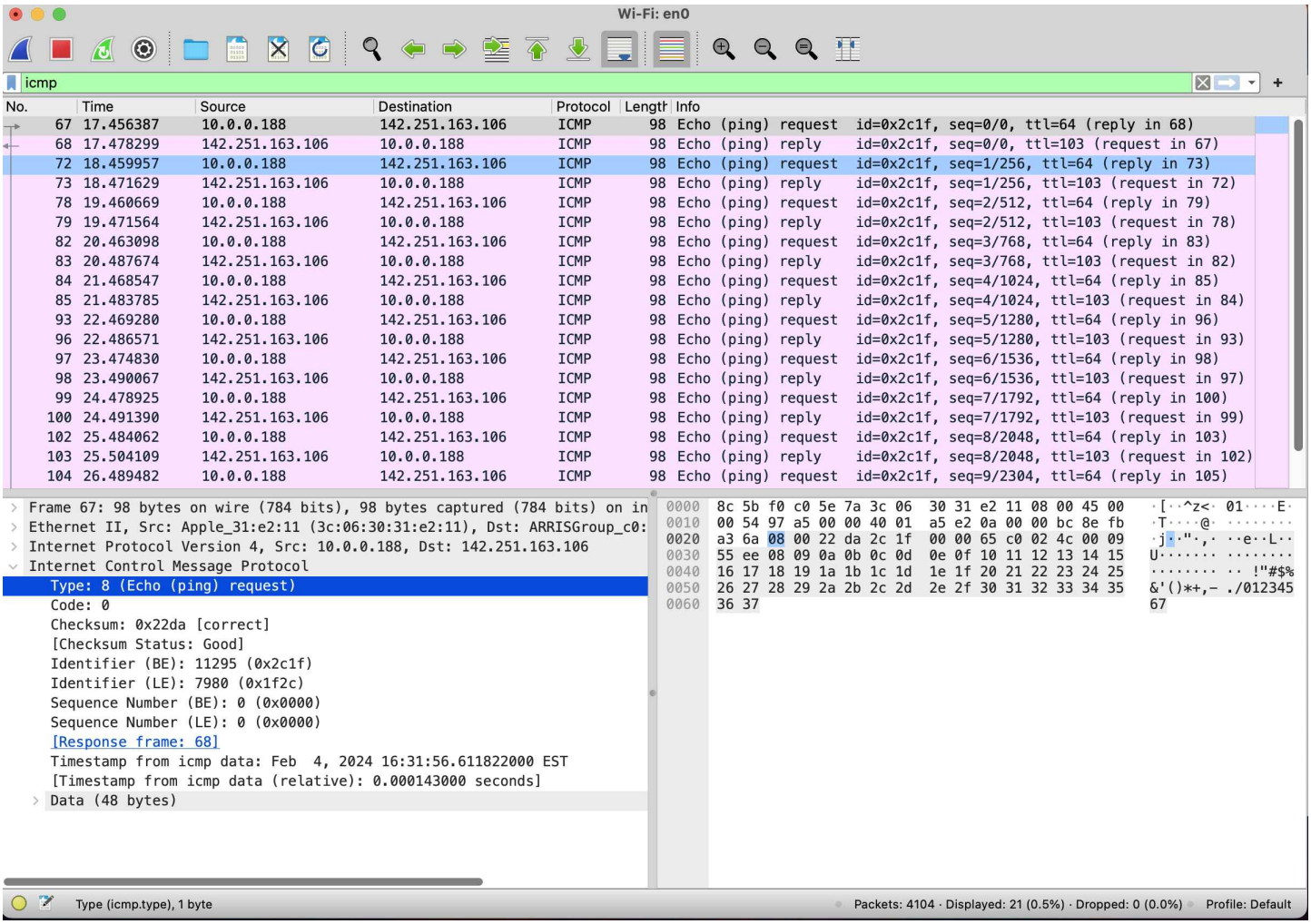
613homework0

Questions for ICMP Part:



1. Examine one of the ping request packets sent by your host. What are the ICMP type and code numbers?

- Type: 8
- Code: 0

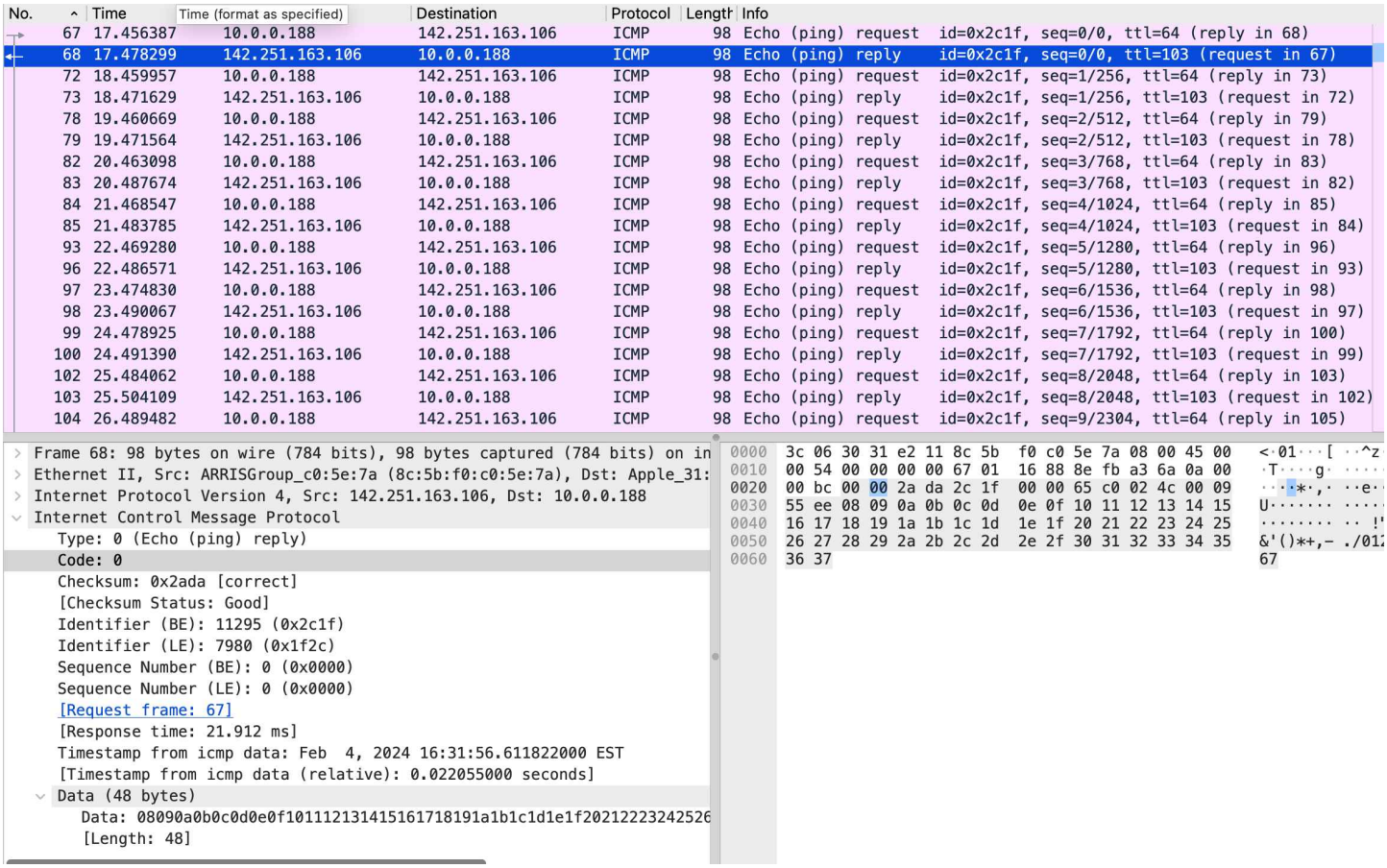
What other fields does this ICMP packet have?

- Checksum: 0x22da
- Identifier (BE): 11295 (0x2c1f)
- Identifier (LE): 7980 (0x1f2c)

- Sequence Number (BE): 0 (0x0000)
- Sequence Number (LE): 0 (0x0000)
- Response frame: 68
- Timestamp from icmp data: Feb 4, 2024 16:31:56.611822000 EST
- Timestamp from icmp data (relative): 0.000143000 seconds
- Data (48 bytes)
  Data:
  08090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f202122232425262728292a2b2c2d2e2f3031323334353637
  [Length: 48]

How many bytes are the checksum, sequence number and identifier fields? (10 points)

- Checksum: 2 bytes
- Identifier: 2 bytes
- Sequence Number: 2 bytes



2. Examine the corresponding ping reply packet. What are the ICMP type and code numbers?

- Type: 0 (Echo (ping) reply)
- Code: 0

What other fields does this ICMP packet have?

- Checksum: 0x2ada [correct]
- Identifier (BE): 11295 (0x2c1f)
- Identifier (LE): 7980 (0x1f2c)
- Sequence Number (BE): 0 (0x0000)
- Sequence Number (LE): 0 (0x0000)
- Request frame: 67
- Response time: 21.912 ms
- Timestamp from icmp data: Feb 4, 2024 16:31:56.611822000 EST
- Timestamp from icmp data (relative): 0.022055000 seconds
- Data (48 bytes)
  Data:
  08090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f20212223242526272829 2a2b2c2d2e2f3031323334353637
  [Length: 48]

How many bytes are the checksum, sequence number and identifier fields? (10 points)

- Checksum: 2 bytes
- Identifier: 2 bytes
- Sequence Number: 2 bytes

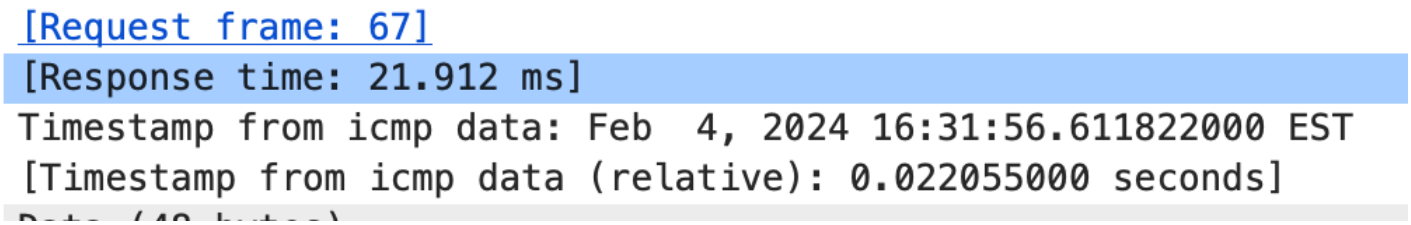
---

3. Examine the consecutive ICMP packets. Verify the RTT time reported in the command window is the same as the timestamps you observe via Wireshark.(Providing screenshots for command windows is helpful to demonstrate your results.) (20 points)

Those 2 screenshots show an example of how I gathered my data.

```
yjs@s-MacBook-Pro ~ % ping -c 10 www.google.com
PING www.google.com (142.251.163.106): 56 data bytes
64 bytes from 142.251.163.106: icmp_seq=0 ttl=103 time=22.190 ms
64 bytes from 142.251.163.106: icmp_seq=1 ttl=103 time=12.116 ms
64 bytes from 142.251.163.106: icmp_seq=2 ttl=103 time=11.257 ms
64 bytes from 142.251.163.106: icmp_seq=3 ttl=103 time=25.010 ms
64 bytes from 142.251.163.106: icmp_seq=4 ttl=103 time=15.690 ms
64 bytes from 142.251.163.106: icmp_seq=5 ttl=103 time=17.656 ms
64 bytes from 142.251.163.106: icmp_seq=6 ttl=103 time=15.850 ms
64 bytes from 142.251.163.106: icmp_seq=7 ttl=103 time=13.295 ms
64 bytes from 142.251.163.106: icmp_seq=8 ttl=103 time=20.717 ms
64 bytes from 142.251.163.106: icmp_seq=9 ttl=103 time=19.004 ms

--- www.google.com ping statistics ---
10 packets transmitted, 10 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 11.257/17.278/25.010/4.268 ms
```



This is the report that shows the difference in MS. The differences are the acceptance range. The biggest difference of RTT time in Terminal compare to Timestamps in Wireshark is 0.83ms.

	Terminal: RTT time	Wireshark: timestamps	Difference: ms
Track 0:	22.190	21.912	0.278
Track 1:	12.116	11.672	0.444
Track 2:	11.257	10.895	0.362
Track 3:	25.010	24.576	0.434
Track 4:	15.690	15.238	0.452
Track 5:	17.656	17.291	0.365
Track 6:	15.850	15.237	0.613
Track 7:	13.295	12.465	0.830
Track 8:	20.717	20.047	0.670
Track 9:	19.004	18.555	0.449

Questions for HTTP Part:

---

1. List up to 10 different protocols that appear in the protocol column in the unfiltered packet listing window before applying the filter. (10 points)

TCP: Transmission Control Protocal

ICMPv6: Internet Control Message Protocol v6

TLSv1.2: Transport Layer Security, TLSv1.2 Record Layer

TLSv1.3: Transport Layer Security, TLSv1.3 Record Layer

ARP: Address Resolution Protocol

MDNS: Multicase Domain Name System

DNS: Domain Name System

QUIC: Quick UDP Internet Connections

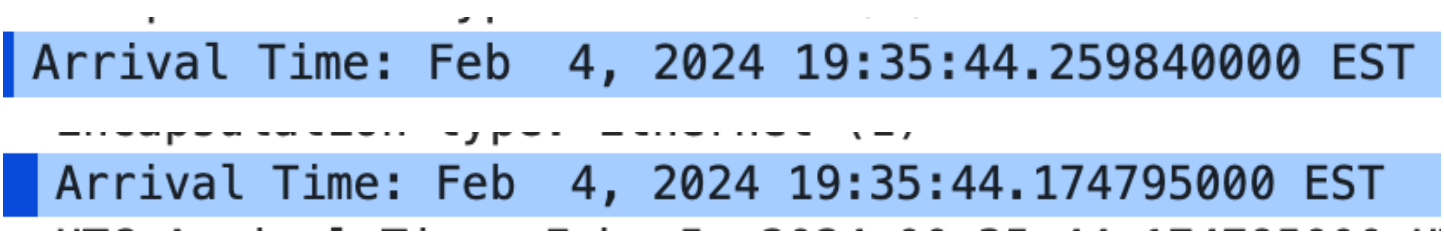
SSDP: Simple Service Discovery Protocol

UDP: User Datagram Protocol

HTTP: Hypertext Transfer Protocol

---

2. 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 The 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. (10 points)



0.259840000 - 0.174795000 = 0.085045 seconds = 85.045 milliseconds

---

3. What is the Internet address of the testingmcafeesites.com? What is the Internet

address of your computer? (10 points)

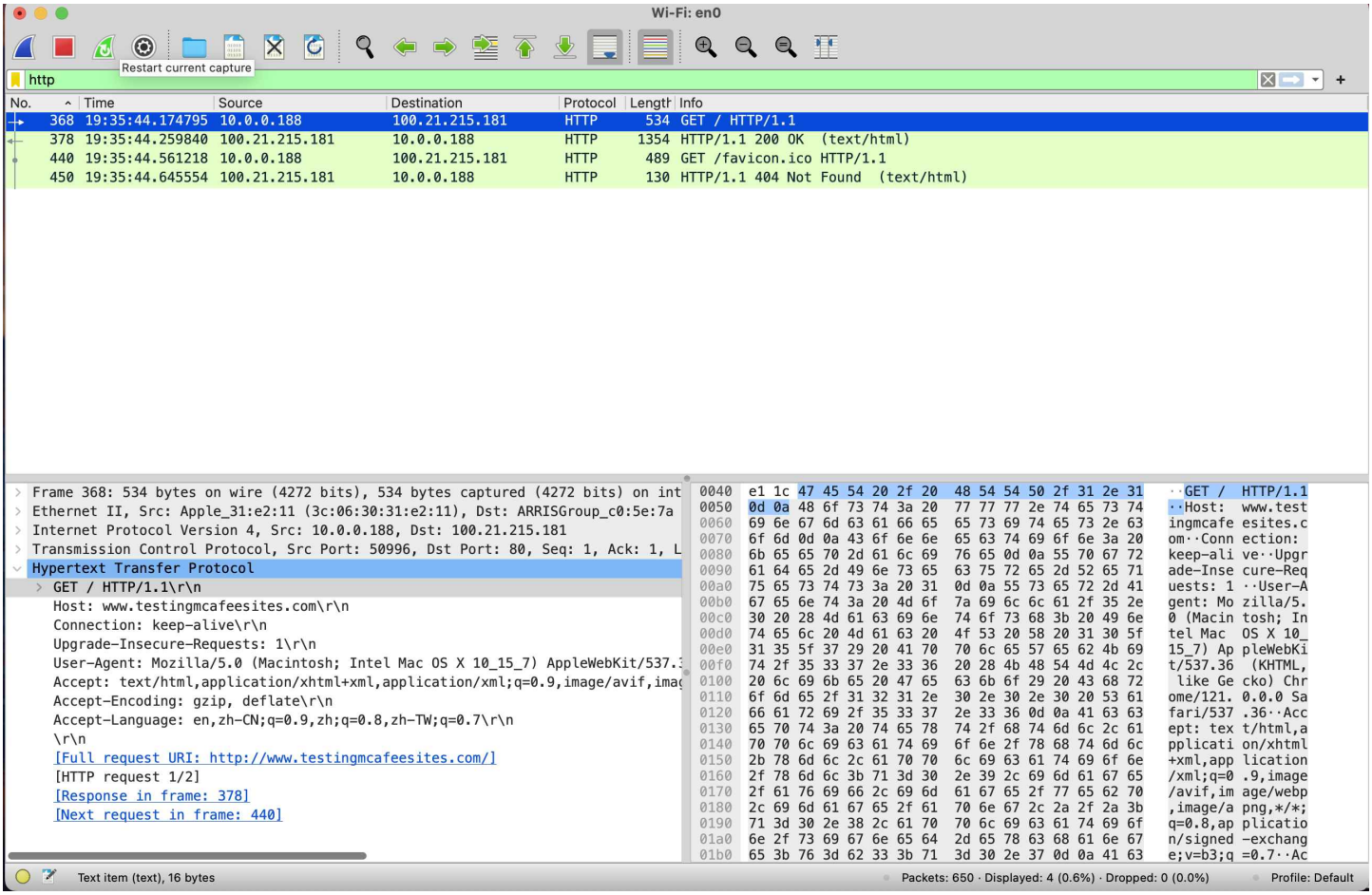
Source	Destination
10.0.0.188	100.21.215.181
100.21.215.181	10.0.0.188

My Internet Address: 10.0.0.188

Website Internet Address: 100.21.215.181

---

4. Provide a similar screenshot to Figure 7 with Wireshark running on your computer. (10 points)



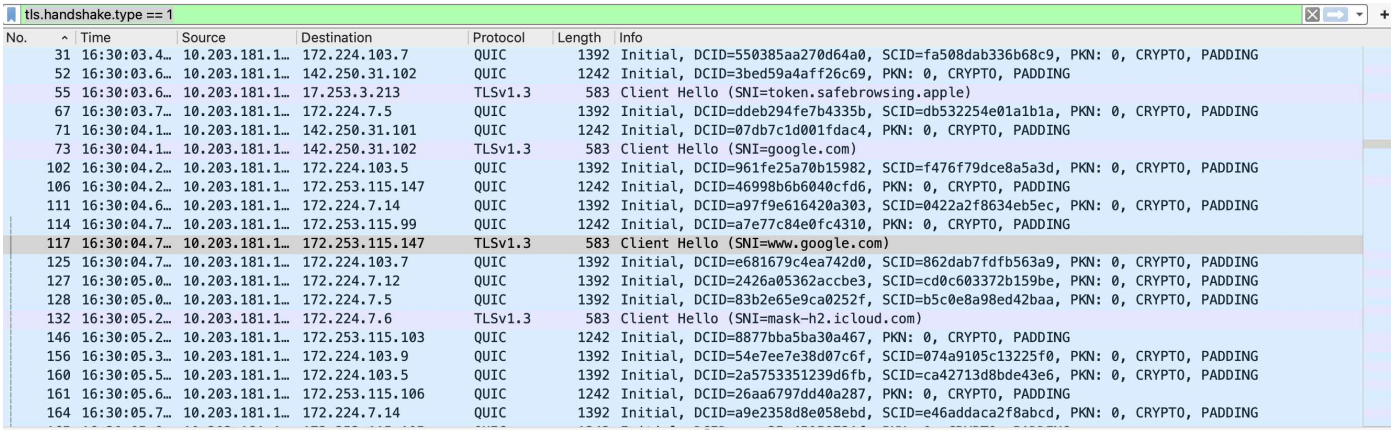
---

5. What happens if you open google.com? Why? (10 points)

I partially used this site to guide me through.

https://en.wikiversity.org/wiki/Wireshark/HTTPS

1. Start capturing packets in Wireshark.
2. Open Safari browser, clean all the cookies and caches, then open http://google.com
3. Since google uses HTTPS instead HTTP. I could not find anything under `http` FILTER, unlike what I did for the previous question.
4. I then put `tls.handshake.type == 1` in the filter to show the successful handshake connections. I found several TLS Handshake Protocols with Google.



No.	Time	Source	Destination	Protocol	Length	Info
31	16:30:03.4…	10.203.181.1…	172.224.103.7	QUIC	1392	Initial, DCID=550385aa270d64a0, SCID=fa508dab336b68c9, PKN: 0, CRYPTO, PADDING
52	16:30:03.6…	10.203.181.1…	142.250.31.102	QUIC	1242	Initial, DCID=3bed59a4aff26c69, PKN: 0, CRYPTO, PADDING
55	16:30:03.6…	10.203.181.1…	17.253.3.213	TLSv1.3	583	Client Hello (SNI=token.safebrowsing.apple)
67	16:30:03.7…	10.203.181.1…	172.224.7.5	QUIC	1392	Initial, DCID=ddeb294fe7b4335b, SCID=db532254e01a1b1a, PKN: 0, CRYPTO, PADDING
71	16:30:04.1…	10.203.181.1…	142.250.31.101	QUIC	1242	Initial, DCID=07db7c1d001fdac4, PKN: 0, CRYPTO, PADDING
73	16:30:04.1…	10.203.181.1…	142.250.31.102	TLSv1.3	583	Client Hello (SNI=google.com)
102	16:30:04.2…	10.203.181.1…	172.224.103.5	QUIC	1392	Initial, DCID=961fe25a70b15982, SCID=f476f79dce8a5a3d, PKN: 0, CRYPTO, PADDING
106	16:30:04.2…	10.203.181.1…	172.253.115.147	QUIC	1242	Initial, DCID=46998b6b6040cfd6, PKN: 0, CRYPTO, PADDING
111	16:30:04.6…	10.203.181.1…	172.224.7.14	QUIC	1392	Initial, DCID=a97f9e616420a303, SCID=0422a2f8634eb5ec, PKN: 0, CRYPTO, PADDING
114	16:30:04.7…	10.203.181.1…	172.253.115.99	QUIC	1242	Initial, DCID=a7e77c84e0fc4310, PKN: 0, CRYPTO, PADDING
117	16:30:04.7…	10.203.181.1…	172.253.115.147	TLSv1.3	583	Client Hello (SNI=www.google.com)
125	16:30:04.7…	10.203.181.1…	172.224.103.7	QUIC	1392	Initial, DCID=e681679c4ea742d0, SCID=862dab7fdfb563a9, PKN: 0, CRYPTO, PADDING
127	16:30:05.0…	10.203.181.1…	172.224.7.12	QUIC	1392	Initial, DCID=2426a05362accbe3, SCID=cd0c603372b159be, PKN: 0, CRYPTO, PADDING
128	16:30:05.0…	10.203.181.1…	172.224.7.5	QUIC	1392	Initial, DCID=83b2e65e9ca0252f, SCID=b5c0e8a98ed42baa, PKN: 0, CRYPTO, PADDING
132	16:30:05.2…	10.203.181.1…	172.224.7.6	TLSv1.3	583	Client Hello (SNI=mask-h2.icloud.com)
146	16:30:05.2…	10.203.181.1…	172.253.115.103	QUIC	1242	Initial, DCID=8877bba5ba30a467, PKN: 0, CRYPTO, PADDING
156	16:30:05.3…	10.203.181.1…	172.224.103.9	QUIC	1392	Initial, DCID=54e7ee7e38d07c6f, SCID=074a9105c13225f0, PKN: 0, CRYPTO, PADDING
160	16:30:05.5…	10.203.181.1…	172.224.103.5	QUIC	1392	Initial, DCID=2a5753351239d6fb, SCID=ca42713d8bde43e6, PKN: 0, CRYPTO, PADDING
161	16:30:05.6…	10.203.181.1…	172.253.115.106	QUIC	1242	Initial, DCID=26aa6797dd40a287, PKN: 0, CRYPTO, PADDING
164	16:30:05.7…	10.203.181.1…	172.224.7.14	QUIC	1392	Initial, DCID=a9e2358d8e058ebd, SCID=e46addaca2f8abcd, PKN: 0, CRYPTO, PADDING

73 Client Hello (SNI=google.com)

Internet Protocol Version 4, Src: 10.203.181.169, Dst: 142.250.31.102

117 Client Hello (SNI = www.google.com)

Internet Protocol Version 4, Src: 10.203.181.169, Dst: 172.253.115.147

962 Client Hello (SNI=apis.google.com)

Internet Protocol Version 4, Src: 10.203.181.169, Dst: 172.253.115.102

1014 Client Hello (SNI=ogs.google.com)

Internet Protocol Version 4, Src: 10.203.181.169, Dst: 172.253.122.113

1365 Client Hello (SNI=play.google.com)

Internet Protocol Version 4, Src: 10.203.181.169, Dst: 142.250.31.113

1398 Client Hello (SNI=adservice.google.com)

Internet Protocol Version 4, Src: 10.203.181.169, Dst: 142.251.163.154

1449 Client Hello (SNI=googleads.g.doubleclick.net)

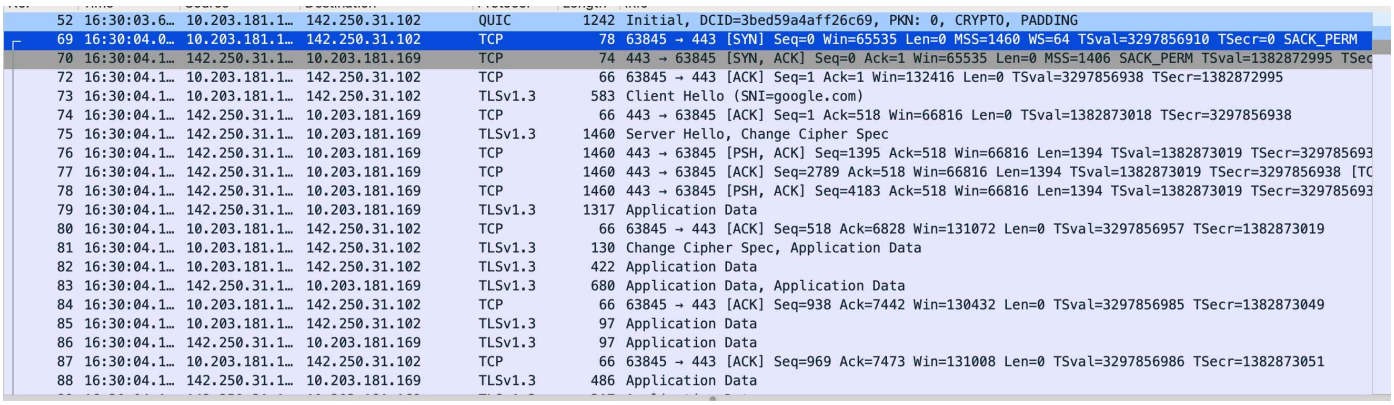
Internet Protocol Version 4, Src: 10.203.181.169, Dst: 172.253.122.155

Those 4 Server Name Indication leads to the advertisement or other platform of Google.

adservice.google.com, googleads.g.doubleclick.net, play.google.com, ogs.google.com

We will focus on analysis of the first three, google.com, www.google.com, ogs.google.com

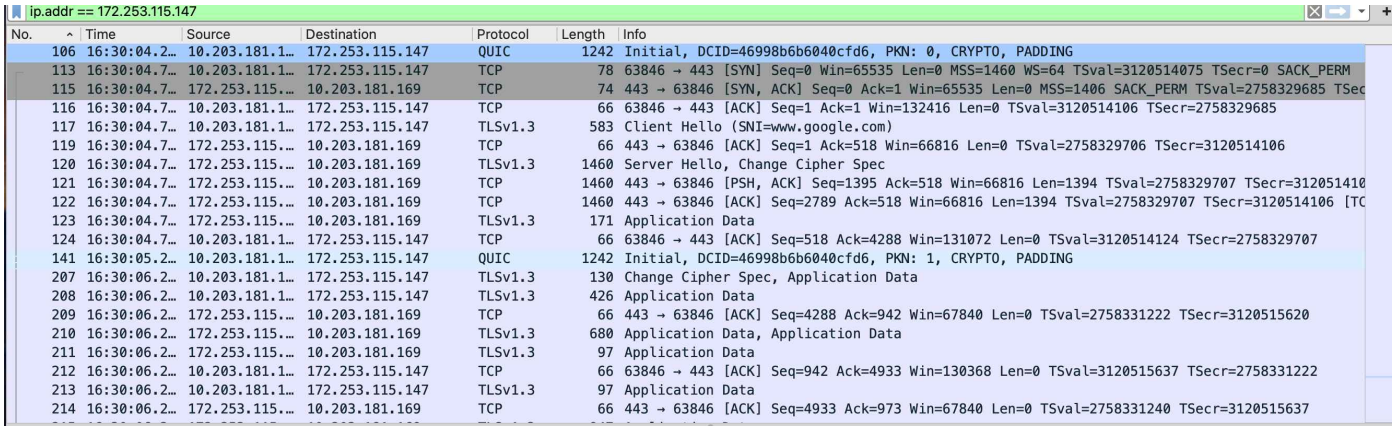
5. Since we already capture the IP address of those connections, we can use `ip.addr == xxx` to find more related information. An interesting thing at this step is that when you type google.com in your web browser. It will automatically convert to https with IPv6 address. When you type http://google.com, the Request URL will be http://www.google.co.in/ and redirect to https://www.google.co.in/. However, the packets in Wireshark will show the IP address in IPv4 instead of IPv6. Which I don't know exactly why :)
6. Let take a look at this part: 73 Client Hello (SNI=google.com) Internet Protocol Version 4, Src: 10.203.181.169, Dst: 142.250.31.102.



52	16:30:03.6…	10.203.181.1…	142.250.31.102	QUIC	1242	Initial, DCID=3bed59a4aff26c69, PKN: 0, CRYPTO, PADDING
69	16:30:04.0…	10.203.181.1…	142.250.31.102	TCP	78	63845 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TSval=3297856910 TSecr=0 SACK_PERM
70	16:30:04.1…	142.250.31.1…	10.203.181.169	TCP	74	443 → 63845 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1406 SACK_PERM TSval=1382872995 TSec
72	16:30:04.1…	10.203.181.1…	142.250.31.102	TCP	66	63845 → 443 [ACK] Seq=1 Ack=1 Win=132416 Len=0 TSval=3297856938 TSecr=1382872995
73	16:30:04.1…	10.203.181.1…	142.250.31.102	TLSv1.3	583	Client Hello (SNI=google.com)
74	16:30:04.1…	142.250.31.1…	10.203.181.169	TCP	66	443 → 63845 [ACK] Seq=1 Ack=518 Win=66816 Len=0 TSval=1382873018 TSecr=3297856938
75	16:30:04.1…	142.250.31.1…	10.203.181.169	TLSv1.3	1460	Server Hello, Change Cipher Spec
76	16:30:04.1…	142.250.31.1…	10.203.181.169	TCP	1460	443 → 63845 [PSH, ACK] Seq=1395 Ack=518 Win=66816 Len=1394 TSval=1382873019 TSecr=329785693
77	16:30:04.1…	142.250.31.1…	10.203.181.169	TCP	1460	443 → 63845 [ACK] Seq=2789 Ack=518 Win=66816 Len=1394 TSval=1382873019 TSecr=3297856938 [TC
78	16:30:04.1…	142.250.31.1…	10.203.181.169	TCP	1460	443 → 63845 [PSH, ACK] Seq=4183 Ack=518 Win=66816 Len=1394 TSval=1382873019 TSecr=329785693
79	16:30:04.1…	142.250.31.1…	10.203.181.169	TLSv1.3	1317	Application Data
80	16:30:04.1…	10.203.181.1…	142.250.31.102	TCP	66	63845 → 443 [ACK] Seq=518 Ack=6828 Win=131072 Len=0 TSval=3297856957 TSecr=1382873019
81	16:30:04.1…	10.203.181.1…	142.250.31.102	TLSv1.3	130	Change Cipher Spec, Application Data
82	16:30:04.1…	10.203.181.1…	142.250.31.102	TLSv1.3	422	Application Data
83	16:30:04.1…	142.250.31.1…	10.203.181.169	TLSv1.3	680	Application Data, Application Data
84	16:30:04.1…	10.203.181.1…	142.250.31.102	TCP	66	63845 → 443 [ACK] Seq=938 Ack=7442 Win=130432 Len=0 TSval=3297856985 TSecr=1382873049
85	16:30:04.1…	10.203.181.1…	142.250.31.102	TLSv1.3	97	Application Data
86	16:30:04.1…	142.250.31.1…	10.203.181.169	TLSv1.3	97	Application Data
87	16:30:04.1…	10.203.181.1…	142.250.31.102	TCP	66	63845 → 443 [ACK] Seq=969 Ack=7473 Win=131008 Len=0 TSval=3297856986 TSecr=1382873051
88	16:30:04.1…	142.250.31.1…	10.203.181.169	TLSv1.3	486	Application Data

My device send a TCP to google.com, it respond a TCP to me, then another TCP send from my end. Now the three-way handshake is done. My device send a Client Hello to google.com. Then I received a Change Cipher Spec Protocol(This protocol is used for the encryption of TLS connections), and a bunch of Application Data Protocols. Application Data Protocol is used to transmit data after a secured connection.

7. Basically, the same thing happen to the connection with this address: 117 Client Hello (SNI = www.google.com) Internet Protocol Version 4, Src: 10.203.181.169, Dst: 172.253.115.147. Three-way handshake -> Client Hello -> Change Cipher Spec. But the amount of Application Data I received is 10 times more than step 6. I guess this was the one website that popped up on my browser, since it had a lot more Application Data.



8. apis.google.com went throught the same process. I guess the connect with apis.google.com allows me to use google apis and other services at the google page.
9. Finally the site has loaded. I would not find any other info that relates to Google.