COL334 - Assignment 1 - Report

Nikhil Zawar - 2022CS11106

August 19, 2024

Contents

1 Measurement Tools - Ping & Traceroute

1.1 Ping

```
nikhil@nikhil-laptop:~$ ping -4 -c 10 google.com
PING google.com (142.250.206.142) 56(84) bytes of data.
64 bytes from del11s21-in-f14.1e100.net (142.250.206.142): icmp_seq=1 ttl=116 time=6.83 ms
64 bytes from del11s21-in-f14.1e100.net (142.250.206.142): icmp_seq=2 ttl=116 time=12.0 ms
64 bytes from del11s21-in-f14.1e100.net (142.250.206.142): icmp_seq=3 ttl=116 time=7.66 ms
64 bytes from del11s21-in-f14.1e100.net (142.250.206.142): icmp_seq=4 ttl=116 time=12.9 ms
64 bytes from del11s21-in-f14.1e100.net (142.250.206.142): icmp_seq=5 ttl=116 time=17.6 ms
64 bytes from del11s21-in-f14.1e100.net (142.250.206.142): icmp_seq=6 ttl=116 time=27.6 ms
64 bytes from del11s21-in-f14.1e100.net (142.250.206.142): icmp_seq=7 ttl=116 time=15.3 ms
64 bytes from del11s21-in-f14.1e100.net (142.250.206.142): icmp_seq=8 ttl=116 time=9.20 ms
64 bytes from del11s21-in-f14.1e100.net (142.250.206.142): icmp_seq=9 ttl=116 time=24.3 ms
64 bytes from del11s21-in-f14.1e100.net (142.250.206.142): icmp_seq=10 ttl=116 time=14.0 ms

--- google.com ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9017ms
rtt min/avg/max/mdev = 6.832/14.737/27.588/6.475 ms
nikhil@nikhil-laptop:~$
```

Ping **google.com** with 10 data packets on **IITD-Wifi**

Approximate RTT in milli-seconds:
Minimum = 4 ms
Maximum = 10 ms
Average = 5 ms

```
nikhil@nikhil-laptop:~$ ping -4 -c 10 sigcomm.org
PING sigcomm.org (190.92.158.4) 56(84) bytes of data.
64 bytes from server.hosting3.acm.org (190.92.158.4): icmp_seq=1 ttl=48 time=312 ms
64 bytes from server.hosting3.acm.org (190.92.158.4): icmp_seq=2 ttl=48 time=403 ms
64 bytes from server.hosting3.acm.org (190.92.158.4): icmp_seq=3 ttl=48 time=424 ms
64 bytes from server.hosting3.acm.org (190.92.158.4): icmp_seq=4 ttl=48 time=350 ms
64 bytes from server.hosting3.acm.org (190.92.158.4): icmp_seq=5 ttl=48 time=370 ms
64 bytes from server.hosting3.acm.org (190.92.158.4): icmp_seq=6 ttl=48 time=310 ms
64 bytes from server.hosting3.acm.org (190.92.158.4): icmp_seq=7 ttl=48 time=310 ms
64 bytes from server.hosting3.acm.org (190.92.158.4): icmp_seq=8 ttl=48 time=430 ms
64 bytes from server.hosting3.acm.org (190.92.158.4): icmp_seq=9 ttl=48 time=360 ms
64 bytes from server.hosting3.acm.org (190.92.158.4): icmp_seq=10 ttl=48 time=391 ms

--- sigcomm.org ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9008ms
rtt min/avg/max/mdev = 310.218/366.066/430.129/43.460 ms
nikhil@nikhil-laptop:~$
```

Ping **sigcomm.org** with 10 data packets on **IITD-Wifi**

Approximate RTT in milli-seconds:
Minimum = 301 ms
Maximum = 329 ms
Average = 309 ms

```
nikhil@nikhil-laptop:~$ ping -4 -c 10 google.com
PING google.com (142.250.206.142) 56(84) bytes of data.
64 bytes from del11s21-in-f14.1e100.net (142.250.206.142): icmp_seq=1 ttl=111 time=29.0 ms
64 bytes from del11s21-in-f14.1e100.net (142.250.206.142): icmp_seq=2 ttl=111 time=47.2 ms
64 bytes from del11s21-in-f14.1e100.net (142.250.206.142): icmp_seq=3 ttl=111 time=74.5 ms
64 bytes from del11s21-in-f14.1e100.net (142.250.206.142): icmp_seq=4 ttl=111 time=42.3 ms
64 bytes from del11s21-in-f14.1e100.net (142.250.206.142): icmp_seq=5 ttl=111 time=40.5 ms
64 bytes from del11s21-in-f14.1e100.net (142.250.206.142): icmp_seq=6 ttl=111 time=42.0 ms
64 bytes from del11s21-in-f14.1e100.net (142.250.206.142): icmp_seq=7 ttl=111 time=36.4 ms
64 bytes from del11s21-in-f14.1e100.net (142.250.206.142): icmp_seq=8 ttl=111 time=49.0 ms
64 bytes from del11s21-in-f14.1e100.net (142.250.206.142): icmp_seq=9 ttl=111 time=44.3 ms
64 bytes from del11s21-in-f14.1e100.net (142.250.206.142): icmp_seq=10 ttl=111 time=61.8 ms

--- google.com ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9016ms
rtt min/avg/max/mdev = 28.978/46.685/74.469/12.296 ms
nikhil@nikhil-laptop:~$
```

Ping **google.com** with 10 data packets on **Mobile Data**

Approximate RTT in milli-seconds:
Minimum = 43 ms
Maximum = 79 ms
Average = 56 ms

```
nikhil@nikhil-laptop:~$ ping -4 -c 10 sigcomm.org
PING sigcomm.org (190.92.158.4) 56(84) bytes of data.
64 bytes from server.hosting3.acm.org (190.92.158.4): icmp_seq=1 ttl=45 time=307 ms
64 bytes from server.hosting3.acm.org (190.92.158.4): icmp_seq=2 ttl=45 time=318 ms
64 bytes from server.hosting3.acm.org (190.92.158.4): icmp_seq=3 ttl=45 time=323 ms
64 bytes from server.hosting3.acm.org (190.92.158.4): icmp_seq=4 ttl=45 time=302 ms
64 bytes from server.hosting3.acm.org (190.92.158.4): icmp_seq=5 ttl=45 time=316 ms
64 bytes from server.hosting3.acm.org (190.92.158.4): icmp_seq=6 ttl=45 time=307 ms
64 bytes from server.hosting3.acm.org (190.92.158.4): icmp_seq=7 ttl=45 time=319 ms
64 bytes from server.hosting3.acm.org (190.92.158.4): icmp_seq=8 ttl=45 time=304 ms
64 bytes from server.hosting3.acm.org (190.92.158.4): icmp_seq=9 ttl=45 time=313 ms
64 bytes from server.hosting3.acm.org (190.92.158.4): icmp_seq=10 ttl=45 time=341 ms

--- sigcomm.org ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9014ms
rtt min/avg/max/mdev = 301.858/315.009/340.903/10.805 ms
nikhil@nikhil-laptop:~$
```

Ping **sigcomm.org** with 10 data packets on **Mobile Data**

Approximate RTT in milli-seconds:
Minimum = 344 ms
Maximum = 407 ms
Average = 371 ms

Table 1: Table of Ping Results

1.1.1 A. Average Ping Latencies for varying networks & websites

1. **Comparing Average Ping Latencies different websites with the same network**
   Answer:

   - **Geographical Distance:** Google.com likely has servers distributed globally, including data centers(Australia) close to our location as compared to data centers of sigcomm.org(USA). This proximity reduces the round-trip time because the data packets travel shorter distances in case of google.com.
   - **Network route:** The number of hops between my laptop and google.com is 9 hops on IITD-Wifi, while sigcomm.org takes 15 hops on the same network. More hops generally implies a longer path and more points of congestion, which increases latency.
   - **Server Response Time:** Google has to handle vast amounts of data and traffic, with minimum delay. So the google servers are highly optimized to respond almost instantly. While, sigcomm.org might be hosted on servers with slower processing time which contribute to higher latency.
   - **Internet Peering:** Google, being so large, might have direct peering connections with many ISPs, which make it faster and allows more efficient routing. While, sigcomm.org may rely on standard internet transit routes, which can be slower due to additional routing that is required and the peering points.

2. **Comparing Average Ping Latencies of same website with different networks**
   Answer:

   - **Network Infrastructure:** IITD-Wifi is likely a low-latency network optimized for academic use. It probably has a direct connection to a local ISP and this connection being high-bandwidth. Mobile networks require increased number of hops from the mobile device to the cell tower and then to the internet backbone, and also these are wireless, which cause higher latency.
   - **Wireless Medium:** The mobile-data connections have radio-nature. So the factors like signal-strength, network congestion all affect the ping latency significantly. The IITD-Wifi network is likely a wired network after a few hops, while the hops in case of mobile networks is more. That is why IITD-wifi has lower latency.
   - **Network congestion and load:** Mobile networks, during peak hours, can experience significant congestion, which increases latency. This can explain why the ping times on mobile data are consistently higher than on IITD-Wifi. Wi-Fi networks, in a institute, may have more consistent performance and less variation in load, contributing to lower latency.

1.1.2 B. Ping Protocol & Theretical Upper Limt

1. **Protocol used by Ping**
   Answer:

   - The ping tool uses ICMP - Internet Control Message Protocol to measure the round-trip-time(RTT) between two devices. This is a network layer protocol, designed primarily for diagnostic and control purposes. It is different from TCP/UDP which are used for data transfer.
   - The ping command on running, sends a ICMP Echo Request message to the target host. The target receives the request message and responds with ICMP Echo Reply message. When the sender recieves the Echo Reply, it calculates the RTT by subtracting the time-stamps

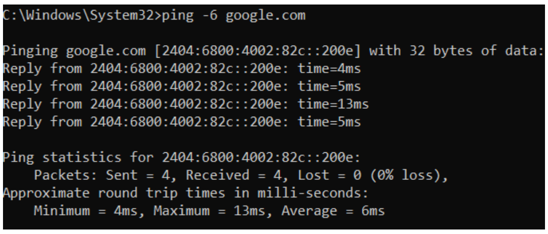
- Proof of ICMP protocol - found using wireshark while ping command is running:

```
126 11.496546   2001:df4:e000:3fc1:…  2404:6800:4002:82c:…  ICMPv6   94 Echo (ping) request id=0x0001, seq=304, hop limit=128 (reply in 127)
127 11.500586   2404:6800:4002:82c:…  2001:df4:e000:3fc1:…  ICMPv6   94 Echo (ping) reply id=0x0001, seq=304, hop limit=117 (request in 126)
142 12.509476   2001:df4:e000:3fc1:…  2404:6800:4002:82c:…  ICMPv6   94 Echo (ping) request id=0x0001, seq=305, hop limit=128 (reply in 143)
143 12.513502   2404:6800:4002:82c:…  2001:df4:e000:3fc1:…  ICMPv6   94 Echo (ping) reply id=0x0001, seq=305, hop limit=117 (request in 142)
159 13.527302   2001:df4:e000:3fc1:…  2404:6800:4002:82c:…  ICMPv6   94 Echo (ping) request id=0x0001, seq=306, hop limit=128 (reply in 160)
160 13.531929   2404:6800:4002:82c:…  2001:df4:e000:3fc1:…  ICMPv6   94 Echo (ping) reply id=0x0001, seq=306, hop limit=117 (request in 159)
189 14.532651   2001:df4:e000:3fc1:…  2404:6800:4002:82c:…  ICMPv6   94 Echo (ping) request id=0x0001, seq=307, hop limit=128 (reply in 191)
191 14.536257   2404:6800:4002:82c:…  2001:df4:e000:3fc1:…  ICMPv6   94 Echo (ping) reply id=0x0001, seq=307, hop limit=117 (request in 189)
469 15.535573   2001:df4:e000:3fc1:…  2404:6800:4002:82c:…  ICMPv6   94 Echo (ping) request id=0x0001, seq=308, hop limit=128 (reply in 472)
472 15.539370   2404:6800:4002:82c:…  2001:df4:e000:3fc1:…  ICMPv6   94 Echo (ping) reply id=0x0001, seq=308, hop limit=117 (request in 469)
491 16.539704   2001:df4:e000:3fc1:…  2404:6800:4002:82c:…  ICMPv6   94 Echo (ping) request id=0x0001, seq=309, hop limit=128 (reply in 492)
492 16.544172   2404:6800:4002:82c:…  2001:df4:e000:3fc1:…  ICMPv6   94 Echo (ping) reply id=0x0001, seq=309, hop limit=117 (request in 491)
493 17.558634   2001:df4:e000:3fc1:…  2404:6800:4002:82c:…  ICMPv6   94 Echo (ping) request id=0x0001, seq=310, hop limit=128 (reply in 494)
494 17.562899   2404:6800:4002:82c:…  2001:df4:e000:3fc1:…  ICMPv6   94 Echo (ping) reply id=0x0001, seq=310, hop limit=117 (request in 493)
498 18.573155   2001:df4:e000:3fc1:…  2404:6800:4002:82c:…  ICMPv6   94 Echo (ping) request id=0x0001, seq=311, hop limit=128 (reply in 499)
499 18.577224   2404:6800:4002:82c:…  2001:df4:e000:3fc1:…  ICMPv6   94 Echo (ping) reply id=0x0001, seq=311, hop limit=117 (request in 498)
```

2. **Theoretical upper limit of packet size for ping protocol**
   Answer:

   - For IPv4, the total packet size, can be up to 65535 bytes($2^{16} - 1$). This includes IP header(20 bytes) and ICMP header(8 bytes), which implies the maximum ICMP payload to be 65507 bytes. Theoretically, this is the maximum amount of data(65507 bytes) that the ping command can send in a request.
   - The ICMP payload size can vary, but it is limited by the MTU - Maximum Transmission unit of the network. The MTU is the largest packet size that can be sent without fragmentation. Common MTU size is around 5000 bytes for Ethernet, which is very near to the max data that can be sent using ping command experimentally.
   - On pinging with packet larger than MTU, it either leads to fragmentation or if the "Don't Fragment"(DF) flag is set, it leads to packet drop and ICMP error message.

1.1.3 C. Ping with IPv6



```
C:\Windows\System32>ping -6 google.com

Pinging google.com [2404:6800:4002:82c::200e] with 32 bytes of data:
Reply from 2404:6800:4002:82c::200e: time=4ms
Reply from 2404:6800:4002:82c::200e: time=5ms
Reply from 2404:6800:4002:82c::200e: time=13ms
Reply from 2404:6800:4002:82c::200e: time=5ms

Ping statistics for 2404:6800:4002:82c::200e:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 4ms, Maximum = 13ms, Average = 6ms
```

Figure 1: Ping IPv6 google.com using IITD-Wifi

```
C:\Windows\System32>ping -6 google.com

Pinging google.com [2404:6800:4007:826::200e] with 32 bytes of data:
Reply from 2404:6800:4007:826::200e: time=55ms
Reply from 2404:6800:4007:826::200e: time=58ms
Reply from 2404:6800:4007:826::200e: time=123ms
Reply from 2404:6800:4007:826::200e: time=82ms

Ping statistics for 2404:6800:4007:826::200e:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 55ms, Maximum = 123ms, Average = 79ms
```

Figure 2: Ping IPv6 google.com using Mobile Data

```
C:\Windows\System32>ping -6 sigcomm.org
Ping request could not find host sigcomm.org. Please check the name and try again.
```

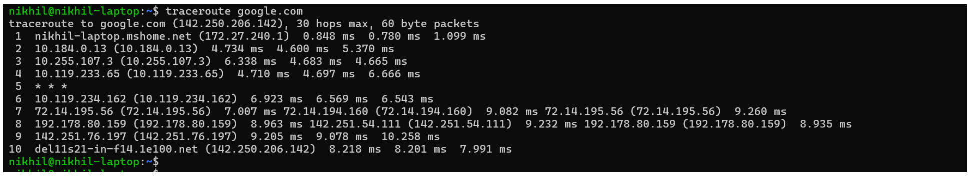
Figure 3: Ping IPv6 sigcomm.org using IITD-Wifi

```
C:\Windows\System32>ping -6 sigcomm.org
Ping request could not find host sigcomm.org. Please check the name and try again.
```

Figure 4: Ping IPv6 sigcomm.org using Mobile Data

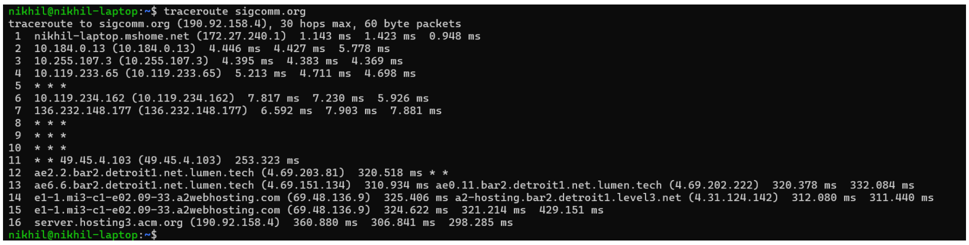
From the above screenshots we can conclude that google.com can be pinged with IPv6 and the only difference we notice is in ping latency. The major difference we notice is that *sigcomm.org* cannot be pinged with IPv6, because the Ping request cannot find the host sigcomm.org. The domain sigcomm.org does not have an IPv6 address.

1.2 Traceroute



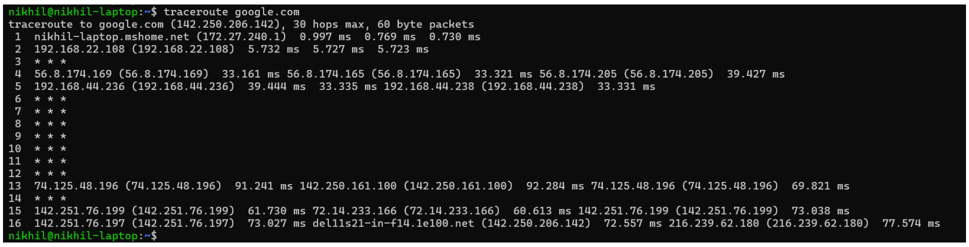
```
nikhil@nikhil-laptop:~$ traceroute google.com
traceroute to google.com (142.250.206.142), 30 hops max, 60 byte packets
 1  nikhil-laptop.mshome.net (172.27.240.1)  0.848 ms  0.780 ms  1.099 ms
 2  10.184.0.13 (10.184.0.13)  4.734 ms  4.600 ms  5.370 ms
 3  10.255.107.3 (10.255.107.3)  6.338 ms  4.683 ms  4.665 ms
 4  10.119.233.65 (10.119.233.65)  4.710 ms  4.697 ms  6.666 ms
 5  * * *
 6  10.119.234.162 (10.119.234.162)  6.923 ms  6.569 ms  6.543 ms
 7  72.14.195.56 (72.14.195.56)  7.007 ms 72.14.194.160 (72.14.194.160)  9.082 ms 72.14.195.56 (72.14.195.56)  9.260 ms
 8  192.178.80.159 (192.178.80.159)  8.963 ms 142.251.54.111 (142.251.54.111)  9.232 ms 192.178.80.159 (192.178.80.159)  8.935 ms
 9  142.251.76.197 (142.251.76.197)  9.205 ms  9.078 ms  10.258 ms
10  del11s21-in-f14.1e100.net (142.250.206.142)  8.218 ms  8.201 ms  7.991 ms
nikhil@nikhil-laptop:~$
```

Traceroute **google.com** on **IITD-Wifi**
Number of hops = 9



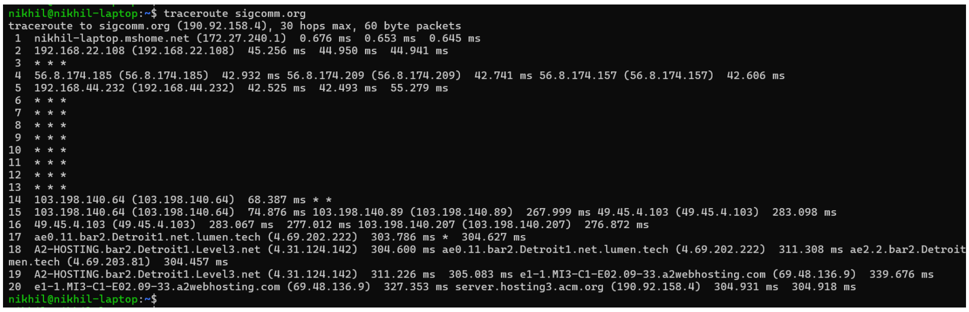
```
nikhil@nikhil-laptop:~$ traceroute sigcomm.org
traceroute to sigcomm.org (190.92.158.4), 30 hops max, 60 byte packets
 1  nikhil-laptop.mshome.net (172.27.240.1)  1.143 ms  1.423 ms  0.948 ms
 2  10.184.0.13 (10.184.0.13)  4.446 ms  4.427 ms  5.778 ms
 3  10.255.107.3 (10.255.107.3)  4.395 ms  4.383 ms  4.369 ms
 4  10.119.233.65 (10.119.233.65)  5.213 ms  4.711 ms  4.698 ms
 5  * * *
 6  10.119.234.162 (10.119.234.162)  7.817 ms  7.230 ms  5.926 ms
 7  136.232.148.177 (136.232.148.177)  6.592 ms  7.903 ms  7.881 ms
 8  * * *
 9  * * *
10  * * *
11  * * 49.45.4.103 (49.45.4.103)  253.323 ms
12  ae2.2.bar2.detroit1.net.lumen.tech (4.69.203.81)  320.518 ms * *
13  ae6.6.bar2.detroit1.net.lumen.tech (4.69.151.134)  310.934 ms ae0.11.bar2.detroit1.net.lumen.tech (4.69.202.222)  320.378 ms  332.084 ms
14  e1-1.mi3-c1-e02.09-33.a2webhosting.com (69.48.136.9)  325.406 ms a2-hosting.bar2.detroit1.level3.net (4.31.124.142)  312.080 ms  311.440 ms
15  e1-1.mi3-c1-e02.09-33.a2webhosting.com (69.48.136.9)  324.622 ms  321.214 ms  429.151 ms
16  server.hosting3.acm.org (190.92.158.4)  360.880 ms  306.841 ms  298.285 ms
nikhil@nikhil-laptop:~$
```

Traceroute **sigcomm.org** on **IITD-Wifi**
Number of hops = 15



```
nikhil@nikhil-laptop:~$ traceroute google.com
traceroute to google.com (142.250.206.142), 30 hops max, 60 byte packets
 1  nikhil-laptop.mshome.net (172.27.240.1)  0.997 ms  0.769 ms  0.730 ms
 2  192.168.22.108 (192.168.22.108)  5.732 ms  5.727 ms  5.723 ms
 3  * * *
 4  56.8.174.169 (56.8.174.169)  33.161 ms 56.8.174.165 (56.8.174.165)  33.321 ms 56.8.174.205 (56.8.174.205)  39.427 ms
 5  192.168.44.236 (192.168.44.236)  39.444 ms  33.335 ms 192.168.44.238 (192.168.44.238)  33.331 ms
 6  * * *
 7  * * *
 8  * * *
 9  * * *
10  * * *
11  * * *
12  * * *
13  74.125.48.196 (74.125.48.196)  91.241 ms 142.250.161.100 (142.250.161.100)  92.284 ms 74.125.48.196 (74.125.48.196)  69.821 ms
14  * * *
15  142.251.76.199 (142.251.76.199)  61.730 ms 72.14.233.166 (72.14.233.166)  60.613 ms 142.251.76.199 (142.251.76.199)  73.038 ms
16  142.251.76.197 (142.251.76.197)  73.027 ms del11s21-in-f14.1e100.net (142.250.206.142)  72.557 ms 216.239.62.180 (216.239.62.180)  77.574 ms
nikhil@nikhil-laptop:~$
```

Traceroute **google.com** on **Mobile Data**
Number of hops = 11



```
nikhil@nikhil-laptop:~$ traceroute sigcomm.org
traceroute to sigcomm.org (190.92.158.4), 30 hops max, 60 byte packets
 1  nikhil-laptop.mshome.net (172.27.240.1)  0.676 ms  0.653 ms  0.645 ms
 2  192.168.22.108 (192.168.22.108)  45.256 ms  44.950 ms  44.941 ms
 3  * * *
 4  56.8.174.185 (56.8.174.185)  42.932 ms 56.8.174.209 (56.8.174.209)  42.741 ms 56.8.174.157 (56.8.174.157)  42.606 ms
 5  192.168.44.232 (192.168.44.232)  42.525 ms  42.493 ms  55.279 ms
 6  * * *
 7  * * *
 8  * * *
 9  * * *
10  * * *
11  * * *
12  * * *
13  * * *
14  103.198.140.64 (103.198.140.64)  68.387 ms * *
15  103.198.140.64 (103.198.140.64)  74.876 ms 103.198.140.89 (103.198.140.89)  267.999 ms 49.45.4.103 (49.45.4.103)  283.098 ms
16  49.45.4.103 (49.45.4.103)  283.067 ms  277.012 ms 103.198.140.207 (103.198.140.207)  276.872 ms
17  ae0.11.bar2.Detroit1.net.lumen.tech (4.69.202.222)  303.786 ms *  304.627 ms
18  A2-HOSTING.bar2.Detroit1.Level3.net (4.31.124.142)  304.600 ms ae0.11.bar2.Detroit1.net.lumen.tech (4.69.202.222)  311.308 ms ae2.2.bar2.Detroit
men.tech (4.69.203.81)  304.457 ms
19  A2-HOSTING.bar2.Detroit1.Level3.net (4.31.124.142)  311.226 ms  305.083 ms e1-1.MI3-C1-E02.09-33.a2webhosting.com (69.48.136.9)  339.676 ms
20  e1-1.MI3-C1-E02.09-33.a2webhosting.com (69.48.136.9)  327.353 ms server.hosting3.acm.org (190.92.158.4)  304.931 ms  304.918 ms
nikhil@nikhil-laptop:~$
```

Traceroute **sigcomm.org** on **Mobile Data**
Number of hops = 18

Table 2: Table of Traceroute results

1.2.1 A. IP hops and autonomous systems

The number of IP hops we see in the traceroute and the list of autonomous systems can be derived from the geolocation data in part E of the assignment. The same data was verified by me using the command *curl https://api.hackertarget.com/aslookup/?q=IP address* on the command prompt for each IP address

1. traceroute google.com using IITD-Wifi
   - Number of IP hops - 9
   - Autonomous Systems - IITD-Wifi, Google LLC
2. traceroute sigcomm.org using IITD-Wifi
   - Number of IP hops - 15
   - Autonomous Systems - IITD-Wifi, Reliance Jio, Lumen, A2-Hosting.
3. traceroute google.com using Mobile Internet
   - Number of IP hops - 9
   - Autonomous Systems - Reliance Jio, Some Private(Probably Airtel), Google LLC
4. traceroute sigcomm.org using Mobile Internet
   - Number of IP hops - 9
   - Autonomous Systems - Reliance Jio, Some Private(Probably Airtel), Lumen, A2-Hosting

1.2.2 B. Explaintion for the '*' in output

If the packet is successfully returned, the RTT is displayed. If the packet is lost or filtered, a '*' is shown instead. This means that the traceroute did not recieve reponse from the router within the timeout period.
If the router responds to the traceroute request, its IP address or DNS-resolved hostname is displayed. If the router does not respond, i.e., when all the three probes are not received, this field becomes empty, and the "**Request timed out**" message appears.
The hops that do not return any response in all three RTT columns could be due to packet drops, ICMP filtering(Firewall), network security policies, or devices that are set not to respond to traceroute.

- **Packet Loss:** Occurs due to multiple reasons like - network congestion, hardware issues, etc. Network Congestion happens when the network is overloaded with data, then some packets are droped to manage the traffic. This happens when the networks have limited bandwidth during peak usage times. Also, faulty network hardware, such as routers, switches can cause packet loss during transmission.
- **Packet Filtering using firewalls:** A firewall might block ICMP Echo Request packets (sent by traceroute) or ICMP Echo Reply packets (sent by the final dest). If these packets are blocked, traceroute cannot complete the trace, leading to * symbol

1.2.3 C. Multiple IP addresses for the same hop count

- **Load Balancing:** Load balancing is a technique used to distribute network traffic across multiple servers or network paths to ensure no single device is overwhelmed. The routers we use are specifically configures to distribute the network traffic across multiple paths to optimize performance. Due to this, when we run a traceroute command, the packets might take different paths through the network, leading to different routers responding at the same hop.
- **Traceroute Behavior:** Traceroute works by sending out multiple packets with incrementally increasing TTL (Time-to-Live) values. If a router has multiple paths to the destination, the packets may take different paths, resulting in different IP addresses being returned for the same hop

1.2.4 D. Ping the first hop IP address(when tracerouting google.com using IITD-WiFi) using mobile data

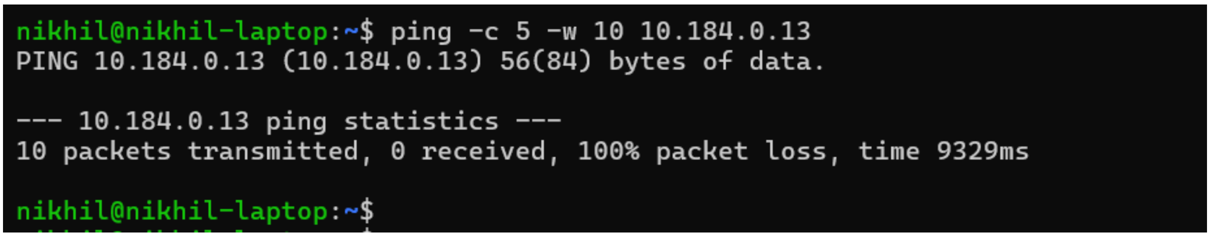


Figure 5: Ping IITD-Wifi using Mobile Network

- The IP address of the first hop router when tracerouting google.com using IITD-Wifi is : 10.184.0.13
- When I ping this IP address using the Mobile Data network(Reliance Jio), my computer does not receive any response, which results in 100% packet loss. The IP address is 10.184.0.13, which is a private network(IITD-Wifi), which means that it is not detectable on public network(Mobile-Data)
- We expect the same, to not receive the response. Because the private networks are designed to keep certain parts of a network isolated from the broader internet for security and management reasons. This will ensure that the internal network traffic is present only within the internal networks and also it prevents unauthorized access from external sources.

1.2.5 E. 2-tiered/3-tiered Internet architecture

1. **To google.com using IITD-Wifi**
   Tier 3 : IITD-Wifi
   Tier 1 : Google LLC
2. **To sigcomm.org using IITD-Wifi**
   Tier 3 : IITD-Wifi
   Tier 2 : Reliance Jio & Lumen
   Tier 1 : A2-Hosting
3. **To google.com using Mobile Data**
   Tier 2 : Reliance Jio
   Tier 1 : Google LLC

4. **To sigcomm.org using Mobile Data**
   Tier 3 : Reliance Jio
   Tier 2 : Lumen
   Tier 1 : A2-Hosting

Why do we not observe 3-tiered system everywhere?
Direct Peering: Large organizations and service providers have direct peering agreements with each other, bypassing the need for intermediate Tier 2 or Tier 3 networks. These complex peering arrangements leads to path that do not fit in the 3-tiered model.
Mobile Data Networks: Mobile networks (like Reliance Jio) mostly operate as Tier 2 networks in specific regions but may also connect to Tier 1 providers for global reach. Their role can shift based on regional agreements.

1.2.6 F. Using geolocation to reason RTTs

IP Address	Location	Network	Postal Code	Approximate Latitude / Longitude*, and Accuracy Radius	ISP / Organization	Domain	Co Ty
172.27.240.1	The IP address '172.27.240.1' is a reserved IP address (private, multicast, etc.).						
10.184.0.13	The IP address '10.184.0.13' is a reserved IP address (private, multicast, etc.).						
10.255.107.3	The IP address '10.255.107.3' is a reserved IP address (private, multicast, etc.).						
10.119.233.65	The IP address '10.119.233.65' is a reserved IP address (private, multicast, etc.).						
10.119.234.162	The IP address '10.119.234.162' is a reserved IP address (private, multicast, etc.).						
72.14.195.56	United States (US), North America	72.14.194.0/23	-	37.751, -97.822 (1000 km)	Google	-	Co
192.178.80.159	United States (US), North America	192.178.80.0/22	-	37.751, -97.822 (1000 km)	Google	-	Co
142.251.76.197	United States (US), North America	142.251.76.0/22	-	37.751, -97.822 (1000 km)	Google	-	Co
142.250.206.142	Florida, United States (US), North America	142.250.206.0/23	-	28.6344, -81.6221 (1000 km)	Google Servers	1e100.net	Ca

Figure 6: Geolocating google.com using IITD-Wifi

IP Address	Location	Network	Postal Code	Approximate Latitude / Longitude*, and Accuracy Radius	ISP / Organization	Domain
172.27.240.1	The IP address '172.27.240.1' is a reserved IP address (private, multicast, etc.).					
192.168.22.108	The IP address '192.168.22.108' is a reserved IP address (private, multicast, etc.).					
56.8.174.177	United States (US), North America	56.8.0.0/13	-	37.751, -97.822 (1000 km)	-	-
192.168.44.234	The IP address '192.168.44.234' is a reserved IP address (private, multicast, etc.).					
74.125.48.196	United States (US), North America	74.125.48.0/22	-	37.751, -97.822 (1000 km)	Google	-
209.85.148.118	United States (US), North America	209.85.148.0/22	-	37.751, -97.822 (1000 km)	Google	-
142.251.54.100	United States (US), North America	142.251.48.0/20	-	37.751, -97.822 (1000 km)	Google	-
72.14.233.31	Stockton, California, United States (US), North America	72.14.233.16/28	95206	37.9186, -121.3132 (1000 km)	Google	-
172.217.167.14	United States (US), North America	172.217.160.0/20	-	37.751, -97.822 (1000 km)	Google Servers	1e100.net

Figure 7: Geolocating google.com using Mobile Data

IP Address	Location	Network	Postal Code	Approximate Latitude / Longitude*, and Accuracy Radius	ISP / Organization	Domain
172.27.240.1	The IP address '172.27.240.1' is a reserved IP address (private, multicast, etc.).					
10.184.0.13	The IP address '10.184.0.13' is a reserved IP address (private, multicast, etc.).					
10.255.107.3	The IP address '10.255.107.3' is a reserved IP address (private, multicast, etc.).					
10.119.233.65	The IP address '10.119.233.65' is a reserved IP address (private, multicast, etc.).					
10.119.234.162	The IP address '10.119.234.162' is a reserved IP address (private, multicast, etc.).					
136.232.148.177	New Delhi, National Capital Territory of Delhi, India (IN), Asia	136.232.148.0/22	110043	28.652, 77.1663 (5 km)	Jio	-
4.7.26.61	San Bernardino, California, United States (US), North America	4.7.26.0/24	92407	34.2098, -117.3997 (20 km)	Lumen	-
4.69.202.222	United States (US), North America	4.69.200.0/22	-	37.751, -97.822 (1000 km)	Lumen	-
69.48.136.9	United States (US), North America	69.48.136.0/23	-	37.751, -97.822 (1000 km)	A2 Hosting	a2webhosti
190.92.158.4	Michigan, United States (US), North America	190.92.152.0/21	-	42.4652, -83.3713 (1000 km)	A2 Hosting	a2webhosti

Figure 8: Geolocating sigcomm.org using IITD-Wifi

IP Address	Location	Network	Postal Code	Approximate Latitude / Longitude*, and Accuracy Radius	ISP / Organization	Domain
172.27.240.1	The IP address '172.27.240.1' is a reserved IP address (private, multicast, etc.).					
192.168.22.108	The IP address '192.168.22.108' is a reserved IP address (private, multicast, etc.).					
56.8.174.165	United States (US), North America	56.8.0.0/13	-	37.751, -97.822 (1000 km)	-	-
192.168.44.236	The IP address '192.168.44.236' is a reserved IP address (private, multicast, etc.).					
103.198.140.64	Singapore (SG), Asia	103.198.140.0/24	-	1.3673, 103.8014 (1000 km)	Reliance Jio Infocomm Pte Singapore	-
4.69.202.222	United States (US), North America	4.69.200.0/22	-	37.751, -97.822 (1000 km)	Lumen	-
4.31.124.142	Detroit, Michigan, United States (US), North America	4.31.124.128/26	48213	42.3983, -82.992 (50 km)	Lumen	-
69.48.136.9	United States (US), North America	69.48.136.0/23	-	37.751, -97.822 (1000 km)	A2 Hosting	a2webhosting.
190.92.158.4	Michigan, United States (US), North America	190.92.152.0/21	-	42.4652, -83.3713 (1000 km)	A2 Hosting	a2webhosting.

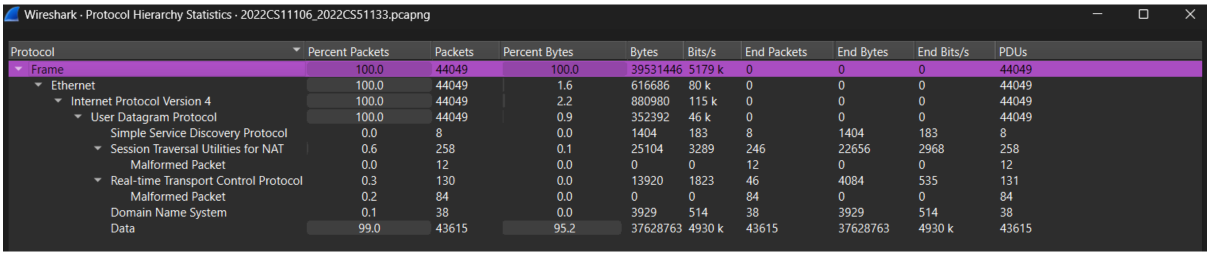
Figure 9: Geolocating sigcomm.org using Mobile Data

The data we get from the geolocation goes hand-in-hand with our intuition. My initial conclusion was that sigcomm.org travels farther distance as compared to google.com which causes higher latency while doing a ping on sigcomm.org. The geolocations also implies that large networks like Google, has servers spread in multiple areas, while smaller networks have their main data centers localised. Also, we observe that some geolocations are not detectable on public networks. These are private networks like IITD-Wifi.

2 Network Data Collection and Header Analysis

Link to pcap file for 60 second video call: https://drive.google.com/file/d/1CMSIPwXJaaHMo-BNtstr7zXBzyOGTzaT/view?usp=sharing

2.1 A. Protocols & Number of Packets


Wireshark · Protocol Hierarchy Statistics · 2022CS11106_2022CS51133.pcapng

Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s	PDUs
Frame	100.0	44049	100.0	39531446	5179 k	0	0	0	44049
Ethernet	100.0	44049	1.6	616686	80 k	0	0	0	44049
Internet Protocol Version 4	100.0	44049	2.2	880980	115 k	0	0	0	44049
User Datagram Protocol	100.0	44049	0.9	352392	46 k	0	0	0	44049
Simple Service Discovery Protocol	0.0	8	0.0	1404	183	8	1404	183	8
Session Traversal Utilities for NAT	0.6	258	0.1	25104	3289	246	22656	2968	258
Malformed Packet	0.0	12	0.0	0	0	12	0	0	12
Real-time Transport Control Protocol	0.3	130	0.0	13920	1823	46	4084	535	131
Malformed Packet	0.2	84	0.0	0	0	84	0	0	84
Domain Name System	0.1	38	0.0	3929	514	38	3929	514	38
Data	99.0	43615	95.2	37628763	4930 k	43615	37628763	4930 k	43615

Figure 10: Protocol Hierarchy Statistics

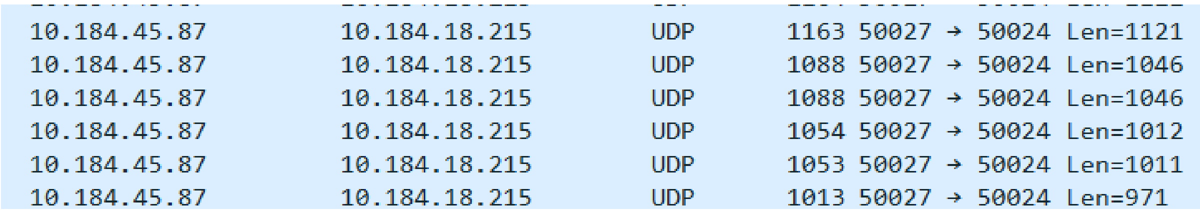
Protocols

1. **Link Layer Protocols:** Ethernet
2. **Network Layer Protocols:** IPv4 - Internet Protocol version 4
3. **Transport Layer Protocols:** UDP - User Datagram Protocol
4. **Application Layer Protocols:** STUN - Session Traversal Utilities for NAT; RTCP - Real-time Transport Control Protocol

Log of number of packets

1. Total Number of Packets during the 1 minute - **44530**
2. Total Number of Packets relevant to teams call: **43991**
   (Filter Used - ip.addr == 10.184.18.215 && ip.addr == 10.184.45.87)
3. UDP packets: **43991 (100%)**
   (Filter Used - (ip.addr == 10.184.18.215 && ip.addr == 10.184.45.87) && udp)
4. STUN packets: 246 **(0.6%)**
   (Filter Used - (ip.addr == (10.184.18.215 && ip.addr == 10.184.45.87) && stun)
5. RTCP Packets: 130 **(0.3%)**
   (Filter Used - (ip.addr == (10.184.18.215 && ip.addr == 10.184.45.87) && rtcp)

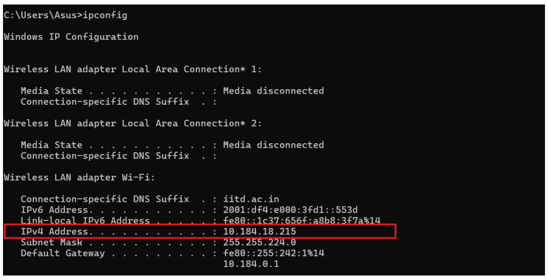
2.2 B. Direct Connection



10.184.45.87	10.184.18.215	UDP	1163 50027 → 50024 Len=1121
10.184.45.87	10.184.18.215	UDP	1088 50027 → 50024 Len=1046
10.184.45.87	10.184.18.215	UDP	1088 50027 → 50024 Len=1046
10.184.45.87	10.184.18.215	UDP	1054 50027 → 50024 Len=1012
10.184.45.87	10.184.18.215	UDP	1053 50027 → 50024 Len=1011
10.184.45.87	10.184.18.215	UDP	1013 50027 → 50024 Len=971

Figure 11: Connection between two end-hosts

Yes, we do observe a direct connection between the two end-hosts. This is seen by comparing the IP addresses of both our systems, and the IP address shown on wireshark corresponding to all the packets. This can be shown with the following screenshots -



```
C:\Users\Asus>ipconfig

Windows IP Configuration


Wireless LAN adapter Local Area Connection* 1:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Wireless LAN adapter Local Area Connection* 2:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . : iitd.ac.in
   IPv6 Address. . . . . . . . . . . : 2001:df4:e000:3fd1::553d
   Link-local IPv6 Address . . . . . : fe80::1c37:656f:a8b8:3f7a%14
   IPv4 Address. . . . . . . . . . . : 10.184.18.215
   Subnet Mask . . . . . . . . . . . : 255.255.224.0
   Default Gateway . . . . . . . . . : fe80::255:242:1%14
                                       10.184.0.1
```

Figure 12: My IP address

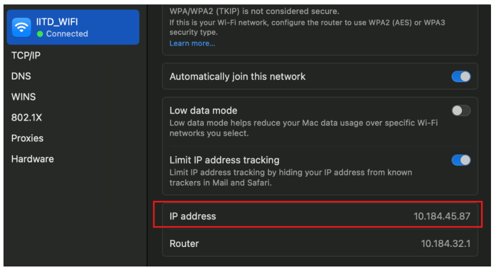
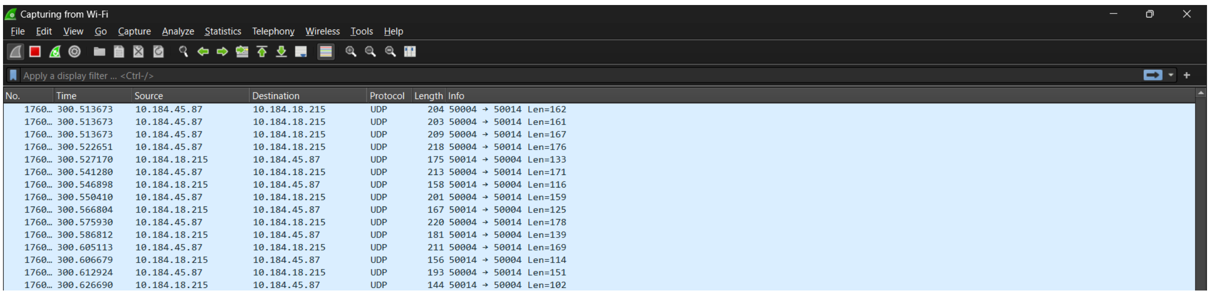


Figure 13: IP address of my partner

2.3 C. Audio and Video Packets

We try to determine the size of the audio and video packets by observation. Running wireshark in different cases like - audio only without speaking, audio only with speaking, video only, audio with video; and then notice the size of patterns in different cases. The following screenshots highlight my analysis:



No.	Time	Source	Destination	Protocol	Length	Info
1760…	300.513673	10.184.45.87	10.184.18.215	UDP	204	50004 → 50014 Len=162
1760…	300.513673	10.184.45.87	10.184.18.215	UDP	203	50004 → 50014 Len=161
1760…	300.513673	10.184.45.87	10.184.18.215	UDP	209	50004 → 50014 Len=167
1760…	300.522651	10.184.45.87	10.184.18.215	UDP	218	50004 → 50014 Len=176
1760…	300.527170	10.184.18.215	10.184.45.87	UDP	175	50014 → 50004 Len=133
1760…	300.541280	10.184.45.87	10.184.18.215	UDP	213	50004 → 50014 Len=171
1760…	300.546898	10.184.18.215	10.184.45.87	UDP	158	50014 → 50004 Len=116
1760…	300.550410	10.184.45.87	10.184.18.215	UDP	201	50004 → 50014 Len=159
1760…	300.566804	10.184.18.215	10.184.45.87	UDP	167	50014 → 50004 Len=125
1760…	300.575930	10.184.45.87	10.184.18.215	UDP	220	50004 → 50014 Len=178
1760…	300.586812	10.184.18.215	10.184.45.87	UDP	181	50014 → 50004 Len=139
1760…	300.605113	10.184.45.87	10.184.18.215	UDP	211	50004 → 50014 Len=169
1760…	300.606679	10.184.18.215	10.184.45.87	UDP	156	50014 → 50004 Len=114
1760…	300.612924	10.184.45.87	10.184.18.215	UDP	193	50004 → 50014 Len=151
1760…	300.626690	10.184.18.215	10.184.45.87	UDP	144	50014 → 50004 Len=102

Figure 14: Audio only open and speaking loudly - (packet size 100 - 500 bytes)

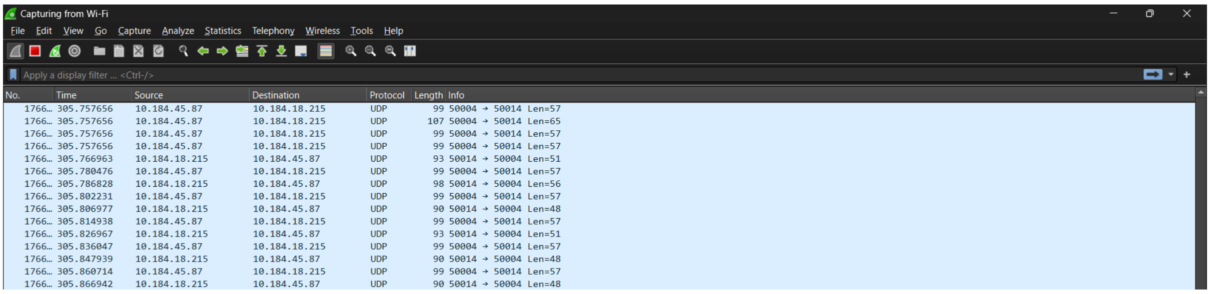


Figure 15: Audio only open and not speaking - (packet size 0 - 80 bytes)

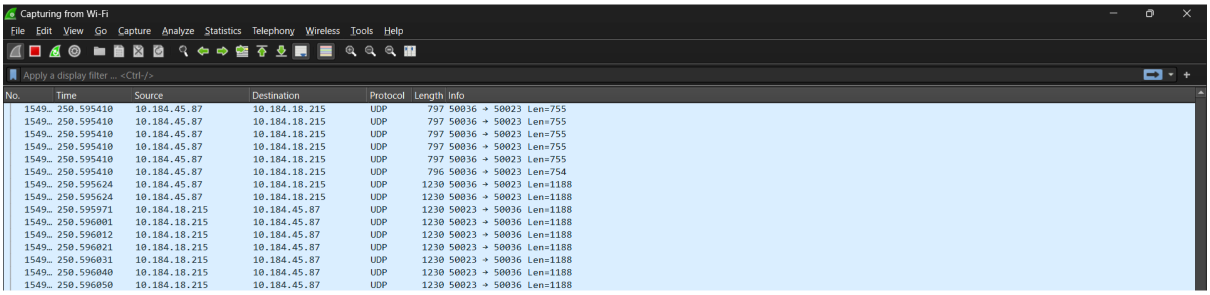


Figure 16: Video Only open - (packet size 600 - 1300 bytes)

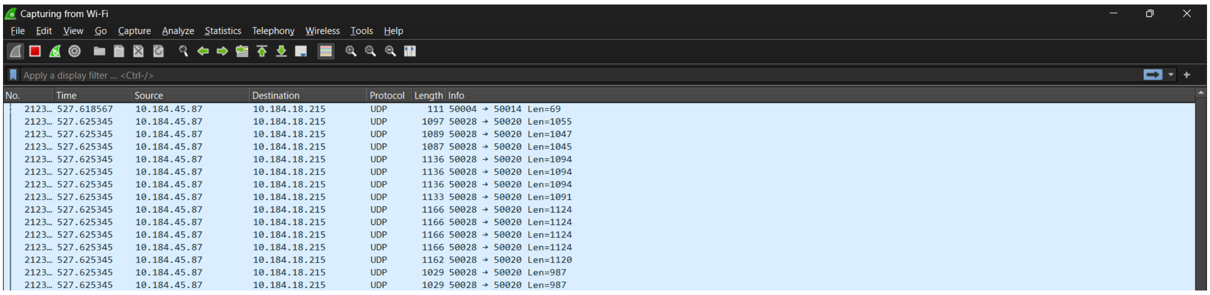


Figure 17: Audio and Video both open - (packet size mixed)

From the above analysis, we can conclude that the video packets are the largest in the range of 600-1300 bytes. Then are the audio packets with high amplitude, which tend to take more space than audio packets with very less amplitude. I have applied filters according to the above range to find the number of packets of each type approximately.

- Video Packets - **37,461 (84.1%)**
  (Filter used - `(ip.addr == 10.184.45.87 && ip.addr == 10.184.18.215) && ((udp.length >= 600) && (udp.length <= 1300))`)

- Audio Packets with no-speaking - **4,424 (9.9%)**
  (Filter used - `(ip.addr == 10.184.45.87 && ip.addr == 10.184.18.215) && (udp.length >= 0 && udp.length <= 100)`)

- Audio Packets with speaking loudly - **2,085 (4.7%)**
  (Filter used - `(ip.addr == 10.184.45.87 && ip.addr == 10.184.18.215) && (udp.length >= 100 && udp.length <= 500)`)

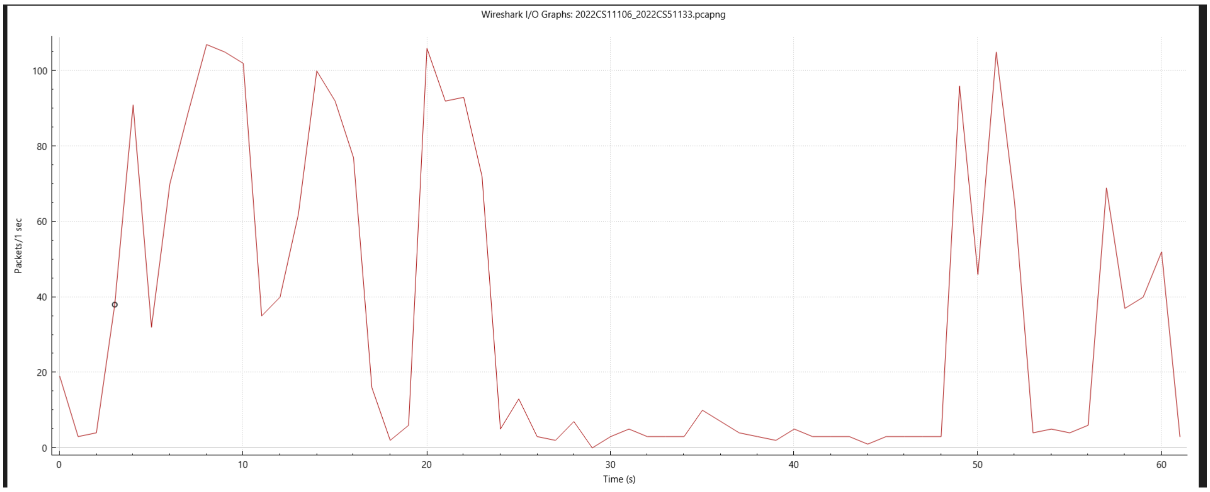


Figure 18: Bandwidth Utilization - Audio Only

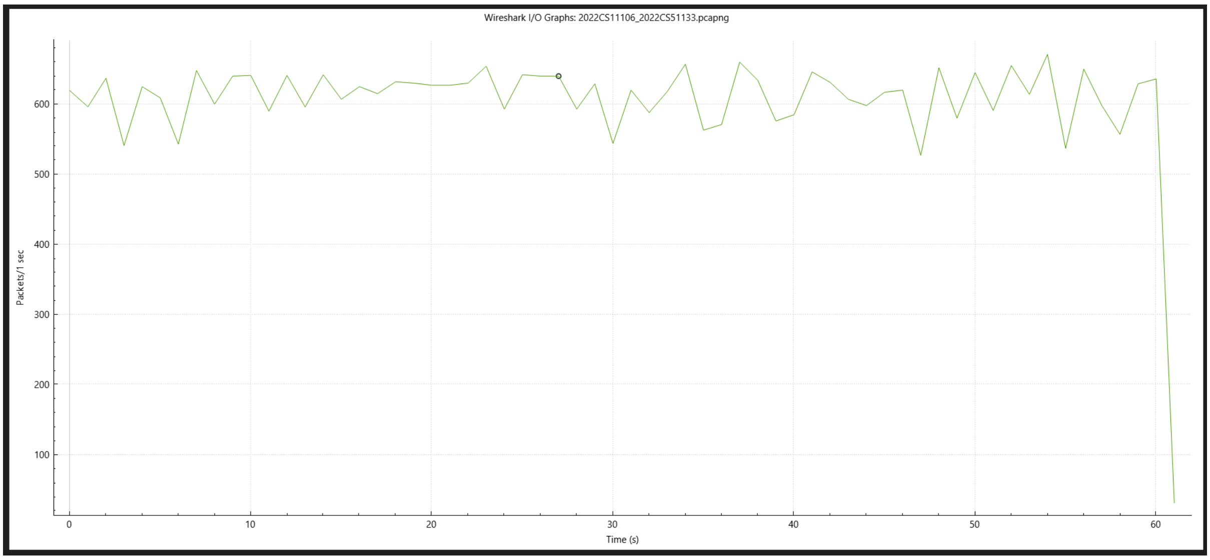


Figure 19: Bandwidth Utilization - Video Only

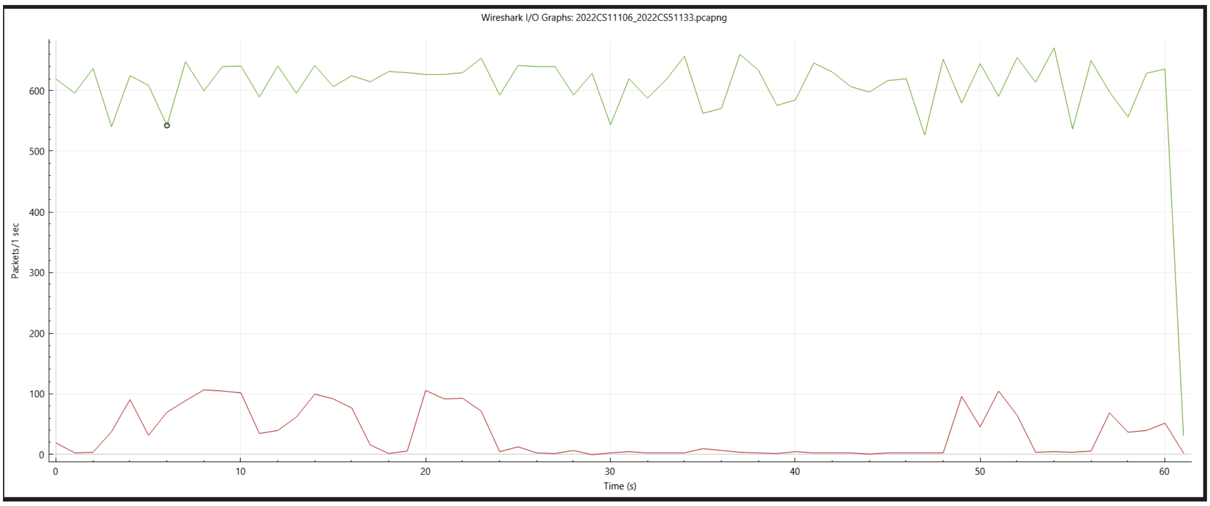


Figure 20: Bandwidth Utilization - Audio & Video Both

3 Traffic Analysis and Network Performance

3.1 Logic

In this part I have coded the file 'speedtest_analysis.py', which takes in a input pcap(pcapng) file and processes it, calculates throughput metrics and generates plots and average speeds for download and upload traffic. I am using the dpkt library for parsing the pcapng file. I am using matplotlib to plot the graph between throughput and time.

1. **Parsing the pcap file**
   The *parce_pcap* function reads the file and extracts all those packages related to port 443(because NDT7 uses this port) and then categorize them into download and upload data based on the direction of the traffic - download or upload. I am also checking other characteristics of the NDT7 speed test, like the TCP protocol and IP6 protocol.
   In the same function I am also making the map *download_data* & *upload_data* which consists of time_stamps as keys and the packet_lengths as values. These dictionaries are returned by the function for further use.

2. **Calculating average speed**
   In the function *calculate_average_speed*, I am computing the average speed of download and upload traffic, considering only those packets where the throughput exceeds a specified threshold. This threshold is threshold_ratio*peak_throughput_value.
   The peak throughput value is calculated by another function *calc_peaks()* which takes download_data/upload_data map as input.

3. **Plotting the average throughput per second**
   I have defined a *plot_throughput()* function which takes in the map consisting of packet lengths corresponding to time_stamps. This map is then passed to a function *calculate_average_throughput_per_second()* which aggregates throughput by second for both download and upload data.
   The results are plotted using matplotlib, creating a time-series graph of throughput.

3.2 A. Percentage of traffic relevant to the speed test

- Total Packets: **71113**
- Total Packets transferred on port 443: **61741**
- Percentage of speedtest traffic in terms of packet transfer: **86.82%**
- Total Packets Size: **91623122**
- Total Packets size transferred from port 443: **81043094**
- Percentage of speedtest traffic in terms of bytes transfer: **88.45%**

3.3 B. Plot of throughput vs time

The download and upload packets are seperated on the basis of the source/destination port. The TCP messages with source port 443 are download-type and those with destination port 443 are upload-type. Then for each integer second from the start of the file, the packets are aggreagted to find the average throughput for that second.

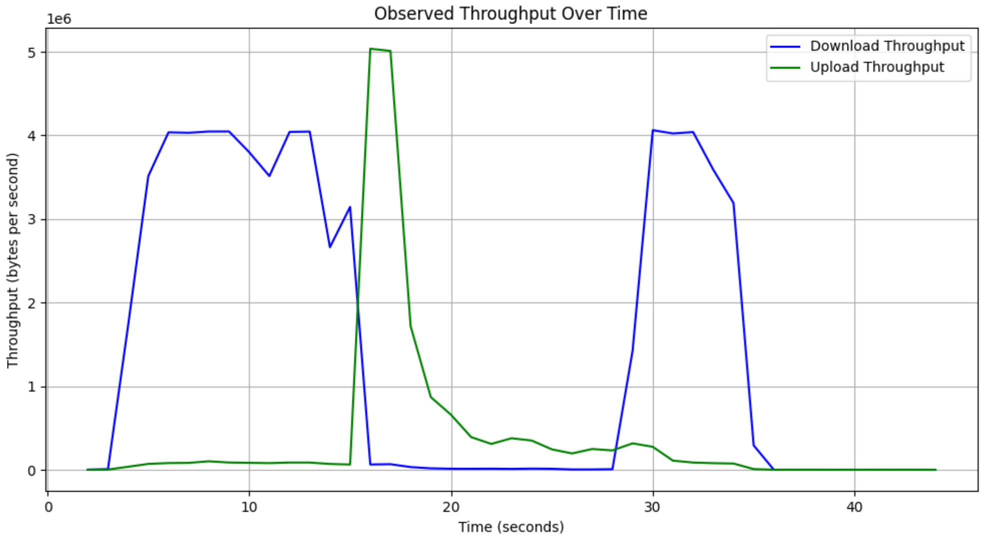


Figure 21: Plot of Throughput vs Time

3.4 C. Average download and upload speeds

I am calculating the average download and upload speeds only for those sections where it is predominant. For this I have defined a threshold throughput = threshold_ratio*peak_throughput. I have tried to approximate the threshold ratio to be 15%, which can be changed as needed.

- **Average Download Speed:** 16.79 Mbps
- **Average Upload Speed:** 33.69 Mbps