CSDS 325 Project 5

Jhean Toledo, jct95

December 2022

Theme

The theme surrounding my observations of the network is the idea of route stability. I wanted to see just how stable or in-stable a certain route is when sending information across the network, measuring things such as whether the same route was used multiple times, or if the network changed routes due to faster RTTs or other various reasons. To test this, I created a simple python script that ran 10,000 trace routes to 10 different domains. The domains I decided to pick is as follows: utexas.edu, google.com, icir.org, nyu.edu, utoronto.ca, gov.za, 46.188.47.18, 82.71.8.205, 212.102.51.18, 43.229.60.176, and 210.5.56.145. For convience, I will refer to the IPs: 46.188.47.18, 82.71.8.205, 212.102.51.18, 43.229.60.176, 210.5.56.145, as Moscow, London, Tokyo, Sydney, and Henan respectively.

Procedure and Data Gathering

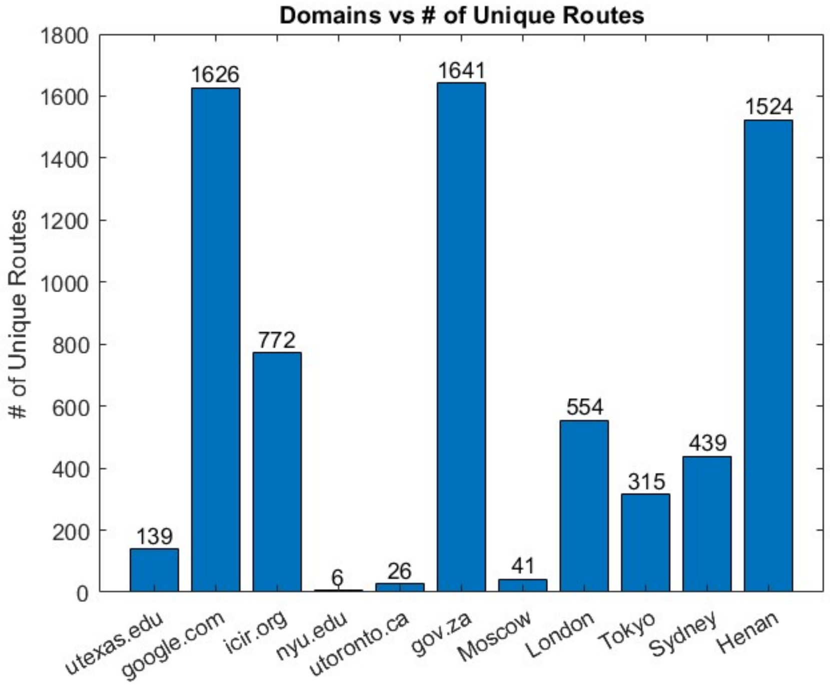
The exact trace route command I used was "traceroute -n -m 45 -A [domain name]." Running this command showed me only the IPs of each router rather than their full hostname, as well as set the max TTL as 45 hops and even displayed the AS number (if the router opted to do so). I ran this command in my python file called traceroutes.py for each of my 11 domains roughly 1600 times. Giving me 17600 different traces. To parse through all of those traces, I created a second python file called tracereader.py where I parsed through the data and analyzed the data in various ways. In order to plot the data in a meaningful way, I used the outputs from tracereader.py and loaded them into a matlab file called project5-plots.m to create plots illustrating various points.

Section I: Stability in terms of IPs:

To test for stability in terms of IPs, I simply analyzed the data created by the traceroutes and ran another script to parse through and find only unique traces in terms of the IP addresses for the routers. The code I used for this is in tracereader.py

1.1: Route variance

One thing worth noting was that some domains had a larger sense of variance when it came to using the exact same trace more than once. For example, from the data I have gathered, running a trace route to google.com seemed to have 1626 unique routes used where as running trace routes to utexas.edu only has 139 routes used. This can be seen more clearly in the graph below.



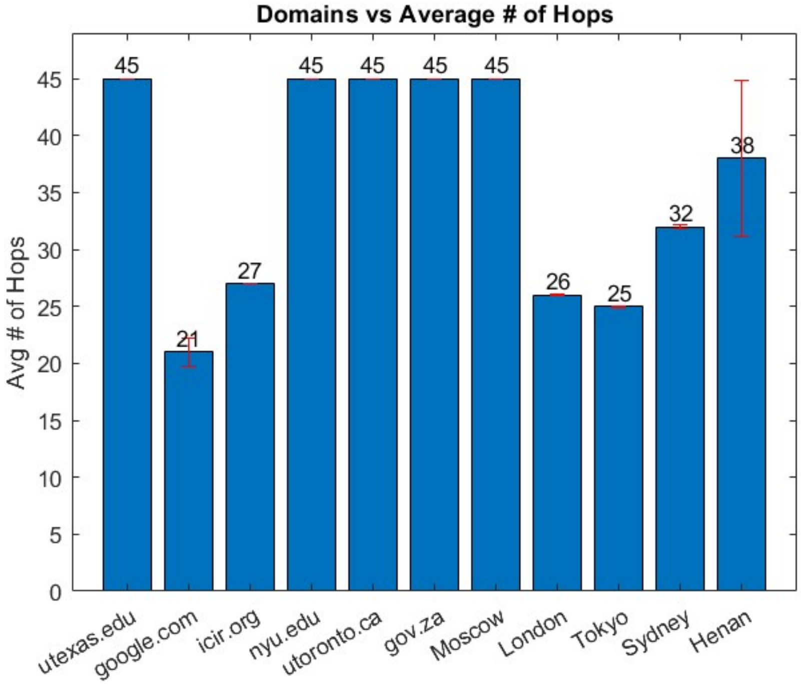
From this data, I can also get a sense of what causes some domains to have a larger number of unique routes as compared to others. Looking at the difference between google.com and utexas.edu, we can see that google.com used a far more different routes than utexas.edu. My guess was that this is because utexas.edu is a lot more of a stable domain than google.com is. Meaning that the route always goes into the same general direction for utexas.edu as compared to google.com. To prove this, I added more stable domains such as icir.org, nyu.edu, and utoronto.ca. To my surprise, the more stable the domain was, the lower the amount of unique routes was seen. To explain the large number of unique routes going to icir.org, I would guess this has to do with the distance between me and California, as there is a larger chance for the route itself to change due to the large distance.

However, I believe it is fair to say with this data I can conclude that non-static domains have a lot more unique routes as compared to a static domain. The other non-static domain I included in my testing was gov.za, a government website in South Africa. I knew that it wasn't static as the destination IP changed quite frequently, which made me want to compare it to the other static domains I have listed. To ensure that the domains I picked would be static, I picked DNS servers for my international traces.

We can see that for the international static domains that the number of unique routes is surprisingly low. This would further prove my point about static vs non-static domains.

1.2: Hop variance

I also wanted to see the average hop count of a given trace for each domain. Below is a bar graph with the average hop count with standard deviations for each domain.



Looking at the graph above, we can see the average hop count going to a given domain, as well as its standard deviation. Something worth noting is that icir.org has a extraordinarily small standard deviation, meaning that it almost always took 27 hops to get to icir.org. Where as going to Henan had a much larger standard deviation, meaning that it had a larger variance in terms of hop count. This would make sense as Henan is located all the way in China, which is nearly 3 times the distance as icir.org, located in Berkeley California.

It would also be fair to say that a route doesn't change in terms of hop size as often as I thought it would. From the last section discussing variance in IPs, I figured that the route would change not only in the routers a given trace is going through, but also the amount of routers. Yet based on the data in the graph, we can see that the standard deviation for each domain is hardly noticeable except for Henan. This would mean that the traces had nearly the same hop count each time. So even though the IPs used in the route changed, the actual amount of routers to get to the destination did not. One potential reason for this is that the route is actually going through the same organizations, for example the same AS's, but just going through different routers in each AS. This will be elaborated on further in the next section.

Section II: AS Stability:

To analyze the Autonomous Systems that each route goes through, I again use tracereader.py to read out only the AS numbers for a given route and stored them in a text file.

2.1: AS In-variance

I wanted to see if the routes may change quite frequently for the IPs they go through, but if they went through the same group of IPs everytime. To do this, I simply took note of the AS numbers that the trace route gave back and associated them to each domain. To my surprise, nearly every single time that a route was ran to the same domain, the same list of AS numbers would come back. Below is a table with the AS numbers received after running to each of the 5 domains located in North America.

AS Numbers for each domain in North America				
utexas.edu	google.com	icir.org	nyu.edu	utoronto.ca
AS32666	AS32666	AS32666	AS32666	AS32666
AS19009	AS19009	AS19009	AS19009	AS19009
AS600	AS600	AS3112	AS3112	AS3112
AS11164	AS11164	AS600	AS600	AS600
N/A	AS15169	AS11537	AS11537	AS11537
N/A	N/A	AS2152	AS3754	AS6509
N/A	N/A	AS198949	AS19905	AS26677
N/A	N/A	AS25	N/A	AS549
N/A	N/A	N/a	N/A	N/A

Table 1: Entries listed with N/A signify that the route was shorter than other domains so there was simply no AS number for the remaining hops. It is important to note that only routes where routers opted to provide the AS number are included here.

Looking at the table listed above, we even see that icir.org, nyu.edu, and utoronto.ca share the first 5 AS numbers. Having similar AS numbers going to nyu.edu and utoronto.ca would make sense as they are in close proximity to each other, so it would be logical that the routes going to either domain end up going through similar AS'.

All domains appear to at least go through AS32666, AS19009, AS600, and AS11164 eventually. This would imply that these AS numbers are responsible for the North East, as my request always stemmed from Ohio. I wanted to see if the same AS numbers would be used if I sent a trace route to another country. Below is another table but with international domains located outside of North America.

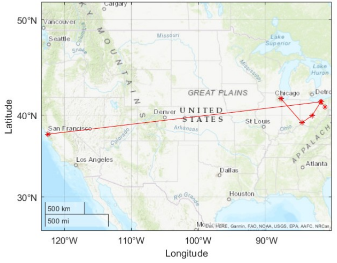
AS Numbers for each international domain					
gov.za	Moscow	London	Tokyo	Sydney	Henan
AS32666	AS32666	AS32666	AS32666	AS	AS32666
AS19009	AS19009	AS19009	AS19009	AS19009	AS19009
AS600	AS600	AS600	AS600	AS3112	AS3112
AS7018	AS7018	AS7018	AS11164	AS600	AS600
AS174	AS3257	AS1299	AS46887	AS11537	AS11164
AS3741	AS28917	AS13037	AS1784	AS3754	AS4637
AS6421	AS39153	N/A	AS3491	AS19905	AS9225
AS37130	N/A	N/A	AS60068	N/A	AS4809
N/A	N/A	N/A	AS212238	N/A	N/A

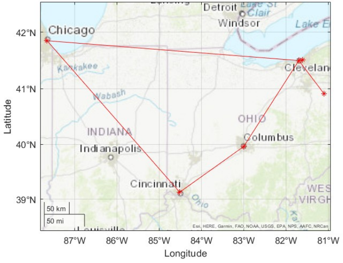
Table 2: Entries listed with N/A signify that the route was shorter than other domains so there was simply no AS number for the remaining hops. It is important to note that only routes where routers opted to provide the AS number are included here.

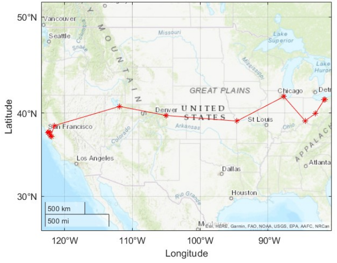
From the table, we again see a similar pattern with the first 4 or so AS numbers, leaving me to believe that yet again those AS numbers are responsible for the northern region of North America. One other interesting thing to note is that the last North American AS number each route went through was AS600. I was unable to get a location response to any of the IPs registered under AS600 but looking up what organization owns AS600 reveals that it belongs to

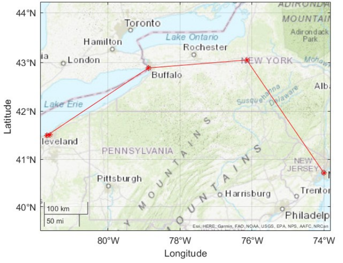
2.2: Phyiscal route consistency

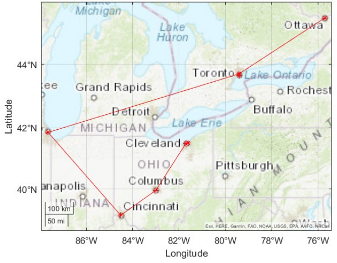
From looking at the AS numbers, I wanted to see visually where the traces were going and figured that the physical route must not change that often if the AS numbers are consistent. Below are visual representations of the trace routes to each of their respective domains.

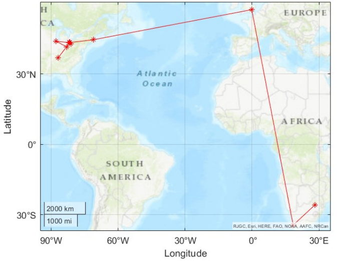

Path used to get to utexas.edu

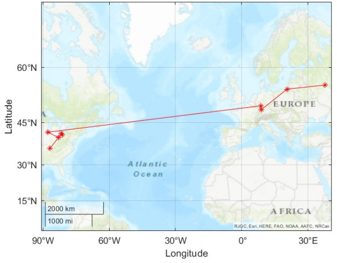

Path used to get to google.com

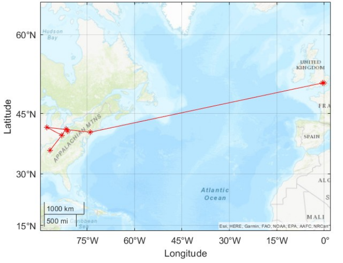

Path used to get to icir.org

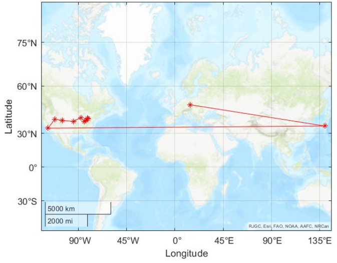

Path used to get to nyu.edu

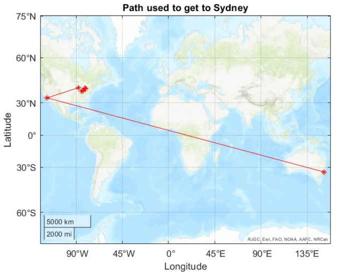

Path used to get to utoronto.ca

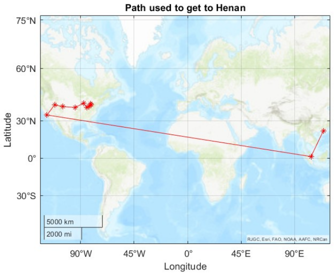

Path used to get to gov.za


Path used to get to Moscow


Path used to get to London


Path used to get to Tokyo


Path used to get to Sydney

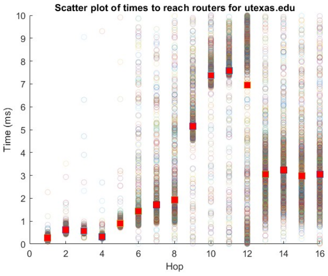


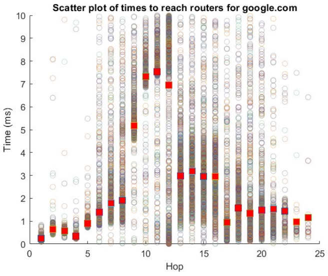
I used a python library created by ipinfo to find the geolocation of the IPs. The library and their website can be found here: `https://ipinfo.io/developers/libraries`

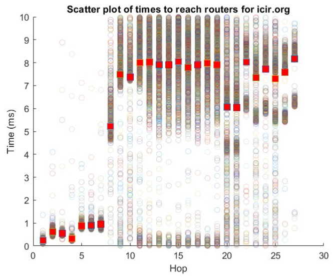
From the visual graphs, we can see that the route is indeed consistent and going towards its destination. Treating Tokyo has an outlier, we can also see approximately where each of the different AS numbers are represented by the points. Granted some AS did not opt to give away their location, but it still gives us a rough idea of where the traffic is going through. To my surprise, it would seem that the route used for each domain was roughly the shortest path from Ohio to any of these locations. I would have assumed that the network is independent of distance, meaning that the network may not route traffic in terms of shortest distance, but rather shortest time to get from one router to another.

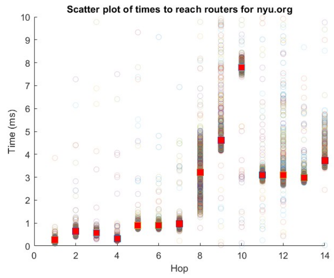
Section III: Latency

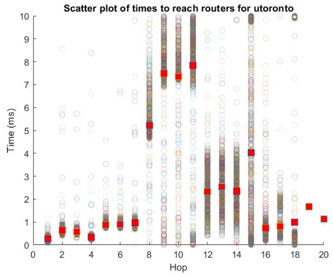
One of the things I wanted to investigate was the time it took for one packet to get to one destination and back. Below are graphs listing the RTTs for each sample plotted over the hop count, with a read square to denote the average RTT for that particular hop.

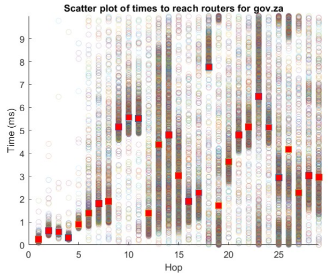


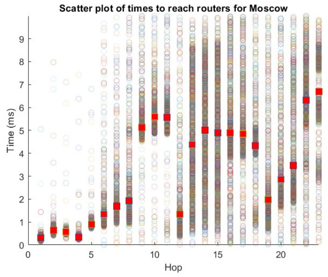


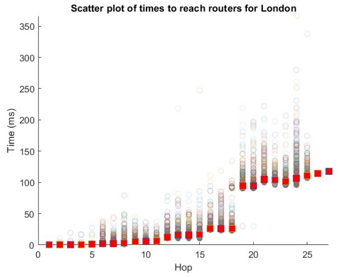

Scatter plot of times to reach routers for icir.org

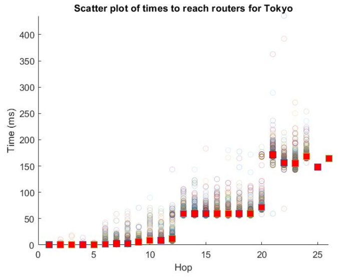

Scatter plot of times to reach routers for nyu.org

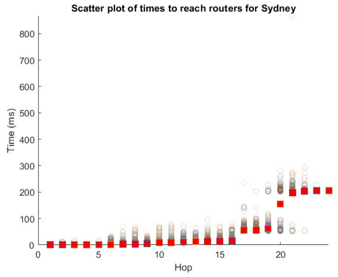

Scatter plot of times to reach routers for utoronto

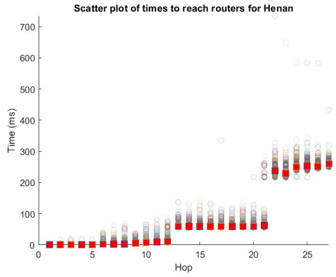

Scatter plot of times to reach routers for gov.za


Scatter plot of times to reach routers for Moscow


Scatter plot of times to reach routers for London


Scatter plot of times to reach routers for Tokyo


Scatter plot of times to reach routers for Sydney


Scatter plot of times to reach routers for Henan

From the graphs above, we can make out some trends. For the domains located in North America, the general trend seems to have routers in the middle with the longest RTTs where as the endpoints are rather fast, with the exception of the routes going to icir.org. Again, I would speculate that this is due to the distance between the endpoints, as the source and destination is not that far between me, nyu.edu, utoronto.ca, and utexas.edu.

We can also tell from the data that the farther a domain is, the longer the response time is. For example, looking at the international domains, we can see that they all exceed roughly 150 ms for the destination to give a response back to me. To no surprise, the domains located in North America have a much more reasonable response times with the average for all North American domains being under 10 ms.

With this, we can conclude that distance does in fact contribute to RTTs.