

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**

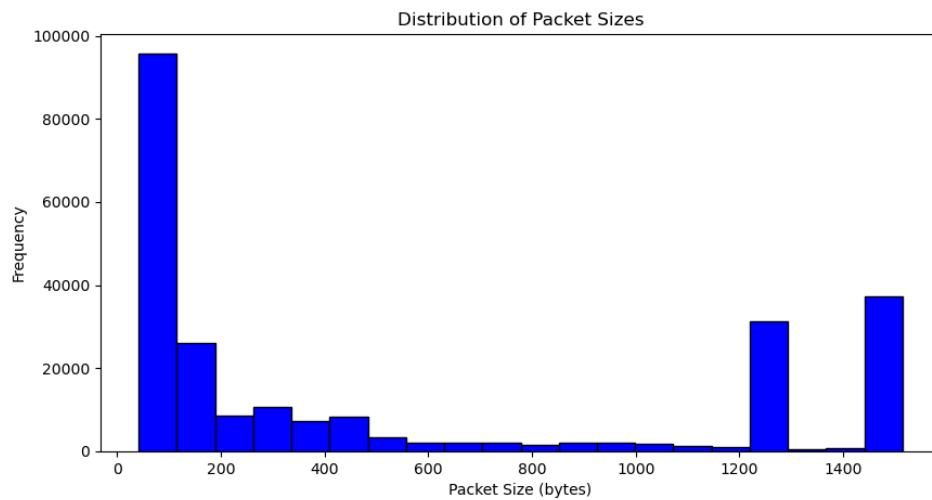


Fig 1. Packet Size Distribution

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 > ...
1 {
2   "249.1.0.0": 2,
3   "147.77.10.7": 2,
4   "10.7.11.235": 2543,
5   "54.236.225.107": 26,
6   "52.1.8.106": 20,
7   "180.149.61.76": 2539,
8   "142.250.192.142": 161,
9   "10.240.8.31": 3149,
10  "142.250.183.78": 49,
11  "142.250.70.100": 3391,
12  "10.0.136.7": 5658,
13  "142.250.194.78": 116,
14  "216.58.200.170": 57,
15  "1.246.10.7": 10,
16  "142.251.42.69": 806,
17  "142.250.192.97": 221,
18  "142.250.70.67": 50,
19  "142.251.12.100": 10,
20  "142.250.194.206": 19,
21  "54.192.142.120": 27,
22  "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 > ...
1 {
2   "0.0.192.228": 2,
3   "11.235.112.228": 2,
4   "54.236.225.107": 27,
5   "10.7.11.235": 7897,
6   "52.1.8.106": 16,
7   "180.149.61.76": 393,
8   "142.250.192.142": 143,
9   "10.240.8.31": 2791,
10  "142.250.183.78": 39,
11  "142.250.70.100": 638,
12  "10.0.136.7": 6726,
13  "142.250.194.78": 111,
14  "216.58.200.170": 49,
15  "0.1.255.255": 10,
16  "142.251.42.69": 414,
17  "142.250.192.97": 174,
18  "142.251.12.100": 8,
19  "142.250.70.67": 36,
20  "142.250.194.206": 17,
21  "54.192.142.120": 24,
22  "142.251.175.188": 4,
23  "142.250.192.3": 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

```
devansh@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

devansh@Dewansh:~/CN_Assignment_1$ 
devansh@Dewansh:~/CN_Assignment_1$ 
devansh@Dewansh:~/CN_Assignment_1$ 
devansh@Dewansh:~/CN_Assignment_1$ 
devansh@Dewansh:~/CN_Assignment_1$ 
devansh@Dewansh:~/CN_Assignment_1$ 
devansh@Dewansh:~/CN_Assignment_1$ 
devansh@Dewansh:~/CN_Assignment_1$ 
devansh@Dewansh:~/CN_Assignment_1$ 
devansh@Dewansh:~/CN_Assignment_1$ 
devansh@Dewansh:~/CN_Assignment_1$ 
devansh@Dewansh:~/CN_Assignment_1$ 

o devansh@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 **enpl1s0** 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: OSCP 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=26214 Len=1176 [TCP RST assembled in 14205]
14205	51.076743	20.247.162.75	10.7.45.49	TLSv1	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	51.077146	10.7.45.49	20.247.162.75	TCP	54	52199 → 443 [ACK] Seq=294 Ack=1921 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=5337 Win=262144 Len=0
14210	51.083599	10.7.45.49	20.247.162.75	TLSv1	134	Change Cipher Spec, Application Data
14211	51.089927	20.247.162.75	10.7.45.49	TCP	60	443 → 52199 [ACK] Seq=5237 Ack=1921 Win=262144 Len=0
14212	51.102203	10.7.45.49	10.6.136.7	DNS	81	Standard query 0x0efc Oneopsec.microsoft.com
14213	51.120444	10.6.136.7	10.7.45.49	DNS	550	Standard query response 0x0efc Oneopsec.microsoft.com CNAME oneopsec.microsoft.com.0-0003.a-msedge.net CNAME a-0003.a-msedge.net.a 204.79.197.203 NS c.root-servers.net H A root-servers.net
14214	51.121575	10.7.45.49	204.79.197.203	TCP	60	52200 → 80 [ESTAB] Seq=0 Win=4240 Len=0 MSS=1460 Win-Scale=3
14215	51.135971	204.79.197.203	10.7.45.49	TCP	66	80 → 52200 [WIN, ACK] Seq=0 Ack=1 Win=55335 Len=0 MSS=1460 Win-Scale=3
14216	51.136110	10.7.45.49	204.79.197.203	TCP	54	52200 → 80 [ACK] Seq=1 Ack=1 Win=12352 Len=0
14217	51.136066	10.7.45.49	204.79.197.203	HTTP	303	GET /oscp/PwqzUgWmTnRgKtGQdImsE=7kZyqcqnpVhP4dcasr23bzrXQzUqfUEF7Yl1fo4uoSdUjCtBmCWQvqdAdJYAAAFaIKSD3/K0ff?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=3608 [TCP RST assembled in 14220]
14220	51.152767	204.79.197.203	10.7.45.49	OSPF	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	51.153958	10.7.45.49	20.247.162.75	TLSv1	357	Application Data
14223	51.153958	20.247.162.75	10.7.45.49	TCP	357	Application Data
14224	51.153959	20.247.162.75	10.7.45.49	TCP	357	Application Data
14225	51.154099	10.7.45.49	20.247.162.75	TCP	54	52199 → 443 [ACK] Seq=374 Ack=5540 Win=261632 Len=0
14226	51.154103	10.7.45.49	20.247.162.75	TCP	54	52199 → 443 [ACK] Seq=374 Ack=5543 Win=261632 Len=0
14227	51.154103	10.7.45.49	20.247.162.75	TLSv1	1513	Application Data
14228	51.160930	10.7.45.49	20.247.162.75	TLSv1	1359	Application Data
14229	51.160933	20.247.162.75	10.7.45.49	TCP	60	443 → 52199 [ACK] Seq=5843 Ack=3138 Win=262144 Len=0
14230	51.162879	20.247.162.75	10.7.45.49	TCP	54	52199 → 443 [ACK] Seq=3138 Ack=7125 Win=262144 Len=0
14231	51.160927	20.247.162.75	10.7.45.49	TCP	54	52199 → 443 [ACK] Seq=3138 Ack=7125 Win=262144 Len=0
14232	51.160930	20.247.162.75	10.7.45.49	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 VdeveiwMP...[20A3832E-4593-ACD8-8326-ED1CFB

Ethernet II, Src: Cisco GC-2d27f7 (88:1c:d3:c0:2d:7f), Dst: Intel F675480 (40:50:56:f7:54:80)

Internet Protocol Version 4, Src: 204.79.197.203, Dst: 10.7.45.49

Transmission Control Protocol, Src Port: 80, Dst Port: 52200, Seq: 1

[Reassembly for Segment 2463 (bytes) : #14219(1408), #14220(1053)]

Hypertext Transfer Protocol

Online Certificate Status Protocol

0250

31 37 20 47 54 Sd 0a 0d 0a 80 82 07 3e 08

.....117 GPT .. B7390

0250

01 00 00 82 07 37 20 47 54 Sd 0a 0d 0a 80 82 07 3e 08

.....70D B7390

0250

06 07 37 20 47 54 Sd 0a 0d 0a 80 82 07 3

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	87.460803	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=4000 Ack=2192 Win=262144 Len=0
17942	87.470191	40.126.18.33	10.7.45.49	TCP	60	443 → 60707 [ACK] Seq=4000 Ack=2132 Win=262144 Len=0
17943	87.471038	10.7.45.49	40.78.187.240	TCP	54	52235 → 443 [ACK] Seq=5357 Ack=32838 Win=113128 Len=0
17944	87.467613	40.126.18.33	10.7.45.49	TCP	60	443 → 60707 [ACK] Seq=4000 Ack=3477 Win=262144 Len=0
17945	87.492219	Cisco-ecf19:01	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-ecf19:01	ARP	42	10.7.45.49 is at ad:59:15b:07:54:80
17947	87.442320	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=12
17948	87.442524	10.7.45.49	146.75.122.172	TCP	54	60708 → 80 [ACK] Seq=1 Ack=1 Win=131328 Len=0
17949	87.442321	10.7.45.49	146.75.122.172	HTTP	304	GET /c/ondomload/update/other/2021/01/33356784_31feaf6d5437de7700835ea2958168ae2a306.cab HTTP/1.1
17950	87.471317	10.7.45.49	10.7.63.53	BROUFSR	243	Host Announcement LADY, Workstation, Server, BT Workstation
17951	87.432128	146.75.122.172	10.7.45.49	TCP	60	80 → 60708 [ACK] Seq=1 Ack=251 Win=147456 Len=0
17952	87.433443	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	87.433443	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	87.433443	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	87.433443	146.75.122.172	10.7.45.49	TCP	1510	80 → 60708 [PSH, ACK] Seq=3539 Ack=251 Win=147456 Len=1456 [TCP PDU reassembled in 17957]
17956	87.433443	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	87.433443	146.75.122.172	10.7.45.49	TCP	1510	TCP Out-of-order: 80 → 60708 [ACK] Seq=5425 Ack=251 Win=147456 Len=1456
17958	87.433608	10.7.45.49	146.75.122.172	TCP	54	60708 → 80 [ACK] Seq=251 Ack=2913 Win=131328 Len=0
17959	87.433608	10.7.45.49	146.75.122.172	TCP	54	60708 → 80 [ACK] Seq=251 Ack=3825 Win=131328 Len=0
17960	17.433112	10.7.45.49	146.75.122.172	TCP	66	[TCP Out-of-order] 80 → 60708 [ACK] Seq=251 Ack=5225 Win=131328 Len=0 SILENT: 588-7525
17961	87.433735	10.7.45.49	146.75.122.172	TCP	54	60708 → 80 [ACK] Seq=251 Ack=7525 Win=131328 Len=0
17962	87.435487	10.7.45.49	146.75.122.172	HTTP	304	GET /c/ondomload/update/other/2021/01/33356787_5096408751d3846dc540a1d3f6141af2bbdb8.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=8041 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]
17967	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]

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	288	Application Data
3215	21.148204	10.7.45.49	142.250.66.10	UDP	1283	61312 → 443 Len=1239
3216	21.148675	10.7.45.49	216.58.196.106	UDP	76	61928 → 443 Len=34
3217	21.182380	142.250.66.10	10.7.45.49	UDP	75	443 → 62132 Len=33
3218	21.182380	10.7.45.49	142.250.66.10	TCP	66	80 → 60708 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 SACK_PERM WS=4
3219	21.182448	10.7.45.49	142.250.66.10	UDP	75	61312 → 443 Len=33
3220	21.187591	10.7.45.49	10.240.8.33	TCP	66	8000 → 50706 [PSH, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 SACK_PERM WS=4
3221	21.187651	10.7.45.49	10.240.8.33	TCP	54	50706 → 8009 [ACK] Seq=1 Ack=1 Win=132352 Len=0
3222	21.193110	10.7.45.49	10.240.8.33	HTTP	1402	5013 Error
3223	21.193110	10.240.8.33	10.7.45.49	TCP	54	8009 → 50706 [ACK] Seq=1 Ack=1409 Win=60480 Len=0
3224	21.193777	10.240.8.33	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=22
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	21.197881	10.240.8.33	10.7.45.49	AJP13	1202	AJP13 Error?
3228	21.195608	10.240.8.33	10.7.45.49	TCP	147	50706 → 8009 [PSH, ACK] Seq=1739 Ack=1149 Win=131072 Len=93 [TCP PDU reassembled in 3231]
3229	21.247449	10.240.8.33	10.7.45.49	TCP	54	8009 → 50706 [ACK] Seq=1149 Ack=1832 Win=71296 Len=0
3230	21.249597	10.240.8.33	10.7.45.49	TCP	200	8009 → 50706 [PSH, ACK] Seq=1149 Ack=1832 Win=71296 Len=226
3231	21.250648	10.7.45.49	10.240.8.33	AJP13	179	AJP13 Error?
3232	21.256928	10.240.8.33	10.7.45.49	TCP	54	8009 → 50706 [ACK] Seq=1375 Ack=1957 Win=71296 Len=0
3233	21.256928	10.240.8.33	10.7.45.49	TCP	1462	8009 → 50706 [ACK] Seq=1375 Ack=1957 Win=71296 Len=1400
3234	21.256928	10.240.8.33	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	10.240.8.33	TCP	54	50706 → 8009 [ACK] Seq=1957 Ack=5597 Win=132352 Len=0
3236	21.258280	10.7.45.49	10.240.8.33	TCP	512	50706 → 8009 [PSH, ACK] Seq=1957 Ack=5597 Win=132352 Len=458
3237	21.258378	10.7.45.49	10.240.8.33	TCP	195	50706 → 8009 [PSH, ACK] Seq=2415 Ack=5597 Win=132352 Len=141

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

Time	Source	Destination	Protocol	Length	Info
1 0.00000	Cisco.cc:19:01	Intel.67:54:80	ARP	60	who has 10.7.45.49? Tell 0.0.0.0
2 0.00005	Intel.67:54:80	Cisco.cc:19:01	ARP	42	10.7.45.49 is at 00:09:50:67:54:80
3 1.4779705	104.215.40.100	104.215.40.100	TLSv1.2	183	Application Data
4 3.49949	104.215.40.100	107.45.49	TCP	60	443 → [ESTAB] [ACK] Seq=14650 Win=4096 Len=0
5 3.610105	104.215.40.100	107.45.49	TLSv1.2	92	Application Data
6 3.610104	107.45.49	104.215.40.100	TCP	54	64343 → 443 [ACK] Seq=508 Acks=39 Win=580 Len=0
7 3.6129734	Intel.67:54:80	Intel.67:54:80	CHNMG	17	CHNMG: I have 10.7.45.49
8 3.6129734	107.45.49	104.215.40.100	DNS	87	Standard query 0x2048 0 assets.activity.windows.com
9 4.786273	104.215.40.100	107.45.49	DNS	545	Standard query response 0x2048 0 assets.activity.windows.com CHNMG asset-edge-trafficmanager.net A 20.44.229.37 NS a.root-servers.net NS b.root-servers.net NS c.root-servers.net NS d.root-servers.net NS e.root-servers.net NS f.root-servers.net NS g.root-servers.net NS h.root-servers.net NS i.root-servers.net NS j.root-servers.net NS k.root-servers.net NS l.root-servers.net NS m.root-servers.net NS n.root-servers.net NS o.root-servers.net NS p.root-servers.net NS q.root-servers.net NS r.root-servers.net NS s.root-servers.net NS t.root-servers.net NS u.root-servers.net NS v.root-servers.net NS w.root-servers.net NS x.root-servers.net NS y.root-servers.net NS z.root-servers.net NS .root-servers.net
10 4.786272	107.45.49	20.44.229.37	TCP	66	20487 → 443 [SYN] Seq=14650400 Len=0 MSS=1460 SACK_PERM=1
11 4.792677	104.224.229.37	107.45.49	TCP	66	443 → 52437 [SYN, ACK] Seq=0 Acks=115553 Len=0 MSS=1460 SACK_PERM=1
12 4.792769	107.45.49	20.44.229.37	TCP	54	52437 → 443 [ACK] Seq=1 Acks=115328 Len=0
13 4.792810	107.45.49	20.44.229.37	TLSv1.3	343	Client Hello (0xassets.activity.windows.com)
14 4.796115	20.44.229.37	107.45.49	TCP	60	[TCP Window update] 443 → 52437 [ACK] Seq=1 Acks=1 Win=262144 Len=0
15 4.805734	20.44.229.37	107.45.49	TCP	443 → 52437 [ACK] Seq=1 Acks=290 Win=261120 Len=0	
16 5.009639	107.45.49	10.7.45.49	DNS	78	Standard query 0x5020 0 edge.microsoft.com
17 5.009639	107.45.49	10.7.45.49	DNS	78	Standard query 0x1414 HTTPS edge.microsoft.com
18 5.009639	10.7.45.49	10.7.45.49	DNS	537	Standard query response 0x5020 0 edge.microsoft.com CHNMG edge-microsoft-com.dual-a-0036-a-sedge.net CHNMG dual-a-0036-a-sedge.net A 13.107.21.239 A 204.79.197.259 NS b.root-servers.net
19 5.012106	10.7.45.49	107.45.49	DNS	66	Standard query response 0x1414 HTTPS edge.microsoft.com CHNMG edge-microsoft-com.dual-a-0036-a-sedge.net CHNMG dual-a-0036-a-sedge.net SOA ns1.a-sedge.net
20 5.012106	107.45.49	10.7.45.49	TCP	66	52438 → 443 [SYN] Seq=1461 Acks=115553 Len=0 MSS=1460 SACK_PERM=1
21 5.015518	13.107.21.239	107.45.49	TCP	66	443 → 52438 [SYN, ACK] Seq=1 Acks=115553 Len=0 MSS=1460 SACK_PERM=1
22 5.015567	107.45.49	13.107.21.239	TCP	54	52438 → 443 [ACK] Seq=1 Acks=115328 Len=0
23 5.015568	107.45.49	13.107.21.239	TLSv1.2	1814	Client Hello (0xedge.microsoft.com)
24 5.018811	13.107.21.239	107.45.49	TCP	60	[TCP Window update] 443 → 52438 [ACK] Seq=1 Acks=1 Win=262144 Len=0
25 5.018811	13.107.21.239	107.45.49	TCP	643 → 52438 [ACK] Seq=1 Acks=1761 Win=208350 Len=0	
26 5.041570	13.107.21.239	107.45.49	TCP	1514	443 → 52438 [PSH, ACK] Seq=1 Acks=1761 Win=262144 Len=1460 [TCP PSH ressembled in 30]
27 5.042977	13.107.21.239	107.45.49	TCP	1514	443 → 52438 [PSH, ACK] Seq=1 Acks=1761 Win=262144 Len=1460 [TCP PSH ressembled in 30]
28 5.042977	107.45.49	13.107.21.239	TCP	1514	443 → 52438 [PSH, ACK] Seq=1 Acks=1761 Win=262144 Len=1460 [TCP PSH ressembled in 30]
29 5.042977	107.45.49	13.107.21.239	TCP	1514	443 → 52438 [PSH, ACK] Seq=1 Acks=1761 Win=262144 Len=1460 [TCP PSH ressembled in 30]
30 5.042977	107.45.49	13.107.21.239	TCP	1514	443 → 52438 [PSH, ACK] Seq=1 Acks=1761 Win=262144 Len=1460 [TCP PSH ressembled in 30]
31 5.042977	107.45.49	13.107.21.239	TCP	1514	443 → 52438 [PSH, ACK] Seq=1 Acks=1761 Win=262144 Len=1460 [TCP PSH ressembled in 30]
32 5.042977	107.45.49	13.107.21.239	TCP	1514	443 → 52438 [PSH, ACK] Seq=1 Acks=1761 Win=262144 Len=1460 [TCP PSH ressembled in 30]
33 5.042977	107.45.49	13.107.21.239	TCP	1514	443 → 52438 [PSH, ACK] Seq=1 Acks=1761 Win=262144 Len=1460 [TCP PSH ressembled in 30]
34 5.042977	107.45.49	13.107.21.239	TCP	1514	443 → 52438 [PSH, ACK] Seq=1 Acks=1761 Win=262144 Len=1460 [TCP PSH ressembled in 30]
35 5.042977	107.45.49	13.107.21.239	TCP	1514	443 → 52438 [PSH, ACK] Seq=1 Acks=1761 Win=262144 Len=1460 [TCP PSH ressembled in 30]
36 5.042977	107.45.49	13.107.21.239	TCP	1514	443 → 52438 [PSH, ACK] Seq=1 Acks=1761 Win=262144 Len=1460 [TCP PSH ressembled in 30]
37 5.042977	107.45.49	13.107.21.239	TCP	1514	443 → 52438 [PSH, ACK] Seq=1 Acks=1761 Win=262144 Len=1460 [TCP PSH ressembled in 30]
38 5.042977	107.45.49	13.107.21.239	TCP	1514	443 → 52438 [PSH, ACK] Seq=1 Acks=1761 Win=262144 Len=1460 [TCP PSH ressembled in 30]
39 5.042977	107.45.49	13.107.21.239	TCP	1514	443 → 52438 [PSH, ACK] Seq=1 Acks=1761 Win=262144 Len=1460 [TCP PSH ressembled in 30]
40 5.042977	107.45.49	13.107.21.239	TCP	1514	443 → 52438 [PSH, ACK] Seq=1 Acks=1761 Win=262144 Len=1460 [TCP PSH ressembled in 30]
41 5.042977	107.45.49	13.107.21.239	TCP	1514	443 → 52438 [PSH, ACK] Seq=1 Acks=1761 Win=262144 Len=1460 [TCP PSH ressembled in 30]
42 5.042977	107.45.49	13.107.21.239	TCP	1514	443 → 52438 [PSH, ACK] Seq=1 Acks=1761 Win=262144 Len=1460 [TCP PSH ressembled in 30]
43 5.042977	107.45.49	13.107.21.239	TCP	1514	443 → 52438 [PSH, ACK] Seq=1 Acks=1761 Win=262144 Len=1460 [TCP PSH ressembled in 30]
44 5.042977	107.45.49	13.107.21.239	TCP	1514	443 → 52438 [PSH, ACK] Seq=1 Acks=1761 Win=262144 Len=1460 [TCP PSH ressembled in 30]
45 5.042977	107.45.49	13.107.21.239	TCP	1514	443 → 52438 [PSH, ACK] Seq=1 Acks=1761 Win=262144 Len=1460 [TCP PSH ressembled in 30]
46 5.042977	107.45.49	13.107.21.239	TCP	1514	443 → 52438 [PSH, ACK] Seq=1 Acks=1761 Win=262144 Len=1460 [TCP PSH ressembled in 30]
47 5.042977	107.45.49	13.107.21.239	TCP	1514	443 → 52438 [PSH, ACK] Seq=1 Acks=1761 Win=262144 Len=1460 [TCP PSH ressembled in 30]
48 5.042977	107.45.49	13.107.21.239	TCP	1514	443 → 52438 [PSH, ACK] Seq=1 Acks=1761 Win=262144 Len=1460 [TCP PSH ressembled in 30]
49 5.042977	107.45.49	13.107.21.239	TCP	1514	443 → 52438 [PSH, ACK] Seq=1 Acks=1761 Win=262144 Len=1460 [TCP PSH ressembled in 30]
50 5.042977	107.45.49	13.107.21.239	TCP	1514	443 → 52438 [PSH, ACK] Seq=1 Acks=1761 Win=262144 Len=1460 [TCP PSH ressembled in 30]
51 5.042977	107.45.49	13.107.21.239	TCP	1514	443 → 52438 [PSH, ACK] Seq=1 Acks=1761 Win=262144 Len=1460 [TCP PSH ressembled in 30]
52 5.042977	107.45.49	13.107.21.239	TCP	1514	443 → 52438 [PSH, ACK] Seq=1 Acks=1761 Win=262144 Len=1460 [TCP PSH ressembled in 30]
53 5.042977	107.45.49	13.107.21.239	TCP	1514	443 → 52438 [PSH, ACK] Seq=1 Acks=1761 Win=262144 Len=1460 [TCP PSH ressembled in 30]
54 5.042977	107.45.49	13.107.21.239	TCP	1514	443 → 52438 [PSH, ACK] Seq=1 Acks=1761 Win=262144 Len=1460 [TCP PSH ressembled in 30]
55 5.042977	107.45.49	13.107.21.239	TCP	1514	443 → 52438 [PSH, ACK] Seq=1 Acks=1761 Win=262144 Len=1460 [TCP PSH ressembled in 30]
56 5.042977	107.45.49	13.107.21.239	TCP	1514	443 → 52438 [PSH, ACK] Seq=1 Acks=1761 Win=262144 Len=1460 [TCP PSH ressembled in 30]
57 5.042977	107.45.49	13.107.21.239	TCP	1514	443 → 52438 [PSH, ACK] Seq=1 Acks=1761 Win=262144 Len=1460 [TCP PSH ressembled in 30]
58 5.042977	107.45.49	13.107.21.239	TCP	1514	443 → 52438 [PSH, ACK] Seq=1 Acks=1761 Win=262144 Len=1460 [TCP PSH ressembled in 30]
59 5.042977	107.45.49	13.107.21.239	TCP	1514	443 → 52438 [PSH, ACK] Seq=1 Acks=1761 Win=262144 Len=1460 [TCP PSH ressembled in 30]
60 5.042977	107.45.49	13.107.21.239	TCP	1514	443 → 52438 [PSH, ACK] Seq=1 Acks=1761 Win=262144 Len=1460 [TCP PSH ressembled in 30]
61 5.042977	107.45.49	13.107.21.239	TCP	1514	443 → 52438 [PSH, ACK] Seq=1 Acks=1761 Win=262144 Len=1460 [TCP PSH ressembled in 30]
62 5.042977	107.45.49	13.107.21.239	TCP	1514	443 → 52438 [PSH, ACK] Seq=1 Acks=1761 Win=262144 Len=1460 [TCP PSH ressembled in 30]
63 5.042977	107.45.49	13.107.21.239	TCP	1514	443 → 52438 [PSH, ACK] Seq=1 Acks=1761 Win=262144 Len=1460 [TCP PSH ressembled in 30]
64 5.042977	107.45.49	13.107.21.239	TCP	1514	443 → 52438 [PSH, ACK] Seq=1 Acks=1761 Win=262144 Len=1460 [TCP PSH ressembled in 30]
65 5.042977	107.45.49	13.107.21.239	TCP	1514	443 → 52438 [PSH, ACK] Seq=1 Acks=1761 Win=262144 Len=1460 [TCP PSH ressembled in 30]
66 5.042977	107.45.49	13.107.21.239	TCP	1514	443 → 52438 [PSH, ACK] Seq=1 Acks=1761 Win=262144 Len=1460 [TCP PSH ressembled in 30]
67 5.042977	107.45.49	13.107.21.239	TCP	1514	443 → 52438 [PSH, ACK] Seq=1 Acks=1761 Win=262144 Len=1460 [TCP PSH ressembled in 30]
68 5.042977	107.45.49	13.107.21.239	TCP	1514	443 → 52438 [PSH, ACK] Seq=1 Acks=1761 Win=262144 Len=1460 [TCP PSH ressembled in 30]
69 5.042977	107.45.49	13.107.21.239	TCP	1514	443 → 52438 [PSH, ACK] Seq=1 Acks=1761 Win=262144 Len=1460 [TCP PSH ressembled in 30]
70 5.042977	107.45.49	13.107.21.239	TCP	1514	443 → 52438 [PSH, ACK] Seq=1 Acks=1761 Win=262144 Len=1460 [TCP PSH ressembled in 30]
71 5.042977	107.45.49	13.107.21.239	TCP	1514	443 → 52438 [PSH, ACK] Seq=1 Acks=1761 Win=262144 Len=1460 [TCP PSH ressembled in 30]
72 5.042977	107.45.49	13.107.21.239	TCP	1514	443 → 52438 [PSH, ACK] Seq=1 Acks=1761 Win=262144 Len=1460 [TCP PSH ressembled in 30]
73 5.042977	107.45.49	13.107.21.239	TCP	1514	443 → 52438 [PSH, ACK] Seq=1 Acks=1761 Win=262144 Len=1460 [TCP PSH ressembled in 30]
74 5.042977	107.45.49	13.107.21.239	TCP	1514	443 → 52438 [PSH, ACK] Seq=1 Acks=1761 Win=262144 Len=1460 [TCP PSH ressembled in 30]
75 5.042977	107.45.49	13.107.21.239	TCP	1514	443 → 52438 [PSH, ACK] Seq=1 Acks=1761 Win=262144 Len=1460 [TCP PSH ressembled in 30]
76 5.042977	107.45.49	13.107.21.239	TCP	1514	443 → 52438 [PSH, ACK] Seq=1 Acks=1761 Win=262144 Len=1460 [TCP PSH ressembled in 30]
77 5.042977	107.45.49	13.107.21.239	TCP	1514	443 → 52438 [PSH, ACK] Seq=1 Acks=1761 Win=262144 Len=1460 [TCP PSH ressembled in 30]
78 5.042977	107.45.49	13.107.21.239	TCP	1514	443 → 52438 [PSH, ACK] Seq=1 Acks=1761 Win=262144 Len=1460 [TCP PSH ressembled in 30]
79 5.042977	107.45.49	13.107.21.239	TCP	1514	443 → 52438 [PSH, ACK] Seq=1 Acks=1761 Win=262144 Len=1460 [TCP PSH ressembled in 30]
80 5.042977	107.45.49	13.107.21.239	TCP	1514	443 → 52438 [PSH, ACK] Seq=1 Acks=1761 Win=262144 Len=1460 [TCP PSH ressembled in 30]
81 5.042977	107.45.49	13.107.21.239	TCP	1514	443 → 52438 [PSH, ACK] Seq=1 Acks=1761 Win=262144 Len=1460 [TCP PSH ressembled in 30]
82 5.042977	107.45.49	13.107.21.239	TCP	1514	443 → 52438 [PSH, ACK] Seq=1 Acks=1761 Win=262144 Len=1460 [TCP PSH ressembled in 30]
83 5.042977	107.45.49	13.107.21.239	TCP	1514	443 → 52438 [PSH, ACK] Seq=1 Acks=1761 Win=262144 Len=1460 [TCP PSH ressembled in 30]
84 5.042977	107.45.49	13.107.21.239	TCP	1514	443 → 52438 [PSH, ACK] Seq=1 Acks=1761 Win=262144 Len=1460 [TCP PSH ressembled in 30]
85 5.042977	107.45.49	13.107.21.239	TCP	1514	443 → 52438 [PSH, ACK] Seq=1 Acks=1761 Win=262144 Len=1460 [TCP PSH ressembled in 30]
86 5.042977	107.45.49	13.107.21.239	TCP	1514	443 → 52438 [PSH, ACK] Seq=1 Acks=1761 Win=262144 Len=1460 [TCP PSH ressembled in 30]
87 5.042977	107.45.49	13.107.21.239	TCP	1514	443 → 52438 [PSH, ACK] Seq=1 Acks=1761 Win=262144 Len=1460 [TCP PSH ressembled in 30]
88 5.042977	107.45.49	13.107.21.239	TCP	1514	443 → 52438 [PSH, ACK] Seq=1 Acks=1761 Win=262144 Len=1460 [TCP PSH ressembled in 30]
89 5.042977	107.45.49	13.107.21.239	TCP	1514	443 → 52438 [PSH, ACK] Seq=1 Acks=1761 Win=262144 Len=1460 [TCP PSH ressembled in 30]
90 5.042977	107.45.49	13.107.21.239	TCP	1514	443 → 52438 [PSH, ACK] Seq=1 Acks=1761 Win=262144 Len=1460 [TCP PSH ressembled in 30]
91 5.042977	107.45.49	13.107.21.239	TCP	1514	443 → 52438 [PSH, ACK] Seq=1 Acks=1761 Win=262144 Len=1460 [TCP PSH ressembled in 30]
92 5.042977	107.45.49	13.107.21.239	TCP	1514	443 → 52438 [PSH, ACK] Seq=1 Acks=1761 Win=262144 Len=1460 [TCP PSH ressembled in 30]
93 5.042977	107.45.49	13.107.21.239	TCP	1514	443 → 52438 [PSH, ACK] Seq=1 Acks=1761 Win=262144 Len=1460 [TCP PSH ressembled in 30]
94 5.042977	107.45.49	13.107.21.239	TCP	1514	443 → 52438 [PSH, ACK] Seq=1 Acks=1761 Win=262144 Len=1460 [TCP PSH ressembled in 30]
95 5.042977	107.45.49	13.107.21.239	TCP	1514	443 → 52438 [PSH, ACK] Seq=1 Acks=1761 Win=262144 Len=1460 [TCP PSH ressembled in 30]
96 5.042977	107.45.49	13.107.21.239	TCP	1514	443 → 52438 [PSH, ACK] Seq=1 Acks=1761 Win=262144 Len=1460 [TCP PSH ressembled in 30]
97 5.042977	107.45.49	13.107.21.239	TCP	1514	443 → 52438 [PSH, ACK] Seq=1 Acks=1761 Win=262144 Len=1460 [TCP PSH ressembled in 30]
98 5.042977	107.45.49	13.107.21.239	TCP	1514	443 → 52438 [PSH, ACK] Seq=1 Acks=1761 Win=262144 Len=1460 [TCP PSH ressembled in 30]
99 5.042977	107.45.49	13.107.21.239	TCP	1514	443 → 52438 [PSH, ACK] Seq=1 Acks=1761 Win=262144 Len=1460 [TCP PSH ressembled in 30]
100 5.042977	107.45.49	13.107.21.239	TCP	1514	443 → 52438 [PSH, ACK] Seq=1 Acks=1761 Win=262144 Len=1460 [TCP PSH ressembled in 30]

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=590 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	1784	Application Data
68	5.241934	10.7.45.49	20.44.229.37	TLSv1.3	818	Application Data
69	5.242436	10.7.45.49	20.44.229.37	TCP	60	443 → 52437 [ACK] Seq=6182 Ack=2184 Win=262144 Len=0
70	5.244316	10.7.45.49	20.44.229.37	TCP	60	443 → 52437 [ACK] Seq=6182 Ack=5180 Win=262144 Len=0
71	5.395231	20.44.229.37	10.7.45.49	TLSv1.3	157	Application Data
72	5.123035	10.7.45.49	20.44.229.37	TLSv1.3	145	Application Data
73	5.132796	10.7.45.49	20.44.229.37	TCP	54	52437 → 443 [ACK] Seq=3188 Ack=6376 Win=130816 Len=0
74	5.134385	10.7.45.49	20.44.229.37	TLSv1.3	168	Application Data
75	5.303234	20.44.229.37	10.7.45.49	TCP	60	443 → 52437 [ACK] Seq=6376 Ack=3382 Win=262144 Len=0
76	5.384438	20.44.229.37	10.7.45.49	TLSv1.3	157	Application Data
77	5.428991	10.7.45.49	20.44.229.37	TCP	54	52437 → 443 [ACK] Seq=3382 Ack=6479 Win=130816 Len=0
78	5.431073	fe80::e884::affc::16	fe80::116	ICMPv6	90	Multicast Listener Report Message v2
79	5.432461	10.7.45.49	224.0.0.2	IGMPv3	54	Membership Report / Join group 239.255.255.258 for any sources
80	5.437475	10.7.45.49	239.255.255.258	SSDP	179	M-SEARCH * HTTP/1.1
81	5.447577	10.7.45.49	239.255.255.258	SSDP	179	M-SEARCH * HTTP/1.1
82	5.666176	10.7.45.49	239.255.255.258	SSDP	179	M-SEARCH * HTTP/1.1
83	5.838087	10.7.45.49	224.0.0.22	IGMPv3	54	Membership Report / Join group 239.255.255.258 for any sources
84	5.838283	fe80::e884::affc::16	fe80::116	ICMPv6	90	Multicast Listener Report Message v2
85	6.555532	10.7.45.49	10.136.7.6	DNS	96	Standard query 0xb92d A function.events.data.microsoft.com
86	6.555881	10.7.45.49	10.136.7.6	DNS	96	Standard query 0xb998 HTTPS function.events.data.microsoft.com
87	6.618935	10.136.7.6	10.7.45.49	DNS	268	Standard query response 0xb92d A function.events.data.microsoft.com CNAME global.asimov.events.data.trafficmanager.net CNAME onedocsprodus1.eastus.cloudapp.azure.com SOA n1-201-az
88	6.618936	10.136.7.6	10.7.45.49	DNS	135	Standard query response 0xb998 A function.events.data.microsoft.com CNAME global.asimov.events.data.trafficmanager.net CNAME onedocsprodus1.eastus.cloudapp.azure.com A 20.42.73.30 NS
89	6.619172	10.7.45.49	20.42.73.30	TCP	66	52441 → 443 [SYN] Seq=6000420 Len=0 Win=65536 SACK_PERM
90	6.660286	20.42.73.30	10.7.45.49	TCP	66	443 → 52441 [ACK] Seq=61400 Len=0 Win=65536 SACK_PERM
91	6.624441	10.7.45.49	20.42.73.30	TCP	60	443 → 443 [ACK] Seq=61400 Len=0 Win=131328 Len=0
92	6.660308	10.7.45.49	20.42.73.30	TCP	60	443 → 443 [ACK] Seq=61400 Len=0 Win=131328 Len=0

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

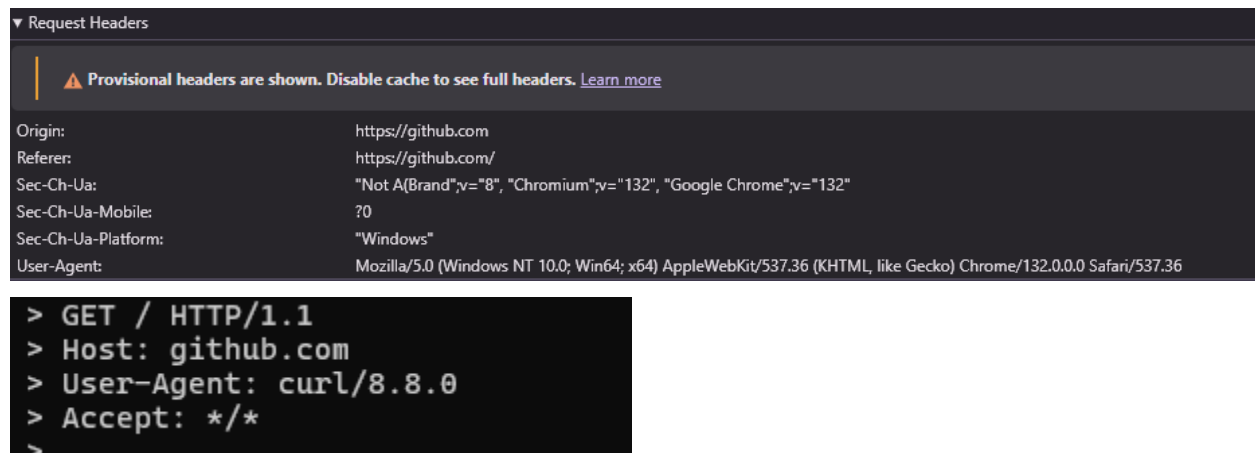
iii) netflix.com

```
(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:ac92:667b: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=BQFmAAEBEDCuHj6Vgx4DeFIkHdEvBZAoo4rwpPIp63gz96X-bjji_QpkQnvaIQ3iowHnFS4TTr0b8chu_2N655ALVn0-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 :



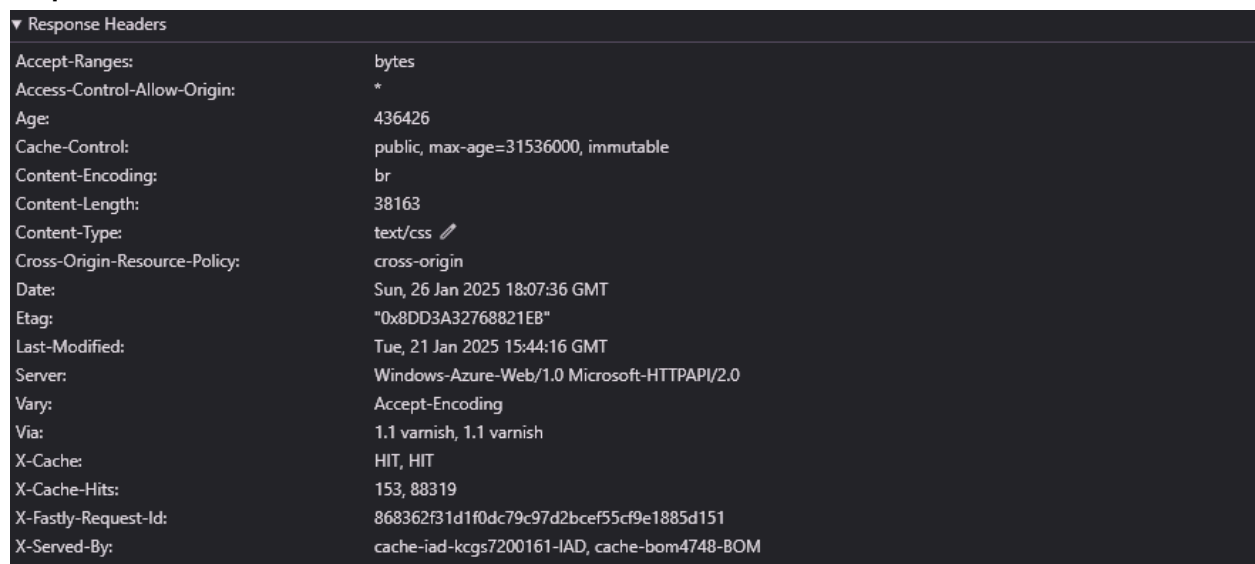
▼ Request Headers

⚠ 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

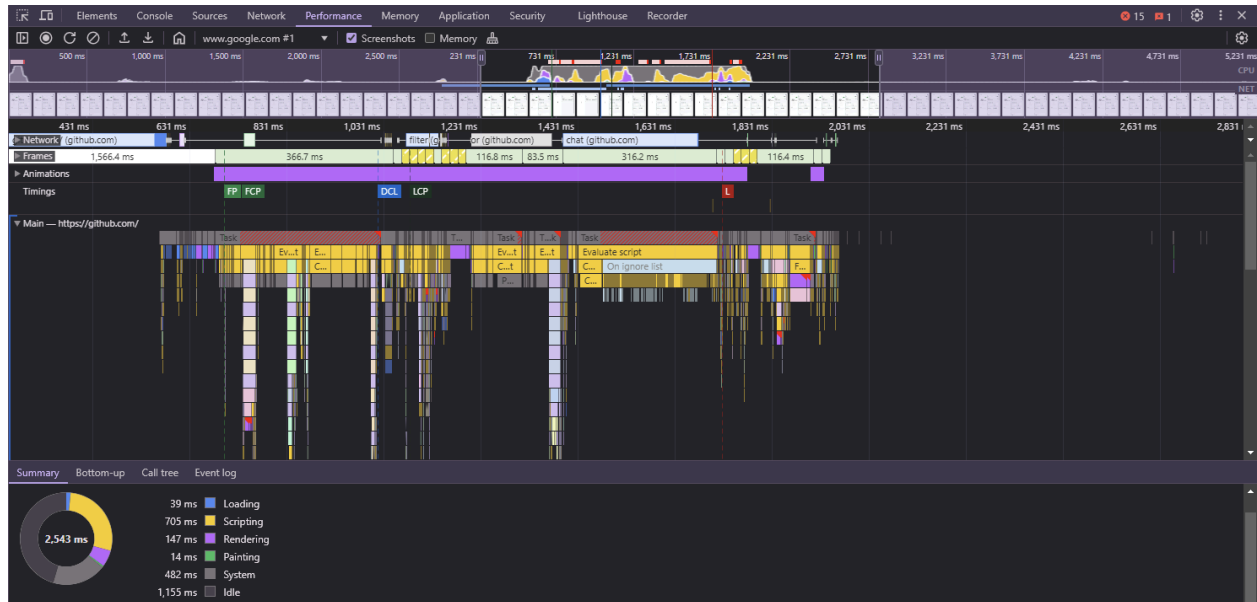
HTTP Error Codes

- **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 :



List of Cookies :

Name	Value	Domain	Path	Expires / ...	Size	HttpOnly	Secure	SameSite	Partition ...	Cross Site	Priority
__Host-user_session_same_site	gkT4Cy28wno5mzl-NcT4l1J8nOCwsfgpJ9_VIOr-QS09F6Y	github.co...	/	2025-02-...	77	✓	✓	Strict			Medium
_device_id	4606a093f8e97635719003ca39efa867	github.co...	/	2026-02-...	42	✓	✓	Lax			Medium
_gh_sess	i9U8ueN88xUw4pbUwR%K28f6%2Btho2ULk8BACxqA5phIP...	github.co...	/	Session	462	✓	✓	Lax			Medium
_ecto	GH1.1.1429292755.1729663225	github.co...	/	2025-10-...	32		✓	Lax			Medium
color_mode	%78%22color_mode%22%3A%22auto%22%3C%22light_the...	github.co...	/	Session	216		✓	Lax			Medium
cpu_bucket	xlg	github.co...	/	Session	13		✓	Lax			Medium
dotcom_user	pranjal15195gaur	github.co...	/	2026-01-...	27	✓	✓	Lax			Medium
logged_in	yes	github.co...	/	2026-01-...	12	✓	✓	Lax			Medium
preferred_color_mode	dark	github.co...	/	Session	24		✓	Lax			Medium
saved_user_sessions	136556627%3AgkT4Cy28wno5mzl-NcT4l1J8nOCwsfgpJ9_V...	github.co...	/	2025-04-...	79	✓	✓	Lax			Medium
tz	Asia%2FCalcutta	github.co...	/	Session	17		✓	Lax			Medium
user_session	gkT4Cy28wno5mzl-NcT4l1J8nOCwsfgpJ9_VIOr-QS09F6Y	github.co...	/	2025-02-...	60	✓	✓	Lax			Medium