

# Assignment 1

Rishabh Kumar (2021CS10103)

Raja Kumar (2021CS10915)

August 2023

## 1 Network Analysis

### 1.1 Traceroute Report: IPv4 Traceroute

Running traceroute from a 4G cellular connection for [www.google.com](http://www.google.com). Below are the IP addresses seen on the path:

```
1 Rishabhs-MacBook-Air:~ rishabhkumar$ traceroute www.google.com
2
3 traceroute to www.google.com (142.250.207.196), 64 hops max, 52 byte packets
4 1 172.20.10.1 (172.20.10.1) 12.605 ms 5.281 ms 6.313 ms
5 2 * * *
6 3 56.8.174.165 (56.8.174.165) 72.937 ms
7 56.8.174.185 (56.8.174.185) 64.685 ms 61.359 ms
8 4 192.168.44.232 (192.168.44.232) 57.096 ms
9 192.168.44.236 (192.168.44.236) 53.689 ms
10 192.168.44.232 (192.168.44.232) 61.728 ms
11 5 * * *
12 6 * * *
13 7 * * *
14 8 * * *
15 9 * * *
16 10 * * *
17 11 * * *
18 12 * * 209.85.148.118 (209.85.148.118) 45.000 ms
19 13 142.251.54.62 (142.251.54.62) 39.695 ms * *
20 14 142.251.52.216 (142.251.52.216) 71.602 ms
21 74.125.244.196 (74.125.244.196) 134.861 ms
22 74.125.37.62 (74.125.37.62) 58.182 ms
23 15 108.170.251.113 (108.170.251.113) 45.494 ms
24 74.125.243.101 (74.125.243.101) 43.371 ms
25 142.251.76.171 (142.251.76.171) 60.299 ms
26 16 108.170.251.97 (108.170.251.97) 34.788 ms 45.499 ms
27 108.170.251.113 (108.170.251.113) 37.550 ms
28 17 del12s10-in-f4.1e100.net (142.250.207.196) 55.977 ms
29 142.251.76.169 (142.251.76.169) 30.721 ms
30 142.251.76.171 (142.251.76.171) 47.168 ms
```

Running traceroute from a 4G cellular connection for [www.iitd.ac.in](http://www.iitd.ac.in). Below are the IP addresses seen on the path:

```
1 Rishabhs-MacBook-Air:~ rishabhkumar$ traceroute www.iitd.ac.in
2
3 traceroute to www.iitd.ac.in (103.27.9.24), 64 hops max, 52 byte packets
4 1 172.20.10.1 (172.20.10.1) 15.527 ms 8.907 ms 7.384 ms
5 2 * * *
6 3 56.8.174.161 (56.8.174.161) 72.320 ms 51.907 ms
7 56.8.174.181 (56.8.174.181) 50.970 ms
8 4 192.168.44.232 (192.168.44.232) 52.408 ms
9 192.168.44.234 (192.168.44.234) 51.014 ms 49.306 ms
10 5 * * *
11 6 * * *
12 7 * * *
13 8 * * *
```

```

14 9 * * *
15 10 * * *
16 11 * 115.244.136.18 (115.244.136.18) 56.332 ms *
17 12 115.244.136.22 (115.244.136.22) 52.395 ms *
18 115.244.136.18 (115.244.136.18) 76.432 ms
19 13 115.244.136.22 (115.244.136.22) 47.255 ms 39.874 ms *
20 14 * * *
21 15 * * *
22 16 * * *
23 17 * * *
24 18 * * *
25 19 * * *
26 20 * * *
27 21 * * *
28 22 * * *
29 23 * * *
30 24 * * *
31 25 * * *
32 26 * * *
33 27 * * *
34 28 * * *
35 29 * * *
36 30 * * *
37 31 * * *
38 32 * * *
39 33 * * *
40 34 * * *
41 35 * * *
42 36 * * *
43 37 * * *
44 38 * * *
45 39 * * *
46 40 * * *
47 41 * * *
48 42 * * *
49 43 * * *
50 44 * * *
51 45 * * *
52 46 * * *
53 47 * * *
54 48 * * *
55 49 * * *
56 50 * * *
57 51 * * *
58 52 * * *
59 53 * * *
60 54 * * *
61 55 * * *
62 56 * * *
63 57 * * *
64 58 * * *
65 59 * * *
66 60 * * *
67 61 * * *
68 62 * * *
69 63 * * *
70 64 * * *

```

### 1.1.1 Observations

For IP Addresses of [www.iitd.ac.in](http://www.iitd.ac.in):

Many hops, including 2, 5, 6, and beyond, exhibit no responses(\* \* \*), possibly due to ICMP blocking or network configuration.

The final hop (64) lacks a specific IP, potentially due to the destination not responding to traceroute. Multiple routes or missing responses suggest complex routing or routers not replying to traceroute requests.

## 1.2 Traceroute Report: IPv6 Traceroute - www.google.com

In MacOS the path defaults to IPv4 and we can force it to IPv6 using `traceroute6` as the command. Below are the IP addresses for path IPv6 for `www.google.com`:

```
1 Rishabhs-MacBook-Air:~ rishabhkumar$ traceroute6 www.google.com
2
3 traceroute6 to www.google.com (2404:6800:4002:822::2004) from 2409:4050:2e3e:159:2cc1
   :184d:90bd:7679, 64 hops max, 12 byte packets
4 1 2409:4050:2e3e:159:a98c:60ac:d223:82b 5.456 ms 5.617 ms 5.204 ms
5 2 * * *
6 3 2405:200:331:eeee:20::484 76.370 ms 62.703 ms 32.710 ms
7 4 2405:200:801:300::e74 53.092 ms
8   2405:200:801:300::e78 56.470 ms
9   2405:200:801:300::e74 49.679 ms
10 5 * * *
11 6 * * *
12 7 * 2405:203:10:8200:130:26:30:99 70.201 ms *
13 8 2001:4860:1:1::1a34 63.179 ms
14   2001:4860:1:1::17ae 53.816 ms 40.271 ms
15 9 2404:6800:8023::1 32.953 ms
16   2404:6800:8107::1 94.093 ms
17   2404:6800:8010::1 55.206 ms
18 10 2001:4860:0:1::306a 49.495 ms
19   del12s05-in-x04.1e100.net 169.147 ms
20   2001:4860:0:1::5e5e 175.452 ms
```

Observations for traceroute path to IPv4:

Private IP address spaces like 10.0.0.0 to 10.255.255.255 (10.0.0.0/8), 172.16.0.0 to 172.31.255.255 (172.16.0.0/12), and 192.168.0.0 to 192.168.255.255 (192.168.0.0/16) are commonly used for internal networks. They are not routable on the public internet. In traceroute outputs, we see these addresses, it's because we are passing through routers within private networks.

## 1.3 Investigating Ping

Based on the results below, it is clear that my system's maximum supported ping packet size is around 65,507 bytes, but even this size encounters issues and timeouts during transmission. This behavior might be due to network restrictions, or other factors affecting the packet size that can be sent successfully.

```
1 Rishabhs-MacBook-Air:~ rishabhkumar$ ping -s 100000 www.google.com
2
3 ping: packet size too large: 100000 > 65507
4 Rishabhs-MacBook-Air:~ rishabhkumar$ ping -s 65507 www.google.com
5 PING www.google.com (142.250.207.196): 65507 data bytes
6 ping: sendto: Message too long
7 ping: sendto: Message too long
8 Request timeout for icmp_seq 0
9 ping: sendto: Message too long
10 Request timeout for icmp_seq 1
11 ^C
12 --- www.google.com ping statistics ---
13 3 packets transmitted, 0 packets received, 100.0% packet loss
14 Rishabhs-MacBook-Air:~ rishabhkumar$ ping -s 65508 www.google.com
15 ping: packet size too large: 65508 > 65507
```

## 2 Replicating the traceroute functionality using ping

Below is the bash script with the help of which we can replicate the functionality of traceroute using the ping command with custom TTL values.

```
1 #!/bin/bash
2
3 destination=$1
4
5 if [ -z "$destination" ]; then
6     echo "Usage: $0 <destination>"
7     exit 1
8 fi
```

```

9
10 max_hops=30
11
12 for ttl in $(seq 1 $max_hops); do
13     ping_result=$(ping -c 1 -t $ttl $destination)
14     echo "$ttl: $ping_result"
15
16     if echo "$ping_result" | grep -q "Time to live exceeded"; then
17         continue
18     fi
19
20     if echo "$ping_result" | grep -q "64 bytes from"; then
21         echo "Traceroute completed"
22         break
23     fi
24 done

```

Executing the above script on [www.google.com](http://www.google.com). Below is the result:

```

1 Rishabhs-MacBook-Air:~ rishabhkumar$ ./custom_traceroute.sh www.google.com
2 1: PING www.google.com (142.250.206.132): 56 data bytes
3 64 bytes from 142.250.206.132: icmp_seq=0 ttl=53 time=48.268 ms
4 --- www.google.com ping statistics ---
5 1 packets transmitted, 1 packet received, 0.0% packet loss
6 round-trip min/avg/max/stddev = 48.268/48.268/48.268/0.000 ms
7 Traceroute completed

```

## 3 Internet Architecture

### 3.1 Number of Hops to Destinations

Destination	Hops to Germany	Hops to USA	Hops to Local Device
<a href="http://www.utah.edu">www.utah.edu</a>	30	30	64
<a href="http://www.uct.ac.za">www.uct.ac.za</a>	30	30	64
<a href="http://www.iitd.ac.in">www.iitd.ac.in</a>	18	30	4
<a href="http://www.google.com">www.google.com</a>	14	12	11
<a href="http://www.facebook.com">www.facebook.com</a>	11	13	13

Table 1: Number of hops from traceroute sources to destinations

#### 3.1.1 Observations based on above results

**Geographical Proximity and Hops:** Geographical proximity between source and destination does not always translate directly into fewer hops. While shorter distances might generally result in fewer hops, the network infrastructure, routing policies, and the organization of the network can play significant roles in determining the number of hops.

**Google vs. Facebook:** In terms of traceroute hops, the paths to both Google and Facebook have several intermediate hops across different networks. However, there's no distinct trend suggesting that one has consistently fewer hops than the other. Both routes seem to traverse various networks before reaching the target.

### 3.2 Latencies between the traceroute sources and the web-servers

#### 3.2.1 Observations

Yes, generally, more hops in a traceroute can result in higher latency. Each hop is a network router or gateway that introduces some level of delay to the traversal of data. As the traceroute packets pass through more hops, the cumulative latency can increase. However, it's important to note that other factors like network congestion and routing inefficiencies can also contribute to latency.

Destination	Germany	USA	Local Device
www.utah.edu	146.710 ms	64.483 ms	314.254 ms
www.uct.ac.za	210.162 ms	191.863 ms	469.471 ms
www.iitd.ac.in	156.209 ms	227.900 ms	4.324 ms
www.google.com	3.48 ms	2.952 ms	7.851 ms
www.facebook.com	8.33 ms	3.870 ms	33.079 ms

Table 2: Latencies of round-trip from traceroute sources to destinations

It's important to note that while latency does tend to increase with more hops, it's not the only factor determining the quality of a network connection. The speed and capacity of each hop, the overall network architecture, and the efficiency of routing algorithms also play crucial roles in determining the final latency and connection quality.

### 3.3 Web-Servers vs IP Addresses

The educational sites like `www.utah.edu` and `www.uct.ac.za` have same IP addresses irrespective of from where you do a traceroute to them. The reason some web-servers are resolved to different IP addresses when queried from different parts of the world is due to a network phenomenon called Anycast.

Anycast is a routing technique where the same IP address is advertised from multiple physical locations. When a user sends a request to that IP address, the routing infrastructure directs the request to the nearest physical location advertising that IP. This is often used for content delivery networks (CDNs) and critical services to improve performance and reliability.

### 3.4 IP Addresses and Paths

Traceroutes from the same starting point to different IP addresses associated with the same web-server can indeed have different paths. This variability in paths can be due to a variety of factors, including the network topology, routing decisions, and current network conditions. Upon analysing `www.google.com`, we observed that the initial hops might be similar, as they are both part of Google's infrastructure. However, the paths could diverge in later hops due to various routing decisions. One of the paths might appear longer due to traffic conditions or routing changes at that particular moment.

When I conducted a traceroute on the IP addresses of hops 1 and 2, the number of hops reported was only 1-2. However, as I progressed to later hops, the number of hops increased to the maximum value of 64. This observation indicates a longer path taken by the traceroute packets as they traverse through the network.

```

1 Rishabhs-MacBook-Air:~ rishabhkumar$ traceroute www.google.com
2 traceroute to www.google.com (142.250.207.196), 64 hops max, 52 byte packets
3  1  10.184.0.13 (10.184.0.13) 49.674 ms 32.649 ms 17.633 ms
4  2  10.254.175.1 (10.254.175.1) 5.951 ms
5  10.254.175.5 (10.254.175.5) 6.986 ms 8.705 ms
6  3  10.255.1.34 (10.255.1.34) 8.077 ms 9.475 ms 4.768 ms
7  4  10.119.233.65 (10.119.233.65) 4.824 ms 9.034 ms 15.277 ms
8  5  * * *
9  6  * * *
10 7  10.119.234.162 (10.119.234.162) 6.145 ms 9.051 ms 4.457 ms
11 8  72.14.195.56 (72.14.195.56) 7.741 ms
12 72.14.194.160 (72.14.194.160) 6.233 ms
13 72.14.195.56 (72.14.195.56) 24.181 ms
14 9  108.170.251.97 (108.170.251.97) 7.041 ms
15 108.170.251.113 (108.170.251.113) 8.350 ms
16 108.170.251.97 (108.170.251.97) 6.577 ms
17 10 142.251.76.169 (142.251.76.169) 8.271 ms 8.353 ms 6.129 ms
18 11 del12s10-in-f4.1e100.net (142.250.207.196) 6.927 ms 6.163 ms 8.609 ms

```

### 3.5 Part E

An indication of a lack of direct peering with Google and Facebook can be derived from a higher number of hops and increased latencies. On the basis of above latencies and number of hops found we can say that India, Germany and USA definitely have their local ISPs directly peered with Google and Facebook.

Let's have a look on tracerouting to Google from traceroute servers of Canada.

```
1 traceroute to www.google.com (172.253.62.104), 30 hops max, 60 byte packets
2 1  gi0-0-0-16.224.nr11.b011027-0.yyz02.atlas.cogentco.com (66.250.250.41) 0.840 ms
   0.826 ms
3 2  te0-0-2-2.agr11.yyz02.atlas.cogentco.com (154.24.42.65) 0.962 ms te0-0-2-2.agr12.
   yyz02.atlas.cogentco.com (154.24.42.69) 0.911 ms
4 3  te0-0-0-9.ccr32.yyz02.atlas.cogentco.com (154.54.3.89) 0.765 ms te0-0-1-9.ccr31.
   yyz02.atlas.cogentco.com (154.54.3.145) 0.855 ms
5 4  be2994.ccr22.cle04.atlas.cogentco.com (154.54.31.233) 7.446 ms 7.395 ms
6 5  be2718.ccr42.ord01.atlas.cogentco.com (154.54.7.129) 14.012 ms 13.961 ms
7 6  be2832.ccr22.mci01.atlas.cogentco.com (154.54.44.169) 25.794 ms 25.565 ms
8 7  be2433.ccr32.dfw01.atlas.cogentco.com (154.54.3.213) 35.820 ms be2432.ccr31.dfw01.
   atlas.cogentco.com (154.54.3.133) 35.778 ms
9 8  be2764.ccr41.dfw03.atlas.cogentco.com (154.54.47.214) 35.660 ms be2763.ccr41.dfw03.
   atlas.cogentco.com (154.54.28.74) 35.803 ms
10 9  tata.dfw03.atlas.cogentco.com (154.54.12.106) 35.433 ms 35.385 ms
11 10 66.110.56.139 (66.110.56.139) 35.405 ms 209.85.172.106 (209.85.172.106) 37.478 ms
12 11 * *
13 12 108.170.240.129 (108.170.240.129) 36.606 ms 142.251.60.52 (142.251.60.52) 35.583
   ms
14 13 108.170.252.131 (108.170.252.131) 35.966 ms 108.170.240.145 (108.170.240.145)
   35.918 ms
15 14 216.239.63.253 (216.239.63.253) 36.024 ms *
16 15 108.170.229.87 (108.170.229.87) 37.795 ms *
17 16 142.251.230.208 (142.251.230.208) 62.253 ms 142.251.67.138 (142.251.67.138) 46.467
   ms
18 17 216.239.40.131 (216.239.40.131) 64.153 ms 216.239.40.133 (216.239.40.133) 44.915
   ms
19 18 192.178.44.102 (192.178.44.102) 44.954 ms 216.239.56.72 (216.239.56.72) 44.079 ms
20 19 * 142.251.77.138 (142.251.77.138) 48.302 ms
21 20 142.251.244.142 (142.251.244.142) 45.021 ms 142.251.244.162 (142.251.244.162)
   50.837 ms
22 21 72.14.239.165 (72.14.239.165) 45.177 ms 172.253.68.81 (172.253.68.81) 44.891 ms
23 22 * *
24 23 * *
25 24 * *
26 25 * *
27 26 * *
28 27 * *
29 28 * bc-in-f104.1e100.net (172.253.62.104) 46.245 ms
```

Here is the tracerouting of Facebook from traceroute server of Greece.

```
1 traceroute to www.facebook.com (31.13.81.36), 30 hops max, 60 byte packets
2 1  gi0-0-0-11.4.agr11.ath01.atlas.cogentco.com (130.117.254.217) 0.856 ms 0.915 ms
3 2  be3233.rcr21.ath01.atlas.cogentco.com (130.117.0.90) 0.747 ms 0.850 ms
4 3  be2047.rcr51.skg01.atlas.cogentco.com (154.54.37.141) 7.743 ms 7.837 ms
5 4  be2046.ccr31.sof02.atlas.cogentco.com (130.117.0.189) 12.020 ms 12.031 ms
6 5  be3421.ccr51.beg03.atlas.cogentco.com (130.117.0.94) 17.111 ms 17.007 ms
7 6  be3422.ccr31.bud01.atlas.cogentco.com (130.117.0.125) 22.442 ms 22.336 ms
8 7  be3261.ccr21.bts01.atlas.cogentco.com (130.117.3.137) 24.652 ms be3263.ccr22.bts01.
   atlas.cogentco.com (154.54.59.177) 24.734 ms
9 8  be2477.ccr21.waw01.atlas.cogentco.com (130.117.51.2) 37.869 ms be2478.ccr21.waw01.
   atlas.cogentco.com (130.117.51.58) 37.822 ms
10 9  be2486.rcr21.b016833-0.waw01.atlas.cogentco.com (154.54.37.42) 38.181 ms 57.808 ms
11 10 149.14.232.58 (149.14.232.58) 38.200 ms 149.14.232.42 (149.14.232.42) 37.966 ms
12 11 po204.asw01.waw1.tfbnw.net (147.75.216.152) 37.441 ms po205.asw02.waw1.tfbnw.net
   (147.75.216.162) 37.503 ms
13 12 po233.psw04.waw1.tfbnw.net (147.75.216.115) 37.472 ms po202.psw03.waw1.tfbnw.net
   (147.75.216.57) 37.418 ms
14 13 157.240.38.149 (157.240.38.149) 37.391 ms 157.240.38.245 (157.240.38.245) 37.485
   ms
15 14 edge-star-mini-shv-01-waw1.facebook.com (31.13.81.36) 37.229 ms 37.395 ms
```

Here, the presence of numerous hops and relatively high latencies suggests that the connection might not have direct peering with Google's servers for Canada and Facebook's server for Greece. The initial queries appear to be routed to the UK before reaching the destination, indicating a potential absence of local ISP peering.

## 4 Packet Analysis

### 4.1 Part A

Request time : 62.415374 seconds  
Response time : 62.429513 seconds  
Total time taken : 0.014139 seconds

508	62.415374	10.184.4.80	10.10.1.4	DNS	76	Standard query 0x61f2 A act4d.iitd.ac.in
509	62.415524	10.184.4.80	10.10.1.4	DNS	76	Standard query 0x670e HTTPS act4d.iitd.ac.in
510	62.429502	10.10.1.4	10.184.4.80	DNS	92	Standard query response 0x61f2 A act4d.iitd.ac.in A 10.237.26.108
512	62.429513	10.10.1.4	10.184.4.80	DNS	129	Standard query response 0x670e HTTPS act4d.iitd.ac.in SOA intdns.iitd.ac.in

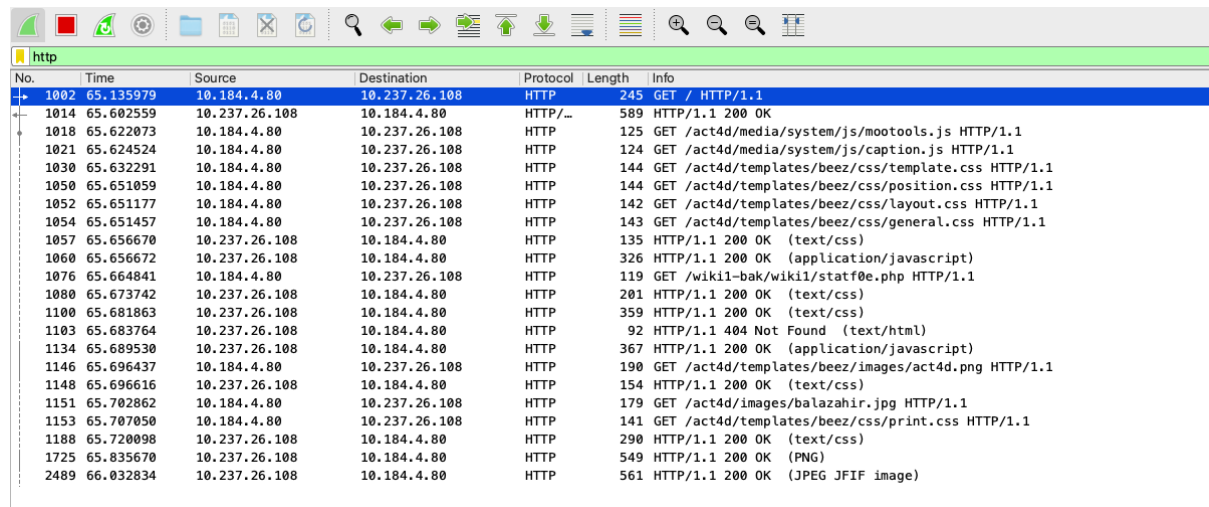
Figure 1: Reference Image for result

### 4.2 Part B

When I applied an “http” filter, I found many different packets. These packets include both HTTP requests and HTTP responses. Each HTTP request includes resources like HTML files, images, CSS file, javascript file, text, PNG, JPEG etc.

From this, we can conclude that web pages are structured as a collection of different resources, as given above. Each of these resources contributes to the final page.

Browsers use progressive rendering to display content as soon as it’s available. Images are fetched as separate resources towards the end, suggesting that more essential components of pages (like text) are prioritized than others.



The image shows a Wireshark packet capture window with the filter 'http' applied. The packet list pane displays 24 packets, all of which are HTTP requests or responses. The packet details pane shows the selected packet (No. 1002) as a GET request for '/ HTTP/1.1' from 10.184.4.80 to 10.237.26.108. The packet bytes pane shows the raw data of the selected packet.

No.	Time	Source	Destination	Protocol	Length	Info
1002	65.135979	10.184.4.80	10.237.26.108	HTTP	245	GET / HTTP/1.1
1014	65.602559	10.237.26.108	10.184.4.80	HTTP	589	HTTP/1.1 200 OK
1018	65.622073	10.184.4.80	10.237.26.108	HTTP	125	GET /act4d/media/system/js/mootools.js HTTP/1.1
1021	65.624524	10.184.4.80	10.237.26.108	HTTP	124	GET /act4d/media/system/js/caption.js HTTP/1.1
1030	65.632291	10.184.4.80	10.237.26.108	HTTP	144	GET /act4d/templates/beeze/css/template.css HTTP/1.1
1050	65.651059	10.184.4.80	10.237.26.108	HTTP	144	GET /act4d/templates/beeze/css/position.css HTTP/1.1
1052	65.651177	10.184.4.80	10.237.26.108	HTTP	142	GET /act4d/templates/beeze/css/layout.css HTTP/1.1
1054	65.651457	10.184.4.80	10.237.26.108	HTTP	143	GET /act4d/templates/beeze/css/general.css HTTP/1.1
1057	65.656670	10.237.26.108	10.184.4.80	HTTP	135	HTTP/1.1 200 OK (text/css)
1060	65.656672	10.237.26.108	10.184.4.80	HTTP	326	HTTP/1.1 200 OK (application/javascript)
1076	65.664841	10.184.4.80	10.237.26.108	HTTP	119	GET /wiki1-bak/wiki1/statf0e.php HTTP/1.1
1080	65.673742	10.237.26.108	10.184.4.80	HTTP	201	HTTP/1.1 200 OK (text/css)
1100	65.681863	10.237.26.108	10.184.4.80	HTTP	359	HTTP/1.1 200 OK (text/css)
1103	65.683764	10.237.26.108	10.184.4.80	HTTP	92	HTTP/1.1 404 Not Found (text/html)
1134	65.689530	10.237.26.108	10.184.4.80	HTTP	367	HTTP/1.1 200 OK (application/javascript)
1146	65.696437	10.184.4.80	10.237.26.108	HTTP	190	GET /act4d/templates/beeze/images/act4d.png HTTP/1.1
1148	65.696616	10.237.26.108	10.184.4.80	HTTP	154	HTTP/1.1 200 OK (text/css)
1151	65.702862	10.184.4.80	10.237.26.108	HTTP	179	GET /act4d/images/balazahir.jpg HTTP/1.1
1153	65.707050	10.184.4.80	10.237.26.108	HTTP	141	GET /act4d/templates/beeze/css/print.css HTTP/1.1
1188	65.720098	10.237.26.108	10.184.4.80	HTTP	290	HTTP/1.1 200 OK (text/css)
1725	65.835670	10.237.26.108	10.184.4.80	HTTP	549	HTTP/1.1 200 OK (PNG)
2489	66.032834	10.237.26.108	10.184.4.80	HTTP	561	HTTP/1.1 200 OK (JPEG JFIF image)

Figure 2: Reference Image for result

### 4.3 Part C

My laptop to server: 6 connections: Port Numbers: 626(15, 16, 24, 25, 26, 27) to 80 Server to My laptop: 6 connections: Port Numbers: 80 to 626(15, 16, 24, 25, 26, 27) - the reverse of the previous one.

The number of TCP connections is not the same as the number of HTTP requests. A single TCP connection can be reused for multiple HTTP requests. Yes, For multiple contents objects to be fetched over the same TCP connection. It has been used for improving performance and reducing latency.



No.	Time	Source	Destination	Protocol	Length	Info
511	62.429511	10.237.26.108	10.184.4.80	TCP	74	80 → 62616 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=536 SACK_PERM TSval=1757894347 TSecr=3028494414
513	62.429515	10.237.26.108	10.184.4.80	TCP	74	80 → 62615 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=536 SACK_PERM TSval=1757894347 TSecr=4235976456
1003	65.143467	10.237.26.108	10.184.4.80	TCP	66	80 → 62616 [ACK] Seq=1 Ack=525 Win=6848 Len=0 TSval=1757895027 TSecr=3028497140
1004	65.147397	10.237.26.108	10.184.4.80	TCP	66	80 → 62616 [ACK] Seq=1 Ack=704 Win=7936 Len=0 TSval=1757895028 TSecr=3028497140
1005	65.596829	10.237.26.108	10.184.4.80	TCP	590	80 → 62616 [ACK] Seq=1 Ack=704 Win=7936 Len=524 TSval=1757895113 TSecr=3028497140 [TCP segment of a
1006	65.596837	10.237.26.108	10.184.4.80	TCP	590	80 → 62616 [ACK] Seq=525 Ack=704 Win=7936 Len=524 TSval=1757895113 TSecr=3028497140 [TCP segment of
1007	65.596839	10.237.26.108	10.184.4.80	TCP	590	80 → 62616 [ACK] Seq=1049 Ack=704 Win=7936 Len=524 TSval=1757895113 TSecr=3028497140 [TCP segment of
1008	65.596840	10.237.26.108	10.184.4.80	TCP	590	80 → 62616 [ACK] Seq=1573 Ack=704 Win=7936 Len=524 TSval=1757895113 TSecr=3028497140 [TCP segment of
1011	65.602550	10.237.26.108	10.184.4.80	TCP	590	80 → 62616 [ACK] Seq=2097 Ack=704 Win=7936 Len=524 TSval=1757895142 TSecr=3028497601 [TCP segment of
1012	65.602556	10.237.26.108	10.184.4.80	TCP	590	80 → 62616 [ACK] Seq=2621 Ack=704 Win=7936 Len=524 TSval=1757895142 TSecr=3028497601 [TCP segment of
1013	65.602557	10.237.26.108	10.184.4.80	TCP	590	80 → 62616 [ACK] Seq=3145 Ack=704 Win=7936 Len=524 TSval=1757895142 TSecr=3028497601 [TCP segment of
1014	65.602559	10.237.26.108	10.184.4.80	HTTP/_	589	HTTP/1.1. 200 OK
1025	65.628747	10.237.26.108	10.184.4.80	TCP	66	80 → 62616 [ACK] Seq=4192 Ack=1228 Win=8960 Len=0 TSval=1757895148 TSecr=3028497626
1026	65.631948	10.237.26.108	10.184.4.80	TCP	74	80 → 62624 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=536 SACK_PERM TSval=1757895149 TSecr=3670636945
1027	65.631952	10.237.26.108	10.184.4.80	TCP	66	80 → 62616 [ACK] Seq=4192 Ack=1287 Win=8960 Len=0 TSval=1757895149 TSecr=3028497626
1031	65.646153	10.237.26.108	10.184.4.80	TCP	74	80 → 62625 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=536 SACK_PERM TSval=1757895150 TSecr=3022032354
1032	65.646157	10.237.26.108	10.184.4.80	TCP	74	80 → 62626 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=536 SACK_PERM TSval=1757895150 TSecr=3119899352
1033	65.646158	10.237.26.108	10.184.4.80	TCP	66	80 → 62615 [ACK] Seq=1 Ack=525 Win=6848 Len=0 TSval=1757895150 TSecr=4235979671
1034	65.646158	10.237.26.108	10.184.4.80	TCP	74	80 → 62627 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=536 SACK_PERM TSval=1757895150 TSecr=2081531520
1035	65.646159	10.237.26.108	10.184.4.80	TCP	66	80 → 62615 [ACK] Seq=1 Ack=583 Win=6848 Len=0 TSval=1757895150 TSecr=4235979671
1036	65.646159	10.237.26.108	10.184.4.80	TCP	590	80 → 62616 [ACK] Seq=4192 Ack=1287 Win=8960 Len=524 TSval=1757895152 TSecr=3028497626 [TCP segment o
1037	65.646160	10.237.26.108	10.184.4.80	TCP	590	80 → 62616 [ACK] Seq=4716 Ack=1287 Win=8960 Len=524 TSval=1757895152 TSecr=3028497626 [TCP segment o
1038	65.646160	10.237.26.108	10.184.4.80	TCP	590	80 → 62616 [ACK] Seq=5248 Ack=1287 Win=8960 Len=524 TSval=1757895152 TSecr=3028497626 [TCP segment o
1039	65.646161	10.237.26.108	10.184.4.80	TCP	590	80 → 62616 [ACK] Seq=5764 Ack=1287 Win=8960 Len=524 TSval=1757895152 TSecr=3028497626 [TCP segment o
1040	65.646162	10.237.26.108	10.184.4.80	TCP	590	80 → 62616 [ACK] Seq=6288 Ack=1287 Win=8960 Len=524 TSval=1757895152 TSecr=3028497626 [TCP segment o
1041	65.646163	10.237.26.108	10.184.4.80	TCP	590	80 → 62616 [ACK] Seq=6812 Ack=1287 Win=8960 Len=524 TSval=1757895152 TSecr=3028497626 [TCP segment o
1042	65.646163	10.237.26.108	10.184.4.80	TCP	66	80 → 62624 [ACK] Seq=1 Ack=525 Win=6848 Len=0 TSval=1757895152 TSecr=3670636954
1043	65.646164	10.237.26.108	10.184.4.80	TCP	66	80 → 62624 [ACK] Seq=1 Ack=603 Win=6848 Len=0 TSval=1757895152 TSecr=3670636954
1055	65.656665	10.237.26.108	10.184.4.80	TCP	590	80 → 62624 [ACK] Seq=1 Ack=603 Win=6848 Len=524 TSval=1757895154 TSecr=3670636954 [TCP segment of a
1056	65.656669	10.237.26.108	10.184.4.80	TCP	590	80 → 62624 [ACK] Seq=525 Ack=603 Win=6848 Len=524 TSval=1757895154 TSecr=3670636954 [TCP segment of
1057	65.656670	10.237.26.108	10.184.4.80	HTTP	135	HTTP/1.1. 200 OK (text/css)
1058	65.656670	10.237.26.108	10.184.4.80	TCP	590	80 → 62615 [ACK] Seq=1 Ack=583 Win=6848 Len=524 TSval=1757895154 TSecr=4235979671 [TCP segment of a
1059	65.656671	10.237.26.108	10.184.4.80	TCP	590	80 → 62615 [ACK] Seq=525 Ack=583 Win=6848 Len=524 TSval=1757895154 TSecr=4235979671 [TCP segment of
1060	65.656672	10.237.26.108	10.184.4.80	HTTP	326	HTTP/1.1. 200 OK (application/javascript)
1063	65.661666	10.237.26.108	10.184.4.80	TCP	590	80 → 62616 [ACK] Seq=7336 Ack=1287 Win=8960 Len=524 TSval=1757895155 TSecr=3028497651 [TCP segment o
1064	65.661674	10.237.26.108	10.184.4.80	TCP	590	80 → 62616 [ACK] Seq=7860 Ack=1287 Win=8960 Len=524 TSval=1757895155 TSecr=3028497651 [TCP segment o
1065	65.661675	10.237.26.108	10.184.4.80	TCP	590	80 → 62616 [ACK] Seq=8384 Ack=1287 Win=8960 Len=524 TSval=1757895155 TSecr=3028497651 [TCP segment o
1066	65.661676	10.237.26.108	10.184.4.80	TCP	590	80 → 62616 [ACK] Seq=8908 Ack=1287 Win=8960 Len=524 TSval=1757895155 TSecr=3028497651 [TCP segment o
1067	65.661677	10.237.26.108	10.184.4.80	TCP	590	80 → 62616 [ACK] Seq=9432 Ack=1287 Win=8960 Len=524 TSval=1757895155 TSecr=3028497651 [TCP segment o
1068	65.661678	10.237.26.108	10.184.4.80	TCP	590	80 → 62616 [ACK] Seq=9956 Ack=1287 Win=8960 Len=524 TSval=1757895155 TSecr=3028497651 [TCP segment o
1069	65.661679	10.237.26.108	10.184.4.80	TCP	590	80 → 62616 [ACK] Seq=10480 Ack=1287 Win=8960 Len=524 TSval=1757895155 TSecr=3028497651 [TCP segment o
1070	65.661680	10.237.26.108	10.184.4.80	TCP	66	80 → 62625 [ACK] Seq=1 Ack=525 Win=6848 Len=0 TSval=1757895156 TSecr=3022032380
1071	65.661680	10.237.26.108	10.184.4.80	TCP	66	80 → 62625 [ACK] Seq=1 Ack=603 Win=6848 Len=0 TSval=1757895156 TSecr=3022032380

Figure 3: Reference Image for result

## 4.4 Part D

I could not see any HTTP request traffic being generated. This is because Indian Express uses HTTPS, which is secure and encrypted.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	13.126.221.44	10.184.4.80	HTTP	135	GET / HTTP/1.1

Figure 4: Reference Image for result