CS212 : Computer Networks Lab
Assignment 3

T Satwik
190030043

1 Question 1

The protocols used by the application at different layers are:

- OSCP (Application Layer) (hangouts supports OCSP packets, instead of HTTP, hence in wireshark we get the corresponding OCSP packets)
- DNS (Application layer, uses UDP protocol)
- TCP (Transport Layer)
- TLS (Transport Layer) (TLS is TLSv1.2 and TLSv1.3 in wireshark)
- IP (Network Layer) (used to know the source and destiantion IP addresses)

2 Question 2

The observed values for various fields of the various protocols are:

OSCP: OSCP is like HTTP hence we can get various data like, Request Method, Request URL, Request version, Accept-Language, User-Agent, etc. These values are in the screenshot in Figure 1

DNS: DNS gives the IP address corresponding the given URL. It used the UDP protocol. We can see the values in the Figure 2

TCP: TCP is a handshaking protocol, and establishes a 3-step-protocols. In the wireshark interface, we can see the various details like the source and destination ports, sequence numbers, acknowledgement numbers, windows size sum, checksum values, etc. We can see all these values in Figure 3

TLS: In addition to the TCP data as above, TLS (TLSv1.2 and TLSv 1.3) would give a transport layer security section which contains

- client hello,
- server hello change cipher spec and encrypted handshake message.
- change cipher spec and encrypted handshake message.

These add the additional security to the packets. We can see these packets and the values, in the Figure 4

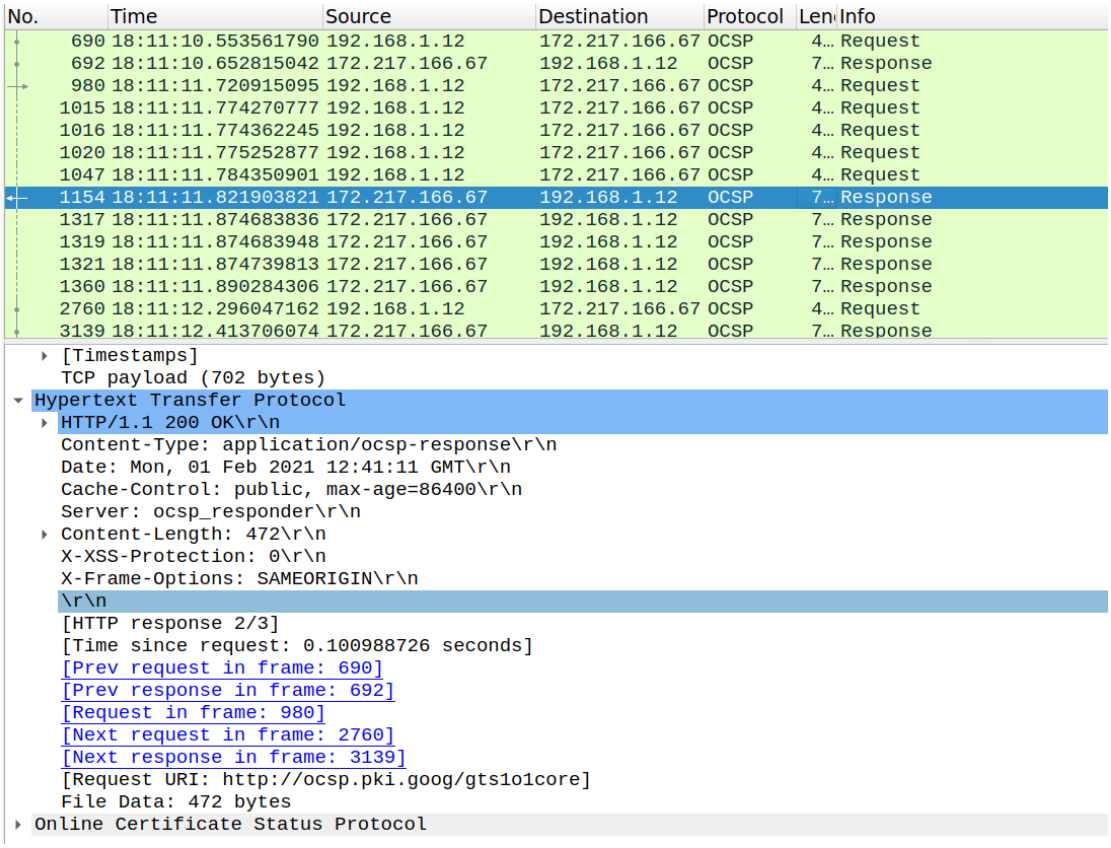


Figure 1: OCSP packets screenshot

Internet Protocol Version 4 This is a section which contains the information regarding the source and destination ports. The corresponding values are in the Figure 5 .

Note: The Screenshot images are also attached in the zip files

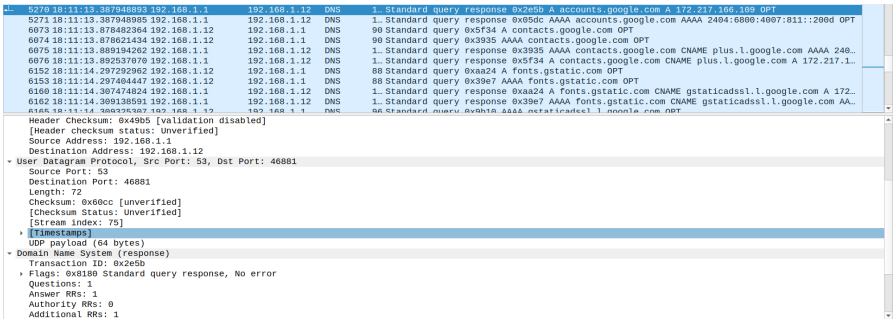


Figure 2: DNS packets screenshot

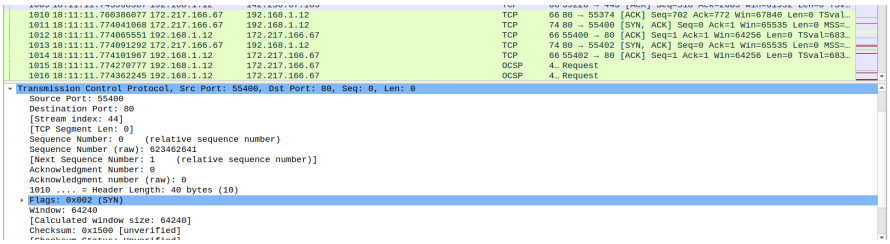


Figure 3: TCP packets screenshot

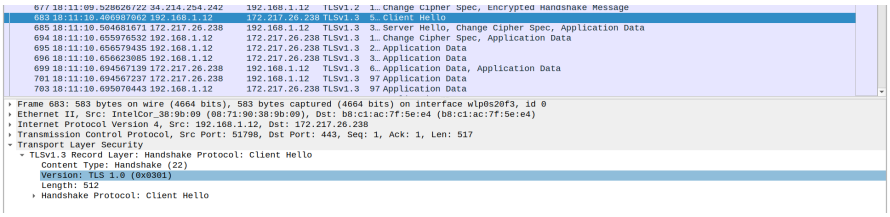


Figure 4: TLS packets screenshot

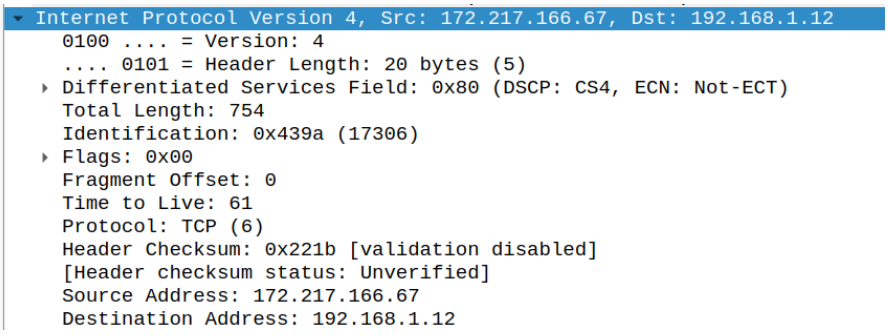


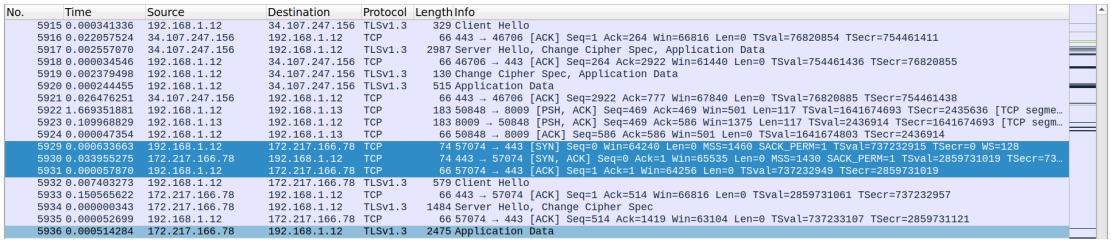
Figure 5: IP values screenshot

3 Question 3

The available functionalities for google hangout application/website are login make a call, send a message etc. When we use any of these functionalities, the number of TCP and TLSv1.2 values increases, and hence we can say that there are handshaking sequences involved in the packets corresponding to the functionality. I have attached another trace file in the zip file which corresponds to the trace after we click on the video call option in the website.

In the Figure 6, we can see that there is packet with SYN Flag from (192.)(i.e my system/client) to 172. (i.e google hangouts/server) and then there is packet from hangouts to my computer with SYN and ACK flags, which means that the server has acknowledged the syn packet from my computer, then there is ACK packet from my laptop to the hangouts server with ACK flag, acknowledging the syn packet from the server.

Hence there are 3 packets with SYN(client→server), SYN and ACK(server→client), ACK(client→server), hence it is a 3-step handshaking protocol.



No.	Time	Source	Destination	Protocol	Length	Info
5915	0.000341336	192.168.1.12	34.107.247.156	TLSv1.3	329	Client Hello
5916	0.022057524	34.107.247.156	192.168.1.12	TCP	66	443 → 46706 [ACK] Seq=1 Ack=264 Win=66816 Len=0 TSval=76820854 TSecr=754461411
5917	0.002557070	34.107.247.156	192.168.1.12	TLSv1.3	2987	Server Hello, Change Cipher Spec, Application Data
5918	0.000034546	192.168.1.12	34.107.247.156	TCP	66	46706 → 443 [ACK] Seq=264 Ack=2922 Win=61440 Len=0 TSval=754461436 TSecr=76820855
5919	0.002379498	192.168.1.12	34.107.247.156	TLSv1.3	130	Change Cipher Spec, Application Data
5920	0.000244455	192.168.1.12	34.107.247.156	TLSv1.3	515	Application Data
5921	0.026476251	34.107.247.156	192.168.1.12	TCP	66	443 → 46706 [ACK] Seq=2922 Ack=777 Win=67840 Len=0 TSval=76820885 TSecr=754461438
5922	1.669351881	192.168.1.12	192.168.1.13	TCP	183	50848 → 8009 [PSH, ACK] Seq=469 Ack=469 Win=501 Len=117 TSval=1641674693 TSecr=2435636 [TCP segme…
5923	0.109968829	192.168.1.13	192.168.1.12	TCP	183	8009 → 50848 [PSH, ACK] Seq=469 Ack=586 Win=1375 Len=117 TSval=2436914 TSecr=1641674693 [TCP segm…
5924	0.000047354	192.168.1.12	192.168.1.13	TCP	66	50848 → 8009 [ACK] Seq=586 Ack=586 Win=501 Len=0 TSval=1641674803 TSecr=2436914
5929	0.000633663	192.168.1.12	172.217.166.78	TCP	74	57074 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=737232915 TSecr=0 WS=128
5930	0.033955275	172.217.166.78	192.168.1.12	TCP	74	443 → 57074 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1430 SACK_PERM=1 TSval=2859731019 TSecr=73…
5931	0.000057870	192.168.1.12	172.217.166.78	TCP	66	57074 → 443 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=737232949 TSecr=2859731019
5932	0.007403273	192.168.1.12	172.217.166.78	TLSv1.3	579	Client Hello
5933	0.150565622	172.217.166.78	192.168.1.12	TCP	66	443 → 57074 [ACK] Seq=1 Ack=514 Win=66816 Len=0 TSval=2859731061 TSecr=737232957
5934	0.000000343	172.217.166.78	192.168.1.12	TLSv1.3	1484	Server Hello, Change Cipher Spec
5935	0.000052699	192.168.1.12	172.217.166.78	TCP	66	57074 → 443 [ACK] Seq=514 Ack=1419 Win=63104 Len=0 TSval=737233107 TSecr=2859731121
5936	0.000514284	172.217.166.78	192.168.1.12	TLSv1.3	2475	Application Data

Figure 6: Functionality packets screenshot

4 Question 4

The protocols like HTTP(OCSP), DNS, TCP, UDP, TLS are important because, they each perform a different task and are necessary for the functioning of the application as whole. DNS is used to get the IP address corresponding to the google.hangout url, so that we can establish a connection. HTTP gets data like request methods of the packets, accept language, browser information etc. TCP establishes a reliable and secure connection for data transfer with the handshaking protocol. TLS adds another layer of security by encrypting the data. UDP is useful for sending data not so securely, but with high speeds.

Hence as we can see each protocol performs a different task and hence are relevent for the overall functioning of the application.

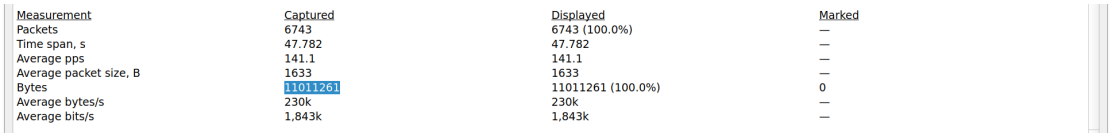
5 Question 5

Statistics from your traces while performing experiments is shown in Figure 7

Throughput=11011261/47.782=230447.888326

RTT is different for different packets, hence I found the min and max values by arranging them in ascending order.

RTT ranges from 0.000000029 to 2.989372596.



Measurement	Captured	Displayed	Marked
Packets	6743	6743 (100.0%)	—
Time span, s	47.782	47.782	—
Average pps	141.1	141.1	—
Average packet size, B	1633	1633	—
Bytes	11011261	11011261 (100.0%)	0
Average bytes/s	230k	230k	—
Average bits/s	1,843k	1,843k	—

Figure 7: Statistics screenshot

Packet Size is different for different packets, hence I found the min and max values by arranging them in ascending order.
Packet size ranges from 54 (for non ARP packets) to 15664.

Number of TCP packets lost = 54(obtained by the filter tcp.analysis.lost_segment)

Number of TCP packets = 6476 (cosidering the TLSv packets also)
Number of UDP packets = 203 (considering the DNS, MDNS packets also)

Number of packets with source as my laptop = 2850 (obtained by filter ip.src==192.168.1.12)
Number of packets with destination as my laptop = 3819 (obtained by filter ip.dst==192.168.1.12)

Number of responses received with respect to one request sent = 3819/2850
Number of responses received with respect to one request sent=1.34

All the values are tabulated in Table 1

Sl. No	Parameter	Value
1	Throughput	230447.888326
2	RTT	0.000000029 to 2.989372596
3	Packet Size	54 to 15664
4	TCP Packets Lost	54
5	Number of TCP packets	6476
6	Number of UDP packets	203
6	Number of responses per request	1.34

Table 1: Statistics

6 Question 6

The whole content is being sent from multiple locations, the list of those IP addresses is as follows:

- 172.217.166.109 (main)
- 142.250.67.163
- 216.58.203.46

It is common for servers to have multiple IP addresses, because when one of the IP address is down, the others can act as backup and also each IP address has its own merits and reputability, using different IP addresses would give the benefits of each of them.