



VIT®

Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

Computer Science & Engineering

CSE3501 – Information Security Analysis and Audit

LAB ASSIGNMENT 5

Submitted to Prof. RAJA SP

NAME: PUNIT MIDDHA

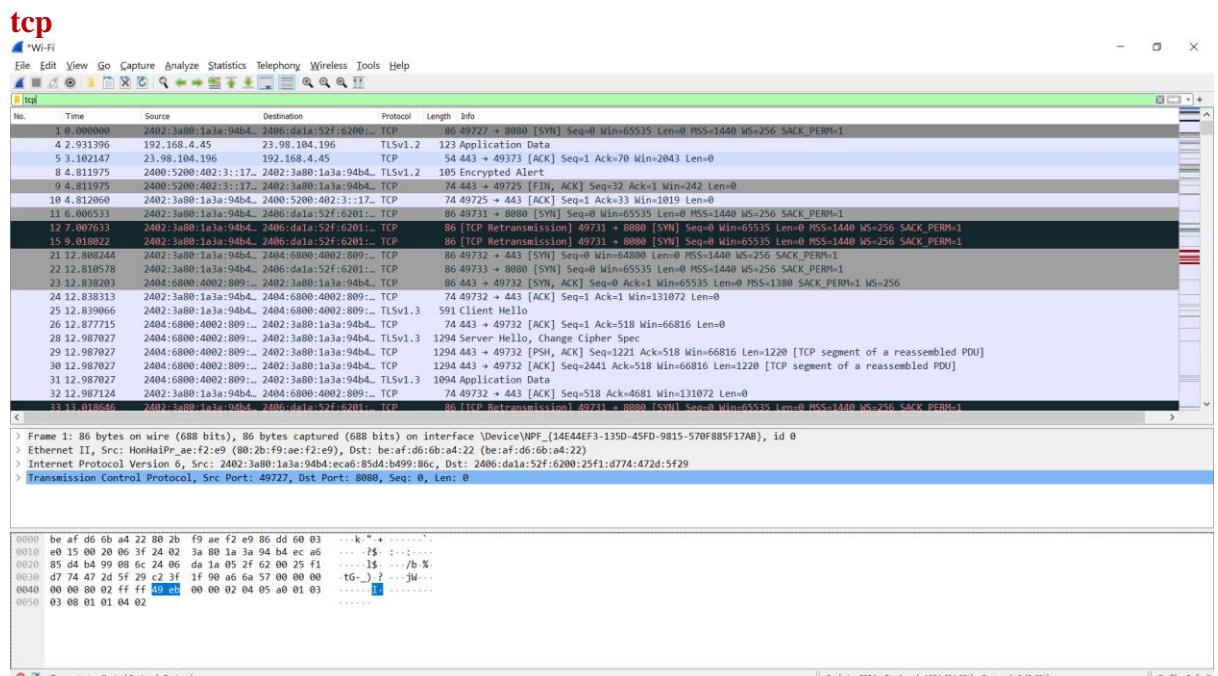
REG.NO: 19BCE2060

SLOT: L39+L40

DATE: 21/10/2021

Network Packets Sniffing using Wireshark

1. Filtering the packets by specifying a protocol



2. Filter the packets based on the port

tcp.port==443

No.	Time	Source	Destination	Protocol	Length	Info
4	2.931396	192.168.4.45	23.98.104.196	TLSv1.2	123	Application Data
5	3.102147	23.98.104.196	192.168.4.45	TCP	54	443 + 49373 [ACK] Seq=1 Ack=70 Win=2043 Len=0
8	4.811975	2400:5200:402::3::17:	2402:3a80:1a3a:94b4..	TLSv1.2	105	Encrypted Alert
9	4.811975	2400:5200:402::3::17:	2402:3a80:1a3a:94b4..	TCP	74	443 + 49725 [FIN, ACK] Seq=32 Ack=1 Win=242 Len=0
10	4.812060	2402:3a80:1a3a:94b4..	2400:5200:402::3::17:	TCP	86	443 + 49732 [SYN] Seq=0 Win=64800 Len=0 MSS=1440 WS=256 SACK_PERM=1
21	12.888244	2402:3a80:1a3a:94b4..	2404:6800:4002:899..	TCP	86	49732 + 443 [SYN] Seq=0 Win=64800 Len=0 MSS=1440 WS=256 SACK_PERM=1
23	12.888203	2404:6800:4002:899..	2402:3a80:1a3a:94b4..	TCP	86	443 + 49732 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1380 SACK_PERM=1 WS=256
24	12.888313	2402:3a80:1a3a:94b4..	2404:6800:4002:899..	TCP	74	49732 + 443 [ACK] Seq=1 Ack=1 Win=131072 Len=0
25	12.889666	2402:3a80:1a3a:94b4..	2404:6800:4002:899..	TLSv1.3	591	Client Hello
26	12.877715	2404:6800:4002:899..	2402:3a80:1a3a:94b4..	TCP	74	443 + 49732 [ACK] Seq=1 Ack=518 Win=66816 Len=0
28	12.987027	2404:6800:4002:899..	2402:3a80:1a3a:94b4..	TLSv1.3	1294	Server Hello, Change Cipher Spec
29	12.987027	2404:6800:4002:899..	2402:3a80:1a3a:94b4..	TCP	1294	443 + 49732 [PSH, ACK] Seq=1221 Ack=518 Win=66816 Len=1220 [TCP segment of a reassembled PDU]
30	12.987027	2404:6800:4002:899..	2402:3a80:1a3a:94b4..	TCP	1294	443 + 49732 [ACK] Seq=2441 Ack=518 Win=66816 Len=1220 [TCP segment of a reassembled PDU]
31	12.987027	2404:6800:4002:899..	2402:3a80:1a3a:94b4..	TLSv1.3	1094	Application Data
32	12.987124	2402:3a80:1a3a:94b4..	2404:6800:4002:899..	TCP	74	49732 + 443 [ACK] Seq=518 Ack=4681 Win=131072 Len=0
34	13.018804	2404:6800:4002:899..	2402:3a80:1a3a:94b4..	TCP	1094	[TCP Spurious Retransmission] 443 + 49732 [PSH, ACK] Seq=3661 Ack=518 Win=66816 Len=1020 Resassembly error, protocol TCP: New frame
35	13.018839	2402:3a80:1a3a:94b4..	2404:6800:4002:899..	TCP	86	[TCP Dup ACK 32:1] 49732 + 443 [ACK] Seq=518 Ack=4681 Win=131072 Len=0 SLE=3661 SRE=4681
36	13.155426	2402:3a80:1a3a:94b4..	2404:6800:4002:899..	TLSv1.3	138	Change Cipher Spec, Application Data
37	13.155943	2402:3a80:1a3a:94b4..	2404:6800:4002:899..	TLSv1.3	165	Application Data
38	13.156293	2402:3a80:1a3a:94b4..	2404:6800:4002:899..	TLSv1.3	380	Application Data
39	13.206469	2404:6800:4002:899..	2402:3a80:1a3a:94b4..	TLSv1.3	682	Application Data, Application Data

tcp.dstport==443

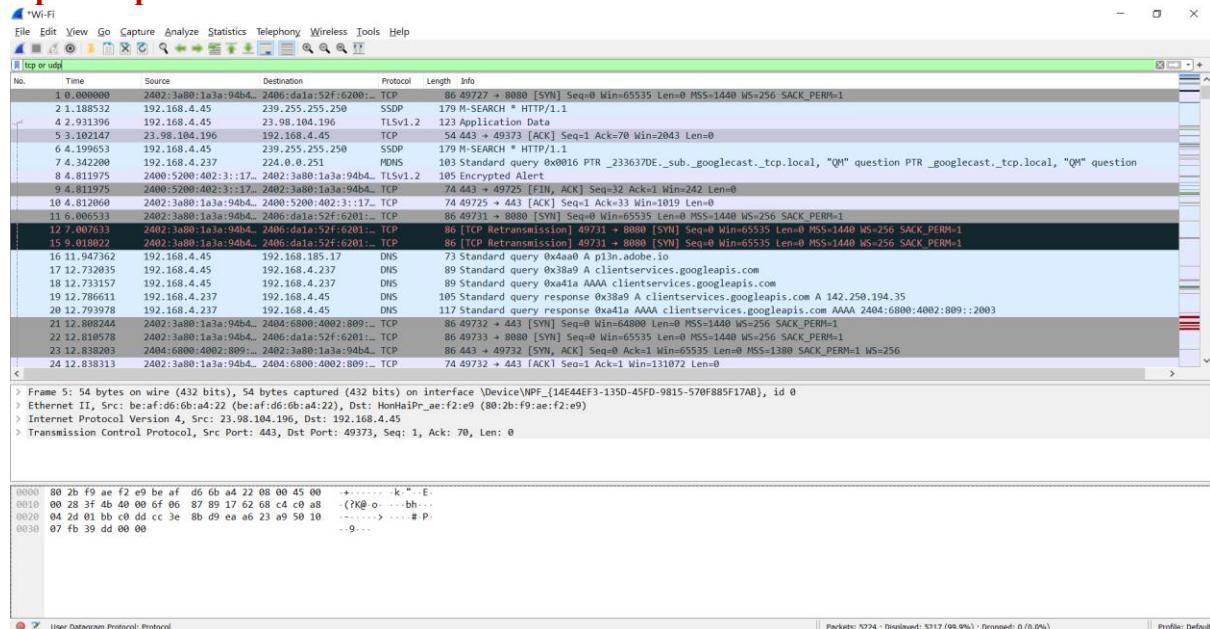
No.	Time	Source	Destination	Protocol	Length	Info
4	2.931396	192.168.4.45	23.98.104.196	TLSv1.2	123	Application Data
10	4.812060	23.98.104.196	192.168.4.45	TCP	74	443 + 49732 [ACK] Seq=1 Ack=33 Win=1019 Len=0
21	12.888244	2402:3a80:1a3a:94b4..	2404:6800:4002:899..	TCP	86	49732 + 443 [SYN] Seq=0 Win=64800 Len=0 MSS=1440 WS=256 SACK_PERM=1
24	12.888313	2402:3a80:1a3a:94b4..	2404:6800:4002:899..	TLSv1.3	591	Client Hello
25	12.889666	2402:3a80:1a3a:94b4..	2404:6800:4002:899..	TLSv1.3	1294	Server Hello, Change Cipher Spec
32	12.987124	2402:3a80:1a3a:94b4..	2404:6800:4002:899..	TCP	74	49732 + 443 [ACK] Seq=518 Ack=4681 Win=131072 Len=0
35	13.018839	2402:3a80:1a3a:94b4..	2404:6800:4002:899..	TCP	86	[TCP Dup ACK 32:1] 49732 + 443 [ACK] Seq=518 Ack=4681 Win=131072 Len=0 SLE=3661 SRE=4681
36	13.155426	2402:3a80:1a3a:94b4..	2404:6800:4002:899..	TLSv1.3	138	Change Cipher Spec, Application Data
37	13.155943	2402:3a80:1a3a:94b4..	2404:6800:4002:899..	TLSv1.3	165	Application Data
38	13.156293	2402:3a80:1a3a:94b4..	2404:6800:4002:899..	TLSv1.3	380	Application Data
39	13.206469	2404:6800:4002:899..	2402:3a80:1a3a:94b4..	TLSv1.3	682	Application Data, Application Data

tcp.srport==443

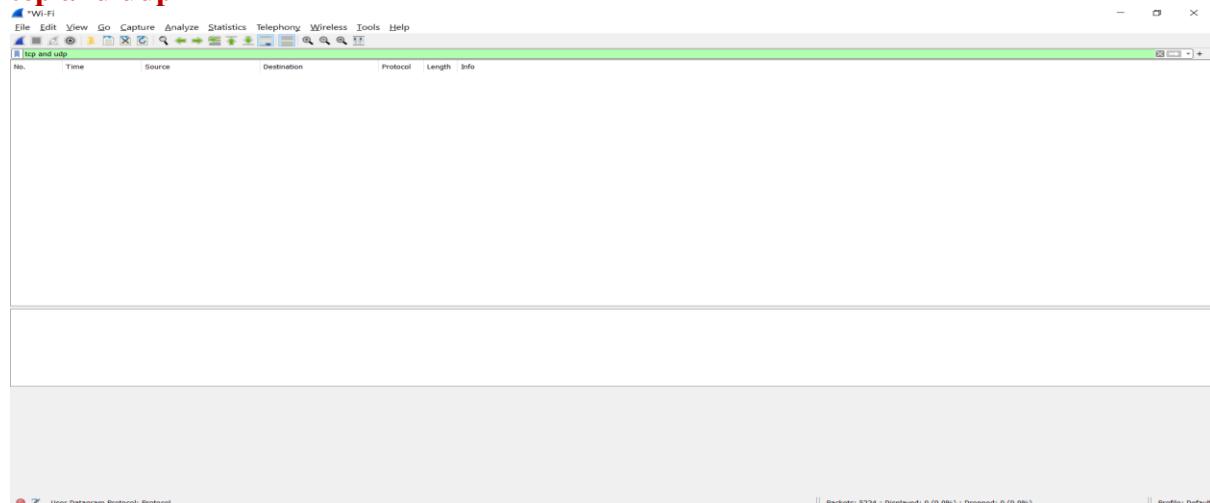
No.	Time	Source	Destination	Protocol	Length	Info
5	3.102147	23.98.104.196	192.168.4.45	TCP	54	443 + 49373 [ACK] Seq=1 Ack=70 Win=2043 Len=0
8	4.811975	2400:5200:402::3::17:	2402:3a80:1a3a:94b4..	TLSv1.2	105	Encrypted Alert
9	4.811975	2400:5200:402::3::17:	2402:3a80:1a3a:94b4..	TCP	74	443 + 49725 [FIN, ACK] Seq=32 Ack=1 Win=242 Len=0
23	12.888203	2404:6800:4002:899..	2402:3a80:1a3a:94b4..	TCP	86	443 + 49732 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1380 SACK_PERM=1 WS=256
26	12.877715	2404:6800:4002:899..	2402:3a80:1a3a:94b4..	TCP	74	49732 + 443 [ACK] Seq=518 Ack=4681 Win=131072 Len=0
28	12.987027	2404:6800:4002:899..	2402:3a80:1a3a:94b4..	TLSv1.3	1294	Server Hello, Change Cipher Spec
29	12.987027	2404:6800:4002:899..	2402:3a80:1a3a:94b4..	TCP	1294	443 + 49732 [PSH, ACK] Seq=1221 Ack=518 Win=66816 Len=1220 [TCP segment of a reassembled PDU]
30	12.987027	2404:6800:4002:899..	2402:3a80:1a3a:94b4..	TCP	1294	443 + 49732 [ACK] Seq=2441 Ack=518 Win=66816 Len=1220 [TCP segment of a reassembled PDU]
31	12.987027	2404:6800:4002:899..	2402:3a80:1a3a:94b4..	TLSv1.3	1094	Application Data
34	13.018804	2404:6800:4002:899..	2402:3a80:1a3a:94b4..	TCP	1094	[TCP Spurious Retransmission] 443 + 49732 [PSH, ACK] Seq=3661 Ack=518 Win=66816 Len=1020 Resassembly error, protocol TCP: New frame
39	13.206469	2404:6800:4002:899..	2402:3a80:1a3a:94b4..	TLSv1.3	682	Application Data, Application Data
40	13.206469	2404:6800:4002:899..	2402:3a80:1a3a:94b4..	TCP	74	443 + 49732 [ACK] Seq=4681 Ack=518 Win=66816 Len=0
41	13.206780	2404:6800:4002:899..	2402:3a80:1a3a:94b4..	TLSv1.3	105	Application Data
46	13.253205	2404:6800:4002:899..	2402:3a80:1a3a:94b4..	TCP	74	443 + 49732 [ACK] Seq=5320 Ack=1011 Win=67840 Len=0
47	13.417936	2404:6800:4002:899..	2402:3a80:1a3a:94b4..	TLSv1.3	334	Application Data
48	13.420592	2404:6800:4002:899..	2402:3a80:1a3a:94b4..	TLSv1.3	105	Application Data
49	13.458399	2404:6800:4002:899..	2402:3a80:1a3a:94b4..	TLSv1.3	113	Application Data
107	13.931899	64:ffff:36c0:a652	2402:3a80:1a3a:94b4..	TCP	86	443 + 49737 [ACK] Seq=5650 Ack=1050 Win=67840 Len=0
108	13.931899	64:ffff:82d3:1e36	2402:3a80:1a3a:94b4..	TCP	86	443 + 49738 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1380 SACK_PERM=1 WS=256

3. Filter results based on 'or / and'

tcp or udp

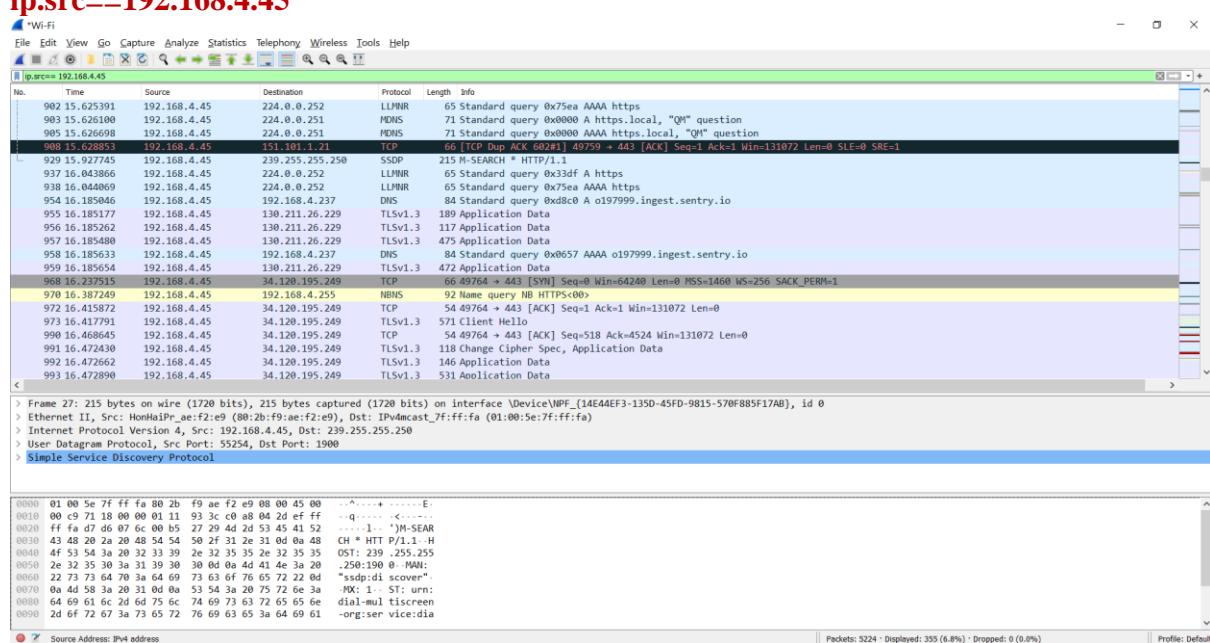


tcp and udp



4. Filter results based on IP addresses

ip.src==192.168.4.45



ip.dst== 192.168.4.237

This screenshot shows a Wireshark capture window with the filter set to "ip.dst== 192.168.4.237". The list view displays numerous DNS requests from various IP addresses to the destination 192.168.4.237. The details view shows the query for "www.googleapis.com" at frame 263. The bytes view shows the raw hex and ASCII data for this request.

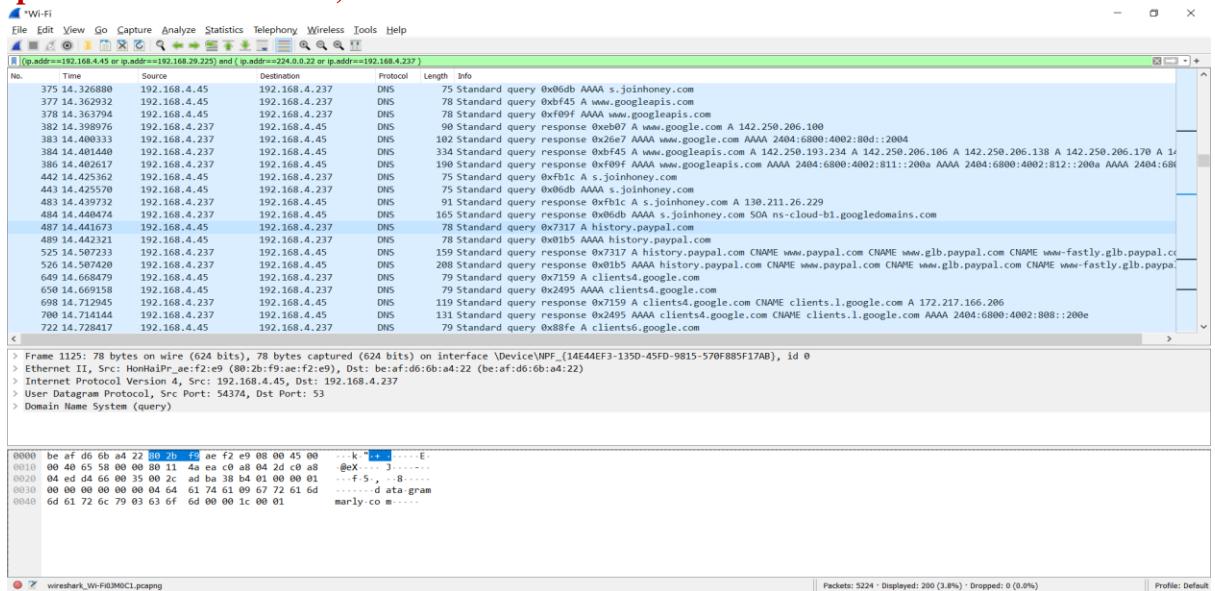
5. Filter results based on the byte sequence tcp contains 23:00:00

This screenshot shows a Wireshark capture window with the filter set to "tcp contains 23:00:00". It displays a series of TLS handshakes between multiple clients and a single server. The details view shows the "Client Hello" message from a client at frame 246, which includes the byte sequence 23:00:00 in its list of supported cipher suites.

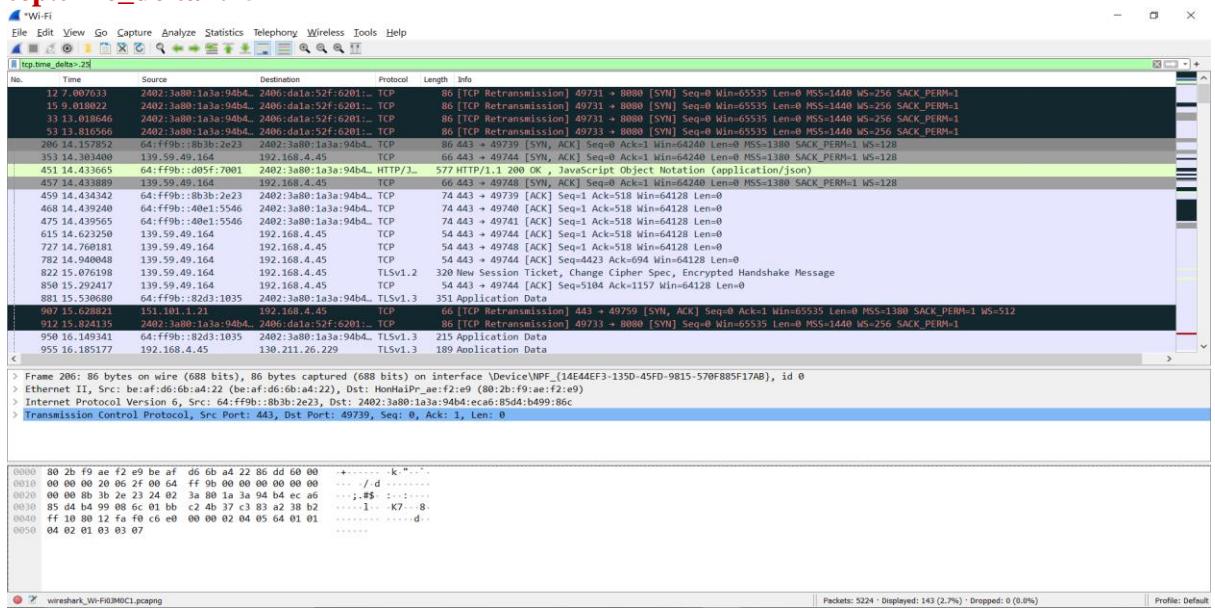
6. Filter results based on two IP addresses ip.addr==192.168.4.45 && ip.addr==130.211.26.229

This screenshot shows a Wireshark capture window with the filter set to "ip.addr==192.168.4.45 && ip.addr==130.211.26.229". It shows a series of TCP connections between the two specified IP addresses. The details view shows a segment of a reassembled PDU at frame 486, which is part of a larger conversation between the two hosts.

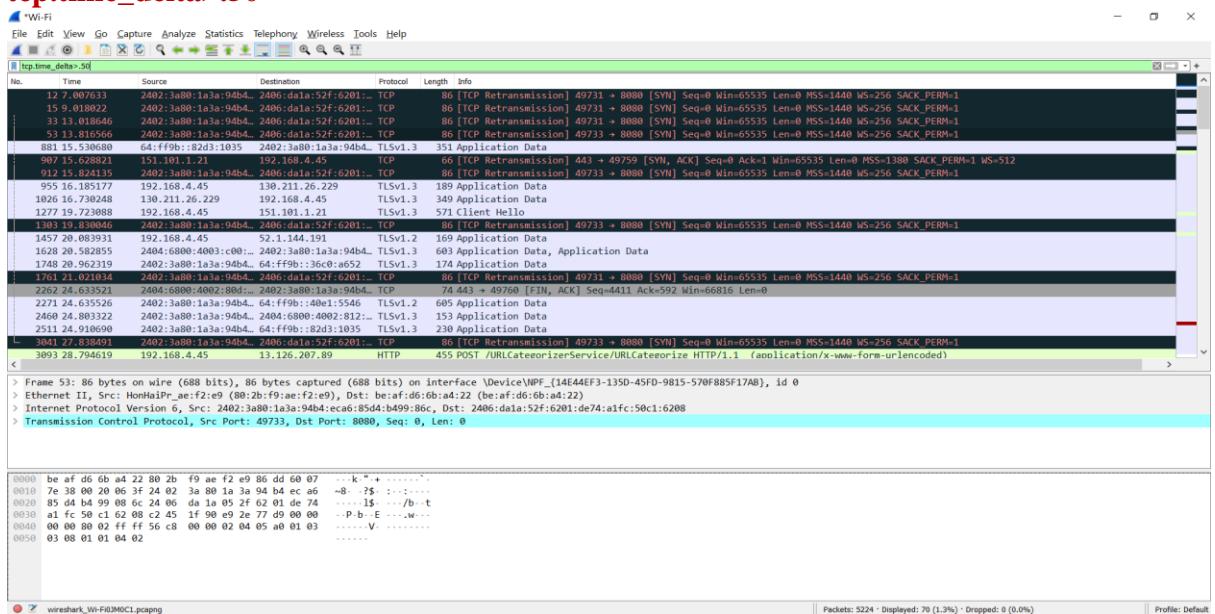
(ip.addr==192.168.4.45 or ip.addr==192.168.29.225) and (ip.addr==224.0.0.22 or ip.addr==192.168.4.237)



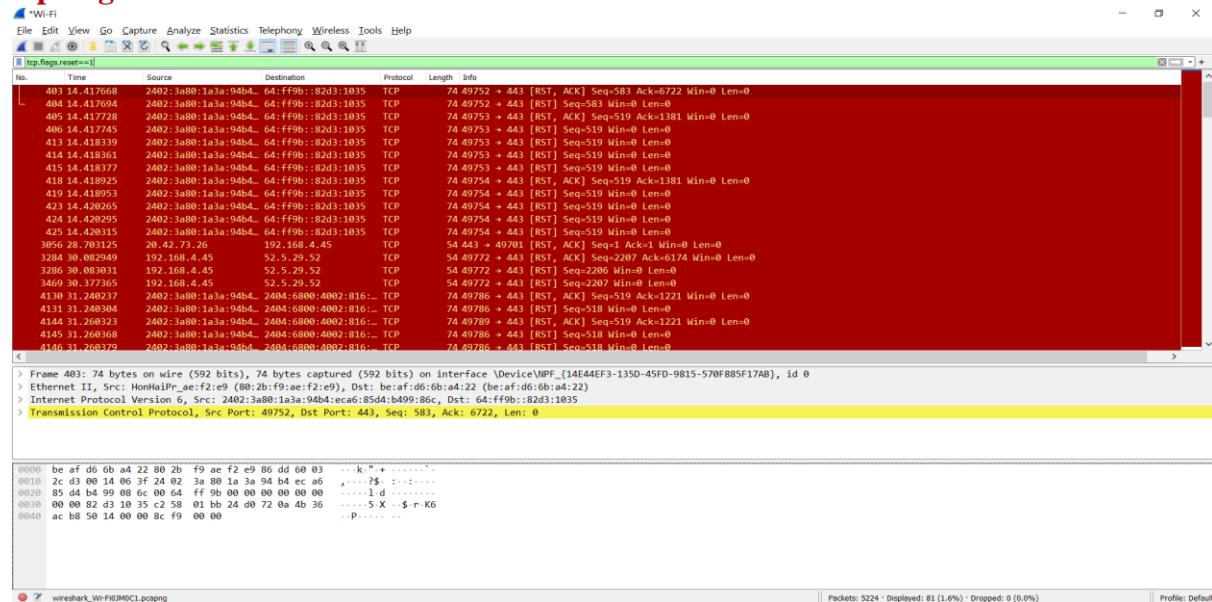
7. Filter the results based on the timestamp $\text{tcp.time_delta} > .25$



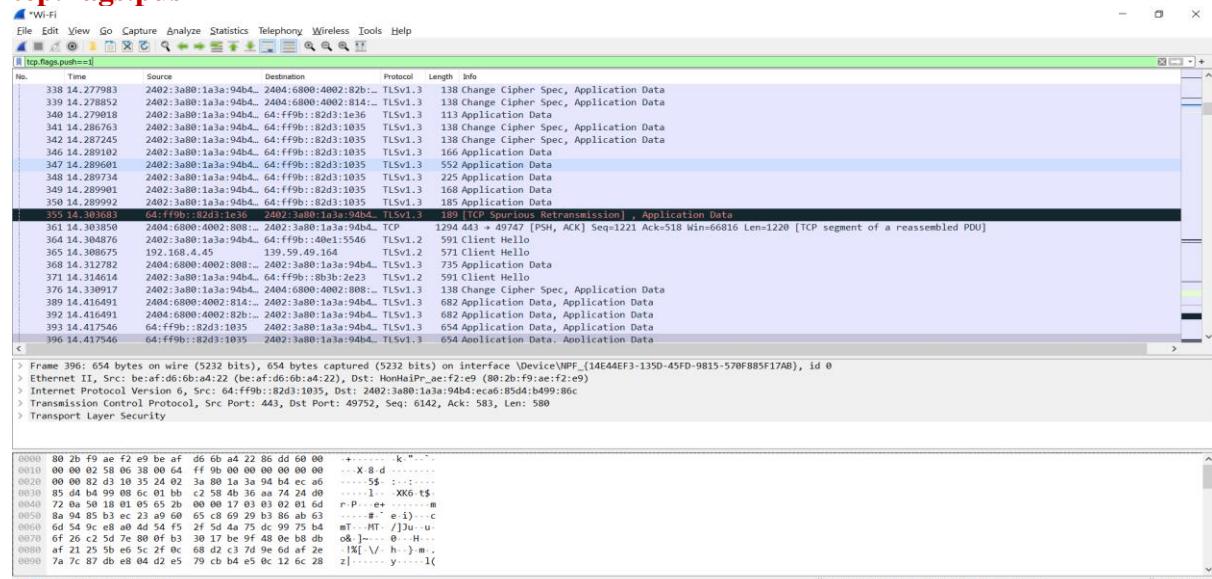
tcp.time_delta > .50



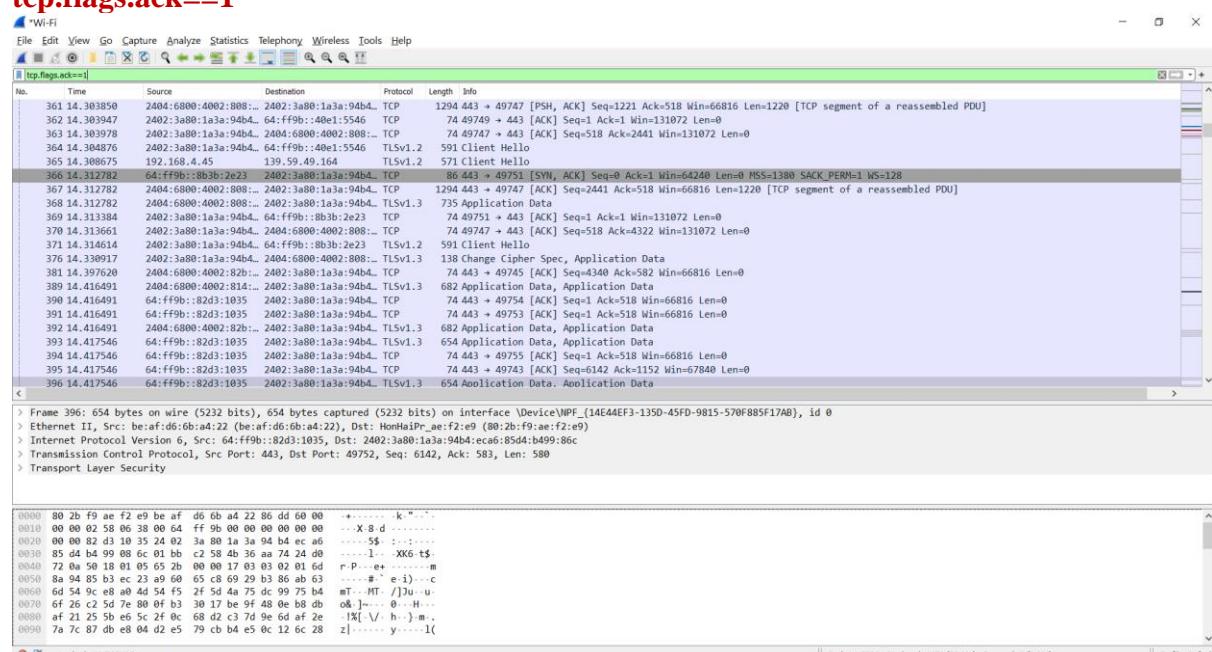
8. Filter the results based on the flags(ACK, SYN, PSH, RST)
tcp.flags.reset==1



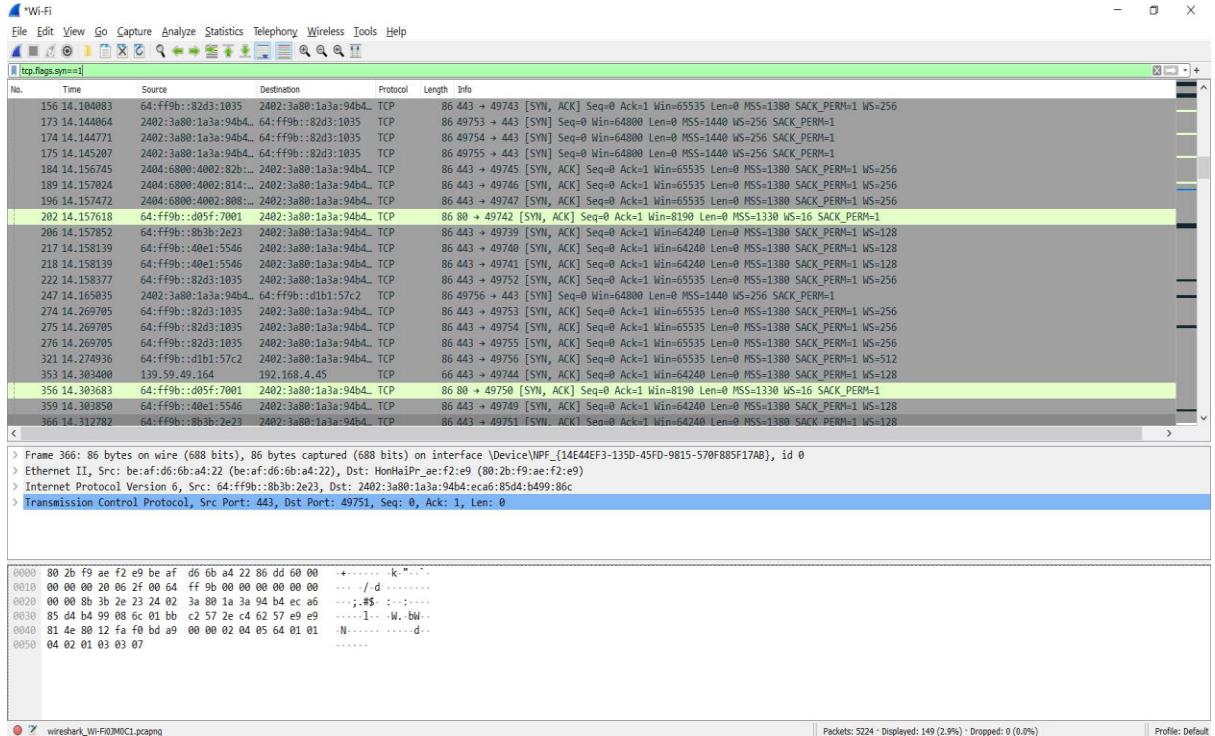
tcp.flags.push==1



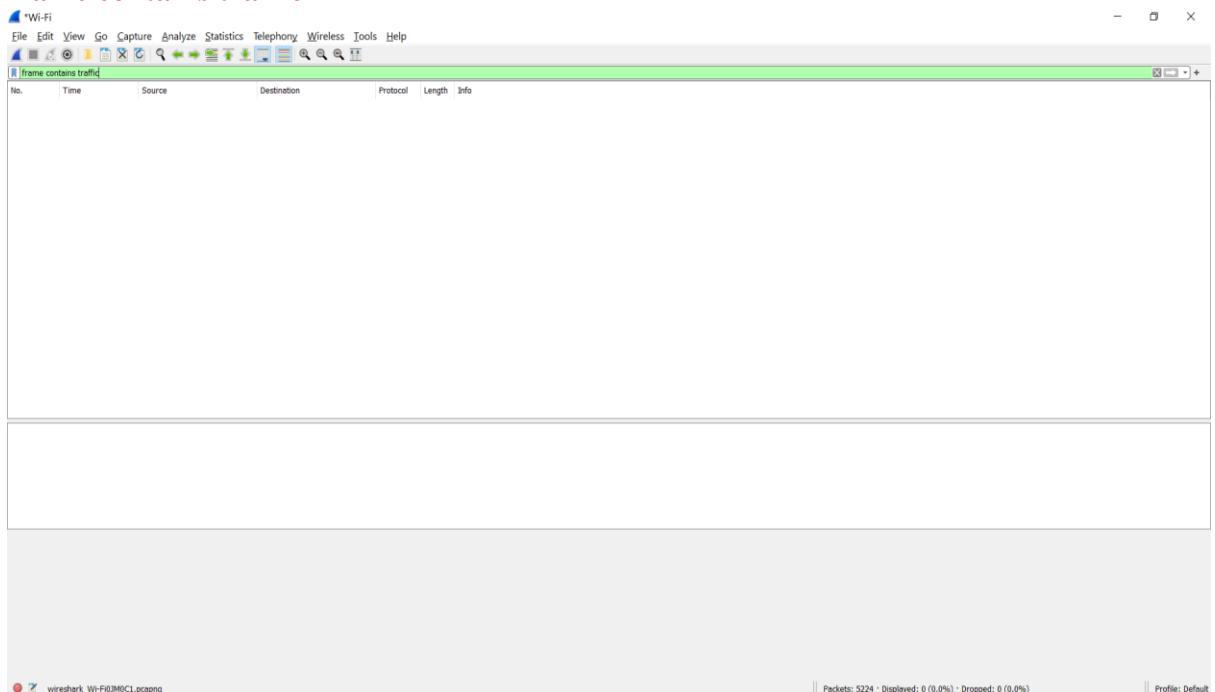
tcp.flags.ack==1



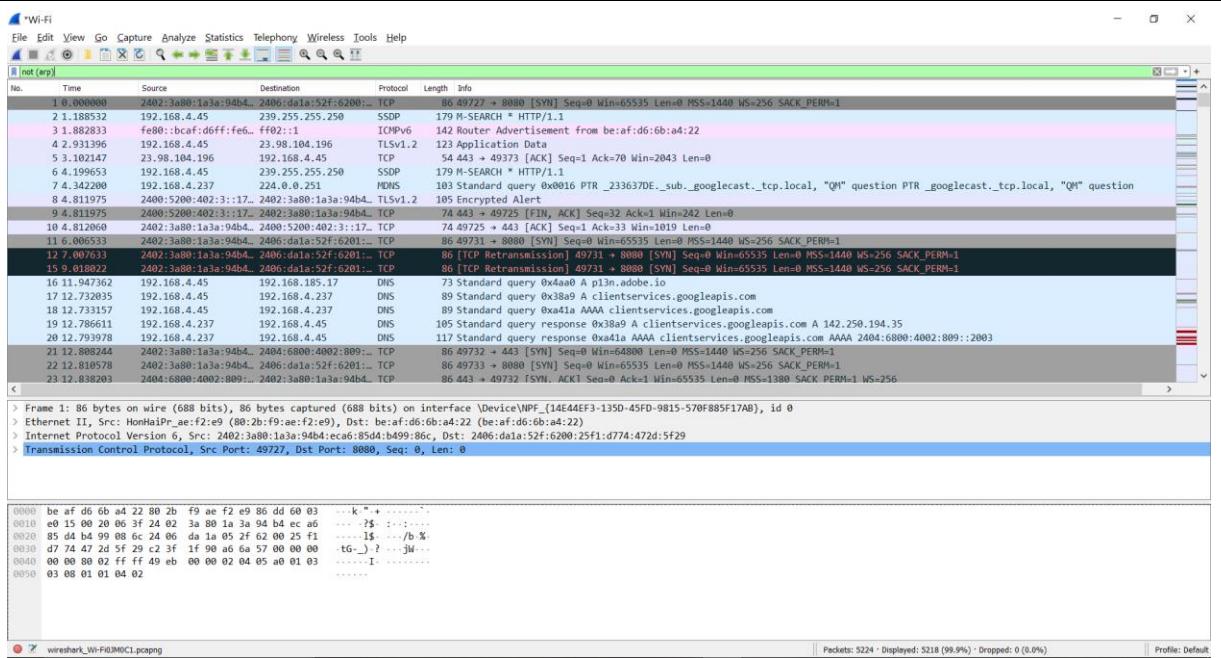
tcp.flags.syn==1



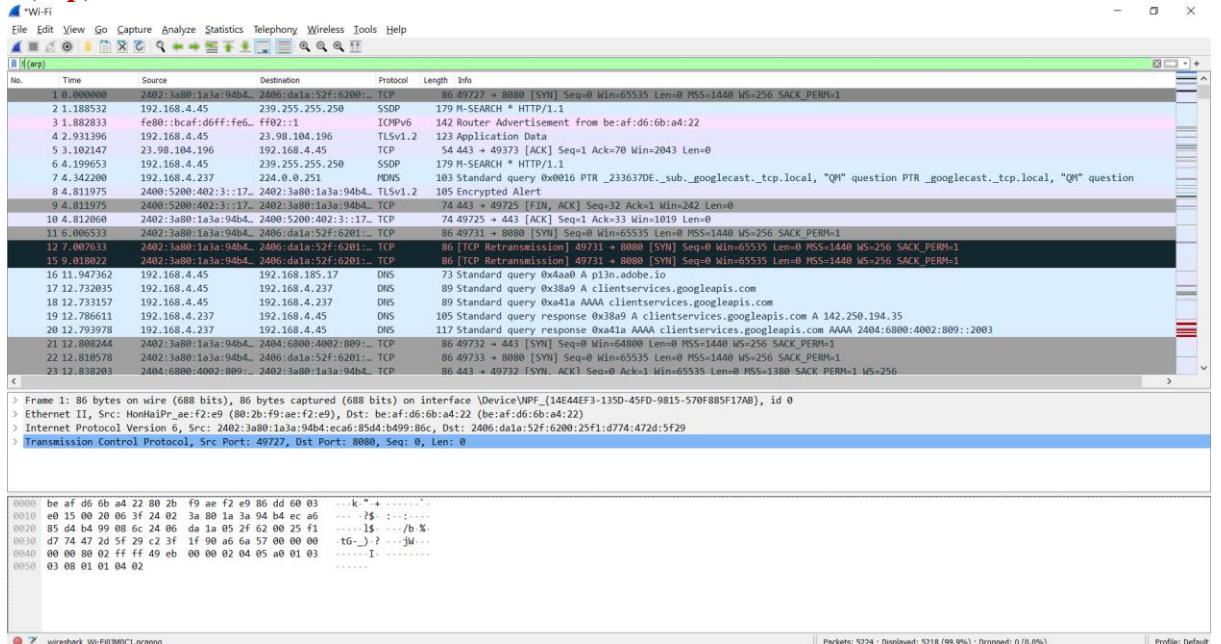
frame contains traffic



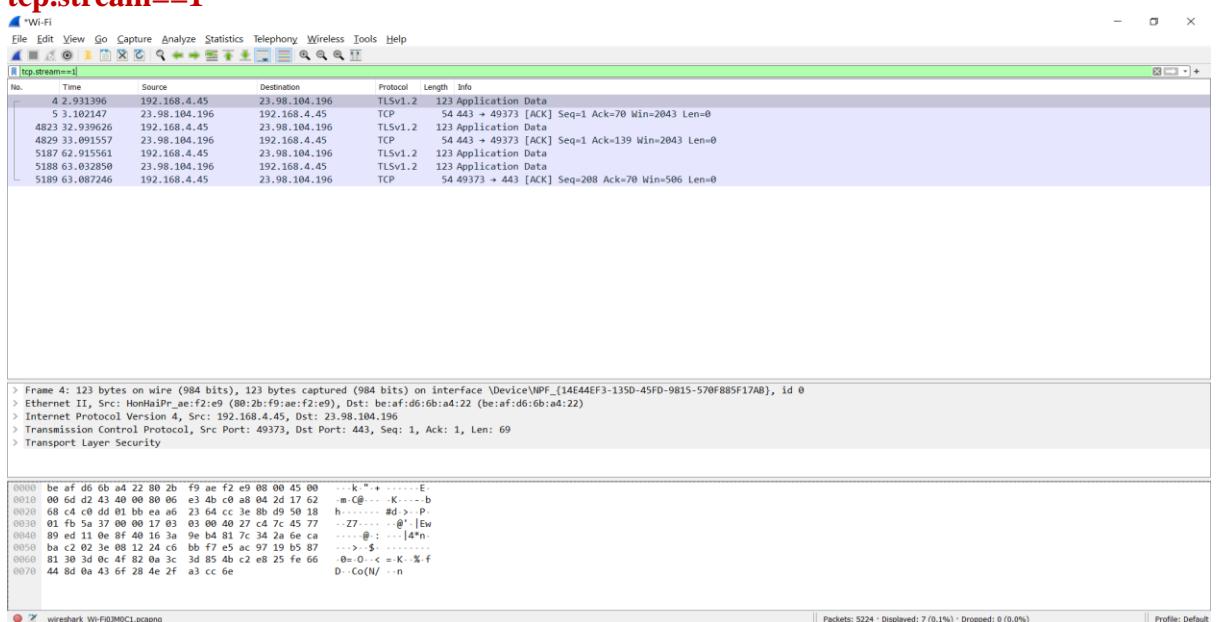
10. Filter the packets based on NOT (!) not (arp)



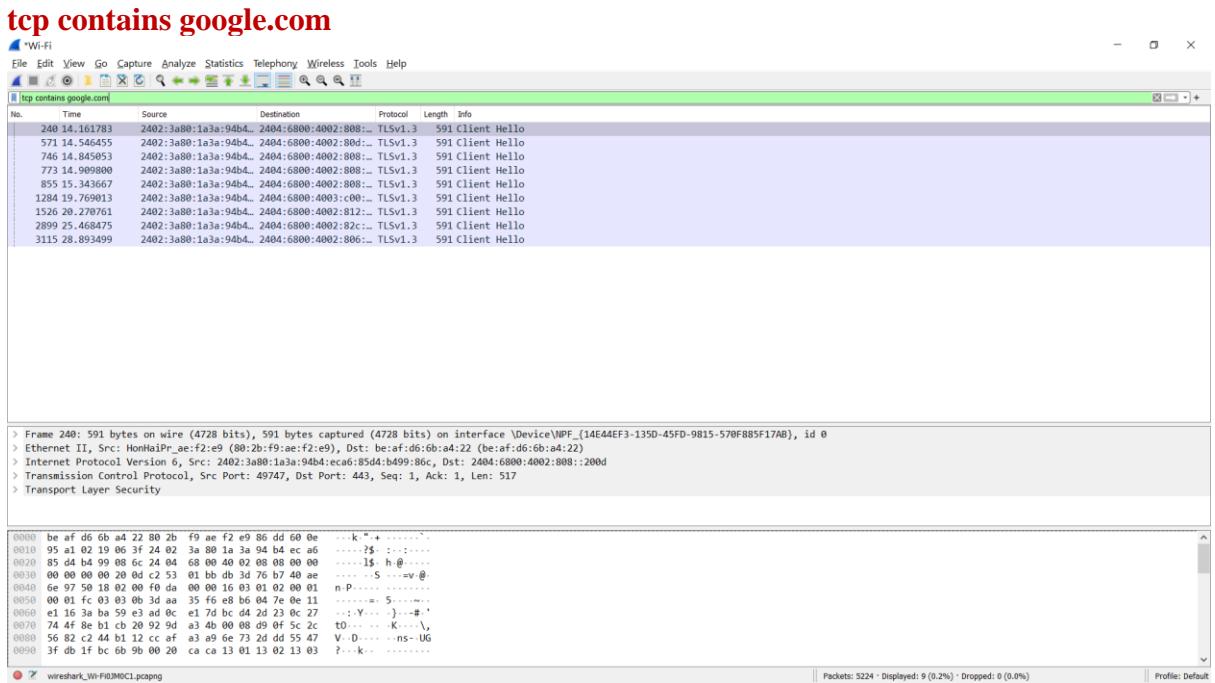
! (arp)



11. Filter the packets based on stream `tcp.stream==1`



12. Filter based on the website



13. Filter the packets based on retransmission and duplicate acks

