

# COL334: Computer Networks

## Assignment 1

Anirudha Kulkarni  
2019CS50421

August 22, 2021

### 1. Networking Tools

---

- (a) IP address of the machine changes with different routers. Each router assigns a private IP address to each device to recognize it between systems connected to network, 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
  - (e) IP address of www.google.com with cisco OpenDNS (208.67.222.222) : 142.250.66.14
  - (f) IP address of www.facebook.com with cisco OpenDNS (208.67.222.222) : 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)

Pinging with different values of TTL gives information about the path taken by packet to the destination. Packets with less TTL values expire in the transit exposing the intermediate IP addresses. Ping to www.iitd.ac.in expire till TTL=18. It requires 19 hops to reach the destination.

```
ping iitd.ac.in -t 18
PING iitd.ac.in (103.27.9.24) 56(84) bytes of data.
From 103.27.9.24 (103.27.9.24) icmp_seq=1 Time to live exceeded      35
From 103.27.9.24 (103.27.9.24) icmp_seq=2 Time to live exceeded    36
^C                                                                    37
--- iitd.ac.in ping statistics ---
 2 packets transmitted, 0 received, +2 errors, 100% packet loss, time 1002ms
```

Figure 1: TTL = 18 packet expires before reaching destination

```

ping iitd.ac.in -t 19
PING iitd.ac.in (103.27.9.24) 56(84) bytes of data.
64 bytes from 103.27.9.24 (103.27.9.24): icmp_seq=1 ttl=51 time=53.7 ms
^C
--- iitd.ac.in ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 53.679/53.679/53.679/0.000 ms

```

Figure 2: TTL = 19 packet reaches destination

(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.

```

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 3: UDP traceroute - many servers did not reply

- (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 -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 4: 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 5: 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.
- (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.

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-thumbnail.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
2.25507000	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/craill/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/livvy/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 6: HTTP requests

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 7: Permanently moved response

### 3. Implement Traceroute using Ping

Traceroute involves following steps:

1. initialize  $t=1$
2. ping destination with  $t=t$  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 8: traceroute command for iitd.ac.in (left) vs custom code (right)

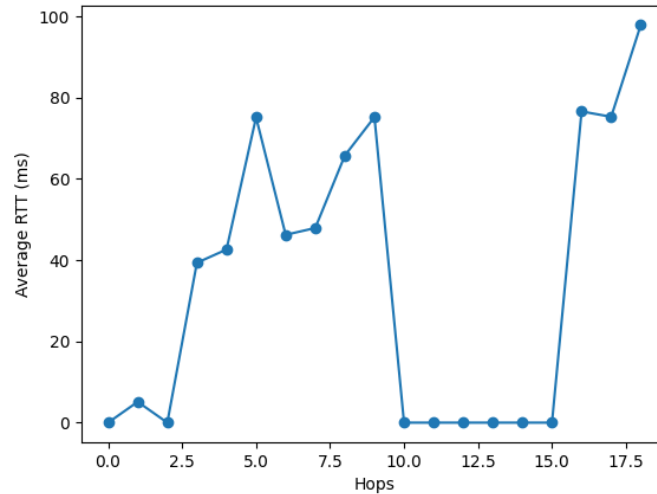


Figure 9: RTT vs Hops graph for iitd.ac.in (servers that did not respond to ping are marked with 0 RTT)

```

[sudo] password for anirudha:
traceroute to google.com (142.250.66.14), 30 hops max, 60 byte packets
1 * * *
2 192.168.1.1 (192.168.1.1) 0.828 ms 2.004 ms 2.026 ms
3 * * *
4 10.206.232.245 (10.206.232.245) 53.540 ms 53.538 ms 53.519 ms
5 10.61.37.58 (10.61.37.58) 78.592 ms 72.022 ms 62.350 ms
6 125.19.2.41 (125.19.2.41) 46.137 ms 45.346 ms 44.171 ms
7 116.119.73.92 (116.119.73.92) 47.711 ms 30.832 ms 30.964 ms
8 72.14.212.48 (72.14.212.48) 32.978 ms 36.299 ms 36.449 ms
9 209.85.246.11 (209.85.246.11) 44.416 ms 44.346 ms 44.404 ms
10 142.251.70.57 (142.251.70.57) 40.879 ms 40.733 ms 35.552 ms
11 bom07s35-in-f14.1e100.net (142.250.66.14) 22.422 ms 24.305 ms 18.402 ms

1 * * * Request timed out
2 192.168.1.1 0.682 ms 0.605 ms 0.969 ms
ping: google.com: Temporary failure in name resolution
3 * * * Request timed out
4 10.0.225.133 28.279 ms 10.50.90.201 28.935 ms 10.50.90.201 56.532 ms
5 10.61.37.62 42.396 ms 10.61.37.54 37.851 ms 10.61.37.62 47.693 ms
6 125.19.2.41 32.899 ms 144.782 ms 41.326 ms
7 182.79.239.125 28.996 ms 182.79.239.125 41.328 ms 116.119.44.115 37.924 ms
ping: google.com: Temporary failure in name resolution
8 72.14.212.48 31.094 ms 72.14.212.48 38.026 ms *
9 209.85.247.65 73.778 ms 41.43 ms 53.663 ms
10 142.251.70.57 26.772 ms 142.251.70.57 103.166 ms 216.239.47.149 48.56 ms
11 172.217.160.206 49.695 ms 172.217.160.206 55.117 ms 142.250.76.206 69.507 ms

```

Figure 10: traceroute command for google.com (left) vs custom code (right)

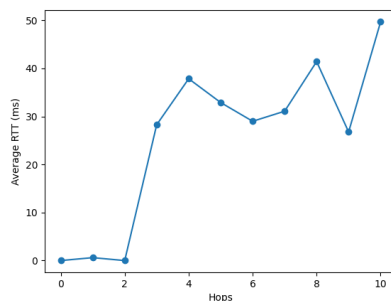


Figure 11: RTT vs Hops graph for google.com (servers that did not respond to ping are marked with 0 RTT)

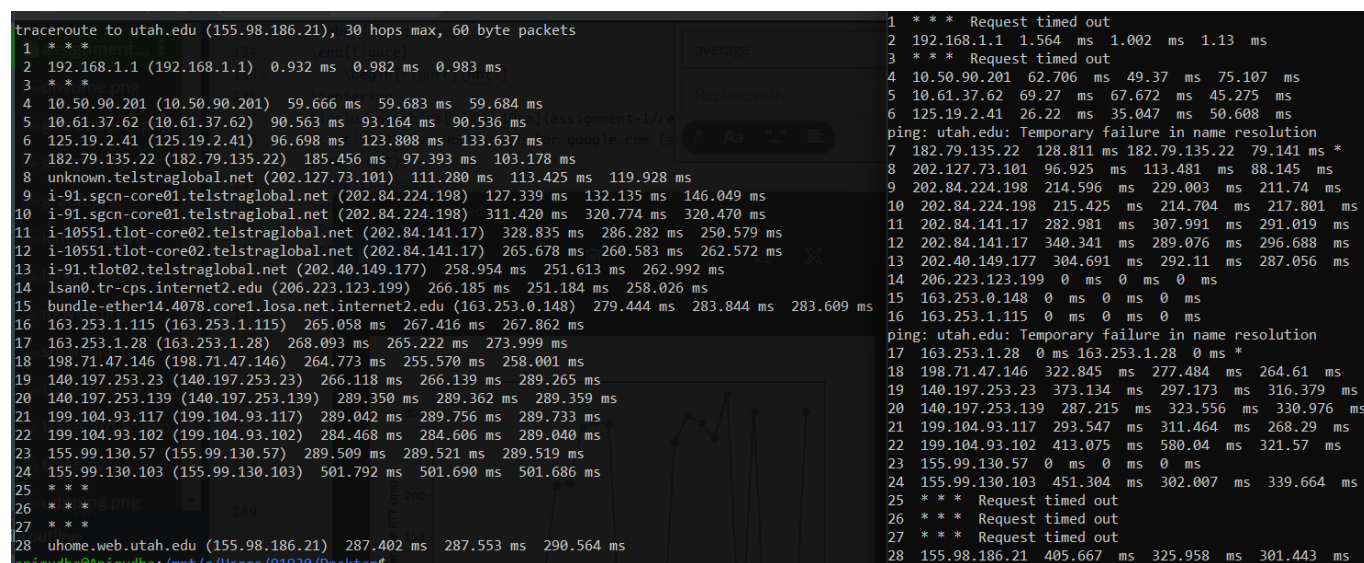


Figure 12: traceroute command for utah.edu (left) vs custom code (right)

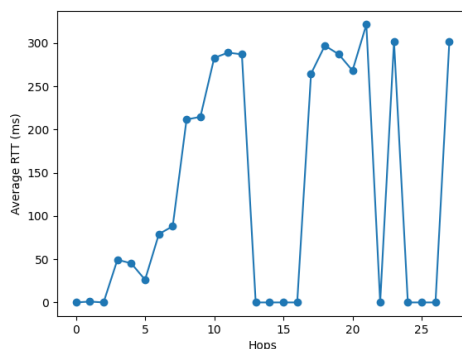


Figure 13: RTT vs Hops graph for utah.edu (servers that did not respond to ping are marked with 0 RTT)

## 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 minimum.
2. Currently default is minimum 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.

```

1 # imports
2 import sys
3 import subprocess
4 import time
5 import re
6 import matplotlib.pyplot as plt
7 # global constants
8 max_ttl=56
9 done=False
10 # function definitions
11
12
13 def get_ip(output):
14     # print(output)
15     result=re.search("Time to live exceeded",output)
16     if(result is None):
17         global done
18         pattern = "[0-9]*\.[0-9]*\.[0-9]*\.[0-9]*: i"
19         result = re.search(pattern,output)
20         if result is not None:
21             done=True
22             return result.group()[0:-3]
23         pattern = "\([0-9]*\.[0-9]*\.[0-9]*\.[0-9]*\): i"
24         result = re.search(pattern,output)
25         if result is not None:
26             done=True
27             return result.group()[1:-4]
28         return "*"
29
30     pattern = "\([0-9]*\.[0-9]*\.[0-9]*\.[0-9]*\) i"
31     result = re.search(pattern,output)
32     if result is not None:
33         return result.group()[1:-3]
34     pattern = "[0-9]*\.[0-9]*\.[0-9]*\.[0-9]* i"
35     result = re.search(pattern,output)
36     if result is not None:
37         return result.group()[0:-2]
38
39
40
41 def calling(hostname,ttl):
42     # get the ip address at current TTL
43     proc = subprocess.Popen("ping -c 1 -W 1 "+hostname+" -t "+str(ttl), shell=True,stdout=
44     subprocess.PIPE)
45     (out, err) = proc.communicate()
46     ip=get_ip(str(out))
47     if(ip=="*"):

```

```

47     return ["*",0]
48     # get time required to reach the destination
49     proc = subprocess.Popen("ping -c 2 -W 1 "+ip+" -t "+str(max_ttl), shell=True,stdout=
subprocess.PIPE)
50     (out, err) = proc.communicate()
51     result = re.search("/[0-9]*\.[0-9]*/",str(out))
52     if not result:
53         return [ip,0]
54     return [ip,float(result.group()[1:-1])]
55
56 # main
57 domain=sys.argv[1]
58 max_hop=30
59 total_hops=0
60 finalarr=[]
61 for i in range(1,max_hop):
62     if done:
63         break
64     total_hops+=1
65     ans=[]
66     # send 3 packets
67     for j in range(3):
68         ans+=[calling(domain,i)]
69     finalarr+=[ans]
70     # format the output
71     print(str(i),end=" ")
72     if (ans[0][0]==ans[1][0] and ans[0][0]==ans[2][0]):
73         if (ans[0][0]=="*"):
74             print(" * * * Request timed out")
75         else:
76             print(ans[0][0] , " " , ans[0][1] , " ms " , ans[1][1] , " ms " , ans[2][1] , " ms ")
77
78
79     else:
80         for i in range(3):
81             if (ans[i][0]=="*"):
82                 print(" * " , end=" ")
83             else:
84                 print(ans[i][0] , " " , ans[i][1] , end=" ms ")
85         print()
86 # Plot the graph
87 garr=[]
88 for i in range(total_hops):
89     garr+=[(finalarr[i][0][1]+finalarr[i][1][1]+finalarr[i][2][1])/3]
90 plt.plot([i for i in range(total_hops)],garr,marker='o' )
91 plt.xlabel('Hops')
92 plt.ylabel('Average RTT (ms)')
93 plt.savefig("trace-"+domain+".png")
94 plt.show()
95 print("Saved RTT vs hops graph at: trace-"+domain+".png")

```

Listing 1: Python implementation for traceroute using ping

#### 4. List of Figures

1	TTL = 18 packet expires before reaching destination . . . . .	1
2	TTL = 19 packet reaches destination . . . . .	2
3	UDP traceroute - many servers did not reply . . . . .	2
4	ICMP traceroute . . . . .	3
5	DNS request response . . . . .	3
6	HTTP requests . . . . .	4



7	Permanently moved response . . . . .	4
8	traceroute command for iitd.ac.in (left) vs custom code (right) . . . . .	5
9	RTT vs Hops graph for iitd.ac.in (servers that did not respond to ping are marked with 0 RTT) . . . . .	5
10	traceroute command for google.com (left) vs custom code (right) . . . . .	5
11	RTT vs Hops graph for google.com (servers that did not respond to ping are marked with 0 RTT) . . . . .	6
12	traceroute command for utah.edu (left) vs custom code (right) . . . . .	6
13	RTT vs Hops graph for utah.edu (servers that did not respond to ping are marked with 0 RTT) . . . . .	6

---

**5. Listings**

---

1	Python implementation for traceroute using ping . . . . .	7
---	---	---