COL334: Computer Networks

Assignment 1

Anirudha Kulkarni
2019CS50421

August 21, 2021

1. Networking Tools

(a) IP address of the machine changes with different routers. Each router assigns a private IP address to device which can be different.

(a) IP Address with Airtel router: 192.168.1.101

(b) IP Address with BSNL router: 192.168.1.12

(c) IP Address with Jio mobile hotspot: 192.168.43.179

(b) (a) IP address of www.google.com with google dns (8.8.8.8) : 142.250.182.206

(b) IP address of www.facebook.com with google dns (8.8.8.8) : 157.240.1.35

(c) IP address of www.google.com with multiplay.bsnl.in (218.248.114.1) : 216.58.221.46

(d) IP address of www.facebook.com with multiplay.bsnl.in (218.248.114.1) : 31.13.79.35

(c) TTL limits the number of hops a packet can cross. Lower values gives time to live exceeded error as the packet can not reach the destination in limited hops. Received packets have fixed TTL values as it is the response from server. Actual packet size include 8 bytes for ICMP packet header and 20 bytes for IP header.

(a) Max size for www.iitd.ac.in: 1472 bytes (1500 bytes total)

(b) Max size for www.google.com: 68 bytes (96 bytes total)

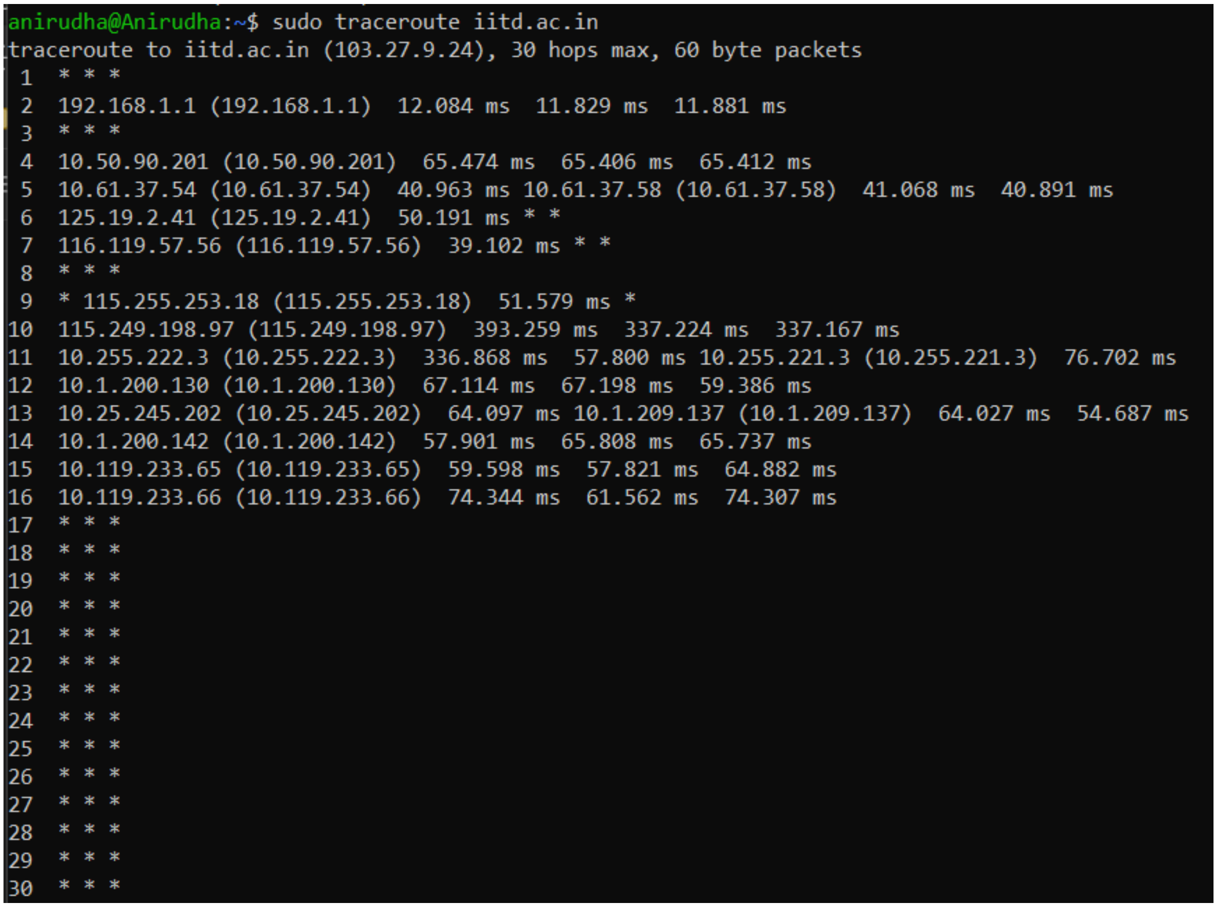
(c) Max size for www.facebook.com: 1472 bytes (1500 bytes total)

(d) Observations:

(a) UDP based traceroute generally require large number of hops before reaching to destination as most of the routers do not reply to UDP protocol as it is unreliable protocol. The Internet Control Message Protocol (ICMP) is a network layer protocol used by network devices to diagnose network communication issues which is default in Windows. Linux, by default, uses UDP.

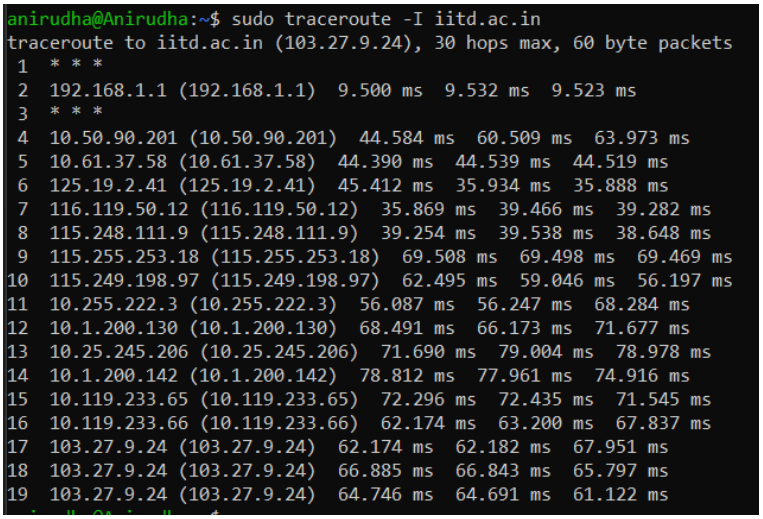
(b) -I parameter in Linux can make traceroute to send ICMP packets

(c) Some paths by default use IPv6 and can be made to use IPv4 with -4 argument. This works only when resolving a host name returns both IPv4 and IPv6 addresses. Similarly -6 forces to use IPv6.



```
anirudha@Anirudha:~$ sudo traceroute iitd.ac.in
traceroute to iitd.ac.in (103.27.9.24), 30 hops max, 60 byte packets
 1  * * *
 2  192.168.1.1 (192.168.1.1)  12.084 ms  11.829 ms  11.881 ms
 3  * * *
 4  10.50.90.201 (10.50.90.201)  65.474 ms  65.406 ms  65.412 ms
 5  10.61.37.54 (10.61.37.54)  40.963 ms 10.61.37.58 (10.61.37.58)  41.068 ms  40.891 ms
 6  125.19.2.41 (125.19.2.41)  50.191 ms * *
 7  116.119.57.56 (116.119.57.56)  39.102 ms * *
 8  * * *
 9  * 115.255.253.18 (115.255.253.18)  51.579 ms *
10  115.249.198.97 (115.249.198.97)  393.259 ms  337.224 ms  337.167 ms
11  10.255.222.3 (10.255.222.3)  336.868 ms  57.800 ms 10.255.221.3 (10.255.221.3)  76.702 ms
12  10.1.200.130 (10.1.200.130)  67.114 ms  67.198 ms  59.386 ms
13  10.25.245.202 (10.25.245.202)  64.097 ms 10.1.209.137 (10.1.209.137)  64.027 ms  54.687 ms
14  10.1.200.142 (10.1.200.142)  57.901 ms  65.808 ms  65.737 ms
15  10.119.233.65 (10.119.233.65)  59.598 ms  57.821 ms  64.882 ms
16  10.119.233.66 (10.119.233.66)  74.344 ms  61.562 ms  74.307 ms
17  * * *
18  * * *
19  * * *
20  * * *
21  * * *
22  * * *
23  * * *
24  * * *
25  * * *
26  * * *
27  * * *
28  * * *
29  * * *
30  * * *
```

Figure 1: UDP traceroute - many servers did not reply



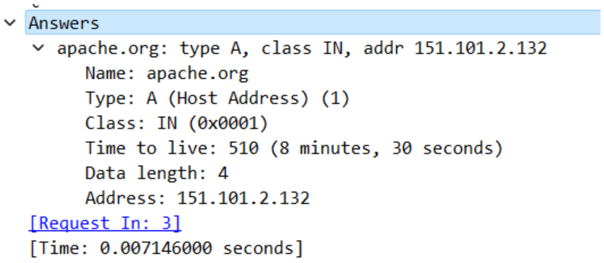
```
anirudha@Anirudha:~$ sudo traceroute -I iitd.ac.in
traceroute to iitd.ac.in (103.27.9.24), 30 hops max, 60 byte packets
 1  * * *
 2  192.168.1.1 (192.168.1.1)  9.500 ms  9.532 ms  9.523 ms
 3  * * *
 4  10.50.90.201 (10.50.90.201)  44.584 ms  60.509 ms  63.973 ms
 5  10.61.37.58 (10.61.37.58)  44.390 ms  44.539 ms  44.519 ms
 6  125.19.2.41 (125.19.2.41)  45.412 ms  35.934 ms  35.888 ms
 7  116.119.50.12 (116.119.50.12)  35.869 ms  39.466 ms  39.282 ms
 8  115.248.111.9 (115.248.111.9)  39.254 ms  39.538 ms  38.648 ms
 9  115.255.253.18 (115.255.253.18)  69.508 ms  69.498 ms  69.469 ms
10  115.249.198.97 (115.249.198.97)  62.495 ms  59.046 ms  56.197 ms
11  10.255.222.3 (10.255.222.3)  56.087 ms  56.247 ms  68.284 ms
12  10.1.200.130 (10.1.200.130)  68.491 ms  66.173 ms  71.677 ms
13  10.25.245.206 (10.25.245.206)  71.690 ms  79.004 ms  78.978 ms
14  10.1.200.142 (10.1.200.142)  78.812 ms  77.961 ms  74.916 ms
15  10.119.233.65 (10.119.233.65)  72.296 ms  72.435 ms  71.545 ms
16  10.119.233.66 (10.119.233.66)  62.174 ms  63.200 ms  67.837 ms
17  103.27.9.24 (103.27.9.24)  62.174 ms  62.182 ms  67.951 ms
18  103.27.9.24 (103.27.9.24)  66.885 ms  66.843 ms  65.797 ms
19  103.27.9.24 (103.27.9.24)  64.746 ms  64.691 ms  61.122 ms
```

Figure 2: ICMP traceroute

2. Packet Analysis

Corresponding wire-shark snapshot is attached with the zip file.

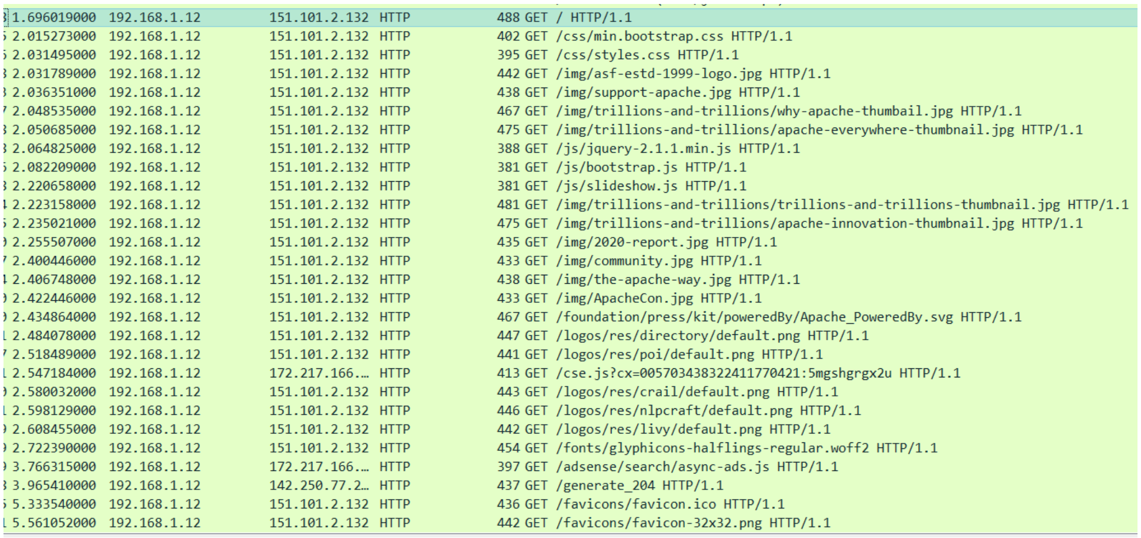
(a) DNS request took 1.686420000 - 1.679274000 = 0.007146 sec = 7.146 milliseconds.



```
Answers
  apache.org: type A, class IN, addr 151.101.2.132
      Name: apache.org
      Type: A (Host Address) (1)
      Class: IN (0x0001)
      Time to live: 510 (8 minutes, 30 seconds)
      Data length: 4
      Address: 151.101.2.132
[Request In: 3]
[Time: 0.007146000 seconds]
```

Figure 3: DNS request response

(b) Approximately 28 http requests were generated. Most of the initial requests (25) are made to 151.101.2.232 which is IP address of the http://apache.org. These requests fetch the HTML content then CSS for styling, images and javascript for dynamic behaviour. Towards the end HTTP requests are made to 172.217.166.238 for google search optimization and 142.250.77.238 for advertisements on the website.

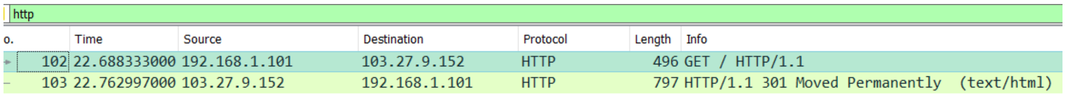


```
1.696019000  192.168.1.12    151.101.2.132  HTTP    488 GET / HTTP/1.1
2.015273000  192.168.1.12    151.101.2.132  HTTP    402 GET /css/min.bootstrap.css HTTP/1.1
2.031495000  192.168.1.12    151.101.2.132  HTTP    395 GET /css/styles.css HTTP/1.1
2.031789000  192.168.1.12    151.101.2.132  HTTP    442 GET /img/asf-estd-1999-logo.jpg HTTP/1.1
2.036351000  192.168.1.12    151.101.2.132  HTTP    438 GET /img/support-apache.jpg HTTP/1.1
2.048535000  192.168.1.12    151.101.2.132  HTTP    467 GET /img/trillions-and-trillions/why-apache-thumbail.jpg HTTP/1.1
2.050685000  192.168.1.12    151.101.2.132  HTTP    475 GET /img/trillions-and-trillions/apache-everywhere-thumbnail.jpg HTTP/1.1
2.064825000  192.168.1.12    151.101.2.132  HTTP    388 GET /js/jquery-2.1.1.min.js HTTP/1.1
2.082209000  192.168.1.12    151.101.2.132  HTTP    381 GET /js/bootstrap.js HTTP/1.1
2.220658000  192.168.1.12    151.101.2.132  HTTP    381 GET /js/slideshow.js HTTP/1.1
2.223158000  192.168.1.12    151.101.2.132  HTTP    481 GET /img/trillions-and-trillions/trillions-and-trillions-thumbnail.jpg HTTP/1.1
2.235021000  192.168.1.12    151.101.2.132  HTTP    475 GET /img/trillions-and-trillions/apache-innovation-thumbnail.jpg HTTP/1.1
2.255507000  192.168.1.12    151.101.2.132  HTTP    435 GET /img/2020-report.jpg HTTP/1.1
2.400446000  192.168.1.12    151.101.2.132  HTTP    433 GET /img/community.jpg HTTP/1.1
2.406748000  192.168.1.12    151.101.2.132  HTTP    438 GET /img/the-apache-way.jpg HTTP/1.1
2.422446000  192.168.1.12    151.101.2.132  HTTP    433 GET /img/ApacheCon.jpg HTTP/1.1
2.434864000  192.168.1.12    151.101.2.132  HTTP    467 GET /foundation/press/kit/poweredBy/Apache_PoweredBy.svg HTTP/1.1
2.484078000  192.168.1.12    151.101.2.132  HTTP    447 GET /logos/res/directory/default.png HTTP/1.1
2.518489000  192.168.1.12    151.101.2.132  HTTP    441 GET /logos/res/poi/default.png HTTP/1.1
2.547184000  192.168.1.12    172.217.166.…  HTTP    413 GET /cse.js?cx=005703438322411770421:5mgshgrgx2u HTTP/1.1
2.580032000  192.168.1.12    151.101.2.132  HTTP    443 GET /logos/res/crail/default.png HTTP/1.1
2.598129000  192.168.1.12    151.101.2.132  HTTP    446 GET /logos/res/nlpcraft/default.png HTTP/1.1
2.608455000  192.168.1.12    151.101.2.132  HTTP    442 GET /logos/res/livy/default.png HTTP/1.1
2.722390000  192.168.1.12    151.101.2.132  HTTP    454 GET /fonts/glyphicons-halflings-regular.woff2 HTTP/1.1
3.766315000  192.168.1.12    172.217.166.…  HTTP    397 GET /adsense/search/async-ads.js HTTP/1.1
3.965410000  192.168.1.12    142.250.77.2…  HTTP    437 GET /generate_204 HTTP/1.1
5.333540000  192.168.1.12    151.101.2.132  HTTP    436 GET /favicons/favicon.ico HTTP/1.1
5.561052000  192.168.1.12    151.101.2.132  HTTP    442 GET /favicons/favicon-32x32.png HTTP/1.1
```

Figure 4: HTTP requests

(c) Last packet (2084th packet) arrival time (including google search manager and advertisement resources) : 5.838546000 sec. Total time taken = 5.838546000 - 1.679274000 = 4.159272 seconds.

(d) There is only 1 request and corresponding response via HTTP protocol. GET request to http://www.cse.iitd.ac.in was responded with 301 response i.e. web-page moved permanently to https://www.cse.iitd.ac.in which uses HTTPS protocol. HTTPS traffic is encrypted using TLS protocol and hence can not be intercepted in clear text. The encrypted data shared can be filtered with TLS filter.



http

o.	Time	Source	Destination	Protocol	Length	Info
102	22.688333000	192.168.1.101	103.27.9.152	HTTP	496	GET / HTTP/1.1
103	22.762997000	103.27.9.152	192.168.1.101	HTTP	797	HTTP/1.1 301 Moved Permanently (text/html)

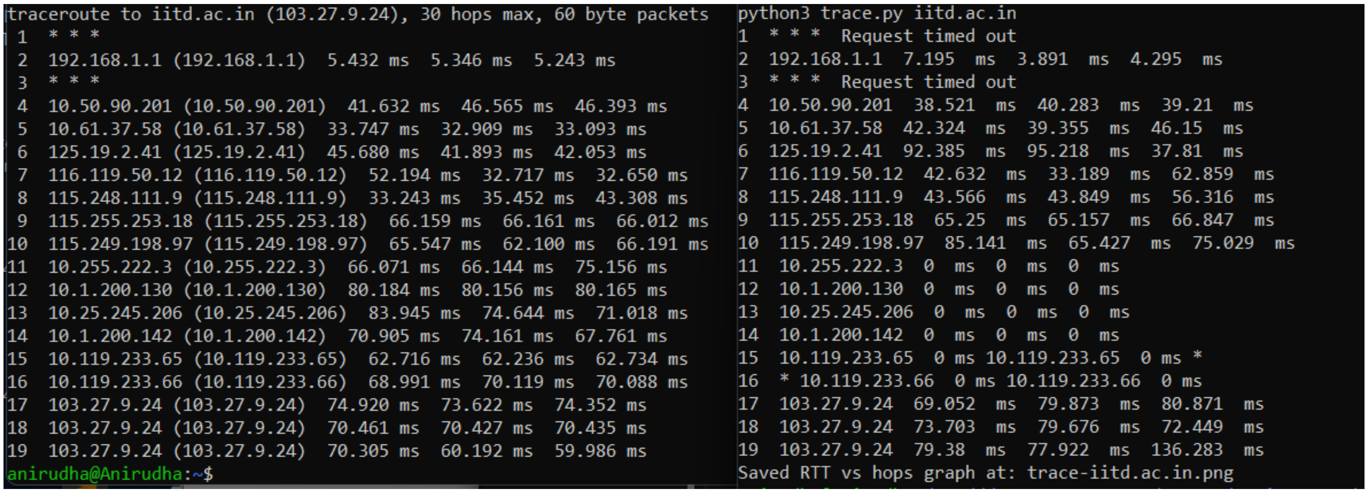
Figure 5: Permanently moved response

3. Implement Traceroute using Ping

Traceroute involves following steps:

1. initialize t=1
2. ping destination with ttl=t and get the IP address at which the Time To Live exceeded message appears. This is the hop where packet expired.
3. ping the IP address from (2) with a large TTL (say 50) to get RTT for the IP.
4. repeat 2-3 with t=t+1 until the destination is reached.
5. repeat 1-2-3-4 steps 3 times to remove any bias or alternate path taken by the packet.

Traceroute implementation with ping command in python hop by hop Usage: python3 trace.py <domain>



```
traceroute to iitd.ac.in (103.27.9.24), 30 hops max, 60 byte packets
 1  * * *
 2  192.168.1.1 (192.168.1.1)  5.432 ms  5.346 ms  5.243 ms
 3  * * *
 4  10.50.90.201 (10.50.90.201)  41.632 ms  46.565 ms  46.393 ms
 5  10.61.37.58 (10.61.37.58)  33.747 ms  32.909 ms  33.093 ms
 6  125.19.2.41 (125.19.2.41)  45.680 ms  41.893 ms  42.053 ms
 7  116.119.50.12 (116.119.50.12)  52.194 ms  32.717 ms  32.650 ms
 8  115.248.111.9 (115.248.111.9)  33.243 ms  35.452 ms  43.308 ms
 9  115.255.253.18 (115.255.253.18)  66.159 ms  66.161 ms  66.012 ms
10  115.249.198.97 (115.249.198.97)  65.547 ms  62.100 ms  66.191 ms
11  10.255.222.3 (10.255.222.3)  66.071 ms  66.144 ms  75.156 ms
12  10.1.200.130 (10.1.200.130)  80.184 ms  80.156 ms  80.165 ms
13  10.25.245.206 (10.25.245.206)  83.945 ms  74.644 ms  71.018 ms
14  10.1.200.142 (10.1.200.142)  70.905 ms  74.161 ms  67.761 ms
15  10.119.233.65 (10.119.233.65)  62.716 ms  62.236 ms  62.734 ms
16  10.119.233.66 (10.119.233.66)  68.991 ms  70.119 ms  70.088 ms
17  103.27.9.24 (103.27.9.24)  74.920 ms  73.622 ms  74.352 ms
18  103.27.9.24 (103.27.9.24)  70.461 ms  70.427 ms  70.435 ms
19  103.27.9.24 (103.27.9.24)  70.305 ms  60.192 ms  59.986 ms
anirudha@Anirudha:~$
```

```
python3 trace.py iitd.ac.in
1  * * *  Request timed out
2  192.168.1.1  7.195  ms  3.891  ms  4.295  ms
3  * * *  Request timed out
4  10.50.90.201  38.521  ms  40.283  ms  39.21  ms
5  10.61.37.58  42.324  ms  39.355  ms  46.15  ms
6  125.19.2.41  92.385  ms  95.218  ms  37.81  ms
7  116.119.50.12  42.632  ms  33.189  ms  62.859  ms
8  115.248.111.9  43.566  ms  43.849  ms  56.316  ms
9  115.255.253.18  65.25  ms  65.157  ms  66.847  ms
10  115.249.198.97  85.141  ms  65.427  ms  75.029  ms
11  10.255.222.3  0  ms  0  ms  0  ms
12  10.1.200.130  0  ms  0  ms  0  ms
13  10.25.245.206  0  ms  0  ms  0  ms
14  10.1.200.142  0  ms  0  ms  0  ms
15  10.119.233.65  0 ms 10.119.233.65  0 ms *
16  * 10.119.233.66  0 ms 10.119.233.66  0 ms
17  103.27.9.24  69.052  ms  79.873  ms  80.871  ms
18  103.27.9.24  73.703  ms  79.676  ms  72.449  ms
19  103.27.9.24  79.38  ms  77.922  ms  136.283  ms
Saved RTT vs hops graph at: trace-iitd.ac.in.png
```

Figure 6: traceroute command (left) vs custom code (right)

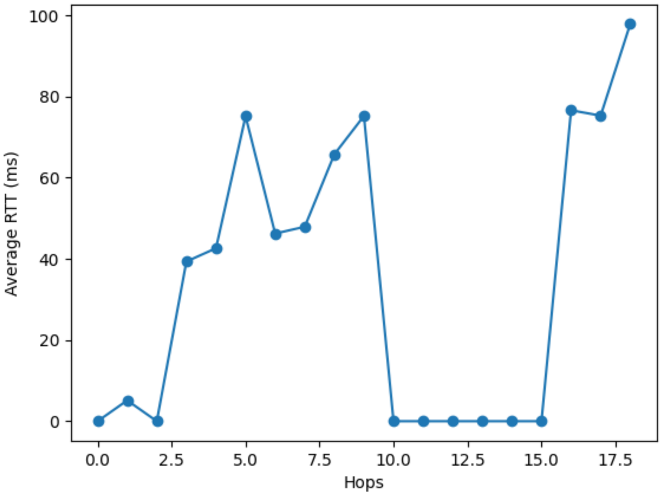


Figure 7: RTT vs Hops graph (servers that did not respond to ping are marked with 0 RTT)

Complexity vs Efficiency trade-off:

1. RTT of some intermediate servers is more noisy due to variation of load in routers. This can be removed by sending multiple packets across each iteration and taking average.
2. Currently default is average of 3 path each sending 2 packets. This can be changed in the program. Sending 3 packets in each iterations severely affects the complexity causing time to shoot up to 2 minutes
3. Sending single packet at 3 paths gives lot of noise and graph is difficult to make any inferences. Hence we choose 2 packets at 3 paths to get most optimal solution
4. Time complexity is further reduced by adding timeout of 1 seconds to reduce time spent in waiting for servers to respond that are not responding. The maximum RTT observed was less than 500ms hence giving 1 second timeout safely reduces the time by a significant factor.

```python
# imports
import sys
import subprocess
import time
import re
import matplotlib.pyplot as plt
# global constants
max_ttl=56
done=False
# function definitions


def get_ip(output):
    # print(output)
    result=re.search("Time to live exceeded",output)
    if(result is None):
        global done
        pattern ="[0-9]*\.[0-9]*\.[0-9]*\.[0-9]*: i"
        result = re.search(pattern,output)
        if result is not None:
            done=True
            return result.group()[0:-3]
        pattern ="\([0-9]*\.[0-9]*\.[0-9]*\.[0-9]*\): i"
        result = re.search(pattern,output)
        if result is not None:
            done=True
            return result.group()[1:-4]
        return "*"

    pattern ="\([0-9]*\.[0-9]*\.[0-9]*\.[0-9]*\) i"
    result = re.search(pattern,output)
    if result is not None:
        return result.group()[1:-3]
    pattern ="[0-9]*\.[0-9]*\.[0-9]*\.[0-9]* i"
    result = re.search(pattern,output)
    if result is not None:
        return result.group()[0:-2]



def callping(hostname,ttl):
    # get the ip address at current TTL
    proc = subprocess.Popen("ping -c 1 -W 1 "+hostname+" -t "+str(ttl), shell=True,stdout=subprocess.PIPE)
    (out, err) = proc.communicate()
    ip=get_ip(str(out))
    if(ip=="*"):
        return ["*",0]
    # get time required to reach the destination
    proc = subprocess.Popen("ping -c 2 -W 1 "+ip+" -t "+str(max_ttl), shell=True,stdout=subprocess.PIPE)
    (out, err) = proc.communicate()
    result = re.search("/[0-9]*\.[0-9]*/",str(out))
    if not result:
        return [ip,0]
    return [ip,float(result.group()[1:-1])]

# main
domain=sys.argv[1]
max_hop=30
total_hops=0
finalarr=[]
for i in range(1,max_hop):
    if done:
        break
    total_hops+=1
    ans=[]
    # send 3 packets
    for j in range(3):
        ans+=[callping(domain,i)]
    finalarr+=[ans]
    # format the output
    print(str(i),end="  ")
    if(ans[0][0]==ans[1][0] and ans[0][0]==ans[2][0]):
        if(ans[0][0]=="*"):
            print("* * *  Request timed out")
        else:
            print(ans[0][0],"",ans[0][1]," ms ",ans[1][1]," ms ",ans[2][1]," ms")


    else:
        for i in range(3):
            if(ans[i][0]=="*"):
                print("*",end=" ")
            else:
                print(ans[i][0],"",ans[i][1],end=" ms ")
        print()
# Plot the graph
garr=[]
for i in range(total_hops):
    garr+=[(finalarr[i][0][1]+finalarr[i][1][1]+finalarr[i][2][1])/3]
plt.plot([i for i in range(total_hops)],garr,marker='o' )
plt.xlabel('Hops')
plt.ylabel('Average RTT (ms)')
plt.savefig("trace-"+domain+".png")
plt.show()
print("Saved RTT vs hops graph at: trace-"+domain+".png")
```

Listing 1: Python implementation for traceroute using ping

4. List of Figures

5. Listings