CA Assignment 4 – Wireshark 18317256

```
Command Prompt

Microsoft Windows [Version 10.0.22621.1555]
(c) Microsoft Corporation. All rights reserved.

C:\Users\nicho>ping 8.8.8.8 -n 10

Pinging 8.8.8.8 with 32 bytes of data:
Reply from 8.8.8.8: bytes=32 time=14ms TTL=55
Reply from 8.8.8.8: bytes=32 time=27ms TTL=55
Reply from 8.8.8.8: bytes=32 time=16ms TTL=55
Reply from 8.8.8.8: bytes=32 time=13ms TTL=55
Reply from 8.8.8.8: bytes=32 time=13ms TTL=55
Reply from 8.8.8.8: bytes=32 time=13ms TTL=55
Reply from 8.8.8.8: bytes=32 time=14ms TTL=55
Reply from 8.8.8.8: bytes=32 time=15ms TTL=55
Reply from 8.8.8.8: bytes=32 time=15ms TTL=55
Reply from 8.8.8.8: bytes=32 time=28ms TTL=55

Ping statistics for 8.8.8.8:
    Packets: Sent = 10, Received = 10, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 13ms, Maximum = 28ms, Average = 16ms

C:\Users\nicho>nslookup www.ucd.ie
Server:  UnKnown
Address:  192.168.3.1

Non-authoritative answer:
Name:    www.ucd.ie
Addresses:  54.229.218.22
          52.17.106.26


C:\Users\nicho>ftp ftp.sunet.se
Connected to sunet.ftp.acc.umu.se.
220 Please use http://ftp.acc.umu.se/ whenever possible.
200 Always in UTF8 mode.
User (sunet.ftp.acc.umu.se:(none)): anonymous
331 Please specify the password.
Password:
230 Login successful.
ftp> quit
221 Goodbye.

C:\Users\nicho>
```

Questions

1) What is the IP address of the network interface that you selected? How did you check the IP address?

192.168.3.129 is the address of the wifi network interface. I checked the IP address by filtering to only HTTP requests and any of the GET requests came from this IP. This is also verified by using ipconfig in the command prompt.

```
Wireless LAN adapter WiFi:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::2f03:39e4:222a:d9d%21
   IPv4 Address. . . . . . . . . . . : 192.168.3.129
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.3.1
```

2) Fill in the following table using the details of the created PCAP file.

Time span, s: 270.247s

Total packets in the capture: 125222

Bytes, MiB: 134828718

Average packet size, B: 1077

Average packets per seconds, pps: 463.4

Average bits per second, b/s: 3991 k

Statistics

Measurement	Captured	Displayed	Marked
Packets	125222	255 (0.2%)	—
Time span, s	270.247	267.002	—
Average pps	463.4	1.0	—
Average packet size, B	1077	104	—
Bytes	134828718	26498 (0.0%)	0
Average bytes/s	498 k	99	—
Average bits/s	3991 k	793	—

3) a) What is the command for filtering all the packets received (inbound) to the selected interface?

Ip.dest == 192.168.3.129

b) What is the command for filtering all the packets exited (outbound) from the selected interface?

Ip.src == 192.168.3.129

4) a) Why do we use ping command?

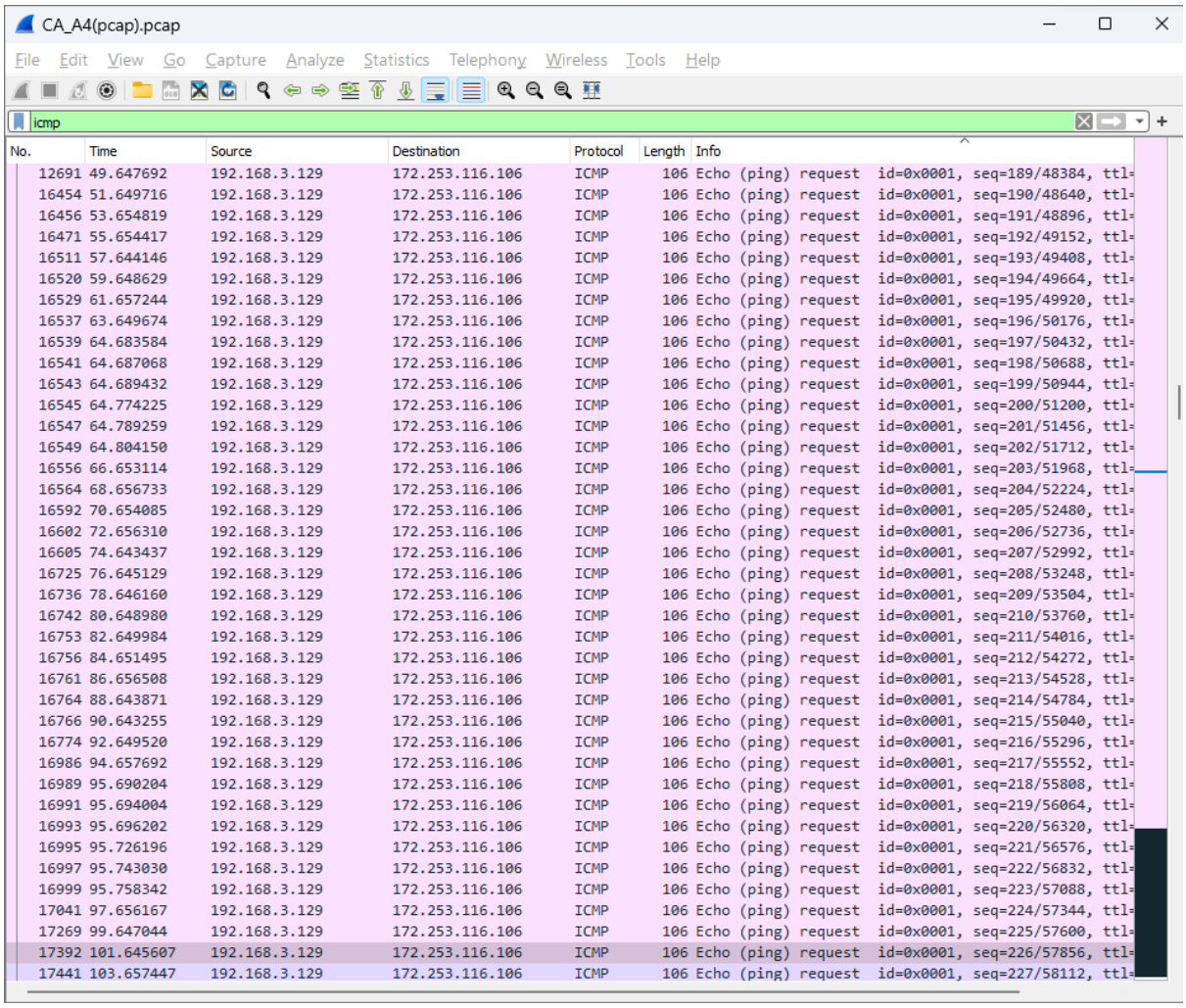
To test connectivity between the devices.

b) What is the underlying protocol related to ping?

ICMP (internet control message protocol)

c) What is the wireshark filter for ping frames?

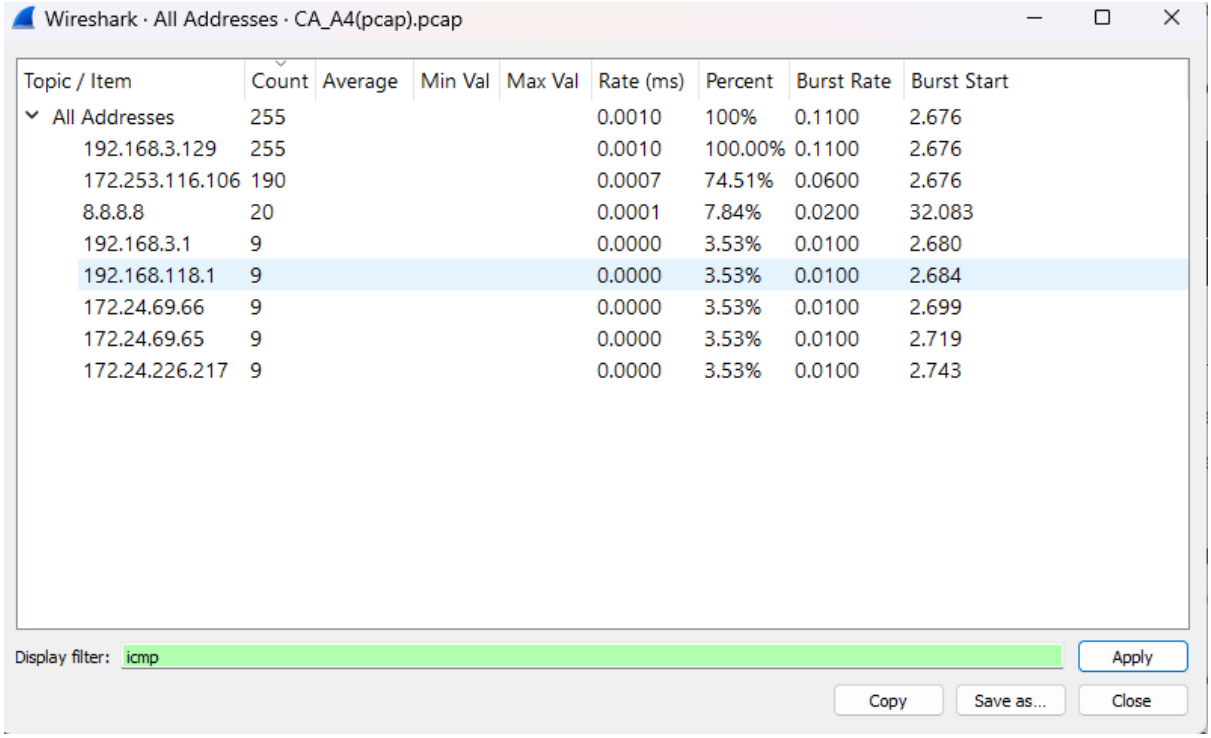
Icmp



No.	Time	Source	Destination	Protocol	Length	Info
12691	49.647692	192.168.3.129	172.253.116.106	ICMP	106	Echo (ping) request id=0x0001, seq=189/48384, ttl=
16454	51.649716	192.168.3.129	172.253.116.106	ICMP	106	Echo (ping) request id=0x0001, seq=190/48640, ttl=
16456	53.654819	192.168.3.129	172.253.116.106	ICMP	106	Echo (ping) request id=0x0001, seq=191/48896, ttl=
16471	55.654417	192.168.3.129	172.253.116.106	ICMP	106	Echo (ping) request id=0x0001, seq=192/49152, ttl=
16511	57.644146	192.168.3.129	172.253.116.106	ICMP	106	Echo (ping) request id=0x0001, seq=193/49408, ttl=
16520	59.648629	192.168.3.129	172.253.116.106	ICMP	106	Echo (ping) request id=0x0001, seq=194/49664, ttl=
16529	61.657244	192.168.3.129	172.253.116.106	ICMP	106	Echo (ping) request id=0x0001, seq=195/49920, ttl=
16537	63.649674	192.168.3.129	172.253.116.106	ICMP	106	Echo (ping) request id=0x0001, seq=196/50176, ttl=
16539	64.683584	192.168.3.129	172.253.116.106	ICMP	106	Echo (ping) request id=0x0001, seq=197/50432, ttl=
16541	64.687068	192.168.3.129	172.253.116.106	ICMP	106	Echo (ping) request id=0x0001, seq=198/50688, ttl=
16543	64.689432	192.168.3.129	172.253.116.106	ICMP	106	Echo (ping) request id=0x0001, seq=199/50944, ttl=
16545	64.774225	192.168.3.129	172.253.116.106	ICMP	106	Echo (ping) request id=0x0001, seq=200/51200, ttl=
16547	64.789259	192.168.3.129	172.253.116.106	ICMP	106	Echo (ping) request id=0x0001, seq=201/51456, ttl=
16549	64.804150	192.168.3.129	172.253.116.106	ICMP	106	Echo (ping) request id=0x0001, seq=202/51712, ttl=
16556	66.653114	192.168.3.129	172.253.116.106	ICMP	106	Echo (ping) request id=0x0001, seq=203/51968, ttl=
16564	68.656733	192.168.3.129	172.253.116.106	ICMP	106	Echo (ping) request id=0x0001, seq=204/52224, ttl=
16592	70.654085	192.168.3.129	172.253.116.106	ICMP	106	Echo (ping) request id=0x0001, seq=205/52480, ttl=
16602	72.656310	192.168.3.129	172.253.116.106	ICMP	106	Echo (ping) request id=0x0001, seq=206/52736, ttl=
16605	74.643437	192.168.3.129	172.253.116.106	ICMP	106	Echo (ping) request id=0x0001, seq=207/52992, ttl=
16725	76.645129	192.168.3.129	172.253.116.106	ICMP	106	Echo (ping) request id=0x0001, seq=208/53248, ttl=
16736	78.646160	192.168.3.129	172.253.116.106	ICMP	106	Echo (ping) request id=0x0001, seq=209/53504, ttl=
16742	80.648980	192.168.3.129	172.253.116.106	ICMP	106	Echo (ping) request id=0x0001, seq=210/53760, ttl=
16753	82.649984	192.168.3.129	172.253.116.106	ICMP	106	Echo (ping) request id=0x0001, seq=211/54016, ttl=
16756	84.651495	192.168.3.129	172.253.116.106	ICMP	106	Echo (ping) request id=0x0001, seq=212/54272, ttl=
16761	86.656508	192.168.3.129	172.253.116.106	ICMP	106	Echo (ping) request id=0x0001, seq=213/54528, ttl=
16764	88.643871	192.168.3.129	172.253.116.106	ICMP	106	Echo (ping) request id=0x0001, seq=214/54784, ttl=
16766	90.643255	192.168.3.129	172.253.116.106	ICMP	106	Echo (ping) request id=0x0001, seq=215/55040, ttl=
16774	92.649520	192.168.3.129	172.253.116.106	ICMP	106	Echo (ping) request id=0x0001, seq=216/55296, ttl=
16986	94.657692	192.168.3.129	172.253.116.106	ICMP	106	Echo (ping) request id=0x0001, seq=217/55552, ttl=
16989	95.690204	192.168.3.129	172.253.116.106	ICMP	106	Echo (ping) request id=0x0001, seq=218/55808, ttl=
16991	95.694004	192.168.3.129	172.253.116.106	ICMP	106	Echo (ping) request id=0x0001, seq=219/56064, ttl=
16993	95.696202	192.168.3.129	172.253.116.106	ICMP	106	Echo (ping) request id=0x0001, seq=220/56320, ttl=
16995	95.726196	192.168.3.129	172.253.116.106	ICMP	106	Echo (ping) request id=0x0001, seq=221/56576, ttl=
16997	95.743030	192.168.3.129	172.253.116.106	ICMP	106	Echo (ping) request id=0x0001, seq=222/56832, ttl=
16999	95.758342	192.168.3.129	172.253.116.106	ICMP	106	Echo (ping) request id=0x0001, seq=223/57088, ttl=
17041	97.656167	192.168.3.129	172.253.116.106	ICMP	106	Echo (ping) request id=0x0001, seq=224/57344, ttl=
17269	99.647044	192.168.3.129	172.253.116.106	ICMP	106	Echo (ping) request id=0x0001, seq=225/57600, ttl=
17392	101.645607	192.168.3.129	172.253.116.106	ICMP	106	Echo (ping) request id=0x0001, seq=226/57856, ttl=
17441	103.657447	192.168.3.129	172.253.116.106	ICMP	106	Echo (ping) request id=0x0001, seq=227/58112, ttl=

d) How many packets can you see when you apply this filter?

255



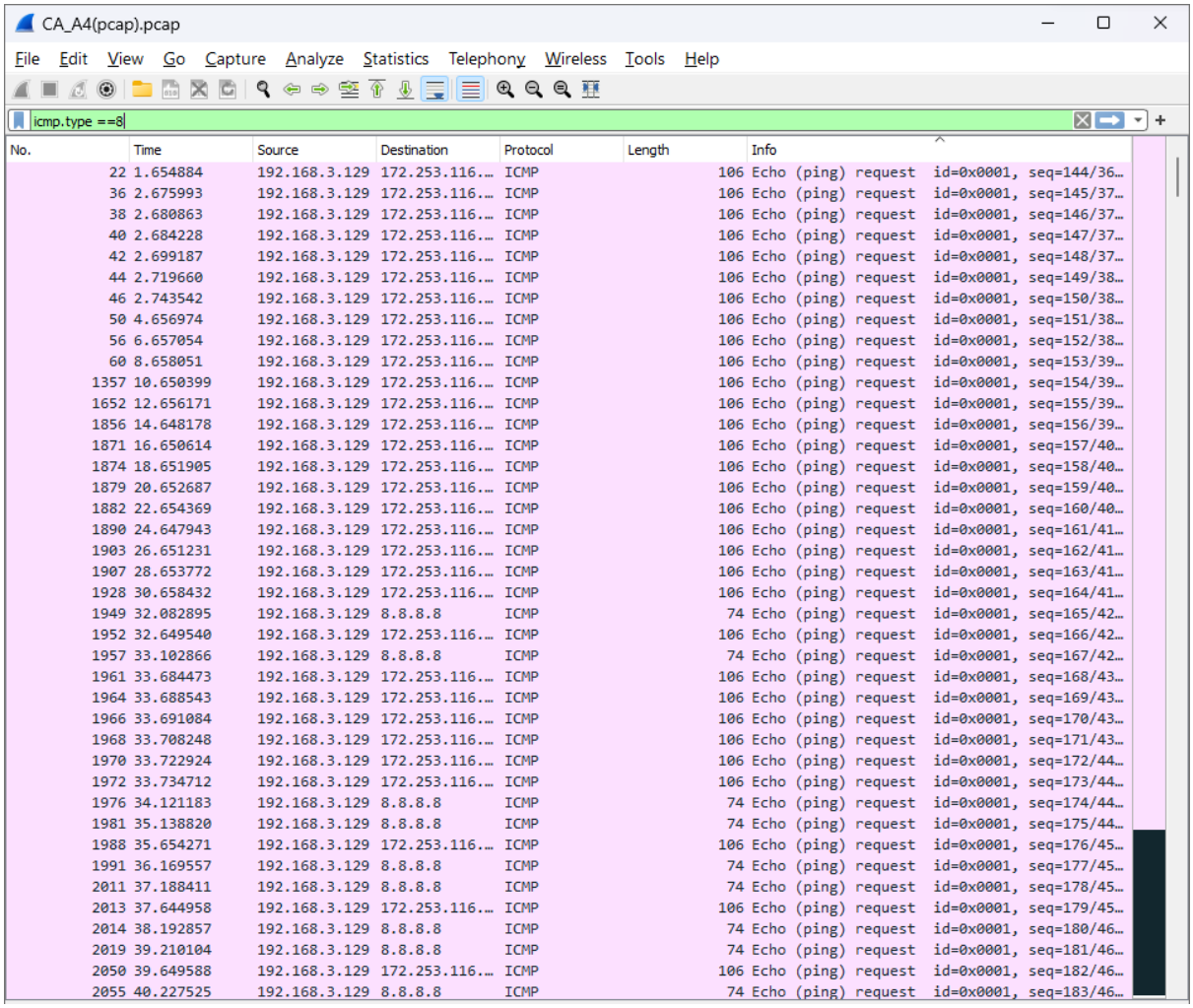
Topic / Item	Count	Average	Min Val	Max Val	Rate (ms)	Percent	Burst Rate	Burst Start
All Addresses	255				0.0010	100%	0.1100	2.676
192.168.3.129	255				0.0010	100.00%	0.1100	2.676
172.253.116.106	190				0.0007	74.51%	0.0600	2.676
8.8.8.8	20				0.0001	7.84%	0.0200	32.083
192.168.3.1	9				0.0000	3.53%	0.0100	2.680
192.168.118.1	9				0.0000	3.53%	0.0100	2.684
172.24.69.66	9				0.0000	3.53%	0.0100	2.699
172.24.69.65	9				0.0000	3.53%	0.0100	2.719
172.24.226.217	9				0.0000	3.53%	0.0100	2.743

e) What is the reason for seeing more packets than ping requests (here we sent 10 requests)?

Because other processes outside of the ping requests can utilise ICMP protocol in the background

f) How can we filter only ping request frames (sent 10 requests) using wireshark?

Use the filter icmp.type == 8



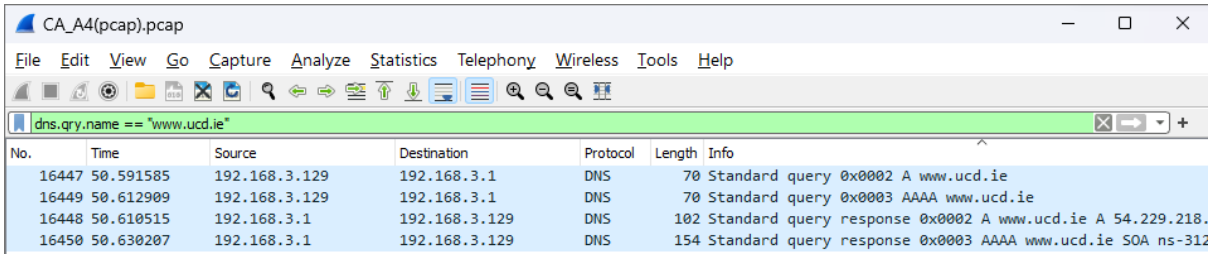
No.	Time	Source	Destination	Protocol	Length	Info
22	1.654884	192.168.3.129	172.253.116.…	ICMP	106	Echo (ping) request id=0x0001, seq=144/36…
36	2.675993	192.168.3.129	172.253.116.…	ICMP	106	Echo (ping) request id=0x0001, seq=145/37…
38	2.680863	192.168.3.129	172.253.116.…	ICMP	106	Echo (ping) request id=0x0001, seq=146/37…
40	2.684228	192.168.3.129	172.253.116.…	ICMP	106	Echo (ping) request id=0x0001, seq=147/37…
42	2.699187	192.168.3.129	172.253.116.…	ICMP	106	Echo (ping) request id=0x0001, seq=148/37…
44	2.719660	192.168.3.129	172.253.116.…	ICMP	106	Echo (ping) request id=0x0001, seq=149/38…
46	2.743542	192.168.3.129	172.253.116.…	ICMP	106	Echo (ping) request id=0x0001, seq=150/38…
50	4.656974	192.168.3.129	172.253.116.…	ICMP	106	Echo (ping) request id=0x0001, seq=151/38…
56	6.657054	192.168.3.129	172.253.116.…	ICMP	106	Echo (ping) request id=0x0001, seq=152/38…
60	8.658051	192.168.3.129	172.253.116.…	ICMP	106	Echo (ping) request id=0x0001, seq=153/39…
1357	10.650399	192.168.3.129	172.253.116.…	ICMP	106	Echo (ping) request id=0x0001, seq=154/39…
1652	12.656171	192.168.3.129	172.253.116.…	ICMP	106	Echo (ping) request id=0x0001, seq=155/39…
1856	14.648178	192.168.3.129	172.253.116.…	ICMP	106	Echo (ping) request id=0x0001, seq=156/39…
1871	16.650614	192.168.3.129	172.253.116.…	ICMP	106	Echo (ping) request id=0x0001, seq=157/40…
1874	18.651905	192.168.3.129	172.253.116.…	ICMP	106	Echo (ping) request id=0x0001, seq=158/40…
1879	20.652687	192.168.3.129	172.253.116.…	ICMP	106	Echo (ping) request id=0x0001, seq=159/40…
1882	22.654369	192.168.3.129	172.253.116.…	ICMP	106	Echo (ping) request id=0x0001, seq=160/40…
1890	24.647943	192.168.3.129	172.253.116.…	ICMP	106	Echo (ping) request id=0x0001, seq=161/41…
1903	26.651231	192.168.3.129	172.253.116.…	ICMP	106	Echo (ping) request id=0x0001, seq=162/41…
1907	28.653772	192.168.3.129	172.253.116.…	ICMP	106	Echo (ping) request id=0x0001, seq=163/41…
1928	30.658432	192.168.3.129	172.253.116.…	ICMP	106	Echo (ping) request id=0x0001, seq=164/41…
1949	32.082895	192.168.3.129	8.8.8.8	ICMP	74	Echo (ping) request id=0x0001, seq=165/42…
1952	32.649540	192.168.3.129	172.253.116.…	ICMP	106	Echo (ping) request id=0x0001, seq=166/42…
1957	33.102866	192.168.3.129	8.8.8.8	ICMP	74	Echo (ping) request id=0x0001, seq=167/42…
1961	33.684473	192.168.3.129	172.253.116.…	ICMP	106	Echo (ping) request id=0x0001, seq=168/43…
1964	33.688543	192.168.3.129	172.253.116.…	ICMP	106	Echo (ping) request id=0x0001, seq=169/43…
1966	33.691084	192.168.3.129	172.253.116.…	ICMP	106	Echo (ping) request id=0x0001, seq=170/43…
1968	33.708248	192.168.3.129	172.253.116.…	ICMP	106	Echo (ping) request id=0x0001, seq=171/43…
1970	33.722924	192.168.3.129	172.253.116.…	ICMP	106	Echo (ping) request id=0x0001, seq=172/44…
1972	33.734712	192.168.3.129	172.253.116.…	ICMP	106	Echo (ping) request id=0x0001, seq=173/44…
1976	34.121183	192.168.3.129	8.8.8.8	ICMP	74	Echo (ping) request id=0x0001, seq=174/44…
1981	35.138820	192.168.3.129	8.8.8.8	ICMP	74	Echo (ping) request id=0x0001, seq=175/44…
1988	35.654271	192.168.3.129	172.253.116.…	ICMP	106	Echo (ping) request id=0x0001, seq=176/45…
1991	36.169557	192.168.3.129	8.8.8.8	ICMP	74	Echo (ping) request id=0x0001, seq=177/45…
2011	37.188411	192.168.3.129	8.8.8.8	ICMP	74	Echo (ping) request id=0x0001, seq=178/45…
2013	37.644958	192.168.3.129	172.253.116.…	ICMP	106	Echo (ping) request id=0x0001, seq=179/45…
2014	38.192857	192.168.3.129	8.8.8.8	ICMP	74	Echo (ping) request id=0x0001, seq=180/46…
2019	39.210104	192.168.3.129	8.8.8.8	ICMP	74	Echo (ping) request id=0x0001, seq=181/46…
2050	39.649588	192.168.3.129	172.253.116.…	ICMP	106	Echo (ping) request id=0x0001, seq=182/46…
2055	40.227525	192.168.3.129	8.8.8.8	ICMP	74	Echo (ping) request id=0x0001, seq=183/46…

5) a) What is the usage of nslookup protocol?

Returns the network's domain name/ip address

b) What is the filter that can be used to capture packets related to nslookup command execution for www.ucd.ie?

dns.qry.name == "www.ucd.ie"



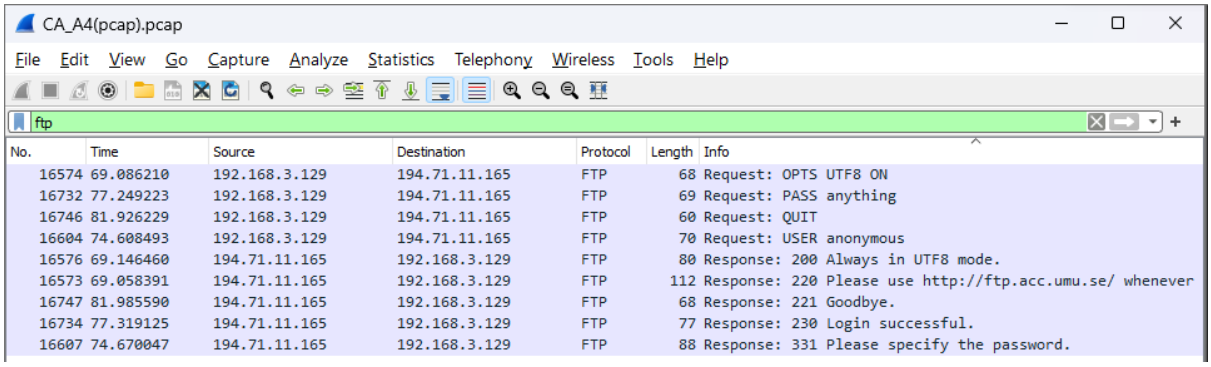
No.	Time	Source	Destination	Protocol	Length	Info
16447	50.591585	192.168.3.129	192.168.3.1	DNS	70	Standard query 0x0002 A www.ucd.ie
16449	50.612909	192.168.3.129	192.168.3.1	DNS	70	Standard query 0x0003 AAAA www.ucd.ie
16448	50.610515	192.168.3.1	192.168.3.129	DNS	102	Standard query response 0x0002 A www.ucd.ie A 54.229.218.
16450	50.630207	192.168.3.1	192.168.3.129	DNS	154	Standard query response 0x0003 AAAA www.ucd.ie SOA ns-312

6) a) What is the functionality of ftp command?

File transfer between server and client

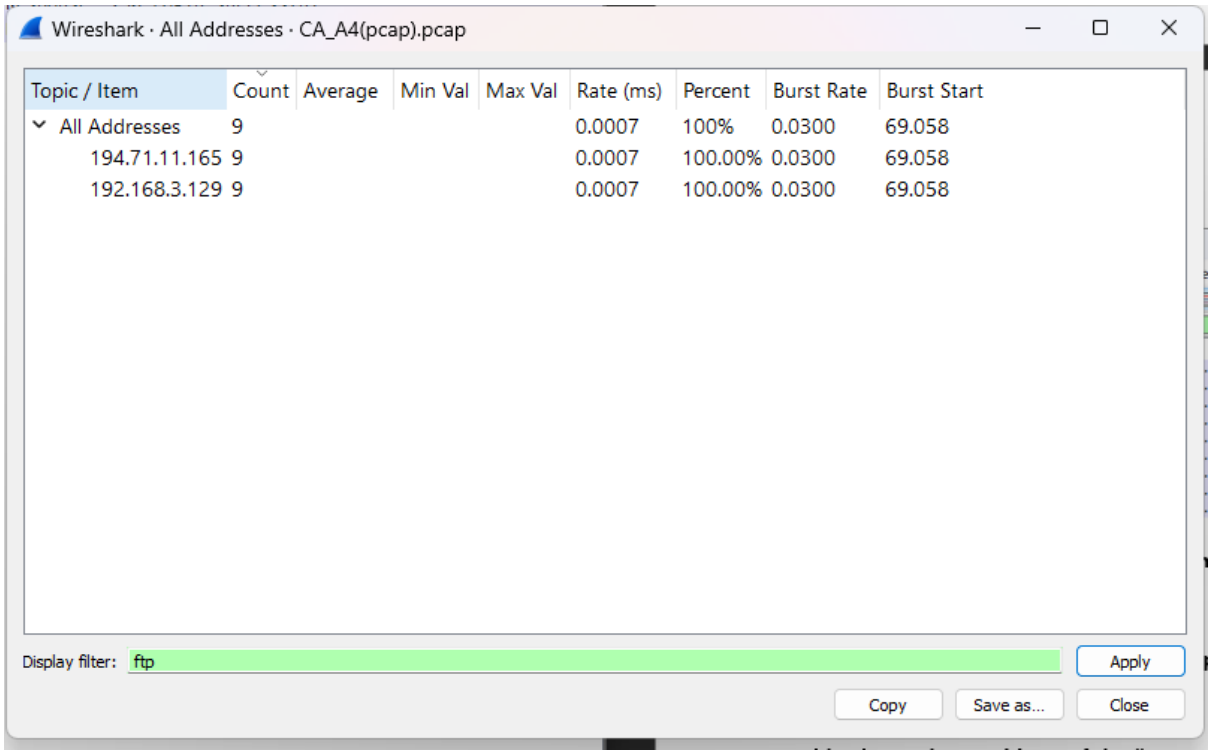
b) How to filter ftp packets in wireshark?

ftp



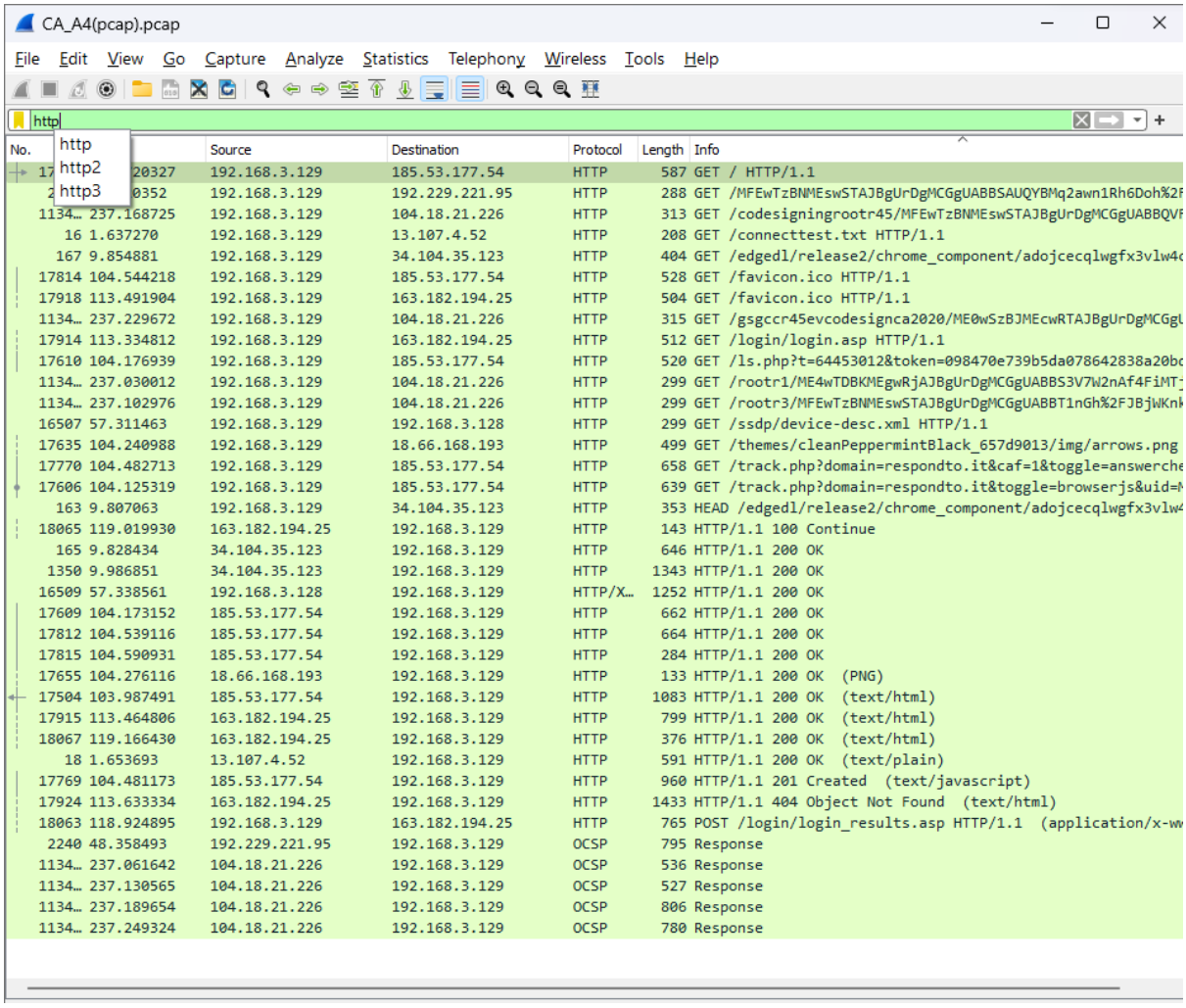
c) How many packets can you see when you apply this filter?

9



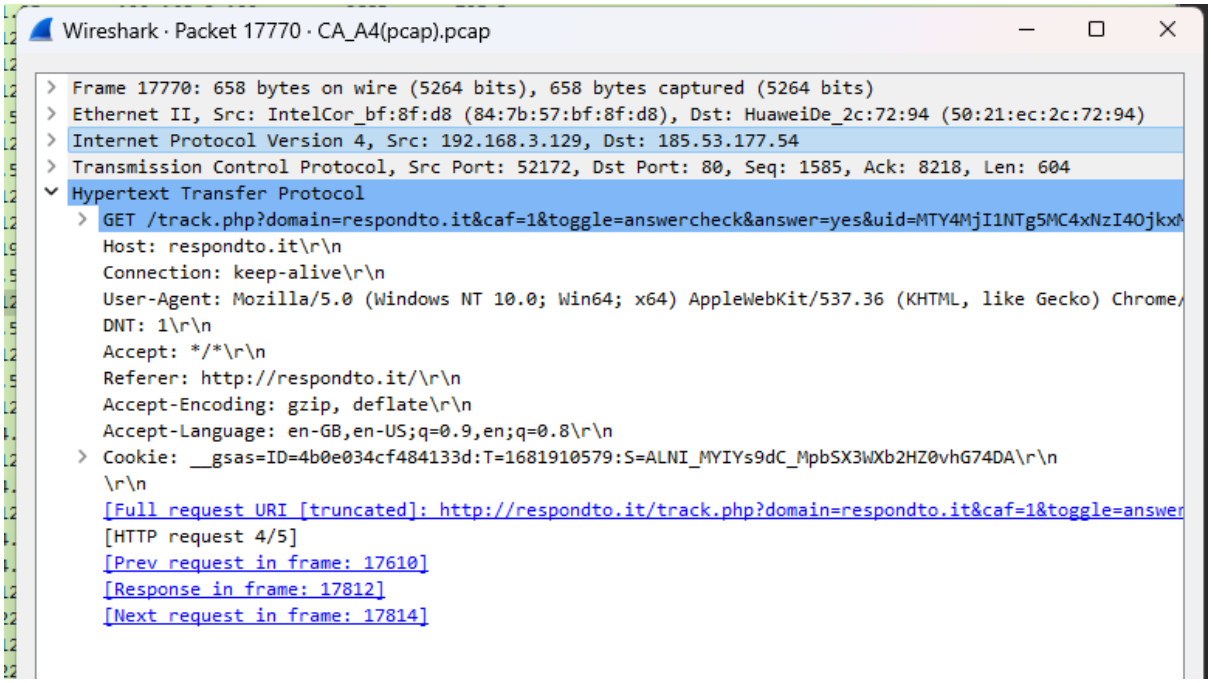
7) a) How to filter frames related to http sessions?

Filter with ‘http’ command



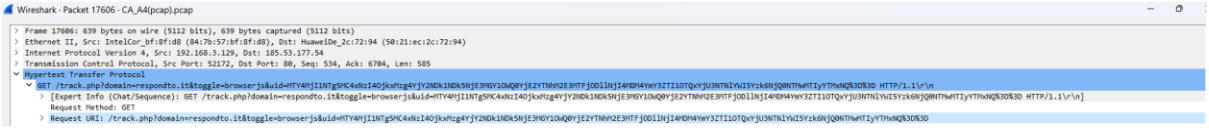
b) What is the IP address of the "respondto.it" url?

185.53.177.54



c) In which header can you find the url of the “respondto.it”? (hint: filter packets related to this url query, select a packet, and explore it)

You can find it in HTTP > GET > Request URI



d) What are the http methods that you can see when you apply the filter in part b?

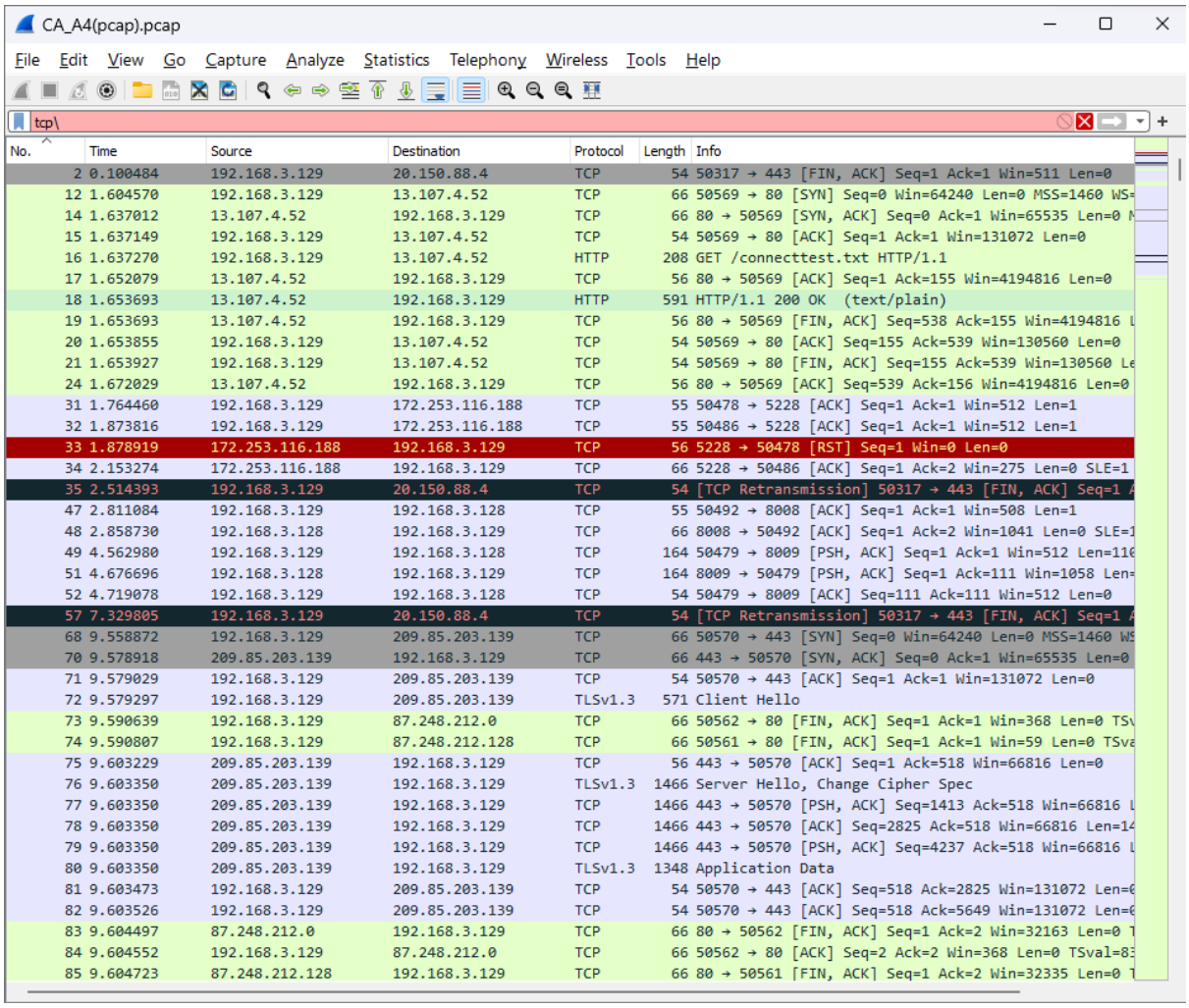
get and post

e) What is the protocol used to find the IP address corresponding to a domain name?

DNS domain name service

8) a) Can you see any TCP frames in your pcap file? (hint: apply filter for TCP packets)

Yes



No.	Time	Source	Destination	Protocol	Length	Info
2	0.100484	192.168.3.129	20.150.88.4	TCP	54	50317 → 443 [FIN, ACK] Seq=1 Ack=1 Win=511 Len=0
12	1.604570	192.168.3.129	13.107.4.52	TCP	66	50569 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=
14	1.637012	13.107.4.52	192.168.3.129	TCP	66	80 → 50569 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 M
15	1.637149	192.168.3.129	13.107.4.52	TCP	54	50569 → 80 [ACK] Seq=1 Ack=1 Win=131072 Len=0
16	1.637270	192.168.3.129	13.107.4.52	HTTP	208	GET /connecttest.txt HTTP/1.1
17	1.652079	13.107.4.52	192.168.3.129	TCP	56	80 → 50569 [ACK] Seq=1 Ack=155 Win=4194816 Len=0
18	1.653693	13.107.4.52	192.168.3.129	HTTP	591	HTTP/1.1 200 OK (text/plain)
19	1.653693	13.107.4.52	192.168.3.129	TCP	56	80 → 50569 [FIN, ACK] Seq=538 Ack=155 Win=4194816 L
20	1.653855	192.168.3.129	13.107.4.52	TCP	54	50569 → 80 [ACK] Seq=155 Ack=539 Win=130560 Len=0
21	1.653927	192.168.3.129	13.107.4.52	TCP	54	50569 → 80 [FIN, ACK] Seq=155 Ack=539 Win=130560 Le
24	1.672029	13.107.4.52	192.168.3.129	TCP	56	80 → 50569 [ACK] Seq=539 Ack=156 Win=4194816 Len=0
31	1.764460	192.168.3.129	172.253.116.188	TCP	55	50478 → 5228 [ACK] Seq=1 Ack=1 Win=512 Len=1
32	1.873816	192.168.3.129	172.253.116.188	TCP	55	50486 → 5228 [ACK] Seq=1 Ack=1 Win=512 Len=1
33	1.878919	172.253.116.188	192.168.3.129	TCP	56	5228 → 50478 [RST] Seq=1 Win=0 Len=0
34	2.153274	172.253.116.188	192.168.3.129	TCP	66	5228 → 50486 [ACK] Seq=1 Ack=2 Win=275 Len=0 SLE=1
35	2.514393	192.168.3.129	20.150.88.4	TCP	54	[TCP Retransmission] 50317 → 443 [FIN, ACK] Seq=1 A
47	2.811084	192.168.3.129	192.168.3.128	TCP	55	50492 → 8008 [ACK] Seq=1 Ack=1 Win=508 Len=1
48	2.858730	192.168.3.128	192.168.3.129	TCP	66	8008 → 50492 [ACK] Seq=1 Ack=2 Win=1041 Len=0 SLE=1
49	4.562980	192.168.3.129	192.168.3.128	TCP	164	50479 → 8009 [PSH, ACK] Seq=1 Ack=1 Win=512 Len=110
51	4.676696	192.168.3.128	192.168.3.129	TCP	164	8009 → 50479 [PSH, ACK] Seq=1 Ack=111 Win=1058 Len=
52	4.719078	192.168.3.129	192.168.3.128	TCP	54	50479 → 8009 [ACK] Seq=111 Ack=111 Win=512 Len=0
57	7.329805	192.168.3.129	20.150.88.4	TCP	54	[TCP Retransmission] 50317 → 443 [FIN, ACK] Seq=1 A
68	9.558872	192.168.3.129	209.85.203.139	TCP	66	50570 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS
70	9.578918	209.85.203.139	192.168.3.129	TCP	66	443 → 50570 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0
71	9.579029	192.168.3.129	209.85.203.139	TCP	54	50570 → 443 [ACK] Seq=1 Ack=1 Win=131072 Len=0
72	9.579297	192.168.3.129	209.85.203.139	TLSv1.3	571	Client Hello
73	9.590639	192.168.3.129	87.248.212.0	TCP	66	50562 → 80 [FIN, ACK] Seq=1 Ack=1 Win=368 Len=0 TS\
74	9.590807	192.168.3.129	87.248.212.128	TCP	66	50561 → 80 [FIN, ACK] Seq=1 Ack=1 Win=59 Len=0 TSva
75	9.603229	209.85.203.139	192.168.3.129	TCP	56	443 → 50570 [ACK] Seq=1 Ack=518 Win=66816 Len=0
76	9.603350	209.85.203.139	192.168.3.129	TLSv1.3	1466	Server Hello, Change Cipher Spec
77	9.603350	209.85.203.139	192.168.3.129	TCP	1466	443 → 50570 [PSH, ACK] Seq=1413 Ack=518 Win=66816 L
78	9.603350	209.85.203.139	192.168.3.129	TCP	1466	443 → 50570 [ACK] Seq=2825 Ack=518 Win=66816 Len=14
79	9.603350	209.85.203.139	192.168.3.129	TCP	1466	443 → 50570 [PSH, ACK] Seq=4237 Ack=518 Win=66816 L
80	9.603350	209.85.203.139	192.168.3.129	TLSv1.3	1348	Application Data
81	9.603473	192.168.3.129	209.85.203.139	TCP	54	50570 → 443 [ACK] Seq=518 Ack=2825 Win=131072 Len=0
82	9.603526	192.168.3.129	209.85.203.139	TCP	54	50570 → 443 [ACK] Seq=518 Ack=5649 Win=131072 Len=0
83	9.604497	87.248.212.0	192.168.3.129	TCP	66	80 → 50562 [FIN, ACK] Seq=1 Ack=2 Win=32163 Len=0 T
84	9.604552	192.168.3.129	87.248.212.0	TCP	66	50562 → 80 [ACK] Seq=2 Ack=2 Win=368 Len=0 TSval=83
85	9.604723	87.248.212.128	192.168.3.129	TCP	66	80 → 50561 [FIN, ACK] Seq=1 Ack=2 Win=32335 Len=0 T

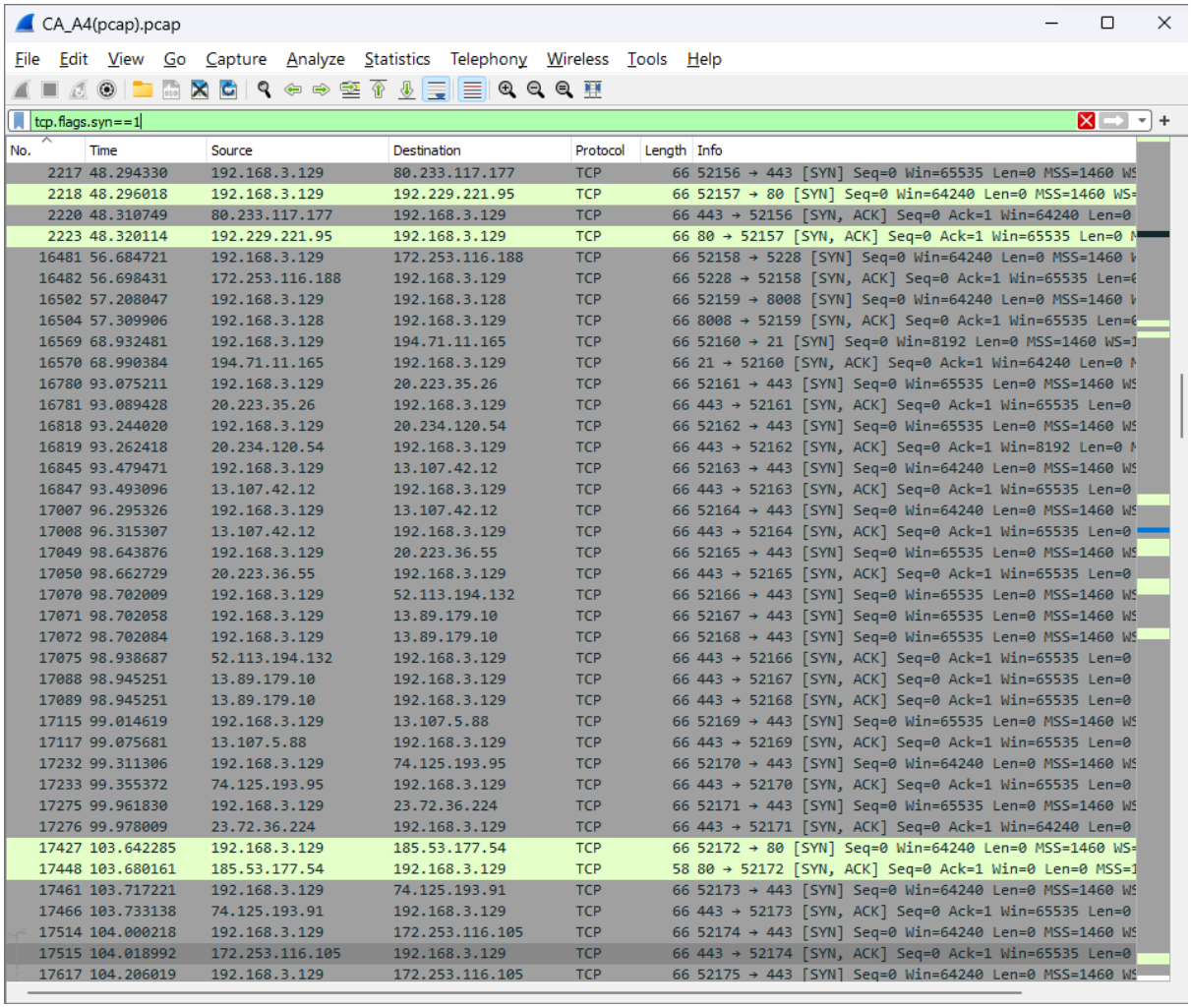
b) What is a three-way handshake in TCP?

The three way handshake includes the client sending a SYN request packet to establish a connection. The receiver sends back a SYN ACK packet confirming that it got the original SYN request and agrees

to establish connection. The client then replies with a final ACK, to show it received the receiver's SYN ACK message and the handshake is complete

c) What is the filter required to filter only TCP SYN frames?

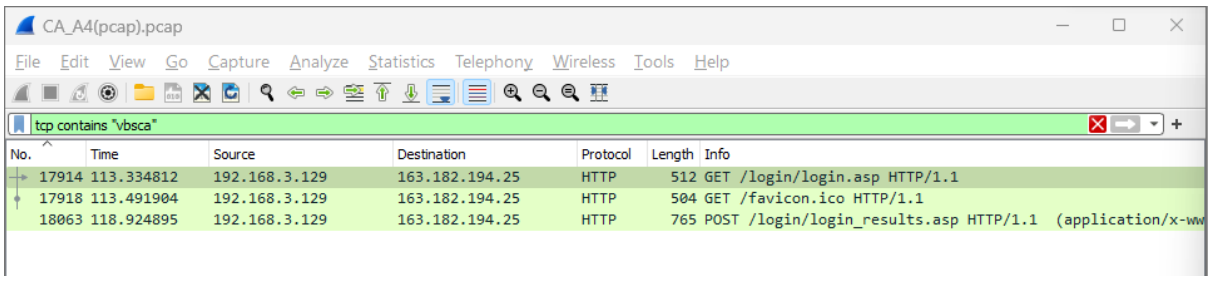
tcp.flags.syn==1



No.	Time	Source	Destination	Protocol	Length	Info
2217	48.294330	192.168.3.129	80.233.117.177	TCP	66	52156 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS
2218	48.296018	192.168.3.129	192.229.221.95	TCP	66	52157 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=
2220	48.310749	80.233.117.177	192.168.3.129	TCP	66	443 → 52156 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0
2223	48.320114	192.229.221.95	192.168.3.129	TCP	66	80 → 52157 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 M
16481	56.684721	192.168.3.129	172.253.116.188	TCP	66	52158 → 5228 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 W
16482	56.698431	172.253.116.188	192.168.3.129	TCP	66	5228 → 52158 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=6
16502	57.208047	192.168.3.129	192.168.3.128	TCP	66	52159 → 8008 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 W
16504	57.309906	192.168.3.128	192.168.3.129	TCP	66	8008 → 52159 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=6
16569	68.932481	192.168.3.129	194.71.11.165	TCP	66	52160 → 21 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=1
16570	68.990384	194.71.11.165	192.168.3.129	TCP	66	21 → 52160 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 M
16780	93.075211	192.168.3.129	20.223.35.26	TCP	66	52161 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS
16781	93.089428	20.223.35.26	192.168.3.129	TCP	66	443 → 52161 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0
16818	93.244020	192.168.3.129	20.234.120.54	TCP	66	52162 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS
16819	93.262418	20.234.120.54	192.168.3.129	TCP	66	443 → 52162 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 M
16845	93.479471	192.168.3.129	13.107.42.12	TCP	66	52163 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS
16847	93.493096	13.107.42.12	192.168.3.129	TCP	66	443 → 52163 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0
17007	96.295326	192.168.3.129	13.107.42.12	TCP	66	52164 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS
17008	96.315307	13.107.42.12	192.168.3.129	TCP	66	443 → 52164 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0
17049	98.643876	192.168.3.129	20.223.36.55	TCP	66	52165 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS
17050	98.662729	20.223.36.55	192.168.3.129	TCP	66	443 → 52165 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0
17070	98.702009	192.168.3.129	52.113.194.132	TCP	66	52166 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS
17071	98.702058	192.168.3.129	13.89.179.10	TCP	66	52167 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS
17072	98.702084	192.168.3.129	13.89.179.10	TCP	66	52168 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS
17075	98.938687	52.113.194.132	192.168.3.129	TCP	66	443 → 52166 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0
17088	98.945251	13.89.179.10	192.168.3.129	TCP	66	443 → 52167 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0
17089	98.945251	13.89.179.10	192.168.3.129	TCP	66	443 → 52168 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0
17115	99.014619	192.168.3.129	13.107.5.88	TCP	66	52169 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS
17117	99.075681	13.107.5.88	192.168.3.129	TCP	66	443 → 52169 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0
17232	99.311306	192.168.3.129	74.125.193.95	TCP	66	52170 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS
17233	99.355372	74.125.193.95	192.168.3.129	TCP	66	443 → 52170 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0
17275	99.961830	192.168.3.129	23.72.36.224	TCP	66	52171 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS
17276	99.978009	23.72.36.224	192.168.3.129	TCP	66	443 → 52171 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0
17427	103.642285	192.168.3.129	185.53.177.54	TCP	66	52172 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=
17448	103.680161	185.53.177.54	192.168.3.129	TCP	58	80 → 52172 [SYN, ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1
17461	103.717221	192.168.3.129	74.125.193.91	TCP	66	52173 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS
17466	103.733138	74.125.193.91	192.168.3.129	TCP	66	443 → 52173 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0
17514	104.000218	192.168.3.129	172.253.116.105	TCP	66	52174 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS
17515	104.018992	172.253.116.105	192.168.3.129	TCP	66	443 → 52174 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0
17617	104.206019	192.168.3.129	172.253.116.105	TCP	66	52175 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS

d) What is the filter for identifying tcp packets that contain "vbsca"?

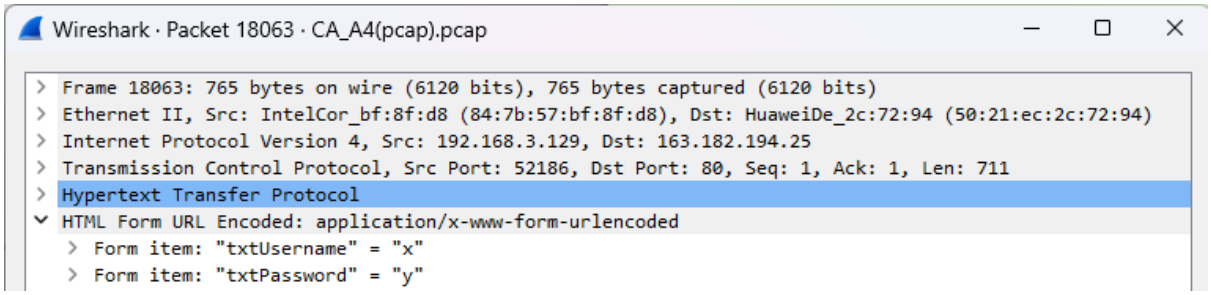
tcp contains "vbsca"



No.	Time	Source	Destination	Protocol	Length	Info
17914	113.334812	192.168.3.129	163.182.194.25	HTTP	512	GET /login/login.asp HTTP/1.1
17918	113.491904	192.168.3.129	163.182.194.25	HTTP	504	GET /favicon.ico HTTP/1.1
18063	118.924895	192.168.3.129	163.182.194.25	HTTP	765	POST /login/login_results.asp HTTP/1.1 (application/x-ww

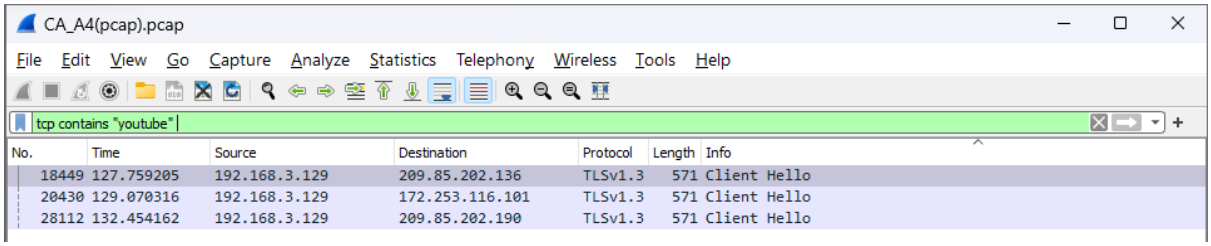
e) Where can you see your submitted username and password in a packet related to this filter?

In the HTML Form URL section as form items. I entered x and y for my username and password below



9) a) How to identify TCP packets which contains "youtube" in wireshark?

tcp contains "youtube"



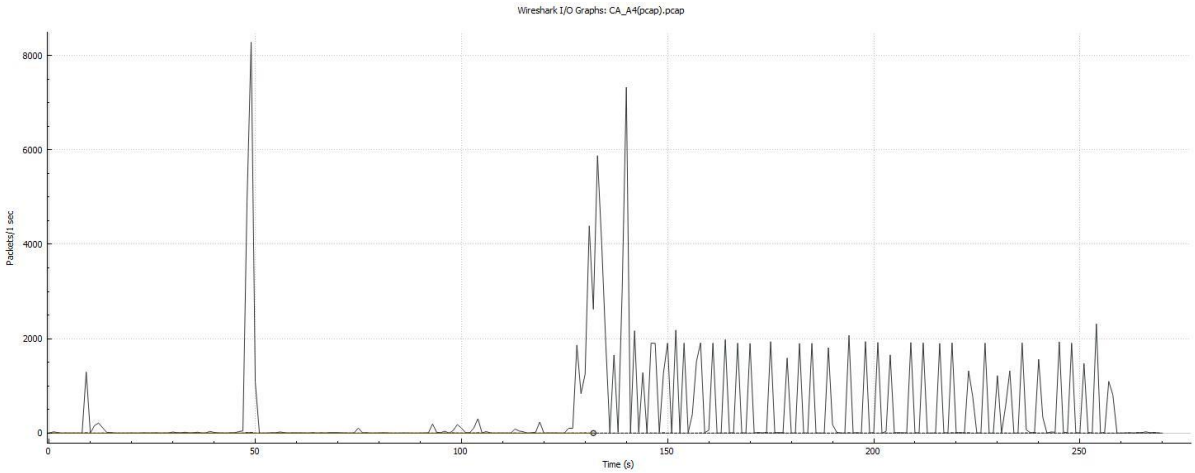
b) What are the different protocols that you can see when you apply this filter?

TLS (Transport Layer Security) as well as Ethernet II, IP, TCP

c) Explain the difference between one-way SSL (Secure Socket Layer) and two-way SSL.

One way SSL verifies the identity of the server for the client but does not authenticate the clients identity for the server. In two-way SSL both the client and server's identities are authenticated for each other.

10) Draw the IO graph related to this PCAP file.



PCAP Drive Link:

https://drive.google.com/file/d/1YbQ35VAfoPXQzoE9NSp-Af5DobkdDpzA/view?usp=sharing