
    "180.149.61.76": 393,
    "142.250.192.142": 143,
    "10.240.8.31": 2791,
    "142.250.183.78": 39,
    "142.250.70.100": 638,
    "10.0.136.7": 6726,
    "142.250.194.78": 111,
    "216.58.200.170": 49,
    "0.1.255.255": 10,
    "142.251.42.69": 414,
    "142.250.192.97": 174,
    "142.251.12.100": 8,
    "142.250.70.67": 36,
    "142.250.194.206": 17,
    "54.192.142.120": 24,
    "142.251.175.188": 4,
    "142.250.193.3": 84
```

Fig 4. Snapshot of Flowcounts for destination IP
(Complete JSON file is in GitHub repo named flow_count_dest.json)

Top Flow: 23.52.40.154:443 -> 10.240.0.249:59231 transferred 19798738 bytes

4. List the top speed in terms of `pps` and `mbps` that your program is able to capture the content without any loss of data when i) running both tcpreplay and your program on the same machine (VM), and ii) when running on different machines: Two student group should run the program on two different machines eg. tcpreplay on physical-machine of student1 and sniffer program physical-machine of student2. Single students should run between two VMs.

I) ON SAME Machine



```
dewansh@Dewansh:~/CN_Assignment_1$ sudo tcpreplay -i eth0 --pps=10000 7.pcap
Actual: 246519 packets (134996148 bytes) sent in 24.65 seconds
Rated: 5476115.0 Bps, 43.80 Mbps, 10000.03 pps
Flows: 9631 flows, 390.68 fps, 208120 unique flow packets, 38397 unique non-f
low packets
Statistics for network device: eth0
        Successful packets:        246519
        Failed packets:            0
        Truncated packets:         0
        Retried packets (ENOBUFS): 0
        Retried packets (EAGAIN):  0
dewansh@Dewansh:~/CN_Assignment_1$
dewansh@Dewansh:~/CN_Assignment_1$
dewansh@Dewansh:~/CN_Assignment_1$
dewansh@Dewansh:~/CN_Assignment_1$
dewansh@Dewansh:~/CN_Assignment_1$
dewansh@Dewansh:~/CN_Assignment_1$
dewansh@Dewansh:~/CN_Assignment_1$
dewansh@Dewansh:~/CN_Assignment_1$
dewansh@Dewansh:~/CN_Assignment_1$
dewansh@Dewansh:~/CN_Assignment_1$
dewansh@Dewansh:~/CN_Assignment_1$
dewansh@Dewansh:~/CN_Assignment_1$
dewansh@Dewansh:~/CN_Assignment_1$
```

```
dewansh@Dewansh:~/CN_Assignment_1$ sudo python3 sniffer2.py -i eth0 -o output
.txt
Listening on eth0
Duration of 30 seconds reached. Exiting...

===== Packet Statistics =====
Total Packets: 246519
Total Data Transferred: 134996148 bytes
Min Packet Size: 42 bytes
Max Packet Size: 1514 bytes
Average Packet Size: 547 bytes

===== Unique Source-Destination Pairs =====
9603

===== Flow Counts Per Source IP =====
1226

===== Flow Counts Per Destination IP =====
1294

Top Flow: 23.52.40.154:443 -> 10.240.0.249:59231 transferred 19798738 bytes
```

Fig 5. Snapshot of Terminal showing the maximum speed attained by the WSL eth0 port without any packet loss is 1000 packets per second (pps) or 43.80 Mbps

Our Observations:

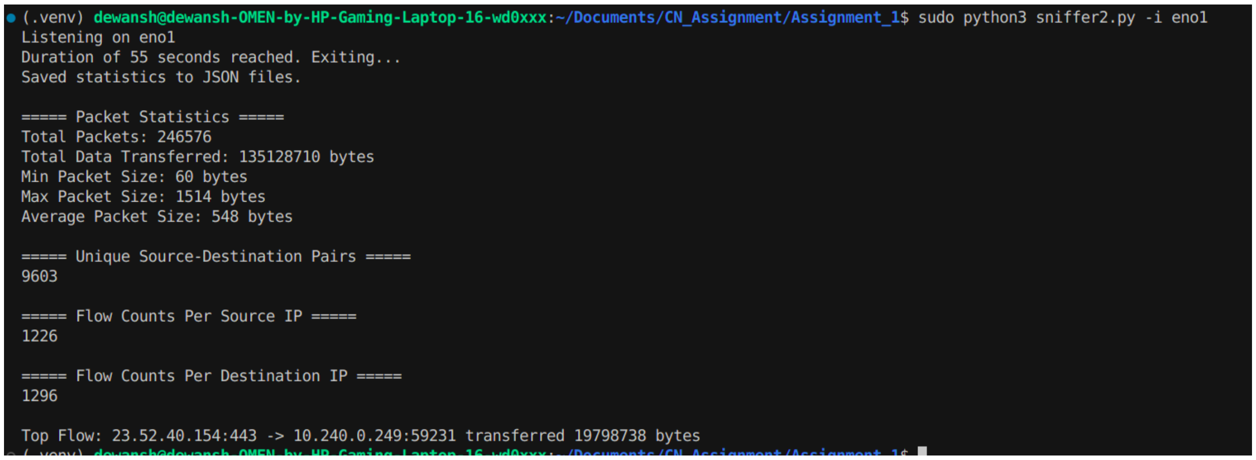
We have tried replaying on the network at speeds 12500,15000, 17500, and 20000 pps but always received packets which were lesser than the packets transmitted in the eth0 network interface.

Also, on replaying the .pcap file on speed 5000, 7500 pps we observed that sometimes the packets received were the same, and sometimes the packets received were greater than the original number of packets.

On replaying the network packets using tcpreplay 10 times with a speed of 10000 pps, we found the exact number of packets that were transmitted by tcpreplay.

II) ON DIFFERENT machine

Using two different machines connected by the RJ45 port using CAT-6 cable, the networks packet transferred were **246519** using **enp1s0** network interface in one system to **eno1** port of my system at the speed of **5000pps**. The network packet summary is shown below. However, the packets received were larger than the actual packets transferred (one reason can be due to the intercommunication between two systems, which results in transmission of some packets)



```
(.venv) dewansh@dewansh-OMEN-by-HP-Gaming-Laptop-16-wd0xxx:~/Documents/CN_Assignment/Assignment_1$ sudo python3 sniffer2.py -i eno1
Listening on eno1
Duration of 55 seconds reached. Exiting...
Saved statistics to JSON files.

===== Packet Statistics =====
Total Packets: 246576
Total Data Transferred: 135128710 bytes
Min Packet Size: 60 bytes
Max Packet Size: 1514 bytes
Average Packet Size: 548 bytes

===== Unique Source-Destination Pairs =====
9603

===== Flow Counts Per Source IP =====
1226

===== Flow Counts Per Destination IP =====
1296

Top Flow: 23.52.40.154:443 -> 10.240.0.249:59231 transferred 19798738 bytes
(.venv) dewansh@dewansh-OMEN-by-HP-Gaming-Laptop-16-wd0xxx:~/Documents/CN_Assignment/Assignment_1$
```

Snapshot of packets transmitted from one device to another

Network packet statistics were **almost similar** with the previous results achieved on single system transfer (significant change being in **Min packet size shifted from 42 bytes to 60 bytes**)

Our Observations:

We have tried replaying on the network at speeds 7500, 10000, 12500 and 15000 pps but always received packets that were lesser than the packets transmitted.

Also, on replaying the .pcap file on speed 4000, 5000 pps we observed that sometimes the packets received were the same, and sometimes the packets received were greater than the original number of packets.

On replaying the network packets using tcpreplay 10 times with a speed of 5000 pps, we found the exact number of packets that were transmitted by tcpreplay.

Part 2: Catch Me If You Can (40 points)

Q1: TCP Packet with ACK & PSH set, sum of ports = 60303

Count: 0

Q2: SYN Set, Source Port % 11 == 0, Sequence Number > 100000

Count: 223

Source: 10.7.11.235:53669 -> Destination: 10.240.8.31:8009
Source: 10.7.11.235:53669 -> Destination: 10.240.8.31:8009
Source: 10.7.11.235:53669 -> Destination: 10.240.8.31:8009
Source: 10.7.11.235:53669 -> Destination: 10.240.8.31:8009
Source: 10.7.11.235:53680 -> Destination: 3.111.224.186:443
Source: 10.7.11.235:53691 -> Destination: 142.250.199.170:443
Source: 10.240.0.249:55968 -> Destination: 10.0.136.7:53
....

(All source and destination IP address are in the part_2_answers.txt file in the github repo)

Q3: Source IP 18.234.xx.xxx, Prime Src Port, Dest Port % 11 == 0

Count: 11

Source: 18.234.0.179:443 -> Destination: 10.7.11.235:53251
Source: 18.234.0.179:443 -> Destination: 10.7.11.235:53251
Source: 18.234.0.179:443 -> Destination: 10.7.11.235:53251
Source: 18.234.0.179:443 -> Destination: 10.7.11.235:53251
Source: 18.234.0.179:443 -> Destination: 10.7.11.235:53251
Source: 18.234.0.179:443 -> Destination: 10.7.11.235:53251
Source: 18.234.0.179:443 -> Destination: 10.7.11.235:53251
Source: 18.234.0.179:443 -> Destination: 10.7.11.235:53251
Source: 18.234.0.179:443 -> Destination: 10.7.11.235:53251
Source: 18.234.0.179:443 -> Destination: 10.7.11.235:53251
Source: 18.234.0.179:443 -> Destination: 10.7.11.235:53251

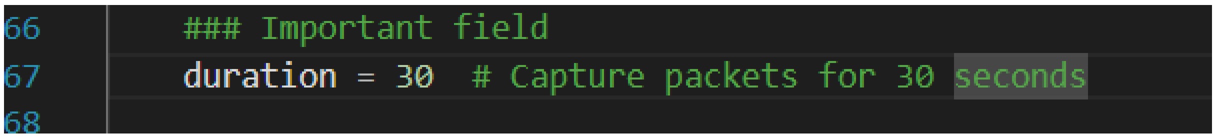
Q4: Sequence + Ack = 2512800625, Checksum ends in 70

Count: 1

Source: 10.240.8.31:8009 -> Destination: 10.7.11.235:53669

NOTE: These answers are generated by replaying the networks at 10000 pps using TCP replay and running the script part_2.py for duration of 30 seconds

(part_2.py file is provided in the github repository with network interface set as "eth0" provided by WSL)



```
66      ### Important field
67      duration = 30  # Capture packets for 30 seconds
68
```

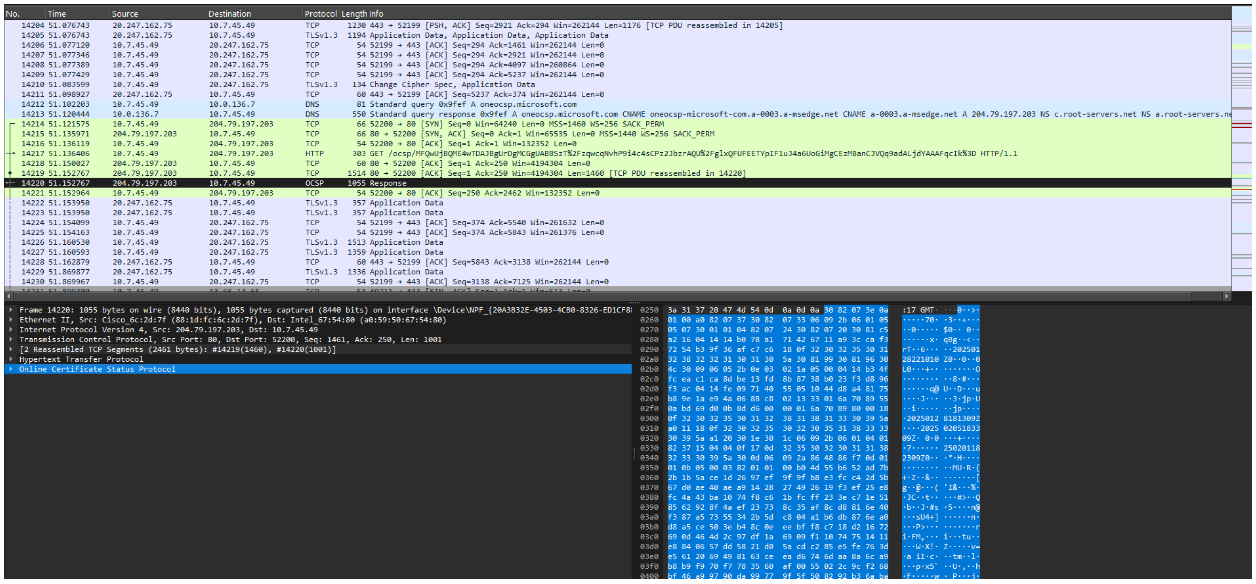
Fig 7. Snapshot of code from part_2.py showing the line number of duration variable for time adjustments

Part 3: Capture the packets (20 points)

Q1. Run the Wireshark tool and capture the trace of the network packets on your host device. We expect you would be connected to the Internet and perform regular network activities. a. List at-least 5 different application layer protocols that we have not discussed so far in the classroom and describe in 1-2 sentences the operation/usage of protocol and its layer of operation and indicate the associated RFC number if any.

Below are the five application layer protocols :

1. OCSP :
   OCSP (Online Certificate Status Protocol)
   Layer: Application Layer (Layer 7)
   Operation/Usage: OCSP provides real-time verification of digital certificate status (valid, revoked, or unknown) to ensure the validity of SSL/TLS certificates.
   RFC: RFC 6960

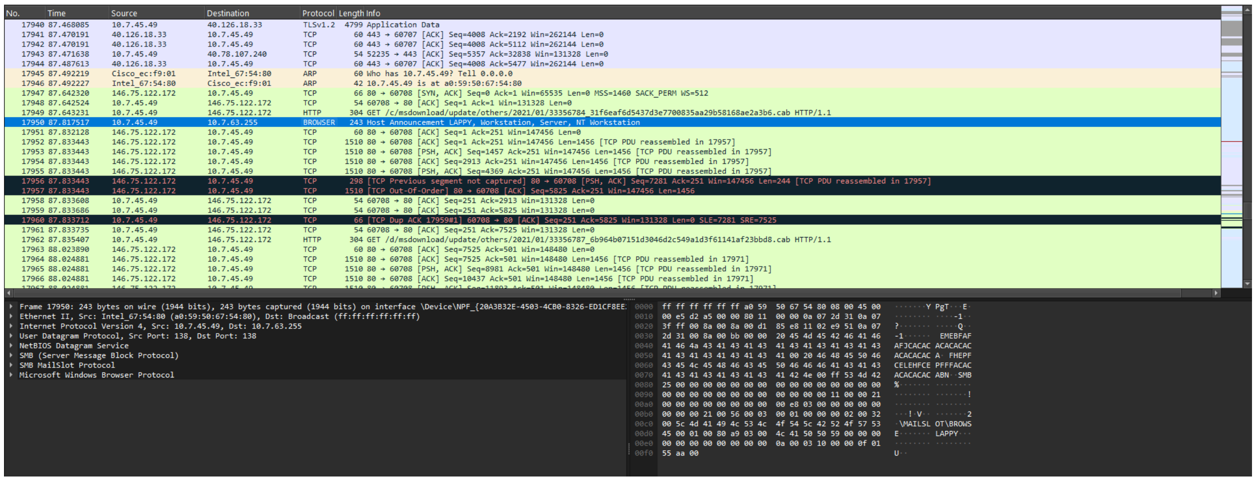


```
No.   Time      Source          Destination     Protocol Length Info
14204 51.076743 20.247.162.75   10.7.45.49      TCP     1230 443 → 52199 [PSH, ACK] Seq=2921 Ack=294 Win=262144 Len=1176 [TCP PDU reassembled in 14205]
14205 51.076743 20.247.162.75   10.7.45.49      TLSv1.3 1194 Application Data, Application Data, Application Data
14206 51.077120 10.7.45.49      20.247.162.75   TCP       54 52199 → 443 [ACK] Seq=294 Ack=1461 Win=262144 Len=0
14207 51.077346 10.7.45.49      20.247.162.75   TCP       54 52199 → 443 [ACK] Seq=294 Ack=2921 Win=262144 Len=0
14208 51.077389 10.7.45.49      20.247.162.75   TCP       54 52199 → 443 [ACK] Seq=294 Ack=4097 Win=260864 Len=0
14209 51.077429 10.7.45.49      20.247.162.75   TCP       54 52199 → 443 [ACK] Seq=294 Ack=5237 Win=262144 Len=0
14210 51.083599 10.7.45.49      20.247.162.75   TLSv1.3  134 Change Cipher Spec, Application Data
14211 51.098927 20.247.162.75   10.7.45.49      TCP       60 443 → 52199 [ACK] Seq=5237 Ack=374 Win=262144 Len=0
14212 51.102203 10.7.45.49      10.0.136.7      DNS       81 Standard query 0x9fef A oneocsp.microsoft.com
14213 51.120444 10.0.136.7      10.7.45.49      DNS      550 Standard query response 0x9fef A oneocsp.microsoft.com CNAME oneocsp-microsoft-com.a-0003.a-msedge.net CNAME a-0003.a-msedge.net A 204.79.197.203 NS c.root-servers.net NS a.root-servers.ne
14214 51.121575 10.7.45.49      204.79.197.203  TCP       66 52200 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
14215 51.135971 204.79.197.203  10.7.45.49      TCP       66 80 → 52200 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1440 WS=256 SACK_PERM
14216 51.136119 10.7.45.49      204.79.197.203  TCP       54 52200 → 80 [ACK] Seq=1 Ack=1 Win=132352 Len=0
14217 51.136406 10.7.45.49      204.79.197.203  HTTP     303 GET /ocsp/MFQwUjBQME4wTDAJBgUrDgMCGgUABBSzT%2FzqwcqNvhP9i4c4sCPz2JbzrAQU%2FglxQFUFEETYpIF1uJ4a6UoGiMgCEzMBanCJVQq9adALjdYAAAFqcIk%3D HTTP/1.1
14218 51.150027 204.79.197.203  10.7.45.49      TCP       60 80 → 52200 [ACK] Seq=1 Ack=250 Win=4194304 Len=0
14219 51.152767 204.79.197.203  10.7.45.49      TCP     1514 80 → 52200 [ACK] Seq=1 Ack=250 Win=4194304 Len=1460 [TCP PDU reassembled in 14220]
14220 51.152767 204.79.197.203  10.7.45.49      OCSP    1055 Response
14221 51.152964 10.7.45.49      204.79.197.203  TCP       54 52200 → 80 [ACK] Seq=250 Ack=2462 Win=132352 Len=0
14222 51.153950 20.247.162.75   10.7.45.49      TLSv1.3  357 Application Data
14223 51.153950 20.247.162.75   10.7.45.49      TLSv1.3  357 Application Data
14224 51.154099 10.7.45.49      20.247.162.75   TCP       54 52199 → 443 [ACK] Seq=374 Ack=5540 Win=261632 Len=0
14225 51.154163 10.7.45.49      20.247.162.75   TCP       54 52199 → 443 [ACK] Seq=374 Ack=5843 Win=261376 Len=0
14226 51.160530 10.7.45.49      20.247.162.75   TLSv1.3 1513 Application Data
14227 51.160593 10.7.45.49      20.247.162.75   TLSv1.3 1359 Application Data
14228 51.162879 20.247.162.75   10.7.45.49      TCP       60 443 → 52199 [ACK] Seq=5843 Ack=3138 Win=262144 Len=0
14229 51.869877 20.247.162.75   10.7.45.49      TLSv1.3 1336 Application Data
14230 51.869967 10.7.45.49      20.247.162.75   TCP       54 52199 → 443 [ACK] Seq=3138 Ack=7125 Win=262144 Len=0
```

```
Frame 14220: 1055 bytes on wire (8440 bits), 1055 bytes captured (8440 bits) on interface \Device\NPF_{20A3B32E-4503-4CB0-8326-ED1CF8
Ethernet II, Src: Cisco_6c:2d:7f (88:1d:fc:6c:2d:7f), Dst: Intel_67:54:80 (a0:59:50:67:54:80)
Internet Protocol Version 4, Src: 204.79.197.203, Dst: 10.7.45.49
Transmission Control Protocol, Src Port: 80, Dst Port: 52200, Seq: 1461, Ack: 250, Len: 1001
[2 Reassembled TCP Segments (2461 bytes): #14219(1460), #14220(1001)]
Hypertext Transfer Protocol
Online Certificate Status Protocol
```

```
0250  3a 31 37 20 47 4d 54 0d  0a 0d 0a 30 82 07 3e 0a   :17 GMT· ···0··>·
0260  01 00 a0 82 07 37 30 82  07 33 06 09 2b 06 01 05   ·····70· ·3··+···
0270  05 07 30 01 01 04 82 07  24 30 82 07 20 30 81 c5   ··0····· $0·· 0··
0280  a2 16 04 14 14 b0 78 a1  71 42 67 11 a9 3c ca f3   ······x· qBg··<··
0290  72 54 b3 9f 36 af c7 c6  18 0f 32 30 32 35 30 31   rT··6··· ··202501
02a0  32 38 32 32 31 30 31 30  5a 30 81 99 30 81 96 30   28221010 Z0··0··0
02b0  4c 30 09 06 05 2b 0e 03  02 1a 05 00 04 14 b3 4f   L0···+·· ·······O
02c0  fc ea c1 ca 8d be 13 fd  8b 87 38 b0 23 f3 d8 96   ········ ··8·#···
02d0  f3 ac 04 14 fe 09 71 40  55 05 10 44 d8 a4 81 75   ······q@ U··D···u
02e0  b8 9e 1a e9 4a 06 88 c8  02 13 33 01 6a 70 89 55   ····J··· ··3·jp·U
02f0  0a bd 69 d0 0b 8d d6 00  00 01 6a 70 89 80 00 18   ··i····· ··jp····
0300  0f 32 30 32 35 30 31 32  38 31 38 31 33 30 39 5a   ·2025012 8181309Z
0310  a0 11 18 0f 32 30 32 35  30 32 30 35 31 38 33 33   ····2025 02051833
0320  30 39 5a a1 20 30 1e 30  1c 06 09 2b 06 01 04 01   09Z· 0·0 ···+····
0330  82 37 15 04 04 0f 17 0d  32 35 30 32 30 31 31 38   ·7······ 25020118
0340  32 33 30 39 5a 30 0d 06  09 2a 86 48 86 f7 0d 01   2309Z0·· ·*·H····
0350  01 0b 05 00 03 82 01 01  00 b0 4d 55 b6 52 ad 7b   ········ ··MU·R·{
0360  2b 1b 5a ce 1d 26 97 ef  9f 9f b8 e3 fc c4 2d 5b   +·Z··&·· ·······-[
0370  67 d0 ae 40 ae a9 14 28  27 49 26 19 f3 ef 25 e8   g··@···( 'I&···%·
0380  fc 4a 43 ba 10 74 f8 c6  1b fc ff 23 3e c7 1e 51   ·JC··t·· ···#>··Q
0390  85 62 92 8f 4a ef 23 73  8c 35 af 8c d8 81 6e 40   ·b··J·#s ·5····n@
03a0  f3 87 a5 73 55 34 2b 5d  c8 04 a1 b6 db 87 6e a0   ···sU4+] ······n·
03b0  d8 a5 ce 50 3e b4 8c 0e  ee bf f8 c7 18 d2 16 72   ···P>··· ·······r
03c0  69 0d 46 4d 2c 97 df 1a  69 09 f1 10 74 75 14 11   i·FM,··· i···tu··
03d0  e8 84 06 57 dd 58 21 d0  5a cd c2 85 e5 fe 76 3d   ···W·X!· Z·····v=
03e0  e5 61 20 69 49 81 63 ce  ea d6 74 6d aa 8a 6c a9   ·a iI·c· ··tm··l·
03f0  b8 b9 f9 70 f7 78 35 60  af 00 55 02 2c 9c f2 68   ···p·x5` ··U·,··h
0400  bf 46 a9 97 90 da 99 77  9f 5f 50 82 92 b3 6a ba   ·F·····w · P···j·
```

2. **BROWSER**
   BROWSER (Windows Browser Protocol)
   Layer: Application Layer (Layer 7)
   Operation/Usage: The Windows Browser Protocol allows devices in a local network to discover shared resources like files and printers, particularly in older Windows environments.
   RFC: RFC 1001 and RFC 1002

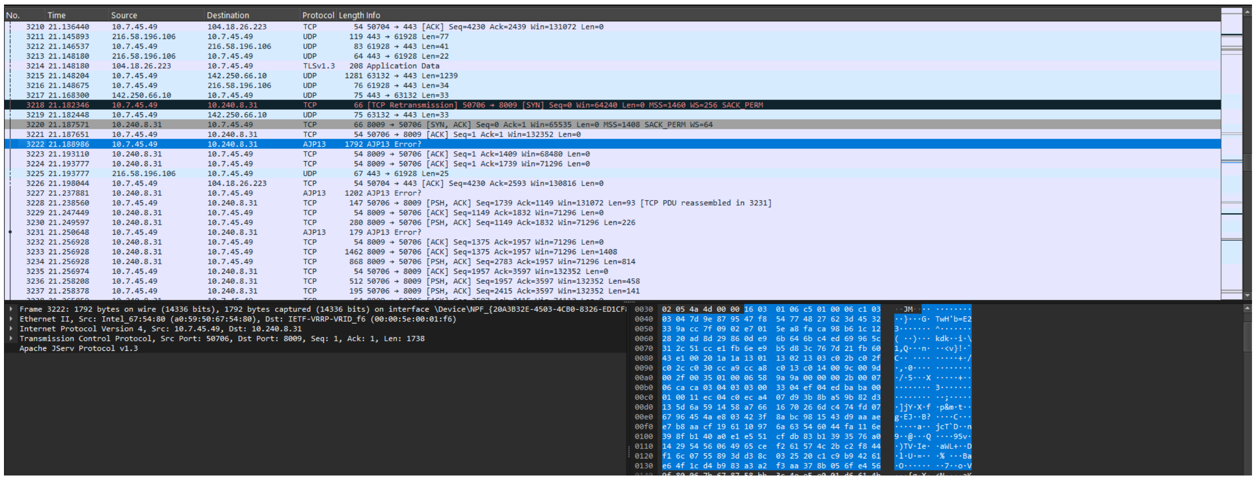


```
No.   Time      Source          Destination     Protocol Length Info
17940 87.468085 10.7.45.49      40.126.18.33    TLSv1.2  4799 Application Data
17941 87.470191 40.126.18.33    10.7.45.49      TCP        60 443 → 60707 [ACK] Seq=4008 Ack=2192 Win=262144 Len=0
17942 87.470191 40.126.18.33    10.7.45.49      TCP        60 443 → 60707 [ACK] Seq=4008 Ack=5112 Win=262144 Len=0
17943 87.471638 10.7.45.49      40.78.107.240   TCP        54 52235 → 443 [ACK] Seq=5357 Ack=32838 Win=131328 Len=0
17944 87.487613 40.126.18.33    10.7.45.49      TCP        60 443 → 60707 [ACK] Seq=4008 Ack=5477 Win=262144 Len=0
17945 87.492219 Cisco_ec:f9:01  Intel_67:54:80  ARP        60 Who has 10.7.45.49? Tell 0.0.0.0
17946 87.492227 Intel_67:54:80  Cisco_ec:f9:01  ARP        42 10.7.45.49 is at a0:59:50:67:54:80
17947 87.642320 146.75.122.172  10.7.45.49      TCP        66 80 → 60708 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 SACK_PERM WS=512
17948 87.642524 10.7.45.49      146.75.122.172  TCP        54 60708 → 80 [ACK] Seq=1 Ack=1 Win=131328 Len=0
17949 87.643231 10.7.45.49      146.75.122.172  HTTP      304 GET /c/msdownload/update/others/2021/01/33356784_31f6eaf6d5437d3e7700835aa29b58168ae2a3b6.cab HTTP/1.1
17950 87.817517 10.7.45.49      10.7.63.255     BROWSER   243 Host Announcement LAPPY, Workstation, Server, NT Workstation
17951 87.832128 146.75.122.172  10.7.45.49      TCP        60 80 → 60708 [ACK] Seq=1 Ack=251 Win=147456 Len=0
17952 87.833443 146.75.122.172  10.7.45.49      TCP      1510 80 → 60708 [ACK] Seq=1 Ack=251 Win=147456 Len=1456 [TCP PDU reassembled in 17957]
17953 87.833443 146.75.122.172  10.7.45.49      TCP      1510 80 → 60708 [PSH, ACK] Seq=1457 Ack=251 Win=147456 Len=1456 [TCP PDU reassembled in 17957]
17954 87.833443 146.75.122.172  10.7.45.49      TCP      1510 80 → 60708 [ACK] Seq=2913 Ack=251 Win=147456 Len=1456 [TCP PDU reassembled in 17957]
17955 87.833443 146.75.122.172  10.7.45.49      TCP      1510 80 → 60708 [PSH, ACK] Seq=4369 Ack=251 Win=147456 Len=1456 [TCP PDU reassembled in 17957]
17956 87.833443 146.75.122.172  10.7.45.49      TCP       298 [TCP Previous segment not captured] 80 → 60708 [PSH, ACK] Seq=7281 Ack=251 Win=147456 Len=244 [TCP PDU reassembled in 17957]
17957 87.833443 146.75.122.172  10.7.45.49      TCP      1510 [TCP Out-Of-Order] 80 → 60708 [ACK] Seq=5825 Ack=251 Win=147456 Len=1456
17958 87.833608 10.7.45.49      146.75.122.172  TCP        54 60708 → 80 [ACK] Seq=251 Ack=2913 Win=131328 Len=0
17959 87.833686 10.7.45.49      146.75.122.172  TCP        54 60708 → 80 [ACK] Seq=251 Ack=5825 Win=131328 Len=0
17960 87.833712 10.7.45.49      146.75.122.172  TCP        66 [TCP Dup ACK 17959#1] 60708 → 80 [ACK] Seq=251 Ack=5825 Win=131328 Len=0 SLE=7281 SRE=7525
17961 87.833735 10.7.45.49      146.75.122.172  TCP        54 60708 → 80 [ACK] Seq=251 Ack=7525 Win=131328 Len=0
17962 87.835407 10.7.45.49      146.75.122.172  HTTP      304 GET /d/msdownload/update/others/2021/01/33356787_6b964b07151d3046d2c549a1d3f61141af23bbd8.cab HTTP/1.1
17963 88.023890 146.75.122.172  10.7.45.49      TCP        60 80 → 60708 [ACK] Seq=7525 Ack=501 Win=148480 Len=0
17964 88.024881 146.75.122.172  10.7.45.49      TCP      1510 80 → 60708 [ACK] Seq=7525 Ack=501 Win=148480 Len=1456 [TCP PDU reassembled in 17971]
17965 88.024881 146.75.122.172  10.7.45.49      TCP      1510 80 → 60708 [PSH, ACK] Seq=8981 Ack=501 Win=148480 Len=1456 [TCP PDU reassembled in 17971]
17966 88.024881 146.75.122.172  10.7.45.49      TCP      1510 80 → 60708 [ACK] Seq=10437 Ack=501 Win=148480 Len=1456 [TCP PDU reassembled in 17971]

> Frame 17950: 243 bytes on wire (1944 bits), 243 bytes captured (1944 bits) on interface \Device\NPF_{20A3B32E-4503-4CB0-8326-ED1CF8EE
> Ethernet II, Src: Intel_67:54:80 (a0:59:50:67:54:80), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
> Internet Protocol Version 4, Src: 10.7.45.49, Dst: 10.7.63.255
> User Datagram Protocol, Src Port: 138, Dst Port: 138
> NetBIOS Datagram Service
> SMB (Server Message Block Protocol)
> SMB MailSlot Protocol
> Microsoft Windows Browser Protocol

0000  ff ff ff ff ff ff a0 59  50 67 54 80 08 00 45 00   ·······Y PgT···E·
0010  00 e5 d2 a5 00 00 80 11  00 00 0a 07 2d 31 0a 07   ········ ····-1··
0020  3f ff 00 8a 00 8a 00 d1  85 e8 11 02 e9 51 0a 07   ?······· ·····Q··
0030  2d 31 00 8a 00 bb 00 00  20 45 4d 45 42 46 41 46   -1······  EMEBFAF
0040  41 46 4a 43 41 43 41 43  41 43 41 43 41 43 41 43   AFJCACAC ACACACAC
0050  41 43 41 43 41 43 41 43  41 00 20 46 48 45 50 46   ACACACAC A· FHEPF
0060  43 45 4c 45 48 46 43 45  50 46 46 46 41 43 41 43   CELEHFCE PFFFACAC
0070  41 43 41 43 41 43 41 43  41 42 4e 00 ff 53 4d 42   ACACACAC ABN··SMB
0080  25 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   %······· ········
0090  00 00 00 00 00 00 00 00  00 00 00 00 11 00 00 21   ········ ·······!
00a0  00 00 00 00 00 00 00 00  00 e8 03 00 00 00 00 00   ········ ········
00b0  00 00 00 21 00 56 00 03  00 01 00 00 00 02 00 32   ···!·V·· ·······2
00c0  00 5c 4d 41 49 4c 53 4c  4f 54 5c 42 52 4f 57 53   ·\MAILSL OT\BROWS
00d0  45 00 01 00 80 a9 03 00  4c 41 50 50 59 00 00 00   E······· LAPPY···
00e0  00 00 00 00 00 00 00 00  0a 00 03 10 00 00 0f 01   ········ ········
00f0  55 aa 00                                           U··
```

3. **AJP13 (Apache JServ Protocol version 1.3)**
   Layer: Application Layer (Layer 7)
   Operation/Usage: AJP13 facilitates communication between web servers (e.g., Apache HTTP Server) and application servers (e.g., Tomcat) to efficiently handle servlet and JSP requests, commonly used in reverse proxy and load balancing setups.
   RFC: RFC 2048



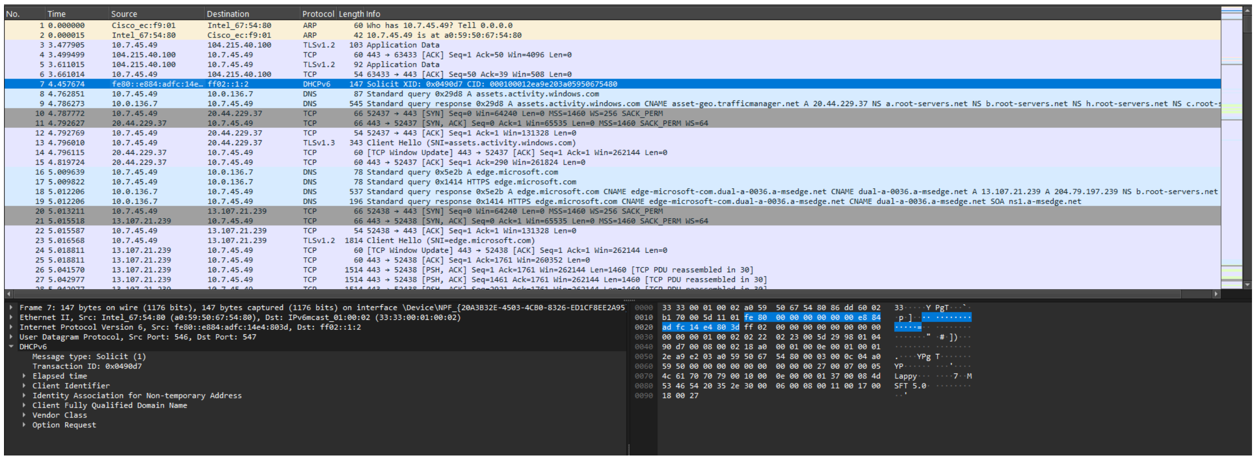
```
No.  Time      Source          Destination     Protocol Length Info
3210 21.136440 10.7.45.49      104.18.26.223   TCP        54 50704 → 443 [ACK] Seq=4230 Ack=2439 Win=131072 Len=0
3211 21.145893 216.58.196.106  10.7.45.49      UDP       119 443 → 61928 Len=77
3212 21.146537 10.7.45.49      216.58.196.106  UDP        83 61928 → 443 Len=41
3213 21.148180 216.58.196.106  10.7.45.49      UDP        64 443 → 61928 Len=22
3214 21.148180 104.18.26.223   10.7.45.49      TLSv1.3   208 Application Data
3215 21.148204 10.7.45.49      142.250.66.10   UDP      1281 63132 → 443 Len=1239
3216 21.148675 10.7.45.49      216.58.196.106  UDP        76 61928 → 443 Len=34
3217 21.168300 142.250.66.10   10.7.45.49      UDP        75 443 → 63132 Len=33
3218 21.182346 10.7.45.49      10.240.8.31     TCP        66 [TCP Retransmission] 50706 → 8009 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
3219 21.182448 10.7.45.49      142.250.66.10   UDP        75 63132 → 443 Len=33
3220 21.187571 10.240.8.31     10.7.45.49      TCP        66 8009 → 50706 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1408 SACK_PERM WS=64
3221 21.187651 10.7.45.49      10.240.8.31     TCP        54 50706 → 8009 [ACK] Seq=1 Ack=1 Win=132352 Len=0
3222 21.188986 10.7.45.49      10.240.8.31     AJP13    1792 AJP13 Error?
3223 21.193110 10.240.8.31     10.7.45.49      TCP        54 8009 → 50706 [ACK] Seq=1 Ack=1409 Win=68480 Len=0
3224 21.193777 10.240.8.31     10.7.45.49      TCP        54 8009 → 50706 [ACK] Seq=1 Ack=1739 Win=71296 Len=0
3225 21.193777 216.58.196.106  10.7.45.49      UDP        67 443 → 61928 Len=25
3226 21.198044 10.7.45.49      104.18.26.223   TCP        54 50704 → 443 [ACK] Seq=4230 Ack=2593 Win=130816 Len=0
3227 21.237881 10.240.8.31     10.7.45.49      AJP13    1202 AJP13 Error?
3228 21.238560 10.7.45.49      10.240.8.31     TCP       147 50706 → 8009 [PSH, ACK] Seq=1739 Ack=1149 Win=131072 Len=93 [TCP PDU reassembled in 3231]
3229 21.247449 10.240.8.31     10.7.45.49      TCP        54 8009 → 50706 [ACK] Seq=1149 Ack=1832 Win=71296 Len=0
3230 21.249597 10.240.8.31     10.7.45.49      TCP       280 8009 → 50706 [PSH, ACK] Seq=1149 Ack=1832 Win=71296 Len=226
3231 21.250648 10.7.45.49      10.240.8.31     AJP13     179 AJP13 Error?
3232 21.256928 10.240.8.31     10.7.45.49      TCP        54 8009 → 50706 [ACK] Seq=1375 Ack=1957 Win=71296 Len=0
3233 21.256928 10.240.8.31     10.7.45.49      TCP      1462 8009 → 50706 [ACK] Seq=1375 Ack=1957 Win=71296 Len=1408
3234 21.256928 10.240.8.31     10.7.45.49      TCP       868 8009 → 50706 [PSH, ACK] Seq=2783 Ack=1957 Win=71296 Len=814
3235 21.256974 10.7.45.49      10.240.8.31     TCP        54 50706 → 8009 [ACK] Seq=1957 Ack=3597 Win=132352 Len=0
3236 21.258208 10.7.45.49      10.240.8.31     TCP       512 50706 → 8009 [PSH, ACK] Seq=1957 Ack=3597 Win=132352 Len=458
3237 21.258378 10.7.45.49      10.240.8.31     TCP       195 50706 → 8009 [PSH, ACK] Seq=2415 Ack=3597 Win=132352 Len=141

> Frame 3222: 1792 bytes on wire (14336 bits), 1792 bytes captured (14336 bits) on interface \Device\NPF_{20A3B32E-4503-4CB0-8326-ED1CF
> Ethernet II, Src: Intel_67:54:80 (a0:59:50:67:54:80), Dst: IETF-VRRP-VRID_f6 (00:00:5e:00:01:f6)
> Internet Protocol Version 4, Src: 10.7.45.49, Dst: 10.240.8.31
> Transmission Control Protocol, Src Port: 50706, Dst Port: 8009, Seq: 1, Ack: 1, Len: 1738
  Apache JServ Protocol v1.3

0030  02 05 4a 4d 00 00 16 03  01 06 c5 01 00 06 c1 03   ··JM···· ········
0040  03 04 7d 9e 87 95 47 f8  54 77 48 27 62 3d 45 32   ··}···G· TwH'b=E2
0050  33 9a cc 7f 09 02 e7 01  5e a8 fa ca 98 b6 1c 12   3······· ^·······
0060  28 20 ad 8d 29 86 0d e9  6b 64 6b c4 ed 69 96 5c   ( ··)··· kdk··i·\
0070  31 2c 51 cc e1 fb 6e e9  b5 d8 3c 76 7d 21 fb 60   1,Q···n· ··<v}!·`
0080  43 e1 00 20 1a 1a 13 01  13 02 13 03 c0 2b c0 2f   C·· ···· ·····+·/
0090  c0 2c c0 30 cc a9 cc a8  c0 13 c0 14 00 9c 00 9d   ·,·0···· ········
00a0  00 2f 00 35 01 00 06 58  9a 9a 00 00 00 2b 00 07   ·/·5···X ·····+··
00b0  06 ca ca 03 04 03 03 00  33 04 ef 04 ed ba ba 00   ········ 3·······
00c0  01 00 11 ec 04 c0 ec a4  07 d9 3b 8b a5 9b 82 d3   ········ ··;·····
00d0  13 5d 6a 59 14 58 a7 66  16 70 26 6d c4 74 fd 07   ·]jY·X·f ·p&m·t··
00e0  67 96 45 4a e8 03 42 3f  8a bc 98 15 43 d9 aa ae   g·EJ··B? ····C···
00f0  e7 b8 aa cf 19 61 10 97  6a 63 54 60 44 fa 11 6e   ·····a·· jcT`D··n
0100  39 8f b1 40 a0 e1 e5 51  cf db 83 b1 39 35 76 a0   9··@···Q ····95v·
0110  14 29 54 56 06 49 65 ce  f2 61 57 4c 2b c2 f8 44   ·)TV·Ie· ·aWL+··D
0120  f1 6c 07 55 89 3d d3 8c  03 25 20 c1 c9 b9 42 61   ·l·U·=·· ·% ···Ba
0130  e6 4f 1c d4 b9 83 a3 a2  f3 aa 37 8b 05 6f e4 56   ·O······ ··7··o·V
```

4. **DHCP (Dynamic Host Configuration Protocol)**
   Layer: Application Layer (Layer 7)
   Operation/Usage: DHCP dynamically assigns IP addresses and network configuration details (e.g., subnet mask, gateway) to devices on a network, eliminating the need for manual configuration.

RFC: RFC 213



```
No.   Time       Source              Destination       Protocol Length Info
    1 0.000000   Cisco_ec:f9:01      Intel_67:54:80    ARP        60 Who has 10.7.45.49? Tell 0.0.0.0
    2 0.000015   Intel_67:54:80      Cisco_ec:f9:01    ARP        42 10.7.45.49 is at a0:59:50:67:54:80
    3 3.477905   10.7.45.49          104.215.40.100    TLSv1.2   103 Application Data
    4 3.499499   104.215.40.100      10.7.45.49        TCP        60 443 → 63433 [ACK] Seq=1 Ack=50 Win=4096 Len=0
    5 3.611015   104.215.40.100      10.7.45.49        TLSv1.2    92 Application Data
    6 3.661014   10.7.45.49          104.215.40.100    TCP        54 63433 → 443 [ACK] Seq=50 Ack=39 Win=508 Len=0
    7 4.457674   fe80::e884:adfc:14e… ff02::1:2        DHCPv6    147 Solicit XID: 0x0490d7 CID: 000100012ea9e203a05950675480
    8 4.762851   10.7.45.49          10.0.136.7        DNS        87 Standard query 0x29d8 A assets.activity.windows.com
    9 4.786273   10.0.136.7          10.7.45.49        DNS       545 Standard query response 0x29d8 A assets.activity.windows.com CNAME asset-geo.trafficmanager.net A 20.44.229.37 NS a.root-servers.net NS b.root-servers.net NS h.root-servers.net NS c.root-s
   10 4.787772   10.7.45.49          20.44.229.37      TCP        66 52437 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
   11 4.792627   20.44.229.37        10.7.45.49        TCP        66 443 → 52437 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 SACK_PERM WS=64
   12 4.792769   10.7.45.49          20.44.229.37      TCP        54 52437 → 443 [ACK] Seq=1 Ack=1 Win=131328 Len=0
   13 4.796010   10.7.45.49          20.44.229.37      TLSv1.3   343 Client Hello (SNI=assets.activity.windows.com)
   14 4.796115   20.44.229.37        10.7.45.49        TCP        60 [TCP Window Update] 443 → 52437 [ACK] Seq=1 Ack=1 Win=262144 Len=0
   15 4.819724   20.44.229.37        10.7.45.49        TCP        60 443 → 52437 [ACK] Seq=1 Ack=290 Win=261824 Len=0
   16 5.009639   10.7.45.49          10.0.136.7        DNS        78 Standard query 0x5e2b A edge.microsoft.com
   17 5.009822   10.7.45.49          10.0.136.7        DNS        78 Standard query 0x1414 HTTPS edge.microsoft.com
   18 5.012206   10.0.136.7          10.7.45.49        DNS       537 Standard query response 0x5e2b A edge.microsoft.com CNAME edge-microsoft-com.dual-a-0036.a-msedge.net CNAME dual-a-0036.a-msedge.net A 13.107.21.239 A 204.79.197.239 NS b.root-servers.net
   19 5.012206   10.0.136.7          10.7.45.49        DNS       196 Standard query response 0x1414 HTTPS edge.microsoft.com CNAME edge-microsoft-com.dual-a-0036.a-msedge.net CNAME dual-a-0036.a-msedge.net SOA ns1.a-msedge.net
   20 5.013211   10.7.45.49          13.107.21.239     TCP        66 52438 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
   21 5.015518   13.107.21.239       10.7.45.49        TCP        66 443 → 52438 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 SACK_PERM WS=64
   22 5.015587   10.7.45.49          13.107.21.239     TCP        54 52438 → 443 [ACK] Seq=1 Ack=1 Win=131328 Len=0
   23 5.016568   10.7.45.49          13.107.21.239     TLSv1.2  1814 Client Hello (SNI=edge.microsoft.com)
   24 5.018811   13.107.21.239       10.7.45.49        TCP        60 [TCP Window Update] 443 → 52438 [ACK] Seq=1 Ack=1 Win=262144 Len=0
   25 5.018811   13.107.21.239       10.7.45.49        TCP        60 443 → 52438 [ACK] Seq=1 Ack=1761 Win=260352 Len=0
   26 5.041570   13.107.21.239       10.7.45.49        TCP      1514 443 → 52438 [PSH, ACK] Seq=1 Ack=1761 Win=262144 Len=1460 [TCP PDU reassembled in 30]
   27 5.042977   13.107.21.239       10.7.45.49        TCP      1514 443 → 52438 [PSH, ACK] Seq=1461 Ack=1761 Win=262144 Len=1460 [TCP PDU reassembled in 30]
```

```
▸ Frame 7: 147 bytes on wire (1176 bits), 147 bytes captured (1176 bits) on interface \Device\NPF_{20A3B32E-4503-4CB0-8326-ED1CF8EE2A95
▸ Ethernet II, Src: Intel_67:54:80 (a0:59:50:67:54:80), Dst: IPv6mcast_01:00:02 (33:33:00:01:00:02)
▸ Internet Protocol Version 6, Src: fe80::e884:adfc:14e4:803d, Dst: ff02::1:2
▸ User Datagram Protocol, Src Port: 546, Dst Port: 547
▾ DHCPv6
    Message type: Solicit (1)
    Transaction ID: 0x0490d7
  ▸ Elapsed time
  ▸ Client Identifier
  ▸ Identity Association for Non-temporary Address
  ▸ Client Fully Qualified Domain Name
  ▸ Vendor Class
  ▸ Option Request
```

```
0000  33 33 00 01 00 02 a0 59  50 67 54 80 86 dd 60 02   33····Y PgT···`·
0010  b1 70 00 5d 11 01 fe 80  00 00 00 00 00 00 e8 84   ·p·]···· ········
0020  ad fc 14 e4 80 3d ff 02  00 00 00 00 00 00 00 00   ·····=·· ········
0030  00 00 00 01 00 02 02 22  02 23 00 5d 29 98 01 04   ·······" ·#·])···
0040  90 d7 00 08 00 02 18 a0  00 01 00 0e 00 01 00 01   ········ ········
0050  2e a9 e2 03 a0 59 50 67  54 80 00 03 00 0c 04 a0   .····YPg T·······
0060  59 50 00 00 00 00 00 00  00 00 00 27 00 07 00 05   YP······ ···'····
0070  4c 61 70 70 79 00 10 00  0e 00 00 01 37 00 08 4d   Lappy··· ····7··M
0080  53 46 54 20 35 2e 30 00  06 00 08 00 11 00 17 00   SFT 5.0· ········
0090  18 00 27                                           ··'
```

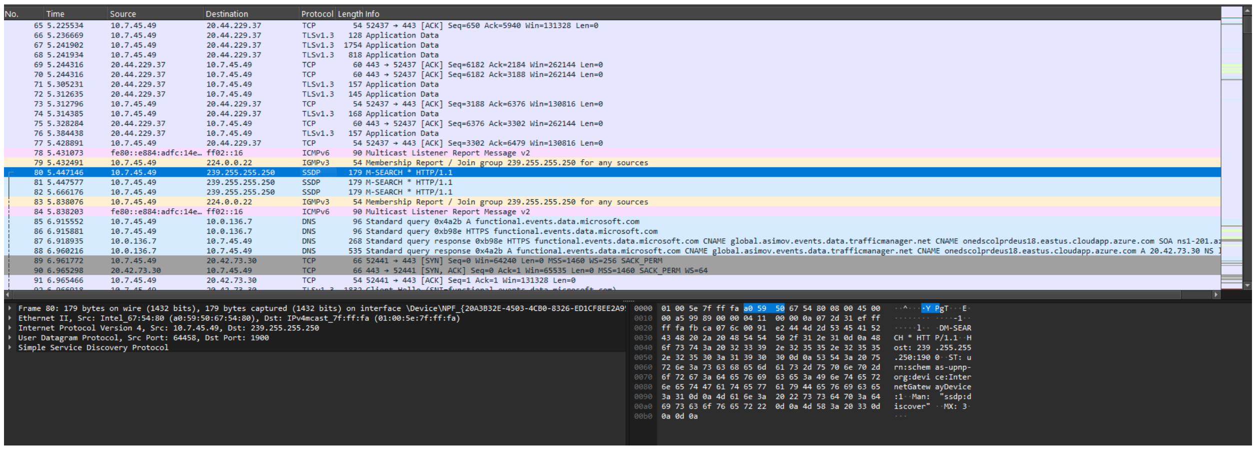
1

5. SSDP (Simple Service Discovery Protocol)

Layer: Application Layer (Layer 7)

Operation/Usage: SSDP is used for the discovery of network devices and services, such as printers or smart TVs, in a local network using multicast.

RFC: RFC 6776



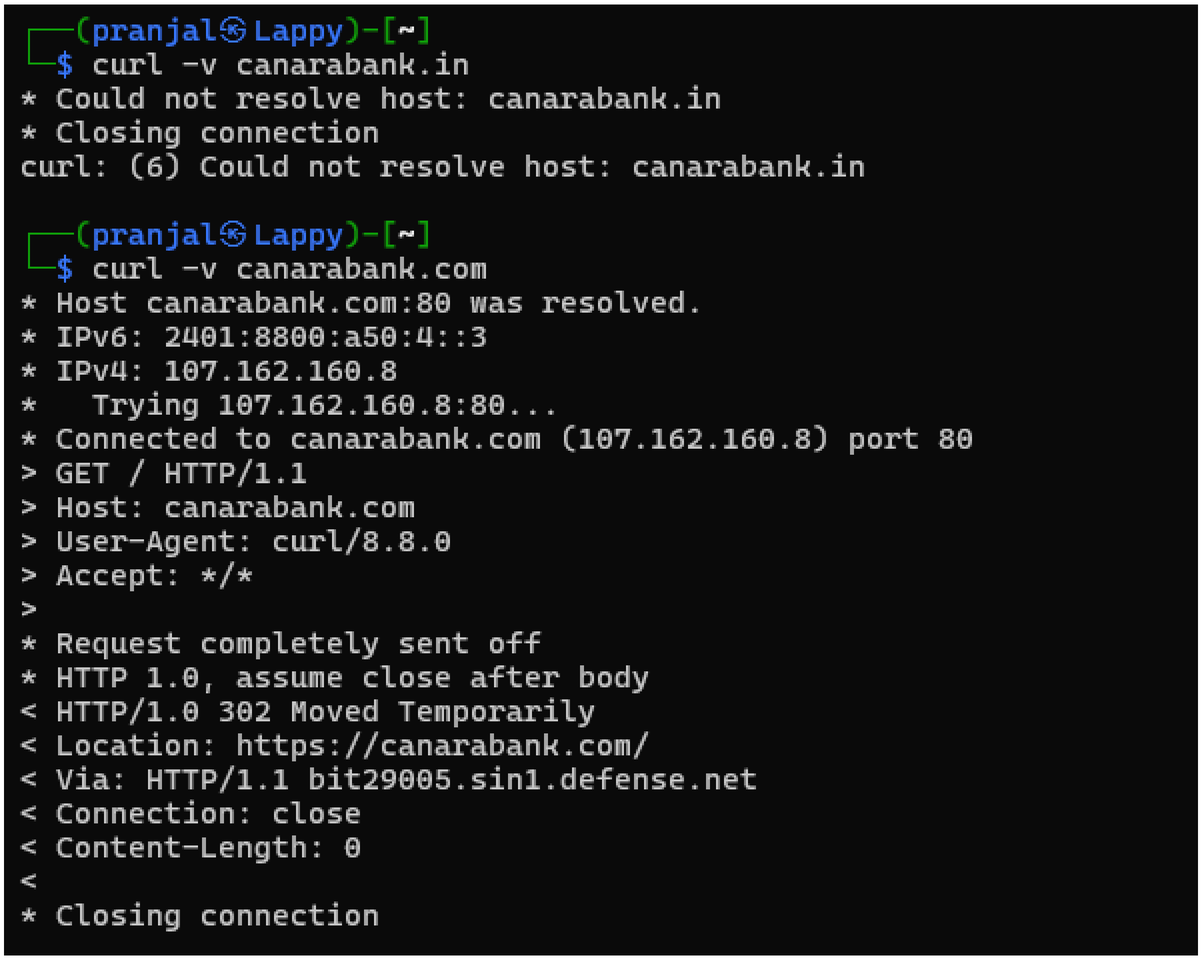
```
No.   Time       Source              Destination       Protocol Length Info
   65 5.225534   10.7.45.49          20.44.229.37      TCP        54 52437 → 443 [ACK] Seq=650 Ack=5940 Win=131328 Len=0
   66 5.236669   10.7.45.49          20.44.229.37      TLSv1.3   128 Application Data
   67 5.241902   10.7.45.49          20.44.229.37      TLSv1.3  1754 Application Data
   68 5.241934   10.7.45.49          20.44.229.37      TLSv1.3   818 Application Data
   69 5.244316   20.44.229.37        10.7.45.49        TCP        60 443 → 52437 [ACK] Seq=6182 Ack=2184 Win=262144 Len=0
   70 5.244316   20.44.229.37        10.7.45.49        TCP        60 443 → 52437 [ACK] Seq=6182 Ack=3188 Win=262144 Len=0
   71 5.305231   20.44.229.37        10.7.45.49        TLSv1.3   157 Application Data
   72 5.312635   20.44.229.37        10.7.45.49        TLSv1.3   145 Application Data
   73 5.312796   10.7.45.49          20.44.229.37      TCP        54 52437 → 443 [ACK] Seq=3188 Ack=6376 Win=130816 Len=0
   74 5.314385   10.7.45.49          20.44.229.37      TLSv1.3   168 Application Data
   75 5.328284   20.44.229.37        10.7.45.49        TCP        60 443 → 52437 [ACK] Seq=6376 Ack=3302 Win=262144 Len=0
   76 5.384438   20.44.229.37        10.7.45.49        TLSv1.3   157 Application Data
   77 5.428891   10.7.45.49          20.44.229.37      TCP        54 52437 → 443 [ACK] Seq=3302 Ack=6479 Win=130816 Len=0
   78 5.431073   fe80::e884:adfc:14e… ff02::16         ICMPv6     90 Multicast Listener Report Message v2
   79 5.432491   10.7.45.49          224.0.0.22        IGMPv3     54 Membership Report / Join group 239.255.255.250 for any sources
   80 5.447146   10.7.45.49          239.255.255.250   SSDP      179 M-SEARCH * HTTP/1.1
   81 5.447577   10.7.45.49          239.255.255.250   SSDP      179 M-SEARCH * HTTP/1.1
   82 5.666176   10.7.45.49          239.255.255.250   SSDP      179 M-SEARCH * HTTP/1.1
   83 5.838076   10.7.45.49          224.0.0.22        IGMPv3     54 Membership Report / Join group 239.255.255.250 for any sources
   84 5.838203   fe80::e884:adfc:14e… ff02::16         ICMPv6     90 Multicast Listener Report Message v2
   85 6.915552   10.7.45.49          10.0.136.7        DNS        96 Standard query 0x4a2b A functional.events.data.microsoft.com
   86 6.915881   10.7.45.49          10.0.136.7        DNS        96 Standard query 0xb98e HTTPS functional.events.data.microsoft.com
   87 6.918935   10.0.136.7          10.7.45.49        DNS       268 Standard query response 0xb98e HTTPS functional.events.data.microsoft.com CNAME global.asimov.events.data.trafficmanager.net CNAME onedscolprdeus18.eastus.cloudapp.azure.com SOA ns1-201.a
   88 6.960216   10.0.136.7          10.7.45.49        DNS       535 Standard query response 0x4a2b A functional.events.data.microsoft.com CNAME global.asimov.events.data.trafficmanager.net CNAME onedscolprdeus18.eastus.cloudapp.azure.com A 20.42.73.30 NS 
   89 6.961772   10.7.45.49          20.42.73.30       TCP        66 52441 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
   90 6.965298   20.42.73.30         10.7.45.49        TCP        66 443 → 52441 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 SACK_PERM WS=64
   91 6.965466   10.7.45.49          20.42.73.30       TCP        54 52441 → 443 [ACK] Seq=1 Ack=1 Win=131328 Len=0
```

```
▸ Frame 80: 179 bytes on wire (1432 bits), 179 bytes captured (1432 bits) on interface \Device\NPF_{20A3B32E-4503-4CB0-8326-ED1CF8EE2A9
▸ Ethernet II, Src: Intel_67:54:80 (a0:59:50:67:54:80), Dst: IPv4mcast_7f:ff:fa (01:00:5e:7f:ff:fa)
▸ Internet Protocol Version 4, Src: 10.7.45.49, Dst: 239.255.255.250
▸ User Datagram Protocol, Src Port: 64458, Dst Port: 1900
▸ Simple Service Discovery Protocol
```

```
0000  01 00 5e 7f ff fa a0 59  50 67 54 80 08 00 45 00   ··^····Y PgT···E·
0010  00 a5 99 89 00 00 04 11  00 00 0a 07 2d 31 ef ff   ········ ····-1··
0020  ff fa fb ca 07 6c 00 91  e2 44 4d 2d 53 45 41 52   ·····l·· ·DM-SEAR
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 22  0d 0a 4d 58 3a 20 33 0d   iscover" ··MX: 3·
00b0  0a 0d 0a                                           ···
```

Q2:
Analyze the following details by visiting the following websites in your favorite browser.
i) canarabank.in
ii) github.com
iii) netflix.com
a. Identify `the request line` with the version of the application layer protocol and the IP address. Also, identify whether the connection(s) is/are persistent or not.
b. For any one of the websites, list any three header field names and corresponding values in the request and response message. Any three HTTP error codes obtained while loading one of the pages with a brief description.
c. Capture the Performance metrics that your browser records when a page is loaded and also report the list the cookies used and the associated flags in the request and response headers. Please report the browser name and screenshot of the performance metrics reported for any one of the page loads

Ans a :
i) canara bank

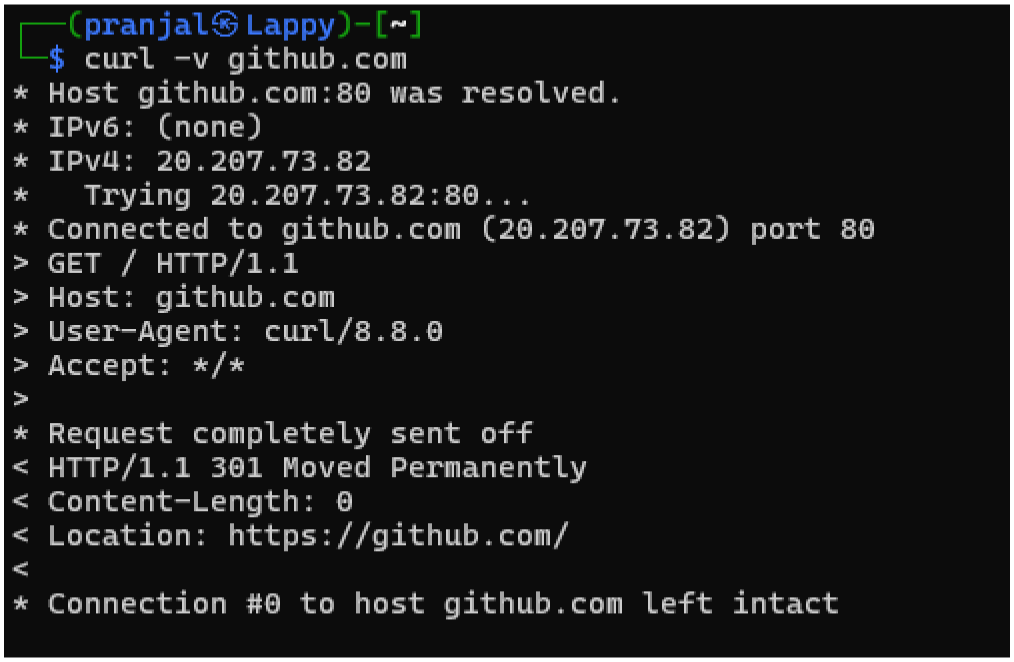


```
┌──(pranjal㉿Lappy)-[~]
└─$ curl -v canarabank.in
* Could not resolve host: canarabank.in
* Closing connection
curl: (6) Could not resolve host: canarabank.in

┌──(pranjal㉿Lappy)-[~]
└─$ curl -v canarabank.com
* Host canarabank.com:80 was resolved.
* IPv6: 2401:8800:a50:4::3
* IPv4: 107.162.160.8
*   Trying 107.162.160.8:80...
* Connected to canarabank.com (107.162.160.8) port 80
> GET / HTTP/1.1
> Host: canarabank.com
> User-Agent: curl/8.8.0
> Accept: */*
>
* Request completely sent off
* HTTP 1.0, assume close after body
< HTTP/1.0 302 Moved Temporarily
< Location: https://canarabank.com/
< Via: HTTP/1.1 bit29005.sin1.defense.net
< Connection: close
< Content-Length: 0
<
* Closing connection
```

Request Line : GET / HTTP/1.1
IP Address : 107.162.160.8
Persistent : No

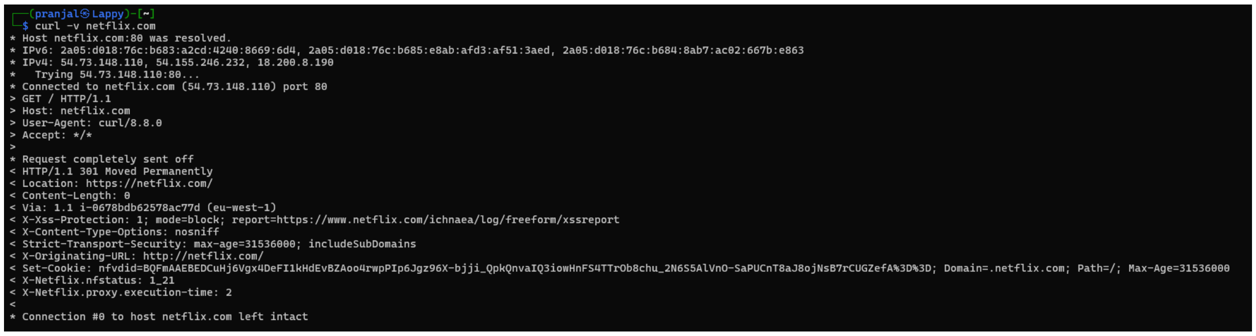
ii) github.com



```
┌──(pranjal㉿Lappy)-[~]
└─$ curl -v github.com
* Host github.com:80 was resolved.
* IPv6: (none)
* IPv4: 20.207.73.82
*   Trying 20.207.73.82:80...
* Connected to github.com (20.207.73.82) port 80
> GET / HTTP/1.1
> Host: github.com
> User-Agent: curl/8.8.0
> Accept: */*
>
* Request completely sent off
< HTTP/1.1 301 Moved Permanently
< Content-Length: 0
< Location: https://github.com/
<
* Connection #0 to host github.com left intact
```

Request Line: GET / HTTP/1.1
IP Address : 20.207.73.82
Persistent : Yes

iii) netflix.com

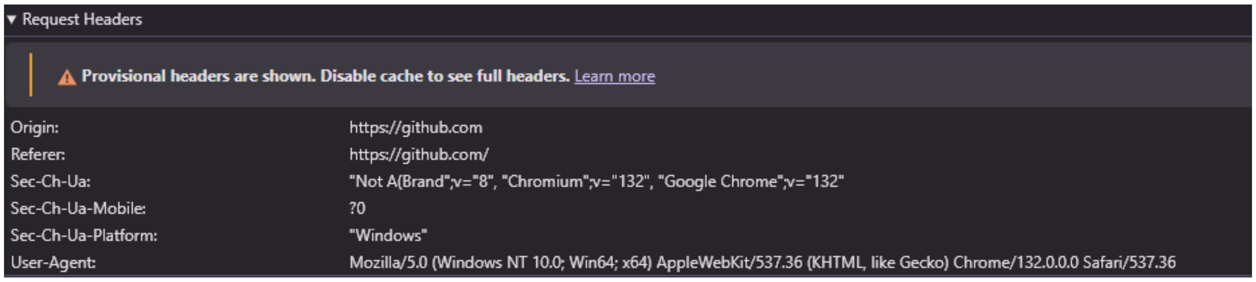


```
┌──(pranjal㉿Lappy)-[~]
└─$ curl -v netflix.com
* Host netflix.com:80 was resolved.
* IPv6: 2a05:d018:76c:b683:a2cd:4240:8669:6d4, 2a05:d018:76c:b685:e8ab:afd3:af51:3aed, 2a05:d018:76c:b684:8ab7:ac02:667b:e863
* IPv4: 54.73.148.110, 54.155.246.232, 18.200.8.190
*   Trying 54.73.148.110:80...
* Connected to netflix.com (54.73.148.110) port 80
> GET / HTTP/1.1
> Host: netflix.com
> User-Agent: curl/8.8.0
> Accept: */*
>
* Request completely sent off
< HTTP/1.1 301 Moved Permanently
< Location: https://netflix.com/
< Content-Length: 0
< Via: 1.1 i-0678bdb62578ac77d (eu-west-1)
< X-Xss-Protection: 1; mode=block; report=https://www.netflix.com/ichnaea/log/freeform/xssreport
< X-Content-Type-Options: nosniff
< Strict-Transport-Security: max-age=31536000; includeSubDomains
< X-Originating-URL: http://netflix.com/
< Set-Cookie: nfvdid=BQFmAAEBEDCuHj6Vgx4DeFI1kHdEvBZAoo4rwpPIp6Jgz96X-bjji_QpkQnvaIQ3iowHnFS4TTrOb8chu_2N6S5AlVnO-SaPUCnT8aJ8ojNsB7rCUGZefA%3D%3D; Domain=.netflix.com; Path=/; Max-Age=31536000
< X-Netflix.nfstatus: 1_21
< X-Netflix.proxy.execution-time: 2
<
* Connection #0 to host netflix.com left intact
```

Request Line : GET / HTTP/1.1
IP Address :   54.73.148.110, 54.155.246.232, 18.200.8.190
Persistent: Yes

Ans b) For github.com

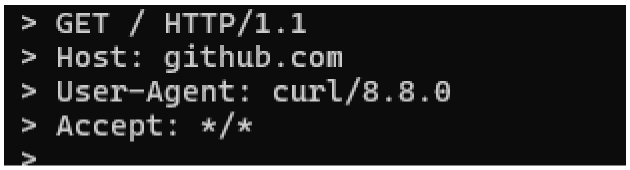
Request header :



▼ Request Headers

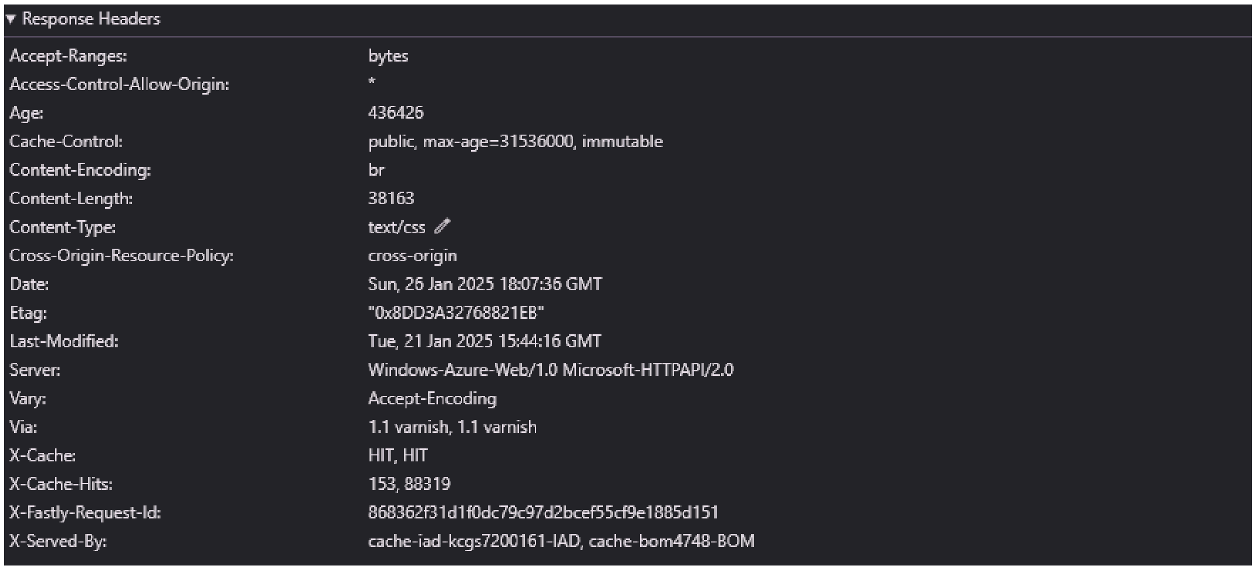
⚠ **Provisional headers are shown. Disable cache to see full headers.** Learn more

Origin:	https://github.com
Referer:	https://github.com/
Sec-Ch-Ua:	"Not A(Brand";v="8", "Chromium";v="132", "Google Chrome";v="132"
Sec-Ch-Ua-Mobile:	?0
Sec-Ch-Ua-Platform:	"Windows"
User-Agent:	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/132.0.0.0 Safari/537.36



```
> GET / HTTP/1.1
> Host: github.com
> User-Agent: curl/8.8.0
> Accept: */*
>
```

Response Header :



▼ Response Headers

Accept-Ranges:	bytes
Access-Control-Allow-Origin:	*
Age:	436426
Cache-Control:	public, max-age=31536000, immutable
Content-Encoding:	br
Content-Length:	38163
Content-Type:	text/css
Cross-Origin-Resource-Policy:	cross-origin
Date:	Sun, 26 Jan 2025 18:07:36 GMT
Etag:	"0x8DD3A32768821EB"
Last-Modified:	Tue, 21 Jan 2025 15:44:16 GMT
Server:	Windows-Azure-Web/1.0 Microsoft-HTTPAPI/2.0
Vary:	Accept-Encoding
Via:	1.1 varnish, 1.1 varnish
X-Cache:	HIT, HIT
X-Cache-Hits:	153, 88319
X-Fastly-Request-Id:	868362f31d1f0dc79c97d2bcef55cf9e1885d151
X-Served-By:	cache-iad-kcgs7200161-IAD, cache-bom4748-BOM

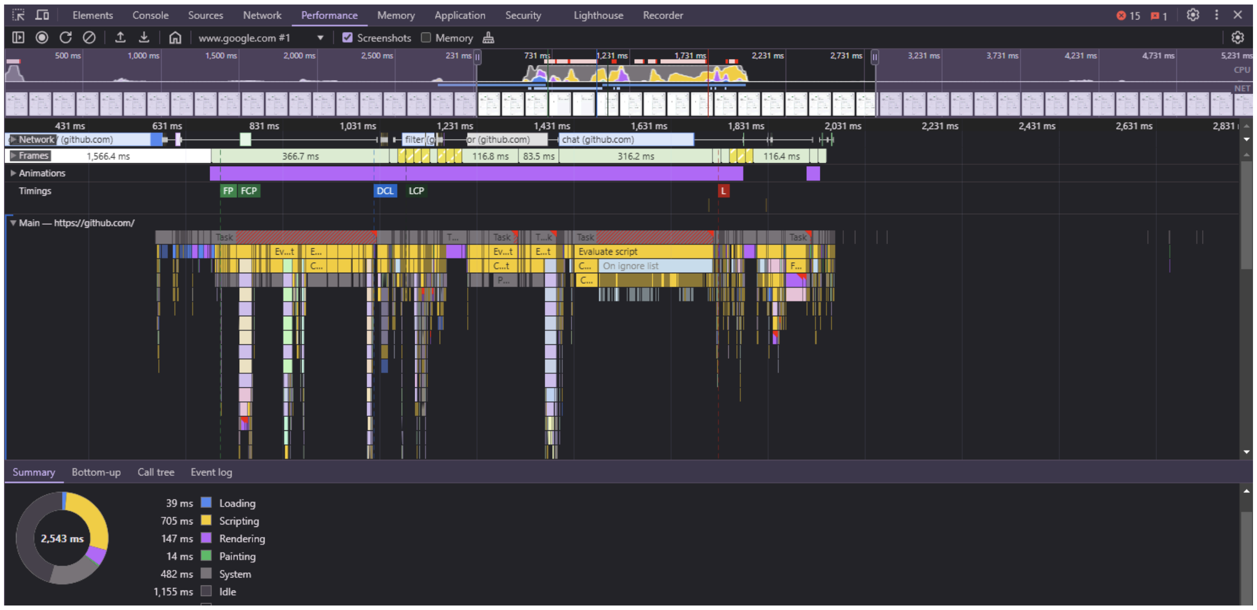
HTTP Error Codes

• **404 Not Found:** This error occurs when the requested resource is not found on the server.
• **500 Internal Server Error:** A general server-side error indicating that something went wrong on the server while processing the request.
• **403 Forbidden:** The server understands the request but refuses to authorize it. This usually happens due to permission issues.

Ans c)

For github.com :

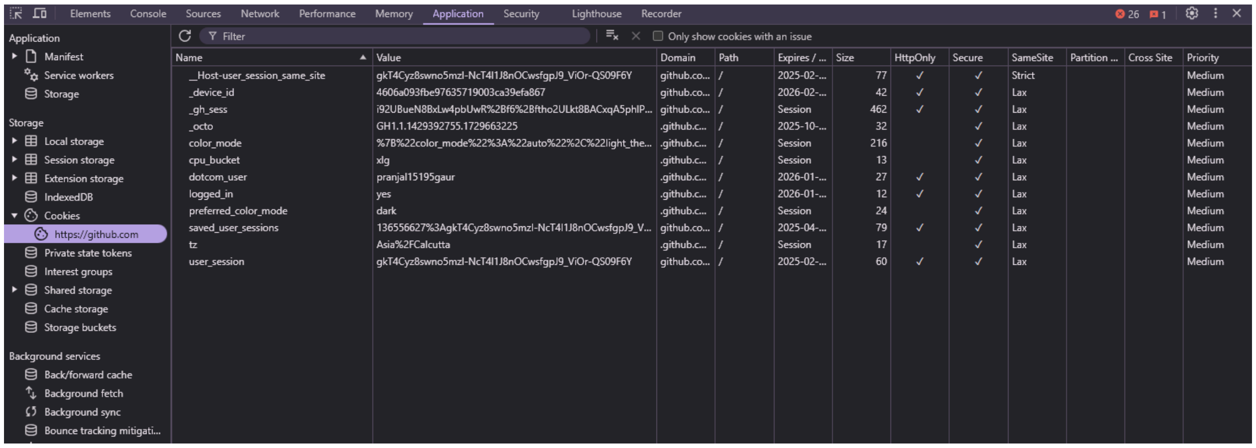
Performance Metric :



39 ms	Loading
705 ms	Scripting
147 ms	Rendering
14 ms	Painting
482 ms	System
1,155 ms	Idle

Total: 2,543 ms

List of Cookies :



Name	Value	Domain	Path	Expires / ...	Size	HttpOnly	Secure	SameSite	Partition ...	Cross Site	Priority
__Host-user_session_same_site	gkT4Cyz8swno5mzl-NcT4l1J8nOCwsfgpJ9_ViOr-QS09F6Y	github.co...	/	2025-02-...	77	✓	✓	Strict			Medium
_device_id	4606a093fbe97635719003ca39efa867	github.co...	/	2026-02-...	42	✓	✓	Lax			Medium
_gh_sess	i92UBueN8BxLw4pbUwR%2Bf6%2Bftho2ULkt8BACxqA5phIP...	github.co...	/	Session	462	✓	✓	Lax			Medium
_octo	GH1.1.1429392755.1729663225	.github.c...	/	2025-10-...	32		✓	Lax			Medium
color_mode	%7B%22color_mode%22%3A%22auto%22%2C%22light_the...	.github.c...	/	Session	216		✓	Lax			Medium
cpu_bucket	xlg	.github.c...	/	Session	13		✓	Lax			Medium
dotcom_user	pranjal15195gaur	.github.c...	/	2026-01-...	27	✓	✓	Lax			Medium
logged_in	yes	.github.c...	/	2026-01-...	12	✓	✓	Lax			Medium
preferred_color_mode	dark	.github.c...	/	Session	24		✓	Lax			Medium
saved_user_sessions	136556627%3AgkT4Cyz8swno5mzl-NcT4l1J8nOCwsfgpJ9_V...	github.co...	/	2025-04-...	79	✓	✓	Lax			Medium
tz	Asia%2FCalcutta	.github.c...	/	Session	17		✓	Lax			Medium
user_session	gkT4Cyz8swno5mzl-NcT4l1J8nOCwsfgpJ9_ViOr-QS09F6Y	github.co...	/	2025-02-...	60	✓	✓	Lax			Medium