

# Computer Networks Assignment-1

1.02.2024

Birudugadda Srivibhav (22110050)

Srivaths Vamsi Chaturvedula (22110260)

Computer Science and Engineering

IIT Gandhinagar

## Introduction

We have created a packet sniffer using the Python `socket` library, `dpkt`, and other utilities. The sniffer captures network packets, analyzes traffic, and provides insights into network activity. Github\_link for sniffer code: <https://github.com/Sparky1743/CN-Assignments>

## Part-1

In this section, we analyze the data captured during the replay of a PCAP file using `tcpreplay` and a custom Python sniffer. We executed the experiment on two different Ubuntu Linux versions (20.04 and 24.04) and ensured that both Wi-Fi and Bluetooth were disabled during the data transfer.

## Experimental Setup

- **Sender (Ubuntu 20.04):** Used `tcpreplay` to send the PCAP file (`2.pcap`).
- **Receiver (Ubuntu 24.04):** Used our Python sniffer program to capture the traffic.
- **Network Interface:** Ethernet (`eth0`).

### Commands Used:

- **Sender:**

```
birud_ubuntu_20.04@chinnu:~$ sudo tcpreplay --pps=10000 -i eth0 2.pcap
```

- **Receiver:**

```
(base) birud_ubuntu_24.04@chinnu:~/CN/Computer-Networks/Packet-Sniffer$ sudo /home/birud_ubuntu/miniconda3/envs/cn/bin/python3 python_sniffer.py -i eth0 -d 10000 -p capture.pcap
```

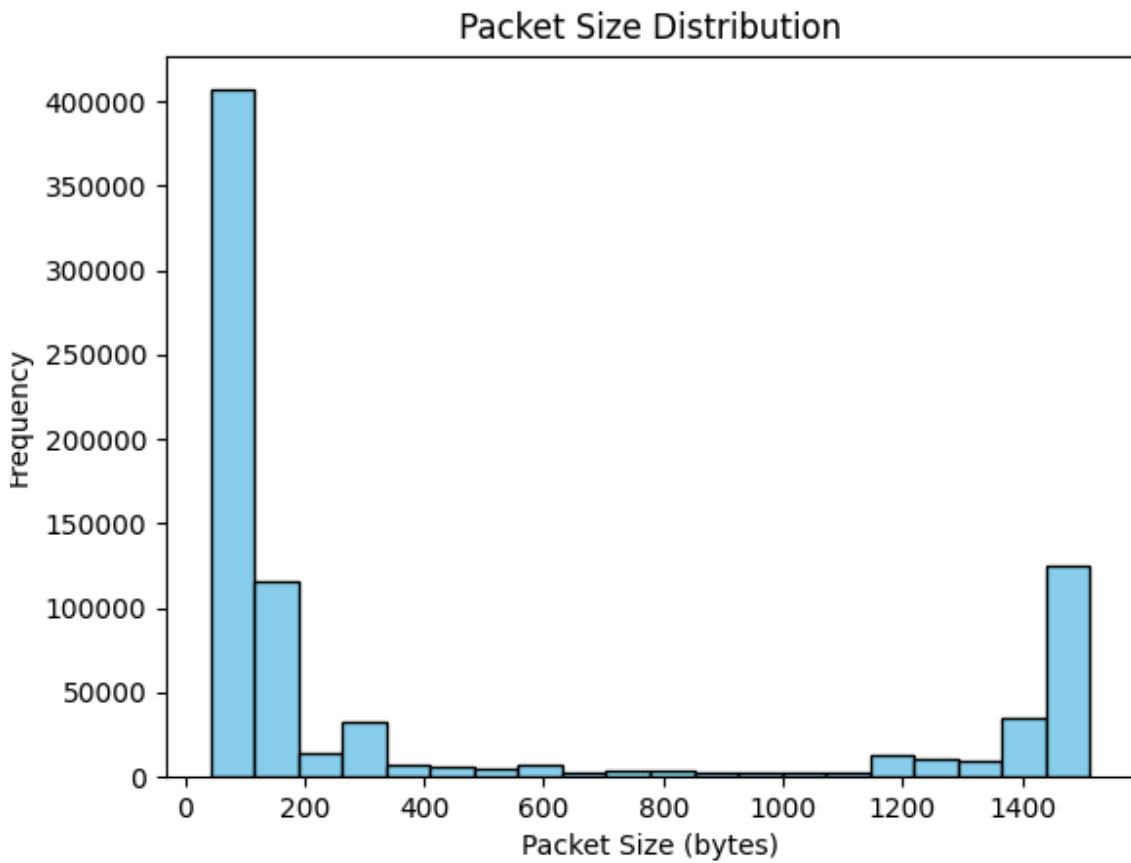


## 1) Original 2.pcap analysis

In this section, we analyze the statistics in the original `2.pcap` file and then compare these statistics with the results obtained using our sniffer code.

```
Total Data: 364641996 bytes
Total Packets: 805997
Min Packet Size: 42 bytes
Max Packet Size: 1514 bytes
Average Packet Size: 452.41 bytes
```

Histogram of packet size distribution



```
Unique Source-Destination Pairs: 41899
('96.43.146.22:443', '172.16.133.29:60614')
('172.16.133.116:53504', '172.16.139.250:5440')
('172.16.133.84:58955', '172.16.139.250:5440')
('96.43.146.22:443', '172.16.133.63:54135')
('8.8.8.8:53', '172.16.133.6:63245')
('172.16.133.67:55957', '172.16.139.250:5440')
('172.16.128.169:4449', '172.16.133.248:161')
('172.16.128.169:4400', '172.16.133.242:161')
('172.16.133.41:52978', '172.16.139.250:5440')
('66.220.149.32:80', '192.168.3.131:56048')
```

The complete data of unique source-destination pairs for the original 2.pcap file can be found in Original\_2.pcap\_statistics/part1\_step2.txt in the GitHub repository.

<b>Destination IP -&gt; Total Flows:</b> 192.168.3.131: 6184 10.0.2.15: 808 207.46.0.109: 29 65.54.95.68: 664 172.16.255.1: 1219 204.14.234.85: 740 65.54.95.140: 495 172.16.0.1: 10 147.31.122.1: 90 66.235.139.121: 39 109.227.83.224: 36	<b>Source IP -&gt; Total Flows:</b> 65.54.95.68: 1275 65.54.95.75: 766 65.54.95.140: 658 204.14.234.85: 1036 65.54.186.19: 66 192.168.3.131: 4294 72.14.213.105: 6 65.54.189.173: 126 207.46.0.109: 63 184.85.226.161: 222
--	--

The complete data of the total flows can be found in Original\_2.pcap\_statistics/part1\_step3.txt in the GitHub repository.

```
Source-Destination Pair Transferring the Most Data:  

('172.16.133.95:49358', '157.56.240.102:443'): 17342229 bytes
```

## 2) Captured data analysis

In our approach, we analyze the captured network traffic in real time while simultaneously storing the captured packets in [capture.pcap](#) for further analysis in Part 2. This enables us to monitor and process network statistics dynamically as the program is running, ensuring immediate insights into data transfer patterns.

Once the user decides to stop sniffing, they need to press [Ctrl+C](#). At this point, the program prints the final statistics and transitions into analysis mode, providing the following options:

- Press '**a**' to analyze packets.
- Press '**h**' to generate a histogram.
- Press '**u**' to display unique source-destination pairs.
- Press '**m**' to analyze flows.
- Press '**q**' to quit.

This interactive approach ensures that users can further investigate captured data efficiently while maintaining real-time monitoring capabilities.

```
birud_ubuntu_20.04@chinnu:~$ sudo tcpreplay --pps=10000 -i eth0 2.pcap
Actual: 805997 packets (364642055 bytes) sent in 80.59 seconds
Rated: 4524116.1 Bps, 36.19 Mbps, 10000.00 pps
Statistics for network device: eth0
    Successful packets:          805997
    Failed packets:              0
    Truncated packets:           0
    Retried packets (ENOBUFS):   0
    Retried packets (EAGAIN):    0
```

```
Current PPS: 10000.07, Mbps: 34.10
Current PPS: 10000.07, Mbps: 34.05
Current PPS: 10000.07, Mbps: 34.25
Current PPS: 10000.22, Mbps: 34.40
No packets received in the last second, continuing...
^C
Capture interrupted by user. Printing final statistics...

Final Statistics:
Total Packets: 806013
Total Bytes Transferred: 364644415 bytes
Average PPS: 9450.70
Average Mbps: 32.62
Total Duration: 85.29 seconds
Peak PPS: 12483.05
Peak Mbps: 127.42

Press 'a' to analyze packets, 'h' to make a histogram, 'u' for unique source-destination pairs, 'm' to analyze
flows, or 'q' to quit.
Enter your choice: 
```

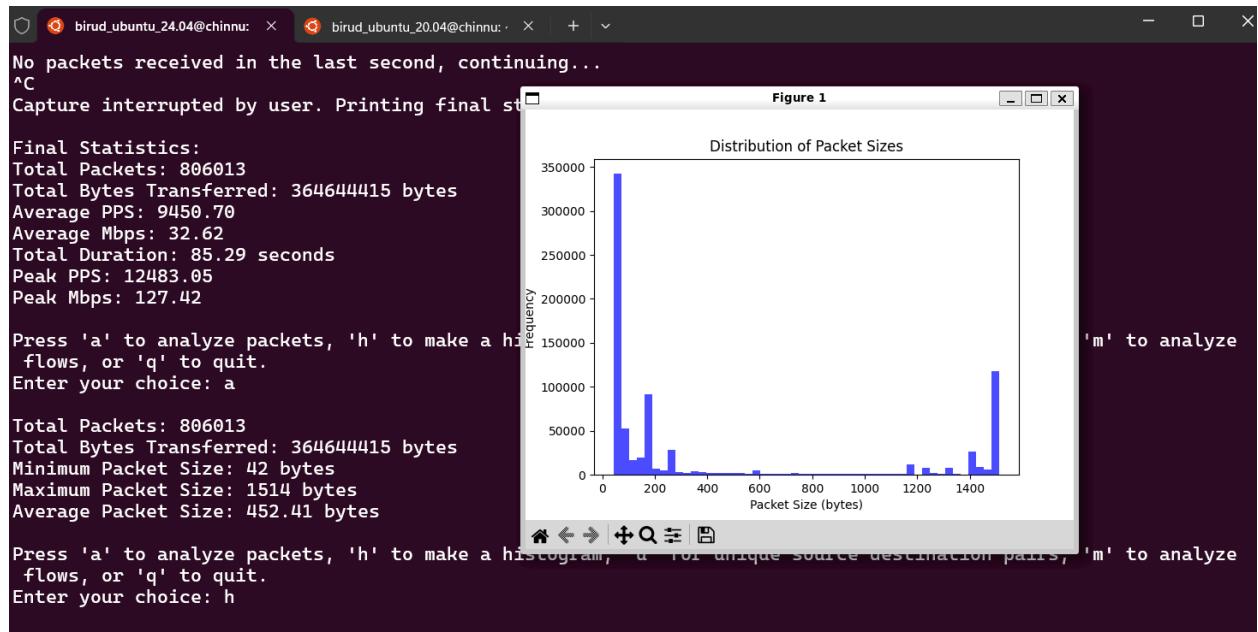
In the above image, the sender has completed sending the packets in 80.59 seconds, so there are no more packets to receive, so the user has pressed **Ctrl+C** to analyze further the captured packets.

- **Analyzing captured packets**

```
Enter your choice: a

Total Packets: 806013
Total Bytes Transferred: 364644415 bytes
Minimum Packet Size: 42 bytes
Maximum Packet Size: 1514 bytes
Average Packet Size: 452.41 bytes
```

- **Plotting Histogram**



- **Unique source-destination pairs**

The unique source-destination pairs will be written to a text file and saved to disk, which can be found in the `Packet-Sniffer/unique_pairs.txt` path of the GitHub repository.

```
Unique Source-Destination Pairs: 41900
('8.8.4.4:53', '172.16.133.6:63104')
('172.16.133.55:57701', '172.16.139.250:5440')
('96.43.146.48:443', '172.16.133.37:60769')
('173.194.43.38:80', '172.16.133.115:53733')
('172.16.133.28:65345', '172.16.139.250:5440')
('172.16.133.43:57817', '172.16.139.250:5440')
('172.16.133.26:53121', '96.43.146.176:443')
('180.153.31.250:443', '172.16.133.40:50296')
('172.16.133.116:53802', '172.16.139.250:5440')
('172.16.133.20:54795', '172.16.128.202:53')
('172.16.133.78:59066', '172.16.139.250:5440')
('172.16.133.92:57645', '172.16.139.250:5440')
```

- **Analyze flows**

The complete total flows data will also be written to disk as a text file, which can be found in the `Packet-Sniffer/total_flows.txt` path of the GitHub repository.

Destination IP -> Total Flows:
192.168.3.131:56368: 124
192.168.3.131:56427: 90
192.168.3.131:56132: 29
192.168.3.131:56093: 23
192.168.3.131:56233: 28
192.168.3.131:57244: 242
10.0.2.15:2526: 66
207.46.0.109:80: 29
192.168.3.131:57721: 4
65.54.95.68:80: 664

Source IP -> Total Flows:
65.54.95.68:80: 1275
65.54.95.75:80: 766
65.54.95.140:80: 658
204.14.234.85:8443: 516
65.54.186.19:5443: 33
192.168.3.131:56320: 15
72.14.213.105:443: 4
192.168.3.131:56511: 56
65.54.189.173:61863: 63
207.46.0.109:80: 63
184.85.226.161:443: 222
204.14.234.85:443: 516

Source-Destination Pair Transferring the Most Data:
('172.16.133.95:49358', '157.56.240.102:443'): 17342229 bytes

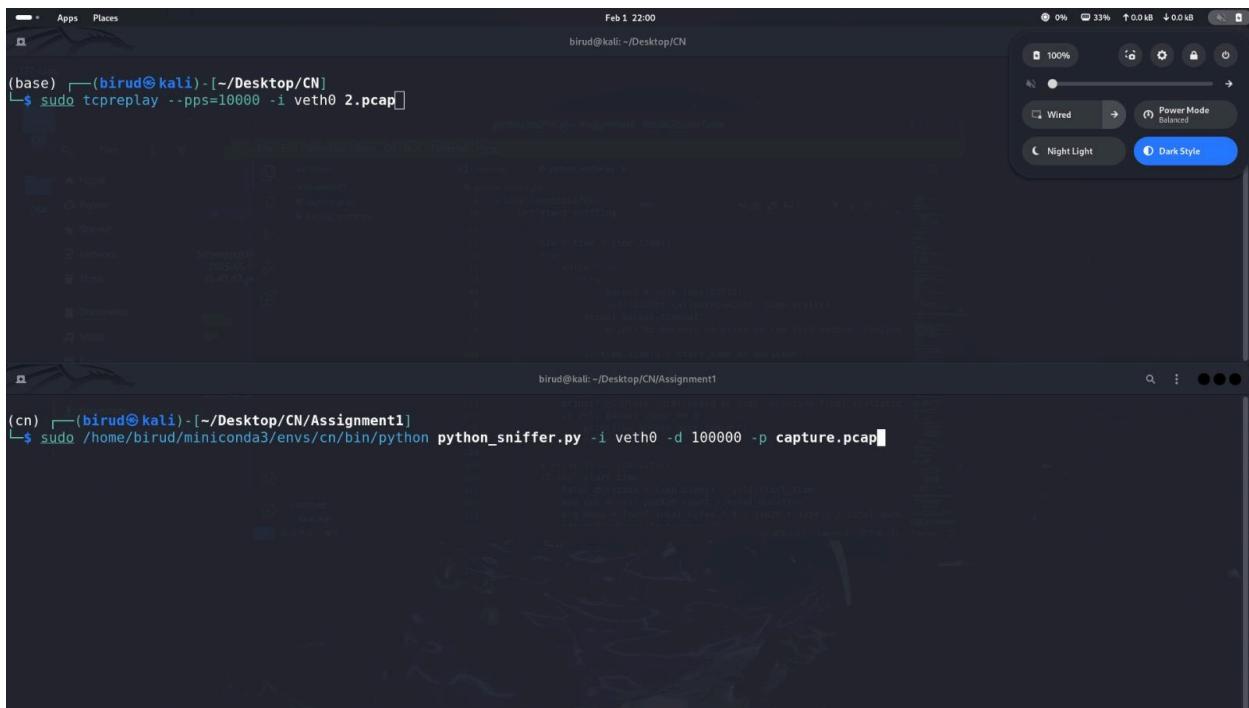
- **Running both tcpreplay and our program on the same machine**

For this process, we are using a Kali Linux virtual machine (VM). To ensure an isolated testing environment, we have disabled all wired connections and created a virtual network interface that is not connected to the internet. This setup allows us to replay network traffic using `tcpreplay` and simultaneously capture and analyze the flows using our program, all within the same machine without external interference.

This was done on linux using the following commands

```
(base) └─(birud㉿kali)-[~/Desktop/CN]
└$ sudo ip link add veth0 type dummy
```

```
(base) └─(birud㉿kali)-[~/Desktop/CN]
└$ sudo ip link set veth0 up
```



There are two terminals here: the upper one is the sender, and the second one is the receiver.

```

Warning in flows.c:flow_decode() line 244:
  flow_decode: packet 48002 IPv6 header version should be 6 but instead is 3
Warning in flows.c:flow_decode() line 244:
  flow_decode: packet 48153 IPv6 header version should be 6 but instead is 3
Warning in flows.c:flow_decode() line 244:
  flow_decode: packet 48154 IPv6 header version should be 6 but instead is 3
Warning in flows.c:flow_decode() line 244:
  flow_decode: packet 49571 IPv6 header version should be 6 but instead is 3
Warning in flows.c:flow_decode() line 244:
  flow_decode: packet 51255 IPv6 header version should be 6 but instead is 3
Warning in flows.c:flow_decode() line 244:
  flow_decode: packet 51523 IPv6 header version should be 6 but instead is 3
Warning in flows.c:flow_decode() line 244:
  flow_decode: packet 52863 IPv6 header version should be 6 but instead is 3
Warning in flows.c:flow_decode() line 244:
  flow_decode: packet 53080 IPv6 header version should be 6 but instead is 3

(birud㉿kali)-[~/Desktop/CN/Assignment1]
└─$ sudo /home/birud/miniconda3/envs/cn/bin/python python_sniffer.py -i veth0 -d 100000 -p capture.pcap
Starting packet capture on veth0 for 100000 seconds...
Successfully bound to veth0
No packets received in the last second, continuing...
No packets received in the last second, continuing...
No packets received in the last second, continuing...
Current PPS: 10002.90, Mbps: 25.20
Current PPS: 10002.10, Mbps: 30.80
Current PPS: 10001.04, Mbps: 26.98
Current PPS: 9999.63, Mbps: 29.71
Current PPS: 10000.67, Mbps: 30.11

```

```

flow decode: packet 790673 needs at least 62 bytes for ICMP header but only 60 available
Warning in flows.c:flow_decode() line 300:
flow decode: packet 790710 needs at least 62 bytes for ICMP header but only 60 available
Actual: 805997 packets (364642055 bytes) sent in 80.59 seconds
Rated: 4524117.3 Bps, 36.19 Mbps, 10000.01 pps
Flows: 41719 flows, 517.60 fps, 805298 unique flow packets, 454 unique non-flow packets
Statistics for network device: veth0
  Successful packets: 805997
  Failed packets: 0
  Truncated packets: 0
  Retried packets (ENOBUFS): 0
  Retried packets (EAGAIN): 0

Current PPS: 10000.02, Mbps: 34.25
Current PPS: 10000.00, Mbps: 34.40
No packets received in the last second, continuing...
Capture interrupted by user. Printing final statistics...

Final Statistics:
Total Packets: 805997
Total Bytes Transferred: 364642055 bytes
Average PPS: 8877.53
Average Mbps: 30.64
Total Duration: 90.79 seconds
Peak PPS: 13819.78
Peak Mbps: 146.15

```

Clearly, we can see that the number of packets sent is equal to the number of packets received, demonstrating high accuracy even in a VM.

The peak values are unusually high, likely due to calculating the instantaneous rate, where consecutive small packets resulted in a high packets-per-second (pps) value.

- **Running tcpreplay and our program on different machines**

We executed the experiment on two different Ubuntu Linux versions (20.04 and 24.04) and ensured that both Wi-Fi and Bluetooth were disabled during the data transfer.

## Experimental Setup

- **Sender (Ubuntu 20.04):** Used `tcpreplay` to send the PCAP file (`2.pcap`).
- **Receiver (Ubuntu 24.04):** Used our Python sniffer program to capture the traffic.
- **Network Interface:** Ethernet (`eth0`).

## Commands Used:

- Sender:

```
birud_ubuntu_20.04@chinnu:~$ sudo tcpreplay --pps=10000 -i eth0 2.pcap
```

- Receiver:

```
(base) birud_ubuntu_24.04@chinnu:~/CN/Computer-Networks/Packet-Sniffer$ sudo /home/birud_ubuntu/miniconda3/envs/cn/bin/python3 python_sniffer.py -i eth0 -d 10000 -p capture.pcap
```

## Results:

- Sender:

```
birud_ubuntu_20.04@chinnu:~$ sudo tcpreplay --pps=10000 -i eth0 2.pcap
Actual: 805997 packets (364642055 bytes) sent in 80.59 seconds
Rated: 4524116.1 Bps, 36.19 Mbps, 10000.00 pps
Statistics for network device: eth0
    Successful packets:      805997
    Failed packets:          0
    Truncated packets:       0
    Retried packets (ENOBUFS): 0
    Retried packets (EAGAIN): 0
```

- Receiver:

```
Current PPS: 10000.07, Mbps: 34.10
Current PPS: 10000.07, Mbps: 34.05
Current PPS: 10000.07, Mbps: 34.25
Current PPS: 10000.22, Mbps: 34.40
No packets received in the last second, continuing...
^C
Capture interrupted by user. Printing final statistics...

Final Statistics:
Total Packets: 806013
Total Bytes Transferred: 364644415 bytes
Average PPS: 9450.70
Average Mbps: 32.62
Total Duration: 85.29 seconds
Peak PPS: 12483.05
Peak Mbps: 127.42

Press 'a' to analyze packets, 'h' to make a histogram, 'u' for unique source-destination pairs, 'm' to analyze flows, or 'q' to quit.
Enter your choice: 
```

We have observed that the code is able to receive packets with negligible loss when TCP replay is sending up to 10,000 packets. However, beyond this limit, the code is unable to properly receive packets, resulting in some packet loss.

## Part-2

The following are the answers to the CTF questions. We saved the packets received by our sniffer in a new pcap file named ***capture.pcap*** and then used our program to solve the CTF challenges. Additionally, we also cross-checked whether our answers were correct using Linux commands.

```
Q1. My IP address: 10.1.2.200
Q2. Number of packets with IP 10.1.2.200: 80
Q3a. laptop = lenovo
Q3b. TCP checksum of that packet: 2657
Q4. Number of packets with 'Order successful': 40
```

## Verification with linux Commands:

- 1) The following command was used to find “my ip address”

```
(base) └─(birud㉿kali)-[~/Desktop/CN/Assignment1]
└$ strings capture.pcap | grep "10.1.2.200"
```

My ip address = <10.1.2.200>?M

- 2) Now to find the number of packets with this ip address:

```
(base) └─(birud㉿kali)-[~/Desktop/CN/Assignment1]
└$ tshark -r capture.pcap -Y "ip.addr == 10.1.2.200" | wc -l
80
```

- 3) Finding the name of the laptop

```
(base) └─(birud㉿kali)-[~/Desktop/CN/Assignment1]
└$ strings capture.pcap | grep "laptop"
ve had great local technical support from HP Korea, and we have seen big improvements since using this new solution. With most employees using Wi-Fi phones/laptops/smart phones, we can access patient data much faster and diagnose t
GET /t5/image/serverpage/avatar-name/icon_laptop2/avatar-theme/classic/avatar-collection/computers/avatar-display-size/message HTTP/1.1
ve had great local technical support from HP Korea, and we have seen big improvements since using this new solution. With most employees using Wi-Fi phones/laptops/smart phones, we can access patient data much faster and diagnose t
GET /resources/images/pages/hero-device-screenshots-laptop.png HTTP/1.1
    <div class="content-text marginBottom15"><p>The HP BYOD solution delivers a robust, simple and secure way for your enterprise to allow users to access your network as well as applications from their own laptop, tablet or smartphone.</p></div>
    PC, laptop, smartphone, etc.</p>
desks, laptops, and servers protecting system state, file data and application data like MS SQL, Exchange and
desks, laptops, and servers protecting system state, file data and application data like MS SQL, Exchange and
The name of laptop = lenovo?M
```

```
(base) └─(birud㉿kali)-[~/Desktop/CN/Assignment1]
└$ tshark -r capture.pcap -Y 'tcp contains "lenovo"' -T fields -e tcp.checksum
0x0a61
```

The laptop name is **lenovo**

The checksum of the packet is 0x0a61 (or 2657)

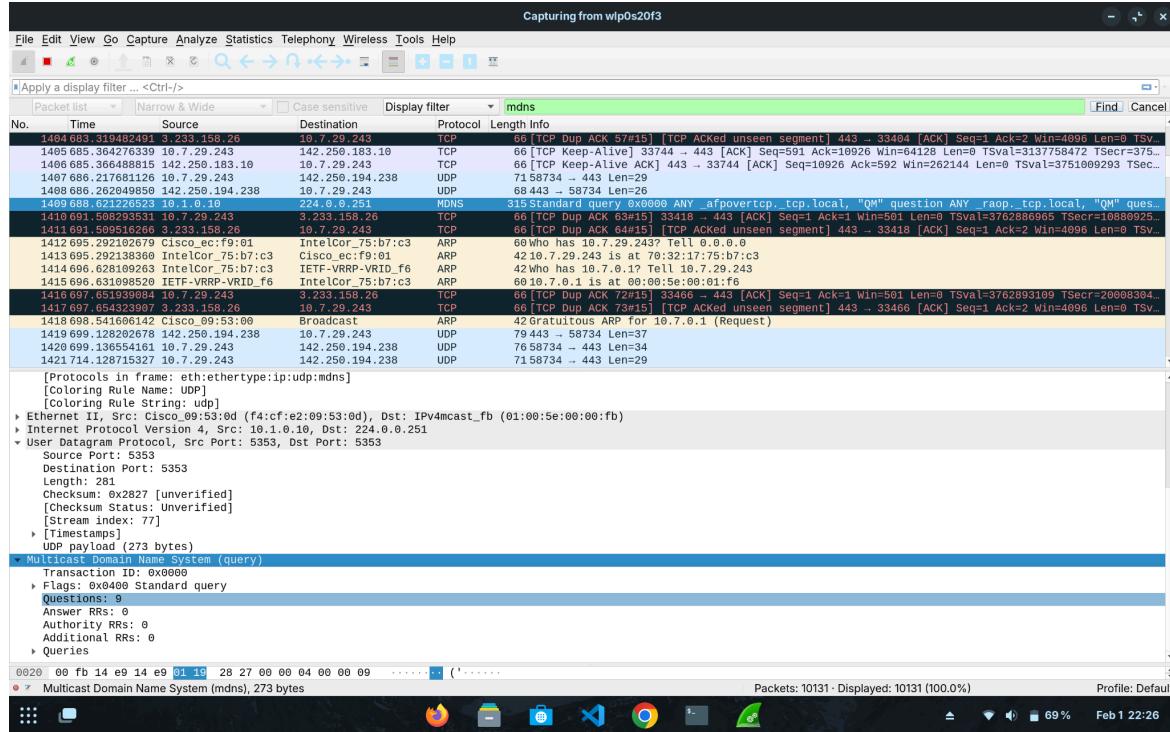
4) Find the number of packets which contain the message “Order successful”.

```
02 (base) └─(birud㉿kali)-[~/Desktop/CN/Assignment1]
└$ strings capture.pcap | grep -c "Order Successful"
40
```

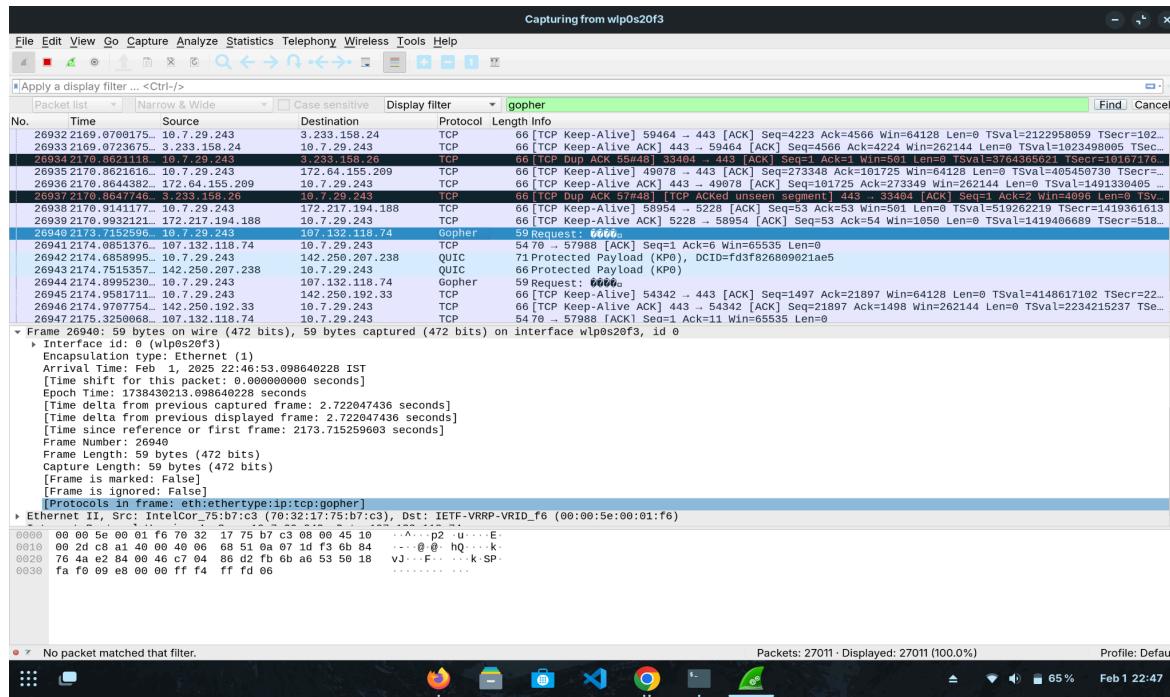
# Part 3

## Question 1:

MDNS:



Gopher:



## XMPP:

Capturing from wlp0s20f3

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter... <Ctrl-/>

Packet List Narrow & Wide Case sensitive Display filter xmpp Find Cancel

No.	Time	Source	Destination	Protocol	Length Info
+ 1973734626.671153...	10.7.29.243	57.144.177.32		XMP/P.X.	131 UNKNOWN PACKET
187728 4682.8437294...	Cisco_ed:f1:be	Broadcast	WLCP	138 U, func=UI; SNAP, OUI 0x0040996 [Cisco Systems, Inc], PID 0x0000	
187285 4622.7869214...	Cisco_ed:f1:be	Broadcast	WLCP	138 U, func=UI; SNAP, OUI 0x0040996 [Cisco Systems, Inc], PID 0x0000	
182962 4562.8347560...	Cisco_ed:f1:be	Broadcast	WLCP	138 U, func=UI; SNAP, OUI 0x0040996 [Cisco Systems, Inc], PID 0x0000	
182661 4582.7776249...	Cisco_ed:f1:be	Broadcast	WLCP	138 U, func=UI; SNAP, OUI 0x0040996 [Cisco Systems, Inc], PID 0x0000	
182485 4442.7626912...	Cisco_ed:f1:be	Broadcast	WLCP	138 U, func=UI; SNAP, OUI 0x0040996 [Cisco Systems, Inc], PID 0x0000	
182286 4382.7645086...	Cisco_ed:f1:be	Broadcast	WLCP	138 U, func=UI; SNAP, OUI 0x0040996 [Cisco Systems, Inc], PID 0x0000	
182612 4322.7766762...	Cisco_ed:f1:be	Broadcast	WLCP	138 U, func=UI; SNAP, OUI 0x0040996 [Cisco Systems, Inc], PID 0x0000	
181272 4262.6670417...	Cisco_ed:f1:be	Broadcast	WLCP	138 U, func=UI; SNAP, OUI 0x0040996 [Cisco Systems, Inc], PID 0x0000	
99812 4082.8014769...	Cisco_ed:f1:be	Broadcast	WLCP	138 U, func=UI; SNAP, OUI 0x0040996 [Cisco Systems, Inc], PID 0x0000	
99567 4142.8545750...	Cisco_ed:f1:be	Broadcast	WLCP	138 U, func=UI; SNAP, OUI 0x0040996 [Cisco Systems, Inc], PID 0x0000	
99181 4082.8014769...	Cisco_ed:f1:be	Broadcast	WLCP	138 U, func=UI; SNAP, OUI 0x0040996 [Cisco Systems, Inc], PID 0x0000	
98992 4022.8464102...	Cisco_ed:f1:be	Broadcast	WLCP	138 U, func=UI; SNAP, OUI 0x0040996 [Cisco Systems, Inc], PID 0x0000	
98651 3962.8464102...	Cisco_ed:f1:be	Broadcast	WLCP	138 U, func=UI; SNAP, OUI 0x0040996 [Cisco Systems, Inc], PID 0x0000	
89499 3909.7858214...	Cisco_ed:f1:be	Broadcast	WLCP	138 U, func=UI; SNAP, OUI 0x0040996 [Cisco Systems, Inc], PID 0x0000	
89320 3842.7914080...	Cisco_ed:f1:be	Broadcast	WLCP	138 U, func=UI; SNAP, OUI 0x0040996 [Cisco Systems, Inc], PID 0x0000	
[Time shift for this packet: 0.00000000 seconds]					
Epoch Time: 1738432666.054533669 seconds					
[Time delta from previous captured frame: 0.003955077 seconds]					
[Time delta from previous displayed frame: 0.003955077 seconds]					
[Time since reference or first frame: 4626.671153044 seconds]					
Frame Number: 107373					
Frame Length: 131 bytes (1048 bits)					
Capture Length: 131 bytes (1048 bits)					
[Frame is marked: False]					
[Frame is ignored: False]					
[Protocols in frame: eth:etherType:ip:tcp:xmpp:xml]					
[Coloring Rule Name: TCP]					
[Coloring Rule String: tcp]					
► Ethernet II, Src: IntelCor_75:b7:c3 (70:32:17:75:b7:c3), Dst: IETF-VRRP-VRID_16 (00:00:5e:00:01:f6)					
► Internet Protocol Version 4, Src: 10.7.29.243, Dst: 57.144.177.32					
▼ Transmission Control Protocol, Src Port: 41614, Dst Port: 5222, Seq: 1903, Ack: 1954, Len: 65					
0000 00 00 5e 00 01 f6 70 32 17 75 b7 c3 08 00 45 00 ..^.. p2 ..u... E...					
0010 00 75 32 e5 40 00 46 06 f4 f3 0a 07 f1 d3 99 u2 @ ..0 ..9...					
0020 b1 20 a2 8e 14 66 cb 8f da 3b d6 44 4e 80 18 ..f... ; JNF...					
0030 01 f5 13 12 00 00 01 01 08 00 ad 06 97 11 2d 7d .....{.....}					
0040 c0 e4 17 03 03 00 3c 89 aa 4b 57 ff c2 c8 26 cb .....<..Kw..&..					
0050 17 b6 02 d6 42 fe c9 7f 6c 81 0f aa 07 e2 27 8e ..B... l....'					
0060 d1 b6 73 fe 77 c0 a8 a8 42 f1 4e 68 7c 3d fc 2d ..s w... B:Nh =..					
Frame (131 bytes) Reassembled TCP (1967 bytes)					
Packets: 117663 - Displayed: 117663 (100.0%)					
Profile: Default					

## SSDP:

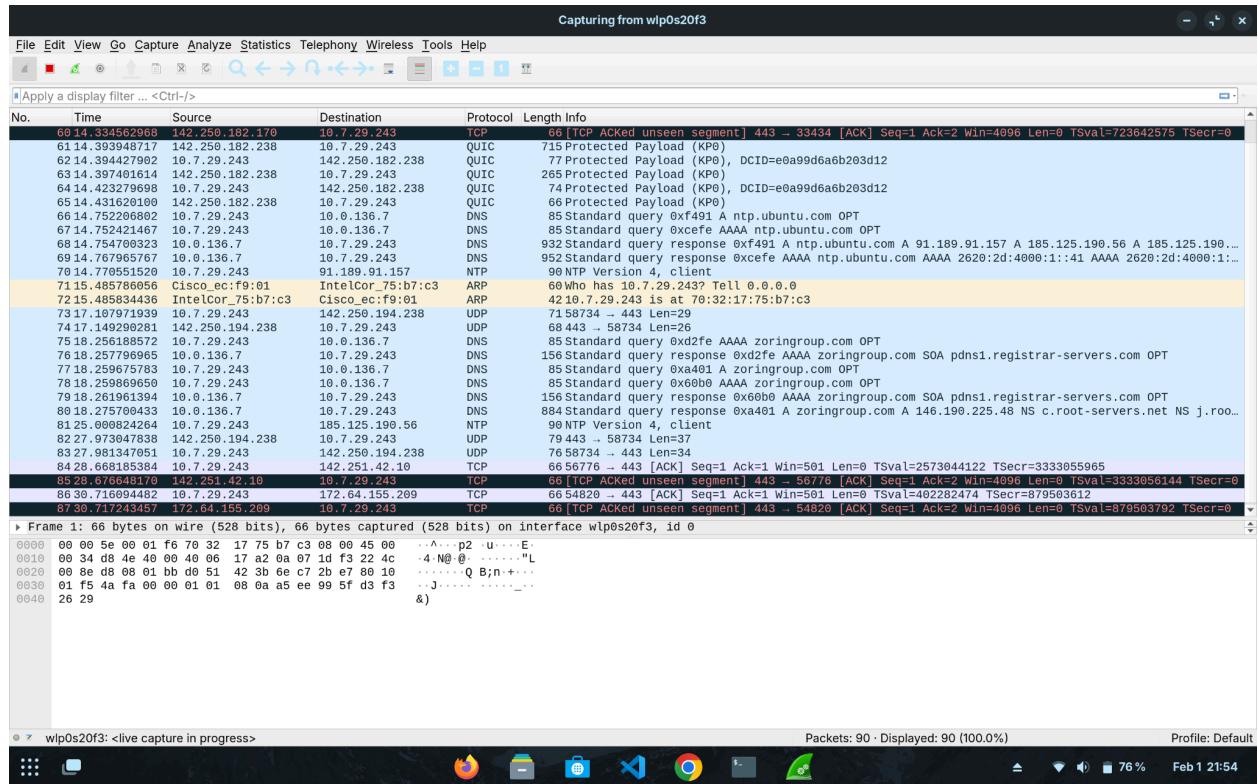
Capturing from wlp0s20f3

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

SS

No.	Time	Source	Destination	Protocol	Length Info
111.672512817	149.82.113.21	16.7.29.243	TLSV1.2	99 Application Data	
121.672646712	10.7.29.243	140.82.113.21	TCP	66 56964 .. 443 [ACK] Seq=40 Ack=103 Win=501 Len=0 TSval=831142253 TSecr=2899967267	
131.672512869	149.82.113.21	16.7.29.243	TCP	66 443 .. 56964 [FIN, ACK] Seq=103 Ack=40 Win=4096 Len=0 TSval=2899967267 TSecr=831141022	
141.672885801	10.7.29.243	140.82.113.21	TLSV1.2	105 Application Data	
151.673173821	10.7.29.243	140.82.113.21	TLSV1.2	99 Application Data	
161.676862484	140.82.113.21	10.7.29.243	TCP	54443 .. 56964 [RST] Seq=104 Win=0 Len=0	
172.693412186	10.7.29.243	239.255.255.250	SSDP	214 M-SEARCH * HTTP/1.1	
182.694764466	10.7.29.243	142.250.183.110	TCP	66 45568 .. 443 [ACK] Seq=1 Ack=1 Win=501 Len=0 TSval=773379650 TSecr=17276665900	
182.695958168	142.250.183.110	10.7.29.243	TCP	66 [TCP ACKed unseen segment] 443 .. 45568 [ACK] Seq=1 Ack=2 Win=4096 Len=0 TSval=1727666883 TSecr=0	
29.2.957871796	10.7.29.243	142.250.286.170	TCP	66 46832 .. 443 [ACK] Seq=1 Ack=1 Win=552 Len=0 TSval=1231549620 TSecr=545968836	
21.2.957979679	142.250.206.170	10.7.29.243	TCP	66 [TCP ACKed unseen segment] 443 .. 46832 [ACK] Seq=1 Ack=2 Win=4096 Len=0 TSval=545968216 TSecr=0	
23.3.004555738	10.7.29.243	239.255.255.250	SSDP	214 M-SEARCH * HTTP/1.1	
234.474243211	10.7.29.243	142.250.183.202	TCP	66 58772 .. 443 [ACK] Seq=1 Ack=1 Win=501 Len=0 TSval=3360510573 TSecr=1919467186	
24.4.742489614	10.7.29.243	142.250.183.110	TCP	66 45582 .. 443 [ACK] Seq=1 Ack=1 Win=501 Len=0 TSval=773381698 TSecr=3374598986	
25.4.745213427	10.7.29.243	35.261.107.59	TCP	66 39998 .. 443 [ACK] Seq=1 Ack=1 Win=501 Len=0 TSval=4181724936 TSecr=1763910926	
26.4.742534473	10.7.29.243	64.233.176.94	TCP	66 60200 .. 443 [ACK] Seq=1 Ack=1 Win=501 Len=0 TSval=2804156196 TSecr=1671813382	
27.4.743529804	142.250.183.202	10.7.29.243	TCP	66 [TCP ACKed unseen segment] 443 .. 58772 [ACK] Seq=1 Ack=2 Win=4096 Len=0 TSval=1919467366 TSecr=0	
28.4.744564013	35.201.107.59	10.7.29.243	TCP	66 [TCP ACKed unseen segment] 443 .. 39998 [ACK] Seq=1 Ack=2 Win=4096 Len=0 TSval=1671811105 TSecr=0	
29.4.744868681	64.233.176.94	10.7.29.243	TCP	66 [TCP ACKed unseen segment] 443 .. 45582 [ACK] Seq=1 Ack=2 Win=4096 Len=0 TSval=3374599173 TSecr=0	
30.4.746046834	142.250.183.110	10.7.29.243	TCP	66 [TCP ACKed unseen segment] 443 .. 45582 [ACK] Seq=1 Ack=2 Win=4096 Len=0 TSval=3374599173 TSecr=0	
31.5.254470181	10.7.29.243	142.250.242.46	TCP	66 37398 .. 443 [ACK] Seq=1 Ack=1 Win=1984 Len=0 TSval=571139751 TSecr=1576839456	
32.5.255542627	142.251.42.46	10.7.29.243	TCP	66 [TCP ACKed unseen segment] 443 .. 37398 [ACK] Seq=1 Ack=2 Win=4096 Len=0 TSval=1576839636 TSecr=0	
33.5.494441666	Cisco_ec:f9:01	IntelCor_75:b7:c3	ARP	66 who has 10.7.29.243? Tell 0.0.0.0	
34.5.494488482	IntelCor_75:b7:c3	Cisco_ec:f9:01	ARP	42 10.7.29.243 is at 70:32:17:75:b7:c3	
35.6.587227249	10.7.29.243	142.250.194.238	UDP	7158734 .. 443 Len=29	
36.6.635269365	142.250.194.238	10.7.29.243	UDP	68 443 .. 58734 Len=26	
37.6.790593886	10.7.29.243	142.251.42.46	TCP	66 55812 .. 443 [ACK] Seq=1 Ack=1 Win=501 Len=0 TSval=571141287 TSecr=2041958618	
38.6.790632898	10.7.29.243	142.250.207.234	TCP	66 34422 .. 443 [ACK] Seq=1 Ack=1 Win=501 Len=0 TSval=111982867 TSecr=3390025477	
Frame 1: 214 bytes on wire (1712 bits), 214 bytes captured (1712 bits) on interface wlp0s20f3, id 0					
0000 01 00 5e 7f ff fa 70 32 17 75 b7 c3 08 00 45 00 ..^.. p2 ..u... E...					
0010 00 98 16 bd 40 00 01 11 6a 73 0a 07 1d f3 ef ff ..@ ..js...					
0020 ff fa 95 a4 07 6c 0d 18 ba 4d 2d 53 45 51 52 l.. M-SEAR...					
0030 43 48 24 2a 28 48 54 54 59 2f 31 2d 31 0d 0a 48 CH * HTTP/1.1 - H					
0040 4f 53 54 3a 28 32 33 39 26 32 35 35 26 32 35 OST: 239.255.255					
0050 2e 32 35 39 3a 31 39 39 30 0d 0a 4d 41 4e 3a 29 ..250.190 0.. MAN:					
0060 22 73 73 64 70 3a 64 69 73 63 6f 76 65 72 22 0d "ssdp:discover"					
0070 0d 4d 58 3a 28 31 0d 0a 53 54 3a 29 75 72 66 3a MX: 1.. ST: urn:					
0080 64 69 61 6c 2d 67 56 7c 74 69 73 63 65 6a 64 69 61 dial-mul tiscreen					
0090 2d 6f 72 67 3a 73 65 72 76 69 63 65 3a 64 69 61 -org:ser vice:dia					
00a0 6c 3a 31 0d 0a 55 53 45 52 2d 41 47 45 44 54 3a l:1.. USE R-AGENT:					
00b0 20 47 6f 6f 67 6c 65 29 43 68 72 6f 6d 5f 2f 31 Google Chrome/1					
00c0 32 31 28 30 26 33 36 37 28 38 35 28 4c 69 66 21.0.616 7.85 Lin UX.....					
00d0 75 78 0d 0a 0d 0a					
Packets: 21101 · Displayed: 21101 (100.0%)					
Profile: Default					

## NTP:



mDNS (Multicast DNS) is a protocol that allows devices on a local network to resolve hostnames to IP addresses without needing a central DNS server. It is commonly used in zero-configuration networking (Zeroconf) for service discovery, such as in Apple's Bonjour and IoT devices. RFC: 6762

Gopher – A text-based protocol from the early internet (RFC 1436) designed for retrieving hierarchical documents, similar to HTTP but with a menu-driven interface. It was largely replaced by the web but still has niche usage.

XMPP (Extensible Messaging and Presence Protocol) is an open standard communication protocol used for real-time messaging, presence notification, and contact list management. It is widely used for instant messaging and chat applications due to its extensibility, scalability, and support for federated communication. RFC 6120

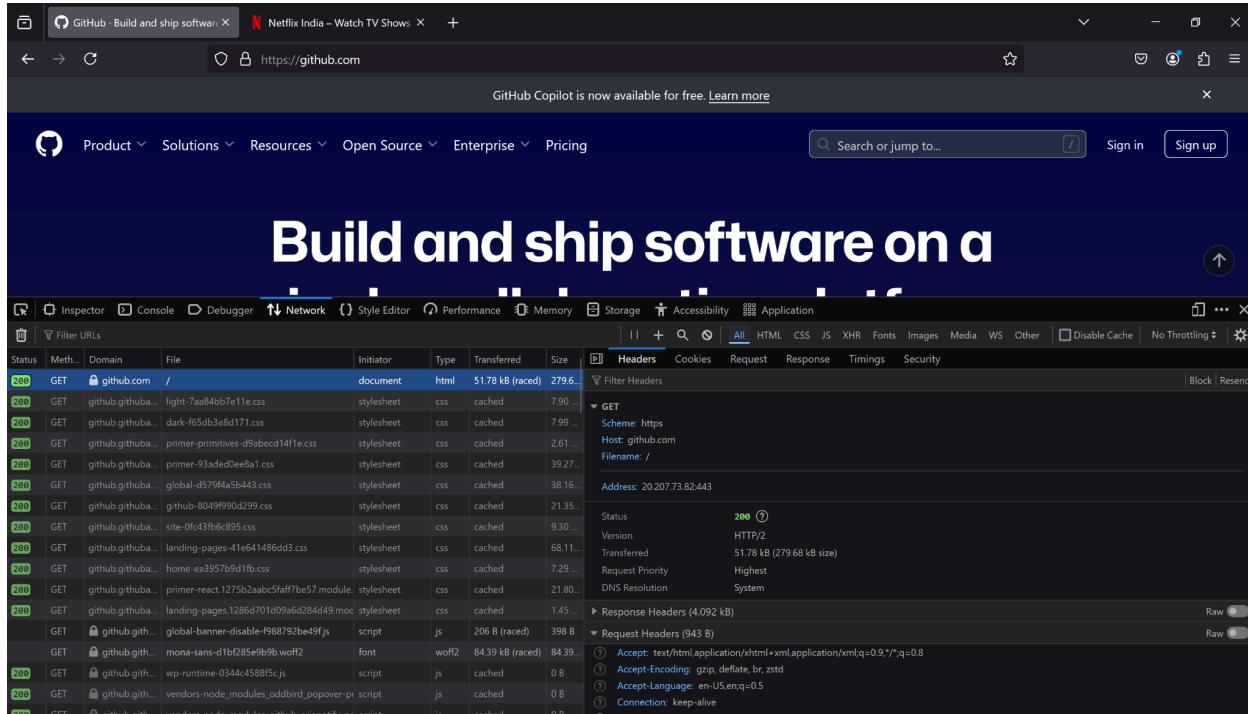
SSDP (Simple Service Discovery Protocol): A network protocol used in UPnP (Universal Plug and Play) to discover devices and services on a local network, often used in smart home and IoT applications. RFC: 2326

NTP (Network Time Protocol): A protocol designed to synchronize clocks across devices on a network with high precision, ensuring accurate timekeeping for distributed systems and security mechanisms. RFC: 5905

## Question 2:

### Part a:

1)github.com



The screenshot shows the Network tab of a browser developer tools interface, specifically the Network tab in Chrome DevTools. It displays a list of requests made to the GitHub website (https://github.com). The table has columns for Status, Method, Domain, File, Initiator, Type, Transferred, and Size. The first request, a GET to the root URL, is highlighted. The details panel on the right shows the request headers, including the IP address 20.207.73.82 and the connection type as persistent. The response headers also indicate a keep-alive connection.

Status	Method	Domain	File	Initiator	Type	Transferred	Size
200	GET	github.com	/	document	html	\$1.78 kB (raced)	279.6...
200	GET	github.github...	light-7aa840b7e11e.css	stylesheet	css	cached	7.90...
200	GET	github.github...	dark-f65cb3e8d171.css	stylesheet	css	cached	7.99...
200	GET	github.github...	primer-primitives-d9abeccd14f1e.css	stylesheet	css	cached	2.61...
200	GET	github.github...	primer-93aded0eeba1.css	stylesheet	css	cached	39.27...
200	GET	github.github...	global-d579f4a5b443.css	stylesheet	css	cached	38.16...
200	GET	github.github...	github-8049f990d299.css	stylesheet	css	cached	21.35...
200	GET	github.github...	site-0fc43fb6c895.css	stylesheet	css	cached	9.30...
200	GET	github.github...	landing-pages-41ee61486dd3.css	stylesheet	css	cached	68.11...
200	GET	github.github...	home-ea3957b9d1fb.css	stylesheet	css	cached	7.29...
200	GET	github.github...	primer-react-1275b2aab5faff7be57.module.css	stylesheet	css	cached	21.80...
200	GET	github.github...	landing-pages-128ed701d0996d284d49.mod...	stylesheet	css	cached	1.45...
GET	GET	github.github...	global-banner-disable-f988792be49f.js	script	js	206 B (raced)	398 B
GET	GET	github.github...	mona-sans-d1b285e9b9.woff2	font	woff2	84.39 kB (raced)	84.39...
200	GET	github.github...	wp-runtime-0344c458bf5c.js	script	js	cached	0 B
200	GET	github.github...	vendors-node_modules_odebird_popover-pi...	script	js	cached	0 B

Keep-alive connection implies it is persistent. IP address, protocol version are visible in the screenshot.

Request Line: GET / HTTP/2

IP Address: 20.207.73.82.443

Connection Type: persistent

## 2)canarabank.com

The screenshot shows the Canara Bank homepage with the title "कनारा बँक" and "Canara Bank". The browser's developer tools Network tab is open, displaying a list of requests made to the site. The table includes columns for Status, Met..., Domain, File, Initiator, Type, Transferred, and Size. Key requests include:

- GET https://canarabank.com/ document html 58.56 kB
- GET cdn.jsdelivr.net/bootstrap.min.css stylesheet css cached 30 kB
- GET cdnjs.cloudflare.com/owl.carousel.css stylesheet css cached 1.1 kB
- GET stackpath.bootstrapcdn.com/fontawesome.min.css stylesheet css cached -1 kB
- GET canarabank.bootstrapcdn.com/custom.css stylesheet css cached 89.7 kB
- GET canarabank.bootstrapcdn.com/simple-slider.css stylesheet css cached 1.6 kB
- GET canarabank.bootstrapcdn.com/logo.webp image webp cached 28.1 kB
- GET canarabank.bootstrapcdn.com/icon-13.svg image svg cached 51.1 kB
- GET canarabank.bootstrapcdn.com/Canaa\_E\_GST-1912.png image png cached 12.1 kB
- GET canarabank.bootstrapcdn.com/product-1.png image png cached 3.4 kB
- GET canarabank.bootstrapcdn.com/product-2.webp image webp cached 4.2 kB
- GET canarabank.bootstrapcdn.com/product-3.webp image webp cached 2.1 kB
- GET canarabank.bootstrapcdn.com/product-4.png image png cached 8.3 kB
- GET canarabank.bootstrapcdn.com/MSME.png image png cached 2.6 kB
- GET canarabank.bootstrapcdn.com/icon-2.png image png cached 2.4 kB
- GET canarabank.bootstrapcdn.com/icon-3.png image png cached 1.9 kB
- GET canarabank.bootstrapcdn.com/icon-preferred.png image png cached 8.3 kB
- GET canarabank.bootstrapcdn.com/Icon\_Quarterly\_Loan\_ac.png image png cached 8.3 kB
- GET canarabank.bootstrapcdn.com/Icon\_Online\_Submission.png image png cached 4.2 kB

At the bottom of the Network tab, it shows 136 requests, 2.76 MB transferred, and a total time of 2.05 s.

## 3)netflix.com

The screenshot shows the Netflix homepage with various movie and TV show thumbnails. The browser's developer tools Network tab is open, showing requests to the logs.netflix.com endpoint. The table includes columns for Status, Met..., Domain, File, Initiator, Type, Transferred, and Size. Key requests include:

- POST https://logs.netflix.com/nmhpfFrame... plain 6 kB 0 B
- POST https://logs.netflix.com/nmhpfFrame... plain 5.70 kB 0 B
- OPT... https://logs.netflix.com fetch plain 2.02 kB 0 B

At the bottom of the Network tab, it shows 3 requests, 0 B transferred, and a total time of 5.40 s.

## Part b(for github.com):

Request and Response Headers (any 3 fields can be chosen):

The screenshot shows the browser's developer tools with two expanded sections: "Response Headers" and "Request Headers".

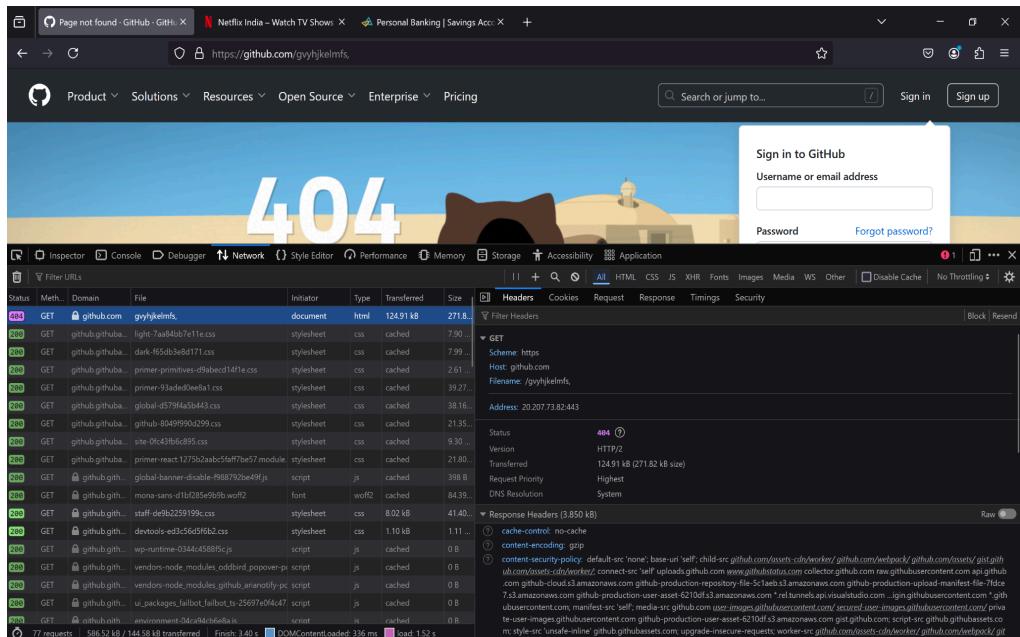
**Response Headers:**

- accept-ranges: bytes
- cache-control: max-age=0, private, must-revalidate
- content-encoding: gzip
- content-language: en-US
- content-security-policy: default-src 'none'; base-uri 'self'; child-src github.com/assets-cdn/worker/ github.com/webpack/ github.com/assets/gis t.github.com/assets-cdn/worker/; connect-src 'self' uploads.github.com www.githubstatus.com collector.github.com raw.githubusercontent.com api.github.com github-cloud.s3.amazonaws.com github-production-repository-file-5c1aeb.s3.amazonaws.com github-production-upload-man ifest-file-7fdce7.s3.amazonaws.com github-production-user-asset-6210df.s3.amazonaws.com \* .rel.tunnels.api.visualstudio.com ...om \* .githubusercontent.com; manifest-src 'self'; media-src github.com user-images.githubusercontent.com / secured-user-images.githubusercontent.com / private-user-images.githubusercontent.com github-production-user-asset-6210df.s3.amazonaws.com gist.github.com github.githubusercontent.com; script-src github.githubusercontent.com; style-src 'unsafe-inline' github.githubusercontent.com; upgrade-insecure-requests; worker-src github.com/assets-cd n/worker/ github.com/webpack/ github.com/assets/gist.github.com/assets-cdn/worker/
- content-type: text/html; charset=utf-8
- date: Sat, 01 Feb 2025 12:56:43 GMT
- etag: W/"f874436886eedca7f59572592fff9848"
- referrer-policy: origin-when-cross-origin, strict-origin-when-cross-origin
- server: GitHub.com
- strict-transport-security: max-age=31536000; includeSubdomains; preload
- vary: X-PJAX, X-PJAX-Container, Turbo-Visit, Turbo-Frame, Accept-Language, Accept-Encoding, Accept, X-Requested-With
- x-content-type-options: nosniff
- X-Firefox-Spdy: h2
- x-frame-options: deny
- x-github-request-id: 9404:293F5C:123A4D0:170AA0D:679E1A14
- x-xss-protection: 0

**Request Headers:**

- Accept: text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8
- Accept-Encoding: gzip, deflate, br, zstd
- Accept-Language: en-US,en;q=0.5
- Connection: keep-alive
- Cookie: \_gh\_sess=LW%2Bg%2Bz8WpGFKYxmJwT%2Bmf9Ygs%2F3ux2eTl0kuZ6tMyDBEqgeB4c1VgjvhWOCgLmst77IYYQYxLyRhsig5v77FBwmzQDVso%2BjEKZisSx700OvwU9n%2FIWkfbl%2BT0o3H%2Bi7rc77PjteCM%2BhuXionn%2FhrdOO5ax8zP%2FjFTV9WVxx6aQERpE94yJ5P%2FJUAHnWaYMWnXhf1MB3mVnPVh0rtWD4rHgUunXmcgTAGQ5tTRB7sPr7zLop%2F2%2FR3SKthDiW%2FNA%2BzfBQqcBoj%2Br1nzx1v70g%3D%3D--AskqrONm0%2Bhr2e3v--8CD5GhHD%2F1XWFUOjp0hsTq%3D%3D; \_octo=GH1.1.626057217.1738414256; logged\_in=no; cpu\_bucket=

## HTTP Error Codes:



The screenshot shows a browser window with a login form for 'Net Banking'. The URL is 'online.canarabank.in/?module=login'. The form has fields for 'Username' (fghbjk), 'Password', 'Image Captcha' (X6FB8), and 'Login'. Below the form are links for 'Create/Reset Login Password', 'Unlock User ID', 'Activate User ID', 'Forgot U...', 'New User Registrati...', 'PFMS Login', and 'TIN2.0 Bulk Pay...'. At the bottom, there are links for '15G/H Submission', 'Card Rewardz', 'Calendar', 'Canara Easy Fee', 'Canara Card Tokenisation', and 'Banking'. A 'secure GlobalSign' logo is also present.

**Network Tab Headers:**

- Request URL: https://online.canarabank.in/digx/j\_security\_check?locale=en
- Request Method: POST
- Status Code: 403 Forbidden
- Remote Address: 103.122.53.3:443
- Referrer Policy: strict-origin-when-cross-origin

**Response Headers:**

- Cache-Control: max-age=0, no-cache, no-store, must-revalidate
- Connection: Keep-Alive
- Content-Length: 525
- Content-Security-Policy: frame-ancestors 'self' 'billipe-sandbox.setu.co'; text/\*; charset=UTF-8
- Content-Type: text/html; charset=UTF-8
- Date: Sat, 01 Feb 2025 19:34:14 GMT
- Expires: Wed, 11 Jan 1984 05:00:00 GMT
- Keep-Alive: timeout=5, max=98
- Pragma: no-cache
- Server-Timing: dtSInfo:desc="0", dtRpId:desc="1765253612", dtTao:desc="1"

**404 Not Found** – The requested resource or webpage could not be found on the server, often due to a broken or incorrect URL.

**400 Bad Request** – The server cannot process the request due to malformed syntax, invalid parameters, or a client error.

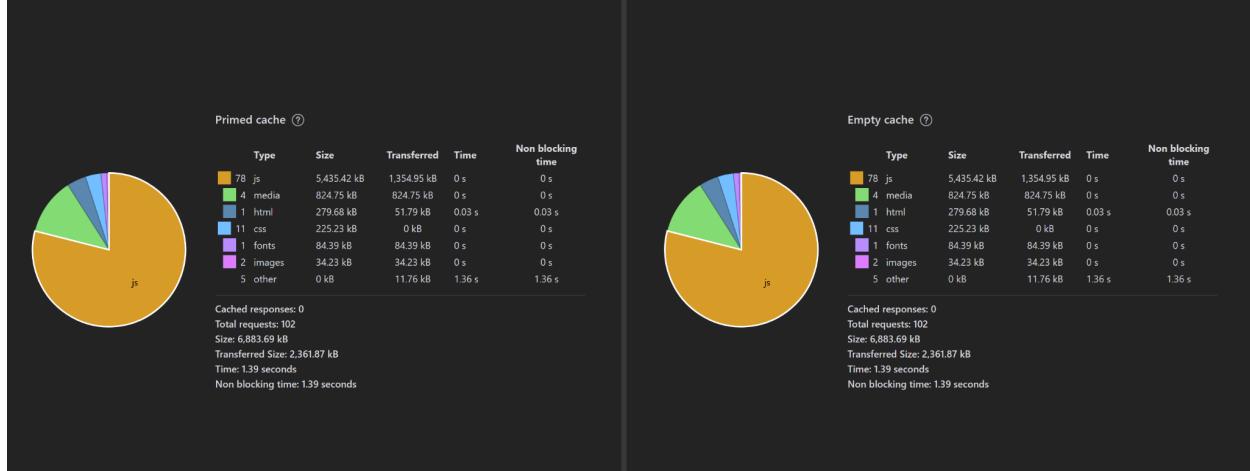
**403 Forbidden** – The server understands the request but refuses to authorize access, usually due to insufficient permissions.

**429 Too Many Requests** – This error occurs when the user has sent too many requests in a given amount of time, often due to rate limiting. It indicates that the client should slow down and try again later.

## Part c(for github.com):

Browser Name: Firefox

Performance Metrics



## List of Cookies

Name	Value	Domain	Path	Expires / Max-Age	Size	HttpOnly	Secure	SameSite	Last Accessed
.gh_sess	LW%2Bg%2Bz8...	github.com	/	Session	378	true	true	Lax	Sat, 01 Feb 2025 13:21:15 GMT
.octo	GH11.62605721...	github.com	/	Sun, 01 Feb 2026 12:00:00 GMT	31	false	true	Lax	Sat, 01 Feb 2025 13:21:15 GMT
cpu_bucket	xlg	github.com	/	Session	13	false	true	Lax	Sat, 01 Feb 2025 13:21:15 GMT
logged_in	no	github.com	/	Sun, 01 Feb 2026 12:00:00 GMT	11	true	true	Lax	Sat, 01 Feb 2025 13:21:15 GMT
preferred...	dark	github.com	/	Session	24	false	true	Lax	Sat, 01 Feb 2025 13:21:15 GMT
tz	Asia%2FKolkata	github.com	/	Session	16	false	true	Lax	Sat, 01 Feb 2025 13:21:15 GMT

Selected cookie details:

- gh\_sess**: LW%2Bg%2Bz8...; Domain: "github.com"; Expires / Max-Age: "Session"; HttpOnly: true; Secure: true; SameSite: Lax; Last Accessed: "Sat, 01 Feb 2025 13:21:15 GMT"; Path: "/"; SameSite: "Lax"; Secure: true; Size: 378
- gh\_sess**: LW%2Bg%2Bz8WpGFKYxmJuwT+Mf9Ygs/3ux2eTL...3SKhDiW/NA+zfbOqcBo+jr1nzc1v70g; Path: "/"; SameSite: "Lax"; Secure: true; Size: 378

## Associated flags in response and request headers

strict-transport-security: max-age=31536000; includeSubdomains; preload
vary: X-Requested-With, X-PJAX-Container, Turbo-Frame, Turbo-Visit, Accept-Encoding, Accept, X-Request-With
x-content-type-options: nosniff
X-Firefox-Spdy: h2
x-frame-options: deny
x-github-request-id: 9A6A:15D61E:DE55E0:1184ADF:679E4953
x-xss-protection: 0

Host: github.com
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: none
Sec-Fetch-User: ?1
TE: trailers
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:122.0) Gecko/20100101 Firefox/122.0