Computer Networks
Assignment-1

HIMANSHU GAURAV SINGH
2019CS10358

August 21, 2021

Contents

1 Networking Tools

1.a IP-address of personal machine

Ran the command *ipconfig getifaddr en0* to get the IP address of the machine - **10.184.20.241**. This was when I was logged in to IITD-Wifi. For my personal mobile phone hotspot (ISP-Reliance Jio), the IP address turns out to be **192.168.43.172**.
The IP address changed on changing the ISP. This is reasonable since it is the router that assigns an IP address to the interface between them.

1.b IP-address of various websites

Ran the command `nslookup <domain-name>` to get the IP address associated with the websites.
On using the default DNS server for the ISP, the IP addresses come as follows -

Webpage	RELIANCE-JIO	IITD-WIFI
www.google.com	142.250.77.206	142.250.182.174
www.facebook.com	157.240.198.35	31.13.79.35

On using the Google public DNS, the results are -

Webpage	RELIANCE-JIO	IITD-WIFI
www.google.com	142.250.195.14	142.250.193.238
www.facebook.com	157.240.198.35	157.240.16.35

1.b.1 Explanation

- Mostly(but not always), the change is in the last two sections of the IP address.
- This is happening probably because different DNS servers point to different host servers attached to the same domain name. It is likely that popular websites such as those of Google and Facebook have multiple servers.

1.c Ping

Ran the command `ping -s <size>-m <ttl-value><domain-name>`.
I did a binary-search for the maximum length of the packets and the minimum ttl value by repeatedly running the above command. The results for the minimum TTL value for which the packet could be sent and the maximum size of the packet that could be sent are as follows :

Domain-name	Minimum TTL	Maximum packet size(bytes)
www.iitd.ac.in	4	8184
www.google.com	10	68
www.facebook.com	12	1472

The packet size here excludes the header of size 8 bytes of the ICMP packet.

1. The lower value of TTL for iitd.ac.in is evident since I was logged on to IITD-WiFi itself implying vicinity within the network to the IITD server.
2. Regarding the maximum packet size, for the IITD server, the maximum packet size that could be sent by my machine, was 8192(2^{13}).
3. The limit of 1472 bytes for facebook.com is probably due to the a cap of 1500 bytes on MTU(Maximum Transmission Unit) over the Ethernet. It corresponds to 1500 - 8 (ICMP header) - 20(IP header) = 1472 bytes.
4. Thus, it is evident that maximum packet size might depend upon the number of routers in between the client and server apart from system specification. Probably, different routers/end-systems have different limitations over the size of the packets they send. Over and above all, we can send messages of at most total length 65536 bytes using IPv4.

1.d Traceroute

Ran the command `traceroute iitd.ac.in`

1.d.1 www.iitd.ac.in

Using personal hotspot (Reliance-JIO), the traceroute comes out as follows:

```
traceroute to iitd.ac.in (103.27.9.24), 64 hops max, 52 byte packets
 1  192.168.43.1 (192.168.43.1) 1032.835 ms 5.549 ms 7.418 ms
 2  * * *
 3  10.71.83.34 (10.71.83.34) 206.575 ms 41.420 ms
    10.71.83.50 (10.71.83.50) 59.662 ms
 4  172.26.100.119 (172.26.100.119) 28.985 ms 73.069 ms 35.736 ms
 5  172.26.100.102 (172.26.100.102) 64.987 ms
    172.26.100.103 (172.26.100.103) 57.922 ms 46.272 ms
 6  192.168.44.26 (192.168.44.26) 38.798 ms
    192.168.44.28 (192.168.44.28) 59.328 ms
    192.168.44.22 (192.168.44.22) 42.704 ms
 7  192.168.44.25 (192.168.44.25) 57.403 ms
    192.168.44.29 (192.168.44.29) 63.647 ms
    192.168.44.23 (192.168.44.23) 37.225 ms
 8  172.26.14.73 (172.26.14.73) 63.038 ms
    172.16.26.5 (172.16.26.5) 64.028 ms
    172.26.14.73 (172.26.14.73) 47.789 ms
 9  172.16.18.0 (172.16.18.0) 47.890 ms 38.507 ms 42.070 ms
10  115.254.77.34 (115.254.77.34) 64.113 ms 71.759 ms
    115.110.210.37.static-delhi.vsnl.net.in (115.110.210.37) 27.637 ms
11  * * 115.255.253.18 (115.255.253.18) 87.734 ms
12  14.140.210.22.static-delhi-vsnl.net.in (14.140.210.22) 65.979 ms
    115.249.198.97 (115.249.198.97) 102.746 ms 111.101 ms
13  * * *
14  * * *
15  * * *
..........
64  * * *
```

Following is the result of tracing a path to iitd.ac.in while being logged on to IIT-D Wifi.

```
traceroute to iitd.ac.in (10.10.211.212), 64 hops max, 52 byte packets
 1  10.194.0.14 (10.194.0.14) 2.368 ms 1.804 ms 1.705 ms
 2  10.254.238.5 (10.254.238.5) 2.560 ms 2.251 ms 2.126 ms
 3  10.254.236.22 (10.254.236.22) 1.995 ms 2.208 ms
    10.254.236.14 (10.254.236.14) 2.640 ms
 4  www.iitd.ac.in (10.10.211.212) 1.768 ms 1.768 ms 2.036 ms
```

The short path is reasonable since the corresponding routers are reasonably close in the network.

1.d.2 www.google.com

Following is the route when the client machine used IITD-WiFi.

```
traceroute to google.com (142.250.192.174), 64 hops max, 52 byte packets
 1  10.194.0.14 (10.194.0.14) 2.936 ms 1.364 ms 1.565 ms
 2  10.254.238.1 (10.254.238.1) 1.738 ms 1.748 ms 1.627 ms
 3  10.255.1.34 (10.255.1.34) 2.046 ms 1.911 ms 1.748 ms
 4  10.119.233.65 (10.119.233.65) 2.207 ms 2.168 ms 1.858 ms
 5  10.1.207.69 (10.1.207.69) 3.532 ms 3.720 ms 3.613 ms
 6  10.119.234.162 (10.119.234.162) 3.759 ms 4.236 ms 4.880 ms
 7  72.14.194.160 (72.14.194.160) 4.042 ms
    72.14.195.56 (72.14.195.56) 4.632 ms 4.408 ms
 8  74.125.243.97 (74.125.243.97) 5.698 ms 6.752 ms 5.975 ms
 9  172.253.73.195 (172.253.73.195) 5.387 ms 5.374 ms 5.305 ms
10  del11s11-in-f14.1e100.net (142.250.192.174) 5.221 ms 5.462 ms 5.034 ms
```

Following is the route when personal smartphone hotspot is used.

```
traceroute to google.com (142.250.77.238), 64 hops max, 52 byte packets
 1  192.168.43.1 (192.168.43.1) 3.055 ms 2.826 ms 2.568 ms
 2  * * *
 3  10.71.83.34 (10.71.83.34) 158.438 ms 60.948 ms 59.493 ms
 4  172.26.100.119 (172.26.100.119) 60.005 ms 60.113 ms 58.518 ms
 5  172.26.100.103 (172.26.100.103) 60.136 ms 46.796 ms
    172.26.100.102 (172.26.100.102) 59.396 ms
 6  192.168.44.22 (192.168.44.22) 46.309 ms
    192.168.44.26 (192.168.44.26) 28.829 ms
    192.168.44.22 (192.168.44.22) 30.452 ms
 7  192.168.44.29 (192.168.44.29) 99.432 ms 42.589 ms
    192.168.44.27 (192.168.44.27) 283.440 ms
 8  172.16.26.5 (172.16.26.5) 65.128 ms
    172.16.18.33 (172.16.18.33) 62.436 ms 57.711 ms
 9  172.16.26.2 (172.16.26.2) 56.066 ms
    172.16.18.2 (172.16.18.2) 57.201 ms
    172.16.26.0 (172.16.26.0) 66.966 ms
10  142.250.168.56 (142.250.168.56) 66.164 ms 55.939 ms
    142.250.47.144 (142.250.47.144) 35.637 ms
11  * * *
12  72.14.233.30 (72.14.233.30) 73.834 ms
    64.233.174.70 (64.233.174.70) 316.269 ms
    142.251.52.210 (142.251.52.210) 30.268 ms
13  142.251.54.77 (142.251.54.77) 37.824 ms
```

```
108.170.251.123 (108.170.251.123) 39.945 ms
del11s09-in-f14.1e100.net (142.250.77.238) 61.703 ms
```

1.d.3 Observations

1. Running `traceroute` multiple times gives different routes each time, which is reasonable since there can be several routes to the same end-system in the network.

2. For the traceroute to IIT-D server, several of the first IP addresses are private addresses of the smartphone/local networks. After that, there are addresses belonging to Reliance(the ISP)(115.254.77.34). After that, we do not get any response, probably because packets to IIT-D servers are being blocked.

3. In this case, 192.168.43.1 is the IP address of my smartphone. Observe that some of the routers in the path(hop 5 and 11) did not respond back, probably due to the smaller TTL value of the packet they sent. `traceroute` uses UDP-datagrams by default which are unreliable in general.

4. In the case of IITD-WiFi, there were several private IP addresses, probably IIT-D servers followed by IP addresses belonging to Google. This shows that some Google-owned routers probably share a direct connection to one of the IIT-D router.

1.d.4 Observations regarding IPv6

1. `traceroute` uses IPv4 by default. To use IPv6, run the command `traceroute6`.

2. `iitd.ac.in` does not respond to IPv6 protocol. The message `traceroute6:nodename nor servname provided, or not known` is received. Running `traceroute6 google.com` while being logged on to IITD-WiFi gives `connect:No route to host`. Evidently, IITD routers and end-systems do not support IPv6.

3. Running the above command while using personal hotspot gave the following result

```
traceroute6 to google.com (2404:6800:4002:81f::200e) from 2409:4050:2e0a:6973:99c5:7b42:774b:e776
    , 64 hops max, 12 byte packets
 1  2409:4050:2e0a:6973::ef 3.256 ms 3.137 ms 2.304 ms
 2  * * *
 3  2405:200:331:eeee:20::506 204.242 ms 32.619 ms 44.323 ms
 4  2405:200:801:300::e74 35.228 ms
    2405:200:801:300::e78 40.288 ms 47.670 ms
 5  2405:200:801:300::e77 50.763 ms 40.116 ms 49.098 ms
 6  2405:200:801:300::dcf 32.674 ms
    2405:200:801:300::dd1 47.584 ms 105.846 ms
 7  2405:200:801:300::75 42.606 ms
    2405:200:801:300::64b 30.728 ms
    2405:200:801:300::75 50.004 ms
 8  2001:4860:1:1::1ef4 40.550 ms 41.583 ms *
 9  2404:6800:8074::1 396.781 ms
    2404:6800:8106::1 30.667 ms *
10  *
    2001:4860:0:9e::1 204.714 ms
    2001:4860:0:1::538a 44.494 ms
11  *
    2001:4860:0:1a::3 191.673 ms
    2001:4860:0:9e::5 42.347 ms
12  2001:4860:0:11dd::1 38.693 ms 40.986 ms
    2001:4860::1c:4000:eaf6 38.954 ms
13  2001:4860:0:1::53ab 41.944 ms
    del12s02-in-x0e.1e100.net 52.155 ms
    2001:4860:0:1::53a9 44.174 ms
```

1.d.5 Attempting to get replies from more routers

1. One possible way to make some of the missing routers to reply is to increase the number of packets sent per ttl value by traceroute. By default, three packets are sent per ttl value which can be modified using the $-q$ option. Since traceroute sends UDP packets which are unreliable, increasing the number of packets sent increases the chances of the servers replying to them. However, this method will not work if there is some problem from the side of the router.

2. Also, we can switch to using ICMP echoes rather than UDP datagrams by turning the -I option on. This improves the responsiveness of the packets, but not all.

```
       traceroute to google.com (142.250.194.174), 64 hops max, 72 byte packets
 1  192.168.43.1 (192.168.43.1) 39.277 ms 6.941 ms 3.303 ms
 2  * * *
 3  * * 56.8.174.189 (56.8.174.189) 257.410 ms
 4  192.168.44.232 (192.168.44.232) 42.685 ms
    192.168.44.234 (192.168.44.234) 56.921 ms 40.665 ms
 5  192.168.44.235 (192.168.44.235) 39.128 ms
    192.168.44.239 (192.168.44.239) 59.026 ms
    192.168.44.235 (192.168.44.235) 37.958 ms
 6  172.26.100.117 (172.26.100.117) 63.602 ms 43.879 ms 56.350 ms
 7  172.26.100.99 (172.26.100.99) 35.368 ms 43.039 ms 58.958 ms
 8  192.168.44.26 (192.168.44.26) 29.773 ms
    192.168.44.28 (192.168.44.28) 71.500 ms
    192.168.44.26 (192.168.44.26) 36.642 ms
 9  192.168.44.29 (192.168.44.29) 29.435 ms * 202.799 ms
10  172.16.26.5 (172.16.26.5) 31.702 ms 39.413 ms 39.080 ms
11  172.16.26.2 (172.16.26.2) 49.471 ms 49.645 ms 35.318 ms
12  142.250.47.144 (142.250.47.144) 38.285 ms 52.689 ms 40.021 ms
13  142.251.66.169 (142.251.66.169) 37.662 ms 44.418 ms 29.927 ms
14  142.251.52.219 (142.251.52.219) 50.826 ms 35.004 ms 37.083 ms
15  del12s06-in-f14.1e100.net (142.250.194.174) 47.935 ms 68.260 ms 41.932 ms
```

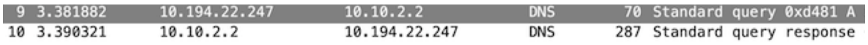
Observe that the number of unresponsive routers is one less than the case in which the default traceroute was used.

2 Packet analysis

Clicking directly on the link to the webpage given in the problem statement first sent the request to a Google server(probably a search request) and after that, to the server of Apache. It was better to browse using the URL itself.

2.a DNS filter

On loading the webpage directly from the browser, several DNS messages corresponding to subdomains(for example, YouTube) were being recorded. To avoid that, I fetched the page using wget command from terminal. The following was the response to the dns filter.



```
 9 3.381882     10.194.22.247     10.10.2.2         DNS      70 Standard query 0xd481 A
10 3.390321     10.10.2.2         10.194.22.247     DNS     287 Standard query response
```

The first packet is a DNS query and the next is the query response corresponding to that sent from the DNS server. The whole process took approximately 8.5 ms.

2.b HTTP filter

The HTTP messages sent/received are as in the figure when the webpage was requesting using Google Chrome. There are approximately 25 request messages and 25 response messages.

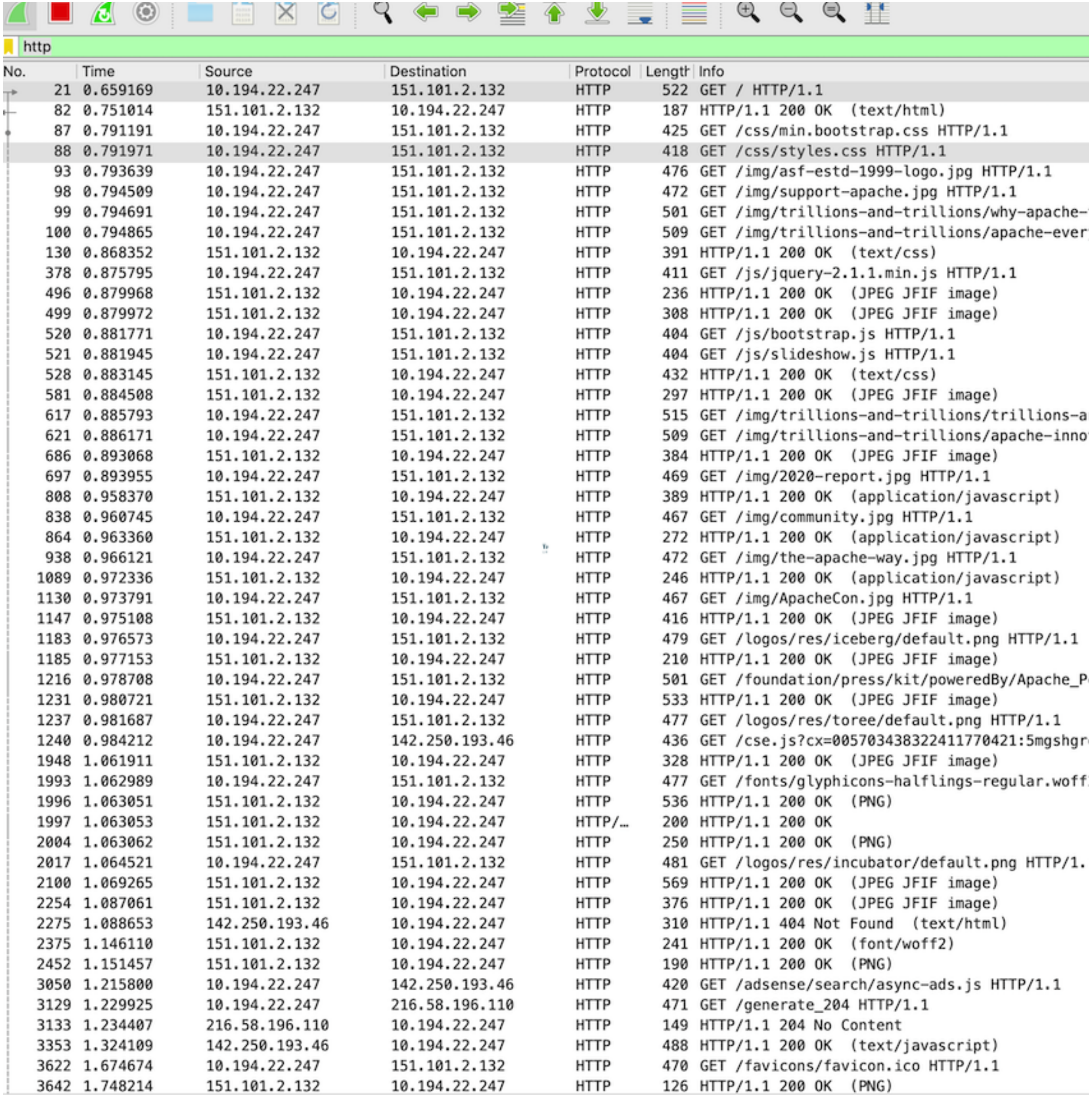
1. Observe that the text, images, the Javascript and CSS of the webpage, all are retrieved separately.

2. Also, when the webpage was fetched using the wget command, the only HTTP messages visible were the first GET message and the next response message corresponding to the base HTML file.

3. This shows that the different components of the webpage, the text, graphics, the fonts etc are all stored in separate files in the server.

The browser first downloads the base(mostly textual) HTML page. Apart from the text segment, it contains links to the various other objects such as the CSS file, the JS file and the images wherever necessary. As the web browser renders the HTML page, it encounters the links to these files and requests for the same.

Figure 1: HTTP messages



http

No.	Time	Source	Destination	Protocol	Length	Info
21	0.659169	10.194.22.247	151.101.2.132	HTTP	522	GET / HTTP/1.1
82	0.751014	151.101.2.132	10.194.22.247	HTTP	187	HTTP/1.1 200 OK (text/html)
87	0.791191	10.194.22.247	151.101.2.132	HTTP	425	GET /css/min.bootstrap.css HTTP/1.1
88	0.791971	10.194.22.247	151.101.2.132	HTTP	418	GET /css/styles.css HTTP/1.1
93	0.793639	10.194.22.247	151.101.2.132	HTTP	476	GET /img/asf-estd-1999-logo.jpg HTTP/1.1
98	0.794509	10.194.22.247	151.101.2.132	HTTP	472	GET /img/support-apache.jpg HTTP/1.1
99	0.794691	10.194.22.247	151.101.2.132	HTTP	501	GET /img/trillions-and-trillions/why-apache-
100	0.794865	10.194.22.247	151.101.2.132	HTTP	509	GET /img/trillions-and-trillions/apache-ever
130	0.868352	151.101.2.132	10.194.22.247	HTTP	391	HTTP/1.1 200 OK (text/css)
378	0.875795	10.194.22.247	151.101.2.132	HTTP	411	GET /js/jquery-2.1.1.min.js HTTP/1.1
496	0.879968	151.101.2.132	10.194.22.247	HTTP	236	HTTP/1.1 200 OK (JPEG JFIF image)
499	0.879972	151.101.2.132	10.194.22.247	HTTP	308	HTTP/1.1 200 OK (JPEG JFIF image)
520	0.881771	10.194.22.247	151.101.2.132	HTTP	404	GET /js/bootstrap.js HTTP/1.1
521	0.881945	10.194.22.247	151.101.2.132	HTTP	404	GET /js/slideshow.js HTTP/1.1
528	0.883145	151.101.2.132	10.194.22.247	HTTP	432	HTTP/1.1 200 OK (text/css)
581	0.884508	151.101.2.132	10.194.22.247	HTTP	297	HTTP/1.1 200 OK (JPEG JFIF image)
617	0.885793	10.194.22.247	151.101.2.132	HTTP	515	GET /img/trillions-and-trillions/trillions-a
621	0.886171	10.194.22.247	151.101.2.132	HTTP	509	GET /img/trillions-and-trillions/apache-inno
686	0.893068	151.101.2.132	10.194.22.247	HTTP	384	HTTP/1.1 200 OK (JPEG JFIF image)
697	0.893955	10.194.22.247	151.101.2.132	HTTP	469	GET /img/2020-report.jpg HTTP/1.1
808	0.958370	151.101.2.132	10.194.22.247	HTTP	389	HTTP/1.1 200 OK (application/javascript)
838	0.960745	10.194.22.247	151.101.2.132	HTTP	467	GET /img/community.jpg HTTP/1.1
864	0.963360	151.101.2.132	10.194.22.247	HTTP	272	HTTP/1.1 200 OK (application/javascript)
938	0.966121	10.194.22.247	151.101.2.132	HTTP	472	GET /img/the-apache-way.jpg HTTP/1.1
1089	0.972336	151.101.2.132	10.194.22.247	HTTP	246	HTTP/1.1 200 OK (application/javascript)
1130	0.973791	10.194.22.247	151.101.2.132	HTTP	467	GET /img/ApacheCon.jpg HTTP/1.1
1147	0.975108	151.101.2.132	10.194.22.247	HTTP	416	HTTP/1.1 200 OK (JPEG JFIF image)
1183	0.976573	10.194.22.247	151.101.2.132	HTTP	479	GET /logos/res/iceberg/default.png HTTP/1.1
1185	0.977153	151.101.2.132	10.194.22.247	HTTP	210	HTTP/1.1 200 OK (JPEG JFIF image)
1216	0.978708	10.194.22.247	151.101.2.132	HTTP	501	GET /foundation/press/kit/poweredBy/Apache_P
1231	0.980721	151.101.2.132	10.194.22.247	HTTP	533	HTTP/1.1 200 OK (JPEG JFIF image)
1237	0.981687	10.194.22.247	151.101.2.132	HTTP	477	GET /logos/res/toree/default.png HTTP/1.1
1240	0.984212	10.194.22.247	142.250.193.46	HTTP	436	GET /cse.js?cx=005703438322411770421:5mgshgr
1948	1.061911	151.101.2.132	10.194.22.247	HTTP	328	HTTP/1.1 200 OK (JPEG JFIF image)
1993	1.062989	10.194.22.247	151.101.2.132	HTTP	477	GET /fonts/glyphicons-halflings-regular.woff
1996	1.063051	151.101.2.132	10.194.22.247	HTTP	536	HTTP/1.1 200 OK (PNG)
1997	1.063053	151.101.2.132	10.194.22.247	HTTP/…	200	HTTP/1.1 200 OK
2004	1.063062	151.101.2.132	10.194.22.247	HTTP	250	HTTP/1.1 200 OK (PNG)
2017	1.064521	10.194.22.247	151.101.2.132	HTTP	481	GET /logos/res/incubator/default.png HTTP/1.
2100	1.069265	151.101.2.132	10.194.22.247	HTTP	569	HTTP/1.1 200 OK (JPEG JFIF image)
2254	1.087061	151.101.2.132	10.194.22.247	HTTP	376	HTTP/1.1 200 OK (JPEG JFIF image)
2275	1.088653	142.250.193.46	10.194.22.247	HTTP	310	HTTP/1.1 404 Not Found (text/html)
2375	1.146110	151.101.2.132	10.194.22.247	HTTP	241	HTTP/1.1 200 OK (font/woff2)
2452	1.151457	151.101.2.132	10.194.22.247	HTTP	190	HTTP/1.1 200 OK (PNG)
3050	1.215800	10.194.22.247	142.250.193.46	HTTP	420	GET /adsense/search/async-ads.js HTTP/1.1
3129	1.229925	10.194.22.247	216.58.196.110	HTTP	471	GET /generate_204 HTTP/1.1
3133	1.234407	216.58.196.110	10.194.22.247	HTTP	149	HTTP/1.1 204 No Content
3353	1.324109	142.250.193.46	10.194.22.247	HTTP	488	HTTP/1.1 200 OK (text/javascript)
3622	1.674674	10.194.22.247	151.101.2.132	HTTP	470	GET /favicons/favicon.ico HTTP/1.1
3642	1.748214	151.101.2.132	10.194.22.247	HTTP	126	HTTP/1.1 200 OK (PNG)

2.c Time taken to load webpage

The time taken to load the webpage was approx 1.18 seconds. It was calculated by subtracting the time when the first `GET` request was sent from the time when final HTTP response was received.

2.d Comparison with cse.iitd.ac.in

The HTTP response to a query to iitd.ac.in was a message with code 301(Permanently moved). I tried it with `twitter.com` and `facebook.com`, both of them showed no activity under HTTP filter.
This is probably due to the fact that the websites use HTTPS rather than HTTP which is more secure and the files transferred over it are encrypted. The corresponding messages can be viewed under the SSL filter.

3 Implementing `traceroute` using `ping`

3.a Algorithm

1. The working of `traceroute` can be found on the man page for it. We implement it in a way similar to that.

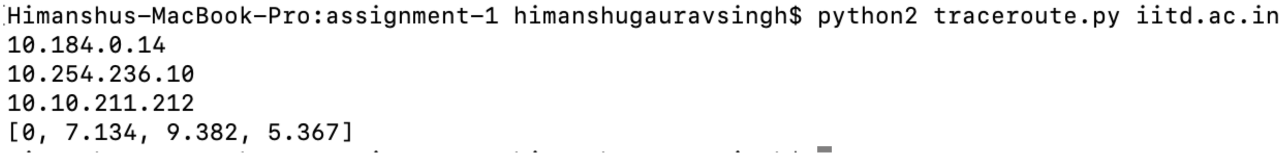
2. We successively ping at the required domain name while incrementing the ttl value of the ping starting from 1. For each ping, if the ttl limit is reached, an error message is received that contains the IP address of the last reached router which can be parsed easily.

3. In the other case, it might happen that the router does not respond to the ping at all. To handle that, we send multiple pings to the router to increase the probability of it responding. In case it does not respond at all. We declare a timeout and proceed to the next ttl value.

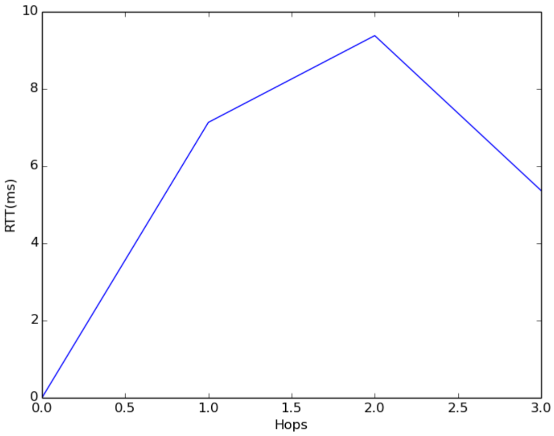
4. The script can be run by `python2 traceroute.py <domain-name>`

3.b Output and plots

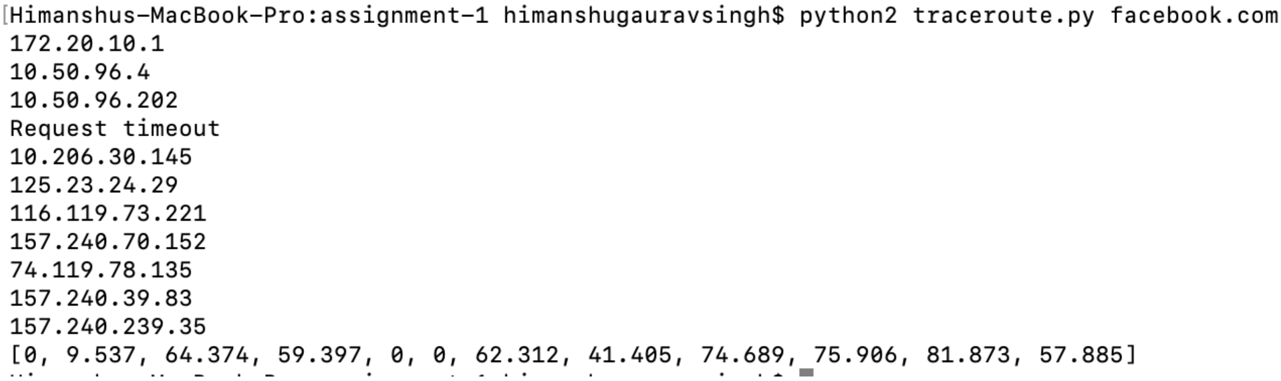
The following is the output for `www.iitd.ac.in`.The machine was connected to IITD-WiFi.



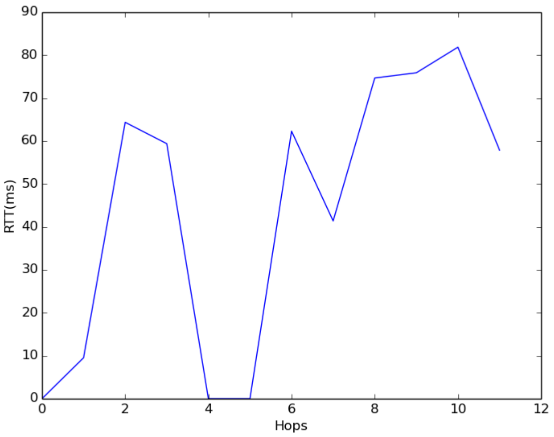
```
Himanshus-MacBook-Pro:assignment-1 himanshugauravsingh$ python2 traceroute.py iitd.ac.in
10.184.0.14
10.254.236.10
10.10.211.212
[0, 7.134, 9.382, 5.367]
```



The following is the output for `www.facebook.com`.The machine was connected to an Airtel-hotspot.



```
Himanshus-MacBook-Pro:assignment-1 himanshugauravsingh$ python2 traceroute.py facebook.com
172.20.10.1
10.50.96.4
10.50.96.202
Request timeout
10.206.30.145
125.23.24.29
116.119.73.221
157.240.70.152
74.119.78.135
157.240.39.83
157.240.239.35
[0, 9.537, 64.374, 59.397, 0, 0, 62.312, 41.405, 74.689, 75.906, 81.873, 57.885]
```



3.b.1 Observation

1. Observe that the plots are not strictly increasing as might be expected. This is probably due to variable latency of the routers and end-systems over time and for different pings.

2. Several of the routers do not respond back to the sender, causing a time-out response. Also, this behaviour is not consistent over multiple runs.