CS588 SYSTEM LAB ASSIGNMENT 2(Zoom Application)

Group 1 Members: Debarpan Jana(224101016) Madhurima Sen(224101034) Mayukh Das(224101036)

Trace File Link -
https://drive.google.com/drive/folders/165RPj21-pEJZy-W1ibsqs_pbpvAroLY7?usp=share_link

Q1.) List out all the protocols used by the application at different layers (only those which you can figure out from traces). Study and briefly describe their packet formats.

In our Zoom application from different packet traces at different places and time and networks, We get the following data:

Connection Establishment (TCP Handshaking)

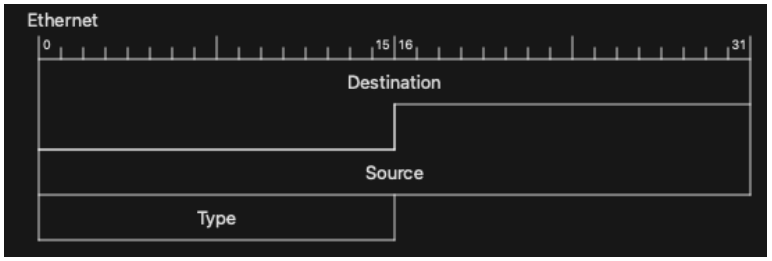
Layers	Protocols
Data Link Layer	Ethernet II
Network Layer	IPV4
Transport Layer	TCP
Session Layer(Security)	TLSv1,TLSv1.2,TLSv1.3

Data Transfer and Connection Termination Phase

Layers	Protocols
Data Link Layer	Ethernet II
Network Layer	IPV4,ICMP,WireGuard(For Layer 3 Security)
Transport Layer	TCP,UDP,QUIC
Session Layer(Security)	TLSv1,TLSv1.2,TLSv1.3

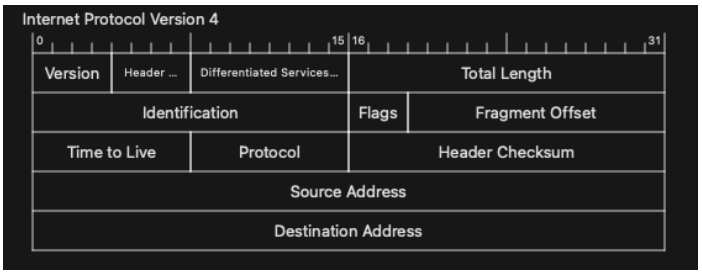
Protocols and their packet format present in some traces :

- ## Data Link Layer
  - ### Ethernet II
    - Ethernet II, also known as DIX Ethernet, is a standard for Ethernet networking. It was one of the original Ethernet standards and defined the format for Ethernet frames, including the structure of the frame header and trailer. Ethernet II uses a 48-bit MAC address and allows for a maximum frame size of 1518 bytes.

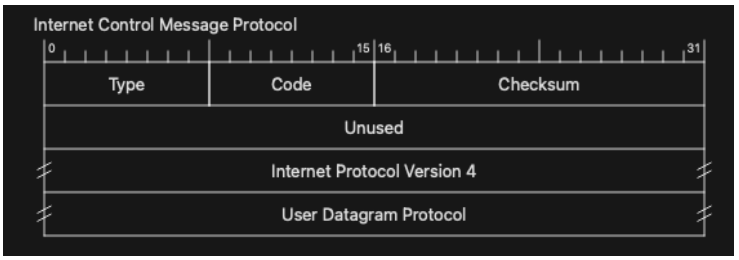


- ## Network Layer
  - ### IPv4

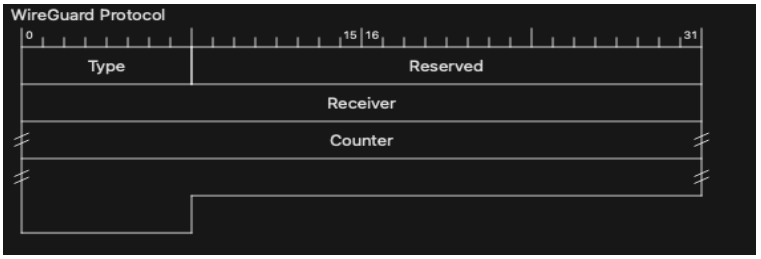
- IPv4 (Internet Protocol version 4) is a widely used protocol for transmitting data across the internet. It is the fourth version of the Internet Protocol (IP) and is the most widely deployed version of IP.An IPv4 address is a 32-bit numerical label that is assigned to each device connected to the internet, allowing data to be sent and received between devices.



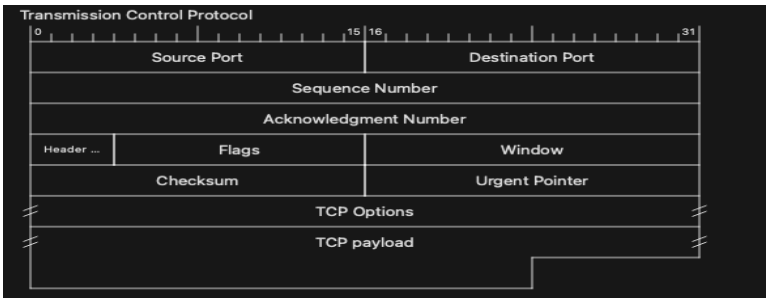
- **ICMP**
  - ICMP (Internet Control Message Protocol) is a network protocol that is used to send error messages or status messages between network devices. It is an integral part of the IP (Internet Protocol) suite and is used to diagnose and troubleshoot network issues.



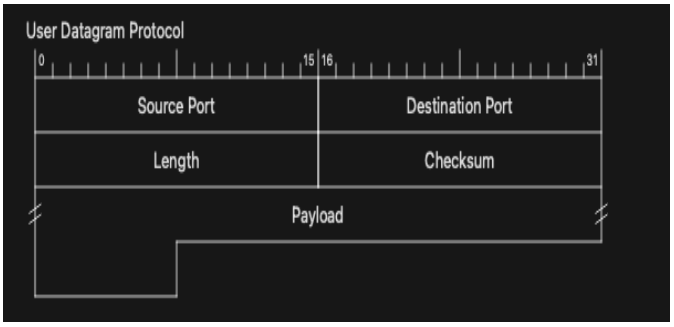
- **WireGuard**
  - WireGuard is a relatively new and modern VPN (Virtual Private Network) protocol that is designed to provide secure, fast, and simple VPN connections. WireGuard was designed with a focus on speed, security, and ease of use, and aims to be a more secure and efficient alternative to traditional VPN protocols such as PPTP, L2TP/IPsec, and OpenVPN.



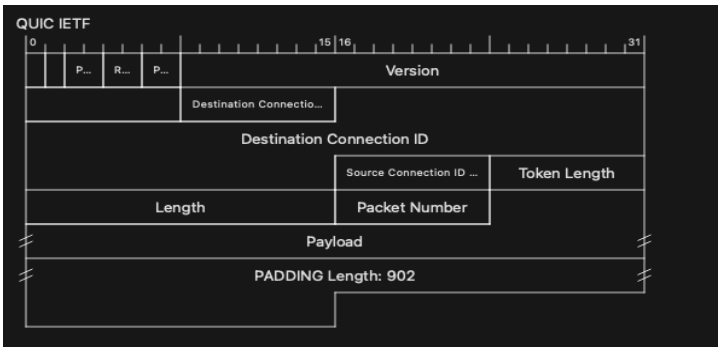
- **Transport Layer Protocols**
  - **TCP**
    - TCP operates at the transport layer of the IP stack and provides a reliable, stream-oriented service to applications. When two devices communicate using TCP, they establish a connection and exchange data by dividing the data into small packets called segments. Each segment is then transmitted individually and reassembled into its original form at the destination



- **UDP**
  - UDP (User Datagram Protocol) is a simple, unreliable, and connectionless network protocol that is part of the Internet Protocol (IP) suite. Unlike TCP, which is a reliable and connection-oriented protocol, UDP does not establish a dedicated connection between devices and does not guarantee the delivery or order of transmitted data.

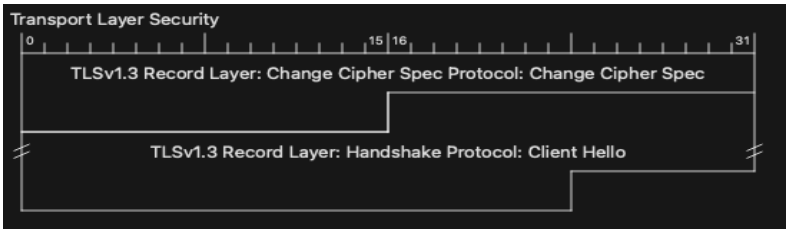


- **QUIC**
  - QUIC (Quick UDP Internet Connections) is a relatively new, experimental transport protocol that is designed to provide low latency and high-throughput connections for the internet. QUIC is built on top of UDP and aims to provide many of the benefits of TCP, such as reliability and flow control, while still retaining the low latency and high-throughput of UDP.



- **Session Layer Protocols**
  - **TLS**

- TLS (Transport Layer Security) is a widely used security protocol that provides encrypted communication between devices over the internet. TLS is the successor to the older SSL (Secure Sockets Layer) protocol and is used to protect sensitive information, such as passwords and credit card numbers, as it is transmitted between devices.



Q2.) Highlight and explain the observed values for various fields of the protocols. Example: Source or destination IP address and port number, Ethernet address, protocol number, etc.

We have taken into consideration a packet (**Application Data Packet**) for describing field values at different layers using different protocols:

- **Data Link Layer**
  - **Ethernet II**
    - Destination MAC address:- 66:16:b5:6b:8c:97
    - Source MAC address:- 14:98:77:3b:c7:78
    - Type: IPv4 (0x0800)
- **Network Layer**
  - **IPv4**
    - 0100 .... = Version: 4
    - .... 0101 = Header Length: 20 bytes (5)
    - Differentiated Services Field: 0x02 (DSCP: CS0, ECN: ECT(0))
      0000 00.. = Differentiated Services Codepoint: Default (0)
      .... ..10 = Explicit Congestion Notification: ECN-Capable Transport codepoint '10' (2)
    - Total Length: 535
    - Identification: 0x0000 (0)
    - 010. .... = Flags: 0x2, Don't fragment
      0... .... = Reserved bit: Not set
      .1.. .... = Don't fragment: Set
      ..0. .... = More fragments: Not set
    - ...0 0000 0000 0000 = Fragment Offset: 0
    - Time to Live: 64
    - Protocol: TCP (6)
    - Header Checksum: 0x7423 [validation disabled]
    - Source Address: 192.168.37.157
    - Destination Address: 170.114.52.4

- **Transport Layer**
  - **TCP**
    - Transmission Control Protocol, Src Port: 49804, Dst Port: 443, Seq: 1495, Ack: 1, Len: 483
    - Source Port: 49804
    - Destination Port: 443
    - [Stream index: 3]
    - [Conversation completeness: Incomplete (60)]
    - [TCP Segment Len: 483]
    - Sequence Number: 1495 (relative sequence number)
    - Sequence Number (raw): 1131176651
    - [Next Sequence Number: 1978 (relative sequence number)]
    - Acknowledgment Number: 1 (relative ack number)

        - Acknowledgment number (raw): 1061589390
        - 1000 .... = Header Length: 32 bytes (8)
        - Flags: 0x018 (PSH, ACK)
        - 000. .... = Reserved: Not set
        - ...0 .... = Accurate ECN: Not set
        - .... 0... .... = Congestion Window Reduced: Not set
        - .... .0.. .... = ECN-Echo: Not set
        - .... ..0. .... = Urgent: Not set
        - ....1 .... = Acknowledgment: Set
        - .... 1... = Push: Set
        - .... .0.. = Reset: Not set
        - ....0. = Syn: Not set
        - ....0 = Fin: Not set
        - [TCP Flags: ·····AP···]
        - Window: 2048
        - [Calculated window size: 2048]
        - [Window size scaling factor: -1 (unknown)]
        - Checksum: 0xb652 [unverified]
        - [Checksum Status: Unverified]
        - Urgent Pointer: 0
        - Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
        - TCP Option - No-Operation (NOP)
        - Kind: No-Operation (1)
        - TCP Option - No-Operation (NOP)
        - Kind: No-Operation (1)
        - TCP Option - Timestamps
        - Kind: Time Stamp Option (8)
        - Length: 10
        - Timestamp value: 143958106: TSval 143958106, TSecr 2945219453
        - Timestamp echo reply: 2945219453
        - [Timestamps]
        - [Time since first frame in this TCP stream: 0.000043000 seconds]
        - [Time since previous frame in this TCP stream: 0.000036000 seconds]
        - [SEQ/ACK analysis]
        - [Bytes in flight: 1977]
        - [Bytes sent since last PSH flag: 483]
        - TCP payload (483 bytes)

- **Session Layer**
    - **TLS**
        - Transport Layer Security
        - TLSv1.2 Record Layer: Application Data Protocol: Hypertext Transfer Protocol
        - Content Type: Application Data (23)
        - Version: TLS 1.2 (0x0303)
        - Length: 478
        - Encrypted Application Data: 2ab2a431b1c4fbe00b4d1ad128f33f25e9fc5549653c07fe5fdd43807c174d7ae8517d0e…
        - [Application Data Protocol: Hypertext Transfer Protocol]

We have taken some other packets which have common layers as above but differ in some layers in Transport or Network Layer

- **ICMP**
    - Internet Control Message Protocol
    - Type: 3 (Destination unreachable)
    - Code: 3 (Port unreachable)
    - Checksum: 0xff93 [correct]

- [Checksum Status: Good]
- Unused: 00000000
- Internet Protocol Version 4, Src: 206.247.158.43, Dst: 192.168.37.157
- User Datagram Protocol, Src Port: 8801, Dst Port: 56038

- **WireGuard**
  - Type: Transport Data (4)
  - Reserved: 000000
  - Receiver: 0x84816905
  - Counter: 1873780222437884126
  - Encrypted Packet

- **UDP**
  - User Datagram Protocol, Src Port: 53397, Dst Port: 8801
  - Source Port: 53397
  - Destination Port: 8801
  - Length: 117
  - Checksum: 0x86fe [unverified]
  - [Checksum Status: Unverified]
  - [Stream index: 10]
  - [Timestamps]
  - [Time since first frame: 7.889801000 seconds]
  - [Time since previous frame: 0.000151000 seconds]
  - UDP payload (109 bytes)

- **QUIC**
  - **QUIC IETF**
  - QUIC Connection information
  - [Packet Length: 1200]
  - 1... .... = Header Form: Long Header (1)
  - .1.. .... = Fixed Bit: True
  - ..00 .... = Packet Type: Initial (0)
  - .... 00.. = Reserved: 0
  - .... ..00 = Packet Number Length: 1 bytes (0)
  - Version: 1 (0x00000001)
  - Destination Connection ID Length: 8
  - Destination Connection ID: 6b0a97db9d0ce044
  - Source Connection ID Length: 0
  - Token Length: 0
  - Length: 1182
  - Packet Number: 0
  - Payload: 3dc8219fdaa4038ae5e0ad89f014ea3e91c1c0ff1d54c8447ca75dd50b0dba604916e051…
  - CRYPTO
  - PADDING Length: 902

Q3.) Explain the sequence of messages exchanged by the application for using the available functionalities in the application. For example: upload, download, play, pause, etc. Check whether there are any handshaking sequences in the application. Briefly explain the handshaking message sequence, if any.

In our assigned application through the traces from tcpdump and analysed it in Wireshark
We get the following information

Zoom uses the client-server model of communication where multiple clients are connected Through the server

As the application is first opened some hello packets are exchanged between client and server The Zoom server IPs are changing in between as the meeting was little longer

288	6.675518	192.168.37.157	206.247.158.43	TLSv1.3	583	Client Hello
295	7.113813	192.168.37.157	206.247.158.43	TLSv1.3	589	Change Cipher Spec, Client Hello
307	7.657198	192.168.37.157	206.247.158.43	TLSv1.3	583	Client Hello
895	10.203770	192.168.37.157	170.114.10.6	TLSv1.3	583	Client Hello
1026	10.670336	192.168.37.157	170.114.10.6	TLSv1.3	589	Change Cipher Spec, Client Hello
1183	11.215130	192.168.37.157	170.114.14.74	TLSv1.2	583	Client Hello
1284	11.590694	170.114.14.74	192.168.37.157	TLSv1.2	1454	Server Hello
9422	70.985814	192.168.37.157	3.13.213.3	TLSv1.2	583	Client Hello
9431	71.375587	3.13.213.3	192.168.37.157	TLSv1.2	1454	Server Hello

Zoom Server has multiple servers . These messages are end to end encrypted and uses TLS as Transport Layer Security for example Change Cipher Spec Protocol Zoom uses

Zoom also uses multiple ports apart from **80(http)** and **443(https)** which is present in several messages. However in our traces most of the packets are for ports 443 or in range 8801-8810.

Protocol	Ports	Source
TCP	80,443	All Zoom clients
TCP	443, 8801, 8802	All Zoom clients
UDP	3478, 3479, 8801 - 8810	All Zoom clients

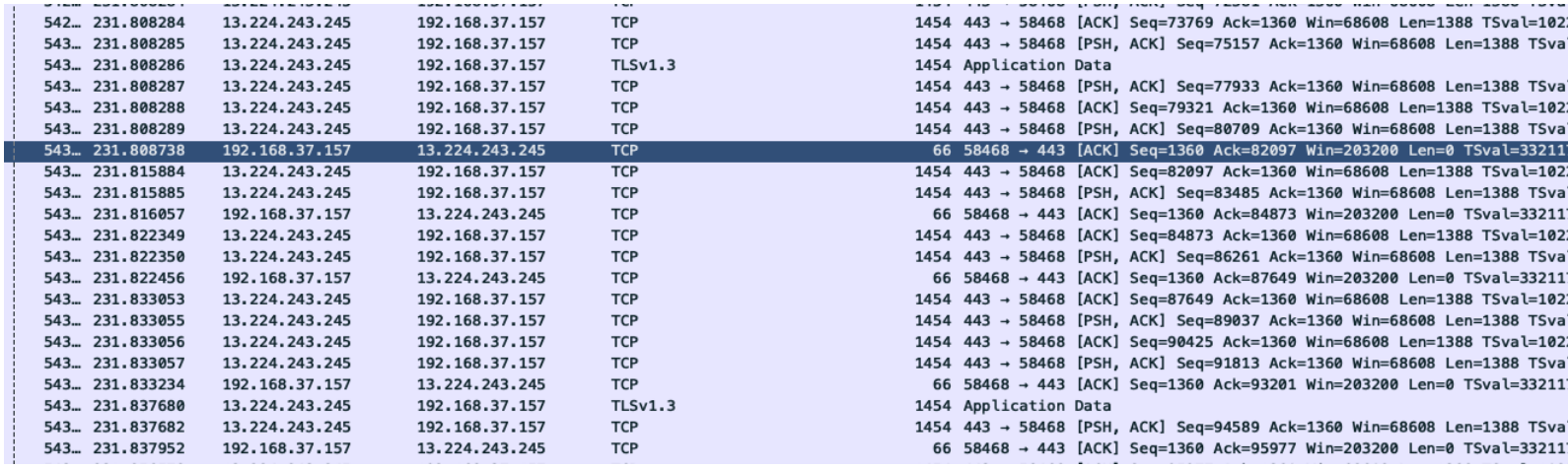
When the meeting is started **TCP-SYN** packets are transmitted

91	4.594049	192.168.37.157	144.195.5.254	TCP	78 49807 → 443 [SYN, ECE, CWR] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TSval=2296058080 TSecr=0 SACK_PERM
93	4.611700	192.168.37.157	144.195.11.254	TCP	78 49808 → 443 [SYN, ECE, CWR] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TSval=2566957588 TSecr=0 SACK_PERM
95	4.652931	192.168.37.157	206.247.158.44	TCP	78 49809 → 443 [SYN, ECE, CWR] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TSval=51282155 TSecr=0 SACK_PERM
97	4.664704	192.168.37.157	206.247.157.253	TCP	78 49810 → 443 [SYN, ECE, CWR] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TSval=4092098973 TSecr=0 SACK_PERM
98	4.959362	206.247.157.253	192.168.37.157	TCP	74 443 → 49810 [SYN, ACK, ECE] Seq=0 Ack=1 Win=43440 Len=0 MSS=1400 SACK_PERM TSval=1770289070 TSecr=4092098973 WS=4096
99	4.959363	144.195.5.254	192.168.37.157	TCP	74 443 → 49807 [SYN, ACK, ECE] Seq=0 Ack=1 Win=43440 Len=0 MSS=1400 SACK_PERM TSval=1002643116 TSecr=2296058080 WS=4096
100	4.959364	206.247.158.44	192.168.37.157	TCP	74 443 → 49809 [SYN, ACK, ECE] Seq=0 Ack=1 Win=43440 Len=0 MSS=1400 SACK_PERM TSval=1770139648 TSecr=51282155 WS=4096

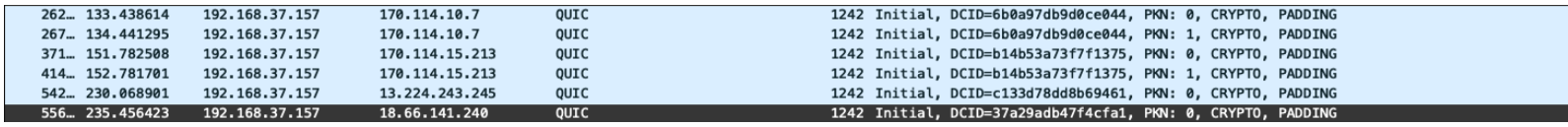
As we switch on audio and video TCP and UDP connections are used depending on network conditions UDP being a connectionless service is widely used in video streaming service.However in zoom live video is being transferred between clients TCP is also used as when network conditions are poor UDP is used.

609	9.405752	192.168.37.157	206.247.158.43	UDP	177 53397 → 8801 Len=135
610	9.405755	192.168.37.157	206.247.158.43	UDP	170 54860 → 8801 Len=128
611	9.448581	192.168.37.157	206.247.158.43	UDP	177 53397 → 8801 Len=135
612	9.466804	206.247.158.43	192.168.37.157	UDP	86 8801 → 56150 Len=44
613	9.478210	206.247.158.43	192.168.37.157	UDP	86 8801 → 54860 Len=44
614	9.478212	206.247.158.43	192.168.37.157	TCP	66 443 → 49815 [ACK] Seq=16571 Ack=48122 Win=65536 Len=0 TSval=1770088533 TSecr=3807274049
615	9.478212	206.247.158.43	192.168.37.157	UDP	56 8801 → 54860 Len=14
616	9.478212	206.247.158.43	192.168.37.157	TLSv1.3	141 Application Data
617	9.478213	206.247.158.43	192.168.37.157	UDP	86 8801 → 57396 Len=44
618	9.478213	206.247.158.43	192.168.37.157	TLSv1.3	1432 Application Data
619	9.478214	206.247.158.43	192.168.37.157	UDP	56 8801 → 57396 Len=14
620	9.478214	206.247.158.43	192.168.37.157	TCP	66 443 → 49815 [ACK] Seq=18012 Ack=48478 Win=65536 Len=0 TSval=1770088533 TSecr=3807274049
621	9.478215	206.247.158.43	192.168.37.157	UDP	56 8801 → 56150 Len=14
622	9.478215	206.247.158.43	192.168.37.157	TLSv1.3	133 Application Data
623	9.478215	206.247.158.43	192.168.37.157	UDP	114 8801 → 56150 Len=72

Zoom also has chat service and reactions which needs quicker actions and these messages are send as soon as possible without waiting for the transmission buffer to be filled so PSH flag is also set and in some packets advanced and relatively new protocol like QUIC is used



542…	231.808284	13.224.243.245	192.168.37.157	TCP	1454 443 → 58468 [ACK] Seq=73769 Ack=1360 Win=68608 Len=1388 TSval=102
543…	231.808285	13.224.243.245	192.168.37.157	TCP	1454 443 → 58468 [PSH, ACK] Seq=75157 Ack=1360 Win=68608 Len=1388 TSva
543…	231.808286	13.224.243.245	192.168.37.157	TLSv1.3	1454 Application Data
543…	231.808287	13.224.243.245	192.168.37.157	TCP	1454 443 → 58468 [PSH, ACK] Seq=77933 Ack=1360 Win=68608 Len=1388 TSva
543…	231.808288	13.224.243.245	192.168.37.157	TCP	1454 443 → 58468 [ACK] Seq=79321 Ack=1360 Win=68608 Len=1388 TSval=102
543…	231.808289	13.224.243.245	192.168.37.157	TCP	1454 443 → 58468 [PSH, ACK] Seq=80709 Ack=1360 Win=68608 Len=1388 TSva
543…	231.808738	192.168.37.157	13.224.243.245	TCP	66 58468 → 443 [ACK] Seq=1360 Ack=82097 Win=203200 Len=0 TSval=33211
543…	231.815884	13.224.243.245	192.168.37.157	TCP	1454 443 → 58468 [ACK] Seq=82097 Ack=1360 Win=68608 Len=1388 TSval=102
543…	231.815885	13.224.243.245	192.168.37.157	TCP	1454 443 → 58468 [PSH, ACK] Seq=83485 Ack=1360 Win=68608 Len=1388 TSva
543…	231.816057	192.168.37.157	13.224.243.245	TCP	66 58468 → 443 [ACK] Seq=1360 Ack=84873 Win=203200 Len=0 TSval=33211
543…	231.822349	13.224.243.245	192.168.37.157	TCP	1454 443 → 58468 [ACK] Seq=84873 Ack=1360 Win=68608 Len=1388 TSval=102
543…	231.822350	13.224.243.245	192.168.37.157	TCP	1454 443 → 58468 [PSH, ACK] Seq=86261 Ack=1360 Win=68608 Len=1388 TSva
543…	231.822456	192.168.37.157	13.224.243.245	TCP	66 58468 → 443 [ACK] Seq=1360 Ack=87649 Win=203200 Len=0 TSval=33211
543…	231.833053	13.224.243.245	192.168.37.157	TCP	1454 443 → 58468 [ACK] Seq=87649 Ack=1360 Win=68608 Len=1388 TSval=102
543…	231.833055	13.224.243.245	192.168.37.157	TCP	1454 443 → 58468 [PSH, ACK] Seq=89037 Ack=1360 Win=68608 Len=1388 TSva
543…	231.833056	13.224.243.245	192.168.37.157	TCP	1454 443 → 58468 [ACK] Seq=90425 Ack=1360 Win=68608 Len=1388 TSval=102
543…	231.833057	13.224.243.245	192.168.37.157	TCP	1454 443 → 58468 [PSH, ACK] Seq=91813 Ack=1360 Win=68608 Len=1388 TSva
543…	231.833234	192.168.37.157	13.224.243.245	TCP	66 58468 → 443 [ACK] Seq=1360 Ack=93201 Win=203200 Len=0 TSval=33211
543…	231.837680	13.224.243.245	192.168.37.157	TLSv1.3	1454 Application Data
543…	231.837682	13.224.243.245	192.168.37.157	TCP	1454 443 → 58468 [PSH, ACK] Seq=94589 Ack=1360 Win=68608 Len=1388 TSva
543…	231.837952	192.168.37.157	13.224.243.245	TCP	66 58468 → 443 [ACK] Seq=1360 Ack=95977 Win=203200 Len=0 TSval=33211



262…	133.438614	192.168.37.157	170.114.10.7	QUIC	1242 Initial, DCID=6b0a97db9d0ce044, PKN: 0, CRYPTO, PADDING
267…	134.441295	192.168.37.157	170.114.10.7	QUIC	1242 Initial, DCID=6b0a97db9d0ce044, PKN: 1, CRYPTO, PADDING
371…	151.782508	192.168.37.157	170.114.15.213	QUIC	1242 Initial, DCID=b14b53a73f7f1375, PKN: 0, CRYPTO, PADDING
414…	152.781701	192.168.37.157	170.114.15.213	QUIC	1242 Initial, DCID=b14b53a73f7f1375, PKN: 1, CRYPTO, PADDING
542…	230.068901	192.168.37.157	13.224.243.245	QUIC	1242 Initial, DCID=c133d78dd8b69461, PKN: 0, CRYPTO, PADDING
556…	235.456423	192.168.37.157	18.66.141.240	QUIC	1242 Initial, DCID=37a29adb47f4cfa1, PKN: 0, CRYPTO, PADDING

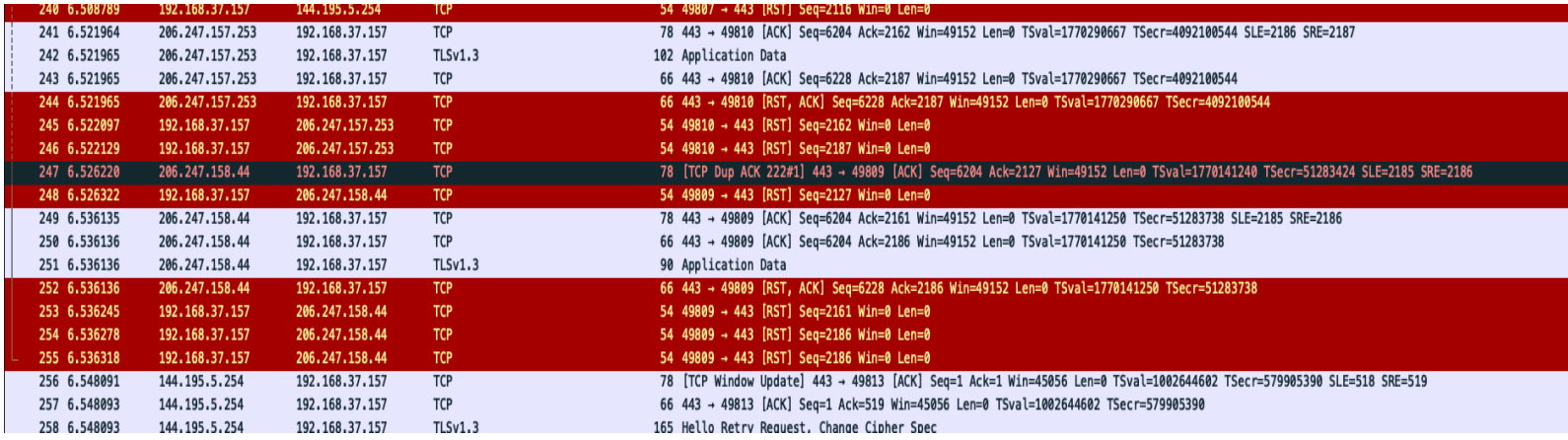
When the meeting is ended **TCP-FIN** packets are transmitted

619…	250.086236	192.168.37.157	18.66.141.240	TCP	66 58474 → 443 [FIN, ACK] Seq=1196 Ack=12736 Win=131072 Len=0 TSval=2669540953 TSecr=3280640255
619…	250.087683	192.168.37.157	18.66.141.240	TCP	66 58472 → 443 [FIN, ACK] Seq=1196 Ack=10550 Win=131072 Len=0 TSval=2337577980 TSecr=4058098086
619…	250.087725	192.168.37.157	18.66.141.240	TCP	66 58471 → 443 [FIN, ACK] Seq=1196 Ack=18310 Win=131072 Len=0 TSval=3494885651 TSecr=543919025
619…	250.087745	192.168.37.157	18.66.141.240	TCP	66 58475 → 443 [FIN, ACK] Seq=1786 Ack=666638 Win=804224 Len=0 TSval=901553802 TSecr=543920335
619…	250.087768	192.168.37.157	18.66.141.240	TCP	66 58473 → 443 [FIN, ACK] Seq=1196 Ack=138492 Win=287872 Len=0 TSval=778353563 TSecr=1679436136
619…	250.087792	192.168.37.157	18.66.141.240	TCP	66 58470 → 443 [FIN, ACK] Seq=3556 Ack=4748123 Win=2852928 Len=0 TSval=2840171654 TSecr=1847451060

When there are some connection issues, sometimes ICMP packets are also sent.

558	9.200478	192.168.37.157	206.247.158.43	ICMP	70 Destination unreachable (Port unreachable)
559	9.200496	192.168.37.157	206.247.158.43	ICMP	70 Destination unreachable (Port unreachable)
568	9.212146	192.168.37.157	206.247.158.43	ICMP	70 Destination unreachable (Port unreachable)
570	9.212206	192.168.37.157	206.247.158.43	ICMP	70 Destination unreachable (Port unreachable)
577	9.249144	192.168.37.157	206.247.158.43	ICMP	70 Destination unreachable (Port unreachable)
578	9.249179	192.168.37.157	206.247.158.43	ICMP	70 Destination unreachable (Port unreachable)

As the network traffic is not that much congested **TCP-DupACK** or **TCP-Retransmission** is not transmitted very often but visible in some traces



240	6.508789	192.168.37.157	144.195.5.254	TCP	54 49807 → 443 [RST] Seq=2116 Win=0 Len=0
241	6.521964	206.247.157.253	192.168.37.157	TCP	78 443 → 49810 [ACK] Seq=6204 Ack=2162 Win=49152 Len=0 TSval=1770290667 TSecr=4092100544 SLE=2186 SRE=2187
242	6.521965	206.247.157.253	192.168.37.157	TLSv1.3	102 Application Data
243	6.521965	206.247.157.253	192.168.37.157	TCP	66 443 → 49810 [ACK] Seq=6228 Ack=2187 Win=49152 Len=0 TSval=1770290667 TSecr=4092100544
244	6.521965	206.247.157.253	192.168.37.157	TCP	66 443 → 49810 [RST, ACK] Seq=6228 Ack=2187 Win=49152 Len=0 TSval=1770290667 TSecr=4092100544
245	6.522097	192.168.37.157	206.247.157.253	TCP	54 49810 → 443 [RST] Seq=2162 Win=0 Len=0
246	6.522129	192.168.37.157	206.247.157.253	TCP	54 49810 → 443 [RST] Seq=2187 Win=0 Len=0
247	6.526220	206.247.158.44	192.168.37.157	TCP	78 [TCP Dup ACK 222#1] 443 → 49809 [ACK] Seq=6204 Ack=2127 Win=49152 Len=0 TSval=1770141240 TSecr=51283424 SLE=2185 SRE=2186
248	6.526322	192.168.37.157	206.247.158.44	TCP	54 49809 → 443 [RST] Seq=2127 Win=0 Len=0
249	6.536135	206.247.158.44	192.168.37.157	TCP	78 443 → 49809 [ACK] Seq=6204 Ack=2161 Win=49152 Len=0 TSval=1770141250 TSecr=51283738 SLE=2185 SRE=2186
250	6.536136	206.247.158.44	192.168.37.157	TCP	66 443 → 49809 [ACK] Seq=6204 Ack=2186 Win=49152 Len=0 TSval=1770141250 TSecr=51283738
251	6.536136	206.247.158.44	192.168.37.157	TLSv1.3	90 Application Data
252	6.536136	206.247.158.44	192.168.37.157	TCP	66 443 → 49809 [RST, ACK] Seq=6228 Ack=2186 Win=49152 Len=0 TSval=1770141250 TSecr=51283738
253	6.536245	192.168.37.157	206.247.158.44	TCP	54 49809 → 443 [RST] Seq=2161 Win=0 Len=0
254	6.536278	192.168.37.157	206.247.158.44	TCP	54 49809 → 443 [RST] Seq=2186 Win=0 Len=0
255	6.536318	192.168.37.157	206.247.158.44	TCP	54 49809 → 443 [RST] Seq=2186 Win=0 Len=0
256	6.548091	144.195.5.254	192.168.37.157	TCP	78 [TCP Window Update] 443 → 49813 [ACK] Seq=1 Ack=1 Win=45056 Len=0 TSval=1002644602 TSecr=579905390 SLE=518 SRE=519
257	6.548093	144.195.5.254	192.168.37.157	TCP	66 443 → 49813 [ACK] Seq=1 Ack=519 Win=45056 Len=0 TSval=1002644602 TSecr=579905390
258	6.548093	144.195.5.254	192.168.37.157	TLSv1.3	165 Hello Retry Request, Change Cipher Spec

Q4). Explain how the particular protocol(s) used by the application is relevant for functioning of the application.

Zoom uses

1. Ethernet II because this is a widely used Data link layer protocol which ensures data is transmitted and received reliably over an Local area network.
2. Uses IPV4 because of its wide availability to any OS and device.
3. Uses ICMP as zoom needs a reliable network connection for smooth video conferencing. ICMP helps to check the quality of the network, it sees if there are any potential network issues like packet loss or latency and also helps to resolve these issues by asking users to change their network connections or troubleshoot.
4. Uses UDP as UDP is a fast,efficient protocol for video and audio streaming related applications. It has less overhead compared to TCP, and has low latency. It can be used over a wide range of network configurations.
5. uses QUIC as QUIC is both reliable and fast. This protocol is built upon UDP. It ensures low latency by reducing connection establishment time.
6. uses TCP for transmission of control messages, file transfer, communication between its servers and backend services as TCP is a highly reliable protocol.
7. Uses wireguard for providing secure and encrypted communication between users.
8. Uses TLSv1,TLSv1.2,TLSv1.3 for providing secure connection. It uses different versions of TLS protocols to make sure that secure connection is available to every user irrespective of their used OS and devices

Q5.) Calculate the following statistics from your traces while performing experiments at different time of the day: Throughput, RTT, Packet size, Number of packets lost, Number of UDP & TCP packets, Number of responses received with respect to one request sent. Report the observed values in your answer, preferably using tables.

	Different Lan	Different network	Lab	Power Cut	Same Lan
Throughput (bytes/sec)	173 k	204 k	42 k	116 k	54 k
RTT (milliseconds)	4.521	4.197	8.087	6.386	10.773
Packet Size (bytes)	784	860	346	746	591
No of Packets Lost	535	67	106	20	3
No of UDP Packets	95762	49296	7893	32984	4925
No of TCP Packets	24719	13311	11683	22108	3997
No of Responses per one request sent	0.848	1.104	1.523	0.623	0.918

Q6.) Check whether the whole content is being sent from same location/source. List out the IP addresses of content providers if multiple sources exist, and explain the reason behind this.

Zoom is sending content from multiple locations and ip addresses. Some of the ip addresses used by zoom are

Different Lan call

170.114.52.4 ,144.195.11.253 ,206.247.148.213 ,144.195.10.253 ,206.247.149.213 ,206.247.141.14 170.114.10.7 ,170.114.14.62 ,192.168.0.101 ,170.114.15.98 ,170.114.3.177 ,3.20.74.61 ,170.114.14.69 170.114.3.168 ,52.15.165.149 ,3.133.0.179

Different network call

170.114.52.4 ,206.247.157.253 ,144.195.5.254 ,206.247.158.44 ,144.195.11.254 ,206.247.158.43 ,170.114.10.6 170.114.14.74 ,192.168.37.157 ,3.13.213.3 ,170.114.10.7 ,170.114.15.213 ,134.224.128.155 ,134.224.251.220

134.224.169.179 ,134.224.158.6 ,134.224.66.185 ,170.114.12.3 ,134.224.157.85 ,170.114.35.162
134.224.175.93 13.59.13.71 ,170.114.52.22 ,13.224.243.245 ,18.66.141.240
Call in lab
170.114.52.4 ,144.195.4.253 ,144.195.5.253 ,206.247.79.253 ,206.247.78.253 ,144.195.9.85 ,170.114.10.7
3.20.74.61 ,172.16.116.136 ,170.114.15.213 ,13.59.13.71
Call during Power Cut
170.114.52.4 ,170.114.15.102 ,192.168.0.101 ,170.114.14.69 ,170.114.15.169 ,170.114.3.163 ,3.80.20.174 144.195.83.213 ,144.195.82.213 ,206.247.10.253 ,206.247.11.253 ,144.195.82.157 ,170.114.10.6 ,52.15.165.149 13.59.13.71
Call in Same lan
170.114.52.4 ,170.114.15.105 ,192.168.0.101 ,170.114.14.72 ,170.114.15.93 ,170.114.10.7 ,3.80.20.174
170.114.3.167 ,144.195.84.213 ,144.195.85.213 ,206.247.72.253 ,206.247.73.253 ,206.247.77.210

Reasons are following

1. Zoom is using multiple IP dresses to distribute its services across different locations, so that it will have low latency irrespective of the user's location.
2. Zoom uses multiple IP dresses for load balancing, so that it can provide service to users and be responsive in the time of high demand or if some IPs are down due to network issues.
3. If anyone wants to attack then s/he has to target all the IP addresses simultaneously. Hence using multiple IP addresses increases the security of zoom.

—----------(*)—----------