CSDS-325 Computer Networks, Project 5 report

Nicolas Slavonia, njs140

November 2022

Introduction

The main theme I want to explore in my project is different forms of delays in the network. I want to understand the time it takes to make contact, ping, retrieve data and more to web servers around the world. I think measuring delays is important because although the network serves multiple purposes, a large reason it was created was to be able to communicate to each other quickly. Analyzing delays can first give us an impression of how fast the network is, but I'd also like to analyze the network during different times of the day to monitor the change in delays and get a small understanding of the limits of the network. In future work, actions can be made on the data retrieved to help minimize some delays.

Organization

I attached two files of code. The 'proj5.c' file is the code I used to get my data from the class servers. It has all my chunks of code in there. The file itself will not run. I would copy and paste my desired method in vi on a class server and let it run. I do similar things with the MatLab code. I comment both files saying which chunk does what. The data can be accessed at: https://drive.google.com/drive/folders/1iwDyLHAs1X9arvLyCkD0kPEx9kvfRCi7. The 'ping' data are my ping measurements. The first column is the time, then each column is for a website. Each row are the ping measurements for each website. The 'curl' data is all the data pertaining to my curl measurements. The first column is the time, second is the DNS measurements, third column is the TCP connection, fourth is the first Byte measurement and the fifth is total time. Each row corresponds to a website at a certain time. First row is case.edu, second is stanford and so on. Once I finish waiting 600 seconds, I take the measurements again. Lastly, the 'wget.c' file holds the times it took get a web object from that website. I explain more regarding this data below but I wanted to explain here how I organized it.

Procedure

I analyze 5 different measurements using *ping*, *curl* and *traceroutes*.

1. The first thing I look at are *ping* measurements to multiple websites around the world over the course of 2 days. I do this to understand how congestion over the network changes the ping measurements with the assumption congestion increases during the day time. So I am looking for fluctuations in the ping measurements. This is my first form of analysis because I refer to this often.

2. The second thing I analyze is Case Western Reserve University's proxy server. My goal here is to understand how CWRU caches recent http requests. With this knowledge, I then measure DNS searches over 2 days using *curl*.

3. The third thing I analyze is the time it takes to set up a TCP connection to websites around the world using *curl*. I compare these times to the ping measurements. I also look at the first byte sent times.

4. Similarly, I monitor the total time *curl* requires to establish a connection. This includes the DNS search, TCP connection time and first byte sent time. I do this for each website over a period of 2 days and compare these to the ping measurements.

5. I then set up wget to fetch similarly sized data from each website again over 2 days. I compare these times to the ping measurements as well as my other curl measurements.

Gathering Data

I gathered all the data myself. I wrote different **C** scripts that would run desired measurements at certain times on the class servers. For the *ping* measurements, I ran a simple while loop that would run for 2 days. I'd have a list of different IPs I wanted to ping, so every 5 minutes I would run a for loop that sends 30 pings over the course of 30 seconds (for each IP) and average the measurements. I would then write the measurements to a file and analyze the data on MatLab.

The *curl* measurements used a similar script to the ping measurements (very similar). In fact, I pretty much copied my ping script and swapped out the 'ping' with 'curl'. I would have the curl command only return the *time_namelookup*, *time_connect*, *time_starttransfer* and the *time_total* which represent the DNS lookup time, TCP connection time, first data byte sent time and the total setup time.
I also ran a script before the one above that used binary search tactics to find how long CWRU caches web searches. I used the DNS lookup time returned by curl.

Analyzing Data

All my analysis was done on MatLab. Whether that was plotting figures, analyzing mean, median, standard deviation, etc. all was done on MatLab. I was mainly looking at the plots and looking at the different time measurements. My data collection was done in such a way that the data was neatly organized in the files I wrote to. So that way on MatLab, it was pretty much just running a plot command.

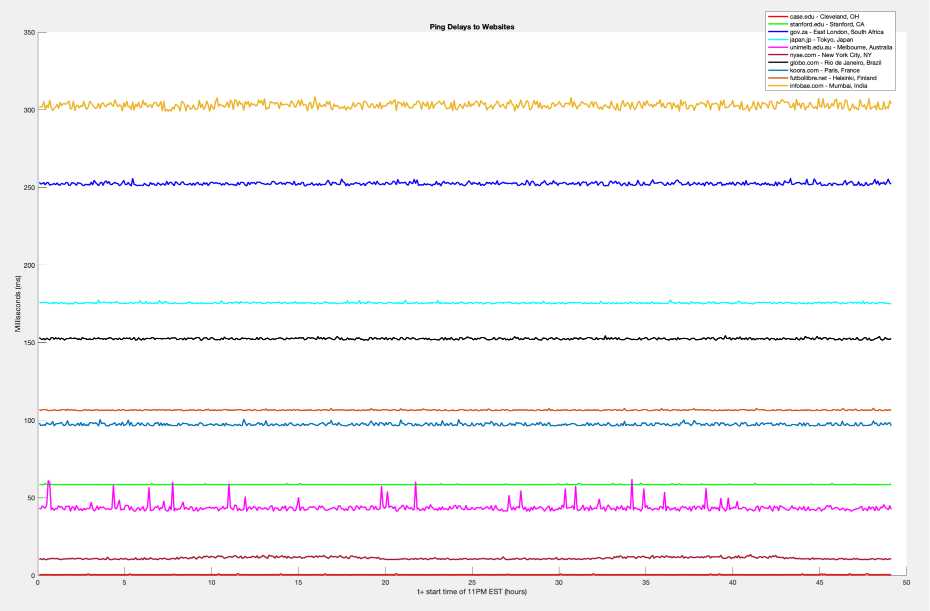
Results

Ping Measurements - Bonus

I wanted to analysis how at different times of the day, the ping measurements may change. I decided to ping 10 websites around the world for 48 hours. For reference, here are the websites, IPs and locations I used.

Website	case.edu	stanford.edu	gov.za	japan.jp	futbollibre.net
IP	129.22.12.21	171.67.215.200	163.195.1.225	157.7.107.95	95.215.19.22
Location	Cleveland, OH	Stanford, CA	East London, South Africa	Tokyo, Japan	Helsinki, Finland
Website	unimelb.edu.au	nyse.com	globo.com	koora.com	infobae.com
IP	43.245.43.59	104.16.103.50	186.192.90.12	37.187.152.154	184.27.197.106
Location	Melbourne, Australia	New York City, NY	Rio de Janeiro, Brazil	Paris, France	Mumbai, India

I use these 10 websites for a large portion of my project. Over the course of 48 hours, I pinged each website 30 times every 5 minutes. When I first started collecting data, I noticed that occasionally I would lose a ping or two during these 30 pings. I would also notice that sometimes one ping took forever. For this portion of my analysis, I wasn't too concerned over these little flaws. So I ended up just averaging the 30 ping measurement times and outputted that number to a file. The data I am referring to is called 'ping.txt'. MatLab has a command called 'readmatrix()' which turns a txt file of numbers into a giant matrix. So using that, I was able to analyze all my data. Here are my ping measurements over the course of 48 hours starting on November 29th at 11PM EST.



Here is a table of the averages and variances of the data. Again, I used MatLab to do this.

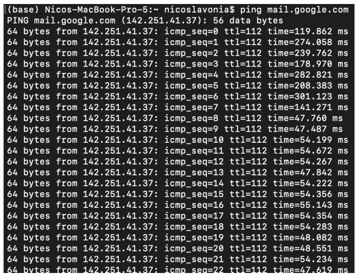
Website	case.edu	stanford.edu	gov.za	japan.jp	futbollibre.net
Mean	0.5382	58.54	252.38	175.54	43.69
Variance	0.0098	0.0159	0.8559	0.1905	7.4088
Website	unimelb.edu.au	nyse.com	globo.com	koora.com	infobae.com
Mean	11.16	152.48	97.31	106.50	302.86
Variance	0.5279	0.3047	0.6856	0.0786	3.77

Let's look at the data. There are a couple of interesting things but first of all I'd like to say that I am a little disappointed in the results. I was personally hoping for large changes in delay during day time vs night time, but that wasn't the case. That is something to note though because this shows that daily volumes of traffic may not affect delays as much as I'd thought.

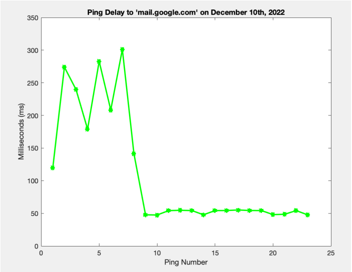
However, we do get a little of bit of this when looking at the New York Stock Exchange. This makes sense because the NYSE is only open from 9:30AM to 4:00PM EST so it would make sense that there are lower delays at night. In the future, I really should have chosen websites that received higher volumes of traffic. Some other interesting things to note are the high variances for Australia. Another interesting thing is that pinging Australia takes very little time. I am pinging Melbourne University and MaxMind is telling me the IP address I pinged is in Australia but I do find it weird that the pings are so low. There are some pretty large occasional variations such as the spikes when pinging Australia and the large fluctuations when pinging India and South Africa. Besides

Australia (which I am suspicious about), it makes sense that the further we are away, the larger the pings. I think it is also interesting that the variance for Australia was so large. Something weird is happening with that ping.

Bonus: All this data was collected and written up before December 10th. However, around 10:20 AM EST I got a text from CWRU telling me that google was having issues delivering mail to inboxes. I quickly thought to ping *mail.google.com* to see the delays and this is what I saw.



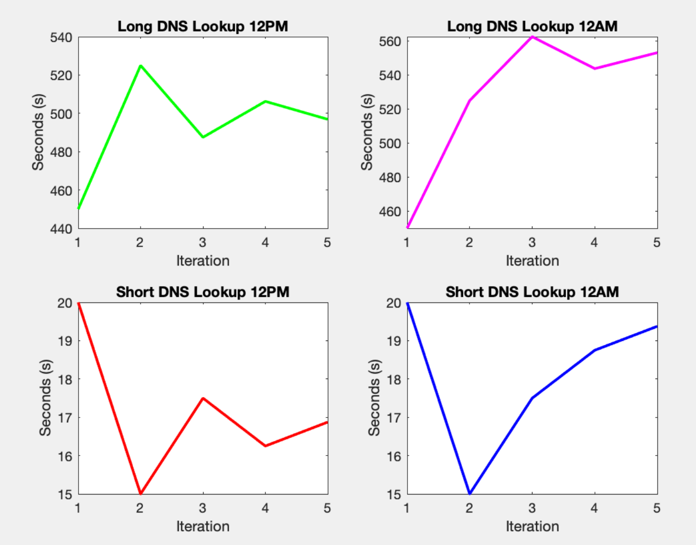
```
[(base) Nicos-MacBook-Pro-5:~ nicoslavonia$ ping mail.google.com
PING mail.google.com (142.251.41.37): 56 data bytes
64 bytes from 142.251.41.37: icmp_seq=0 ttl=112 time=119.862 ms
64 bytes from 142.251.41.37: icmp_seq=1 ttl=112 time=274.058 ms
64 bytes from 142.251.41.37: icmp_seq=2 ttl=112 time=239.762 ms
64 bytes from 142.251.41.37: icmp_seq=3 ttl=112 time=178.970 ms
64 bytes from 142.251.41.37: icmp_seq=4 ttl=112 time=282.821 ms
64 bytes from 142.251.41.37: icmp_seq=5 ttl=112 time=208.383 ms
64 bytes from 142.251.41.37: icmp_seq=6 ttl=112 time=301.123 ms
64 bytes from 142.251.41.37: icmp_seq=7 ttl=112 time=141.271 ms
64 bytes from 142.251.41.37: icmp_seq=8 ttl=112 time=47.760 ms
64 bytes from 142.251.41.37: icmp_seq=9 ttl=112 time=47.487 ms
64 bytes from 142.251.41.37: icmp_seq=10 ttl=112 time=54.199 ms
64 bytes from 142.251.41.37: icmp_seq=11 ttl=112 time=54.672 ms
64 bytes from 142.251.41.37: icmp_seq=12 ttl=112 time=54.267 ms
64 bytes from 142.251.41.37: icmp_seq=13 ttl=112 time=47.842 ms
64 bytes from 142.251.41.37: icmp_seq=14 ttl=112 time=54.222 ms
64 bytes from 142.251.41.37: icmp_seq=15 ttl=112 time=54.356 ms
64 bytes from 142.251.41.37: icmp_seq=16 ttl=112 time=55.143 ms
64 bytes from 142.251.41.37: icmp_seq=17 ttl=112 time=54.354 ms
64 bytes from 142.251.41.37: icmp_seq=18 ttl=112 time=54.283 ms
64 bytes from 142.251.41.37: icmp_seq=19 ttl=112 time=48.082 ms
64 bytes from 142.251.41.37: icmp_seq=20 ttl=112 time=48.551 ms
64 bytes from 142.251.41.37: icmp_seq=21 ttl=112 time=54.234 ms
64 bytes from 142.251.41.37: icmp_seq=22 ttl=112 time=47.619 ms
```



The pings took forever! Now this does not relate to any of my other data because this was not taken on the class server and these are individual ping measurements, not an average. But I wanted to add this because I thought it was cool. The initial pings ranged from 100 to 300 milliseconds. They quickly dropped off as it seems they fixed their issue. For reference, when I run ping now under the same conditions, I am getting an average of 40 milliseconds. I thought this was cool and wanted to add it.

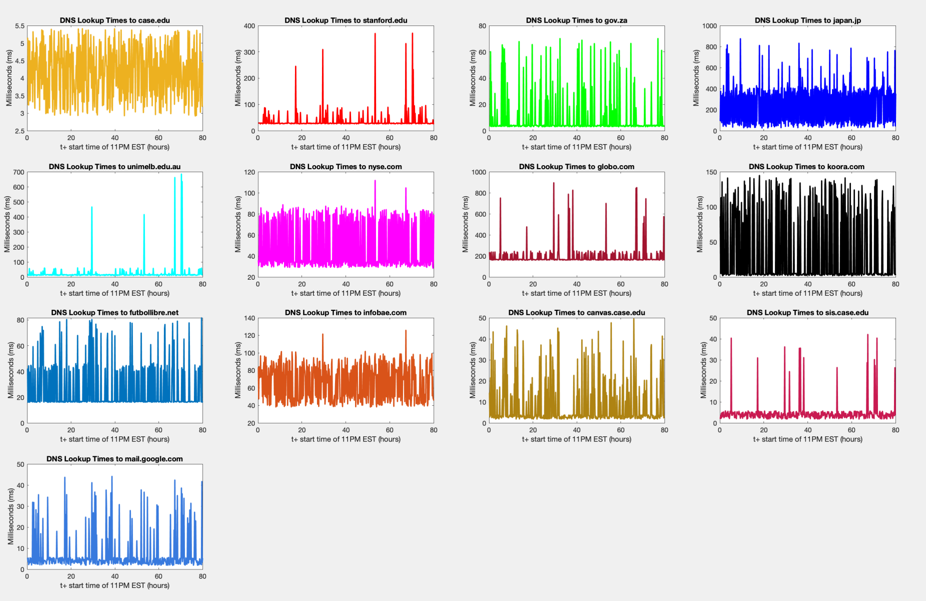
CWRU Cache, DNS Lookup

For this section, I wanted to analysis the DNS lookup times over a period of 2 days. The goal here was to see if the DNS servers produced delays during the day. Because our nearest DNS server is located in the United States, it would be interesting to see if it takes longer to look something up during the day vs at night. First however, I need to make sure that my searches were not being cached at CWRU's proxy server. So the first thing I did was write a script that uses binary search to close in on the period of time CWRU caches a website. Now this doesn't produce a lot of data, so here is the graph.



I got this data by looking up 'japan.jp' hoping no one would look this website up on CWRU Wireless while this was running (luckily no one did). Here is a quick explanation of what is happening. When messing around with the DNS lookup times, I noticed I would get 3 different ranges of times. The first was a couple milliseconds, the second would be 20 or 30 milliseconds, and the third would be a couple hundred milliseconds. So on the graph above, the 'Short' DNS lookup is the time the cached website would go from a few milliseconds to 20 milliseconds. Looking at the data, around lunch time, CWRU would hold onto the exact website for around 17 seconds. Where as at midnight, it would be held for around 19 seconds. The 'Long' DNS lookup is the time it took to go from 20 milliseconds to a couple hundred. Around noon, it would hold onto my website for around 500 seconds where as at midnight, it would hold onto it for around 550 seconds. So this is telling me that CWRU holds onto the information about a website longer at night.

What I really wanted was to find a specific time such that my searches would always require the long DNS lookup time. To be safe, I picked a time of 600 seconds, 10 minutes. With this time, I would be able to lookup the long DNS time for each of my websites over the course of 2 days. Lastly, I also added some new websites. I added websites that CWRU students use a lot such as ’sis.case.edu’, ’canvas.case.edu’ and ’mail.google.com’. I added these out of my own curiosity because if the DNS lookup time was small, that means it was recently used.

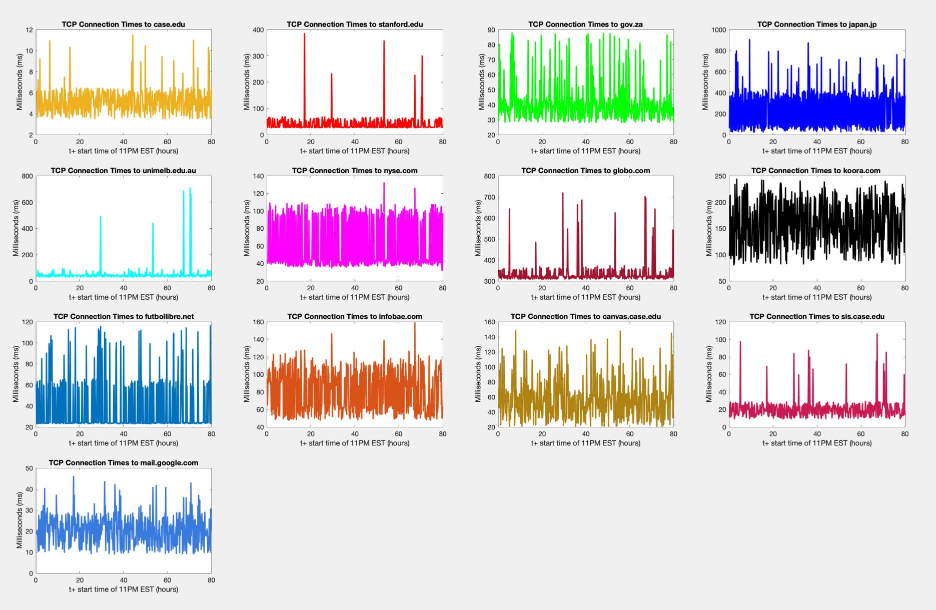


Now I must admit this data doesn’t really make sense to me. I was able to tell that the CWRU proxy server ’forgets’ websites after around 9 minutes so I was a bit shocked when I saw these results. These results show that not only was I apparently wrong, but some websites are stored longer than others. For example, japan.jp was nearly never fully stored (always needed the long DNS search) where as ’gov.za’ was. Sometimes ’unimelb.edu.au’ took nearly a second to look up where as most of the time it was the short lookup time. ’globo.com’ was similar to ’japan.jp’ but occasionally took a long time. Canvas, SIS and gmail seemed to have always been saved on the proxy server, or people were using it (can’t really tell). ’case.edu’ was always super short which just means ’case.edu’ is hosted on campus.

There wasn’t the strongest correlation between ping and DNS lookup times. DNS lookup times were much quicker than ping. Intuitively, this makes sense as ping talks to the actual server where as the DNS lookup time is talking to a nearby server. It is interesting however that ’japan.jp’, ’globo.com’ and ’nyse.com’ took the same amout of time if not longer.

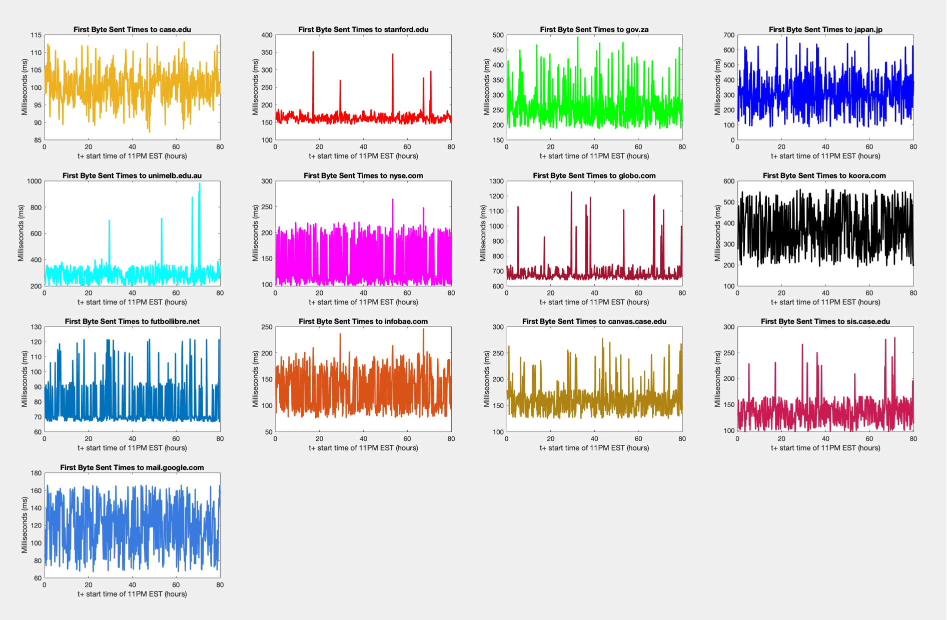
TCP Setup

Another useful piece of information the *curl* command gives is the time it took to set up a TCP connection. With ping and DNS lookups, all I do is send a request and wait for a response. This measurement tells us how long it took to talk to each other since setting up a TCP connection over HTTPS requires multiple packets. So this can tell us the delays received over multiple requests and responses. In addition to this, I will also look at the time it takes to send the first byte of data across the network using curl. Starting with the TCP connection times, here are the plots I got.



This shows us the time it took to setup TCP connections. Looking at the data, there is a pretty strong correlating between ping measurements and TCP connections. What I should mention is that 'case.edu', 'nyse.com' and 'globo.com' needed more time to setup a connection while 'gov.za', 'unimelb.edu.au', 'futbollibre.net' and 'inforbae.com' needed less time. The rest used roughly the same amount of time. This is interesting, I was expecting every website to require more time to setup a connection.

I also measured the time it took to send over the first byte of data.

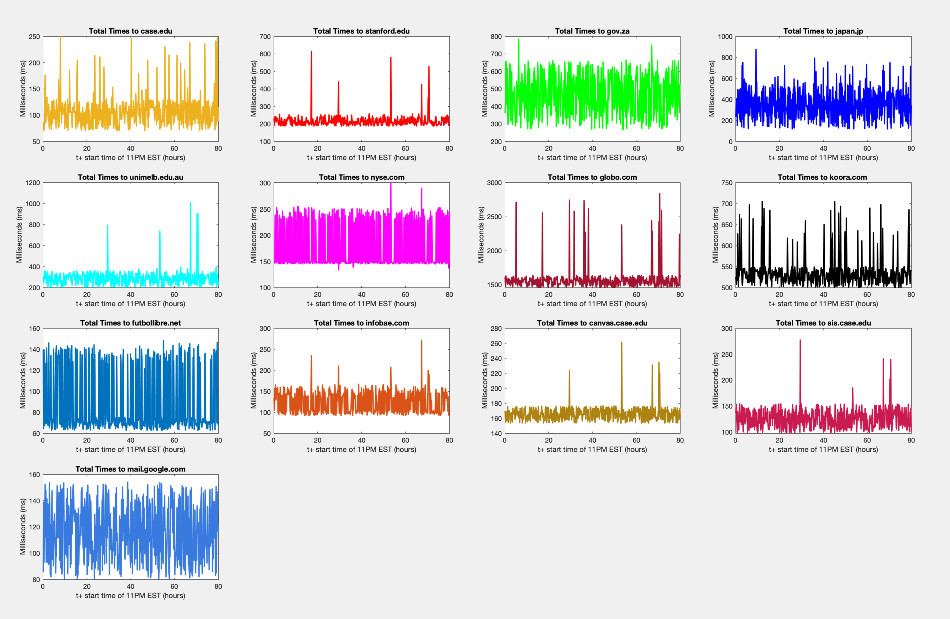


As expected, every website needed more time to accomplish this. 'case.edu' needed a lot more time which was interesting. The TCP connection and first Byte graphs each have the same shape but they are offset in time. This tells me that individual measurements don't differ from each other, but rather they change over time.

For some reason 'futbollibre.net' had the fasted time to send the first byte of data. Even faster than 'case.edu'. Not sure why this true but it was something I thought I should mention.

Total Time

The last thing I do with curl is look at the total time which is the time required to complete the whole process. I am interested in seeing this because this gives the summary of how long it took to cover all necessary setups like the DNS lookup, TCP connection, pre-transfer delays, re-direct delays and more.

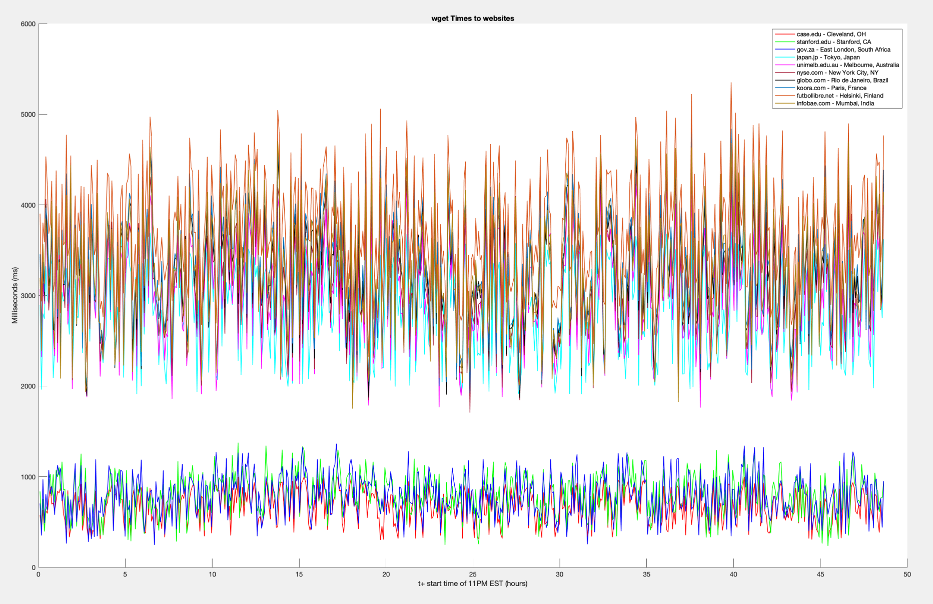


This graph summarizes a lot of the previous information ive said. But I wanted to point out a few more things. For one, some of the graphs have patterns to them. Looking at 'japan.jp', the times were fluctuating. One measurement would be quick, the next would be slow, and then quick again. Some graphs like 'stanford.edu', 'globo.com', 'sis.case.edu' and more were pretty steady the whole way through. Expect for a couple very large spikes in delay. Some graphs had absolute minimums like 'nyse.com', 'inforbae.com' and 'infobae.com'. The measurements wouldn't get any quicker, but they would occasionally get much slower.

Some websites stayed true to their ping measurements while others didn't. 'japan.jp' and 'futbollibre.net' didn't have much of a difference between their ping times and total times. Where as every other website had a massive difference. I am not really sure why this is the case but clearly something good is happening there.

Wget

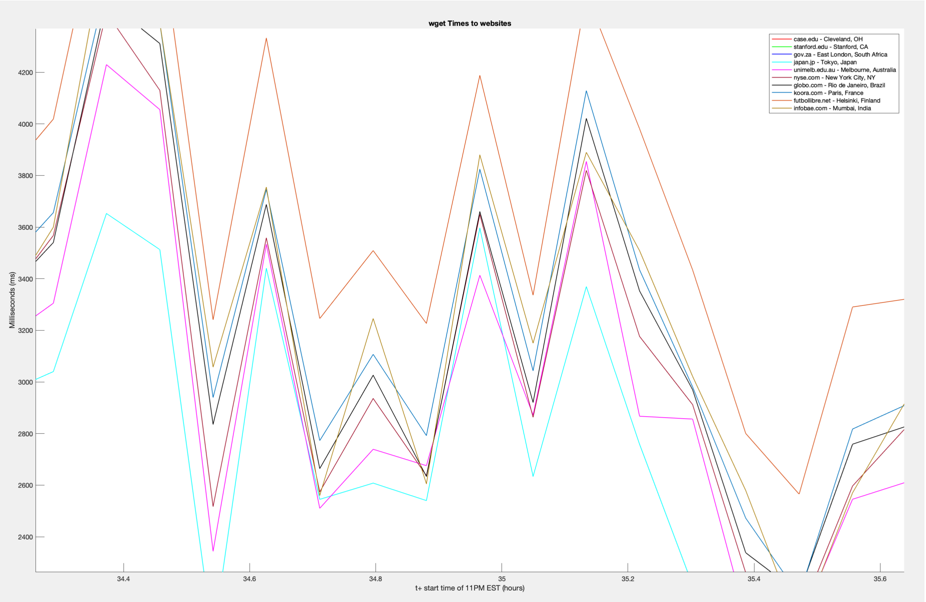
The last thing I wanted to look at was the time it took to receive data across the network. Everything before this was all about starting a connection. This will now monitor how long it takes to get data across the network using the connection. This can show us the delays over a period of multiple packets sent out.

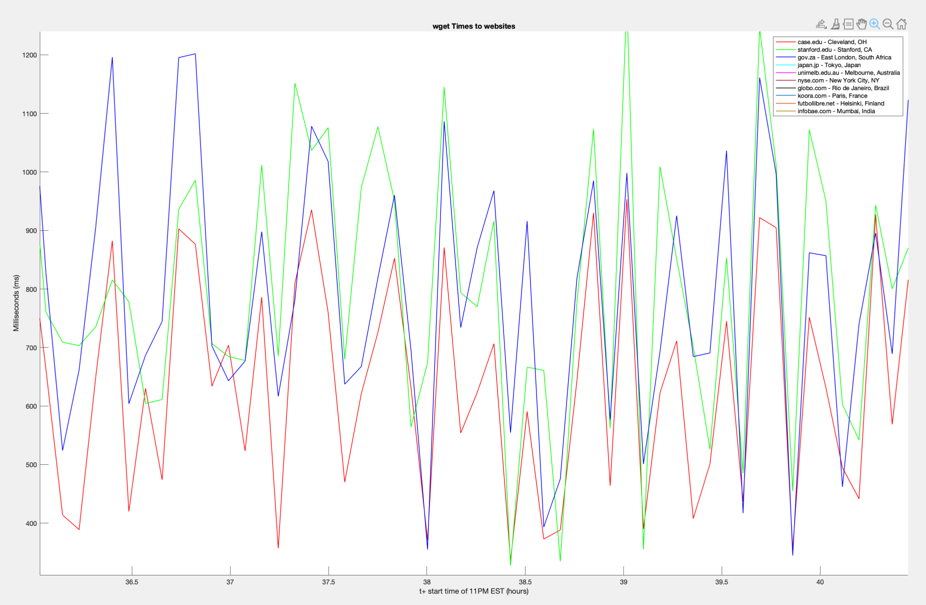


I find this data to be the most interesting. To state the obvious, there is a clear split in wget times. 'case.edu', 'stanford.edu' and 'gov.za' all share really small wget times where as the rest are above. I found a web object on each website in the range of 40 kilobytes to 200 kilobytes. The three websites I mentioned above did not have the smallest objects (so that doesn't explain the low wget times).

Website	case.edu	stanford.edu	gov.za	japan.jp	futbollibre.net
Object size	100K	120K	60K	180K	40K
Website	unimelb.edu.au	nyse.com	globo.com	koora.com	infobae.com
Object size	40K	132K	68K	106K	224K

These wget times don't really correlate to ping measurements which is interesting. I did find something else very interesting. The graph above is kind of a mess but I did that because I wanted to show how closely some websites were to each other. I have zoomed in photos below.





It is pretty apparent in the first photo, but we can see how these websites are almost perfectly offset by a small time. It isn't perfect, but it is pretty neat given that these websites were picked randomly by me, they span the globe and they all have different web object sizes.

Traceroute Time Jumps

I ran out of time because I was focusing on my final exams. I was going to analyze the time it takes for packets to jump from routers to routers to see if I can understand what was happening with my Australia pings. But I'd appreciate it if you could pretend I found something ground-braking here.