

**NAME – INDRANIL BAIN**  
**ENROLLMENT NO. – 2020CSB039**  
**GROUP – GX**  
**SUBJECT – COMPUTER NETWORK LAB**

## 1. Analysing Packets for given commands –

### (i) ping:

I ran the command ping -c google.com to get results like below –

No.	Protocol	Source	Destination	Protocol	Length	Info		
19018	icmp	142.250.66.14	192.168.29.34	ICMP	98	Echo (ping) reply id=0x0e9, seq=38/9728, ttl=112 (request in 2641)		
2643 96.286848		192.168.29.34	142.250.66.14	ICMP	98	Echo (ping) request id=0x0e9, seq=39/9984, ttl=63 (reply in 2644)		
2644 96.331886		142.250.66.14	192.168.29.34	ICMP	98	Echo (ping) reply id=0x0e9, seq=39/9984, ttl=112 (request in 2643)		
2735 97.288252		192.168.29.34	142.250.66.14	ICMP	98	Echo (ping) request id=0x0e9, seq=40/10240, ttl=63 (reply in 2736)		
2736 97.332973		142.250.66.14	192.168.29.34	ICMP	98	Echo (ping) reply id=0x0e9, seq=40/10240, ttl=112 (request in 2735)		
2788 98.289715		192.168.29.34	142.250.66.14	ICMP	98	Echo (ping) request id=0x0e9, seq=41/10496, ttl=63 (reply in 2789)		
2789 98.339772		142.250.66.14	192.168.29.34	ICMP	98	Echo (ping) reply id=0x0e9, seq=41/10496, ttl=112 (request in 2788)		
2829 99.291895		192.168.29.34	142.250.66.14	ICMP	98	Echo (ping) request id=0x0e9, seq=42/10752, ttl=63 (reply in 2830)		
2830 99.335791		142.250.66.14	192.168.29.34	ICMP	98	Echo (ping) reply id=0x0e9, seq=42/10752, ttl=112 (request in 2829)		
2832 100.293767		192.168.29.34	142.250.66.14	ICMP	98	Echo (ping) request id=0x0e9, seq=43/11088, ttl=63 (reply in 2833)		
2833 100.337991		142.250.66.14	192.168.29.34	ICMP	98	Echo (ping) reply id=0x0e9, seq=43/11088, ttl=112 (request in 2832)		
2837 101.295295		192.168.29.34	142.250.66.14	ICMP	98	Echo (ping) request id=0x0e9, seq=44/11264, ttl=63 (reply in 2838)		
2838 101.341799		142.250.66.14	192.168.29.34	ICMP	98	Echo (ping) reply id=0x0e9, seq=44/11264, ttl=112 (request in 2837)		
2839 102.297080		192.168.29.34	142.250.66.14	ICMP	98	Echo (ping) request id=0x0e9, seq=45/11520, ttl=63 (reply in 2840)		
2840 102.342386		142.250.66.14	192.168.29.34	ICMP	98	Echo (ping) request id=0x0e9, seq=45/11520, ttl=112 (request in 2839)		
2843 103.298611		192.168.29.34	142.250.66.14	ICMP	98	Echo (ping) request id=0x0e9, seq=46/11776, ttl=63 (reply in 2844)		
2844 103.343547		142.250.66.14	192.168.29.34	ICMP	98	Echo (ping) reply id=0x0e9, seq=46/11776, ttl=112 (request in 2843)		
2850 104.300765		192.168.29.34	142.250.66.14	ICMP	98	Echo (ping) request id=0x0e9, seq=47/12032, ttl=63 (reply in 2851)		
2851 104.345962		142.250.66.14	192.168.29.34	ICMP	98	Echo (ping) reply id=0x0e9, seq=47/12032, ttl=112 (request in 2850)		
3038 105.302104		192.168.29.34	142.250.66.14	ICMP	98	Echo (ping) request id=0x0e9, seq=48/12288, ttl=63 (reply in 3039)		
3039 105.369129		142.250.66.14	192.168.29.34	ICMP	98	Echo (ping) reply id=0x0e9, seq=48/12288, ttl=112 (request in 3038)		
3052 106.304042		192.168.29.34	142.250.66.14	ICMP	98	Echo (ping) request id=0x0e9, seq=49/12544, ttl=63 (reply in 3054)		
3054 106.357404		142.250.66.14	192.168.29.34	ICMP	98	Echo (ping) reply id=0x0e9, seq=49/12544, ttl=112 (request in 3052)		
3094 107.305324		192.168.29.34	142.250.66.14	ICMP	98	Echo (ping) request id=0x0e9, seq=50/12800, ttl=63 (reply in 3099)		
3099 107.353973		142.250.66.14	192.168.29.34	ICMP	98	Echo (ping) reply id=0x0e9, seq=50/12800, ttl=112 (request in 3094)		

```
> Frame 218: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface \Device\NPF_{BD2FAC1B} 00:00:00:00:00:00
> Ethernet II, Src: AzureWay_70:c0:75 (d8:c0:a6:70:c0:75), Dst: Serverco_b7:21:76 (8c:a3:99:b7:21:76)
> Internet Protocol Version 4, Src: 192.168.29.34, Dst: 142.250.66.14
> Internet Control Message Protocol
Frame 218: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface \Device\NPF_{BD2FAC1B} 00:00:00:00:00:00
  00:00:00:00:00:00  8c a3 99 b7 21 76 d8 c0 a6 70 c0 75 08 00 45 00  ...lv...-p-u-E-
  00:01:00 00 54 1b 78 40 00 03 f1 71 5e c0 a8 1d 22 8e fa  T-x@? q^...-.
  00:02:00 42 0e 08 00 6a 70 03 e8 00 01 1b 29 cd 63 00 00  B...jp...)-c...
  00:03:00 00 00 e2 46 00 00 00 00 00 00 10 11 12 13 14 15  ...F.....
  00:04:00 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25  ....F%...
  00:05:00 26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35  &(')*,-./.012345
  00:06:00 36 37

```

### (ii) traceroute:

Running traceroute 'isro.gov.in' gives the result –

No.	Time	Source	Destination	Protocol	Length	Info		
14325 468.184936		192.168.29.34	192.168.29.1	DNS	71	Standard query 0x9d7 A nasa.gov.in		
14326 468.208962		192.168.29.1	192.168.29.34	DNS	87	Standard query response 0x9d7 A nasa.gov.in A 210.212.115.198		
14327 468.238294		192.168.29.34	210.212.115.198	UDP	51	51627 > 33435 Len-9		
14328 468.240726		192.168.29.34	192.168.29.34	ICMP	79	Time-to-live exceeded (Time to live exceeded in transit)		
14329 468.241385		192.168.29.34	210.212.115.198	UDP	51	51627 > 33435 Len-9		
14330 468.243073		192.168.29.34	192.168.29.34	ICMP	79	Time-to-live exceeded (Time to live exceeded in transit)		
14331 468.243887		192.168.29.34	210.212.115.198	UDP	51	51627 > 33435 Len-9		
14332 468.245228		192.168.29.34	192.168.29.1	ICMP	79	Time-to-live exceeded (Time to live exceeded in transit)		
14333 468.245230		192.168.29.34	210.212.115.198	UDP	51	51627 > 33435 Len-9		
14334 468.250861	10:18:18.1	192.168.29.34	192.168.29.34	ICMP	110	Time-to-live exceeded (Time to live exceeded in transit)		
14335 468.252116	192.168.29.34	210.212.115.198	UDP	51	51627 > 33435 Len-9			
14336 468.256694	10:18:18.1	192.168.29.34	192.168.29.34	ICMP	110	Time-to-live exceeded (Time to live exceeded in transit)		
14337 468.257200		192.168.29.34	210.212.115.198	UDP	51	51627 > 33436 Len-9		
14338 468.264691	10:18:18.1	192.168.29.34	192.168.29.34	ICMP	110	Time-to-live exceeded (Time to live exceeded in transit)		
14339 468.268918		192.168.29.34	210.212.115.198	UDP	51	51627 > 33437 Len-9		
14340 468.270050	172.16.19.9	192.168.29.34	192.168.29.34	ICMP	110	Time-to-live exceeded (Time to live exceeded in transit)		
14341 468.271121		192.168.29.34	210.212.115.198	UDP	51	51627 > 33437 Len-9		
14342 468.278584	172.16.19.9	192.168.29.34	192.168.29.34	ICMP	110	Time-to-live exceeded (Time to live exceeded in transit)		
14343 468.279211		192.168.29.34	210.212.115.198	UDP	51	51627 > 33437 Len-9		
14344 468.303232	172.16.19.9	192.168.29.34	192.168.29.34	ICMP	110	Time-to-live exceeded (Time to live exceeded in transit)		
14345 468.304304		192.168.29.34	210.212.115.198	UDP	51	51627 > 33438 Len-9		
14346 468.346493	192.168.13.228	192.168.29.34	192.168.29.34	ICMP	79	Time-to-live exceeded (Time to live exceeded in transit)		
14347 468.347428		192.168.29.34	210.212.115.198	UDP	51	51627 > 33438 Len-9		
14348 468.392067	192.168.13.230	192.168.29.34	192.168.29.34	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)		
14349 468.392943		192.168.29.34	210.212.115.198	UDP	51	51627 > 33438 Len-9		

```
> Frame 218: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface \Device\NPF_{BD2FAC1B} 00:00:00:00:00:00
> Ethernet II, Src: AzureWay_70:c0:75 (d8:c0:a6:70:c0:75), Dst: Serverco_b7:21:76 (8c:a3:99:b7:21:76)
> Internet Protocol Version 4, Src: 192.168.29.34, Dst: 142.250.66.14
> Internet Control Message Protocol
Frame 218: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface \Device\NPF_{BD2FAC1B} 00:00:00:00:00:00
  00:00:00:00:00:00  8c a3 99 b7 21 76 d8 c0 a6 70 c0 75 08 00 88 00  ...lv...-p-u-E-
  00:01:00 00 54 1b 78 40 00 03 f1 71 5e c0 a8 1d 22 8e fa  T-x@? q^...-.
  00:02:00 42 0e 08 00 6a 70 03 e8 00 01 1b 29 cd 63 00 00  B...jp...)-c...
  00:03:00 00 00 e2 46 00 00 00 00 00 00 10 11 12 13 14 15  ...F.....
  00:04:00 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25  ....F%...
  00:05:00 26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35  &(')*,-./.012345
  00:06:00 36 37

```

(iii) arp:

Running arp gave the following result –

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.29.158	224.0.0.22	IGMPv3	54	Membership Report / Join group 224.0.0.251 for any sources
2	4.370144	fe80::8e3:99ff:feb: ff02::1		ICMPv6	142	Router Advertisement from 8c:a3:99:b7:21:76
3	3.390187	192.168.29.158	224.0.0.251	MDNS	103	Standard query 0x0131 PTR _CASEB412._sub._googlecast._tcp.local, "QM" question PTR _googlecast._tcp.local, "QM" question
No.	Time	Source	Destination	Protocol	Length	Info
8	3.7876761	192.168.29.1	192.168.29.34	DNS	86	Standard query response 0x5057 A github.com A 20.207.73.82
9	3.787047	192.168.29.1	192.168.29.34	DNS	135	Standard query response 0x5df5 AAAA github.com SOA dns1.p08.nsone.net
10	3.788309	192.168.29.34	192.168.29.1	DNS	70	Standard query 0xdfd7 AAAA github.com
11	3.790498	192.168.29.1	192.168.29.34	DNS	70	Standard query response 0xdfd7 AAAA github.com
12	3.793017	192.168.29.34	20.207.73.82	TCP	74	50122 + 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM Tsvl=1065183458 TSecr=0 WS=128
13	3.842659	20.207.73.82	192.168.29.34	TCP	74	443 + 50122 [SYN, ACK] Seq=1 Ack=1 Win=65535 Len=0 MSS=1436 SACK_PERM Tsvl=2031019625 TSecr=1065183458 WS=1024
14	3.844293	192.168.29.34	20.207.73.82	TCP	66	50122 + 443 [ACK] Seq=1 Ack=1 Win=64256 Len=0 Tsvl=1065183510 TSecr=2031019625
15	3.858347	192.168.29.34	20.207.73.82	TLSV1.3	378	Cleint Hello
16	3.908149	20.207.73.82	192.168.29.34	TLSV1.3	1490	Server Hello, Change Cipher Spec, Application Data
17	3.908149	20.207.73.82	192.168.29.34	TLSV1.3	1447	Application Data, Application Data, Application Data
18	3.910318	192.168.29.34	20.207.73.82	TCP	66	50122 + 443 [ACK] Seq=313 Ack=2086 Win=63616 Len=0 Tsvl=1065183576 TSecr=2031019690
19	3.956649	192.168.29.34	20.207.73.82	TLSV1.3	130	Change Cipher Spec, Application Data
20	3.998625	20.207.73.82	192.168.29.34	TLSV1.3	145	Application Data
21	3.998625	20.207.73.82	192.168.29.34	TLSV1.3	272	Application Data
22	3.998624	192.168.29.34	20.207.73.82	TCP	66	443 + 50122 [ACK] Seq=2964 Ack=583 Win=68608 Len=0 Tsvl=2031019874 TSecr=1065183664
23	4.392097	20.207.73.82	192.168.29.34	TLSV1.3	1458	Application Data
24	4.363265	20.207.73.82	192.168.29.34	TLSV1.3	1125	Application Data
25	4.363265	20.207.73.82	192.168.29.34	TCP	66	50122 + 443 [ACK] Seq=583 Ack=5415 Win=63744 Len=0 Tsvl=1065184029 TSecr=2031020145
27	4.366418	192.168.29.34	192.168.29.1	DNS	79	Standard query 0x7783 A codeLoad.github.com
28	4.366418	192.168.29.34	192.168.29.1	DNS	79	Standard query 0x643d AAAA codeLoad.github.com
29	4.394357	192.168.29.34	192.168.29.1	DNS	79	Standard query 0x7783 A codeLoad.github.com
30	4.395124	192.168.29.34	192.168.29.1	DNS	79	Standard query 0x643d AAAA codeLoad.github.com
31	4.481765	192.168.29.1	192.168.29.34	DNS	95	Standard query response 0x7783 A codeLoad.github.com A 20.207.73.88
32	4.419726	192.168.29.1	192.168.29.34	DNS	163	Standard query response 0x643d AAAA codeLoad.github.com SOA ns-1707.awsdns-21.co.uk
33	4.420458	192.168.29.34	192.168.29.1	DNS	79	Standard query 0x0fd5 AAAA codeLoad.github.com
34	4.422216	192.168.29.1	192.168.29.34	DNS	79	Standard query response 0x0fd5 AAAA codeLoad.github.com
35	4.425272	192.168.29.34	20.207.73.88	TCP	74	50156 + 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM Tsvl=2227723457 TSecr=0 WS=128
36	4.483807	20.207.73.88	192.168.29.34	TCP	74	443 + 50156 [SYN, ACK] Seq=1 Ack=1 Win=64256 Len=0 Tsvl=2227723516 TSecr=312283214 TSecr=2227723457 WS=1024
37	4.484765	192.168.29.34	20.207.73.88	TCP	66	50156 + 443 [ACK] Seq=1 Ack=1 Win=64256 Len=0 Tsvl=2227723516 TSecr=312283214
38	4.485524	192.168.29.34	20.207.73.88	TLSV1.3	387	Cleint Hello
39	4.543748	20.207.73.88	192.168.29.34	TLSV1.3	1490	Server Hello, Change Cipher Spec, Application Data
40	4.543748	20.207.73.88	192.168.29.34	TLSV1.3	1444	Application Data, Application Data, Application Data
41	4.545040	192.168.29.34	20.207.73.88	TCP	66	50156 + 443 [ACK] Seq=322 Ack=2083 Win=63616 Len=0 Tsvl=2227723576 TSecr=312283274

(iv) dig:

Running dig ocw.mit.edu gives me the following result –

No.	Time	Source	Destination	Protocol	Length	Info
1	0.0000000	192.168.29.34	239.255.255.250	SSDP	217	M-SEARCH * HTTP/1.1
2	0.492948	192.168.29.34	20.198.119.143	TLSv1.2	97	Application Data
3	0.648461	20.198.119.143	192.168.29.34	TLSv1.2	228	Application Data
4	0.694772	192.168.29.34	20.198.119.143	TCP	54	49723 > 443 [ACK] Seq=44 Ack=175 Win=516 Len=0
5	0.982725	192.168.29.34	192.168.29.1	DNS	71	Standard query 0xd6eb A ocm.mit.edu
6	0.991130	192.168.29.1	192.168.29.34	DNS	135	Standard answer response 0xd6eb A ocm.mit.edu A 151.101.2.133 A 151.101.194.133 A 151.101.130.133 A 151.101.66.133
7	1.008664	192.168.29.34	239.255.255.250	SSDP	217	M-SEARCH * HTTP/1.1
8	2.028294	192.168.29.34	239.255.255.250	SSDP	217	M-SEARCH * HTTP/1.1
9	2.513360	192.168.29.158	224.0.0.251	MDNS	103	Standard query 0x1d27 PTR _CASE8412._sub._googlecast._tcp.local, "QM" question PTR _googlecast._tcp.local, "QM" question
10	3.044475	192.168.29.34	239.255.255.250	SSDP	217	M-SEARCH * HTTP/1.1
11	3.923231	192.168.29.34	224.0.0.251	MDNS	85	Standard query 0x0000 PTR _microsoft_mcc._tcp.local, "QU" question
12	4.306772	192.168.29.1	224.0.0.1	IGMPv3	50	Membership Query, general
13	4.921584	192.168.29.34	224.0.0.251	MDNS	85	Standard query 0x0000 PTR _microsoft_mcc._tcp.local, "QU" question
14	5.467795	Azurewav_70:c0:75	Servicer_b7:21:76	ARP	42	Who has 192.168.29.1? Tell 192.168.29.34
15	5.476761	Servicer_b7:21:76	Azurewav_70:c0:75	ARP	42	192.168.29.1 is at 8c:a3:99:b7:21:76
16	6.468677	192.168.29.34	224.0.0.22	IGMPv3	54	Membership Report / Join group 224.0.0.252 for any sources
17	7.469970	192.168.29.34	224.0.0.22	IGMPv3	54	Membership Report / Join group 239.255.255.250 for any sources
18	10.154027	192.168.29.158	224.0.0.22	IGMPv3	54	Membership Report / Join group 224.0.0.251 for any sources
19	11.680036	fe80::8e3:99ff:feb0:1	ICMPv6	142	Router Advertisement from 8c:a3:99:b7:21:76	
20	12.459237	192.168.29.34	224.0.0.22	IGMPv3	54	Membership Report / Join group 224.0.0.251 for any sources

## (v) wget:

I downloaded a **github** repository using **wget**. The packet capture is shown below –

## 2. [This question cannot be solved because telnet daemon is not available anymore.]

3. While sending SSH request to the local server of IIEST (hamsa) and analysing the network traffic using Wireshark tool, we would be able to see all the packets that were transferred during this session. These packets include the initial connection request, the exchange of authentication information (such as username and password), and the subsequent data transfer between client and server. The data is encrypted hence not human-readable, but we can view the different packets and their corresponding size.

The packet data is shown below-

No.	Time	Source	Destination	Protocol	Length	Info
407	9.447587729	10.2.99.201	10.2.1.40	SSH	102	Client: Encrypted packet (len=36)
408	9.448241023	10.2.1.40	10.2.99.201	SSH	102	Server: Encrypted packet (len=36)
420	9.650892660	10.2.99.201	10.2.1.40	SSH	102	Client: Encrypted packet (len=36)
424	9.651572351	10.2.1.40	10.2.99.201	SSH	102	Server: Encrypted packet (len=36)
430	9.885158465	10.2.99.201	10.2.1.40	SSH	102	Client: Encrypted packet (len=36)
437	9.885915527	10.2.1.40	10.2.99.201	SSH	102	Server: Encrypted packet (len=36)
441	9.886998512	10.2.1.40	10.2.99.201	SSH	118	Server: Encrypted packet (len=52)
468	10.309737106	10.2.99.201	10.2.1.40	SSH	102	Client: Encrypted packet (len=36)
470	10.318573367	10.2.1.40	10.2.99.201	SSH	102	Server: Encrypted packet (len=36)
478	10.438024090	10.2.99.201	10.2.1.40	SSH	102	Client: Encrypted packet (len=36)
479	10.438708161	10.2.1.40	10.2.99.201	SSH	102	Server: Encrypted packet (len=36)
485	10.606265373	10.2.99.201	10.2.1.40	SSH	102	Client: Encrypted packet (len=36)
488	10.607077215	10.2.1.40	10.2.99.201	SSH	102	Server: Encrypted packet (len=36)
518	11.137070886	10.2.99.201	10.2.1.40	SSH	102	Client: Encrypted packet (len=36)
519	11.138941561	10.2.1.40	10.2.99.201	SSH	102	Server: Encrypted packet (len=36)
521	11.143265471	10.2.1.40	10.2.99.201	SSH	110	Server: Encrypted packet (len=44)
523	11.143760441	10.2.1.40	10.2.99.201	SSH	118	Server: Encrypted packet (len=52)

Frame 469: 102 bytes on wire (816 bits), 102 bytes captured (816 bits) on Ethernet II, Src: LCFCHefFe\_8f:d3:ee (8c:8c:aa:8f:d3:ee), Dst: Hewlett\_P\_96 (00:56:31:07:40:00) Internet Protocol Version 4, Src: 10.2.99.201, Dst: 10.2.1.40 Transmission Control Protocol, Src Port: 33788, Dst Port: 22, Seq: 109, Ack: 100, Len: 80  
SSH Protocol  
0000 9c b6 54 96 62 28 8c aa 8f d3 ee 08 00 45 48 .T-b...EH  
0010 00 58 31 07 40 00 06 90 5c 0a 02 63 c9 0a 02 .X1-@-.\n.c...  
0020 01 28 83 fc 00 16 51 8b ce 67 76 ac b1 84 88 18 .(....Q:gv....  
0030 01 f5 79 3f 00 00 01 01 08 00 0e 89 55 f4 52 91 .y?....UR.  
0040 29 1a 5f ac 3c c0 08 56 e2 7a 25 c5 20 d1 44 c4 )\_6-V.z%  
0050 50 83 8a 97 08 83 9c cf c8 76 9b 3c b8 3d e5 55 P.....v-<=U  
0060 a6 03 73 7a 80 50 ..sz.P

4. I entered the URL <http://gaia.cs.umass.edu/wireshark-labs/INTRO-wireshark-file1.html> in Chrome browser and started packet capture. Once the file download confirmation was shown, I stopped capturing packets.

The packet capture is shown below –

No.	Time	Source	Destination	Protocol	Length	Info
3943	103.718771	192.168.29.34	128.119.245.12	HTTP	527	GET /wireshark-labs/INTRO-wireshark-file1.html HTTP/1.1
3962	103.996673	128.119.245.12	192.168.29.34	HTTP	492	HTTP/1.1 200 OK (text/html)
3966	104.065759	192.168.29.34	128.119.245.12	HTTP	473	GET /favicon.ico HTTP/1.1
3971	104.344011	128.119.245.12	192.168.29.34	HTTP	538	HTTP/1.1 404 Not Found (text/html)

- a) We can see from the screenshot that **HTTP GET** was sent at Time **103.718771** and **HTTP OK** was received at Time **103.996673**. So, it took **277.902 milliseconds**.
- b) The IP address of gaia.cs.umass.edu is **128.119.245.12** whereas the IP address of my device is **192.168.29.34**. This is evident from the above screenshot.

5. I started the Wireshark packet capture service and then I typed the URL <https://www.gmail.com> in my browser. I stopped the packet capture after I have given my username and password to log on to Gmail.

The screenshot below shows the capture –

No.	Time	Source	Destination	Protocol	Length	Info
8	6.932335	192.168.29.34	192.168.29.34	DNS	73	Standard query 0xcB49 A www.gmail.com
9	6.932698	192.168.29.34	192.168.29.1	DNS	73	Standard query 0x08ec HTTPS www.gmail.com
10	6.938648	192.168.29.1	192.168.29.34	DNS	89	Standard query response 0xcB49 A www.gmail.com A 142.250.192.133
11	6.951354	192.168.29.34	142.250.192.133	TCP	66	52009 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
12	6.952158	192.168.29.34	142.250.192.133	TCP	66	52010 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
13	6.999554	142.250.192.133	192.168.29.34	TCP	66	443 → 52010 [SYN, ACK] Seq=1 Ack=1 Win=65535 Len=0 MSS=1412 SACK_PERM WS=256
14	6.999715	192.168.29.34	142.250.192.133	TCP	54	52010 → 443 [ACK] Seq=1 Ack=1 Win=131072 Len=0
15	7.000385	192.168.29.34	142.250.192.133	TLSv1.3	571	Client Hello
16	7.002425	142.250.192.133	192.168.29.34	TCP	66	443 → 52009 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1412 SACK_PERM WS=256
17	7.002520	192.168.29.34	142.250.192.133	TCP	54	52009 → 443 [ACK] Seq=1 Ack=1 Win=131072 Len=0
18	7.002810	192.168.29.34	142.250.192.133	TLSv1.3	571	Client Hello
19	7.018837	192.168.29.1	192.168.29.34	DNS	130	Standard query response 0x08ec HTTPS www.gmail.com SOA ns1.google.com
20	7.048512	192.168.29.158	224.0.0.22	IGMPv3	54	Membership Report / Join group 224.0.0.251 for any sources
21	7.048981	142.250.192.133	192.168.29.34	TCP	54	443 → 52010 [ACK] Seq=1 Ack=518 Win=66816 Len=0
22	7.053840	142.250.192.133	192.168.29.34	TCP	54	443 → 52009 [ACK] Seq=1 Ack=518 Win=66816 Len=0
23	7.109978	142.250.192.133	192.168.29.34	TLSv1.3	1466	Server Hello, Change Cipher Spec
24	7.109978	142.250.192.133	192.168.29.34	TCP	1466	443 → 52010 [PSH, ACK] Seq=1413 Ack=518 Win=66816 Len=1412 [TCP segment of a reassembled PDU]
25	7.109978	142.250.192.133	192.168.29.34	TCP	1466	443 → 52010 [ACK] Seq=2825 Ack=518 Win=66816 Len=1412 [TCP segment of a reassembled PDU]
26	7.109978	142.250.192.133	192.168.29.34	TLSv1.3	267	Application Data
27	7.110183	192.168.29.34	142.250.192.133	TCP	54	52010 → 443 [ACK] Seq=518 Ack=4450 Win=131072 Len=0
28	7.115471	142.250.192.133	192.168.29.34	TLSv1.3	1466	Server Hello, Change Cipher Spec
29	7.115471	142.250.192.133	192.168.29.34	TCP	1466	443 → 52009 [PSH, ACK] Seq=1413 Ack=518 Win=66816 Len=1412 [TCP segment of a reassembled PDU]
30	7.115471	142.250.192.133	192.168.29.34	TCP	1466	443 → 52009 [ACK] Seq=2825 Ack=518 Win=66816 Len=1412 [TCP segment of a reassembled PDU]
31	7.115471	142.250.192.133	192.168.29.34	TLSv1.3	265	Application Data
32	7.115701	192.168.29.34	142.250.192.133	TCP	54	52009 → 443 [ACK] Seq=518 Ack=4448 Win=131072 Len=0
33	7.116981	192.168.29.34	142.250.192.133	TLSv1.3	128	Change Cipher Spec, Application Data
34	7.118140	192.168.29.34	142.250.192.133	TLSv1.3	128	Change Cipher Spec, Application Data
35	7.118449	192.168.29.34	142.250.192.133	TCP	54	52009 → 443 [FIN, ACK] Seq=592 Ack=4448 Win=131072 Len=0
36	7.118684	192.168.29.34	142.250.192.133	TLSv1.3	152	Application Data
37	7.122835	192.168.29.34	142.250.192.133	TLSv1.3	714	Application Data
38	7.125186	192.168.29.34	192.168.29.1	DNS	80	Standard query 0x7c60 A beacons.gcp.gvt2.com
39	7.125432	192.168.29.34	192.168.29.1	DNS	80	Standard query 0x08cc HTTPS beacons.gcp.gvt2.com
40	7.130331	192.168.29.1	192.168.29.34	DNS	126	Standard query response 0x7c60 A beacons.gcp.gvt2.com CNAME beacons-handoff.gcp.gvt2.com A 172.217.160.227
41	7.132113	192.168.29.1	192.168.29.34	DNS	167	Standard query response 0x8bcc HTTPS beacons.gcp.gvt2.com CNAME beacons-handoff.gcp.gvt2.com SOA ns1.google.com
42	7.132677	192.168.29.34	172.217.160.227	TCP	66	52011 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
43	7.164001	142.250.192.133	192.168.29.34	TCP	54	444 → 52010 [ACK] Seq=4450 Ack=592 Win=66816 Len=0
44	7.164001	172.217.160.227	192.168.29.34	TCP	66	444 → 52011 [SYN, ACK] Seq=0 Ack=1 Win=131072 Len=0
45	7.164001	142.250.192.133	192.168.29.34	TLSv1.3	1088	Application Data, Application Data
46	7.164306	192.168.29.34	172.217.160.227	TCP	54	52011 → 443 [ACK] Seq=1 Ack=1 Win=131072 Len=0
47	7.164850	192.168.29.34	172.217.160.227	TLSv1.3	571	Client Hello
48	7.165069	192.168.29.34	142.250.192.133	TLSv1.3	85	Application Data

wireshark\_Wi-FiXEPY1.pcapng | Packets: 11606 · Displayed: 11606 (10.0%)

- a) Yes, the current application layer protocol is **TLS**. A primary use case of **TLS** is encrypting the communication between web applications and servers, such as web-browsers loading a website.  
But, in the previous case, it was **HTTP**.
- b) Previously, the protocol used was simple **HTTP**. As a result, sniffers like Wireshark could decode the data. But <https://www.gmail.com> is using **HTTPS** with **TLS**. So, the data is encrypted and Wireshark cannot decode it.

--:-

