

613homework0

Questions for ICMP Part:

Wi-Fi: en0

No.	Time	Source	Destination	Protocol	Length	Info
67	17.456387	10.0.0.188	142.251.163.106	ICMP	98	Echo (ping) request id=0x2c1f, seq=0/0, ttl=64 (reply in 68)
68	17.478299	142.251.163.106	10.0.0.188	ICMP	98	Echo (ping) reply id=0x2c1f, seq=0/0, ttl=103 (request in 67)
72	18.459957	10.0.0.188	142.251.163.106	ICMP	98	Echo (ping) request id=0x2c1f, seq=1/256, ttl=64 (reply in 73)
73	18.471629	142.251.163.106	10.0.0.188	ICMP	98	Echo (ping) reply id=0x2c1f, seq=1/256, ttl=103 (request in 72)
78	19.460669	10.0.0.188	142.251.163.106	ICMP	98	Echo (ping) request id=0x2c1f, seq=2/512, ttl=64 (reply in 79)
79	19.471564	142.251.163.106	10.0.0.188	ICMP	98	Echo (ping) reply id=0x2c1f, seq=2/512, ttl=103 (request in 78)
82	20.463098	10.0.0.188	142.251.163.106	ICMP	98	Echo (ping) request id=0x2c1f, seq=3/768, ttl=64 (reply in 83)
83	20.487674	142.251.163.106	10.0.0.188	ICMP	98	Echo (ping) reply id=0x2c1f, seq=3/768, ttl=103 (request in 82)
84	21.468547	10.0.0.188	142.251.163.106	ICMP	98	Echo (ping) request id=0x2c1f, seq=4/1024, ttl=64 (reply in 85)
85	21.483785	142.251.163.106	10.0.0.188	ICMP	98	Echo (ping) reply id=0x2c1f, seq=4/1024, ttl=103 (request in 84)
93	22.469280	10.0.0.188	142.251.163.106	ICMP	98	Echo (ping) request id=0x2c1f, seq=5/1280, ttl=64 (reply in 96)
96	22.486571	142.251.163.106	10.0.0.188	ICMP	98	Echo (ping) reply id=0x2c1f, seq=5/1280, ttl=103 (request in 93)
97	23.474830	10.0.0.188	142.251.163.106	ICMP	98	Echo (ping) request id=0x2c1f, seq=6/1536, ttl=64 (reply in 98)
98	23.490067	142.251.163.106	10.0.0.188	ICMP	98	Echo (ping) reply id=0x2c1f, seq=6/1536, ttl=103 (request in 97)
99	24.478925	10.0.0.188	142.251.163.106	ICMP	98	Echo (ping) request id=0x2c1f, seq=7/1792, ttl=64 (reply in 100)
100	24.491390	142.251.163.106	10.0.0.188	ICMP	98	Echo (ping) reply id=0x2c1f, seq=7/1792, ttl=103 (request in 99)
102	25.484062	10.0.0.188	142.251.163.106	ICMP	98	Echo (ping) request id=0x2c1f, seq=8/2048, ttl=64 (reply in 103)
103	25.504109	142.251.163.106	10.0.0.188	ICMP	98	Echo (ping) reply id=0x2c1f, seq=8/2048, ttl=103 (request in 102)
104	26.489482	10.0.0.188	142.251.163.106	ICMP	98	Echo (ping) request id=0x2c1f, seq=9/2304, ttl=64 (reply in 105)

> Frame 67: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface en0
> Ethernet II, Src: Apple_31:e2:11 (3c:06:30:31:e2:11), Dst: ARRISGroup_c0:00:00:00:00:00
> Internet Protocol Version 4, Src: 10.0.0.188, Dst: 142.251.163.106
▼ Internet Control Message Protocol
Type: 8 (Echo (ping) request)
Code: 0
Checksum: 0x22da [correct]
[Checksum Status: Good]
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)

Type (icmp.type), 1 byte

Packets: 4104 · Displayed: 21 (0.5%) · Dropped: 0 (0.0%) · Profile: Default

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

No.	Time	Time (format as specified)	Destination	Protocol	Length	Info
67	17.456387	10.0.0.188	142.251.163.106	ICMP	98	Echo (ping) request id=0x2c1f, seq=0/0, ttl=64 (reply in 68)
68	17.478299	142.251.163.106	10.0.0.188	ICMP	98	Echo (ping) reply id=0x2c1f, seq=0/0, ttl=103 (request in 67)
72	18.459957	10.0.0.188	142.251.163.106	ICMP	98	Echo (ping) request id=0x2c1f, seq=1/256, ttl=64 (reply in 73)
73	18.471629	142.251.163.106	10.0.0.188	ICMP	98	Echo (ping) reply id=0x2c1f, seq=1/256, ttl=103 (request in 72)
78	19.460669	10.0.0.188	142.251.163.106	ICMP	98	Echo (ping) request id=0x2c1f, seq=2/512, ttl=64 (reply in 79)
79	19.471564	142.251.163.106	10.0.0.188	ICMP	98	Echo (ping) reply id=0x2c1f, seq=2/512, ttl=103 (request in 78)
82	20.463098	10.0.0.188	142.251.163.106	ICMP	98	Echo (ping) request id=0x2c1f, seq=3/768, ttl=64 (reply in 83)
83	20.487674	142.251.163.106	10.0.0.188	ICMP	98	Echo (ping) reply id=0x2c1f, seq=3/768, ttl=103 (request in 82)
84	21.468547	10.0.0.188	142.251.163.106	ICMP	98	Echo (ping) request id=0x2c1f, seq=4/1024, ttl=64 (reply in 85)
85	21.483785	142.251.163.106	10.0.0.188	ICMP	98	Echo (ping) reply id=0x2c1f, seq=4/1024, ttl=103 (request in 84)
93	22.469280	10.0.0.188	142.251.163.106	ICMP	98	Echo (ping) request id=0x2c1f, seq=5/1280, ttl=64 (reply in 96)
96	22.486571	142.251.163.106	10.0.0.188	ICMP	98	Echo (ping) reply id=0x2c1f, seq=5/1280, ttl=103 (request in 93)
97	23.474830	10.0.0.188	142.251.163.106	ICMP	98	Echo (ping) request id=0x2c1f, seq=6/1536, ttl=64 (reply in 98)
98	23.490067	142.251.163.106	10.0.0.188	ICMP	98	Echo (ping) reply id=0x2c1f, seq=6/1536, ttl=103 (request in 97)
99	24.478925	10.0.0.188	142.251.163.106	ICMP	98	Echo (ping) request id=0x2c1f, seq=7/1792, ttl=64 (reply in 100)
100	24.491390	142.251.163.106	10.0.0.188	ICMP	98	Echo (ping) reply id=0x2c1f, seq=7/1792, ttl=103 (request in 99)
102	25.484062	10.0.0.188	142.251.163.106	ICMP	98	Echo (ping) request id=0x2c1f, seq=8/2048, ttl=64 (reply in 103)
103	25.504109	142.251.163.106	10.0.0.188	ICMP	98	Echo (ping) reply id=0x2c1f, seq=8/2048, ttl=103 (request in 102)
104	26.489482	10.0.0.188	142.251.163.106	ICMP	98	Echo (ping) request id=0x2c1f, seq=9/2304, ttl=64 (reply in 105)

> Frame 68: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on in

> Ethernet II, Src: ARRISGroup_c0:5e:7a (8c:5b:f0:c0:5e:7a), Dst: Apple_31:

> Internet Protocol Version 4, Src: 142.251.163.106, Dst: 10.0.0.188

> Internet Control Message Protocol

Type: 0 (Echo (ping) reply)

Code: 0

Checksum: 0x2ada [correct]

[Checksum Status: Good]

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: 08090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f20212223242526

[Length: 48]

0000 3c 06 30 31 e2 11 8c 5b f0 c0 5e 7a 08 00 45 00 < 01... [^z

0010 00 54 00 00 00 00 67 01 16 88 8e fb a3 6a 0a 00 .T...g... ..

0020 00 bc 00 00 2a da 2c 1f 00 00 65 c0 02 4c 00 09 ...*,. ..e..

0030 55 ee 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15 U..... ..

0040 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25 !

0050 26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35 &'()*+,- ./01:

0060 36 37 67

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:
08090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f202122232425262728292a2b2c2d2e
2f3031323334353637
[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
```

[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 (40 bytes)

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 Protocol

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: Multicast 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 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)

Arrival Time: Feb 4, 2024 19:35:44.259840000 EST

Arrival Time: Feb 4, 2024 19:35:44.174795000 EST

$0.259840000 - 0.174795000 = 0.085045 \text{ seconds} = 85.045 \text{ milliseconds}$

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

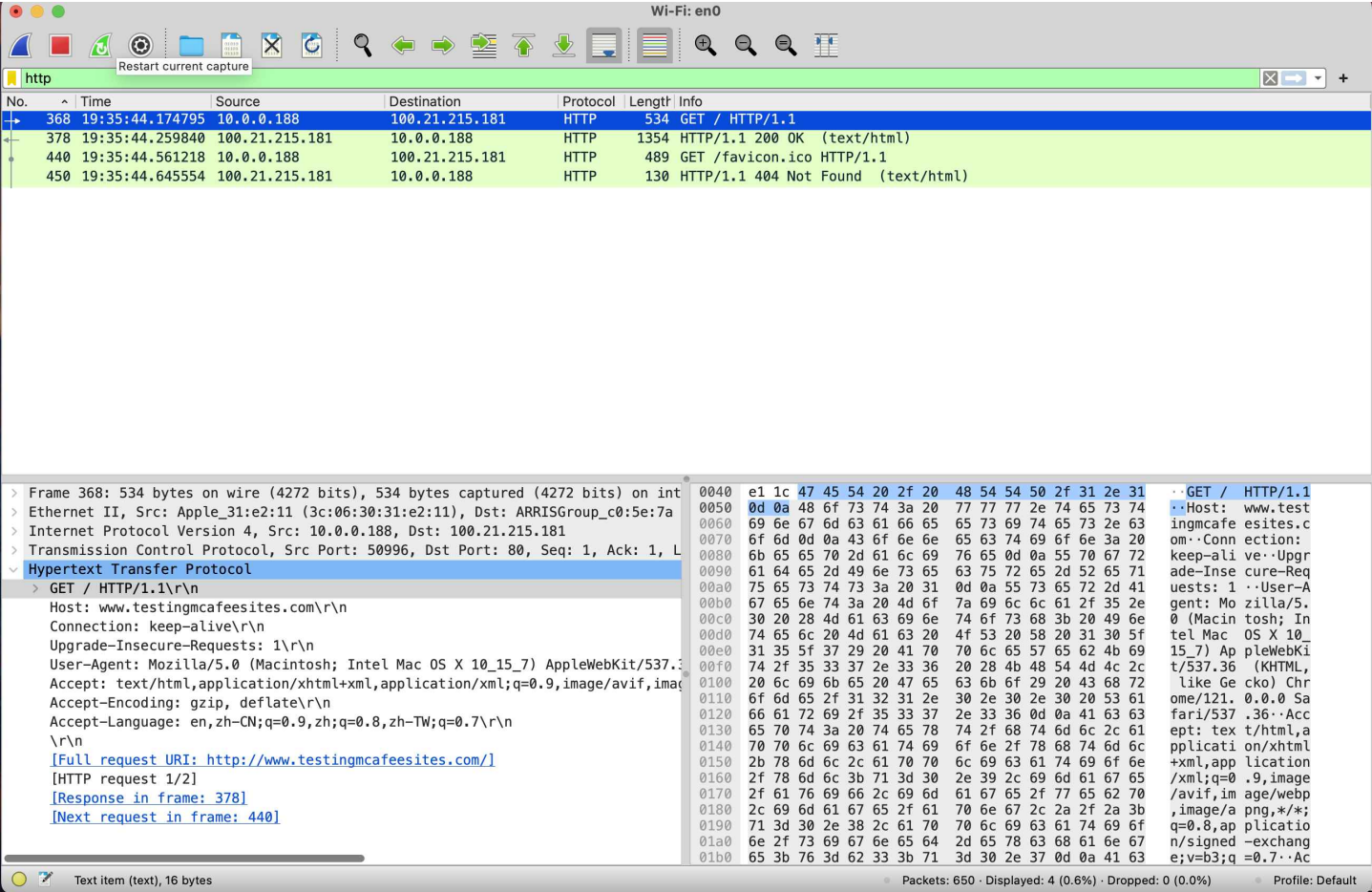
address of your computer? (10 points)

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.

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=862dab7fdff563a9, 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=1460 SACK_PERM TSval=1382872995 TSecr=0
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=3297856938
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
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=3297856938
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.

No.	Time	Source	Destination	Protocol	Length	Info
106	16:30:04.2...	10.203.181.1...	172.253.115.147	QUIC	1242	Initial, DCID=46998b6b6040cfd6, PKN: 0, CRYPTO, PADDING
113	16:30:04.7...	10.203.181.1...	172.253.115.147	TCP	78	63846 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TSval=3120514075 TSecr=0 SACK_PERM
115	16:30:04.7...	172.253.115...	10.203.181.169	TCP	74	443 → 63846 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1406 SACK_PERM TSval=2758329685 TSecr=
116	16:30:04.7...	10.203.181.1...	172.253.115.147	TCP	66	63846 → 443 [ACK] Seq=1 Ack=1 Win=132416 Len=0 TSval=3120514106 TSecr=2758329685
117	16:30:04.7...	10.203.181.1...	172.253.115.147	TLSv1.3	583	Client Hello (SNI=www.google.com)
119	16:30:04.7...	172.253.115...	10.203.181.169	TCP	66	443 → 63846 [ACK] Seq=1 Ack=518 Win=66816 Len=0 TSval=2758329706 TSecr=3120514106
120	16:30:04.7...	172.253.115...	10.203.181.169	TLSv1.3	1460	Server Hello, Change Cipher Spec
121	16:30:04.7...	172.253.115...	10.203.181.169	TCP	1460	443 → 63846 [PSH, ACK] Seq=1395 Ack=518 Win=66816 Len=1394 TSval=2758329707 TSecr=312051410
122	16:30:04.7...	172.253.115...	10.203.181.169	TCP	1460	443 → 63846 [ACK] Seq=2789 Ack=518 Win=66816 Len=1394 TSval=2758329707 TSecr=3120514106 [TC
123	16:30:04.7...	172.253.115...	10.203.181.169	TLSv1.3	171	Application Data
124	16:30:04.7...	10.203.181.1...	172.253.115.147	TCP	66	63846 → 443 [ACK] Seq=518 Ack=4288 Win=131072 Len=0 TSval=3120514124 TSecr=2758329707
141	16:30:05.2...	10.203.181.1...	172.253.115.147	QUIC	1242	Initial, DCID=46998b6b6040cfd6, PKN: 1, CRYPTO, PADDING
207	16:30:06.2...	10.203.181.1...	172.253.115.147	TLSv1.3	130	Change Cipher Spec, Application Data
208	16:30:06.2...	10.203.181.1...	172.253.115.147	TLSv1.3	426	Application Data
209	16:30:06.2...	172.253.115...	10.203.181.169	TCP	66	443 → 63846 [ACK] Seq=4288 Ack=942 Win=67840 Len=0 TSval=2758331222 TSecr=3120515620
210	16:30:06.2...	172.253.115...	10.203.181.169	TLSv1.3	680	Application Data, Application Data
211	16:30:06.2...	172.253.115...	10.203.181.169	TLSv1.3	97	Application Data
212	16:30:06.2...	10.203.181.1...	172.253.115.147	TCP	66	63846 → 443 [ACK] Seq=942 Ack=4933 Win=130368 Len=0 TSval=3120515637 TSecr=2758331222
213	16:30:06.2...	10.203.181.1...	172.253.115.147	TLSv1.3	97	Application Data
214	16:30:06.2...	172.253.115...	10.203.181.169	TCP	66	443 → 63846 [ACK] Seq=4933 Ack=973 Win=67840 Len=0 TSval=2758331240 TSecr=3120515637

8. apis.google.com went through 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.

