Delay as Influenced By Routing Decisions

Prarthana Gajjala, pjg83@case.edu

Table of Contents

1 Introduction

This report aims to investigate delay in web server access, especially due to network layer level decisions. It does so by analyzing various aspects of the delay in accessing the current top 100 websites, as determined by Alexa.[1]

1.1 Motivation

Between 2011 and 2021, the number of internet users increased by an estimated 2725.59 million. This is forecasted to increase by approximately 798.65 million by 2025.[2] Naturally, much of the traffic generated by this increase of users will be geared towards top websites. As such, it is important to look into how delay in accessing these websites can be minimized with increased network traffic.

When datagrams are transported from a source to a destination, they take some particular route through various routers in the network. At each individual router, or “hop”, a decision is made about the next address to which to forward the datagram. However, if

[1] http://s3.amazonaws.com/alexa-static/top-1m.csv.zip
[2] Statista. "Forecast of the number of internet users in the World from 2010 to 2025 (in millions)." Chart. May 25, 2021. Statista. Accessed December 15, 2022. https://www.statista.com/forecasts/1146844/internet-users-in-the-world

datagrams arrive faster than routers can make these decisions or transmit to the next address, routers may have to queue or buffer datagrams. This can lead to delay. As such, an increase in traffic into routers can cause larger queues and buffers can further exacerbate delay. An analysis into which specific addresses currently have delay may serve to minimize this effect. This is particularly important for popular websites.

1.2 Intended Measurements

1.2.1 Average Delay to Top 100 Websites

The average delay to access the top 100 websites will be measured. This establishes a baseline for expected delays, in addition to determining which websites already have high delays.

1.2.2 Number of Router Hops to Top 100 Websites

The number of routers a datagram traverses will be measured for the top 100 websites. This aims to find a correlation between the number of routers data encounters and delay.

1.2.3 Router Delay

The delay that occurs at each individual router hop when sending a datagram to the top 100 websites will be measured.

1.2.4 Hop Delay by Website

The average of delays that occur at individual hops when sending a datagram to each specific website in the top 100 websites will be measured. This aims to find a correlation between average router delay for all hops when accessing a website and average overall delay when accessing that same website.

1.2.5 Router Delay by IP Address

The address of the router affiliated with each hop delay will be recorded. The average delay for each router will be calculated. This aims to determine which specific routers contribute to delay.

1.2.6 Router Delay by Hop Depth

We define hop depth as the number of routers prior to this one that have been visited by a datagram traveling to a particular destination. For each hop depth, the average delay of routers at that depth will be calculated. This aims to find a correlation between the depth of routers and router delay.

2. Procedure

2.1 Raw Data Access

All raw data and project files have a copy living at:
https://drive.google.com/drive/folders/1kxSW_TwOn3yAhujUBWN0e_rosKVE1TuW?usp=sharing.
The scripts folder, graphs.xlsx file, data.txt file, and report file will also be included in the final project submission. For further information, please reference the data.txt file.

2.2 Data Gathering

To gather data, two existing tools were used.[3] Both tools utilize the Internet Control Measurement Protocol (ICMP) and communicate network layer level information. All data was collected while connected to the CaseWireless network. Measurements were gathered on the CSDS325 class server, specifically eecslab-12.case.edu.
Finally, note that while both tools were run on 100 websites, the address of one website, twimg.com could not be located. Thus, all data yielded results for 99 websites.

1. **Ping** sends an echo request to the given web site and listens for an echo response. To gather data, ping was utilized to echo request five packets of data repeatedly from the top 99 webpages for approximately 93 minutes. To minimize network disruption, these requests were not sent concurrently. Instead, five packets were sent to one of the top webpages, followed by a short delay, then five packets were sent to another webpage. The script to run these requests is titled collectPing.py, and can be accessed at pjg83-proj5/collectPing.py. To access data, visit on Google Drive pjg83-proj5/raw_data/pingData.txt.

   Example ping output:
   $ ping -c 5 google.com
   PING google.com (172.217.0.174) 56(84) bytes of data.

[3] In general, facts referenced from the Network Layer Part 3 powerpoint by Mark Allman.

```
64 bytes from yyz08s10-in-f174.1e100.net (172.217.0.174): icmp_seq=1 ttl=111 time=10.4 ms
64 bytes from yyz08s10-in-f174.1e100.net (172.217.0.174): icmp_seq=2 ttl=111 time=10.4 ms
64 bytes from yyz08s10-in-f174.1e100.net (172.217.0.174): icmp_seq=3 ttl=111 time=10.2 ms
64 bytes from yyz08s10-in-f174.1e100.net (172.217.0.174): icmp_seq=4 ttl=111 time=10.2 ms
64 bytes from yyz08s10-in-f174.1e100.net (172.217.0.174): icmp_seq=5 ttl=111 time=10.3 ms
```

2. **Traceroute** sends a series of packets to the destination, each failing by surpassing the max time to live (TTL) at the next router hop. When these routers send back a "TTL expired" response, it includes the ip information of the router. This allows us to trace the route a packet takes to get to its destination and the delay at each router hop. This route may vary. To gather data, traceroute was utilized to trace the route to the top 99 webpages for approximately 175 minutes. Each hop attempted to elicit a response three times. To minimize network disruption, these requests were not sent concurrently. A traceroute called on to one of the top webpages, followed by a short delay, then a traceroute was sent to another webpage. The script to run these requests is titled pjg83-proj5/collectTrace.py. To access data, visit on Google Drive pjg83-proj5/raw_data/traceData.txt.

   Example traceroute output:

```
$traceroute google.com
traceroute to google.com (142.250.191.206), 30 hops max, 60 byte packets
 1  _gateway (129.22.23.1)  0.287 ms  0.796 ms  0.750 ms
 2  129.22.1.226 (129.22.1.226)  0.671 ms  0.663 ms  0.678 ms
 3  10.2.0.22 (10.2.0.22)  0.555 ms 10.2.0.18 (10.2.0.18)  0.554 ms  0.660 ms
 4  * * *
 5  edge0-v103.cwru.edu (192.5.109.1)  1.018 ms  0.831 ms  1.023 ms
 …
```

   Note that sometimes, route details are hidden. These data were omitted from some analyses.

2.3 Data Analysis

All data was parsed into a .csv file and graphed. A spreadsheet with all csv data and graphs can be found pjg83-proj5/graphs.xlsx. Please open on excel.

2.2.1 Average Delay to Top 99 Websites

Each website was pinged approximately 10 times. Each ping was analyzed, and the five response times extracted. After parsing the entire file, the average and standard deviation of the delay for each website was calculated. The script to parse the ping output file and run these calculations is at pjg83-proj5/pingDataParse.py. Data was then fed into a .csv file and graphed. To access this .csv file, visit on OneDrive pjg83-proj5/.csv_files/averageDelays.csv.

2.2.2 Number of Router Hops to Top 99 Websites

Traceroute was run on each website approximately 10 times. The number of hops in each traceroute response was calculated, including hidden addresses. After parsing the entire file, the average and standard deviation of the number of hops for each website was calculated. The script to parse the traceroute output file and run all relevant calculations is at pjg83-proj5/traceDataParse.py. This was then fed into a .csv file and graphed. To access this .csv file, visit on OneDrive pjg83-proj5/.csv_files/numHops.csv.

2.2.3 Router Delay

For each iteration of traceroute, the delay in obtaining a response from each address was recorded. This includes all three attempts to elicit a response from each address. This was then fed into a .csv file and analyzed. To access this .csv file, visit on OneDrive pjg83-proj5/.csv_files/hopDelays.csv.

2.2.4 Hop Delay by Website

For each iteration of traceroute, the delay in obtaining a response from each address was recorded, and affiliated with a website. This includes all three attempts to elicit a response from each address. This was then fed into a .csv file and analyzed. To access this .csv file, visit on OneDrive pjg83-proj5/.csv_files/delaysByWebsite.csv.

2.2.5 Router Delay by IP Address

For each iteration of traceroute, the delay in obtaining a response from each address was recorded, and affiliated with this address. This includes all three attempts to elicit a response from each address. The average hop delay that occurs at each router was calculated. These values were then fed into a .csv file and analyzed. To access this .csv file, visit on OneDrive pjg83-proj5/.csv_files/delaysByIp.csv.

2.2.6 Router Delay by Hop Depth

For each iteration of traceroute, for each hop depth, the delay in obtaining a response from each address was recorded, and affiliated with the hop depth. This includes all three attempts to elicit a response from each address. The average delay that occurs at each hop depth was calculated. These values were then fed into a .csv file and analyzed. To access this .csv file, visit on OneDrive pjg83-proj5/.csv_files/hopDepth.csv**.**

2.4 Script Usage

2.4.1 collectPing.py

- Run $python3 collectPing.py
- Ensure that a directory ../raw_data exists
- Data can be found at ../raw_data/pingData.py

2.4.2 collectTrace.py

- Run $python3 collectTrace.py
- Ensure that a directory ../raw_data exists
- Data can be found at ../raw_data/traceData.py

2.4.3 pingDataParse.py

- Run $python3 pingDataParse.py
- Ensure collectPing.py has been run or a file ../raw_data/pingData.py exists with appropriate data
- Information detailed in section 2.2.1 will be printed

2.4.4 traceDataParse.py

- Run $python3 traceDataParse.py
- Ensure collectPing.py has been run or a file ../raw_data/traceData.py exists with appropriate data
- When prompted, enter the analysis procedure desired, referencing section 2.3
- Appropriate information will be printed

3 Results

3.1 Average Delay to Top 99 Websites

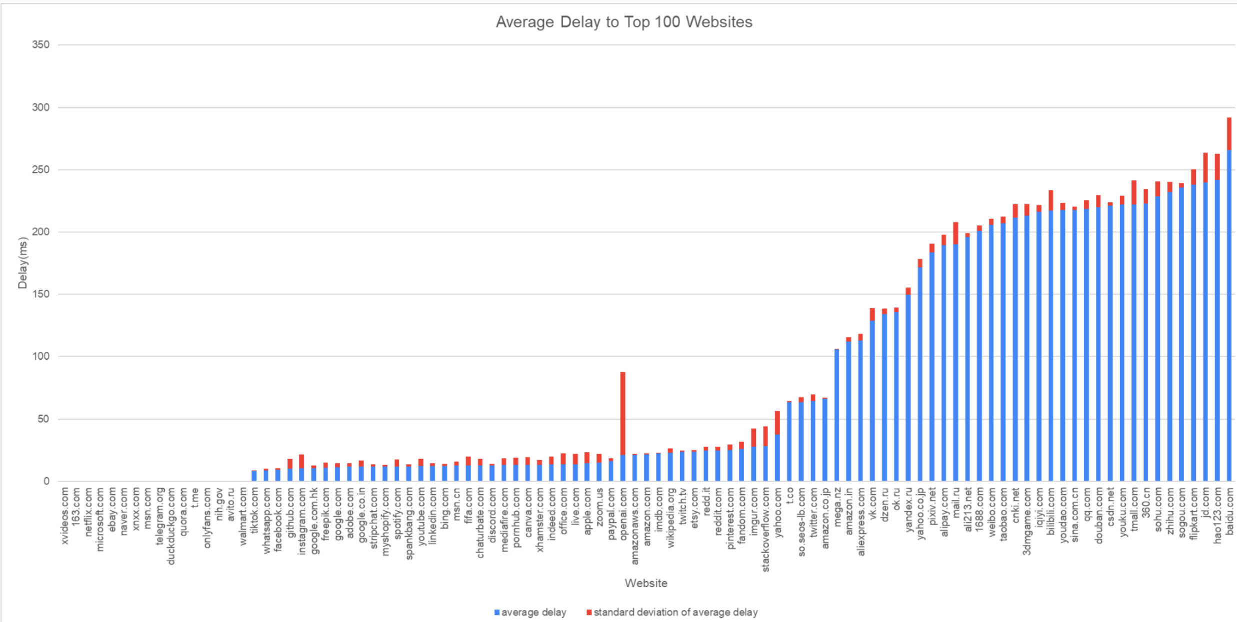

Figure 1: Bar Graph of Average Delay to Top 99 Websites

Figure 1 depicts the average delay when pinging the top 99 websites in 5 packet bursts approximately 10 times. The blue portion of the bar shows these averages, while the red portion shows the standard deviation of the delays.

Statistic	Value	Website
Average mean delay	92.90870943 ms	
Std Dev mean delay	91.99451254 ms	
Average std dev	7.165198523 ms	
Max mean delay	265.9 ms	baidu.com
Min mean delay	8.184 ms	tiktok.com
Max std dev	66.516252 ms	openai.com
Min std dev	0.09209578 ms	tiktok.com

Table 1: Statistics Regarding Average Delay to Top 99 Websites

Table 1 depicts statistics about the average delay when pinging the top 99 websites in 5 packet bursts approximately 10 times.

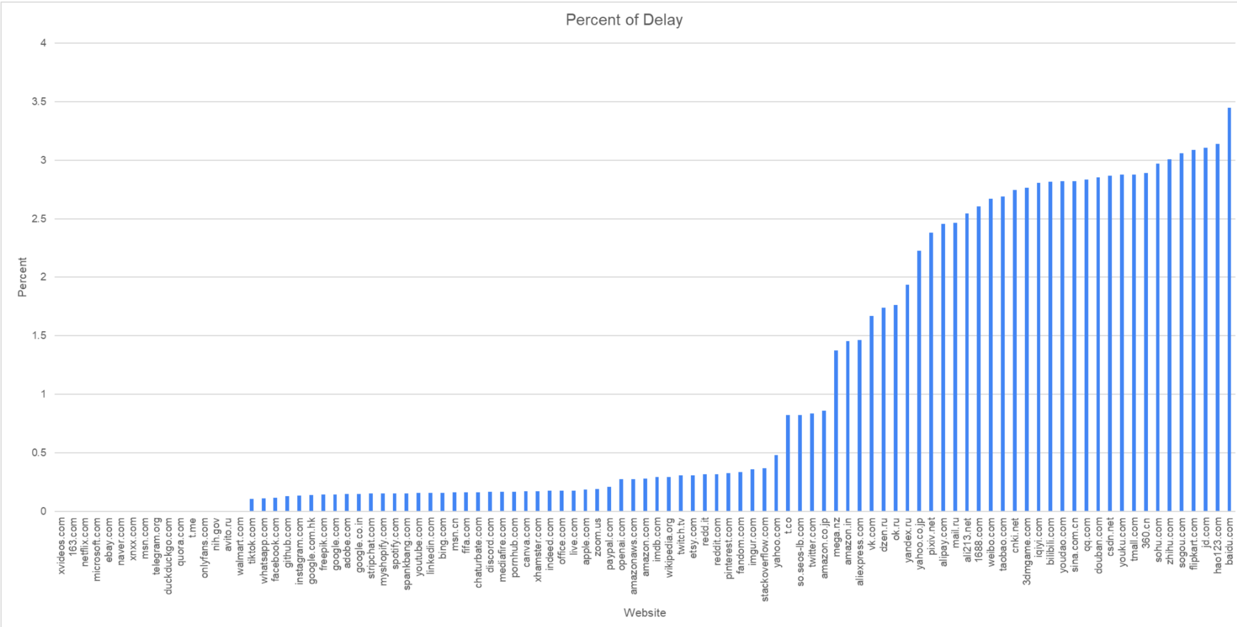

Figure 2: Bar Graph of Percent Delay of each Top 99 Website

Figure 2 depicts the percent delay of the summed average delays each website composes.

Note first that 16 websites have average delays of 0. These websites had 100% packet loss, and as such, delay could not be calculated. It is likely that these websites do not support ICMP. These websites were omitted when calculating the values in Table 1.

As seen in Figure 2, over 60% of websites use, on average, less than 0.5% of the total delay. The majority of delay lies in just 16 websites. Furthermore, while the top two websites, google.com and youtube.com have average delays under 15 ms, the third top website has the largest average delay, of 265.9 ms. Therefore, while connected to the CaseWireless network, most delay can be found in a minority of websites. Efforts to minimize delay can be directed at requests sent to those websites.

3.2 Number of Router Hops

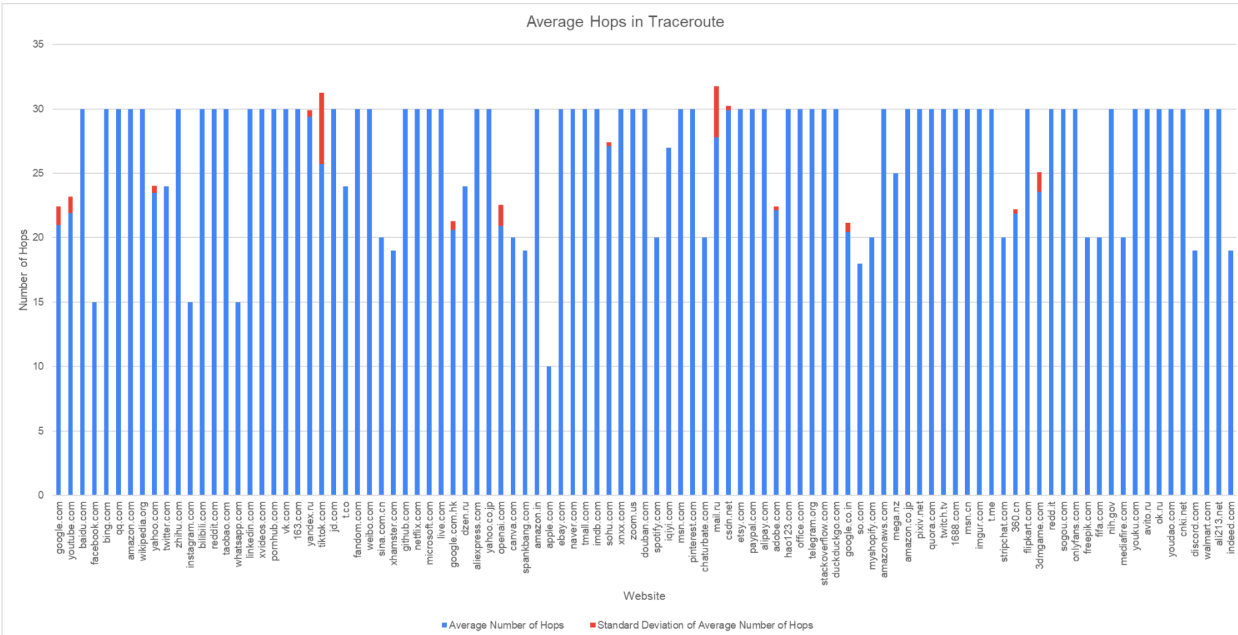

Figure 3: Bar Graph of Number of Router Hops to Top 99 Websites

Figure 3 depicts the average number of hops when running traceroute on the top 99 websites approximately 10 times. The blue portion of the bar shows these averages, while the red portion shows the standard deviation of the delays.

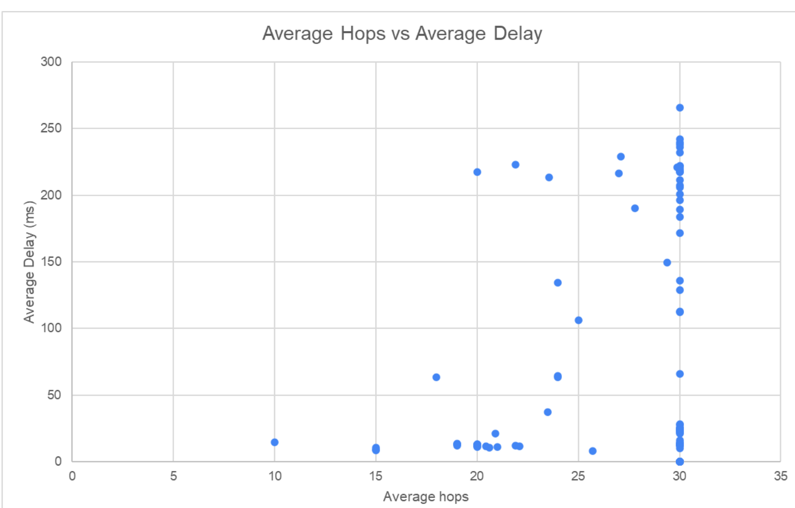

Figure 4: Average Hops vs Average Delay for Top 99 Websites

Figure 4 shows the relationship between the average number of router hops traceroute yields for the top 99 websites, and the average delay to ping those websites as described in section 3.1.

To minimize network disruption, traceroute attempted a maximum of 30 hops on any given website. As seen above in Figures 3 and 4, 62 websites required more than 30 hops to reach the receiver. Thus, there are sharp cutoffs in the data in the graphs above.

However, even without websites whose traceroute was cut off by the 30 hop limit, there is no clear correlation between the number of hops a datagram encounters and the delay in accessing a website, as seen in Figure 4. As such, while connected to the CaseWireless network, though many datagrams going to popular websites are passed through many routers (over 30), encountering a large number of routers does not increase delay.

3.3 Router Delay

Average Router Delay	26.87578 ms
Standard Deviation Router Delay	56.17837 ms
Minimum Router Delay	0.075 ms
Maximum Router Delay	956.919 ms

Table 2: Statistics Regarding Router Delay for Top 99 Websites

Table 2 depicts statistics about the delay in obtaining a response from each router address when running traceroute on the top 99 websites approximately 10 times.

As seen in table 2, when running traceroute on the top 99 websites, the router delay had a large standard deviation, more than twice the average router delay. This hints that there may be some influences in router delay. While this result is not very interesting on its own, when combined with the next few results, it yields some interesting patterns.

3.4 Hop Delay by Website

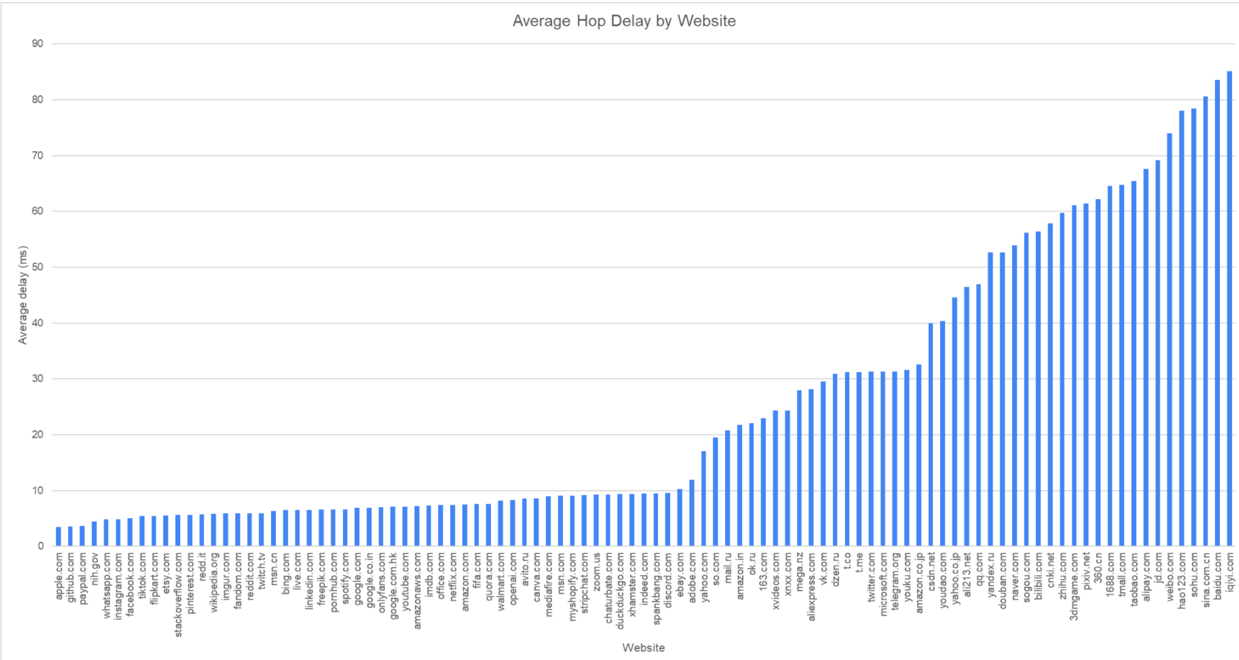

Figure 5: Bar Graph of Average Hop Delay When Running traceroute on Top 99 Websites

Figure 5 depicts the average delay of individual websites in obtaining a response from each hop address when running traceroute on the top 99 websites approximately 10 times.

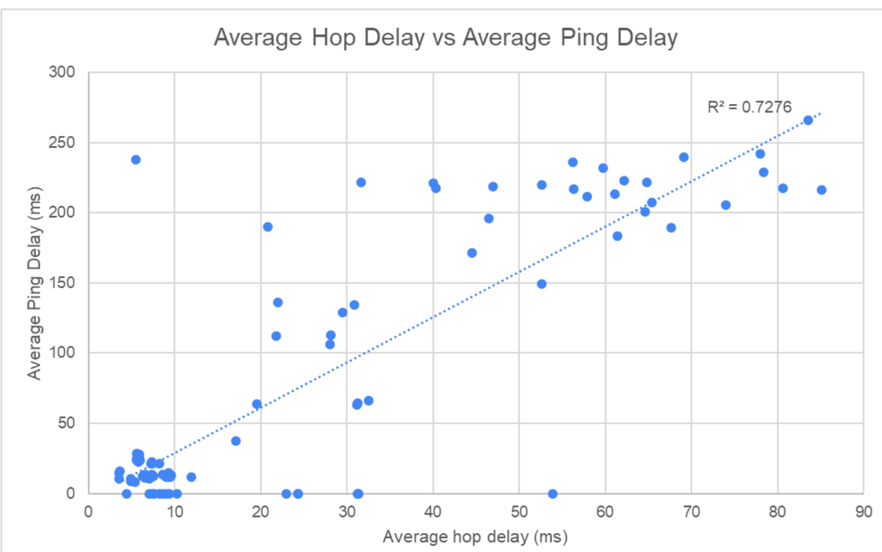

Figure 6: Average Hop Delay by Website vs Average Ping Delay by Website

Figure 6 shows the relationship between the average delay of individual websites in obtaining a response from each hop address[4] and the average delay of when pinging individual websites.[5]

[4] See Figure 5
[5] See Figure 1

Figure 6 shows a strong positive correlation between the average hop delay when running traceroute on a specific website and the average delay merely pinging a specific website. In Section 3.2, we found that the number of router hops a datagram encounters does not correlate with the overall delay in accessing a website. However, here, we determine that when connected to the CaseWireless network the delay of *individual* hops contribute strongly to the overall delay when accessing a website.

Inspecting and decreasing delays in these individual routers, thus, will successfully decrease overall delay in accessing a website. Next, we look at individual routers to determine which routers contribute to delay.

3.5 Router Delay by IP Address

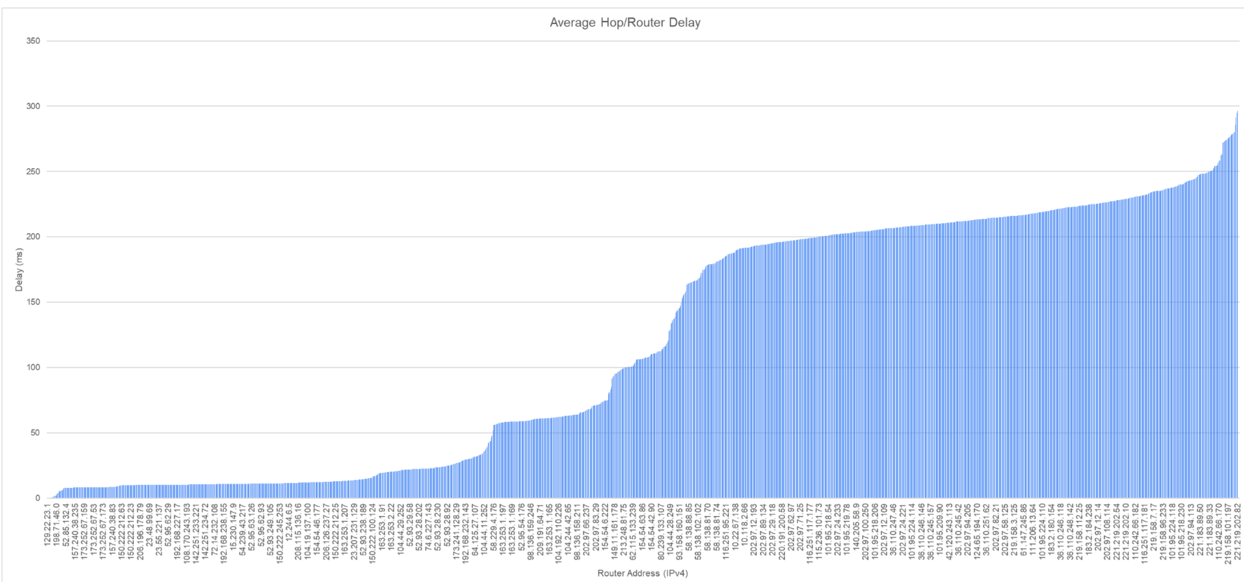

Figure 7: Bar Graph of Average Hop/Router Delay When Running traceroute on Top 99 Websites, by Router Address

Figure 7 depicts the average delay of individual websites in obtaining a response from each router address when running traceroute on the top 99 websites approximately 10 times.

We see here that the majority of routers have a delay of under 100 ms. Furthermore, only a handful of routers have average delay of over 250 ms (36 routers, making up approximately 2.3% of routers encountered). This strengthens the claim that when connected to the CaseWireless network, *individual* router delays contribute largely to overall delay, rather than the number of routers encountered. Focusing on reducing the delay of these individual routers can greatly reduce overall delay for popular websites.

Since data was collected while connected to the CaseWireless network, the first few hops in every traceroute used similar routers. These routers have the smallest hop delay. This informs the next result about hop “depth” - the number of hops in a specific router is in a traceroute.

3.6 Router Delay by Hop Depth

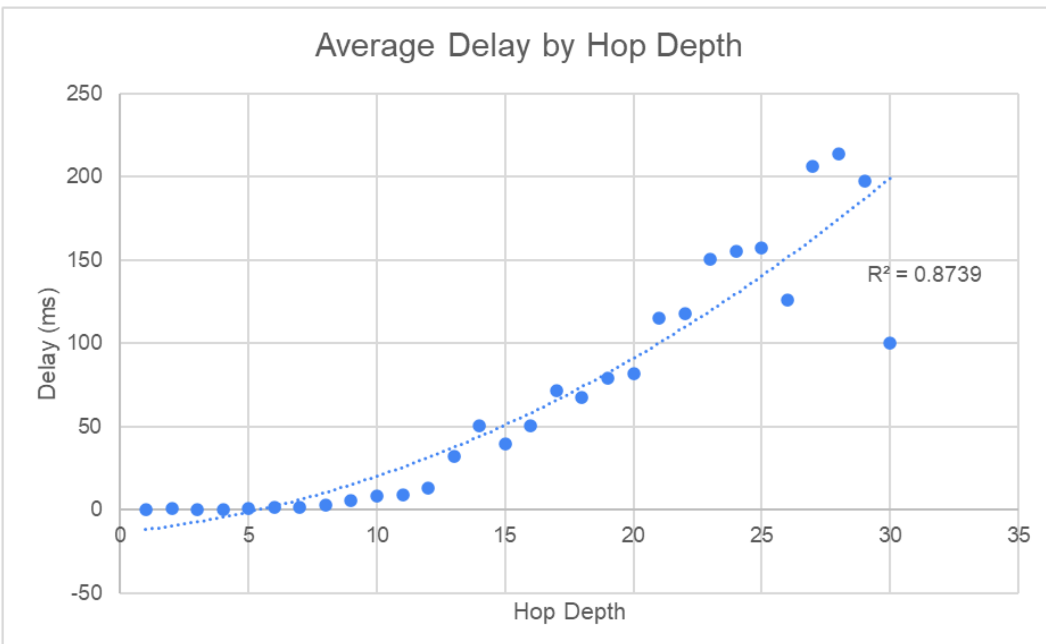

Figure 8: Hop Depth vs Average Delay at that Depth

Figure 8 shows the correlation between hop depth and the average delay of all IP addresses at that hop depth, for all traceroute runs. The trendline fit to this data is polynomial of degree 2.

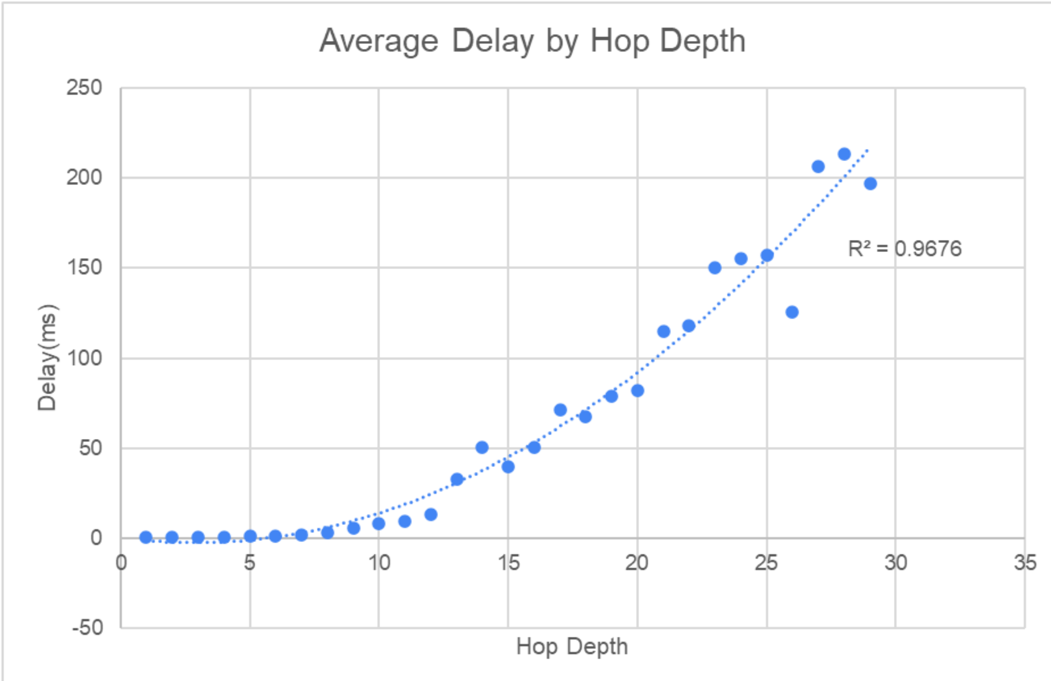

Figure 9: Hop Depth vs Average Delay at that Depth Without Cutoff Point

Figure 9 shows the correlation between hop depth and the average delay of all IP addresses at that hop depth, for all traceroute runs. The point with hop depth of 30 is discarded, since this is the cutoff depth of the traceroute call, and yields an outlier point. Removing this outlier yields a substantially stronger relation. The trendline fit to this data is polynomial of degree 2.

Figure 8 and 9 show a strong polynomial correlation between hop depth and delay. We know from section 3.2 that this does not mean that more hops lead to higher delay, especially because there is a correlation between average hop delay and overall delay.[6] Instead, this is a correlation between the specific routers that lie at larger hop depths and delay. When connected to the CaseWireless network, routers that lie at larger hop depths have parabolically larger delays. This further contributes to the conclusion that only a few routers are largely responsible for delay.

[6] See section 3.4