EECS 325 Project 2 Report

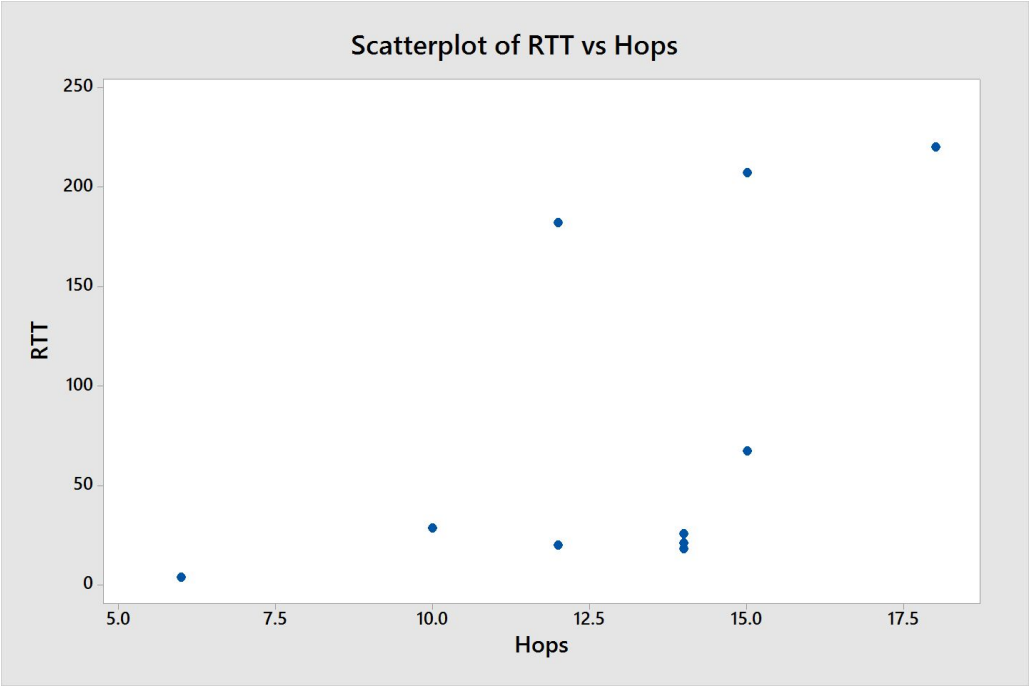
1. Explain how you will match ICMP responses with the probes you sent out

To properly check to make sure the ICMP responses met what probes I sent out there are a bunch of different things I can check. The IP source and Destinations can both be checked for accuracy. The ICMP type and code bytes can also be checked since they should return 3 for our scenario. Lastly, and perhaps most accurately I could check to make sure that my message I sent out was returned. These caused a lot of responses to fail however, making it difficult to gather data. My final submission has this code commented out because for gathering data for the graphs almost nothing was responding and I needed data. Around one in 30 were getting back correctly. Most likely due to network congestion.

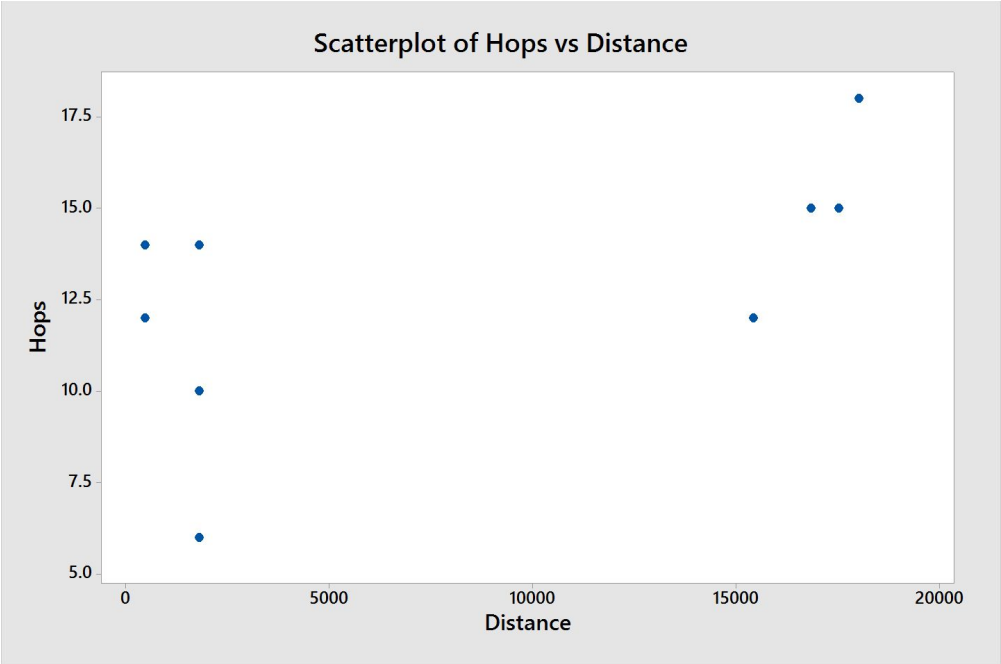
2. Now for reasons the probe could fail:
    a. The destination wouldn't respond in ICMP
    b. Firewall stopped it
    c. UDP packet was lost on the way to destination
    d. ICMP packet was lost on the way back
    e. Destination server crashed
    f. Lost internet connection on either side
    g. Timeout
    h. Timeout after 2 seconds for our tool
    i. Corrupted over physical lines

Site List:
yahoo.com
360.com
aliyun.com
getmyads.com
nikkeibp.co.jp
donga.com
ted.com
google.com
Amazom.com
espn.com

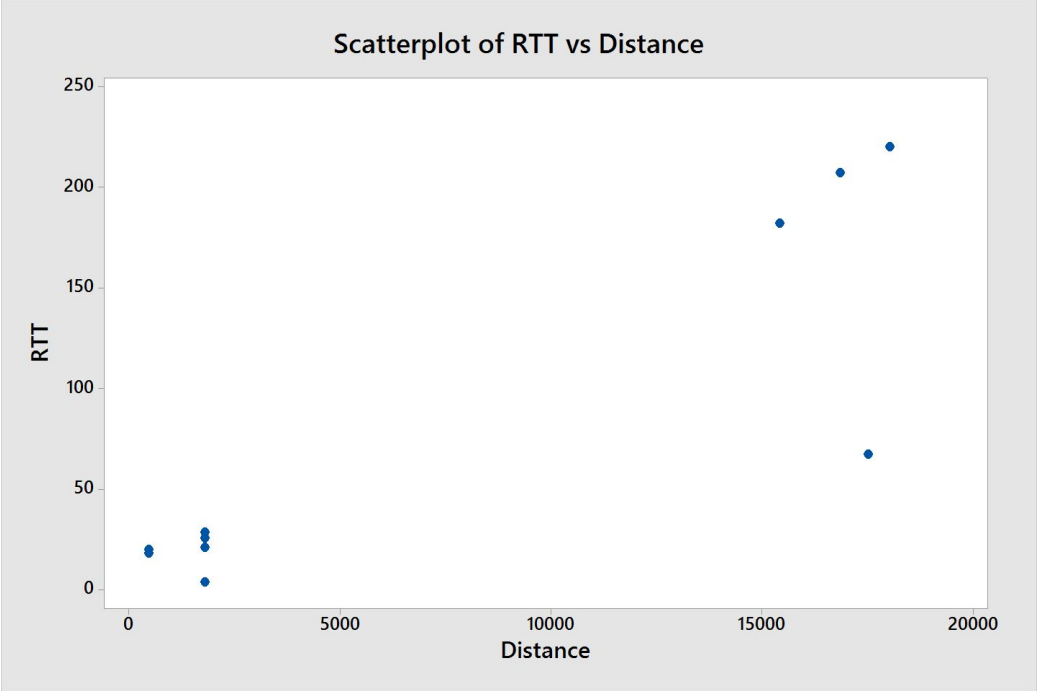
Google ESPN and Amazon were added because a couple of the sites appeared to be down and wouldn't respond at all to anything even my own web browser. They were foreign so it may have been a firewall issue on my end or theirs.



(Pearson) Correlation coefficient was .545



Correlation Coefficient is .542



Coerrelation coefficient is .876

Raw Output from hops and rtt program

Round Trip Time to: yahoo.com (28.5 ms)
Number of Hops to: yahoo.com (10)
Round Trip Time to: 360.com (220.0 ms)
Number of Hops to: 360.com (18)
Round Trip Time to: aliyun.com (67.7 ms)
Number of Hops to: aliyun.com (15)
Round Trip Time to: getmyads.com (20.7 ms)
Number of Hops to: getmyads.com (14)
Round Trip Time to: nikkeibp.co.jp (182.6 ms)
Number of Hops to: nikkeibp.co.jp (12)
Round Trip Time to: donga.com (207.1 ms)
Number of Hops to: donga.com (15)
Round Trip Time to: ted.com (17.9 ms)
Number of Hops to: ted.com (14)
Round Trip Time to: google.com (3.7 ms)
Number of Hops to: google.com (6)
Round Trip Time to: amazon.com (19.8 ms)
Number of Hops to: amazon.com (12)

Raw Output from geographic distance program:

Geographic Distance from: yahoo.com (1802 km)

Geographic Distance from: 360.com (18013 km)

Geographic Distance from: aliyun.com (13360 km)

Geographic Distance from: getmyads.com (1802 km)

Geographic Distance from: nikkeibp.co.jp (10674 km)

Geographic Distance from: donga.com (16834 km)

Geographic Distance from: ted.com (462 km)

Geographic Distance from: google.com (1802 km)

Geographic Distance from: amazon.com (462 km)

Host	RTT(MS)	Hops	Distance
yahoo.com	28.5	10	1802
360.com	220	18	18013
aliyun.com	67	15	13360
getmyads.com	20.7	14	1802
nikkeibp.co.jp	182	12	10674
donga.com	207	15	16834
ted.com	17.9	14	462
google.com	3.7	6	1802
amazon.com	19.8	12	462
espn.com	15.6	14	1802

Conclusion:

Something was definitely off when I was doing my final testing. For some reason many of the response times were extremely sporadic. It was hard to find good data all the time because many of the numbers would jump around rapidly. This may have been due to network

congestion but it is hard to say for sure. Hopefully on another day the times will be more consistent.

The graphs are kind of sporadic, they all have a ton of outliers, so there is some correlation evident, especially on RTT and Distance, but in reality I would expect a much higher correlation. The data wasn't like backwards though or anything, there are just a lot of outlying points that hurt the correlations.

Additionally the geographic data is repetitive, perhaps some of the servers are based out of the same cities because for instance distance to the .jp website makes perfect sense. Same with 360.com which is chinese, and donga.com

I lost a lot of probes, more than I was expecting, but I guess I'm not used to trying to only send a single probe and get a response, I bet the data would be a lot more consistent if a lot of them were sent to each destination instead of just one. Tracert uses more than just one probe to get better data, and I now understand why.