COL334 ASSIGNMENT 1

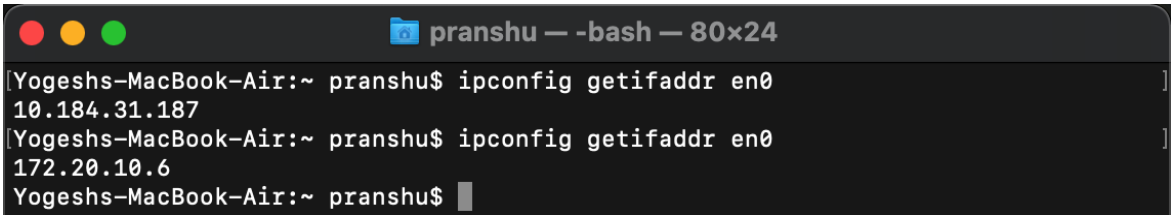
ARPIT CHAUHAN - 2019CS10332

August 22, 2021

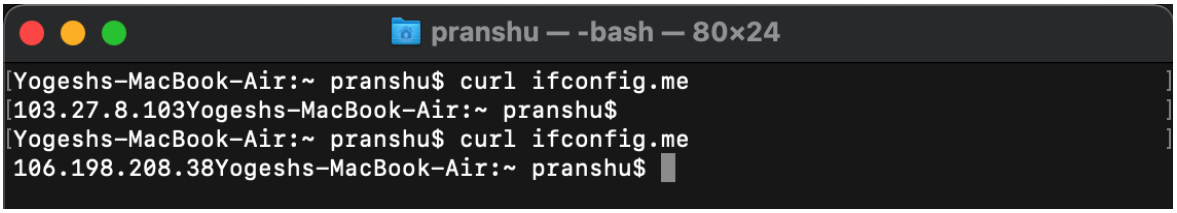
1. Networking Tools

A.

- Local IP address on IITD Wi-Fi : 10.184.31.187
- Local IP address on AIRTEL 4G : 172.20.10.6



- The public IP address on IITD Wi-Fi : 103.27.8.103
- The public IP address on AIRTEL 4G : 106.198.208.38



We observe that the IP address changes on changing the ISP because of a change in the network infrastructure. Here we have used 'ipconfig' which is a utility that communicates with the IPConfiguration agent to retrieve and set IP configuration parameters.

B.

- Here we consider the default DNS server and 3 open DNS servers, 1.1.1.1 (Cloudflare), 8.8.8.8 (Google Public DNS), and 9.9.9.9 (Quad9) .
- Results obtained for default DNS server -

```
Yogeshs-MacBook-Air:~ pranshu$ nslookup www.google.com
Server:         10.10.2.2
Address:        10.10.2.2#53

Non-authoritative answer:
Name:   www.google.com
Address: 142.250.194.196

Yogeshs-MacBook-Air:~ pranshu$ nslookup www.facebook.com
Server:         10.10.2.2
Address:        10.10.2.2#53

Non-authoritative answer:
www.facebook.com        canonical name = star-mini.c10r.facebook.com.
Name:   star-mini.c10r.facebook.com
Address: 157.240.16.35
```

Domain\DNS	10.10.2.2	1.1.1.1	8.8.8.8	9.9.9.9
www.google.com	142.250.194.196	142.250.195.36	142.250.195.4	142.250.204.100
www.facebook.com	157.240.16.35	157.240.16.35	157.240.239.35	157.240.13.35

- Results obtained for www.google.com on open DNS servers -

```
Yogeshs-MacBook-Air:~ pranshu$ nslookup www.google.com 1.1.1.1
Server:         1.1.1.1
Address:        1.1.1.1#53

Non-authoritative answer:
Name:   www.google.com
Address: 142.250.195.36

Yogeshs-MacBook-Air:~ pranshu$ nslookup www.google.com 8.8.8.8
Server:         8.8.8.8
Address:        8.8.8.8#53

Non-authoritative answer:
Name:   www.google.com
Address: 142.250.195.4
```

```
Yogeshs-MacBook-Air:~ pranshu$ nslookup www.google.com 9.9.9.9
Server:         9.9.9.9
Address:        9.9.9.9#53

Non-authoritative answer:
Name:   www.google.com
Address: 142.250.204.100
```

- .Results obtained for www.facebook.com on open DNS servers -

```
Yogeshs-MacBook-Air:~ pranshu$ nslookup www.facebook.com 1.1.1.1
Server:         1.1.1.1
Address:        1.1.1.1#53

Non-authoritative answer:
www.facebook.com        canonical name = star-mini.c10r.facebook.com.
Name:   star-mini.c10r.facebook.com
Address: 157.240.16.35

Yogeshs-MacBook-Air:~ pranshu$ nslookup www.facebook.com 8.8.8.8
Server:         8.8.8.8
Address:        8.8.8.8#53

Non-authoritative answer:
www.facebook.com        canonical name = star-mini.c10r.facebook.com.
Name:   star-mini.c10r.facebook.com
Address: 157.240.239.35

Yogeshs-MacBook-Air:~ pranshu$ nslookup www.facebook.com 9.9.9.9
Server:         9.9.9.9
Address:        9.9.9.9#53

Non-authoritative answer:
www.facebook.com        canonical name = star-mini.c10r.facebook.com.
Name:   star-mini.c10r.facebook.com
Address: 157.240.13.35
```

- Here we are using the **nslookup** command which is a program to query Internet domain name servers.
- We observe that on changing the DNS server, the IP addresses of the domains change but not by much. This change is due to the fact that these domains have multiple host servers which have similar but different addresses.

```
Yogeshs-MacBook-Air:~ pranshu$ host -t ns google.com
google.com name server ns1.google.com.
google.com name server ns3.google.com.
google.com name server ns2.google.com.
google.com name server ns4.google.com.
Yogeshs-MacBook-Air:~ pranshu$ nslookup www.google.com ns1.google.com
Server:         ns1.google.com
Address:        216.239.32.10#53

Name:   www.google.com
Address: 142.250.182.164
```

- Here, a Non-authoritative label means the result of the DNS query is coming indirectly from the authoritative DNS server. If we specify the exact DNS server for the domain, we don't get the Non-authoritative label

C.

- I used the **ping [-m ttl] [-s packetsize] [-c count] host** command for 3 different hosts using IITD Wi-fi and made the following observations.

HOST	Minimum TTL to reach Destination	Maximum Packet Size(data bytes)
www.iitd.ac.in	5	35512
www.google.com	12	1472
www.facebook.com	11	1472

```
[Yogeshs-MacBook-Air:~ pranshu$ sudo ping -s 35512 -c 1  www.iitd.ac.in
PING www.iitd.ac.in (103.27.9.24): 35512 data bytes
35520 bytes from 103.27.9.24: icmp_seq=0 ttl=60 time=11.142 ms

--- www.iitd.ac.in ping statistics ---
1 packets transmitted, 1 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 11.142/11.142/11.142/0.000 ms
[Yogeshs-MacBook-Air:~ pranshu$ sudo ping -s 35513 -c 1  www.iitd.ac.in
PING www.iitd.ac.in (103.27.9.24): 35513 data bytes

--- www.iitd.ac.in ping statistics ---
1 packets transmitted, 0 packets received, 100.0% packet loss
Yogeshs-MacBook-Air:~ pranshu$
```

```
[Yogeshs-MacBook-Air:~ pranshu$ sudo ping -s 1472 -c 1  www.google.com
PING www.google.com (142.250.192.4): 1472 data bytes
76 bytes from 142.250.192.4: icmp_seq=0 ttl=117 time=30.031 ms
wrong total length 96 instead of 1500

--- www.google.com ping statistics ---
1 packets transmitted, 1 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 30.031/30.031/30.031/0.000 ms
[Yogeshs-MacBook-Air:~ pranshu$ sudo ping -s 1473 -c 1  www.google.com
PING www.google.com (142.250.192.4): 1473 data bytes

--- www.google.com ping statistics ---
1 packets transmitted, 0 packets received, 100.0% packet loss
Yogeshs-MacBook-Air:~ pranshu$
```

```
[Yogeshs-MacBook-Air:~ pranshu$ sudo ping -s 1472 -c 1  www.facebook.com
PING star-mini.c10r.facebook.com (157.240.16.35): 1472 data bytes
1480 bytes from 157.240.16.35: icmp_seq=0 ttl=54 time=25.648 ms

--- star-mini.c10r.facebook.com ping statistics ---
1 packets transmitted, 1 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 25.648/25.648/25.648/0.000 ms
[Yogeshs-MacBook-Air:~ pranshu$ sudo ping -s 1473 -c 1  www.facebook.com
PING star-mini.c10r.facebook.com (157.240.16.35): 1473 data bytes

--- star-mini.c10r.facebook.com ping statistics ---
1 packets transmitted, 0 packets received, 100.0% packet loss
Yogeshs-MacBook-Air:~ pranshu$
```

- We observe that ping packets with a much larger packet size were successfully transmitted for www.iitd.ac.in than www.google.com and www.facebook.com. 8 additional bytes of ICMP header data are also sent along with the specified number of data bytes to be sent.
- Here we have used the **ping** utility that uses the ICMP protocol's mandatory ECHO_REQUEST datagram to elicit an ICMP ECHO_RESPONSE from a host or gateway. ECHO_REQUEST datagrams (``pings'') have an IP and ICMP header, followed by a ``struct timeval'' and then an arbitrary number of ``pad'' bytes used to fill out the packet.

D.

- Traceroute to www.google.com using AIRTEL 4G connected via Hotspot to my laptop.

```
Yogeshs-MacBook-Air:~ pranshu$ traceroute -I www.google.com
traceroute to www.google.com (142.250.182.132), 64 hops max, 72 byte packets
 1  172.20.10.1 (172.20.10.1)  6.410 ms  7.046 ms  3.849 ms
 2  10.50.96.4 (10.50.96.4)  51.916 ms  25.278 ms  47.840 ms
 3  10.50.96.202 (10.50.96.202)  32.859 ms  30.812 ms  46.478 ms
 4  * * *
 5  10.206.30.165 (10.206.30.165)  65.667 ms *  64.287 ms
 6  dsl-ncr-dynamic-029.24.23.125.airtelbroadband.in (125.23.24.29)  33.288 ms  27.572 ms  70.316 ms
 7  72.14.217.194 (72.14.217.194)  40.951 ms  34.434 ms  49.928 ms
 8  108.170.251.113 (108.170.251.113)  57.602 ms  44.432 ms  38.607 ms
 9  108.170.251.124 (108.170.251.124)  30.196 ms  37.938 ms  46.312 ms
10  142.250.63.116 (142.250.63.116)  32.727 ms *  59.518 ms
11  72.14.239.59 (72.14.239.59)  65.117 ms  67.514 ms  81.057 ms
12  74.125.242.145 (74.125.242.145)  77.900 ms  62.557 ms  73.127 ms
13  216.239.56.63 (216.239.56.63)  69.814 ms  80.008 ms  61.855 ms
14  maa05s22-in-f4.1e100.net (142.250.182.132)  78.131 ms  66.956 ms  69.986 ms
Yogeshs-MacBook-Air:~ pranshu$
```

- Traceroute to www.google.com using IITD Wi-Fi

```
Yogeshs-MacBook-Air:~ pranshu$ traceroute -I www.google.com
traceroute to www.google.com (142.250.77.68), 64 hops max, 72 byte packets
 1  10.184.0.14 (10.184.0.14)  2.089 ms  3.579 ms  1.722 ms
 2  10.255.1.34 (10.255.1.34)  1.794 ms  1.918 ms  1.797 ms
 3  10.119.233.65 (10.119.233.65)  1.996 ms  3.856 ms  1.928 ms
 4  10.1.207.69 (10.1.207.69)  3.520 ms  3.575 ms  3.434 ms
 5  10.119.234.162 (10.119.234.162)  3.619 ms  3.446 ms  3.432 ms
 6  72.14.195.56 (72.14.195.56)  3.897 ms  4.209 ms  4.402 ms
 7  108.170.251.108 (108.170.251.108)  5.027 ms  4.914 ms  4.759 ms
 8  142.250.230.114 (142.250.230.114)  27.813 ms  27.485 ms  28.524 ms
 9  216.239.54.93 (216.239.54.93)  27.102 ms  27.608 ms  27.090 ms
10  108.170.248.193 (108.170.248.193)  26.937 ms  27.040 ms  26.916 ms
11  142.250.238.197 (142.250.238.197)  29.319 ms  29.460 ms  29.088 ms
12  bom07s27-in-f4.1e100.net (142.250.77.68)  27.526 ms  27.610 ms  28.355 ms
Yogeshs-MacBook-Air:~ pranshu$
```

- The Internet is a large and complex aggregation of network hardware, connected together by gateways. Tracking the route one's packets follow (or finding the miscreant gateway that's discarding your packets) can be difficult. **traceroute** utilizes the IP protocol `time to live' field and attempts to elicit an ICMP TIME_EXCEEDED response from each gateway along the path to some host.
- '* * *' at any hop represents that the particular gateway either doesn't send ICMP "time exceeded" messages or sends them with a TTL too small to reach us (hop 4 in the 1st traceroute above).
- In the above two instances of traceroute, all the gateways use IPv4. In case some paths were to default to IPv6, we could use some additional flags to force traceroute to use IPv4.
- In general, traceroute uses IPv4 by default. We could use **traceroute6** to print the route IPv6 packets will take to a network node
- Traceroute by default uses UDP datagrams which are unreliable and thus some servers in the path may refuse to respond. So, we use the **-I** flag which forces traceroute to use ICMP ECHO instead of UDP datagrams. This change helps some of the missing routers to reply.
- Another possible attempt to make some of the missing routers to reply is by increasing the number of probes per ttl which is generally 3 by default.
- In both the traceroutes, several of the initial IP addresses such as 10.*.*.* and 172.20.10.1 are private i.e. they have been reserved by the **Internet Assigned Numbers Authority** (IANA) for private networks.

```
Yogeshs-MacBook-Air:~ pranshu$ traceroute6 -I www.google.com
connect: No route to host
Yogeshs-MacBook-Air:~ pranshu$ traceroute6 -I www.google.com
traceroute6 to www.google.com (2404:6800:4009:81b::2004) from 2401:4900:4456:34b1:a9c2:516e:3833:bbe6, 64 hops max, 16 byte packets
 1  2401:4900:4456:34b1:2568:f0b4:411:5141  476.543 ms  10.746 ms  3.497 ms
 2  2401:4900:4456:34b1:0:4a:74d7:2340  50.834 ms  326.160 ms  207.614 ms
 3  * * *
 4  2401:4900:0:c001::101  66.806 ms  28.851 ms  39.863 ms
 5  2401:4900:0:c001::17e  45.027 ms  56.019 ms  42.784 ms
 6  2401:4900:0:c001::17b  40.648 ms  36.078 ms  28.799 ms
 7  2404:a800:1a00:500::d  47.076 ms  55.486 ms  36.526 ms
 8  2001:4860:1:1::10c4  31.253 ms  29.699 ms  40.069 ms
 9  2404:6800:8172::1  52.528 ms  51.625 ms  41.032 ms
10  * *
    2001:4860:0:9e::1  51.197 ms
11  2001:4860:0:9e::3  46.332 ms  80.735 ms  26.909 ms
12  2001:4860::9:4000:d772  46.797 ms  89.443 ms  54.590 ms
13  * * *
14  2001:4860:0:1::4c71  81.517 ms  88.434 ms  77.980 ms
15  bom12s10-in-x04.1e100.net  44.955 ms  87.461 ms  63.309 ms
Yogeshs-MacBook-Air:~ pranshu$
```

- traceroute6 calls to www.google.com using IITD Wi-Fi and AIRTEL 4G respectively.

2. Packet Analysis

A.

- As we can observe in the screenshot below, the DNS request-response takes approximately 50ms to complete.
- Here, we have used Wireshark which is a very useful tool to sniff packets on the wire.
- First, we use the **sudo killall -HUP mDNSResponder; sleep 2;** command to flush the local DNS cache and also clear the Safari browser cache and history. Then, we apply a DNS filter using Wireshark which grabs all packets while we visit http://apache.org from our browser.

No.	Time	Source	Destination	Protocol	Length	Info
295	14.550139	10.184.31.187	1.1.1.1	DNS	70	Standard query 0x5996 HTTPS apache.org
296	14.550943	10.184.31.187	1.1.1.1	DNS	70	Standard query 0x8b17 A apache.org
297	14.601401	1.1.1.1	10.184.31.187	DNS	86	Standard query response 0x8b17 A apache.org A 151.101.2.132
301	14.622234	1.1.1.1	10.184.31.187	DNS	154	Standard query response 0x5996 HTTPS apache.org SOA ns-558.awsdns-05.net

B.

- Next, we apply an HTTP filter on the packet trace and visit the same website again after flushing the local DNS cache and clearing the browser cache as well.
- Approx **32 HTTP GET requests** were generated in loading the web page as can be counted from the screenshot attached below.

No.	Time	Source	Destination	Protocol	Length	Info
138	6.402724	10.184.31.187	151.101.2.132	HTTP	418	GET / HTTP/1.1
189	6.414315	151.101.2.132	10.184.31.187	HTTP	105	HTTP/1.1 200 OK (text/html)
191	6.426577	10.184.31.187	151.101.2.132	HTTP	393	GET /css/min.bootstrap.css HTTP/1.1
192	6.428009	10.184.31.187	151.101.2.132	HTTP	386	GET /css/styles.css HTTP/1.1
210	6.445058	10.184.31.187	151.101.2.132	HTTP	448	GET /img/support-apache.jpg HTTP/1.1
211	6.445905	10.184.31.187	151.101.2.132	HTTP	477	GET /img/trillions-and-trillions/why-apache-thumbail.jpg HTTP/1.1
212	6.446183	10.184.31.187	151.101.2.132	HTTP	485	GET /img/trillions-and-trillions/apache-everywhere-thumbnail.jpg HTTP/1.1
215	6.447020	10.184.31.187	151.101.2.132	HTTP	452	GET /img/asf-estd-1999-logo.jpg HTTP/1.1
329	6.462677	151.101.2.132	10.184.31.187	HTTP	290	HTTP/1.1 200 OK (JPEG JFIF image)
346	6.463349	10.184.31.187	151.101.2.132	HTTP	379	GET /js/jquery-2.1.1.min.js HTTP/1.1
524	6.480884	151.101.2.132	10.184.31.187	HTTP	385	HTTP/1.1 200 OK (JPEG JFIF image)
528	6.480890	151.101.2.132	10.184.31.187	HTTP	249	HTTP/1.1 200 OK (application/javascript)
533	6.481814	10.184.31.187	151.101.2.132	HTTP	372	GET /js/bootstrap.js HTTP/1.1
534	6.482012	10.184.31.187	151.101.2.132	HTTP	372	GET /js/slideshow.js HTTP/1.1
569	6.489572	151.101.2.132	10.184.31.187	HTTP	265	HTTP/1.1 200 OK (application/javascript)
573	6.490323	10.184.31.187	151.101.2.132	HTTP	491	GET /img/trillions-and-trillions/trillions-and-trillions-thumbnail.jpg HTTP/1.1
642	6.499428	151.101.2.132	10.184.31.187	HTTP	415	HTTP/1.1 200 OK (JPEG JFIF image)
644	6.499819	10.184.31.187	151.101.2.132	HTTP	485	GET /img/trillions-and-trillions/apache-innovation-thumbnail.jpg HTTP/1.1
697	6.511716	151.101.2.132	10.184.31.187	HTTP	211	HTTP/1.1 200 OK (JPEG JFIF image)
701	6.512261	10.184.31.187	151.101.2.132	HTTP	445	GET /img/2020-report.jpg HTTP/1.1
795	6.525617	151.101.2.132	10.184.31.187	HTTP	536	HTTP/1.1 200 OK (JPEG JFIF image)
800	6.526080	10.184.31.187	151.101.2.132	HTTP	443	GET /img/community.jpg HTTP/1.1
877	6.608284	151.101.2.132	10.184.31.187	HTTP	428	HTTP/1.1 200 OK (text/css)
885	6.608569	151.101.2.132	10.184.31.187	HTTP	395	HTTP/1.1 200 OK (text/css)
889	6.609834	10.184.31.187	151.101.2.132	HTTP	448	GET /img/the-apache-way.jpg HTTP/1.1
897	6.616899	10.184.31.187	151.101.2.132	HTTP	443	GET /img/ApacheCon.jpg HTTP/1.1
1097	6.641575	151.101.2.132	10.184.31.187	HTTP	303	HTTP/1.1 200 OK (JPEG JFIF image)
1098	6.641576	151.101.2.132	10.184.31.187	HTTP	377	HTTP/1.1 200 OK (JPEG JFIF image)
1103	6.642191	10.184.31.187	151.101.2.132	HTTP	456	GET /logos/res/libcloud/default.png HTTP/1.1
1104	6.642624	10.184.31.187	151.101.2.132	HTTP	456	GET /logos/res/geronimo/default.png HTTP/1.1
1269	6.664688	151.101.2.132	10.184.31.187	HTTP	545	HTTP/1.1 200 OK (PNG)
1275	6.665109	10.184.31.187	151.101.2.132	HTTP	452	GET /logos/res/reef/default.png HTTP/1.1
1499	6.720881	151.101.2.132	10.184.31.187	HTTP	564	HTTP/1.1 200 OK (JPEG JFIF image)
1501	6.721418	10.184.31.187	151.101.2.132	HTTP	453	GET /logos/res/mxnet/default.png HTTP/1.1
1550	6.771571	10.184.31.187	216.58.203.14	HTTP	404	GET /cse.js?cx=005703438322411770421:5mgshgrgx2u HTTP/1.1
1714	6.920877	216.58.203.14	10.184.31.187	HTTP	286	HTTP/1.1 404 Not Found (text/html)
1927	7.017655	151.101.2.132	10.184.31.187	HTTP	232	HTTP/1.1 200 OK (JPEG JFIF image)
1935	7.020919	151.101.2.132	10.184.31.187	HTTP	384	HTTP/1.1 200 OK (application/javascript)
1947	7.029377	10.184.31.187	151.101.2.132	HTTP	397	GET /fonts/glyphicons-halflings-regular.woff2 HTTP/1.1
1953	7.033418	10.184.31.187	151.101.2.132	HTTP	456	GET /logos/res/nlpcraft/default.png HTTP/1.1
1990	7.041640	151.101.2.132	10.184.31.187	HTTP	575	HTTP/1.1 200 OK (PNG)
1994	7.068767	10.184.31.187	151.101.2.132	HTTP	453	GET /logos/res/toree/default.png HTTP/1.1
2583	7.173069	151.101.2.132	10.184.31.187	HTTP	262	HTTP/1.1 200 OK (PNG)
2978	7.201740	151.101.2.132	10.184.31.187	HTTP	430	HTTP/1.1 200 OK (PNG)
3423	7.224714	151.101.2.132	10.184.31.187	HTTP	331	HTTP/1.1 200 OK (JPEG JFIF image)
3772	7.266675	151.101.2.132	10.184.31.187	HTTP	253	HTTP/1.1 200 OK (PNG)
4247	7.291697	151.101.2.132	10.184.31.187	HTTP	145	HTTP/1.1 200 OK (PNG)
4423	7.320577	10.184.31.187	216.58.203.14	HTTP	388	GET /adsense/search/async-ads.js HTTP/1.1
4544	7.465459	10.184.31.187	142.250.67.174	HTTP	447	GET /generate_204 HTTP/1.1
4614	7.492198	142.250.67.174	10.184.31.187	HTTP	149	HTTP/1.1 204 No Content
4657	7.513977	216.58.203.14	10.184.31.187	HTTP	274	HTTP/1.1 200 OK (text/javascript)
4716	7.553184	151.101.2.132	10.184.31.187	HTTP	235	HTTP/1.1 200 OK (font/woff2)
5242	8.183366	10.184.31.187	151.101.2.132	HTTP	385	GET /favicons/favicon-194x194.png HTTP/1.1
5245	8.185041	10.184.31.187	151.101.2.132	HTTP	392	GET /favicons/android-chrome-192x192.png HTTP/1.1
5248	8.189653	10.184.31.187	151.101.2.132	HTTP	383	GET /favicons/favicon-96x96.png HTTP/1.1
5254	8.191593	10.184.31.187	151.101.2.132	HTTP	383	GET /favicons/favicon-32x32.png HTTP/1.1
5260	8.195836	10.184.31.187	151.101.2.132	HTTP	377	GET /favicons/favicon.ico HTTP/1.1
5261	8.196333	10.184.31.187	151.101.2.132	HTTP	383	GET /favicons/favicon-16x16.png HTTP/1.1
5266	8.196836	151.101.2.132	10.184.31.187	HTTP	469	HTTP/1.1 200 OK (PNG)
5312	8.372741	151.101.2.132	10.184.31.187	HTTP	311	HTTP/1.1 200 OK (PNG)
5317	8.379765	151.101.2.132	10.184.31.187	HTTP	464	HTTP/1.1 200 OK (PNG)
5346	8.384369	151.101.2.132	10.184.31.187	HTTP	128	HTTP/1.1 200 OK (PNG)
5380	8.734875	151.101.2.132	10.184.31.187	HTTP	128	HTTP/1.1 200 OK (PNG)
5404	8.880952	151.101.2.132	10.184.31.187	HTTP	126	HTTP/1.1 200 OK (PNG)

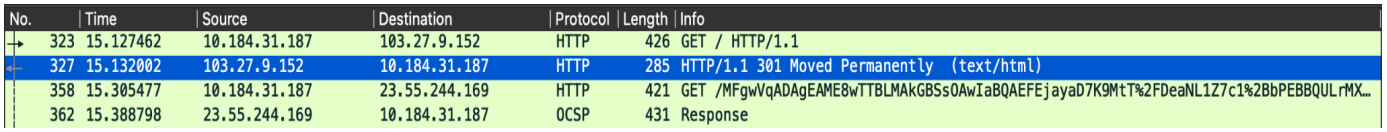
- If we observe the HTTP request-responses in the above output, we can understand how web pages are structured as various components such as images, text, and styles stored separately.
- We can observe that different requests are made to the JavaScript, Image , Logos, and CSS folders for different components of the website.
- The first GET request is made to download the HTML page of the website which contains links to various objects such as the JS file, CSS file, fonts, and images required by the webpage. Subsequently, GET requests corresponding to these objects are made which are then rendered upon receiving the HTTP response.

C.

- The total time is taken to download a webpage is calculated as the time between the first DNS request and the time when the last content object was received.
- To calculate the above time, we use Wireshark to grab all the packets while visiting http://apache.org and then apply DNS and HTTP filters to obtain the time when the first DNS request was made and the time when the last content object was received, respectively.
- It took approximately 2.66 seconds to download the entire webpage.

D.

- Following is the Wireshark output after running a packet trace for http://www.cse.iitd.ac.in with an HTTP filter.



No.	Time	Source	Destination	Protocol	Length	Info
323	15.127462	10.184.31.187	103.27.9.152	HTTP	426	GET / HTTP/1.1
327	15.132002	103.27.9.152	10.184.31.187	HTTP	285	HTTP/1.1 301 Moved Permanently (text/html)
358	15.305477	10.184.31.187	23.55.244.169	HTTP	421	GET /MFgwVqADAgEAME8wTTBLMAkGBSsOAwIaBQAEFEjayaD7K9MtT%2FDeaNL1Z7c1%2BbPEBBQULrMX…
362	15.388798	23.55.244.169	10.184.31.187	OCSP	431	Response

- We observe that we get an **HTTP/1.1 301 Moved Permanently** response which is used for permanent redirecting, meaning current links or records using the URL this response is received for should be updated.
- The 301 redirects are considered a best practice for upgrading users from HTTP to HTTPS.
- The possible reason behind this may be that http://www.cse.iitd.ac.in uses HTTPS instead of HTTP, thus we do not obtain any HTTP traffic while visiting the website.
- We were easily obtaining the HTTP traffic for http://apache.org possibly because it uses HTTP and not HTTPS.
- HTTPS is basically HTTP with encryption. The only difference between the two protocols is that HTTPS uses TLS (SSL) to encrypt normal HTTP requests and responses. As a result, **HTTPS is far more secure than HTTP.**

3. **Implementing** traceroute **using** ping

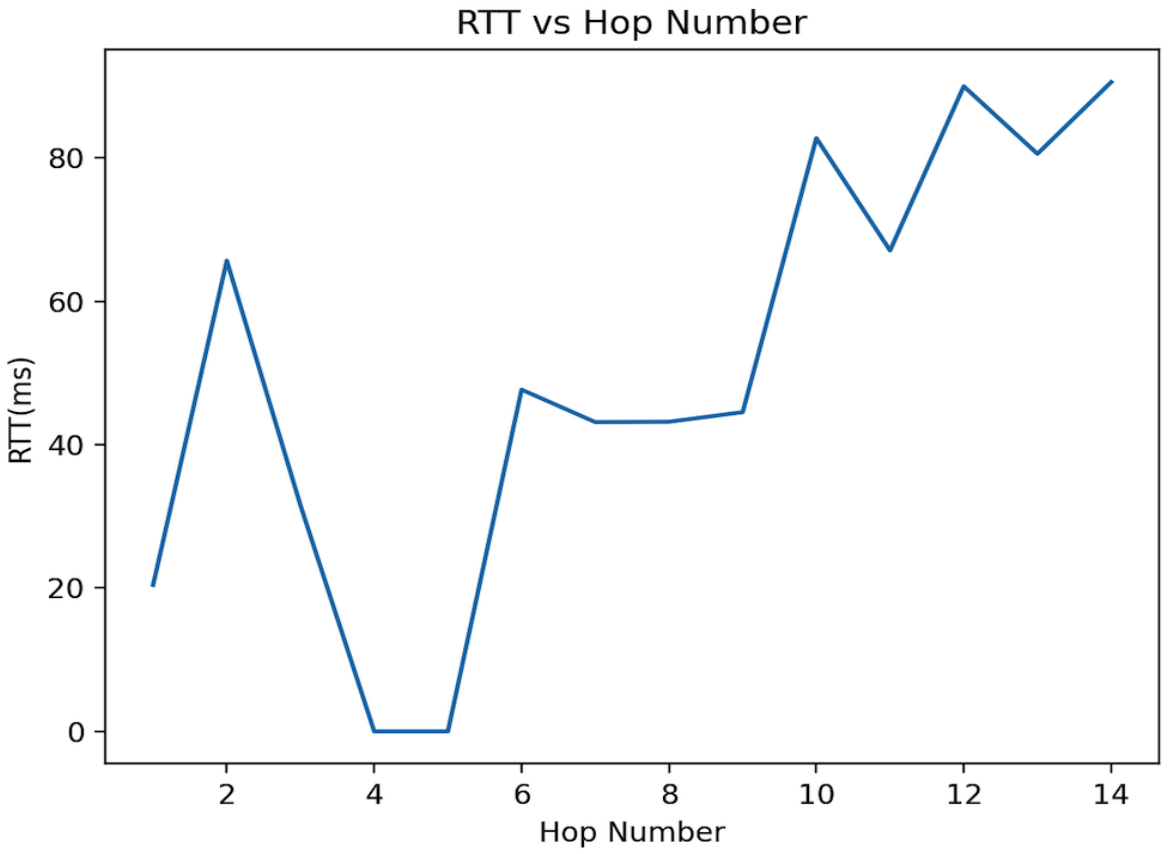
- The task is to implement a traceroute using the ping command available in the terminal. We need to take the domain name as input and print the IP addresses of the hops on the way to the destination IP address and also save the RTT vs Hop plot.

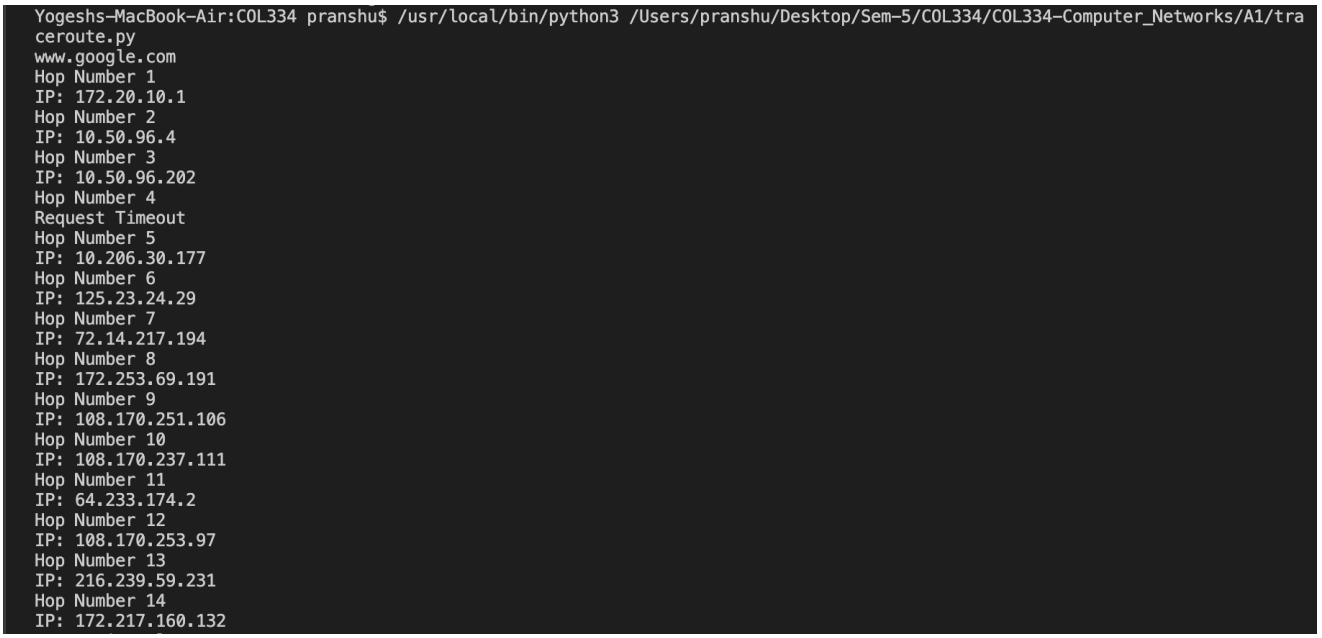
  Algorithm

  - We know that **traceroute** utilizes the IP protocol `time to live' field and attempts to elicit an ICMP TIME_EXCEEDED response from each gateway along the path to some host. We use a similar strategy to implement traceroute using ping, hop by hop.
  - We ping the domain in consideration with varying ttl values starting from ttl = 1 until we reach a point where we can successfully transmit the ping packet to the required destination.
  - For all the ttl values lesser than the minimum number of hops required to reach the destination address, we obtain the following message on the terminal.

```
Yogeshs-MacBook-Air:~ pranshu$ ping -m 1 -c 1 www.google.com
PING www.google.com (142.250.196.4): 56 data bytes
36 bytes from 172.20.10.1: Time to live exceeded
Vr HL TOS  Len   ID Flg  off TTL Pro  cks      Src      Dst
 4  5  00 5400 8322   0 0000  01  01 2d6e 172.20.10.6  142.250.196.4
```

  - We observe that the displayed message contains the IP address of the router up to which the ping packet could reach. We can extract the IP address of the router from this message.
  - Using the above strategy, we store the IP addresses of all the routers on the path to the destination address. Then we ping each such address to obtain the RTT(Round Trip Time) of each server on the way and plot the required graph.
  - The following plot was obtained for www.google.com





Terminal output for www.google.com