CS342 ASSIGNMENT 2

Traces: http://bit.ly/180101093_traces_assign2

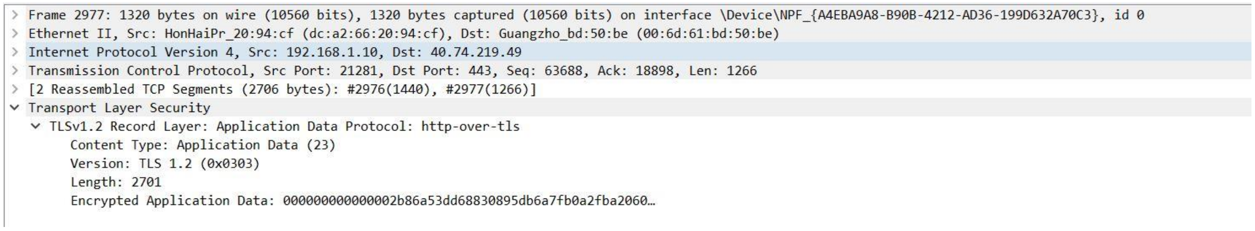
Name: Pulkit Changoiwala   Roll No: 180101093

Ans-1---

See Trace file name Ans1.

Application Layer:

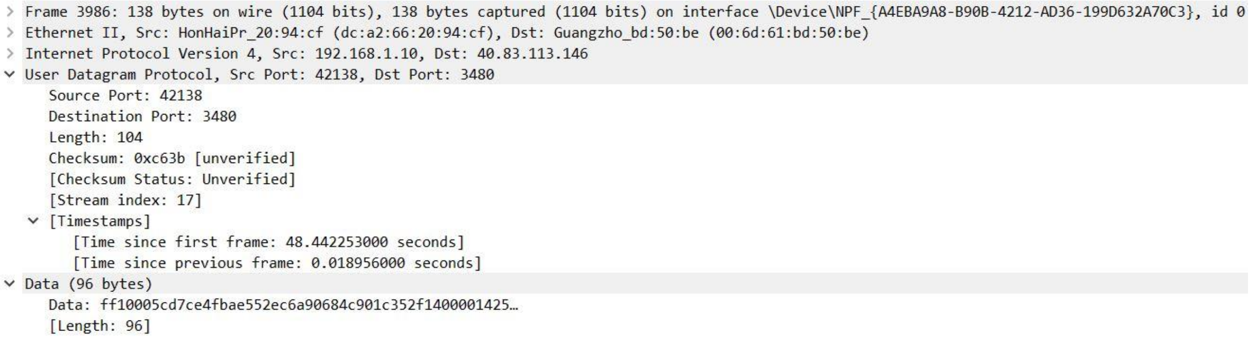
1) TLSv1.2 (Transport Layer Security): Transport Layer Security, or TLS, is a widely adopted security protocol designed to facilitate privacy and data security for communications over the Internet.



```
> Frame 2977: 1320 bytes on wire (10560 bits), 1320 bytes captured (10560 bits) on interface \Device\NPF_{A4EBA9A8-B90B-4212-AD36-199D632A70C3}, id 0
> Ethernet II, Src: HonHaiPr_20:94:cf (dc:a2:66:20:94:cf), Dst: Guangzho_bd:50:be (00:6d:61:bd:50:be)
> Internet Protocol Version 4, Src: 192.168.1.10, Dst: 40.74.219.49
> Transmission Control Protocol, Src Port: 21281, Dst Port: 443, Seq: 63688, Ack: 18898, Len: 1266
> [2 Reassembled TCP Segments (2706 bytes): #2976(1440), #2977(1266)]
v Transport Layer Security
  v TLSv1.2 Record Layer: Application Data Protocol: http-over-tls
      Content Type: Application Data (23)
      Version: TLS 1.2 (0x0303)
      Length: 2701
      Encrypted Application Data: 000000000000002b86a53dd68830895db6a7fb0a2fba2060…
```

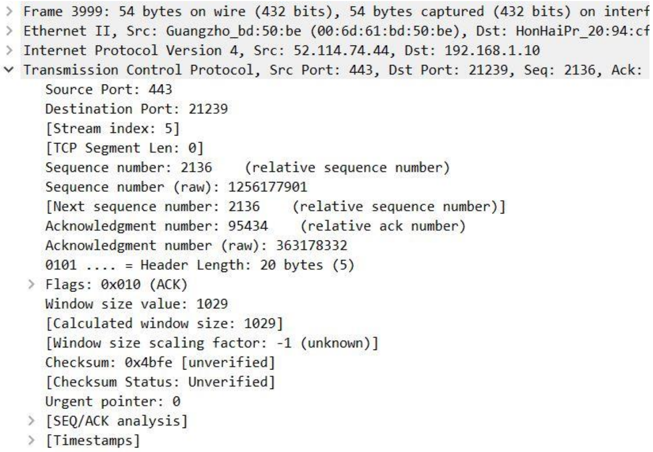
Transport Layer:

1) UDP (User Datagram Protocol): It is the simplest transport layer protocol. It simply takes the datagram from the network layer, attaches its header and sends it to the user.

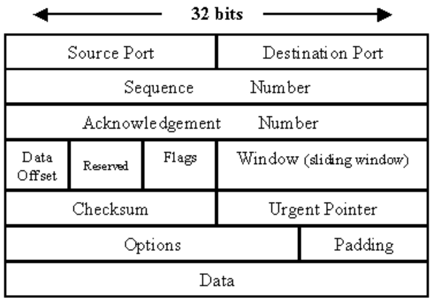


```
> Frame 3986: 138 bytes on wire (1104 bits), 138 bytes captured (1104 bits) on interface \Device\NPF_{A4EBA9A8-B90B-4212-AD36-199D632A70C3}, id 0
> Ethernet II, Src: HonHaiPr_20:94:cf (dc:a2:66:20:94:cf), Dst: Guangzho_bd:50:be (00:6d:61:bd:50:be)
> Internet Protocol Version 4, Src: 192.168.1.10, Dst: 40.83.113.146
v User Datagram Protocol, Src Port: 42138, Dst Port: 3480
    Source Port: 42138
    Destination Port: 3480
    Length: 104
    Checksum: 0xc63b [unverified]
    [Checksum Status: Unverified]
    [Stream index: 17]
  v [Timestamps]
      [Time since first frame: 48.442253000 seconds]
      [Time since previous frame: 0.018956000 seconds]
v Data (96 bytes)
    Data: ff10005cd7ce4fbae552ec6a90684c901c352f1400001425…
    [Length: 96]
```

2) TCP (Transport Control Protocol): It is a set of protocols or rules and procedures that governs communications among computers on the internet.



```
> Frame 3999: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interf
> Ethernet II, Src: Guangzho_bd:50:be (00:6d:61:bd:50:be), Dst: HonHaiPr_20:94:cf
> Internet Protocol Version 4, Src: 52.114.74.44, Dst: 192.168.1.10
v Transmission Control Protocol, Src Port: 443, Dst Port: 21239, Seq: 2136, Ack:
    Source Port: 443
    Destination Port: 21239
    [Stream index: 5]
    [TCP Segment Len: 0]
    Sequence number: 2136    (relative sequence number)
    Sequence number (raw): 1256177901
    [Next sequence number: 2136    (relative sequence number)]
    Acknowledgment number: 95434    (relative ack number)
    Acknowledgment number (raw): 363178332
    0101 .... = Header Length: 20 bytes (5)
  > Flags: 0x010 (ACK)
    Window size value: 1029
    [Calculated window size: 1029]
    [Window size scaling factor: -1 (unknown)]
    Checksum: 0x4bfe [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
  > [SEQ/ACK analysis]
  > [Timestamps]
```



UDP Packets:

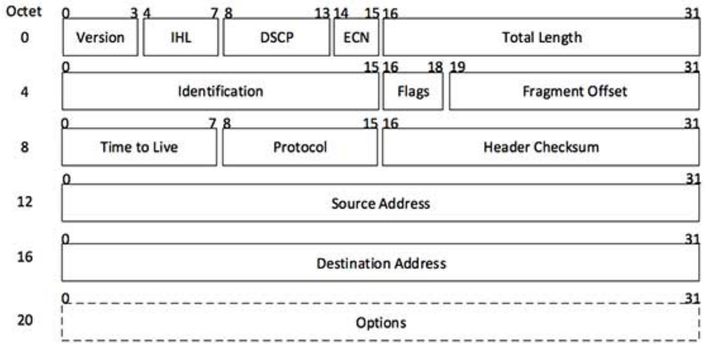
- **Source Port**: Port of the sender, a 16-bit field
- **Destination Port**: Port of receiver application
- **Length**: Combined length of UDP header & encapsulated data.
- **UDP Checksum:** It is an error detection scheme. It is the 16-bit one's complement of the one's complement sum of the UDP header, pseudo header of information from the IP header and the data

TCP Packets:

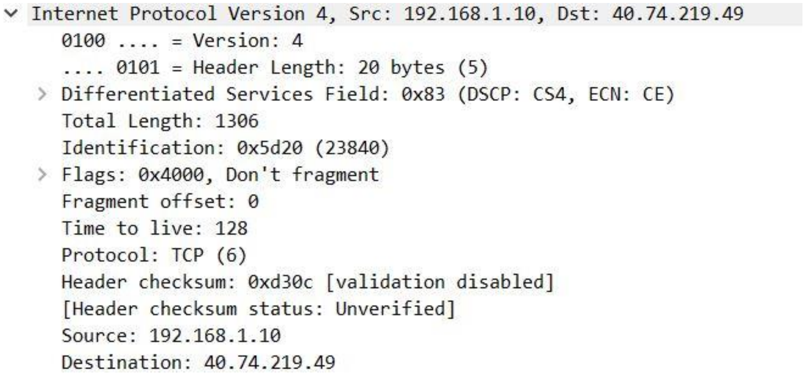
- **Sequence Number:** To mark an order in a group of packets/messages.
- **Ack Number**: It contains the sequence no. of the data byte that the receiver expects to receive next.
- **Flags**: There are total 6 types of flags, each of 1 bit. Ack, Syn, Psh, Urg etc.
- **Header Length**: 4-bit field & it contains the length of TCP header.
- **Window Size**: It shows how much data(in bytes) the sender can receive without acknowledgement.
- **Urgent Pointer**: It indicates how much data in the current segment counting from the first data byte is urgent.

Network Layer:

1) **IPv4:** is the main protocol of standard-based internetworking methods on the internet IP is responsible to deliver data packets.



[Image: IP Header]

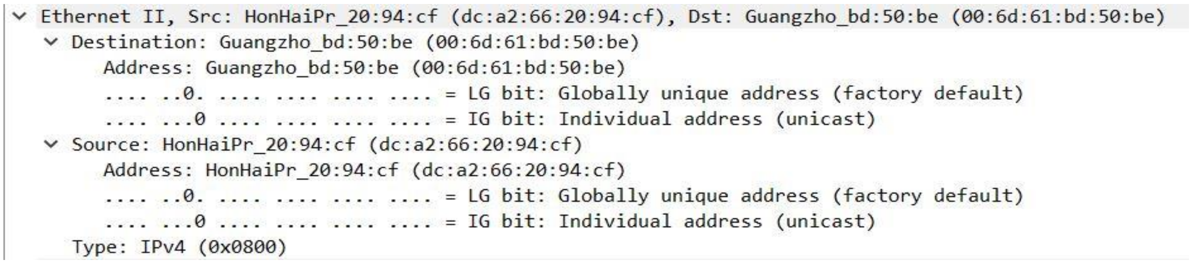


```
⌄ Internet Protocol Version 4, Src: 192.168.1.10, Dst: 40.74.219.49
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  › Differentiated Services Field: 0x83 (DSCP: CS4, ECN: CE)
    Total Length: 1306
    Identification: 0x5d20 (23840)
  › Flags: 0x4000, Don't fragment
    Fragment offset: 0
    Time to live: 128
    Protocol: TCP (6)
    Header checksum: 0xd30c [validation disabled]
    [Header checksum status: Unverified]
    Source: 192.168.1.10
    Destination: 40.74.219.49
```

Version: Indicates the IP version used.
Header Length: Contains the length of the IP header
Types of Services: Used for Quality of Service(QoS).
Total Length: It is a 16-bit field that contains the total length of the datagram (in bytes).
Identification: of the fragments of an original IP datagram.
DF/MF bits: DF bit stands for Do Not Fragment and MF stands for More Fragment bits.
Time to Live: indicates the maximum no. of hops a datagram can take to reach the dest.
Protocol: It tells the network layer at the dest. host to which protocol the IP datagram belongs.
Src/Dest IP addr. : It contains the logical address of sender and receiver of the datagram

Link Layer:

1) **Ethernet:** It is the most common LAN technology.



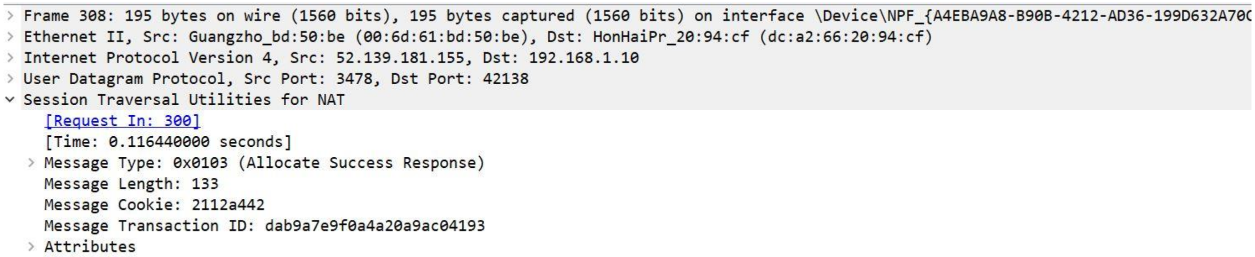
```
⌄ Ethernet II, Src: HonHaiPr_20:94:cf (dc:a2:66:20:94:cf), Dst: Guangzho_bd:50:be (00:6d:61:bd:50:be)
  ⌄ Destination: Guangzho_bd:50:be (00:6d:61:bd:50:be)
      Address: Guangzho_bd:50:be (00:6d:61:bd:50:be)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
  ⌄ Source: HonHaiPr_20:94:cf (dc:a2:66:20:94:cf)
      Address: HonHaiPr_20:94:cf (dc:a2:66:20:94:cf)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: IPv4 (0x0800)
```

Preamble/SFD: This indicates the starting of the frame and allows the sender and receiver to establish bit synchronization.
Dest/Src addr: Both are 6-byte field and contains the MAC address of the receiver/sender machine.
Length: It indicates the length of the entire Ethernet frame.

Data: This is the place where actual data is stored and both IP header and data is stored here in general.
CRC: This is used to detect any in-transit corruption of data.

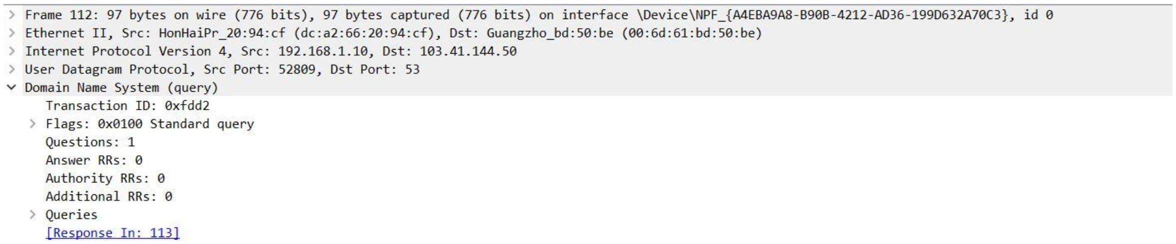
Others Protocols

1) STUN: Service Traversal Utilities for NAT



```
> Frame 308: 195 bytes on wire (1560 bits), 195 bytes captured (1560 bits) on interface \Device\NPF_{A4EBA9A8-B90B-4212-AD36-199D632A70C
> Ethernet II, Src: Guangzho_bd:50:be (00:6d:61:bd:50:be), Dst: HonHaiPr_20:94:cf (dc:a2:66:20:94:cf)
> Internet Protocol Version 4, Src: 52.139.181.155, Dst: 192.168.1.10
> User Datagram Protocol, Src Port: 3478, Dst Port: 42138
v Session Traversal Utilities for NAT
    [Request In: 300]
    [Time: 0.116440000 seconds]
  > Message Type: 0x0103 (Allocate Success Response)
    Message Length: 133
    Message Cookie: 2112a442
    Message Transaction ID: dab9a7e9f0a4a20a9ac04193
  > Attributes
```

2) DNS: Domain name system is a hierarchical and decentralized naming system for computers. It translates more readily memorized domain names to the numerical IP addresses.



```
> Frame 112: 97 bytes on wire (776 bits), 97 bytes captured (776 bits) on interface \Device\NPF_{A4EBA9A8-B90B-4212-AD36-199D632A70C3}, id 0
> Ethernet II, Src: HonHaiPr_20:94:cf (dc:a2:66:20:94:cf), Dst: Guangzho_bd:50:be (00:6d:61:bd:50:be)
> Internet Protocol Version 4, Src: 192.168.1.10, Dst: 103.41.144.50
> User Datagram Protocol, Src Port: 52809, Dst Port: 53
v Domain Name System (query)
    Transaction ID: 0xfdd2
  > Flags: 0x0100 Standard query
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
  > Queries
    [Response In: 113]
```

3) ARP: The Address Resolution Protocol is a communication protocol used for discovering the link-layer address, such as a MAC address, associated with a given internet layer address, typically an IPv4 address.
4) ICMPv6: is an integral part of IPv6 and performs error reporting and diagnostic functions (e.g., ping), and has a framework for extensions to implement future changes. ICMPv6 stands for Internet Control Message Protocol version 6.

Ans 2)

See Trace file name Ans1, voicecall, chat_q3.

1) Video Call: Skype uses UDP for video call feature while starting a video call and ending we see some TCP packets also but during video call UDP packets are predominant. Reason to use UDP:
   a) UDP allows IP multicast thus it is best suited for video call purposes.
   b) UDP is fast, it allows realtime communication without delay. Errors such as packets loss have very little impact on customer experience.
2) Chatting: TCP is used for message communication between two nodes. Because we can't afford packet loss while chatting thus TCP is used as it is reliable and accurate. It guarantees proper delivery of packets.

3) Voice call: It uses UDP for voice call feature. For similar reason mentioned for video calls.

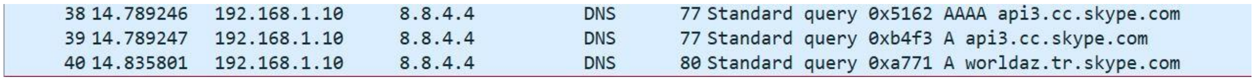
Ans-3)

Exchange of Messages.

Starting a video call :

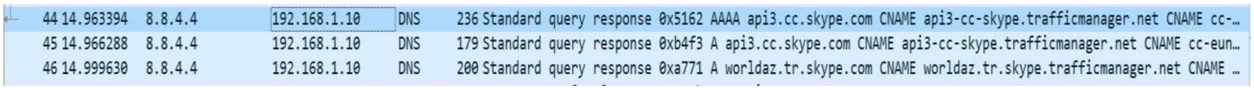
1) **DNS Query:** When we load the site then DNS querying is done by the browser. DNS query is a demand for information sent from user's computer to a DNS server to ask for the IP address associated with the domain name skype.com

Request:



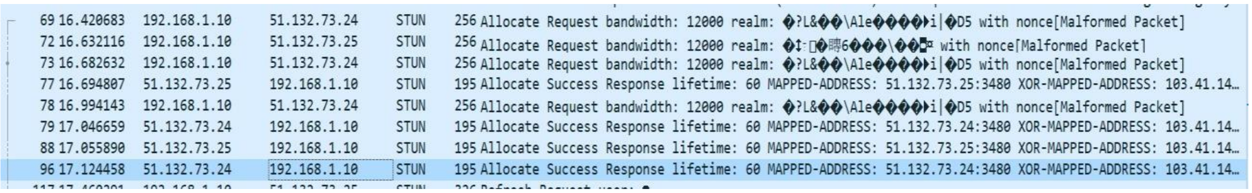
```
38 14.789246   192.168.1.10     8.8.4.4          DNS        77 Standard query 0x5162 AAAA api3.cc.skype.com
39 14.789247   192.168.1.10     8.8.4.4          DNS        77 Standard query 0xb4f3 A api3.cc.skype.com
40 14.835801   192.168.1.10     8.8.4.4          DNS        80 Standard query 0xa771 A worldaz.tr.skype.com
```

Response:



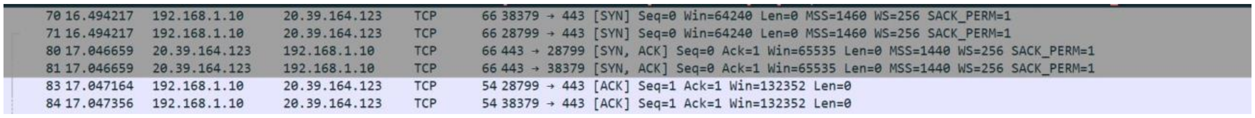
```
44 14.963394  8.8.4.4        192.168.1.10   DNS    236 Standard query response 0x5162 AAAA api3.cc.skype.com CNAME api3-cc-skype.trafficmanager.net CNAME cc-…
45 14.966288  8.8.4.4        192.168.1.10   DNS    179 Standard query response 0xb4f3 A api3.cc.skype.com CNAME api3-cc-skype.trafficmanager.net CNAME cc-eun…
46 14.999630  8.8.4.4        192.168.1.10   DNS    200 Standard query response 0xa771 A worldaz.tr.skype.com CNAME worldaz.tr.skype.trafficmanager.net CNAME …
```

2) **STUN Handshake**: Session Traversal Utilities for NAT
The protocol is used in several different network implementations, one of which is VoIP. STUN is used to resolve the public IP of a device running behind a NAT, to solve problems such as one-way audio during a phone call or phone registration issues when trying to register to a VoIP or an IP PBX residing on a different network.



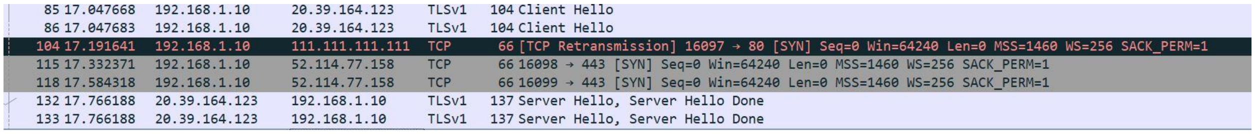
```
69 16.420683  192.168.1.10   51.132.73.24   STUN   256 Allocate Request bandwidth: 12000 realm: �?L&��\Ale����›i|�D5 with nonce[Malformed Packet]
72 16.632116  192.168.1.10   51.132.73.25   STUN   256 Allocate Request bandwidth: 12000 realm: �↕▯�時6���\��¤ with nonce[Malformed Packet]
73 16.682632  192.168.1.10   51.132.73.24   STUN   256 Allocate Request bandwidth: 12000 realm: �?L&��\Ale����›i|�D5 with nonce[Malformed Packet]
77 16.694807  51.132.73.25   192.168.1.10   STUN   195 Allocate Success Response lifetime: 60 MAPPED-ADDRESS: 51.132.73.25:3480 XOR-MAPPED-ADDRESS: 103.41.14…
78 16.994143  192.168.1.10   51.132.73.24   STUN   256 Allocate Request bandwidth: 12000 realm: �?L&��\Ale����›i|�D5 with nonce[Malformed Packet]
79 17.046659  51.132.73.24   192.168.1.10   STUN   195 Allocate Success Response lifetime: 60 MAPPED-ADDRESS: 51.132.73.24:3480 XOR-MAPPED-ADDRESS: 103.41.14…
88 17.055890  51.132.73.25   192.168.1.10   STUN   195 Allocate Success Response lifetime: 60 MAPPED-ADDRESS: 51.132.73.25:3480 XOR-MAPPED-ADDRESS: 103.41.14…
96 17.124458  51.132.73.24   192.168.1.10   STUN   195 Allocate Success Response lifetime: 60 MAPPED-ADDRESS: 51.132.73.24:3480 XOR-MAPPED-ADDRESS: 103.41.14…
```

3) **TCP Handshake:** It is a 3-way process between server and client.
Step-1: The client establishes a connection with the server. It sends the SYN segment to the host.
Step-2: The server responds to the client request with an SYN-ACK signal set.
Step-3: Client acknowledges server response and a connection is established.



```
70 16.494217  192.168.1.10    20.39.164.123   TCP     66 38379 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
71 16.494217  192.168.1.10    20.39.164.123   TCP     66 28799 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
80 17.046659  20.39.164.123   192.168.1.10    TCP     66 443 → 28799 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1440 WS=256 SACK_PERM=1
81 17.046659  20.39.164.123   192.168.1.10    TCP     66 443 → 38379 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1440 WS=256 SACK_PERM=1
83 17.047164  192.168.1.10    20.39.164.123   TCP     54 28799 → 443 [ACK] Seq=1 Ack=1 Win=132352 Len=0
84 17.047356  192.168.1.10    20.39.164.123   TCP     54 38379 → 443 [ACK] Seq=1 Ack=1 Win=132352 Len=0
```

4) **TLS Handshake:** Client sends a Client hello and server responds with server Hello and authentication key.



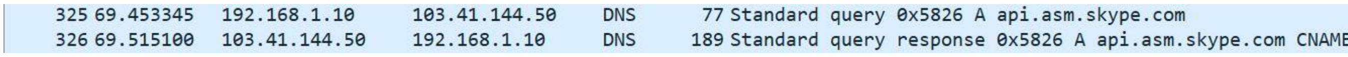
```
 85 17.047668  192.168.1.10    20.39.164.123     TLSv1   104 Client Hello
 86 17.047683  192.168.1.10    20.39.164.123     TLSv1   104 Client Hello
104 17.191641  192.168.1.10    111.111.111.111   TCP      66 [TCP Retransmission] 16097 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
115 17.332371  192.168.1.10    52.114.77.158     TCP      66 16098 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
118 17.584318  192.168.1.10    52.114.77.158     TCP      66 16099 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
132 17.766188  20.39.164.123   192.168.1.10      TLSv1   137 Server Hello, Server Hello Done
133 17.766188  20.39.164.123   192.168.1.10      TLSv1   137 Server Hello, Server Hello Done
```

5) **UDP Packets:** Then UDP packets are used for video calling.

Chat Box:

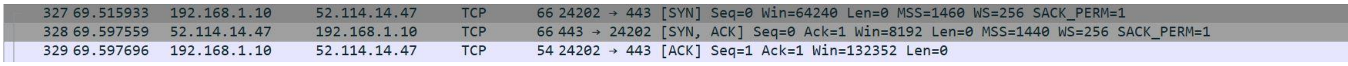
Sending text messages and video messages.

1) **DNS Query:**



```
325 69.453345   192.168.1.10     103.41.144.50    DNS      77 Standard query 0x5826 A api.asm.skype.com
326 69.515100   103.41.144.50    192.168.1.10     DNS     189 Standard query response 0x5826 A api.asm.skype.com CNAME
```

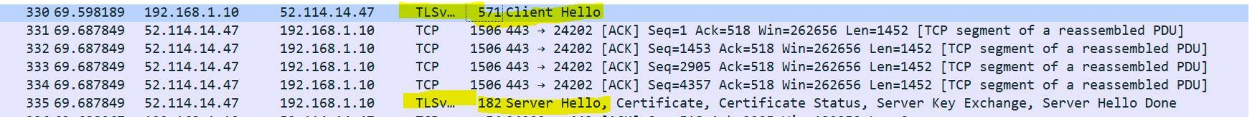
2) **3-Way TCP Handshake:**



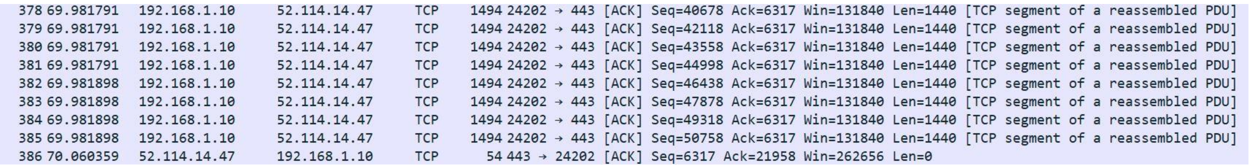
```
327 69.515933   192.168.1.10     52.114.14.47     TCP      66 24202 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
328 69.597559   52.114.14.47     192.168.1.10     TCP      66 443 → 24202 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1440 WS=256 SACK_PERM=1
329 69.597696   192.168.1.10     52.114.14.47     TCP      54 24202 → 443 [ACK] Seq=1 Ack=1 Win=132352 Len=0
```

3) **TLS Handshake:** (Yellow highlighted entries in below-attached image)

4) **TCP Packets** for **Video transfer:**



```
330 69.598189   192.168.1.10    52.114.14.47    TLSv…   571 Client Hello
331 69.687849   52.114.14.47    192.168.1.10    TCP    1506 443 → 24202 [ACK] Seq=1 Ack=518 Win=262656 Len=1452 [TCP segment of a reassembled PDU]
332 69.687849   52.114.14.47    192.168.1.10    TCP    1506 443 → 24202 [ACK] Seq=1453 Ack=518 Win=262656 Len=1452 [TCP segment of a reassembled PDU]
333 69.687849   52.114.14.47    192.168.1.10    TCP    1506 443 → 24202 [ACK] Seq=2905 Ack=518 Win=262656 Len=1452 [TCP segment of a reassembled PDU]
334 69.687849   52.114.14.47    192.168.1.10    TCP    1506 443 → 24202 [ACK] Seq=4357 Ack=518 Win=262656 Len=1452 [TCP segment of a reassembled PDU]
335 69.687849   52.114.14.47    192.168.1.10    TLSv…   182 Server Hello, Certificate, Certificate Status, Server Key Exchange, Server Hello Done
```



```
378 69.981791   192.168.1.10    52.114.14.47    TCP    1494 24202 → 443 [ACK] Seq=40678 Ack=6317 Win=131840 Len=1440 [TCP segment of a reassembled PDU]
379 69.981791   192.168.1.10    52.114.14.47    TCP    1494 24202 → 443 [ACK] Seq=42118 Ack=6317 Win=131840 Len=1440 [TCP segment of a reassembled PDU]
380 69.981791   192.168.1.10    52.114.14.47    TCP    1494 24202 → 443 [ACK] Seq=43558 Ack=6317 Win=131840 Len=1440 [TCP segment of a reassembled PDU]
381 69.981791   192.168.1.10    52.114.14.47    TCP    1494 24202 → 443 [ACK] Seq=44998 Ack=6317 Win=131840 Len=1440 [TCP segment of a reassembled PDU]
382 69.981898   192.168.1.10    52.114.14.47    TCP    1494 24202 → 443 [ACK] Seq=46438 Ack=6317 Win=131840 Len=1440 [TCP segment of a reassembled PDU]
383 69.981898   192.168.1.10    52.114.14.47    TCP    1494 24202 → 443 [ACK] Seq=47878 Ack=6317 Win=131840 Len=1440 [TCP segment of a reassembled PDU]
384 69.981898   192.168.1.10    52.114.14.47    TCP    1494 24202 → 443 [ACK] Seq=49318 Ack=6317 Win=131840 Len=1440 [TCP segment of a reassembled PDU]
385 69.981898   192.168.1.10    52.114.14.47    TCP    1494 24202 → 443 [ACK] Seq=50758 Ack=6317 Win=131840 Len=1440 [TCP segment of a reassembled PDU]
386 70.060359   52.114.14.47    192.168.1.10    TCP      54 443 → 24202 [ACK] Seq=6317 Ack=21958 Win=262656 Len=0
```

Note: Images attached above are for video file sent as a message sent on the skype. A similar exchange of packets occurred for text message exchange and explanation is same as given in video call message exchanges.

Ans-4)

Video Call

Time	Throughput	RTT	Packet Size	No. Packets lost	No of UDP & TCP packets	# response w.r.t. one request sent
9:00	9536	0.028	264 B	0	1011 & 369	0.342
16:00	12k	0.019	232 B	0	2587 & 564	0.312
21:00	10k	1/49	212 B	0	259 & 2424	0.297

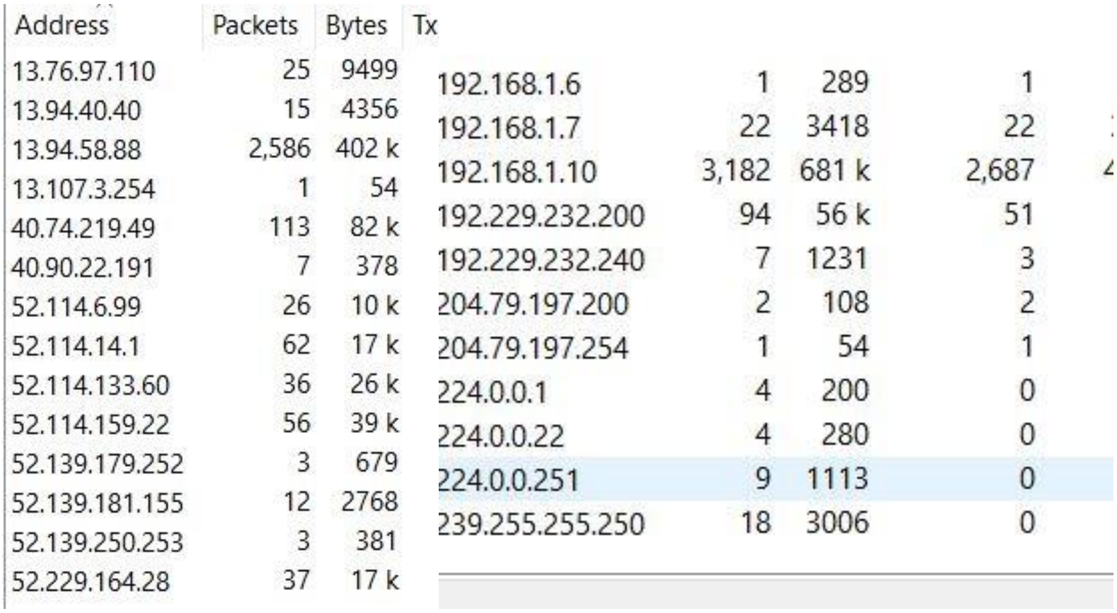
Chat Messages

Time	Throughput	RTT	Packet Size	No. Packets lost	No of UDP & TCP packets	# response w.r.t. one request sent
9:00	38k	0.024	913 B	0	28 & 3938	0.58
16:00	14k	0.043	625 B	0	31 & 925	0.706

21:00	12k	0.050	620 B	0	35 & 831	0.707

Ans-5)

Yes, the IP address of destination changes during different times of the day. This Skype is a global service provider thus it has multiple servers to balance the load, increase the reliability and better network distribution among its users. A server used in morning might be busy in afternoon so packet must go to other server.



Address	Packets	Bytes	Tx
13.76.97.110	25	9499	
13.94.40.40	15	4356	
13.94.58.88	2,586	402 k	
13.107.3.254	1	54	
40.74.219.49	113	82 k	
40.90.22.191	7	378	
52.114.6.99	26	10 k	
52.114.14.1	62	17 k	
52.114.133.60	36	26 k	
52.114.159.22	56	39 k	
52.139.179.252	3	679	
52.139.181.155	12	2768	
52.139.250.253	3	381	
52.229.164.28	37	17 k	
192.168.1.6	1	289	1
192.168.1.7	22	3418	22
192.168.1.10	3,182	681 k	2,687
192.229.232.200	94	56 k	51
192.229.232.240	7	1231	3
204.79.197.200	2	108	2
204.79.197.254	1	54	1
224.0.0.1	4	200	0
224.0.0.22	4	280	0
224.0.0.251	9	1113	0
239.255.255.250	18	3006	0