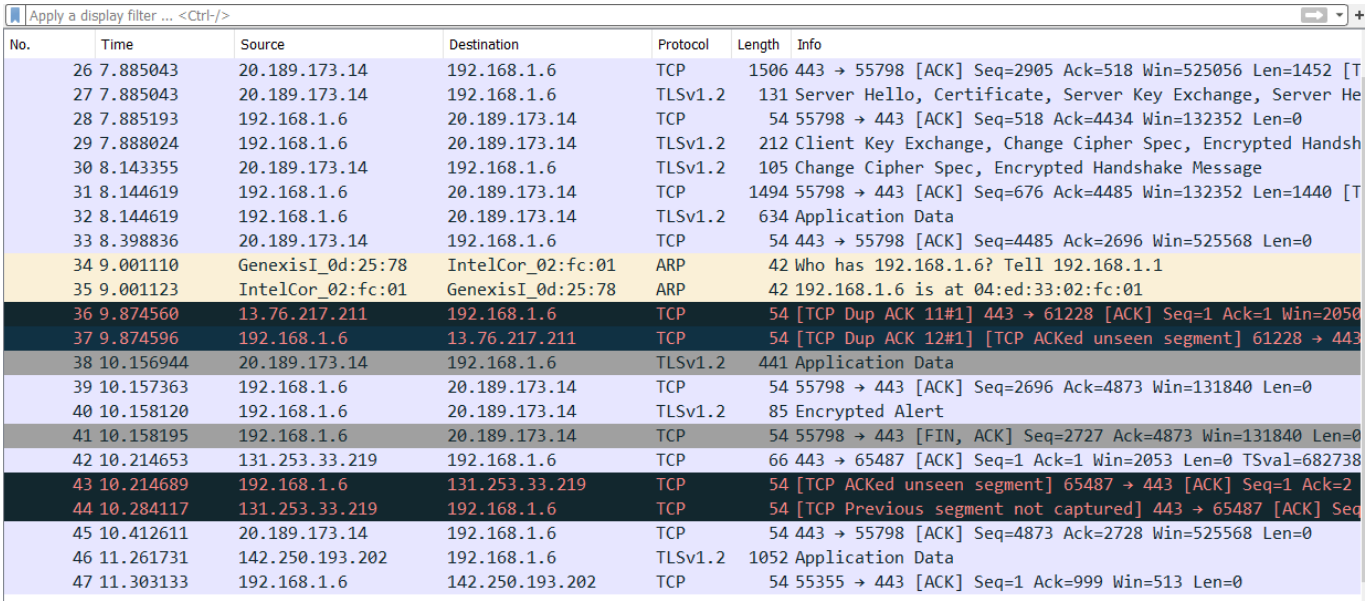
Nirbhay Sharma (B19CSE114)

Computer Networks - Lab - 4

---

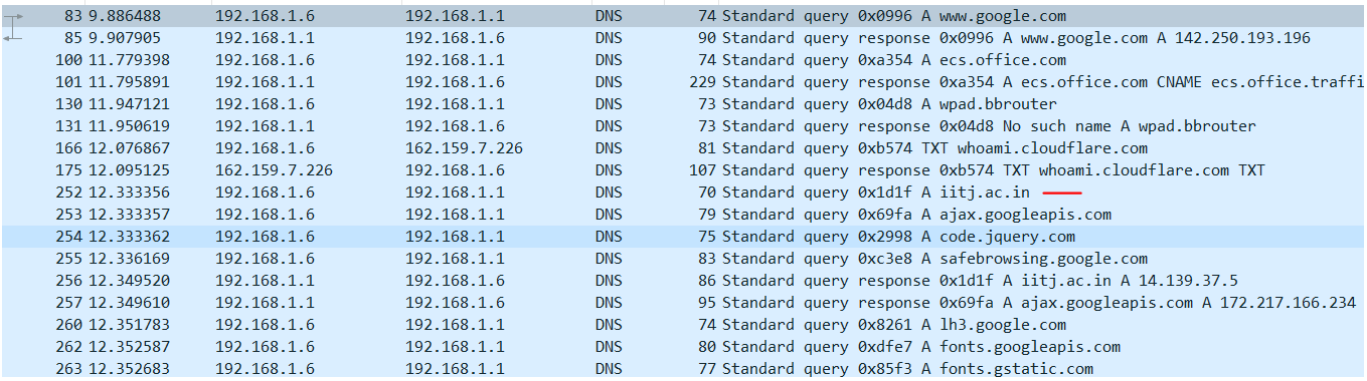
1. after starting packet capture on wireless connection (wifi) it starts to capture various packets related to wifi and it can even capture packets from where the wifi is getting the network screenshot is attached below



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

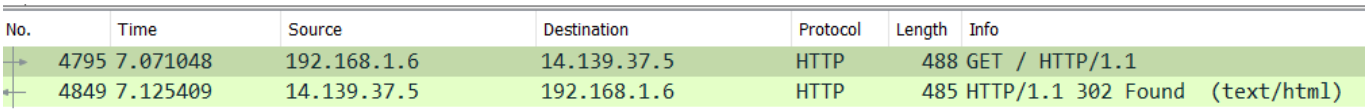
No.	Time	Source	Destination	Protocol	Length	Info
26	7.885043	20.189.173.14	192.168.1.6	TCP	1506	443 → 55798 [ACK] Seq=2905 Ack=518 Win=525056 Len=1452 [T
27	7.885043	20.189.173.14	192.168.1.6	TLSv1.2	131	Server Hello, Certificate, Server Key Exchange, Server He
28	7.885193	192.168.1.6	20.189.173.14	TCP	54	55798 → 443 [ACK] Seq=518 Ack=4434 Win=132352 Len=0
29	7.888024	192.168.1.6	20.189.173.14	TLSv1.2	212	Client Key Exchange, Change Cipher Spec, Encrypted Handsh
30	8.143355	20.189.173.14	192.168.1.6	TLSv1.2	105	Change Cipher Spec, Encrypted Handshake Message
31	8.144619	192.168.1.6	20.189.173.14	TCP	1494	55798 → 443 [ACK] Seq=676 Ack=4485 Win=132352 Len=1440 [T
32	8.144619	192.168.1.6	20.189.173.14	TLSv1.2	634	Application Data
33	8.398836	20.189.173.14	192.168.1.6	TCP	54	443 → 55798 [ACK] Seq=4485 Ack=2696 Win=525568 Len=0
34	9.001110	GenexisI_0d:25:78	IntelCor_02:fc:01	ARP	42	Who has 192.168.1.6? Tell 192.168.1.1
35	9.001123	IntelCor_02:fc:01	GenexisI_0d:25:78	ARP	42	192.168.1.6 is at 04:ed:33:02:fc:01
36	9.874560	13.76.217.211	192.168.1.6	TCP	54	[TCP Dup ACK 11#1] 443 → 61228 [ACK] Seq=1 Ack=1 Win=2050
37	9.874596	192.168.1.6	13.76.217.211	TCP	54	[TCP Dup ACK 12#1] [TCP ACKed unseen segment] 61228 → 443
38	10.156944	20.189.173.14	192.168.1.6	TLSv1.2	441	Application Data
39	10.157363	192.168.1.6	20.189.173.14	TCP	54	55798 → 443 [ACK] Seq=2696 Ack=4873 Win=131840 Len=0
40	10.158120	192.168.1.6	20.189.173.14	TLSv1.2	85	Encrypted Alert
41	10.158195	192.168.1.6	20.189.173.14	TCP	54	55798 → 443 [FIN, ACK] Seq=2727 Ack=4873 Win=131840 Len=0
42	10.214653	131.253.33.219	192.168.1.6	TCP	66	443 → 65487 [ACK] Seq=1 Ack=1 Win=2053 Len=0 TSval=682738
43	10.214689	192.168.1.6	131.253.33.219	TCP	54	[TCP ACKed unseen segment] 65487 → 443 [ACK] Seq=1 Ack=2
44	10.284117	131.253.33.219	192.168.1.6	TCP	54	[TCP Previous segment not captured] 443 → 65487 [ACK] Seq
45	10.412611	20.189.173.14	192.168.1.6	TCP	54	443 → 55798 [ACK] Seq=4873 Ack=2728 Win=525568 Len=0
46	11.261731	142.250.193.202	192.168.1.6	TLSv1.2	1052	Application Data
47	11.303133	192.168.1.6	142.250.193.202	TCP	54	55355 → 443 [ACK] Seq=1 Ack=999 Win=513 Len=0

2. yes, able to see the dns request, screenshot attached below



83	9.886488	192.168.1.6	192.168.1.1	DNS	74	Standard query 0x0996 A www.google.com
85	9.907905	192.168.1.1	192.168.1.6	DNS	90	Standard query response 0x0996 A www.google.com A 142.250.193.196
100	11.779398	192.168.1.6	192.168.1.1	DNS	74	Standard query 0xa354 A ecs.office.com
101	11.795891	192.168.1.1	192.168.1.6	DNS	229	Standard query response 0xa354 A ecs.office.com CNAME ecs.office.traffi
130	11.947121	192.168.1.6	192.168.1.1	DNS	73	Standard query 0x04d8 A wpad.bbrouter
131	11.950619	192.168.1.1	192.168.1.6	DNS	73	Standard query response 0x04d8 No such name A wpad.bbrouter
166	12.076867	192.168.1.6	162.159.7.226	DNS	81	Standard query 0xb574 TXT whoami.cloudflare.com
175	12.095125	162.159.7.226	192.168.1.6	DNS	107	Standard query response 0xb574 TXT whoami.cloudflare.com TXT
252	12.333356	192.168.1.6	192.168.1.1	DNS	70	Standard query 0x1d1f A iitj.ac.in
253	12.333357	192.168.1.6	192.168.1.1	DNS	79	Standard query 0x69fa A ajax.googleapis.com
254	12.333362	192.168.1.6	192.168.1.1	DNS	75	Standard query 0x2998 A code.jquery.com
255	12.336169	192.168.1.6	192.168.1.1	DNS	83	Standard query 0xc3e8 A safebrowsing.google.com
256	12.349520	192.168.1.1	192.168.1.6	DNS	86	Standard query response 0x1d1f A iitj.ac.in A 14.139.37.5
257	12.349610	192.168.1.1	192.168.1.6	DNS	95	Standard query response 0x69fa A ajax.googleapis.com A 172.217.166.234
260	12.351783	192.168.1.6	192.168.1.1	DNS	74	Standard query 0x8261 A lh3.google.com
262	12.352587	192.168.1.6	192.168.1.1	DNS	80	Standard query 0xdfe7 A fonts.googleapis.com
263	12.352683	192.168.1.6	192.168.1.1	DNS	77	Standard query 0x85f3 A fonts.gstatic.com

- http and tcp requests are as follows



No.	Time	Source	Destination	Protocol	Length	Info
4795	7.071048	192.168.1.6	14.139.37.5	HTTP	488	GET / HTTP/1.1
4849	7.125409	14.139.37.5	192.168.1.6	HTTP	485	HTTP/1.1 302 Found (text/html)

- from above image we can see that we have an http GET request and by doing that we get the web page on the browser

tcp and ip.dst==14.139.37.5

No.	Time	Source	Destination	Protocol	Length	Info
7012	5.576752	192.168.1.6	14.139.37.5	TCP	66	61505 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
7013	5.583520	192.168.1.6	14.139.37.5	TCP	66	61506 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
7027	5.632250	192.168.1.6	14.139.37.5	TCP	54	61505 → 80 [ACK] Seq=1 Ack=1 Win=132096 Len=0
7028	5.632863	192.168.1.6	14.139.37.5	HTTP	488	GET / HTTP/1.1
7051	5.644264	192.168.1.6	14.139.37.5	TCP	54	61506 → 80 [ACK] Seq=1 Ack=1 Win=132096 Len=0
7104	5.731901	192.168.1.6	14.139.37.5	TCP	66	61507 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
7110	5.742570	192.168.1.6	14.139.37.5	TCP	54	61505 → 80 [ACK] Seq=435 Ack=432 Win=131584 Len=0
7131	5.788182	192.168.1.6	14.139.37.5	TCP	54	61507 → 443 [ACK] Seq=1 Ack=1 Win=132096 Len=0
7133	5.818601	192.168.1.6	14.139.37.5	TLSv1.2	571	Client Hello
7154	5.871337	192.168.1.6	14.139.37.5	TCP	54	61507 → 443 [ACK] Seq=518 Ack=2021 Win=132096 Len=0
7173	5.878354	192.168.1.6	14.139.37.5	TLSv1.2	180	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
7190	5.937041	192.168.1.6	14.139.37.5	TLSv1.2	746	Application Data
7219	5.997514	192.168.1.6	14.139.37.5	TCP	66	61507 → 443 [ACK] Seq=1336 Ack=11326 Win=132096 Len=0 SLE=14222 SRE=18566
7227	6.041755	192.168.1.6	14.139.37.5	TCP	66	61508 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
7228	6.042128	192.168.1.6	14.139.37.5	TCP	66	61509 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
7229	6.042686	192.168.1.6	14.139.37.5	TCP	66	61510 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
7230	6.042926	192.168.1.6	14.139.37.5	TCP	66	61511 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
7231	6.043451	192.168.1.6	14.139.37.5	TCP	66	61512 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
7244	6.047588	192.168.1.6	14.139.37.5	TCP	54	61507 → 443 [ACK] Seq=1336 Ack=34538 Win=132096 Len=0
7274	6.092727	192.168.1.6	14.139.37.5	TCP	54	61508 → 443 [ACK] Seq=1 Ack=1 Win=132096 Len=0
7275	6.092743	192.168.1.6	14.139.37.5	TCP	54	61512 → 443 [ACK] Seq=1 Ack=1 Win=132096 Len=0
7276	6.092756	192.168.1.6	14.139.37.5	TCP	54	61509 → 443 [ACK] Seq=1 Ack=1 Win=132096 Len=0

- ip for iitj.ac.in = 14.139.37.5

3. packet highlighted with black shows that the packet is lost or 3 dup acks are received for a packet which basically means that the packet needs to be retransmitted. screen shots attached below

30	7.125296	192.168.1.6	20.198.162.78	TCP	155	[TCP Retransmission] 55481 → 443 [PSH, ACK] Seq=1 Ack=1 Win=516 Len=101
31	7.203737	192.168.1.6	13.107.4.52	TCP	54	[TCP Retransmission] 65063 → 80 [FIN, ACK] Seq=155 Ack=540 Win=131840 Len=0
32	7.206693	13.107.4.52	192.168.1.6	TCP	54	80 → 65063 [FIN, ACK] Seq=540 Ack=155 Win=525312 Len=0
33	7.206693	20.198.162.78	192.168.1.6	TLSv1.2	225	Application Data
34	7.206693	13.107.4.52	192.168.1.6	TCP	54	80 → 65063 [ACK] Seq=541 Ack=156 Win=525312 Len=0
35	7.206693	20.198.162.78	192.168.1.6	TCP	225	[TCP Retransmission] 443 → 55481 [PSH, ACK] Seq=1 Ack=102 Win=6754 Len=171
36	7.206839	192.168.1.6	20.198.162.78	TCP	66	55481 → 443 [ACK] Seq=102 Ack=172 Win=515 Len=0 SLE=1 SRE=172
37	7.207336	192.168.1.6	13.107.4.52	TCP	54	65063 → 80 [ACK] Seq=156 Ack=541 Win=131840 Len=0
38	7.212378	20.198.162.78	192.168.1.6	TCP	66	[TCP Dup ACK 33#1] 443 → 55481 [ACK] Seq=172 Ack=102 Win=6754 Len=0 SLE=1 SRE=102
39	7.220728	13.107.4.52	192.168.1.6	TCP	54	[TCP Dup ACK 34#1] 80 → 65063 [ACK] Seq=541 Ack=156 Win=525312 Len=0
61	8.250799	52.163.231.110	192.168.1.6	TCP	66	443 → 54238 [ACK] Seq=1 Ack=1 Win=2047 Len=0 TSval=7788907 TSecr=40023389
62	8.250867	192.168.1.6	52.163.231.110	TCP	54	[TCP ACKed unseen segment] 54238 → 443 [ACK] Seq=1 Ack=2 Win=515 Len=0
63	8.322831	52.163.231.110	192.168.1.6	TCP	54	[TCP Previous segment not captured] 443 → 54238 [ACK] Seq=2 Ack=1 Win=2047 Len=0

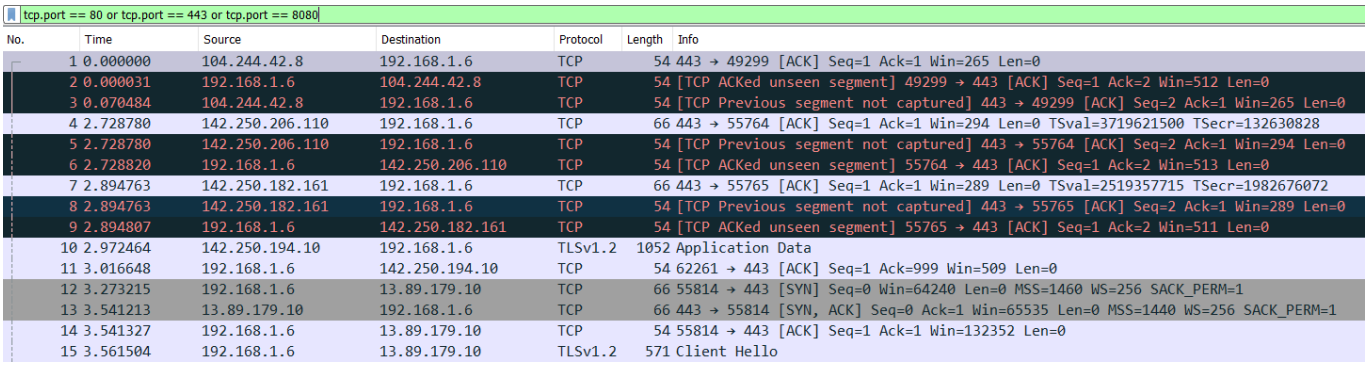
4. various filters and their uses are shown below (start capturing packets then open some websites like (iitj.ac.in, github.com, leetcode.com))

- ip.dst==13.234.176.102 (querying a particular ip)

ip.dst==13.234.176.102

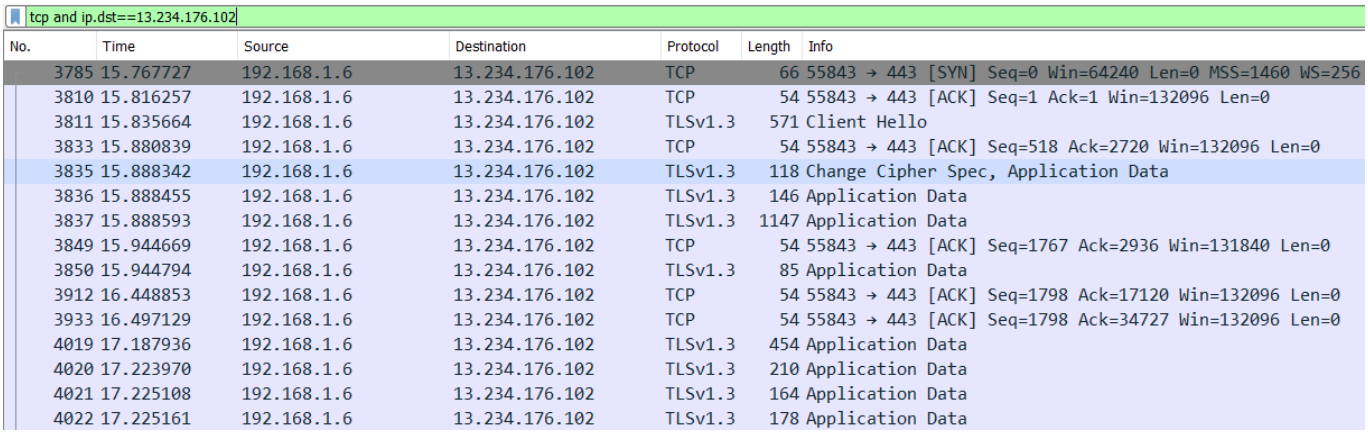
No.	Time	Source	Destination	Protocol	Length	Info
3785	15.767727	192.168.1.6	13.234.176.102	TCP	66	55843 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256
3810	15.816257	192.168.1.6	13.234.176.102	TCP	54	55843 → 443 [ACK] Seq=1 Ack=1 Win=132096 Len=0
3811	15.835664	192.168.1.6	13.234.176.102	TLSv1.3	571	Client Hello
3833	15.880839	192.168.1.6	13.234.176.102	TCP	54	55843 → 443 [ACK] Seq=518 Ack=2720 Win=132096 Len=0
3835	15.888342	192.168.1.6	13.234.176.102	TLSv1.3	118	Change Cipher Spec, Application Data
3836	15.888455	192.168.1.6	13.234.176.102	TLSv1.3	146	Application Data
3837	15.888593	192.168.1.6	13.234.176.102	TLSv1.3	1147	Application Data
3849	15.944669	192.168.1.6	13.234.176.102	TCP	54	55843 → 443 [ACK] Seq=1767 Ack=2936 Win=131840 Len=0
3850	15.944794	192.168.1.6	13.234.176.102	TLSv1.3	85	Application Data
3912	16.448853	192.168.1.6	13.234.176.102	TCP	54	55843 → 443 [ACK] Seq=1798 Ack=17120 Win=132096 Len=0
3933	16.497129	192.168.1.6	13.234.176.102	TCP	54	55843 → 443 [ACK] Seq=1798 Ack=34727 Win=132096 Len=0
4019	17.187936	192.168.1.6	13.234.176.102	TLSv1.3	454	Application Data
4020	17.223970	192.168.1.6	13.234.176.102	TLSv1.3	210	Application Data
4021	17.225108	192.168.1.6	13.234.176.102	TLSv1.3	164	Application Data
4022	17.225161	192.168.1.6	13.234.176.102	TLSv1.3	178	Application Data
4023	17.225339	192.168.1.6	13.234.176.102	TLSv1.3	157	Application Data
4082	17.518751	192.168.1.6	13.234.176.102	TCP	54	55843 → 443 [ACK] Seq=2691 Ack=37075 Win=132096 Len=0
4095	17.540522	192.168.1.6	13.234.176.102	TCP	54	55843 → 443 [ACK] Seq=2691 Ack=39493 Win=132096 Len=0
4096	17.541815	192.168.1.6	13.234.176.102	TLSv1.3	136	Application Data

- tcp.port == 80 or tcp.port == 443 or tcp.port == 8080 (or filter)


tcp.port == 80 or tcp.port == 443 or tcp.port == 8080

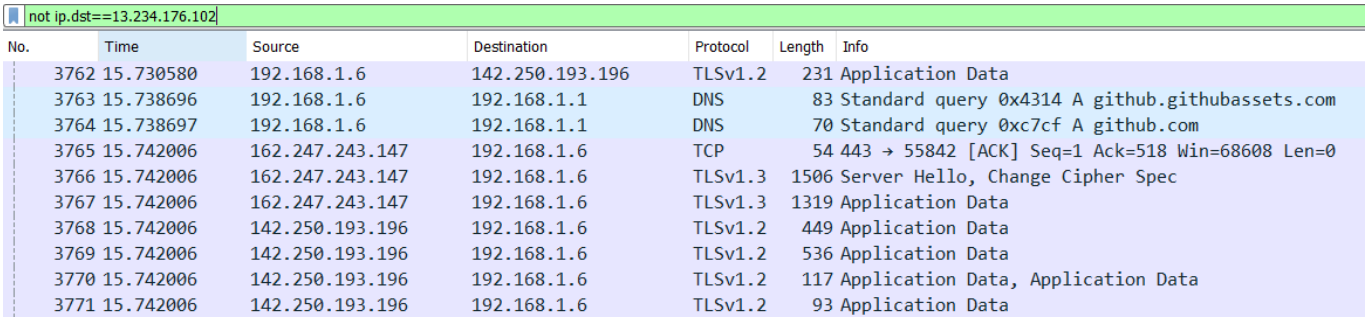
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	104.244.42.8	192.168.1.6	TCP	54	443 → 49299 [ACK] Seq=1 Ack=1 Win=265 Len=0
2	0.000031	192.168.1.6	104.244.42.8	TCP	54	[TCP ACKed unseen segment] 49299 → 443 [ACK] Seq=1 Ack=2 Win=512 Len=0
3	0.070484	104.244.42.8	192.168.1.6	TCP	54	[TCP Previous segment not captured] 443 → 49299 [ACK] Seq=2 Ack=1 Win=265 Len=0
4	2.728780	142.250.206.110	192.168.1.6	TCP	66	443 → 55764 [ACK] Seq=1 Ack=1 Win=294 Len=0 TSval=3719621500 TSecr=132630828
5	2.728780	142.250.206.110	192.168.1.6	TCP	54	[TCP Previous segment not captured] 443 → 55764 [ACK] Seq=2 Ack=1 Win=294 Len=0
6	2.728820	192.168.1.6	142.250.206.110	TCP	54	[TCP ACKed unseen segment] 55764 → 443 [ACK] Seq=1 Ack=2 Win=513 Len=0
7	2.894763	142.250.182.161	192.168.1.6	TCP	66	443 → 55765 [ACK] Seq=1 Ack=1 Win=289 Len=0 TSval=2519357715 TSecr=1982676072
8	2.894763	142.250.182.161	192.168.1.6	TCP	54	[TCP Previous segment not captured] 443 → 55765 [ACK] Seq=2 Ack=1 Win=289 Len=0
9	2.894807	192.168.1.6	142.250.182.161	TCP	54	[TCP ACKed unseen segment] 55765 → 443 [ACK] Seq=1 Ack=2 Win=511 Len=0
10	2.972464	142.250.194.10	192.168.1.6	TLSv1.2	1052	Application Data
11	3.016648	192.168.1.6	142.250.194.10	TCP	54	62261 → 443 [ACK] Seq=1 Ack=999 Win=509 Len=0
12	3.273215	192.168.1.6	13.89.179.10	TCP	66	55814 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
13	3.541213	13.89.179.10	192.168.1.6	TCP	66	443 → 55814 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1440 WS=256 SACK_PERM=1
14	3.541327	192.168.1.6	13.89.179.10	TCP	54	55814 → 443 [ACK] Seq=1 Ack=1 Win=132352 Len=0
15	3.561504	192.168.1.6	13.89.179.10	TLSv1.2	571	Client Hello

- tcp and ip.dst==13.234.176.102 (and filter)


tcp and ip.dst==13.234.176.102

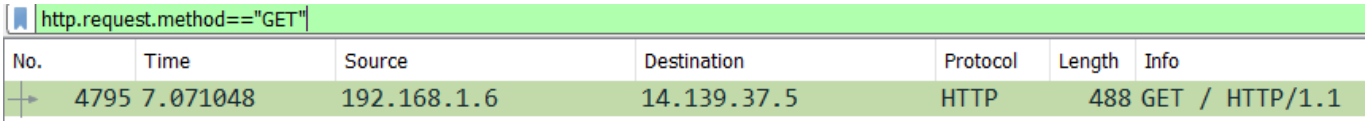
No.	Time	Source	Destination	Protocol	Length	Info
3785	15.767727	192.168.1.6	13.234.176.102	TCP	66	55843 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256
3810	15.816257	192.168.1.6	13.234.176.102	TCP	54	55843 → 443 [ACK] Seq=1 Ack=1 Win=132096 Len=0
3811	15.835664	192.168.1.6	13.234.176.102	TLSv1.3	571	Client Hello
3833	15.880839	192.168.1.6	13.234.176.102	TCP	54	55843 → 443 [ACK] Seq=518 Ack=2720 Win=132096 Len=0
3835	15.888342	192.168.1.6	13.234.176.102	TLSv1.3	118	Change Cipher Spec, Application Data
3836	15.888455	192.168.1.6	13.234.176.102	TLSv1.3	146	Application Data
3837	15.888593	192.168.1.6	13.234.176.102	TLSv1.3	1147	Application Data
3849	15.944669	192.168.1.6	13.234.176.102	TCP	54	55843 → 443 [ACK] Seq=1767 Ack=2936 Win=131840 Len=0
3850	15.944794	192.168.1.6	13.234.176.102	TLSv1.3	85	Application Data
3912	16.448853	192.168.1.6	13.234.176.102	TCP	54	55843 → 443 [ACK] Seq=1798 Ack=17120 Win=132096 Len=0
3933	16.497129	192.168.1.6	13.234.176.102	TCP	54	55843 → 443 [ACK] Seq=1798 Ack=34727 Win=132096 Len=0
4019	17.187936	192.168.1.6	13.234.176.102	TLSv1.3	454	Application Data
4020	17.223970	192.168.1.6	13.234.176.102	TLSv1.3	210	Application Data
4021	17.225108	192.168.1.6	13.234.176.102	TLSv1.3	164	Application Data
4022	17.225161	192.168.1.6	13.234.176.102	TLSv1.3	178	Application Data

- not ip.dst==13.234.176.102 (not filter)


not ip.dst==13.234.176.102

No.	Time	Source	Destination	Protocol	Length	Info
3762	15.730580	192.168.1.6	142.250.193.196	TLSv1.2	231	Application Data
3763	15.738696	192.168.1.6	192.168.1.1	DNS	83	Standard query 0x4314 A github.githubassets.com
3764	15.738697	192.168.1.6	192.168.1.1	DNS	70	Standard query 0xc7cf A github.com
3765	15.742006	162.247.243.147	192.168.1.6	TCP	54	443 → 55842 [ACK] Seq=1 Ack=518 Win=68608 Len=0
3766	15.742006	162.247.243.147	192.168.1.6	TLSv1.3	1506	Server Hello, Change Cipher Spec
3767	15.742006	162.247.243.147	192.168.1.6	TLSv1.3	1319	Application Data
3768	15.742006	142.250.193.196	192.168.1.6	TLSv1.2	449	Application Data
3769	15.742006	142.250.193.196	192.168.1.6	TLSv1.2	536	Application Data
3770	15.742006	142.250.193.196	192.168.1.6	TLSv1.2	117	Application Data, Application Data
3771	15.742006	142.250.193.196	192.168.1.6	TLSv1.2	93	Application Data

- http.request.method=="GET"

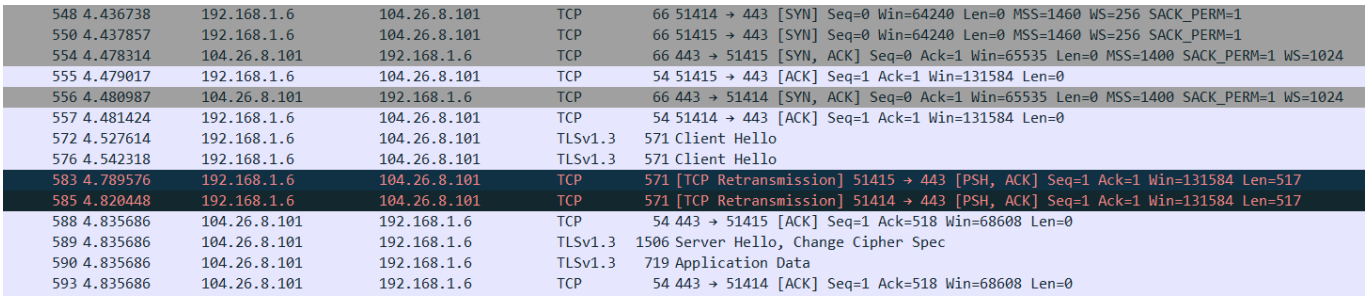

http.request.method=="GET"

No.	Time	Source	Destination	Protocol	Length	Info
4795	7.071048	192.168.1.6	14.139.37.5	HTTP	488	GET / HTTP/1.1

5. **ip.src == 192.168.1.6** (since we need to get all outgoing traffic then we can filter the src ip address of our computer to get what all packets are going out of our ip address)

6. UDP is a faster protocol than TCP (as it requires 3-way handshaking which is slow), in DNS the requests are very small with a small response which also fits in UDP packets and also DNS don't require to maintain connection as in case of TCP and hence DNS uses UDP but in case of HTTP, it requres a reliable data transfer and needs to maintain a longer connection with the server and hence TCP is suitable for it.

7. visit a website (say leetcode.com (ip = 104.26.8.101)) and observe the tcp connection in wireshark



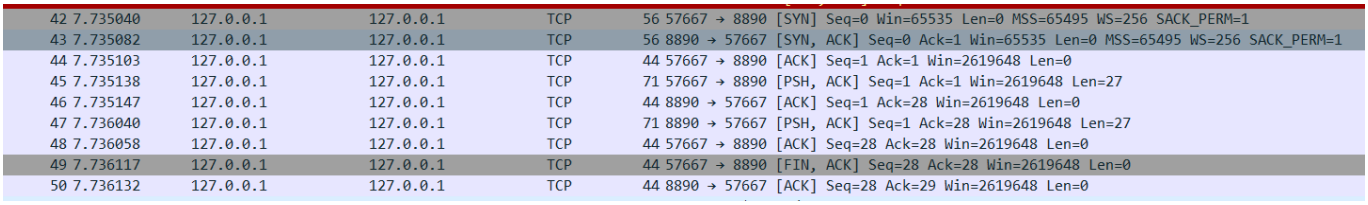
548	4.436738	192.168.1.6	104.26.8.101	TCP	66 51414 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
550	4.437857	192.168.1.6	104.26.8.101	TCP	66 51415 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
554	4.478314	104.26.8.101	192.168.1.6	TCP	66 443 → 51415 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1400 SACK_PERM=1 WS=1024
555	4.479017	192.168.1.6	104.26.8.101	TCP	54 51415 → 443 [ACK] Seq=1 Ack=1 Win=131584 Len=0
556	4.480987	104.26.8.101	192.168.1.6	TCP	66 443 → 51414 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1400 SACK_PERM=1 WS=1024
557	4.481424	192.168.1.6	104.26.8.101	TCP	54 51414 → 443 [ACK] Seq=1 Ack=1 Win=131584 Len=0
572	4.527614	192.168.1.6	104.26.8.101	TLSv1.3	571 Client Hello
576	4.542318	192.168.1.6	104.26.8.101	TLSv1.3	571 Client Hello
583	4.789576	192.168.1.6	104.26.8.101	TCP	571 [TCP Retransmission] 51415 → 443 [PSH, ACK] Seq=1 Ack=1 Win=131584 Len=517
585	4.820448	192.168.1.6	104.26.8.101	TCP	571 [TCP Retransmission] 51414 → 443 [PSH, ACK] Seq=1 Ack=1 Win=131584 Len=517
588	4.835686	104.26.8.101	192.168.1.6	TCP	54 443 → 51415 [ACK] Seq=1 Ack=518 Win=68608 Len=0
589	4.835686	104.26.8.101	192.168.1.6	TLSv1.3	1506 Server Hello, Change Cipher Spec
590	4.835686	104.26.8.101	192.168.1.6	TLSv1.3	719 Application Data
593	4.835686	104.26.8.101	192.168.1.6	TCP	54 443 → 51414 [ACK] Seq=1 Ack=518 Win=68608 Len=0

- from above image we can observe that first client (192.168.1.6) sends SYN signal to server and then in return it receives an ACK from 104.26.0.101 then client sends another ACK to the server in response to the previous ACK by server and hence the connection is established between (192.168.1.6) and (104.26.0.101)

8. Since the client and server program are running on localhost and hence we need to capture localhost traffic so switch from wireless (wifi) to Adapter for loopback packet capture and run the client and socket program

```
PS D:\coding assn sem6\cn assn\assn4> python server2.py 8890
server waiting for connection
127.0.0.1:57667 joined successfully
127.0.0.1:57667's query:  [1, 2, 3, 4, 5, 6, 7, 8, 9]
total connections  1
server waiting for connection
127.0.0.1:57667 leaves
```

- from above image we can see that client has joind with port 57667

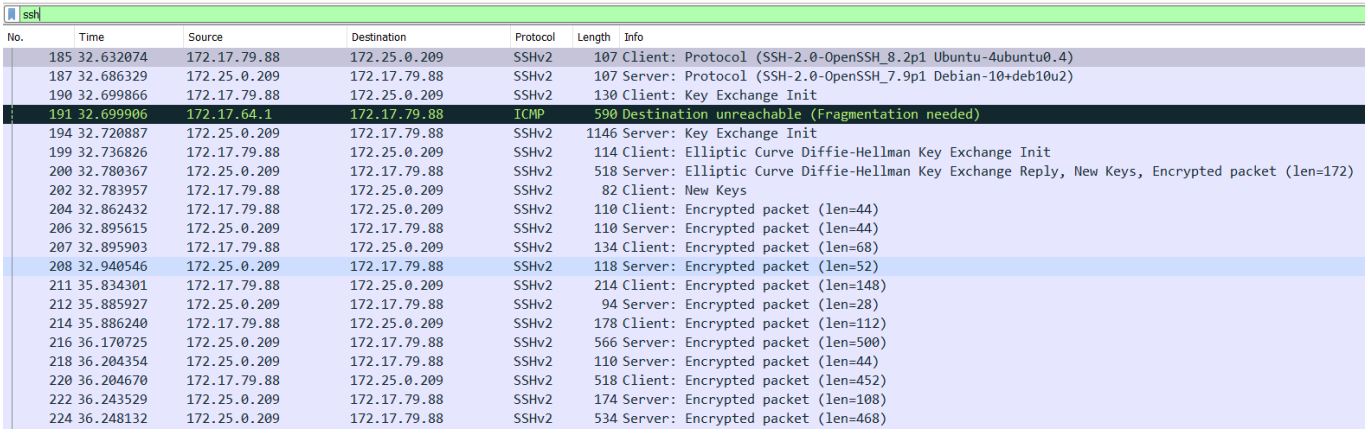


42	7.735040	127.0.0.1	127.0.0.1	TCP	56 57667 → 8890 [SYN] Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM=1
43	7.735082	127.0.0.1	127.0.0.1	TCP	56 8890 → 57667 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM=1
44	7.735103	127.0.0.1	127.0.0.1	TCP	44 57667 → 8890 [ACK] Seq=1 Ack=1 Win=2619648 Len=0
45	7.735138	127.0.0.1	127.0.0.1	TCP	71 57667 → 8890 [PSH, ACK] Seq=1 Ack=1 Win=2619648 Len=27
46	7.735147	127.0.0.1	127.0.0.1	TCP	44 8890 → 57667 [ACK] Seq=1 Ack=28 Win=2619648 Len=0
47	7.736040	127.0.0.1	127.0.0.1	TCP	71 8890 → 57667 [PSH, ACK] Seq=1 Ack=28 Win=2619648 Len=27
48	7.736058	127.0.0.1	127.0.0.1	TCP	44 57667 → 8890 [ACK] Seq=28 Ack=28 Win=2619648 Len=0
49	7.736117	127.0.0.1	127.0.0.1	TCP	44 57667 → 8890 [FIN, ACK] Seq=28 Ack=28 Win=2619648 Len=0
50	7.736132	127.0.0.1	127.0.0.1	TCP	44 8890 → 57667 [ACK] Seq=28 Ack=29 Win=2619648 Len=0

- from above figure we can see the communication between port 8890 (server port) and port 57667, we can observe that first client has send server req to join then server replied with an ack and then finally client responded and hence 3 way handshake is done, so now client has started sending data which has len=27 (from the figure) and then after the client closes the connection, it sends fin ack and the connection closes

9. connect to vpn and logon to iitj server using command

- > ssh sharma59@172.25.0.209
- > apply filter (ssh) to wireshark



ssh

No.	Time	Source	Destination	Protocol	Length	Info
185	32.632074	172.17.79.88	172.25.0.209	SSHv2	107	Client: Protocol (SSH-2.0-OpenSSH_8.2p1 Ubuntu-4ubuntu0.4)
187	32.686329	172.25.0.209	172.17.79.88	SSHv2	107	Server: Protocol (SSH-2.0-OpenSSH_7.9p1 Debian-10+deb10u2)
190	32.699866	172.17.79.88	172.25.0.209	SSHv2	130	Client: Key Exchange Init
191	32.699906	172.17.64.1	172.17.79.88	ICMP	590	Destination unreachable (Fragmentation needed)
194	32.720887	172.25.0.209	172.17.79.88	SSHv2	1146	Server: Key Exchange Init
199	32.736826	172.17.79.88	172.25.0.209	SSHv2	114	Client: Elliptic Curve Diffie-Hellman Key Exchange Init
200	32.780367	172.25.0.209	172.17.79.88	SSHv2	518	Server: Elliptic Curve Diffie-Hellman Key Exchange Reply, New Keys, Encrypted packet (len=172)
202	32.783957	172.17.79.88	172.25.0.209	SSHv2	82	Client: New Keys
204	32.862432	172.17.79.88	172.25.0.209	SSHv2	110	Client: Encrypted packet (len=44)
206	32.895615	172.25.0.209	172.17.79.88	SSHv2	110	Server: Encrypted packet (len=44)
207	32.895903	172.17.79.88	172.25.0.209	SSHv2	134	Client: Encrypted packet (len=68)
208	32.940546	172.25.0.209	172.17.79.88	SSHv2	118	Server: Encrypted packet (len=52)
211	35.834301	172.17.79.88	172.25.0.209	SSHv2	214	Client: Encrypted packet (len=148)
212	35.885927	172.25.0.209	172.17.79.88	SSHv2	94	Server: Encrypted packet (len=28)
214	35.886240	172.17.79.88	172.25.0.209	SSHv2	178	Client: Encrypted packet (len=112)
216	36.170725	172.25.0.209	172.17.79.88	SSHv2	566	Server: Encrypted packet (len=500)
218	36.204354	172.25.0.209	172.17.79.88	SSHv2	110	Server: Encrypted packet (len=44)
220	36.204670	172.17.79.88	172.25.0.209	SSHv2	518	Client: Encrypted packet (len=452)
222	36.243529	172.25.0.209	172.17.79.88	SSHv2	174	Server: Encrypted packet (len=108)
224	36.248132	172.25.0.209	172.17.79.88	SSHv2	534	Server: Encrypted packet (len=468)

- observe here that the destination ip is same as 172.25.0.209 and it establish ssh connection with remote iitj server and starts sending it packets

---