COL334 Networks Assignment-1 Report

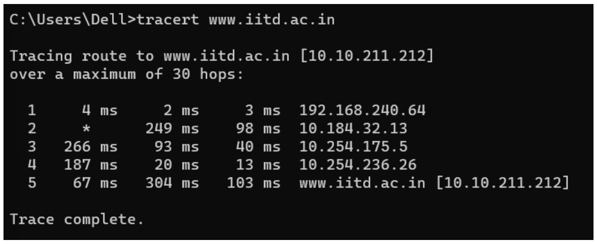
Kanishka Gajbhiye - 2021CS50131
Mrunal Kadhane - 2021CS10109

We performed several tasks to familiarise ourselves with handy tools such as traceroute, nmap, wireshark, ifconfig, etc. used in computer network diagnostics.

1 Network Analysis

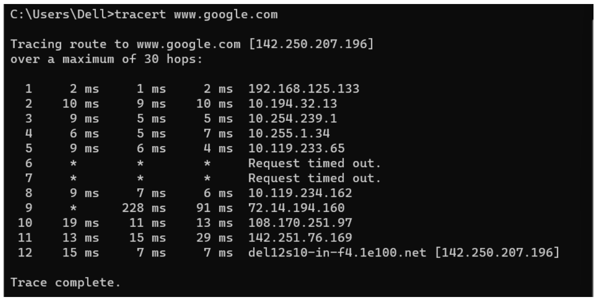
a Some snapshots showing use of ”traceroute(tracert)” command

Used cellular network (Airtel 4G)



```
C:\Users\Dell>tracert www.iitd.ac.in

Tracing route to www.iitd.ac.in [10.10.211.212]
over a maximum of 30 hops:

  1     4 ms     2 ms     3 ms  192.168.240.64
  2     *      249 ms    98 ms  10.184.32.13
  3   266 ms    93 ms    40 ms  10.254.175.5
  4   187 ms    20 ms    13 ms  10.254.236.26
  5    67 ms   304 ms   103 ms  www.iitd.ac.in [10.10.211.212]

Trace complete.
```

Figure 1: Trace to www.iitd.ac.in



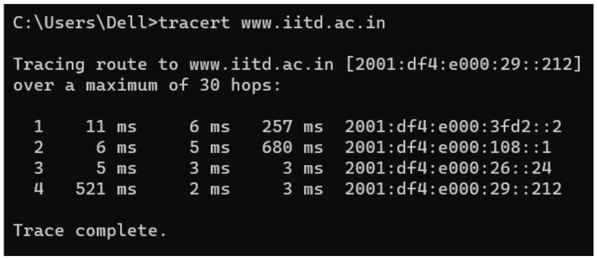
```
C:\Users\Dell>tracert www.google.com

Tracing route to www.google.com [142.250.207.196]
over a maximum of 30 hops:

  1     2 ms     1 ms     2 ms  192.168.125.133
  2    10 ms     9 ms    10 ms  10.194.32.13
  3     9 ms     5 ms     5 ms  10.254.239.1
  4     6 ms     5 ms     7 ms  10.255.1.34
  5     9 ms     6 ms     4 ms  10.119.233.65
  6     *        *        *     Request timed out.
  7     *        *        *     Request timed out.
  8     9 ms     7 ms     6 ms  10.119.234.162
  9     *      228 ms    91 ms  72.14.194.160
 10    19 ms    11 ms    13 ms  108.170.251.97
 11    13 ms    15 ms    29 ms  142.251.76.169
 12    15 ms     7 ms     7 ms  del12s10-in-f4.1e100.net [142.250.207.196]

Trace complete.
```

Figure 2: Trace to www.google.com

b Some interesting observations

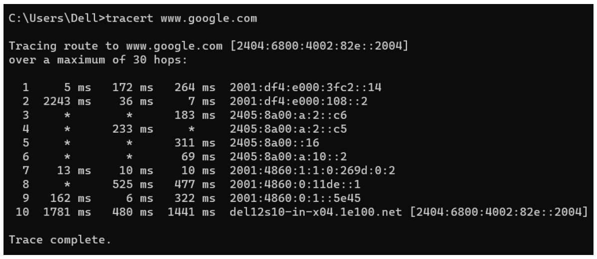
1. When running traceroute to www.iitd.ac.in and google.com sometimes the **path defaulted to IPv6**. Many modern networks are designed to support both IPv4 and IPv6 (dual-stack networks). Whether to use IPv6 or IPv4 depends on preference, network policies, network availability, etc. We can force IPv4/IPv6 using command options:

i. -4 for IPv4
ii. -6 for IPv6



```
C:\Users\Dell>tracert www.iitd.ac.in

Tracing route to www.iitd.ac.in [2001:df4:e000:29::212]
over a maximum of 30 hops:

  1    11 ms     6 ms   257 ms  2001:df4:e000:3fd2::2
  2     6 ms     5 ms   680 ms  2001:df4:e000:108::1
  3     5 ms     3 ms     3 ms  2001:df4:e000:26::24
  4   521 ms     2 ms     3 ms  2001:df4:e000:29::212

Trace complete.
```

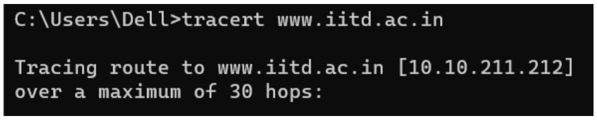
Figure 3: Trace to www.iitd.ac.in (IPv6)



```
C:\Users\Dell>tracert www.google.com

Tracing route to www.google.com [2404:6800:4002:82e::2004]
over a maximum of 30 hops:

  1     5 ms   172 ms   264 ms  2001:df4:e000:3fc2::14
  2  2243 ms    36 ms     7 ms  2001:df4:e000:108::2
  3     *        *      183 ms  2405:8a00:a:2::c6
  4     *      233 ms     *     2405:8a00:a:2::c5
  5     *        *      311 ms  2405:8a00::16
  6     *        *       69 ms  2405:8a00:a:10::2
  7    13 ms    10 ms    10 ms  2001:4860:1:1:0:269d:0:2
  8     *      525 ms   477 ms  2001:4860:0:11de::1
  9   162 ms     6 ms   322 ms  2001:4860:0:1::5e45
 10  1781 ms   480 ms  1441 ms  del12s10-in-x04.1e100.net [2404:6800:4002:82e::2004]

Trace complete.
```

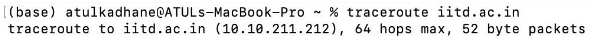
Figure 4: Trace to www.google.com(IPv6)

2. In **MAC, maximum hops are 64 by default whereas in Windows it's 30** by default. This difference is due to the underlying implementation and default behavior of the operating systems.



```
C:\Users\Dell>tracert www.iitd.ac.in

Tracing route to www.iitd.ac.in [10.10.211.212]
over a maximum of 30 hops:
```

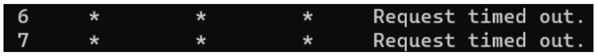
Figure 5: Default hops in Windows



```
(base) atulkadhane@ATULs-MacBook-Pro ~ % traceroute iitd.ac.in
traceroute to iitd.ac.in (10.10.211.212), 64 hops max, 52 byte packets
```

Figure 6: Default hops in MAC

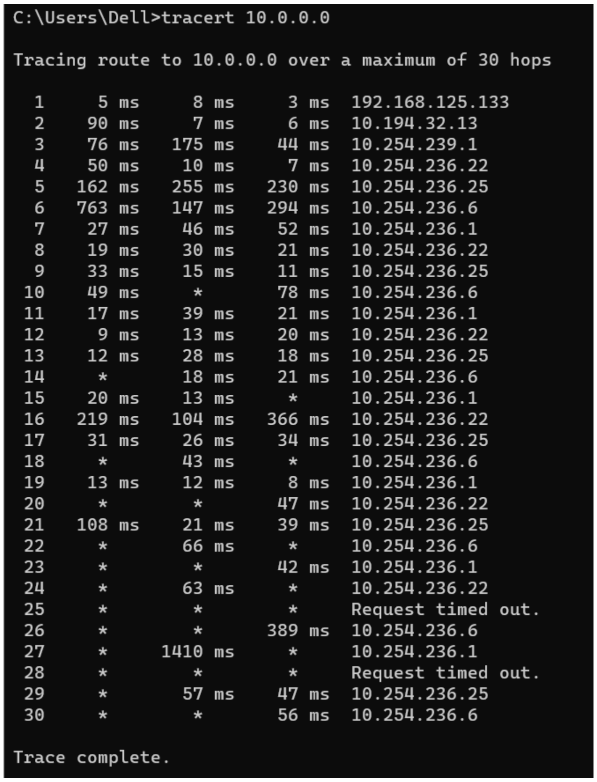
3. We saw **missing routers along the path** that do not seem to reply to the traceroute. These routers are configured to deprioritise or automatically reject ICMP (Internet control message protocol) packets. If we see too many such rows in output, it signifies that network is congested.



```
6     *        *        *     Request timed out.
7     *        *        *     Request timed out.
```

Figure 7: Missing routers

4. **Private IP addresses not traceable**. Upon using traceroute on private IP, often found the command stuck in loops.

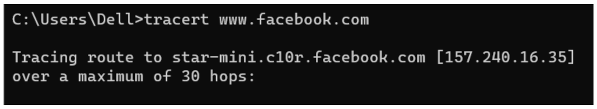


```
C:\Users\Dell>tracert 10.0.0.0

Tracing route to 10.0.0.0 over a maximum of 30 hops

  1     5 ms     8 ms     3 ms  192.168.125.133
  2    90 ms     7 ms     6 ms  10.194.32.13
  3    76 ms   175 ms    44 ms  10.254.239.1
  4    50 ms    10 ms     7 ms  10.254.236.22
  5   162 ms   255 ms   230 ms  10.254.236.25
  6   763 ms   147 ms   294 ms  10.254.236.6
  7    27 ms    46 ms    52 ms  10.254.236.1
  8    19 ms    30 ms    21 ms  10.254.236.22
  9    33 ms    15 ms    11 ms  10.254.236.25
 10    49 ms     *       78 ms  10.254.236.6
 11    17 ms    39 ms    21 ms  10.254.236.1
 12     9 ms    13 ms    20 ms  10.254.236.22
 13    12 ms    28 ms    18 ms  10.254.236.25
 14     *       18 ms    21 ms  10.254.236.6
 15    20 ms    13 ms     *     10.254.236.1
 16   219 ms   104 ms   366 ms  10.254.236.22
 17    31 ms    26 ms    34 ms  10.254.236.25
 18     *       43 ms     *     10.254.236.6
 19    13 ms    12 ms     8 ms  10.254.236.1
 20     *        *       47 ms  10.254.236.22
 21   108 ms    21 ms    39 ms  10.254.236.25
 22     *       66 ms     *     10.254.236.6
 23     *        *       42 ms  10.254.236.1
 24     *       63 ms     *     10.254.236.22
 25     *        *        *     Request timed out.
 26     *        *      389 ms  10.254.236.6
 27     *     1410 ms     *     10.254.236.1
 28     *        *        *     Request timed out.
 29     *       57 ms    47 ms  10.254.236.25
 30     *        *       56 ms  10.254.236.6

Trace complete.
```

Figure 8: Traceroute stuck in loops

5. **Change in IP address of cellular network used**. This happens due to DHCP(Dynamic Host Configuration Protocol). Whereas IP addresses of popular websites and services, including Google, often remain relatively stable over time. This is done so that they are easily accessible.

6. **Reverse DNS lookup giving modified hostnames**. This is due to distributed nature of host's network. Some servers use load balancing and CDNs to optimize performance and distribute traffic.

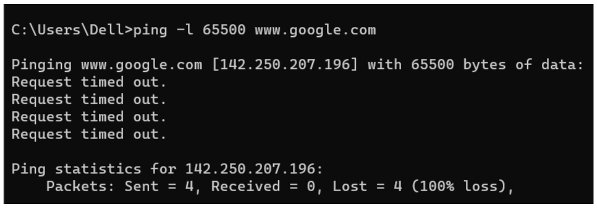


```
C:\Users\Dell>tracert www.facebook.com

Tracing route to star-mini.c10r.facebook.com [157.240.16.35]
over a maximum of 30 hops:
```

Figure 9: Reverse DNS lookup of facebook

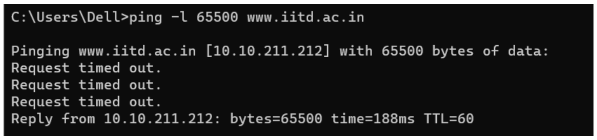
c Maximum size of ping packets

Upon using *ping* command, discovered that the maximum packet size that can be sent is **65500 bytes (On windows)**. Sending a packet of this size often results in "request timed out" almost everytime. Factors like network traffic and route can affect ping output.



```
C:\Users\Dell>ping -l 65500 www.google.com

Pinging www.google.com [142.250.207.196] with 65500 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 142.250.207.196:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

Figure 10: No ping successful



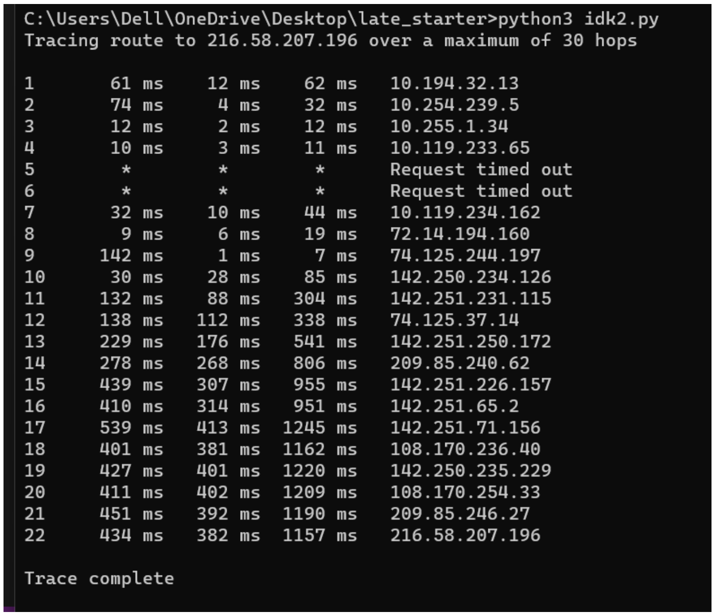
```
C:\Users\Dell>ping -l 65500 www.iitd.ac.in

Pinging www.iitd.ac.in [10.10.211.212] with 65500 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Reply from 10.10.211.212: bytes=65500 time=188ms TTL=60
```

Figure 11: 1 out of 4 pings successful

2 Replicating traceroute functionality using ping

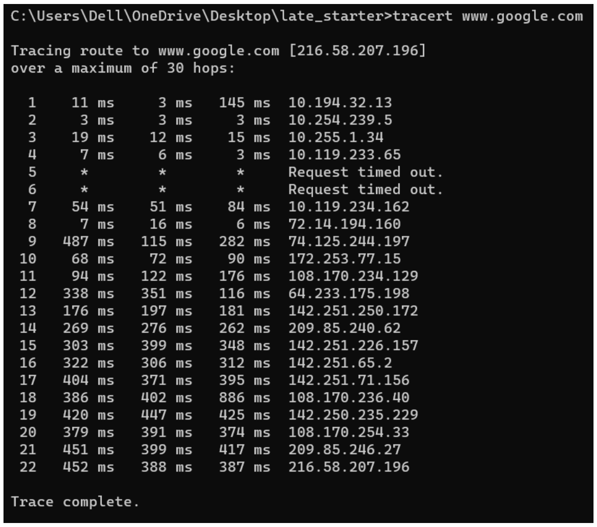
Since the code was made on Windows system, utilized nping to replicate tracert. The snapshots comparing in-built tracert and replicated tracert are shown.(Trace done to www.google.com)

On running in-built tracroute on some ip addresses it leads to failure (eg. 10.119.233.65), although such IPs can be seen if they are in path for some other IP. For such cases, the simulation outputs that the IP can't be traced.



```
C:\Users\Dell\OneDrive\Desktop\late_starter>python3 idk2.py
Tracing route to 216.58.207.196 over a maximum of 30 hops

1       61 ms    12 ms    62 ms   10.194.32.13
2       74 ms     4 ms    32 ms   10.254.239.5
3       12 ms     2 ms    12 ms   10.255.1.34
4       10 ms     3 ms    11 ms   10.119.233.65
5        *        *        *      Request timed out
6        *        *        *      Request timed out
7       32 ms    10 ms    44 ms   10.119.234.162
8        9 ms     6 ms    19 ms   72.14.194.160
9      142 ms     1 ms     7 ms   74.125.244.197
10      30 ms    28 ms    85 ms   142.250.234.126
11     132 ms    88 ms   304 ms   142.251.231.115
12     138 ms   112 ms   338 ms   74.125.37.14
13     229 ms   176 ms   541 ms   142.251.250.172
14     278 ms   268 ms   806 ms   209.85.240.62
15     439 ms   307 ms   955 ms   142.251.226.157
16     410 ms   314 ms   951 ms   142.251.65.2
17     539 ms   413 ms  1245 ms   142.251.71.156
18     401 ms   381 ms  1162 ms   108.170.236.40
19     427 ms   401 ms  1220 ms   142.250.235.229
20     411 ms   402 ms  1209 ms   108.170.254.33
21     451 ms   392 ms  1190 ms   209.85.246.27
22     434 ms   382 ms  1157 ms   216.58.207.196

Trace complete
```

Figure 12: Code o/p replicating traceroute functionality



```
C:\Users\Dell\OneDrive\Desktop\late_starter>tracert www.google.com

Tracing route to www.google.com [216.58.207.196]
over a maximum of 30 hops:

  1    11 ms     3 ms   145 ms  10.194.32.13
  2     3 ms     3 ms     3 ms  10.254.239.5
  3    19 ms    12 ms    15 ms  10.255.1.34
  4     7 ms     6 ms     3 ms  10.119.233.65
  5     *        *        *     Request timed out.
  6     *        *        *     Request timed out.
  7    54 ms    51 ms    84 ms  10.119.234.162
  8     7 ms    16 ms     6 ms  72.14.194.160
  9   487 ms   115 ms   282 ms  74.125.244.197
 10    68 ms    72 ms    90 ms  172.253.77.15
 11    94 ms   122 ms   176 ms  108.170.234.129
 12   338 ms   351 ms   116 ms  64.233.175.198
 13   176 ms   197 ms   181 ms  142.251.250.172
 14   269 ms   276 ms   262 ms  209.85.240.62
 15   303 ms   399 ms   348 ms  142.251.226.157
 16   322 ms   306 ms   312 ms  142.251.65.2
 17   404 ms   371 ms   395 ms  142.251.71.156
 18   386 ms   402 ms   886 ms  108.170.236.40
 19   420 ms   447 ms   425 ms  142.250.235.229
 20   379 ms   391 ms   374 ms  108.170.254.33
 21   451 ms   399 ms   417 ms  209.85.246.27
 22   452 ms   388 ms   387 ms  216.58.207.196

Trace complete.
```

Figure 13: Traceroute output

Note: Python script submitted separately.

3 Internet Architecture

a Commands performed

We used the "Traceroute" command on mac or the "Tracert" command on windows to monitor the path traversed to reach the 5 DNS servers mentioned in the assignment pdf from three different locations

b Observations

A) Consult an AS-IP lookup service to figure out when traffic gets into the local ISP, transits to other intermediate ISPs, and finally into the destination domains.

Answer - AS-IP lookup service refers to a service that allows you to find the **Autonomous System (AS) associated with a specific IP address**. Autonomous Systems are essentially groups of IP networks and routers under the control of a single organization, often an Internet Service Provider (ISP) or a large company. If we closely examine the RTT time from a certain intermediate router to our own device has exceeded like 400ms, we can say that this router is handling heavy traffic and there are chances of packet drops and queing delay. We can use its ip address to find the local area of this router via the AS-IP lookup service.

B) In a neat tabular format, report the number of hops from the (3) traceroute sources to the above (5) destinations. If the pair of (traceroute source, destination) are geographically close to each other, does it roughly translate into fewer hops? Do Google and Facebook differ from the others in the number of hops required to reach them, irrespective of which traceroute source is used? Why would this be so?

Answer - Yes. If the traceroute source is closer to the destination, it results in fewer hops as observed in performing traceroutes to iitd.ac.in via Canada(more than 30) and via my own device in Delhi(4). Traceroutes to Google or Facebook are fewer in number of hops since these are accessed globally and hence it is necessary to reduce the latency for these websites.

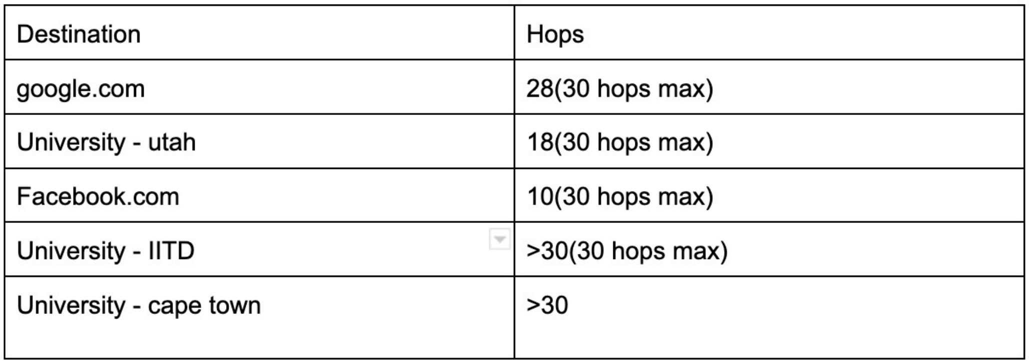
Traceroute 1 - Own device

Destination	Hops
google.com	10(64 hops max)
University - utah	34(64 hops max)
Facebook.com	13(64 hops max)
University - IITD	4(64 hops max)
University - cape town	>64

Traceroute 2 - Canada

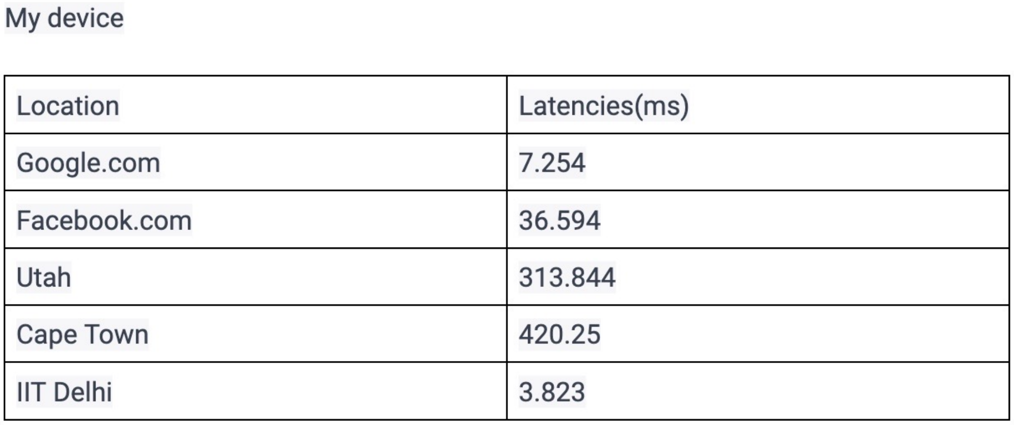
Destination	Hops
google.com	27(30 hops max)
University - utah	21(30 hops max)
Facebook.com	10(30 hops max)
University - IITD	>30(30 hops max)
University - cape town	>30

Traceroute 3 - USA



Destination	Hops
google.com	28(30 hops max)
University - utah	18(30 hops max)
Facebook.com	10(30 hops max)
University - IITD	>30(30 hops max)
University - cape town	>30

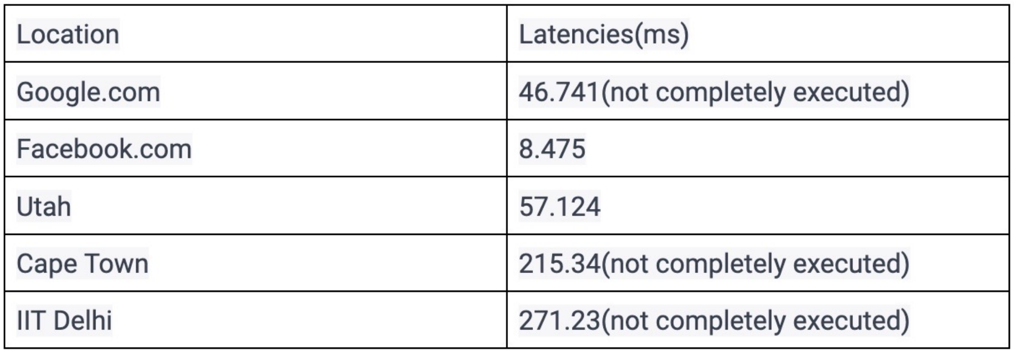
C) Report the latencies between the traceroute sources and the web-servers. Does the latency seem to be related to the number of hops, being higher when there are more hops? Why is this the case?



My device

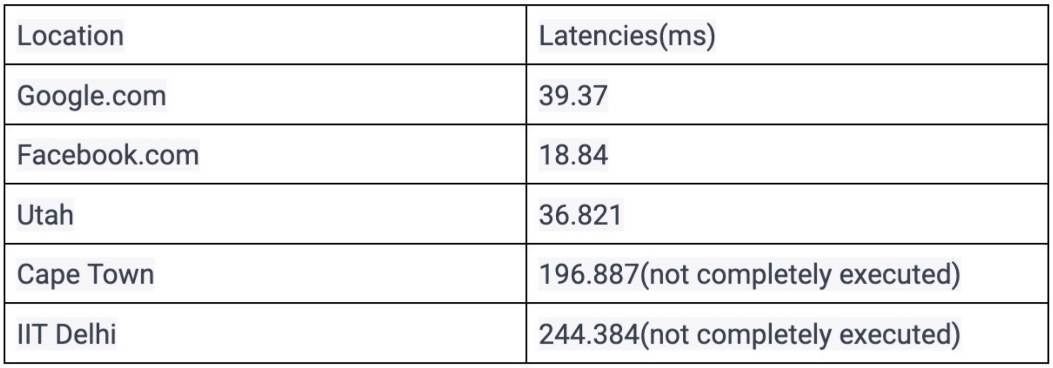
Location	Latencies(ms)
Google.com	7.254
Facebook.com	36.594
Utah	313.844
Cape Town	420.25
IIT Delhi	3.823

Canada - Montreal



Location	Latencies(ms)
Google.com	46.741(not completely executed)
Facebook.com	8.475
Utah	57.124
Cape Town	215.34(not completely executed)
IIT Delhi	271.23(not completely executed)

USA - Chicago



Location	Latencies(ms)
Google.com	39.37
Facebook.com	18.84
Utah	36.821
Cape Town	196.887(not completely executed)
IIT Delhi	244.384(not completely executed)

Answer - The latencies are summarised in the table and these values are approximate values. It is possible that it may increase or decrease if the traceroute is performed again. A "hop" refers to a single point-to-point communication between two devices, often passing through intermediate routers or switches. Latency refers to the time it takes for data to travel from the source to the destination. Each hop introduces some delay due to the processing time of routers, switches, and other network equipment. More hops suggest that there is not a faster route to that server since such routers weren't placed to build a faster connection. This introduces **propagation delay, queuing delay, processing delay and routing delay**.

D) Which of the destination web-servers are resolved to the same IP address irrespective of from where you do a traceroute to them? Why do you think some web-servers are resolved to different IP addresses when queried from different parts of the world?

Answer - Web servers like that of Universities which are regional and not that heavily accessed on a global level have the same IP addresses for a traceroute from any part of the world. Web servers that are resolved to the same IP address regardless of the traceroute source are often the result of using **content delivery networks (CDNs)**. CDNs are distributed networks of servers that work together to deliver web content to users from the server closest to them geographically. This enhances performance and reduces latency by minimizing the distance data needs to travel.

E) If you do traceroutes from the same starting point to different IP addresses you found for the same web-server, do the paths appear different? Which ones are longer?

Answer - Web servers like that of Google, Facebook, Netflix have different IP addresses from the same starting point since these are heavily accessed and it is necessary to create a shorter route to reach them. That depends on the router closet to them leading it towards the destination router. The paths may or may not differ depending on the route chosen by the packet from the lookup table. Accordingly, the latencies will also change.

F) Try tracerouting to Google and Facebook from different countries of traceroute servers around the world. Are you able to find any countries that do not seem to have their local ISP's directly peered with Google and Facebook?

Answer - Yes, as visible from the above table, Canada seems to have a greater number of hops to Google and Facebook which might suggest that their local ISP's are not directly peered at. Although this may not be true since latency can also be the result from network congestion and other factors. Some servers which are directly peered with Google and Facebook show a direct jump from local ISP to routers in these website's network after some hops.

4 Packet Analysis

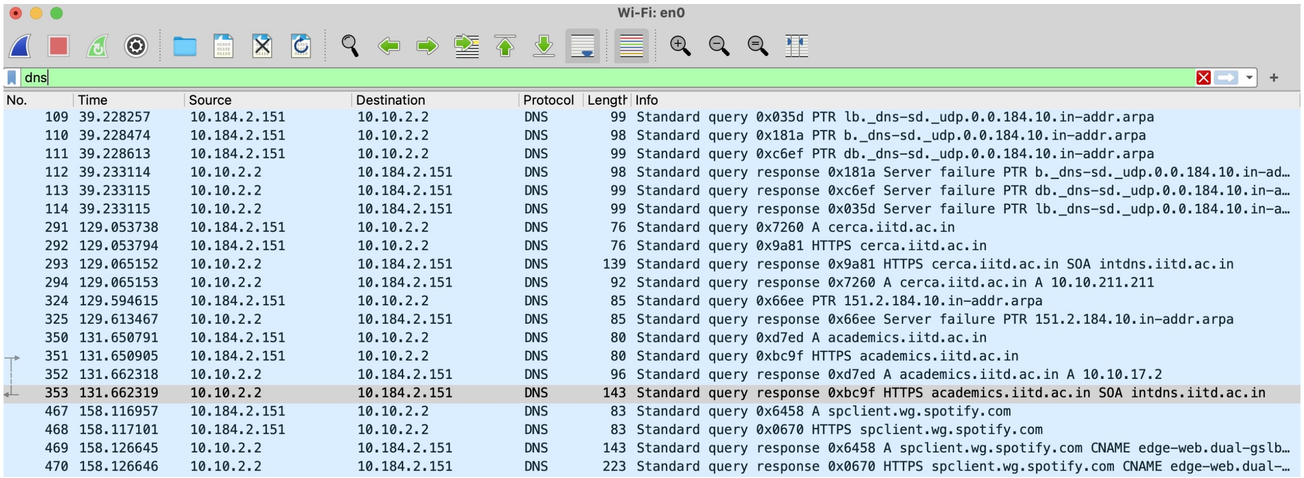
a Commands performed

Used wireshark to grab all packets while visiting the HTTP website such as http://act4d.iitd.ac.in from the browser. Performed a flushdns to clear the local DNS cache.

b Observations

A) Apply a "dns" filter on the packet trace, and see if you can find DNS queries and responses for www.iitd.ac.in. How long did it take for the DNS request-response to complete?

Answer - We received 4 dns query - response pairs for iitd.ac.in. Upon observing, we found that the average time between query and response time is around 10 ms.

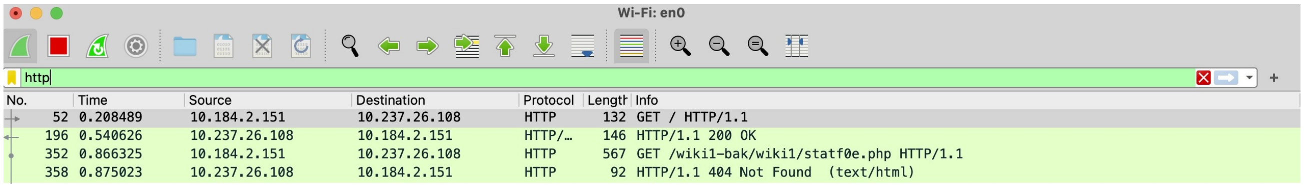


```
No.  Time        Source        Destination   Protocol Length Info
 109 39.228257   10.184.2.151  10.10.2.2     DNS       99 Standard query 0x035d PTR lb._dns-sd._udp.0.0.184.10.in-addr.arpa
 110 39.228474   10.184.2.151  10.10.2.2     DNS       98 Standard query 0x181a PTR b._dns-sd._udp.0.0.184.10.in-addr.arpa
 111 39.228613   10.184.2.151  10.10.2.2     DNS       99 Standard query 0xc6ef PTR db._dns-sd._udp.0.0.184.10.in-addr.arpa
 112 39.233114   10.10.2.2     10.184.2.151  DNS       98 Standard query response 0x181a Server failure PTR b._dns-sd._udp.0.0.184.10.in-ad…
 113 39.233115   10.10.2.2     10.184.2.151  DNS       99 Standard query response 0xc6ef Server failure PTR db._dns-sd._udp.0.0.184.10.in-a…
 114 39.233115   10.10.2.2     10.184.2.151  DNS       99 Standard query response 0x035d Server failure PTR lb._dns-sd._udp.0.0.184.10.in-a…
 291 129.053738  10.184.2.151  10.10.2.2     DNS       76 Standard query 0x7260 A cerca.iitd.ac.in
 292 129.053794  10.184.2.151  10.10.2.2     DNS       76 Standard query 0x9a81 HTTPS cerca.iitd.ac.in
 293 129.065152  10.10.2.2     10.184.2.151  DNS      139 Standard query response 0x9a81 HTTPS cerca.iitd.ac.in SOA intdns.iitd.ac.in
 294 129.065153  10.10.2.2     10.184.2.151  DNS       92 Standard query response 0x7260 A cerca.iitd.ac.in A 10.10.211.211
 324 129.594615  10.184.2.151  10.10.2.2     DNS       85 Standard query 0x66ee PTR 151.2.184.10.in-addr.arpa
 325 129.613467  10.10.2.2     10.184.2.151  DNS       85 Standard query response 0x66ee Server failure PTR 151.2.184.10.in-addr.arpa
 350 131.650791  10.184.2.151  10.10.2.2     DNS       80 Standard query 0xd7ed A academics.iitd.ac.in
 351 131.650905  10.184.2.151  10.10.2.2     DNS       80 Standard query 0xbc9f HTTPS academics.iitd.ac.in
 352 131.662318  10.10.2.2     10.184.2.151  DNS       96 Standard query response 0xd7ed A academics.iitd.ac.in A 10.10.17.2
 353 131.662319  10.10.2.2     10.184.2.151  DNS      143 Standard query response 0xbc9f HTTPS academics.iitd.ac.in SOA intdns.iitd.ac.in
 467 158.116957  10.184.2.151  10.10.2.2     DNS       83 Standard query 0x6458 A spclient.wg.spotify.com
 468 158.117101  10.184.2.151  10.10.2.2     DNS       83 Standard query 0x0670 HTTPS spclient.wg.spotify.com
 469 158.126645  10.10.2.2     10.184.2.151  DNS      143 Standard query response 0x6458 A spclient.wg.spotify.com CNAME edge-web.dual-gslb…
 470 158.126646  10.10.2.2     10.184.2.151  DNS      223 Standard query response 0x0670 HTTPS spclient.wg.spotify.com CNAME edge-web.dual-…
```

Figure 14: DNS Query

B) Apply an "http" filter on the packet trace and report the approximate number of HTTP requests that were generated. What can you tell from this observation about how web- pages are structured, and how browsers render complex pages with multiple images and files?

Answer - We are capturing packets for the site act4d.iitd.ac.in. As this is not a secure connection, we are able to see http request via which data is getting sent to the receiver. There are a minimum of 10 HTTP requests on wireshark. The site has complex pages with multiple images and javascripts which requires us to send it through different HTTP requests. As the browser parses the HTML, it encounters references to **external resources like images, CSS files, and JavaScript files**. It sends separate requests to the server for each of these resources. These resources are fetched in parallel, which helps improve the loading speed of the page.

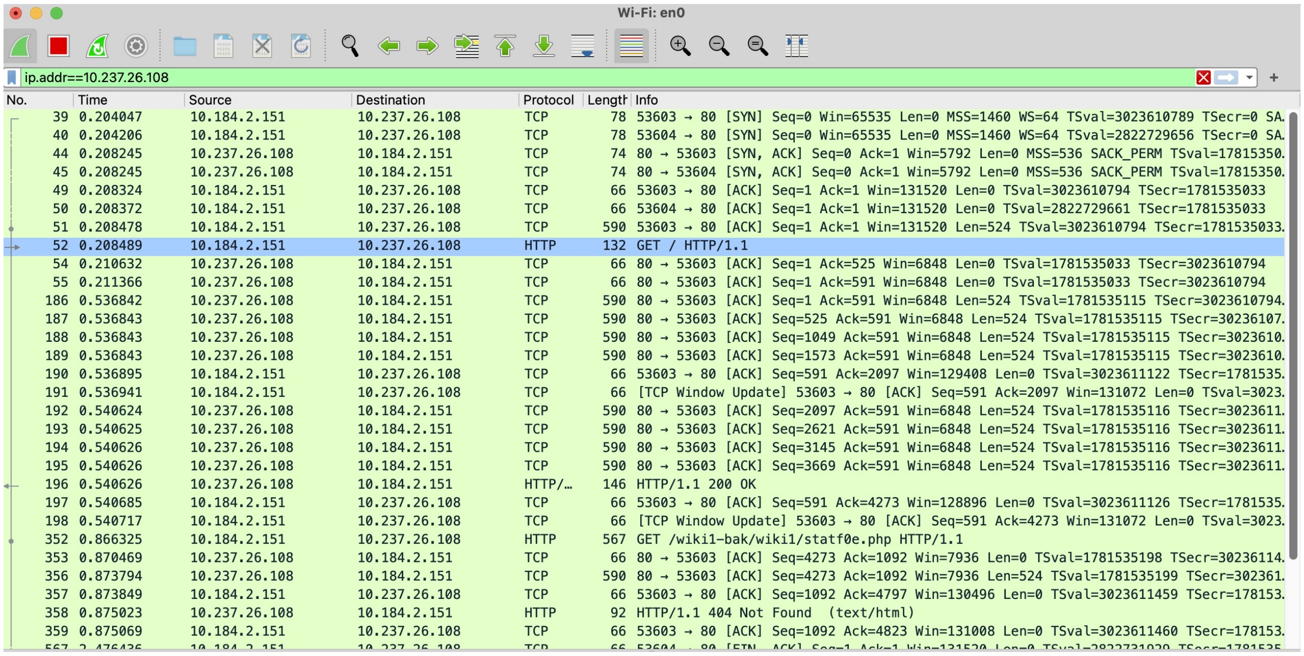


```
No.  Time      Source         Destination    Protocol Length Info
  52 0.208489  10.184.2.151   10.237.26.108  HTTP      132 GET / HTTP/1.1
 196 0.540626  10.237.26.108  10.184.2.151   HTTP/…    146 HTTP/1.1 200 OK
 352 0.866325  10.184.2.151   10.237.26.108  HTTP      567 GET /wiki1-bak/wiki1/statf0e.php HTTP/1.1
 358 0.875023  10.237.26.108  10.184.2.151   HTTP       92 HTTP/1.1 404 Not Found  (text/html)
```

Figure 15: HTTP Request

C) Find the number of TCP connections that were opened between your browser and the web-server. Is this the same as the number of HTTP requests for content objects that you found in the previous part? Do you find that some content objects are fetched over the same TCP connection? Note that TCP connections are distinguished from one another based on the source port and destination port.

Answer - There are a 30 of TCP connection opened between my browser and the web server which is greater than the HTTP requests. Each packet being received by the receiver needs to send an acknowledgemnt message which creates a TCP Connection. The message is chopped at the transport layer and until all the packets are not received for a particular message, the message is not sent out. All the chopped packets for a particular message are assigned the same destination port number and as the packets are received , the transport layer of the receiver checks the sequence numbers along with the port numbers and accordingly waits until it receives that particular packet. The sender is alerted by the receiver through the [ACK] signals for each packet.



Wi-Fi: en0

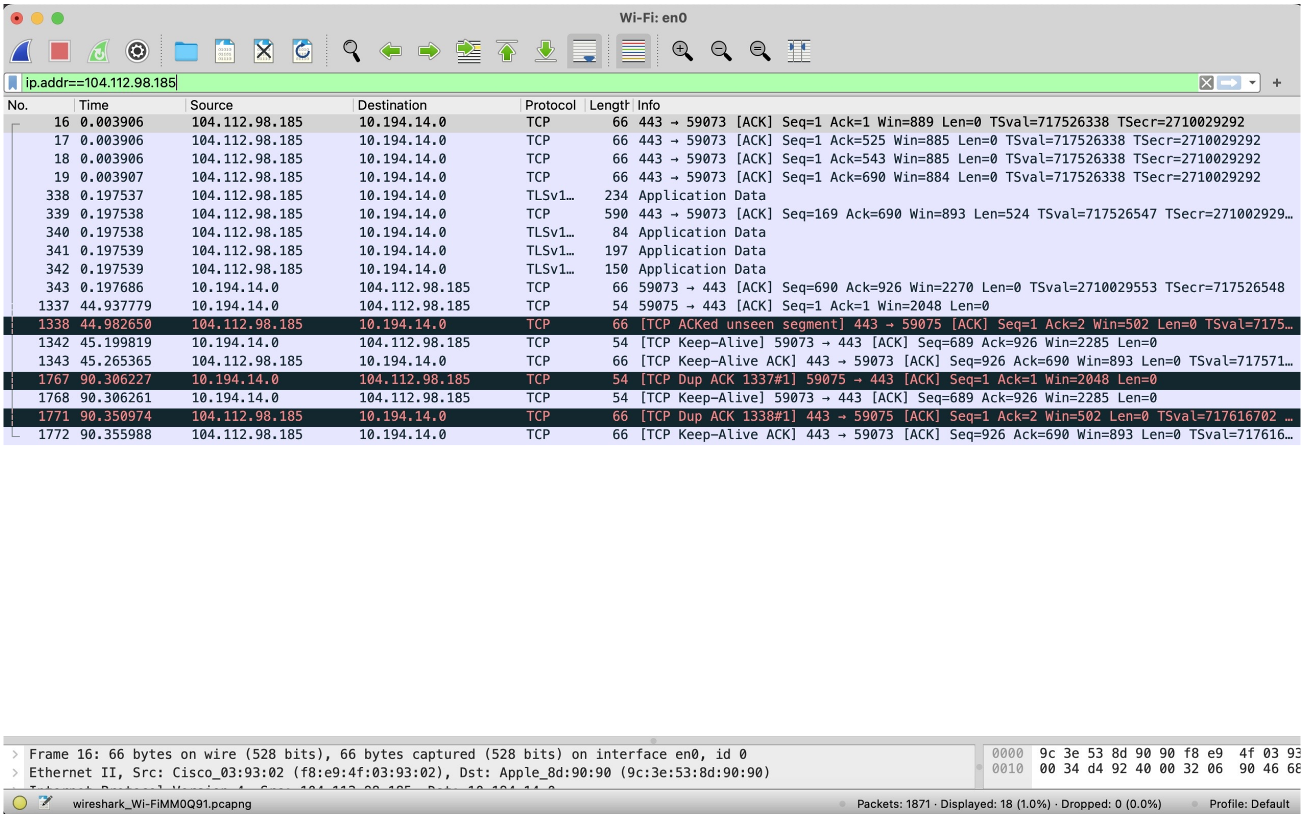
ip.addr==10.237.26.108

No.	Time	Source	Destination	Protocol	Length	Info
39	0.204047	10.184.2.151	10.237.26.108	TCP	78	53603 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TSval=3023610789 TSecr=0 SA.
40	0.204206	10.184.2.151	10.237.26.108	TCP	78	53604 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TSval=2822729656 TSecr=0 SA.
44	0.208245	10.237.26.108	10.184.2.151	TCP	74	80 → 53603 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=536 SACK_PERM TSval=17815350.
45	0.208245	10.237.26.108	10.184.2.151	TCP	74	80 → 53604 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=536 SACK_PERM TSval=17815350.
49	0.208324	10.184.2.151	10.237.26.108	TCP	66	53603 → 80 [ACK] Seq=1 Ack=1 Win=131520 Len=0 TSval=3023610794 TSecr=1781535033
50	0.208372	10.184.2.151	10.237.26.108	TCP	66	53604 → 80 [ACK] Seq=1 Ack=1 Win=131520 Len=0 TSval=2822729661 TSecr=1781535033
51	0.208478	10.184.2.151	10.237.26.108	TCP	590	53603 → 80 [ACK] Seq=1 Ack=1 Win=131520 Len=524 TSval=3023610794 TSecr=1781535033.
52	0.208489	10.184.2.151	10.237.26.108	HTTP	132	GET / HTTP/1.1
54	0.210632	10.237.26.108	10.184.2.151	TCP	66	80 → 53603 [ACK] Seq=1 Ack=525 Win=6848 Len=0 TSval=1781535033 TSecr=3023610794
55	0.211366	10.237.26.108	10.184.2.151	TCP	66	80 → 53603 [ACK] Seq=1 Ack=591 Win=6848 Len=0 TSval=1781535033 TSecr=3023610794
186	0.536842	10.237.26.108	10.184.2.151	TCP	590	80 → 53603 [ACK] Seq=1 Ack=591 Win=6848 Len=524 TSval=1781535115 TSecr=3023610794.
187	0.536843	10.237.26.108	10.184.2.151	TCP	590	80 → 53603 [ACK] Seq=525 Ack=591 Win=6848 Len=524 TSval=1781535115 TSecr=30236107.
188	0.536843	10.237.26.108	10.184.2.151	TCP	590	80 → 53603 [ACK] Seq=1049 Ack=591 Win=6848 Len=524 TSval=1781535115 TSecr=3023610.
189	0.536843	10.237.26.108	10.184.2.151	TCP	590	80 → 53603 [ACK] Seq=1573 Ack=591 Win=6848 Len=524 TSval=1781535115 TSecr=3023610.
190	0.536895	10.184.2.151	10.237.26.108	TCP	66	53603 → 80 [ACK] Seq=591 Ack=2097 Win=129408 Len=0 TSval=3023611122 TSecr=1781535.
191	0.536941	10.184.2.151	10.237.26.108	TCP	66	[TCP Window Update] 53603 → 80 [ACK] Seq=591 Ack=2097 Win=131072 Len=0 TSval=3023.
192	0.540624	10.237.26.108	10.184.2.151	TCP	590	80 → 53603 [ACK] Seq=2097 Ack=591 Win=6848 Len=524 TSval=1781535116 TSecr=3023611.
193	0.540625	10.237.26.108	10.184.2.151	TCP	590	80 → 53603 [ACK] Seq=2621 Ack=591 Win=6848 Len=524 TSval=1781535116 TSecr=3023611.
194	0.540626	10.237.26.108	10.184.2.151	TCP	590	80 → 53603 [ACK] Seq=3145 Ack=591 Win=6848 Len=524 TSval=1781535116 TSecr=3023611.
195	0.540626	10.237.26.108	10.184.2.151	TCP	590	80 → 53603 [ACK] Seq=3669 Ack=591 Win=6848 Len=524 TSval=1781535116 TSecr=3023611.
196	0.540626	10.237.26.108	10.184.2.151	HTTP/…	146	HTTP/1.1 200 OK
197	0.540685	10.184.2.151	10.237.26.108	TCP	66	53603 → 80 [ACK] Seq=591 Ack=4273 Win=128896 Len=0 TSval=3023611126 TSecr=1781535.
198	0.540717	10.184.2.151	10.237.26.108	TCP	66	[TCP Window Update] 53603 → 80 [ACK] Seq=591 Ack=4273 Win=131072 Len=0 TSval=3023.
352	0.866325	10.184.2.151	10.237.26.108	HTTP	567	GET /wiki1-bak/wiki1/statf0e.php HTTP/1.1
353	0.870469	10.237.26.108	10.184.2.151	TCP	66	80 → 53603 [ACK] Seq=4273 Ack=1092 Win=7936 Len=0 TSval=1781535198 TSecr=30236114.
356	0.873794	10.237.26.108	10.184.2.151	TCP	590	80 → 53603 [ACK] Seq=4273 Ack=1092 Win=7936 Len=524 TSval=1781535199 TSecr=302361.
357	0.873849	10.184.2.151	10.237.26.108	TCP	66	53603 → 80 [ACK] Seq=1092 Ack=4797 Win=130496 Len=0 TSval=3023611459 TSecr=178153.
358	0.875023	10.237.26.108	10.184.2.151	HTTP	92	HTTP/1.1 404 Not Found (text/html)
359	0.875069	10.184.2.151	10.237.26.108	TCP	66	53603 → 80 [ACK] Seq=1092 Ack=4823 Win=131008 Len=0 TSval=3023611460 TSecr=178153.
567	2.476436	10.184.2.151	10.237.26.108	TCP	66	53604 → 80 [FIN, ACK] Seq=1 Ack=1 Win=131520 Len=0 TSval=2822731929 TSecr=1781535

Figure 16: TCP Request

D) Now try doing a trace for http://www.indianexpress.com and filter for "http". What do you find, is there any HTTP traffic? Browse through the entire trace without any filters, are you able to see the contents of any HTML and Javascript files being transferred? What just happened?

Answer - Doing a trace for Indian Express website, we see **no HTTP requests** on wireshark. Indian Express has a secure connection and hence all the data being transferred is encrypted. It uses the TLSv1(Transport Layer Security version 1) protocol. TLSv1 is a **cryptographic protocol that provides secure communication over a network. It ensures data confidentiality, integrity, and authenticity between two parties (client and server) by encrypting the data being transmitted**. It uses encryption algorithms and cryptographic techniques to protect the data from eavesdropping and tampering. Hence we are not able to see the contents of the HTML and Javascript files being transferred.



```
ip.addr==104.112.98.185
```

```
No.    Time       Source          Destination     Protocol  Length Info
  16 0.003906   104.112.98.185  10.194.14.0     TCP        66 443 → 59073 [ACK] Seq=1 Ack=1 Win=889 Len=0 TSval=717526338 TSecr=2710029292
  17 0.003906   104.112.98.185  10.194.14.0     TCP        66 443 → 59073 [ACK] Seq=1 Ack=525 Win=885 Len=0 TSval=717526338 TSecr=2710029292
  18 0.003906   104.112.98.185  10.194.14.0     TCP        66 443 → 59073 [ACK] Seq=1 Ack=543 Win=885 Len=0 TSval=717526338 TSecr=2710029292
  19 0.003907   104.112.98.185  10.194.14.0     TCP        66 443 → 59073 [ACK] Seq=1 Ack=690 Win=884 Len=0 TSval=717526338 TSecr=2710029292
 338 0.197537   104.112.98.185  10.194.14.0     TLSv1…    234 Application Data
 339 0.197538   104.112.98.185  10.194.14.0     TCP       590 443 → 59073 [ACK] Seq=169 Ack=690 Win=893 Len=524 TSval=717526547 TSecr=271002929…
 340 0.197538   104.112.98.185  10.194.14.0     TLSv1…     84 Application Data
 341 0.197539   104.112.98.185  10.194.14.0     TLSv1…    197 Application Data
 342 0.197539   104.112.98.185  10.194.14.0     TLSv1…    150 Application Data
 343 0.197686   10.194.14.0     104.112.98.185  TCP        66 59073 → 443 [ACK] Seq=690 Ack=926 Win=2270 Len=0 TSval=2710029553 TSecr=717526548
1337 44.937779  10.194.14.0     104.112.98.185  TCP        54 59075 → 443 [ACK] Seq=1 Ack=1 Win=2048 Len=0
1338 44.982650  104.112.98.185  10.194.14.0     TCP        66 [TCP ACKed unseen segment] 443 → 59075 [ACK] Seq=1 Ack=2 Win=502 Len=0 TSval=7175…
1342 45.199819  10.194.14.0     104.112.98.185  TCP        54 [TCP Keep-Alive] 59073 → 443 [ACK] Seq=689 Ack=926 Win=2285 Len=0
1343 45.265365  104.112.98.185  10.194.14.0     TCP        66 [TCP Keep-Alive ACK] 443 → 59073 [ACK] Seq=926 Ack=690 Win=893 Len=0 TSval=717571…
1767 90.306227  10.194.14.0     104.112.98.185  TCP        54 [TCP Dup ACK 1337#1] 59075 → 443 [ACK] Seq=1 Ack=1 Win=2048 Len=0
1768 90.306261  10.194.14.0     104.112.98.185  TCP        54 [TCP Keep-Alive] 59073 → 443 [ACK] Seq=689 Ack=926 Win=2285 Len=0
1771 90.350974  104.112.98.185  10.194.14.0     TCP        66 [TCP Dup ACK 1338#1] 443 → 59075 [ACK] Seq=1 Ack=2 Win=502 Len=0 TSval=717616702 …
1772 90.355988  104.112.98.185  10.194.14.0     TCP        66 [TCP Keep-Alive ACK] 443 → 59073 [ACK] Seq=926 Ack=690 Win=893 Len=0 TSval=717616…
```

```
> Frame 16: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface en0, id 0
> Ethernet II, Src: Cisco_03:93:02 (f8:e9:4f:03:93:02), Dst: Apple_8d:90:90 (9c:3e:53:8d:90:90)
```

```
0000  9c 3e 53 8d 90 90 f8 e9  4f 03 93
0010  00 34 d4 92 40 00 32 06  90 46 68
```

wireshark_Wi-FiMM0Q91.pcapng
Packets: 1871 · Displayed: 18 (1.0%) · Dropped: 0 (0.0%)
Profile: Default

Figure 17: Analysing trace to Indian Express Website