Joe Jensen – jpj35

EECS 325 – P2

Note: To run instructions are contained in comments at the top of distMeasurement.py

Bold Questions Answered:

1. I did not use the technique to measure hop distance with a single probe, I used the traceroute implementation which sends multiple probes, which slowly increments the ttl until the address received is the address we're trying to reach. To implement this program with a single probe, you'd use the icmp message "Destination Unreachable." In doing so, you'd set the port number incorrectly, and when your probe hits the server you're trying to reach, they send back the destination unreachable icmp message, which contains the packet you sent. You can then parse through the data of the icmp message and find the header with the ttl, thus subtracting from where you started and find the number of hops.

2. How to match the ICMP responses with probes you're sending out: The ICMP message you're receiving will contain the data (as mentioned above) from the packet you sent it. (At least the first portion as it only sends back 64 bytes) From there you can parse through that data and see what was inside! So when initially sending the packet, you can set the message inside to something like "joespacket" and parse through the icmp response to see if "joespacket" is there. If it is, it's most likely the icmp you're looking for.

3. Possible reasons you may not get an answer when probing an arbitrary host: Some hosts may not allow probe's similar to traceroute to get through due to security (firewall) . Additionally they may not let icmp messages like the one I would send to go through either. Another reason you may not get a response is because we're using UDP. UDP can drop packets and you'll never know! So you can't just sit and wait for it to come back, it may have been lost.

The next page shows the terminal output of my program. For 10 IP's listed in targets.txt, the hops, rtt, and dist (distance) are found.

```
Finding hops, RTT, and Geo Dist for www.google.com
216.58.219.228
hops: 17
rtt: 30.403137207 ms
dist: 2156.37457548 miles

Finding hops, RTT, and Geo Dist for www.case.edu
129.22.12.21
hops: 4
rtt: 0.207185745239 ms
dist: 0.273172536354 miles

Finding hops, RTT, and Geo Dist for espn.go.com
199.181.133.61
hops: 19
rtt: 71.1941719055 ms
dist: 2060.22944428 miles

Finding hops, RTT, and Geo Dist for www.cnet.com
192.5.110.39
hops: 9
rtt: 0.671863555908 ms
dist: 0.273172536354 miles

Finding hops, RTT, and Geo Dist for www.weather.com
23.213.232.204
hops: 10
rtt: 14.4920349121 ms
dist: 543.85917725 miles

Finding hops, RTT, and Geo Dist for www.java.com
23.203.158.151
hops: 9
rtt: 9.50598716736 ms
dist: 543.85917725 miles

Finding hops, RTT, and Geo Dist for www.walmart.com
23.203.150.40
hops: 8
rtt: 9.64117050171 ms
dist: 543.85917725 miles

Finding hops, RTT, and Geo Dist for www.usc.edu
128.125.253.146
hops: 20
rtt: 77.4409770966 ms
dist: 2060.22944428 miles

Finding hops, RTT, and Geo Dist for www.target.com
192.5.110.39
hops: 10
rtt: 0.607013702393 ms
dist: 0.273172536354 miles

Finding hops, RTT, and Geo Dist for www.china.com
192.5.110.39
hops: 5
rtt: 1.08599662781 ms
dist: 0.273172536354 miles
```

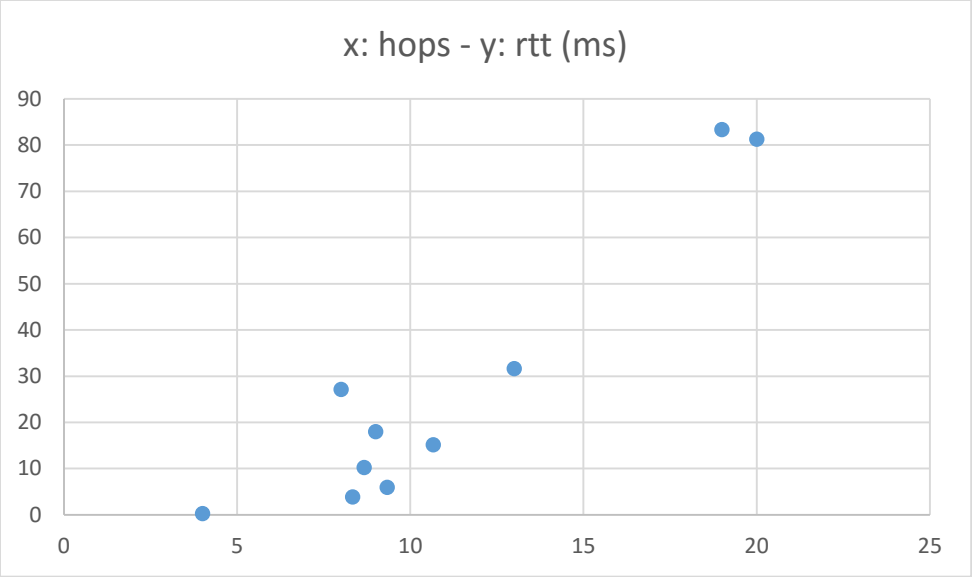
In computing the scatter plots, I ran my program 3 times to get an average for hops and rtt. The below table shows the data:

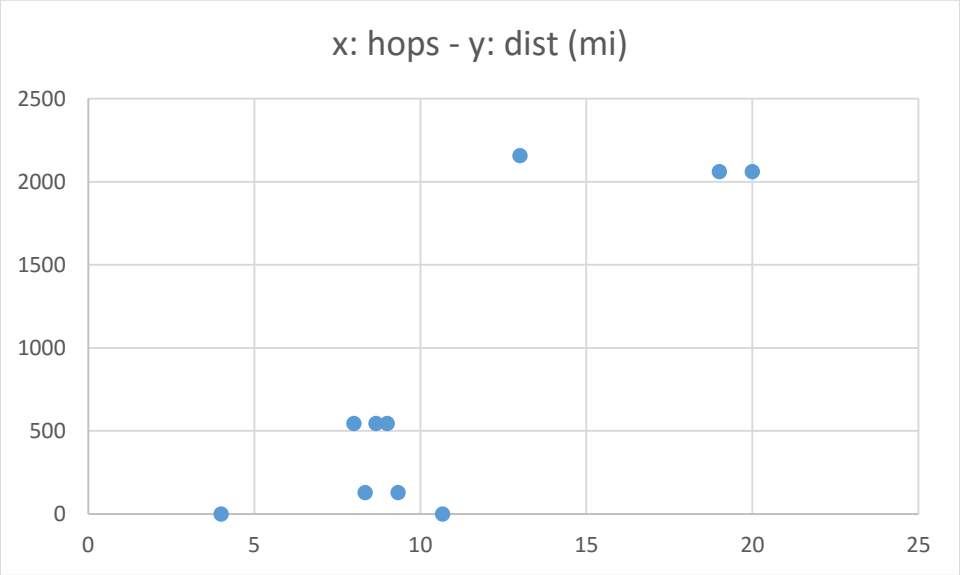
Addr	Hops	RTT (ms)
google	13	31.618
case	4	0.2533
espn	19	83.3477
cnet	10.66667	15.1773
weather	9	18.01397
java	8.666667	10.2377
walmart	8	27.101
usc	20	81.30267
target	8.33333	3.846
china	9.33333	5.934

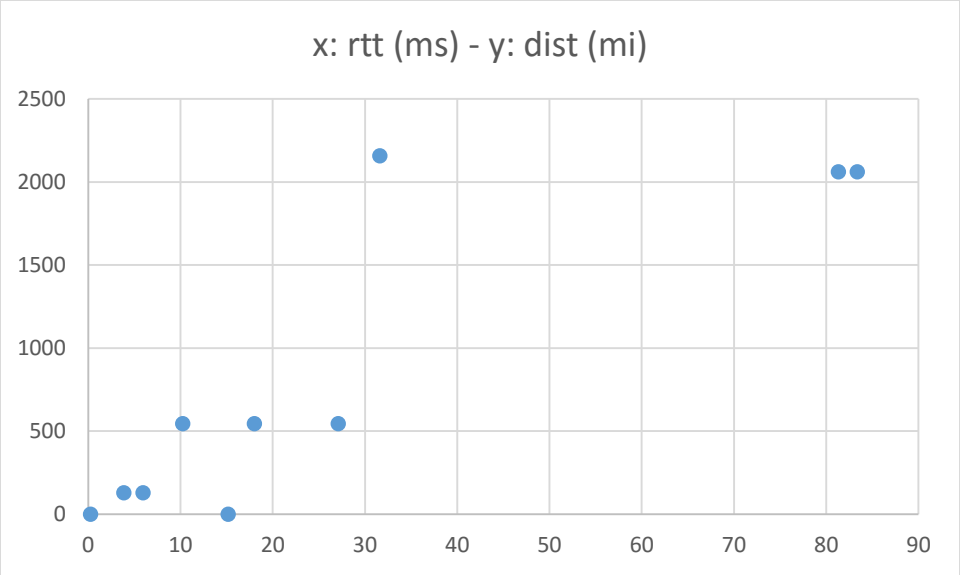
Additionally, I computed (just once) the geographical distance as those values didn't change based on the website "freegeoip.net". The geographical distance data is shown below:

Addr	Distance
google	2156.37
case	0.27317
espn	2060.23
cnet	0.273
weather	543.86
java	543.86
walmart	543.86
usc	2060.23
target	128.93
china	128.935

From this data I computed 3 scatter plots: hops vs rtt, hops vs dist, rtt vs dist (all shown below):


x: hops - y: rtt (ms)
90
80
70
60
50
40
30
20
10
0
0
5
10
15
20
25


x: hops - y: dist (mi)
2500
2000
1500
1000
500
0
0
5
10
15
20
25



These 3 plots produced correlation values as shown below:

Correlation Coeff		
hops-rtt	hops-dist	rtt-dist
0.941431	0.850401	0.856844

All 3 plots have strong correlation, especially hops and rtt. Obviously these should have a strong correlation, especially rtt and distance. The further a router is away, the longer the propagation delay. There is a theoretical limit based on the speed of light, so reaching a router at USC, 2000+ miles away, could never be as fast as reaching case.edu, 0.3 mi away, at today's bandwidths. Distance and rtt have a very strong correlation, also. The further a router is away, the more routers you are likely to travel through to reach it. Routers are all over the place, and if you're going through multiple subnets, companies, and networks, you're going to hit more routers. The strongest correlation I calculated was hops and rtt. A big portion of delay is transmission and queueing. If you encounter more routers as you try to get to a host, the more transmission and queueing you will encounter, which drives rtt up.