COMPUTER NETWORKS LAB

REPORT ASSIGNMENT-5

NAME: SOHAM LAHIRI

CLASS: BCSE UG-III 5TH SEMESTER

ROLL NO: 002210501107

GROUP: A3

SUBMISSION DATE: 18/11/2024

Problem Statement:

1.Generate some ICMP traffic by using the Ping command line tool to check the connectivity of a neighbouring machine (or router). Note the results in Wireshark. The initial ARP request broadcast from your PC determines the physical MAC address of the network IP Address, and the ARP reply from the neighboring system. After the ARP request, the pings (ICMP echo request and replies) can be seen.

Outputs:

Command Prompt:

```
C:\Users\User>ping 192.168.29.1

Pinging 192.168.29.1 with 32 bytes of data:
Reply from 192.168.29.1: bytes=32 time=183ms TTL=64
Reply from 192.168.29.1: bytes=32 time=2ms TTL=64
Reply from 192.168.29.1: bytes=32 time=2ms TTL=64
Reply from 192.168.29.1: bytes=32 time=2ms TTL=64

Ping statistics for 192.168.29.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 183ms, Average = 47ms
```

Wireshark:

No.	Time	Source	Destination	Protocol	Length	Info
→ 9	2.160932	192.168.29.31	192.168.29.1	ICMP	74	Echo (ping) request id=0x0001, seq=1/256, ttl=128 (reply in 11)
← 11	2.344279	192.168.29.1	192.168.29.31	ICMP	74	Echo (ping) reply id=0x0001, seq=1/256, ttl=64 (request in 9)
14	3.178662	192.168.29.31	192.168.29.1	ICMP	74	Echo (ping) request id=0x0001, seq=2/512, ttl=128 (reply in 15)
15	3.180560	192.168.29.1	192.168.29.31	ICMP	74	Echo (ping) reply id=0x0001, seq=2/512, ttl=64 (request in 14)
20	4.200104	192.168.29.31	192.168.29.1	ICMP	74	Echo (ping) request id=0x0001, seq=3/768, ttl=128 (reply in 21)
21	4.202399	192.168.29.1	192.168.29.31	ICMP	74	Echo (ping) reply id=0x0001, seq=3/768, ttl=64 (request in 20)
23	5.226076	192.168.29.31	192.168.29.1	ICMP	74	Echo (ping) request id=0x0001, seq=4/1024, ttl=128 (reply in 24)
24	5.228366	192.168.29.1	192.168.29.31	ICMP	74	Echo (ping) reply id=0x0001, seq=4/1024, ttl=64 (request in 23)

Wireshark · Packet 9 · Wi-Fi

```
> Frame 9: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface \Device\NPF_{CB2D87BB-2C08-4B5A-B2DC-F2893C56B2BB}, id 0
> Ethernet II, Src: ChongqingFug_d6:f6:2f (40:5b:d8:d6:f6:2f), Dst: ServercomPri_a5:c0:66 (b4:a7:c6:a5:c0:66)
> Internet Protocol Version 4, Src: 192.168.29.31, Dst: 192.168.29.1
> Internet Control Message Protocol
```

```
0000  b4 a7 c6 a5 c0 66 40 5b  d8 d6 f6 2f 08 00 45 00   ·····f@[ ···/··E·
0010  00 3c 3c 89 00 00 80 01  42 c7 c0 a8 1d 1f c0 a8   ·<<····· B·······
0020  1d 01 08 00 4d 5a 00 01  00 01 61 62 63 64 65 66   ····MZ·· ··abcdef
0030  67 68 69 6a 6b 6c 6d 6e  6f 70 71 72 73 74 75 76   ghijklmn opqrstuv
0040  77 61 62 63 64 65 66 67  68 69                    wabcdefg hi
```

No.: 9 · Time: 2.160932 · Source: 192.168.29.31 · Destination: 192.168.29.1 · Protocol: ICMP · Length: 74 · Info: Echo (ping) request id=0x0001, seq=1/256, ttl=128 (reply in 11)

Wireshark · Packet 11 · Wi-Fi

```
> Frame 11: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface \Device\NPF_{CB2D87BB-2C08-4B5A-B2DC-F2893C56B2BB}, id 0
> Ethernet II, Src: ServercomPri_a5:c0:66 (b4:a7:c6:a5:c0:66), Dst: ChongqingFug_d6:f6:2f (40:5b:d8:d6:f6:2f)
> Internet Protocol Version 4, Src: 192.168.29.1, Dst: 192.168.29.31
> Internet Control Message Protocol
```

```
0000  40 5b d8 d6 f6 2f b4 a7  c6 a5 c0 66 08 00 45 00   @[···/·· ···f··E·
0010  00 3c ca 12 00 00 40 01  f5 3d c0 a8 1d 01 c0 a8   ·<····@· ·=······
0020  1d 1f 00 00 55 5a 00 01  00 01 61 62 63 64 65 66   ····UZ·· ··abcdef
0030  67 68 69 6a 6b 6c 6d 6e  6f 70 71 72 73 74 75 76   ghijklmn opqrstuv
0040  77 61 62 63 64 65 66 67  68 69                    wabcdefg hi
```

No.: 11 · Time: 2.344279 · Source: 192.168.29.1 · Destination: 192.168.29.31 · Protocol: ICMP · Length: 74 · Info: Echo (ping) reply id=0x0001, seq=1/256, ttl=64 (request in 9)

2. Generate some web traffic and

a. find the list of the different protocols that appear in the protocol column in the unfiltered packet-listing window of Wireshark.

Wireshark · Protocol Hierarchy Statistics · Wi-Fi

Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s	PDUs
Frame	100.0	28	100.0	16569	52 k	0	0	0	28
Ethernet	100.0	28	2.4	392	1236	0	0	0	28
Internet Protocol Version 6	14.3	4	1.0	160	504	0	0	0	4
Transmission Control Protocol	14.3	4	0.5	80	252	0	0	0	4
Hypertext Transfer Protocol	14.3	4	207.8	34425	108 k	2	708	2232	4
Media Type	3.6	1	559.1	92629	292 k	1	92629	292 k	1
Line-based text data	3.6	1	0.8	134	422	1	134	422	1
Internet Protocol Version 4	85.7	24	2.9	480	1513	0	0	0	24
Transmission Control Protocol	85.7	24	2.9	480	1513	0	0	0	24
Hypertext Transfer Protocol	85.7	24	153.9	25499	80 k	12	4424	13 k	24
Portable Network Graphics	3.6	1	16.5	2736	8629	1	2736	8629	1
Media Type	17.9	5	104.9	17377	54 k	5	17377	54 k	5
Line-based text data	21.4	6	68.7	11387	35 k	6	11387	35 k	6

b. How long did it take from when the HTTP GET message was sent until the HTTP OK reply was received? (By default, the value of the Time column in the packet-listing window is the amount of time, in seconds, since Wireshark tracing began. To display the Time field in time-of-day format, select the Wireshark View pull down menu, then select Time Display Format, then select Time-of-day.)

No.	Time	Source	Destination	Protocol	Length	Info
5709	29.746898	192.168.29.31	44.228.249.3	HTTP	490	GET / HTTP/1.1
5721	30.011139	44.228.249.3	192.168.29.31	HTTP	1278	HTTP/1.1 200 OK (text/html)

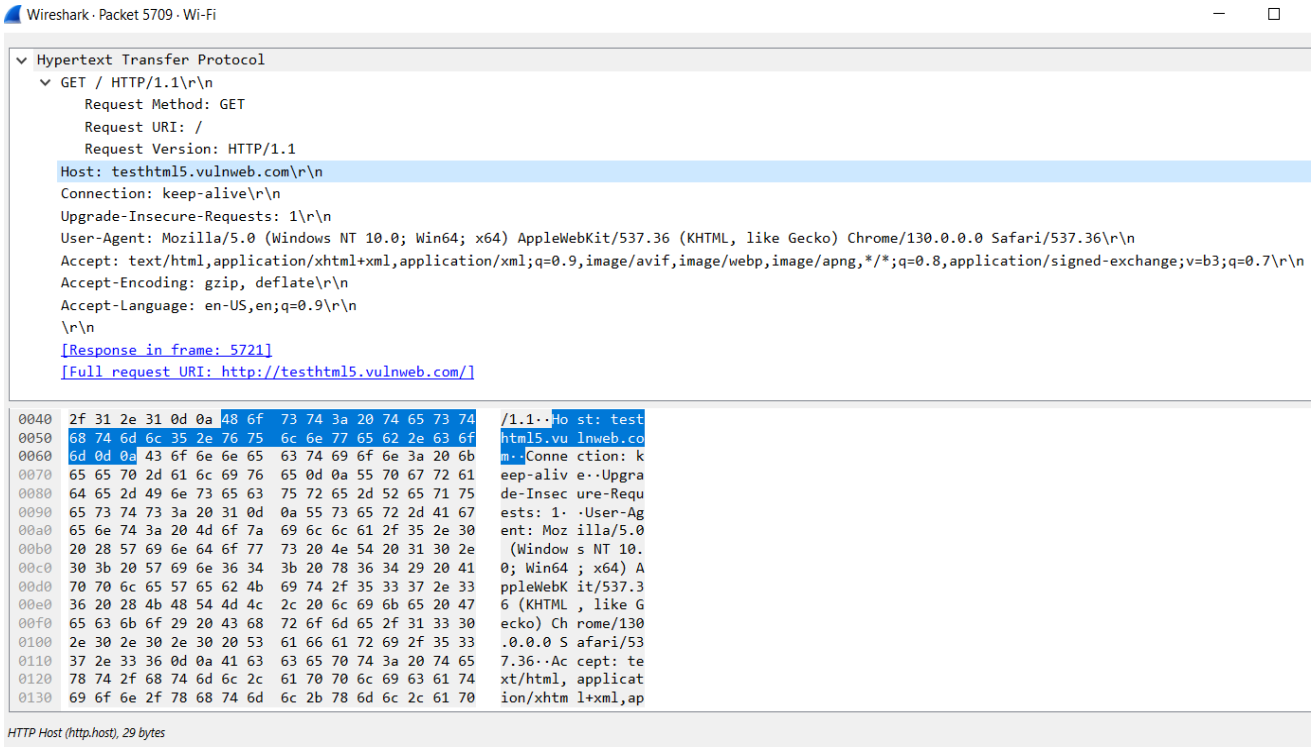
According to the Time column in above snapshot, it took 0.264241seconds approx. to get the HTTP response.

c. What is the Internet address of the website? What is the Internet address of your computer?

The local IP is 192.168.29.31 and the server IP is 44.228.249.3 .

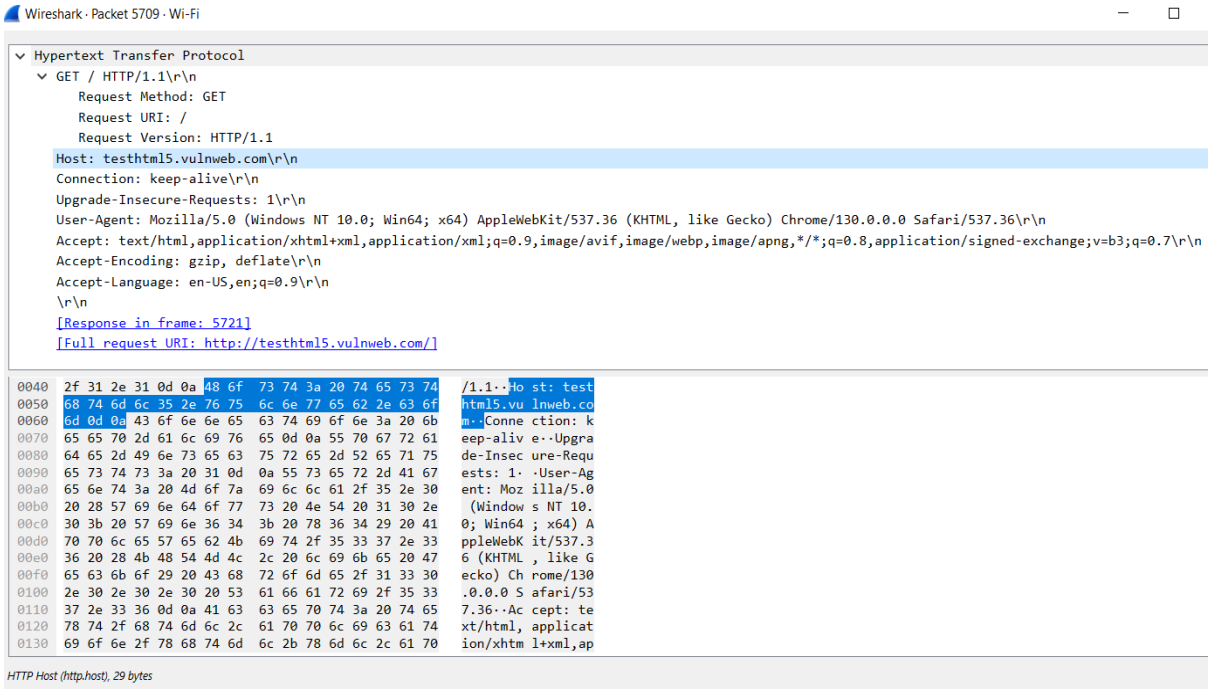
d. Search back through your capture, and find an HTTP packet containing a GET command. Click on the packet in the Packet List Panel. Then expand the HTTP layer in the Packet Details Panel, from the packet.

e. Find out the value of the Host from the Packet Details Panel, within the GET command.



The value of Host as shown above is: testhtml5.vulnweb.com .

3. Highlight the Hex and ASCII representations of the packet in the Packet Bytes Panel.



(Highlighted in blue)

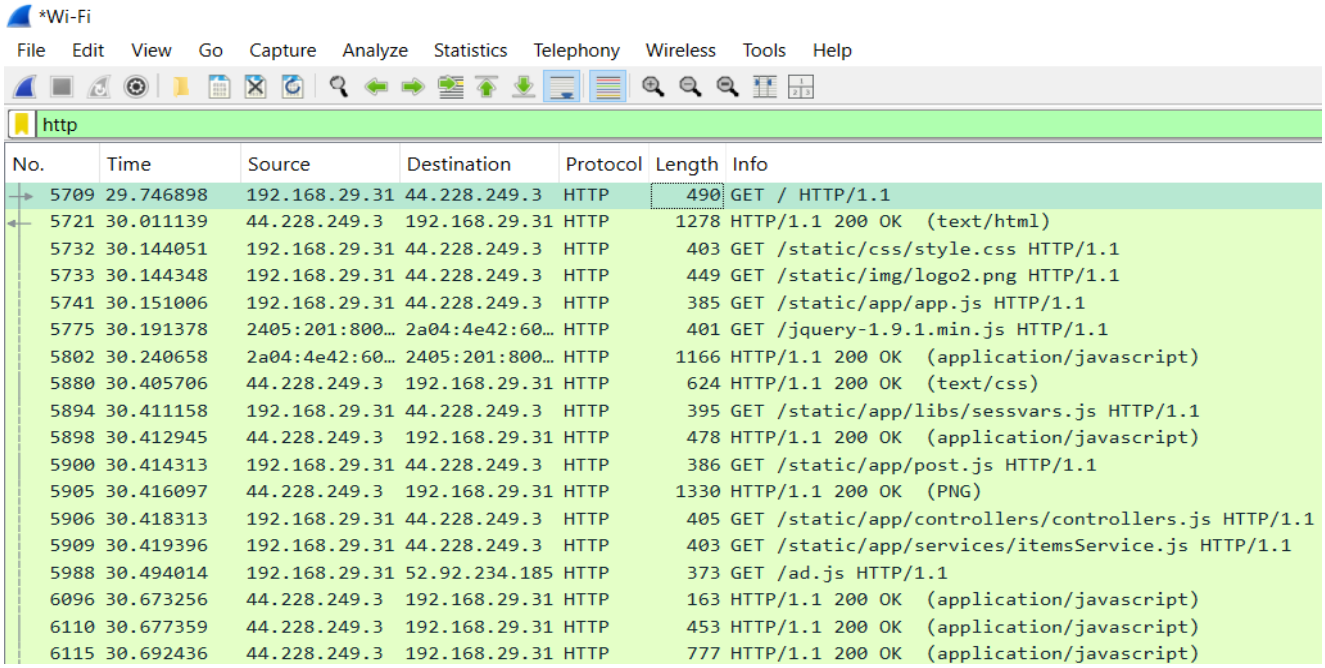
4. Find out the first 4 bytes of the Hex value of the Host parameter from the Packet Bytes Panel.

From the previous snippet, the first 4 bytes of the Hex value of the Host parameter from the Packet Bytes Panel are: 48 6f 73 74

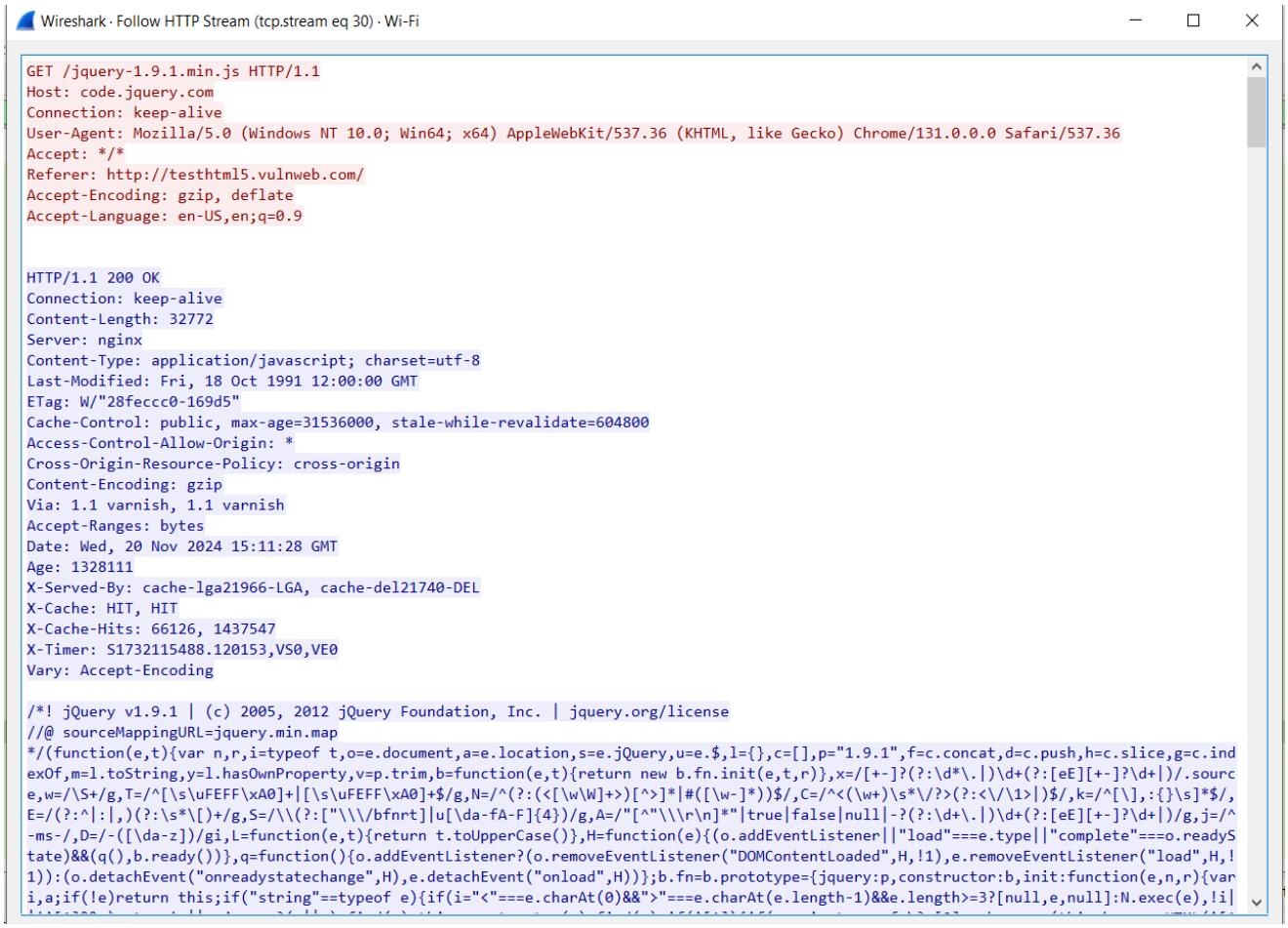
5. Filter packets with http, TCP, DNS and other protocols.

a. Find out what those packets contain by following one of the conversations (also called network flows), select one of the packets and press the right mouse button..click on follow.

Filtered by HTTP:

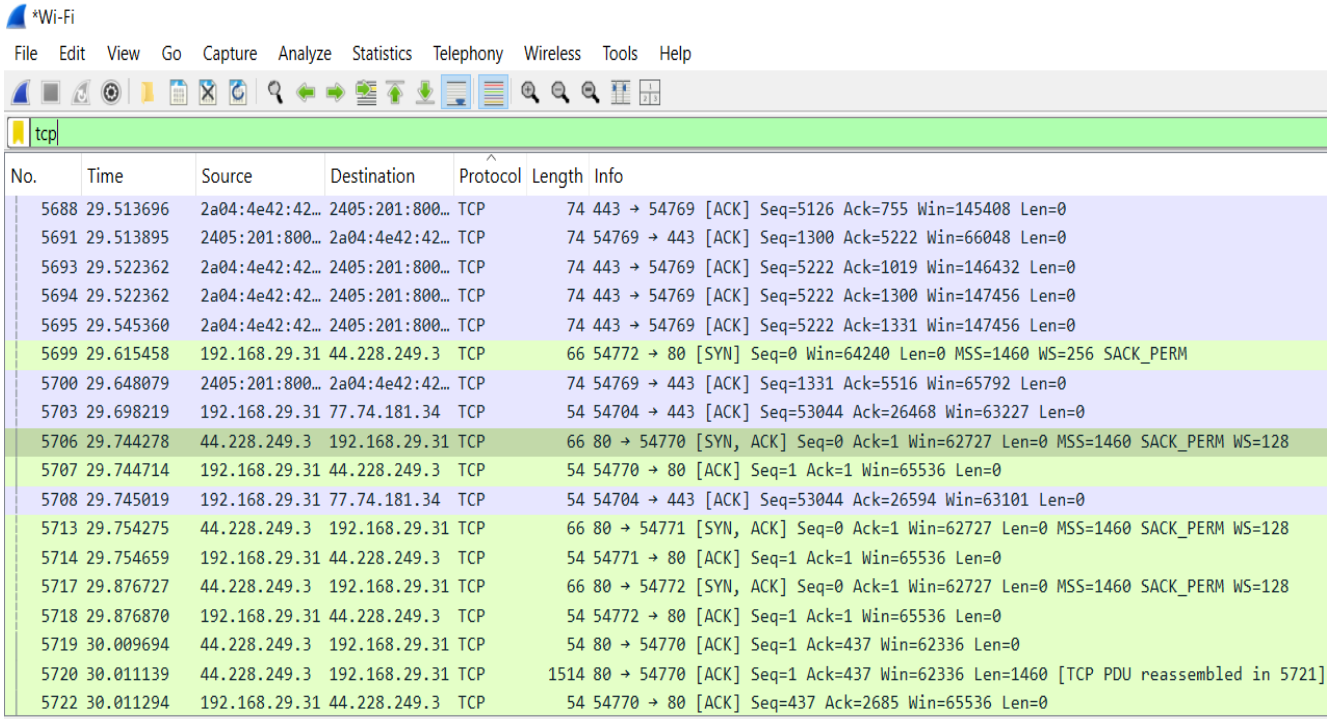


No.	Time	Source	Destination	Protocol	Length	Info
5709	29.746898	192.168.29.31	44.228.249.3	HTTP	490	GET / HTTP/1.1
5721	30.011139	44.228.249.3	192.168.29.31	HTTP	1278	HTTP/1.1 200 OK (text/html)
5732	30.144051	192.168.29.31	44.228.249.3	HTTP	403	GET /static/css/style.css HTTP/1.1
5733	30.144348	192.168.29.31	44.228.249.3	HTTP	449	GET /static/img/logo2.png HTTP/1.1
5741	30.151006	192.168.29.31	44.228.249.3	HTTP	385	GET /static/app/app.js HTTP/1.1
5775	30.191378	2405:201:800…	2a04:4e42:60…	HTTP	401	GET /jquery-1.9.1.min.js HTTP/1.1
5802	30.240658	2a04:4e42:60…	2405:201:800…	HTTP	1166	HTTP/1.1 200 OK (application/javascript)
5880	30.405706	44.228.249.3	192.168.29.31	HTTP	624	HTTP/1.1 200 OK (text/css)
5894	30.411158	192.168.29.31	44.228.249.3	HTTP	395	GET /static/app/libs/sessvars.js HTTP/1.1
5898	30.412945	44.228.249.3	192.168.29.31	HTTP	478	HTTP/1.1 200 OK (application/javascript)
5900	30.414313	192.168.29.31	44.228.249.3	HTTP	386	GET /static/app/post.js HTTP/1.1
5905	30.416097	44.228.249.3	192.168.29.31	HTTP	1330	HTTP/1.1 200 OK (PNG)
5906	30.418313	192.168.29.31	44.228.249.3	HTTP	405	GET /static/app/controllers/controllers.js HTTP/1.1
5909	30.419396	192.168.29.31	44.228.249.3	HTTP	403	GET /static/app/services/itemsService.js HTTP/1.1
5988	30.494014	192.168.29.31	52.92.234.185	HTTP	373	GET /ad.js HTTP/1.1
6096	30.673256	44.228.249.3	192.168.29.31	HTTP	163	HTTP/1.1 200 OK (application/javascript)
6110	30.677359	44.228.249.3	192.168.29.31	HTTP	453	HTTP/1.1 200 OK (application/javascript)
6115	30.692436	44.228.249.3	192.168.29.31	HTTP	777	HTTP/1.1 200 OK (application/javascript)

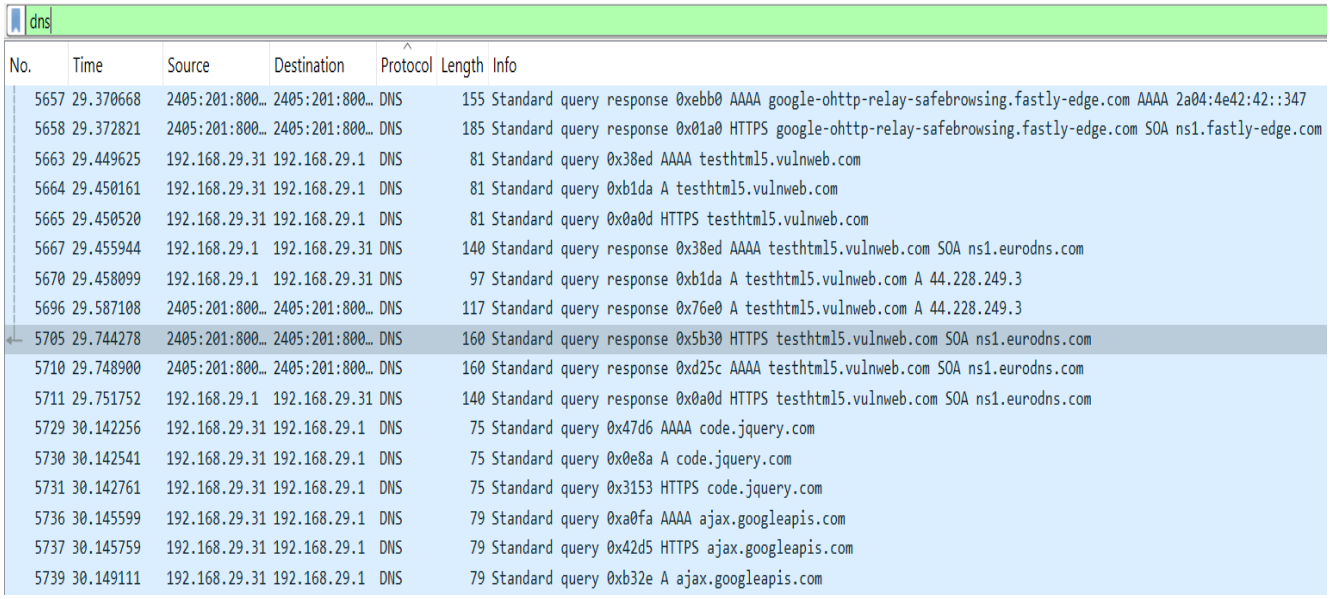


Following a HTTP Packet

Filtered by TCP:



Filtered by DNS:

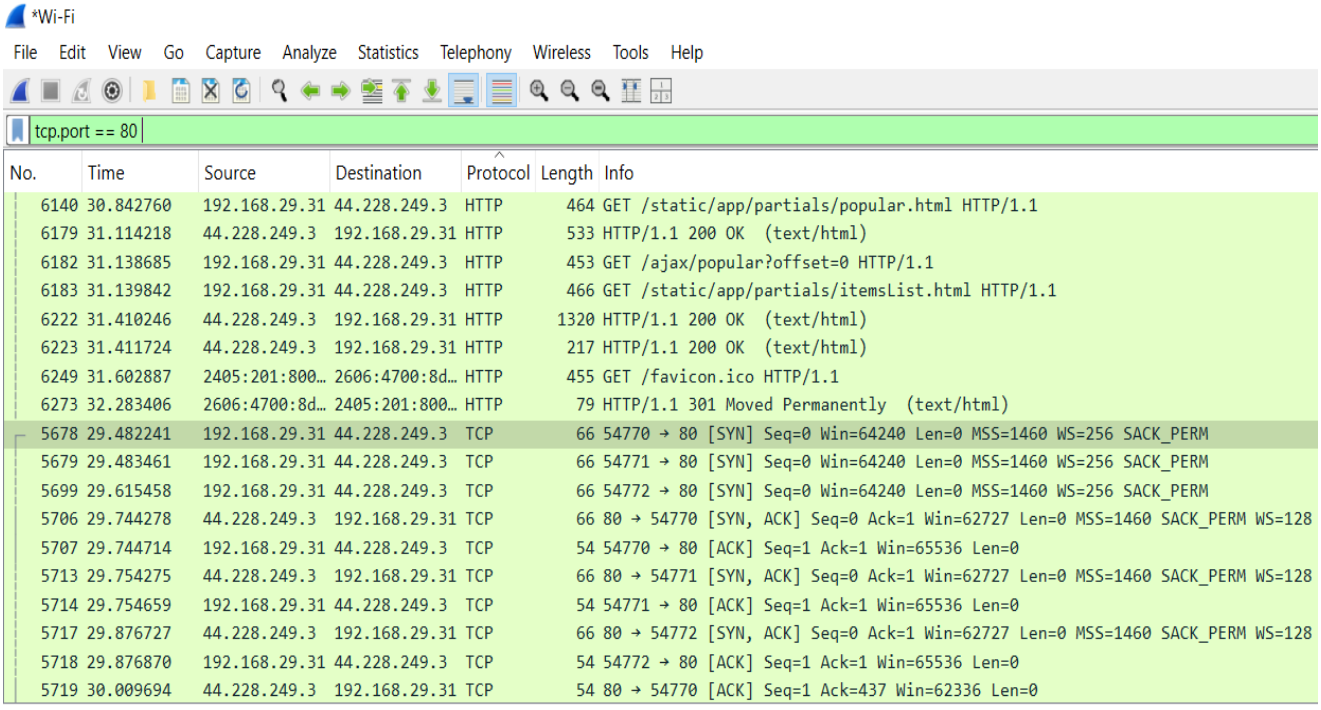


Filtered by TLS:

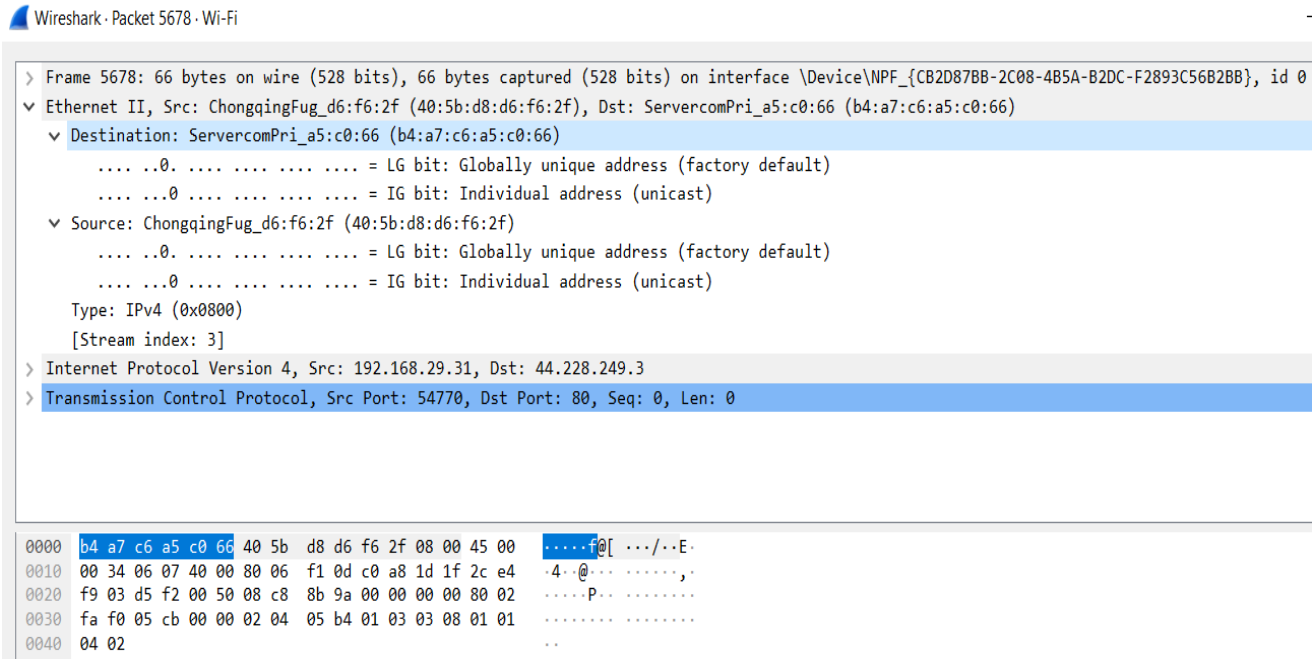
tls

No.	Time	Source	Destination	Protocol	Length	Info
5640	29.016802	192.168.29.31	77.74.181.34	TLSv1.2	132	Application Data
5641	29.016978	192.168.29.31	77.74.181.34	TLSv1.2	800	Application Data
5647	29.227742	77.74.181.34	192.168.29.31	TLSv1.2	109	Application Data
5648	29.230035	77.74.181.34	192.168.29.31	TLSv1.2	261	Application Data
5680	29.484800	192.168.29.31	77.74.181.34	TLSv1.2	131	Application Data
5681	29.484988	192.168.29.31	77.74.181.34	TLSv1.2	216	Application Data
5701	29.697782	77.74.181.34	192.168.29.31	TLSv1.2	109	Application Data
5702	29.697782	77.74.181.34	192.168.29.31	TLSv1.2	195	Application Data
5704	29.699294	77.74.181.34	192.168.29.31	TLSv1.2	180	Application Data
5715	29.810990	192.168.29.31	77.74.181.34	TLSv1.2	131	Application Data
5716	29.811219	192.168.29.31	77.74.181.34	TLSv1.2	183	Application Data
5723	30.023525	77.74.181.34	192.168.29.31	TLSv1.2	109	Application Data
5724	30.026236	77.74.181.34	192.168.29.31	TLSv1.2	294	Application Data
5734	30.144829	192.168.29.31	77.74.181.34	TLSv1.2	132	Application Data
5735	30.145234	192.168.29.31	77.74.181.34	TLSv1.2	329	Application Data
5806	30.247103	192.168.29.31	77.74.181.34	TLSv1.2	131	Application Data
5807	30.247342	192.168.29.31	77.74.181.34	TLSv1.2	163	Application Data
5808	30.247790	192.168.29.31	77.74.181.34	TLSv1.2	131	Application Data

6. Search through your capture, and find an HTTP packet coming back from the server (TCP Source Port == 80). Expand the Ethernet layer in the Packet Details Panel.



No.	Time	Source	Destination	Protocol	Length	Info
6140	30.842760	192.168.29.31	44.228.249.3	HTTP	464	GET /static/app/partials/popular.html HTTP/1.1
6179	31.114218	44.228.249.3	192.168.29.31	HTTP	533	HTTP/1.1 200 OK (text/html)
6182	31.138685	192.168.29.31	44.228.249.3	HTTP	453	GET /ajax/popular?offset=0 HTTP/1.1
6183	31.139842	192.168.29.31	44.228.249.3	HTTP	466	GET /static/app/partials/itemsList.html HTTP/1.1
6222	31.410246	44.228.249.3	192.168.29.31	HTTP	1320	HTTP/1.1 200 OK (text/html)
6223	31.411724	44.228.249.3	192.168.29.31	HTTP	217	HTTP/1.1 200 OK (text/html)
6249	31.602887	2405:201:800…	2606:4700:8d…	HTTP	455	GET /favicon.ico HTTP/1.1
6273	32.283406	2606:4700:8d…	2405:201:800…	HTTP	79	HTTP/1.1 301 Moved Permanently (text/html)
5678	29.482241	192.168.29.31	44.228.249.3	TCP	66	54770 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
5679	29.483461	192.168.29.31	44.228.249.3	TCP	66	54771 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
5699	29.615458	192.168.29.31	44.228.249.3	TCP	66	54772 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
5706	29.744278	44.228.249.3	192.168.29.31	TCP	66	80 → 54770 [SYN, ACK] Seq=0 Ack=1 Win=62727 Len=0 MSS=1460 SACK_PERM WS=128
5707	29.744714	192.168.29.31	44.228.249.3	TCP	54	54770 → 80 [ACK] Seq=1 Ack=1 Win=65536 Len=0
5713	29.754275	44.228.249.3	192.168.29.31	TCP	66	80 → 54771 [SYN, ACK] Seq=0 Ack=1 Win=62727 Len=0 MSS=1460 SACK_PERM WS=128
5714	29.754659	192.168.29.31	44.228.249.3	TCP	54	54771 → 80 [ACK] Seq=1 Ack=1 Win=65536 Len=0
5717	29.876727	44.228.249.3	192.168.29.31	TCP	66	80 → 54772 [SYN, ACK] Seq=0 Ack=1 Win=62727 Len=0 MSS=1460 SACK_PERM WS=128
5718	29.876870	192.168.29.31	44.228.249.3	TCP	54	54772 → 80 [ACK] Seq=1 Ack=1 Win=65536 Len=0
5719	30.009694	44.228.249.3	192.168.29.31	TCP	54	80 → 54770 [ACK] Seq=1 Ack=437 Win=62336 Len=0

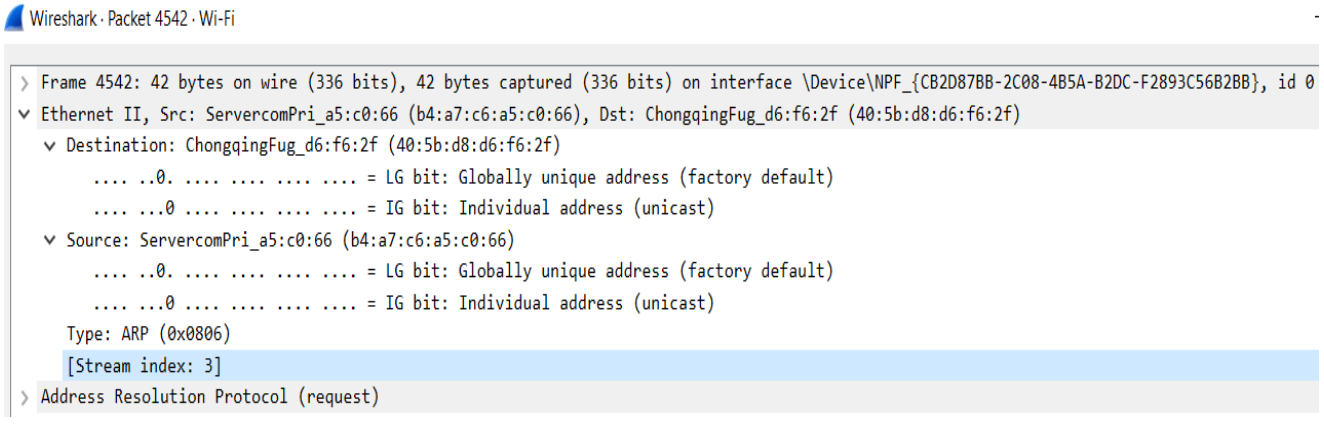


7. What are the manufacturers of your PC's Network Interface Card (NIC), and the servers NIC?

So according to the details, my PC's Network Interface Card (NIC) has the manufacturer: ServercomPri.

And the server's Network Interface Card (NIC) has the manufacturer: ChongqingFug.

8. What are the Hex values (shown in the raw bytes panel) of the two NICS Manufacturers OUIs?



Ans: The hex values (shown the raw bytes panel) of the two NICS Manufacturers OUIs are: b4:a7:c6:a5:c0:66 (my NIC raw bytes) and 40:5b:d8:d6:f6:2f (server NIC).

9. Find the following statistics:

a. What percentage of packets in your capture are TCP, and give an example of the higher level protocol which uses TCP?

b. What percentage of packets in your capture are UDP, and give an example of the higher level protocol which uses UDP?

Wireshark · Protocol Hierarchy Statistics · Wi-Fi

Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s	PDUs
Frame	100.0	6061	100.0	5306614	1107 k	0	0	0	6061
Ethernet	100.0	6061	1.6	84992	17 k	0	0	0	6061
Internet Protocol Version 6	82.6	5007	3.8	200280	41 k	0	0	0	5007
Transmission Control Protocol	82.6	5007	1.9	102588	21 k	3655	75500	15 k	5007
Transport Layer Security	22.2	1347	94.1	4992916	1042 k	1347	1520965	317 k	1604
Hypertext Transfer Protocol	0.1	4	0.6	34425	7186	2	708	147	4
Media Type	0.0	1	1.7	92629	19 k	1	92629	19 k	1
Line-based text data	0.0	1	0.0	134	27	1	134	27	1
Data	0.0	1	0.0	1	0	1	1	0	1
Internet Protocol Version 4	17.4	1054	0.4	21080	4400	0	0	0	1054
Transmission Control Protocol	17.4	1054	0.4	21552	4498	432	9112	1902	1054
Transport Layer Security	9.9	598	4.2	221390	46 k	598	214748	44 k	600
Hypertext Transfer Protocol	0.4	24	0.5	25499	5322	12	4424	923	24
Portable Network Graphics	0.0	1	0.1	2736	571	1	2736	571	1
Media Type	0.1	5	0.3	17377	3627	5	17377	3627	5
Line-based text data	0.1	6	0.2	11387	2377	6	11387	2377	6

From the above statistics, the percentage of TCP packets(in IPV6) is 82.6%. A protocol that uses TCP is HTTP.

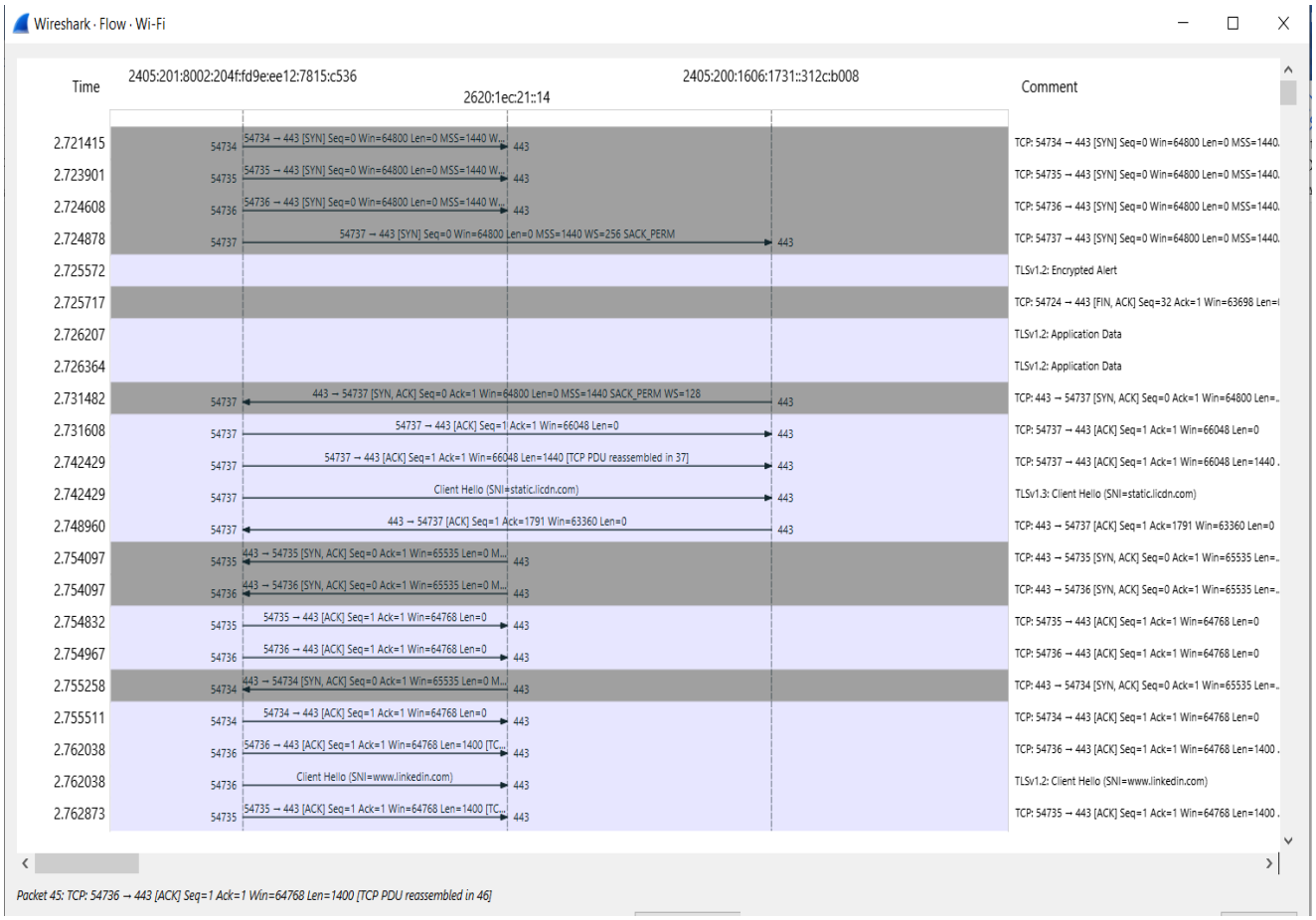
Wireshark · DNS · Wi-Fi

Packet Type	Count	Average	Min Val	Max Val	Rate (ms)	Percent	Burst Rate	Burst Start
Total Packets	252				0.0084	100%	0.3000	30.142
rcode	252				0.0084	100%	0.3000	30.142
opcodes	252				0.0084	100%	0.3000	30.142
Service Stats	0				0.0000	100%	-	-
Response Stats	0				0.0000	100%	-	-
Response	252				0.0084	100%	0.3000	30.142
Query Type	252				0.0084	100%	0.3000	30.142
Query Stats	0				0.0000	100%	-	-
Query Name	0				0.0000	100%	-	-
Payload size	252	73.52	28	305	0.0084	100%	0.3000	30.142
Class	252				0.0084	100%	0.3000	30.142
Answer Type	262				0.0087	100%	0.3200	30.152

A protocol that uses UDP is DNS.

10. Find the traffic flow Select the Statistics->Flow Graph menu option. Choose General Flow and

Network Source options, and click the OK button.



COMMENTS:

This assignment has significantly enhanced my understanding of network traffic analysis using Wireshark. By generating and capturing ICMP traffic, I observed the ARP resolution process and analyzed how ICMP echo requests and replies are exchanged. Additionally, by generating web traffic, I explored various protocols, such as HTTP, TCP, and DNS, and learned how to measure the time between HTTP GET requests and HTTP OK replies. The exercise of filtering packets, analyzing network flows, and examining packet bytes deepened my knowledge of how different protocols interact at the packet level. Moreover, I gained practical insights into identifying NIC manufacturers and analyzing traffic statistics, including the percentage of TCP and UDP packets. This hands-on experience has been invaluable in improving my understanding of network communication and protocol analysis. I am grateful to my teacher for their guidance throughout the assignment, which has greatly contributed to my learning.