CS588: Computer System Lab

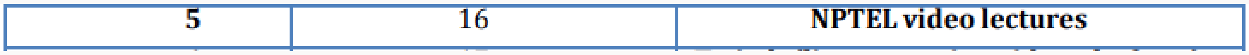
(January-May2023)

Assignment–2: Network Protocol Analysis Using Wireshark

Question:

Wire shark is a free and open source packet sniffer. And network protocol analyser tool. It helps to capture network packets and understand the structure of different networking protocols. Instructions:

- Install Wireshark (download from www.wireshark.org), and learn how to capture packets and filter the required content.
- A specific application is assigned to the group (refer to Table 1 below). Each group needs to perform various activities according to functionalities available in the assigned application and collect the traces for the application using Wireshark. Application-specific activities, if any, are mentioned in the table.
- You should carry out your experiments across different network conditions including different time(s) of the day and locations (e.g., lab or hostel, etc.).
- It is advisable to provide only trace-based descriptions while answering the questions. While answering, provide snapshots of the traces in the report and highlight the content as and when required.
- If something is missing/incorrect in a problem description, clearly mention the assumption with your answer. Be precise with your answers; there is no credit for being unnecessarily verbose (may award you negative marks for the same). Unless specified otherwise, do not describe the tool or application or protocol in general.



5	16	NPTEL video lectures

By

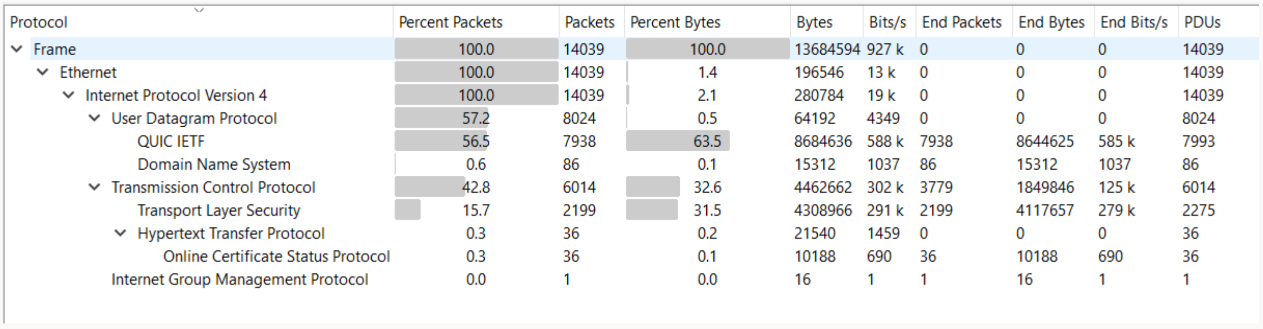
Kaja Gnana Prakash(224101027)

Tejas Chandra Karredula (224101052)

Ashish Kumar Pal(224101009)

1. **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.**

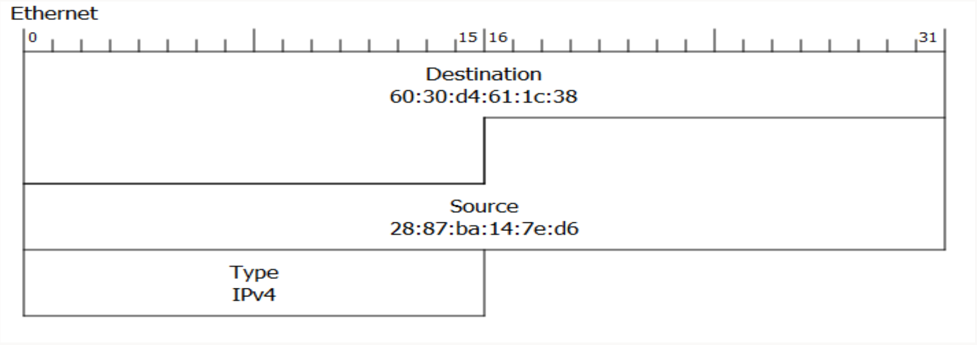
Ans: The protocols used by the application at different layer from the trace:



Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s	PDUs
Frame	100.0	14039	100.0	13684594	927 k	0	0	0	14039
Ethernet	100.0	14039	1.4	196546	13 k	0	0	0	14039
Internet Protocol Version 4	100.0	14039	2.1	280784	19 k	0	0	0	14039
User Datagram Protocol	57.2	8024	0.5	64192	4349	0	0	0	8024
QUIC IETF	56.5	7938	63.5	8684636	588 k	7938	8644625	585 k	7993
Domain Name System	0.6	86	0.1	15312	1037	86	15312	1037	86
Transmission Control Protocol	42.8	6014	32.6	4462662	302 k	3779	1849846	125 k	6014
Transport Layer Security	15.7	2199	31.5	4308966	291 k	2199	4117657	279 k	2275
Hypertext Transfer Protocol	0.3	36	0.2	21540	1459	0	0	0	36
Online Certificate Status Protocol	0.3	36	0.1	10188	690	36	10188	690	36
Internet Group Management Protocol	0.0	1	0.0	16	1	1	16	1	1

Protocols Used in the Trace: **Ethernet**, **IPV4**, UDP, **QUIC**, **TCP**, HTTP,IGMP

<u>Data Link Layer:</u> Ethernet Format



Ethernet

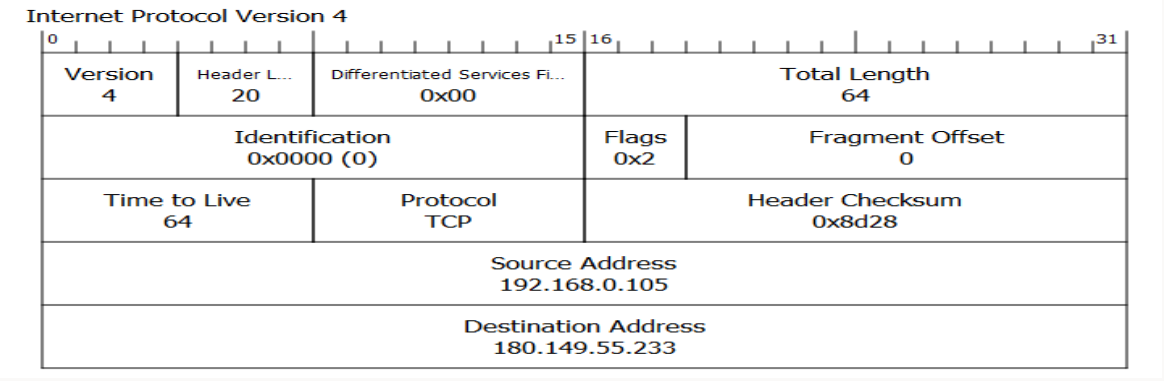
Destination
60:30:d4:61:1c:38

Source
28:87:ba:14:7e:d6

Type
IPv4

(Image 1.1)

<u>Network Layer:</u> IP Packet Format



Internet Protocol Version 4

Version
4

Header L...
20

Differentiated Services Fi...
0x00

Total Length
64

Identification
0x0000 (0)

Flags
0x2

Fragment Offset
0

Time to Live
64

Protocol
TCP

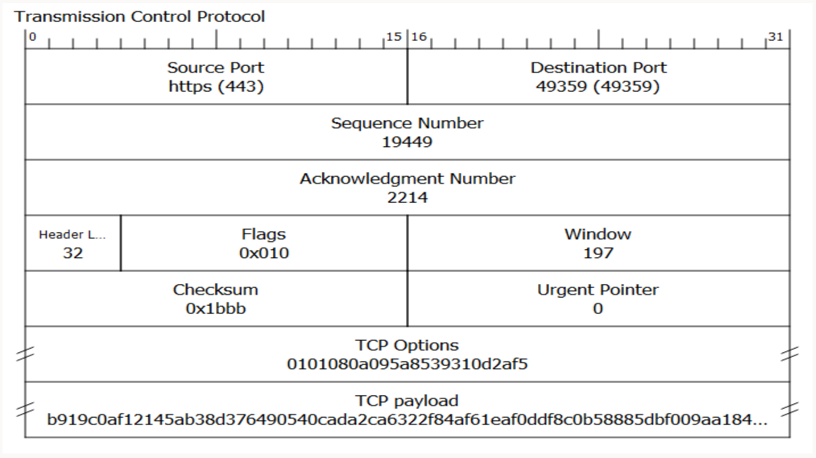
Header Checksum
0x8d28

Source Address
192.168.0.105

Destination Address
180.149.55.233

(Image 1.2)

Transport Layer: TCP Format



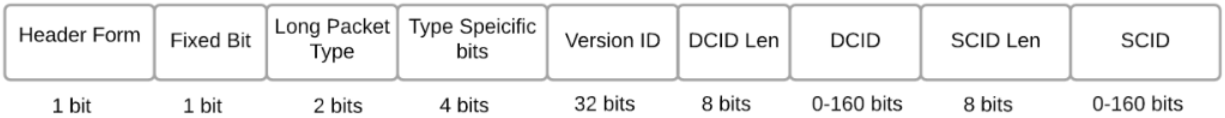
Transmission Control Protocol

Source Port https (443)		Destination Port 49359 (49359)
Sequence Number 19449		
Acknowledgment Number 2214		
Header L... 32	Flags 0x010	Window 197
Checksum 0x1bbb		Urgent Pointer 0
TCP Options 0101080a095a8539310d2af5		
TCP payload b919c0af12145ab38d376490540cada2ca6322f84af61eaf0ddf8c0b58885dbf009aa184...		

(Image 1.3)

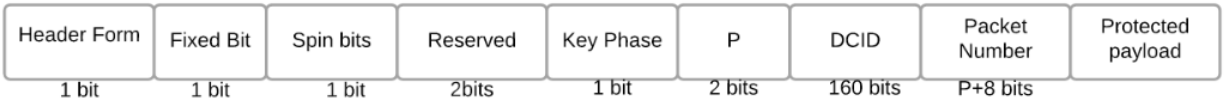
QUIC Format:

Long Header



Header Form	Fixed Bit	Long Packet Type	Type Speicific bits	Version ID	DCID Len	DCID	SCID Len	SCID
1 bit	1 bit	2 bits	4 bits	32 bits	8 bits	0-160 bits	8 bits	0-160 bits

Short Header



Header Form	Fixed Bit	Spin bits	Reserved	Key Phase	P	DCID	Packet Number	Protected payload
1 bit	1 bit	1 bit	2bits	1 bit	2 bits	160 bits	P+8 bits	

(Image 1.4)

2. **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.**

Ans:

Ethernet Field Values: As shown in **(image 1.1)**

Destination Address: Contains 48 bit MAC address of destination youtube server.

Source Address: Contains 48 bit MAC address of client i..e, here(local host). Ie., MAC address of the host machine.

IPV4 field Values:

As shown in **(image 1.2)**

Source Address: Contains 32 bit IP address of the host/client machine i.e, 192.168.0.105

Destination Address: Contains 32 bit IP address of the destination (here one of the youtuber server) ie., 180.149.55.233

Protocol: The protocol used for the packet picked is TCP.

TTL: How many hops that the current packet can do (here it is 64).

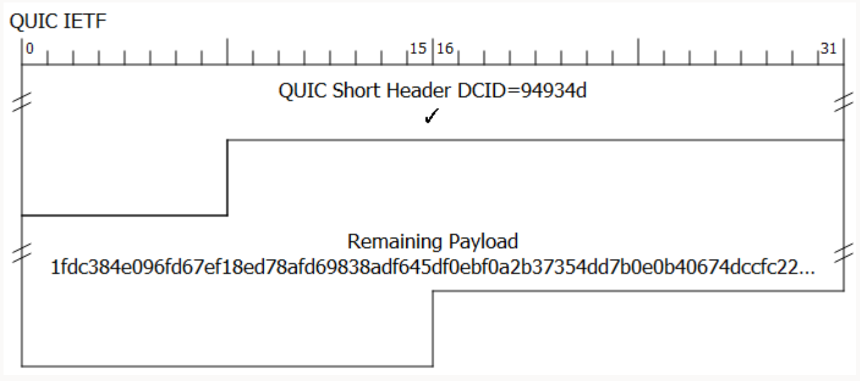
TCP Field Values:

As shown in **(image 1.3)** above:

Destination Port: The port used at client side is 49358(a random socket which is free on the host side).

Source Port: The port with destination communicates with the client here 443(https) port.

QUIC Field Values:

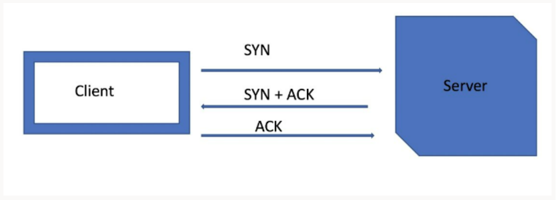

QUIC IETF
0
15
16
31
QUIC Short Header DCID=94934d
✓
Remaining Payload
1fdc384e096fd67ef18ed78afd69838adf645df0ebf0a2b37354dd7b0e0b40674dccfc22...

DCID: Destination Connection ID is 94934d. Here QUIC uses a short header it means the connection has been established.QUIC has two different types of headers. The long header is used prior to the connection establishment. The short header is used after the first connection is established.

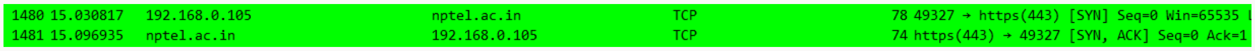
3. **Explain the sequence of messages exchanged by the application for using the available functionalities in the application. For example: upload, download, play, pause, etc. Check whether there are any handshaking sequences in the application. Briefly explain the handshaking message sequence, if any**

Ans:

Three way Handshake:

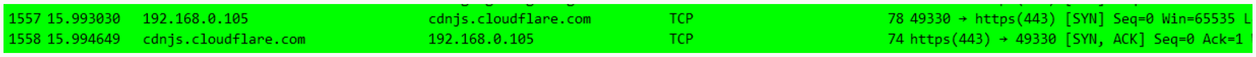


1. First there will be a three way handshake between the client and nptel.ac.in site as shown below. The client sends SYN=1 to the server and the server responds with SYN+ACK. After the connection is established data exchange can start.



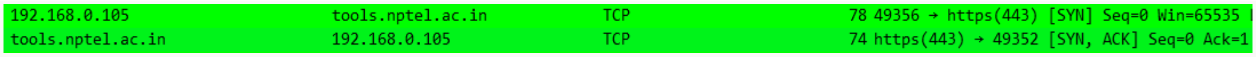
```
1480 15.030817   192.168.0.105          nptel.ac.in          TCP          78 49327 → https(443) [SYN] Seq=0 Win=65535 L
1481 15.096935   nptel.ac.in            192.168.0.105        TCP          74 https(443) → 49327 [SYN, ACK] Seq=0 Ack=1
```

2. Afterwards there will be a three way transfer of message or handshake between cdnjs.cloudflare which is a content delivery network.



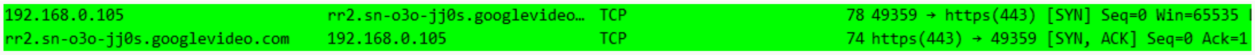
```
1557 15.993030   192.168.0.105          cdnjs.cloudflare.com   TCP          78 49330 → https(443) [SYN] Seq=0 Win=65535 L
1558 15.994649   cdnjs.cloudflare.com   192.168.0.105          TCP          74 https(443) → 49330 [SYN, ACK] Seq=0 Ack=1
```

3. Then we communicate with tools.nptel.ac.in to open the youtube video link.



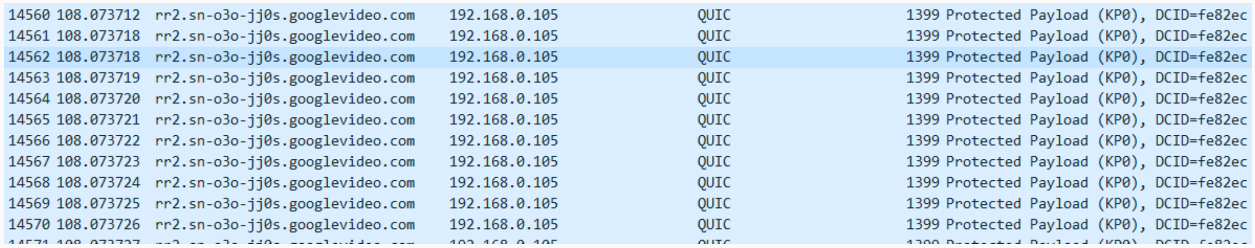
```
192.168.0.105            tools.nptel.ac.in        TCP          78 49356 → https(443) [SYN] Seq=0 Win=65535
tools.nptel.ac.in        192.168.0.105            TCP          74 https(443) → 49352 [SYN, ACK] Seq=0 Ack=1
```

4. After we start **playing** the youtube video, the client makes a three way handshake between client and you tube server(here rr2.sn-30..).



```
192.168.0.105                    rr2.sn-o3o-jj0s.googlevideo…  TCP          78 49359 → https(443) [SYN] Seq=0 Win=65535
rr2.sn-o3o-jj0s.googlevideo.com  192.168.0.105                 TCP          74 https(443) → 49359 [SYN, ACK] Seq=0 Ack=1
```

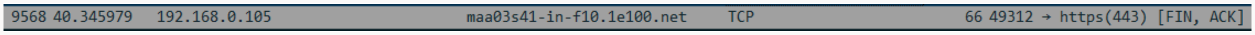
5. Transmission of data from youtube server(here rr2.sn-030..) to Client. Below is the snip of transmission of message or payload from the server to client.



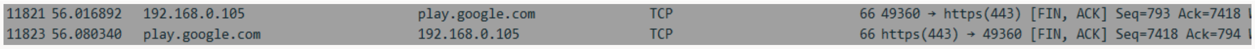
```
14560 108.073712  rr2.sn-o3o-jj0s.googlevideo.com   192.168.0.105      QUIC       1399 Protected Payload (KP0), DCID=fe82ec
14561 108.073718  rr2.sn-o3o-jj0s.googlevideo.com   192.168.0.105      QUIC       1399 Protected Payload (KP0), DCID=fe82ec
14562 108.073718  rr2.sn-o3o-jj0s.googlevideo.com   192.168.0.105      QUIC       1399 Protected Payload (KP0), DCID=fe82ec
14563 108.073719  rr2.sn-o3o-jj0s.googlevideo.com   192.168.0.105      QUIC       1399 Protected Payload (KP0), DCID=fe82ec
14564 108.073720  rr2.sn-o3o-jj0s.googlevideo.com   192.168.0.105      QUIC       1399 Protected Payload (KP0), DCID=fe82ec
14565 108.073721  rr2.sn-o3o-jj0s.googlevideo.com   192.168.0.105      QUIC       1399 Protected Payload (KP0), DCID=fe82ec
14566 108.073722  rr2.sn-o3o-jj0s.googlevideo.com   192.168.0.105      QUIC       1399 Protected Payload (KP0), DCID=fe82ec
14567 108.073723  rr2.sn-o3o-jj0s.googlevideo.com   192.168.0.105      QUIC       1399 Protected Payload (KP0), DCID=fe82ec
14568 108.073724  rr2.sn-o3o-jj0s.googlevideo.com   192.168.0.105      QUIC       1399 Protected Payload (KP0), DCID=fe82ec
14569 108.073725  rr2.sn-o3o-jj0s.googlevideo.com   192.168.0.105      QUIC       1399 Protected Payload (KP0), DCID=fe82ec
14570 108.073726  rr2.sn-o3o-jj0s.googlevideo.com   192.168.0.105      QUIC       1399 Protected Payload (KP0), DCID=fe82ec
```

6.

7. Once we pause the youtube video, the client sends a FIN request to pause the video and we get acknowledgement(FIN+ACK) from the you tube server(here maa03s server).



```
9568 40.345979   192.168.0.105                  maa03s41-in-f10.1e100.net     TCP               66 49312 → https(443) [FIN, ACK]
```



```
11821 56.016892   192.168.0.105       play.google.com       TCP          66 49360 → https(443) [FIN, ACK] Seq=793 Ack=7418
11823 56.080340   play.google.com     192.168.0.105         TCP          66 https(443) → 49360 [FIN, ACK] Seq=7418 Ack=794
```

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

Ans):

IPV4 Protocol: IP stands for **internet protocol**. It is a protocol that is defined in the TCP/IP model . IP provides the fundamental mechanism using which data is delivered between devices which may or may not be in the same network.The Packet uses the Ip addresses of the destination and source to identify the device over the network.

TCP: TCP stands for Transport Control Protocol. It is mainly used to transmit data securely over the network without any loss of data.It is a connection-oriented protocol. It has a handshaking mechanism to ensure the connection between the client and server.

- Pre-fetching and buffering are used in video streaming to ensure seamless video playback, for which TCP provides the buffer. Unlike video streaming, NPTEL uses Youtube for playing videos, which buffers and plays, TCP provides a reliable and quick transfer of packets.

QUIC: QUIC stands for Quick UDP Internet Connections.A protocol developed by google mainly to **reduce the latency compared to TCP**. QUIC can be said to be similar to TCP+TLS+HTTP implemented on UDP.

HTTP: HTTP stands for HyperText Transfer Protocol. It is used to access the data on the World Wide Web (www).HTTP is similar to the FTP as it also transfers the files from one host to another host.

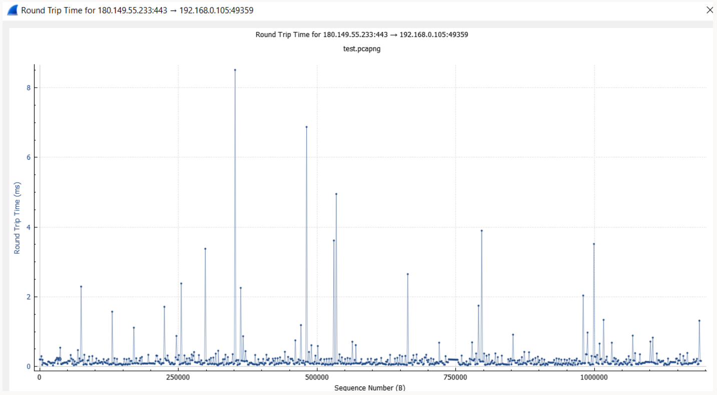
DNS: DNS stands for Domain Naming System. It is mainly used for name resolution. It stores the IP address along with corresponding website names which are easier to remember.

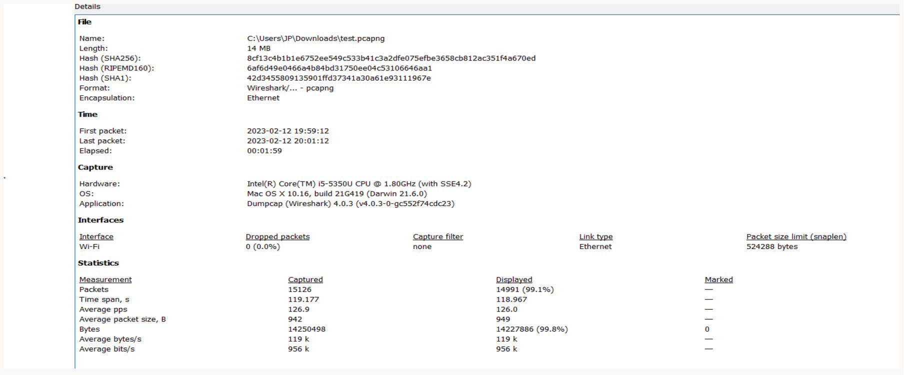
TLS: TLS stands for Transport Layer Security. It is used mainly to encrypt the data sent over the network to ensure no eavesdropping and prevent hacking.

5. **Calculate the following statistics from your traces while performing experiments at different times 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.**

Trace No.	Time Trace was Taken	Average Throughput	RTT	Number of TCP and UDP Packets	Average packet Size	Number of Retransmissions
Trace-1	8:00 PM	119k bytes/sec	Max- 9 ms; Min- 0.15 ms	TCP: 6834; UDP: 8249	949B	50
Trace-2	9:40 PM	32k bytes/sec	Max-0.11 ms Min- 0.105ms	TCP:1733 UDP: 11575 (QUIC packets)	1038 B	32
Trace-3	2:30 PM	83k bytes/sec	Max- 3.25ms Min-0.09ms	TCP: 7109 UDP: 9805	855 B	58

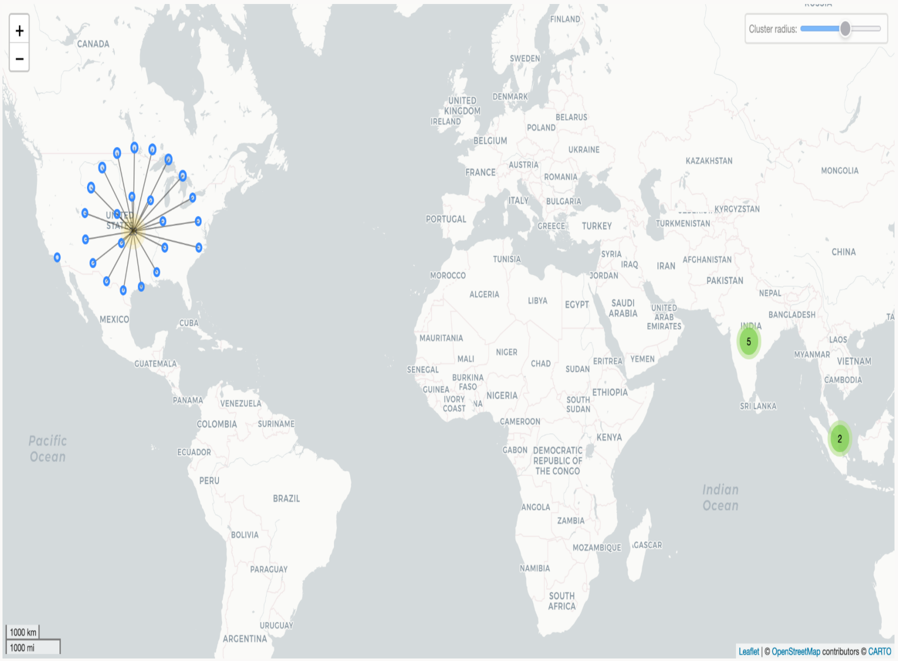
Snip for Trace-1: Round Trip Time for all the Packets. Max for Packet no: 6992 && Throughput

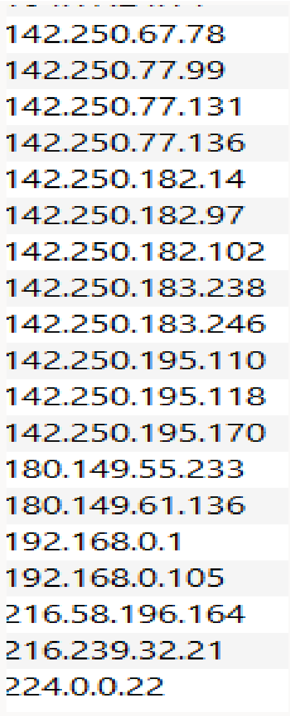



```
Details
File
 Name:                  C:\Users\JP\Downloads\test.pcapng
 Length:                14 MB
 Hash (SHA256):         8cf13c4b1b1e6752ee549c533b41c3a2dfe075efbe3658cb812ac351f4a670ed
 Hash (RIPEMD160):      6af6d49e0466a4b84bd31750ee04c53106646aa1
 Hash (SHA1):           42d3455809135901ffd37341a30a61e93111967e
 Format:                Wireshark/... - pcapng
 Encapsulation:         Ethernet
Time
 First packet:          2023-02-12 19:59:12
 Last packet:           2023-02-12 20:01:12
 Elapsed:               00:01:59
Capture
 Hardware:              Intel(R) Core(TM) i5-5350U CPU @ 1.80GHz (with SSE4.2)
 OS:                    Mac OS X 10.16, build 21G419 (Darwin 21.6.0)
 Application:           Dumpcap (Wireshark) 4.0.3 (v4.0.3-0-gc552f74cdc23)
Interfaces
 Interface              Dropped packets        Capture filter        Link type         Packet size limit (snaplen)
 Wi-Fi                  0 (0.0%)               none                  Ethernet          524288 bytes
Statistics
 Measurement                  Captured                  Displayed                  Marked
 Packets                      15126                     14991 (99.1%)              —
 Time span, s                 119.177                   118.967                    —
 Average pps                  126.9                     126.0                      —
 Average packet size, B       942                       949                        —
 Bytes                        14250498                  14227886 (99.8%)           0
 Average bytes/s              119 k                     119 k                      —
 Average bits/s               956 k                     956 k                      —
```

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

List of IP address:





142.250.67.78
142.250.77.99
142.250.77.131
142.250.77.136
142.250.182.14
142.250.182.97
142.250.182.102
142.250.183.238
142.250.183.246
142.250.195.110
142.250.195.118
142.250.195.170
180.149.55.233
180.149.61.136
192.168.0.1
192.168.0.105
216.58.196.164
216.239.32.21
224.0.0.22

As shown in the figure our client is communicating with multiple servers of youtube(blue dots). i.e.,Youtube communicates with client with multiple different servers at each instance.

Reasons for use multiple servers:

1. To reduce the latency of the system.
2. High level of Reliability and Availability.
3. Implements Load Balancing efficiently.
4. For safekeeping the data even if a server goes down.

One Drive Links for Traces at different time stamps:

Trace1: [Trace1_8PM.pcapng]
Trace2: [Trace2_940PM.pcap]
Trace3: [Trace3_2PM.pcap]