AON PROJECT 2- REPORT

Introduction:

As digital connectivity becomes increasingly essential, the resilience and trustworthiness of the Internet's underlying infrastructure are critical for seamless communication and data exchange. This project delves into the behavior and stability of Internet paths using traceroute, a diagnostic tool that traces the route packets take from a source to a destination across the intricate network of the Internet. Through systematic collection and analysis of path data from specific IP destinations, this study seeks to identify patterns of path stability and variability, offering insights into the intricacies of Internet routing.

Part 1: Measurements/Data collection

The methodology for this study involved the following steps:

1. **Measurement Tool Utilization:**
   Traceroute was employed to record the route of packets to five predetermined IP destinations. This tool provided detailed information on the routers traversed, packet loss, and delay at each hop.
2. **Selection of Destinations:**
   The destinations included popular domain names such as www.google.com, www.pitt.edu, www.yahoo.com, www.github.com, and www.microsoft.com, offering a diverse set of paths for analysis.
3. **Data Collection Script:**
   A Bash script automated the process of data collection, conducting **20 measurements** per destination over a span of **6 days** i.e **120 measurements** per destination to account for potential temporal variations in path and delay.
4. **Data Collection Strategy:**
   Measurements were conducted from the same source machine, using a wired connection to ensure minimal latency fluctuations. The

script was executed to generate text files, each representing a single measurement instance.

5. **Directory Organization:**
   The collected data was meticulously organized into directories, each labeled with the date of measurement to facilitate structured analysis and traceability.
6. **Sleep Interval Management:**
   Between each measurement, a sleep interval ensured that data points were not collected in rapid succession, reducing the risk of temporal bias and providing a more accurate representation of path variability.
7. **Data Output Handling:**
   The output from each traceroute instance was redirected to individual text files named systematically to correspond with the destination and measurement number, simplifying the subsequent data analysis process.
8. **Preparation for Analysis:**
   Following data collection, each router-level path was prepared to be translated into an autonomous-system-level path using tools such as the WHOIS database, setting the stage for a granular analysis of path stability.

Part 2: Data analysis

A. Converting IPs to AS.

The objective of this phase was to convert the router-level paths obtained from traceroute measurements into a representation that aligns with the Internet's Autonomous System (AS) structure. To accomplish this, a Python script has been developed that automates the translation process. The script operates in multiple steps:

1. **Data Extraction**: The script walks through a directory of text files containing traceroute results. Each file corresponds to a set of traceroute measurements for a particular URL. The script identifies the URL from the filename and parses the traceroute data to extract hop numbers, IP addresses, and latency measurements.

2. **Unique IP Collection**: As traceroute data includes IP addresses for each hop, the script collects these into a set to avoid duplicate queries, which prepares for a bulk WHOIS lookup.

3. **Bulk WHOIS Lookup**: Utilizing the WHOIS database service provided by Team Cymru, the script sends a bulk request to translate the collected IP addresses into AS numbers. The response is then sorted and parsed. Each IP address is matched with its corresponding AS number, which is stored in a dictionary for quick access.

4. **Data Aggregation and Output**: Finally, the script revisits the traceroute data, now substituting each IP address with its AS number. The completed dataset includes the date of measurement, hop number, original IP address, AS number, and latency measurements. This dataset is outputted as a CSV file, which provides a structured format suitable for further analysis or visualization.

Through this approach, we can analyze the AS-level paths and observe how data travels through different network domains. This high-level perspective is invaluable for understanding the Internet's structure and the dynamics of path selection, which are governed by the routing policies of individual ASes. The AS-level path data also serves as a foundational element for the subsequent stages of our project, which include stability analysis and delay characterization.

B. Stability of Paths - Router Level and AS Level

Router Level Analysis:

Most dominant paths and their counts:

The table presented below shows the most dominant paths taken to reach each URL as determined by traceroute, along with the frequency of each path's occurrence.

URL	Path	Count
www.github.com	('172.25.192.1', '10.0.0.1', '96.120.62.169', '96.110.215.137', '96.110.120.93', '162.151.65.5', '96.110.42.173', '96.110.38.134', '96.110.39.165', '68.86.84.145', '96.110.32.126', '50.248.119.26', '98.124.190.162')	2
ww.google.com	na	na
www.microsoft.com	('172.25.192.1', '10.0.0.1', '96.120.62.169', '96.110.215.125', '162.151.152.210', '96.110.120.93', '162.151.65.5', '50.216.209.234')	59
www.pitt.edu	('172.25.192.1', '10.0.0.1', '96.120.62.169', '96.110.215.137', '96.110.120.93', '162.151.65.5', '96.110.42.169', '96.110.38.138',	2

	'96.110.32.101', '68.86.84.145', '96.110.32.122')	
www.yahoo.com	('172.25.192.1', '10.0.0.1', '96.120.62.169', '96.110.215.125', '162.151.152.210', '96.110.120.93', '162.151.65.5', '4.68.106.85', '4.16.246.238', '74.6.225.151', '209.73.184.57', '69.147.92.12')	2

1. **www.github.com**: The most common path taken to reach GitHub involved a sequence of 13 hops, starting with IP 172.25.192.1 and ending with IP 98.124.190.162. This particular path was observed 2 times during the measurement period. The fact that there's only a count of 2 suggests that either the path to GitHub changed often or the measurement period captured a limited number of samples.
2. **www.google.com**: There was no single dominant path, the paths were too variable to have a clear "most common" route, which is typical for a global service with a distributed infrastructure.
3. **www.microsoft.com**: The path to Microsoft was more consistent, with the most common path involving 8 hops, starting with IP 172.25.192.1 and ending with IP 50.216.209.234. This path was observed 59 times, indicating a high level of path stability during the measurements.
4. **www.pitt.edu**: The path to the University of Pittsburgh included 11 hops and was observed 2 times. Similar to GitHub, this suggests either a high level of path variability or a small number of samples.

5. **www.yahoo.com**: The path to Yahoo involved a sequence of 12 hops, starting with IP 172.25.192.1 and ending with IP 69.147.92.12. This path was also observed 2 times, which could indicate variability in the routing or limited sampling.

Path Change Frequencies:

URL	Change Frequency
www.github.com	94.1%
www.google.com	99.1%
www.microsoft.com	36.1%
www.pitt.edu	97.5%
www.yahoo.com	98.3%

1. **www.github.com** : The paths to GitHub changed in 94.1% of measurements. This is a high change frequency, suggesting that the route to GitHub was highly variable during the measurement period.
2. **www.google.com** : Google's paths changed in 99.1% of measurements, indicating that nearly every measurement recorded a different path. This could be due to Google's global and distributed network infrastructure, which dynamically adjusts routes.
3. **www.microsoft.com** : The paths to Microsoft had the lowest change frequency, with 36.1% of measurements showing different paths. This indicates a moderate level of path stability relative to the others.
4. **www.pitt.edu** : The paths to the University of Pittsburgh changed in 97.5% of measurements, suggesting that the route to this destination was also highly variable.
5. **www.yahoo.com** : Yahoo's paths changed in 98.3% of measurements, suggesting high variability in the routing paths to Yahoo's network.

Most variable hop for each URL:

The table below describes the "Most Variable Hop" data, which indicates the specific hop (by its sequence number in the path) that experienced the most changes across the traceroute measurements for each URL. This hop is where the path variability was highest, implying either different routers were encountered at this stage in the path, or the path was intermittently missing this hop.

URL	Most Variable Hop	Changes
www.github.com	10	102
www.google.com	16	117
www.microsoft.com	7	42
www.pitt.edu	11	106
www.yahoo.com	10	113

1. **www.github.com**: Hop number 10 for GitHub experienced the most changes with 102 instances of variability. This could be a key transition point in the network where multiple routes converge or diverge.
2. **www.google.com**: Hop number 16 for Google had the highest number of changes, with 117 instances. Given Google's distributed network, this hop could be part of a dynamic segment of the network that frequently re-routes traffic.
3. **www.microsoft.com**: Hop number 7 for Microsoft changed 42 times. This is the lowest among the listed URLs, suggesting that while Microsoft's paths were relatively stable, there was still some variability at this point in the network.
4. **www.pitt.edu**: Hop number 11 for the University of Pittsburgh changed 106 times, indicating significant path variability at this junction.
5. **www.yahoo.com**: Hop number 10 for Yahoo saw 113 changes, suggesting this hop is a point of high variability in the path to Yahoo.

Visualization-Router Level Stability

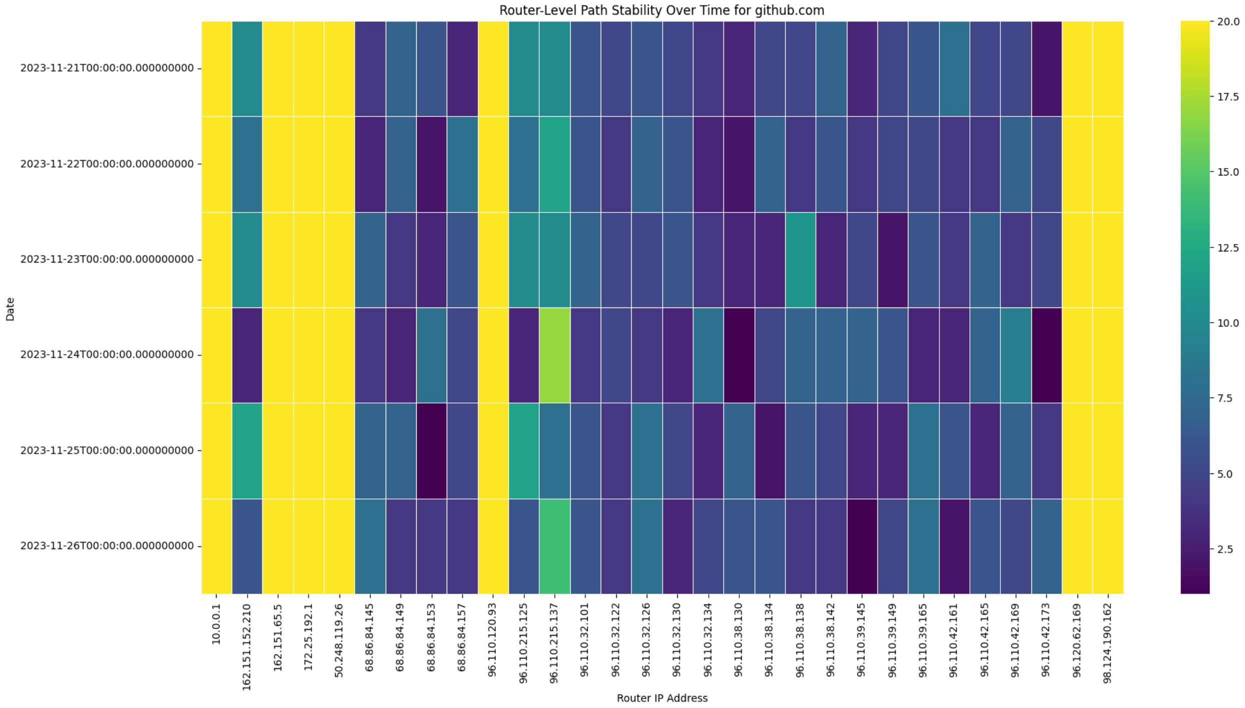
The heat maps created visualize router-level path stability over time for different URLs. Each heat map corresponds to traceroutes to a specific URL (such as github.com, google.com, microsoft.com, pitt.edu, and yahoo.com). They show which routers were encountered on specific days and how frequently each router was part of the path.

X-axis (Router IP Address): Each column represents a unique router identified by its IP address.

Y-axis (Date): Each row represents a date when the traceroute was run.

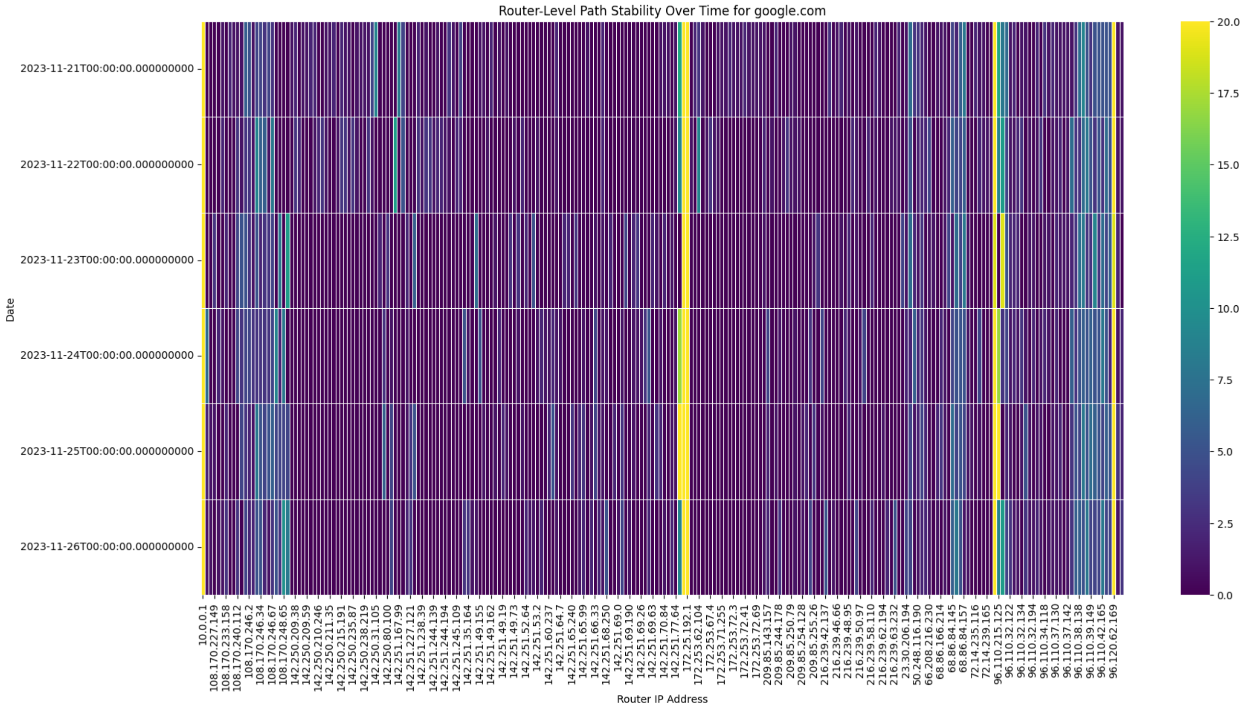
Color Intensity: The color in each cell indicates the number of times a particular router (IP address) was encountered on that date. Darker colors indicate higher frequency, suggesting that the router was a consistent part of the path on that day. Lighter colors or yellow indicate fewer encounters, suggesting less consistency or that the router was not part of the path on those dates.

www.github.com



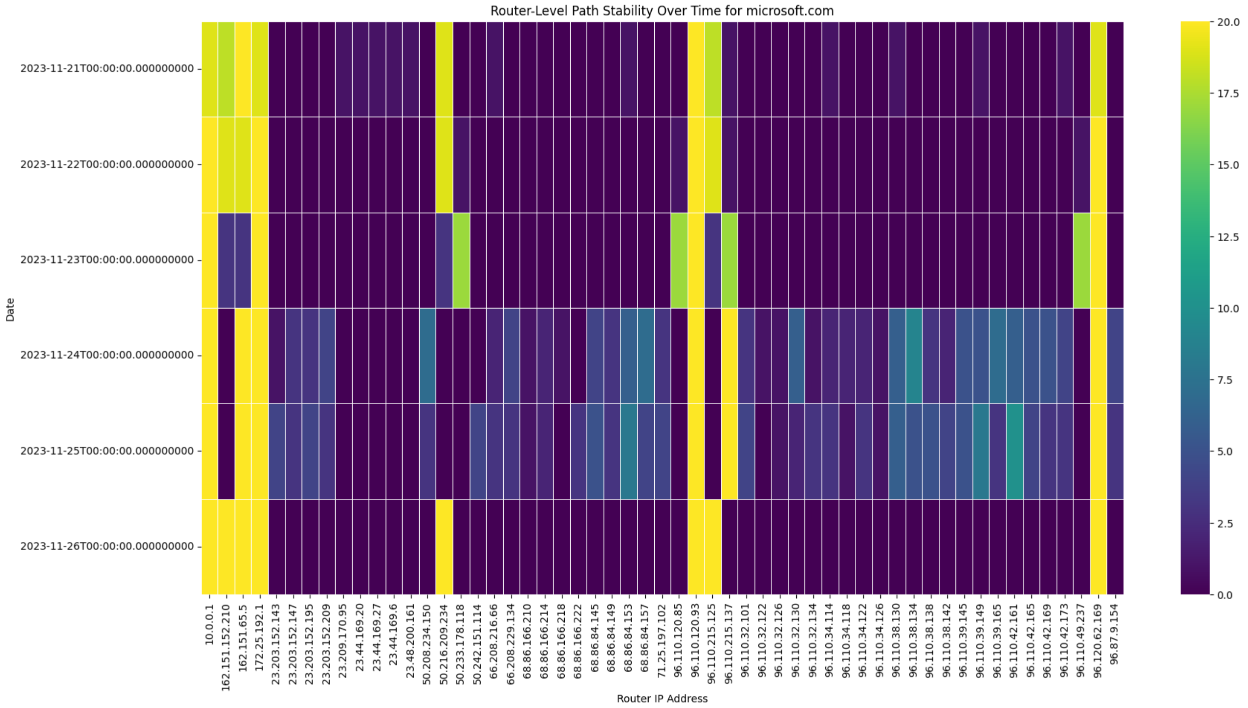
The presence of consistently dark columns across multiple dates suggests that certain routers were consistently part of the path to GitHub. Yellow columns indicate that some routers were not encountered on those days, which could mean a path change or that those routers did not respond to traceroute probes on those dates.

www.google.com



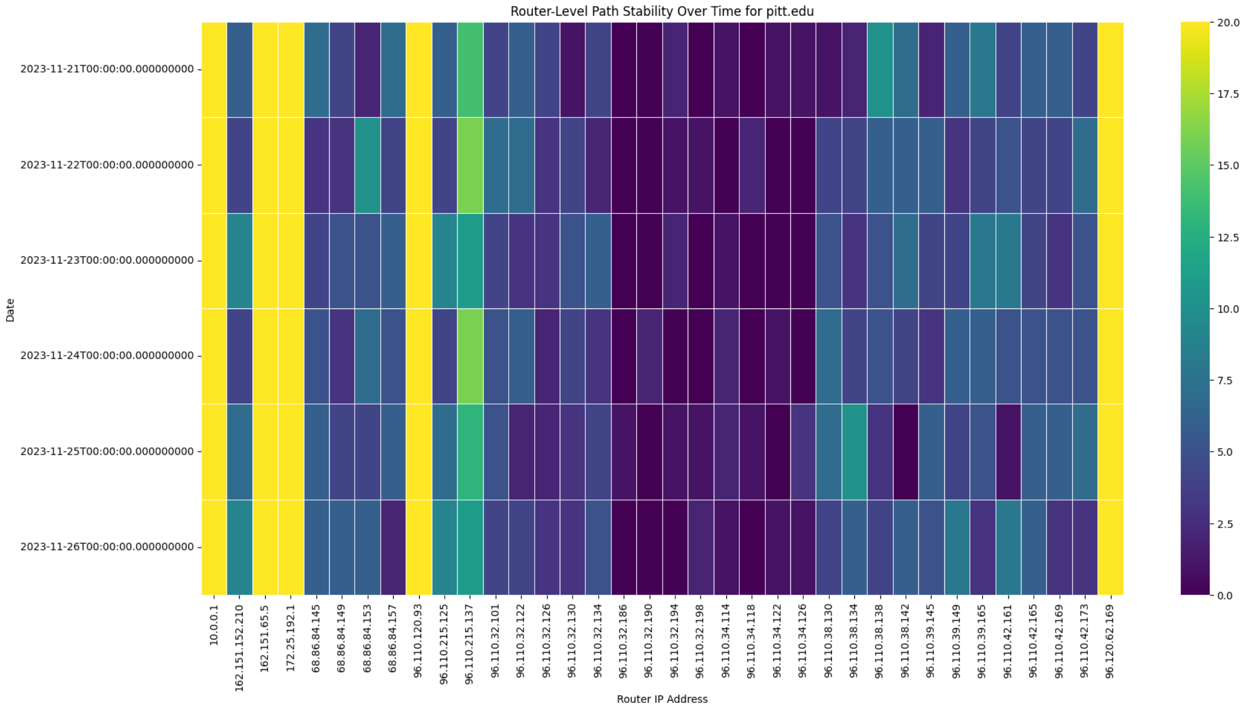
This heatmap shows more variability with alternating colors, indicating that the path to Google changed more frequently over the observed period. It suggests that different routers were used on different days, which is common for services with a global presence and load-balancing across multiple data centers.

www.microsoft.com



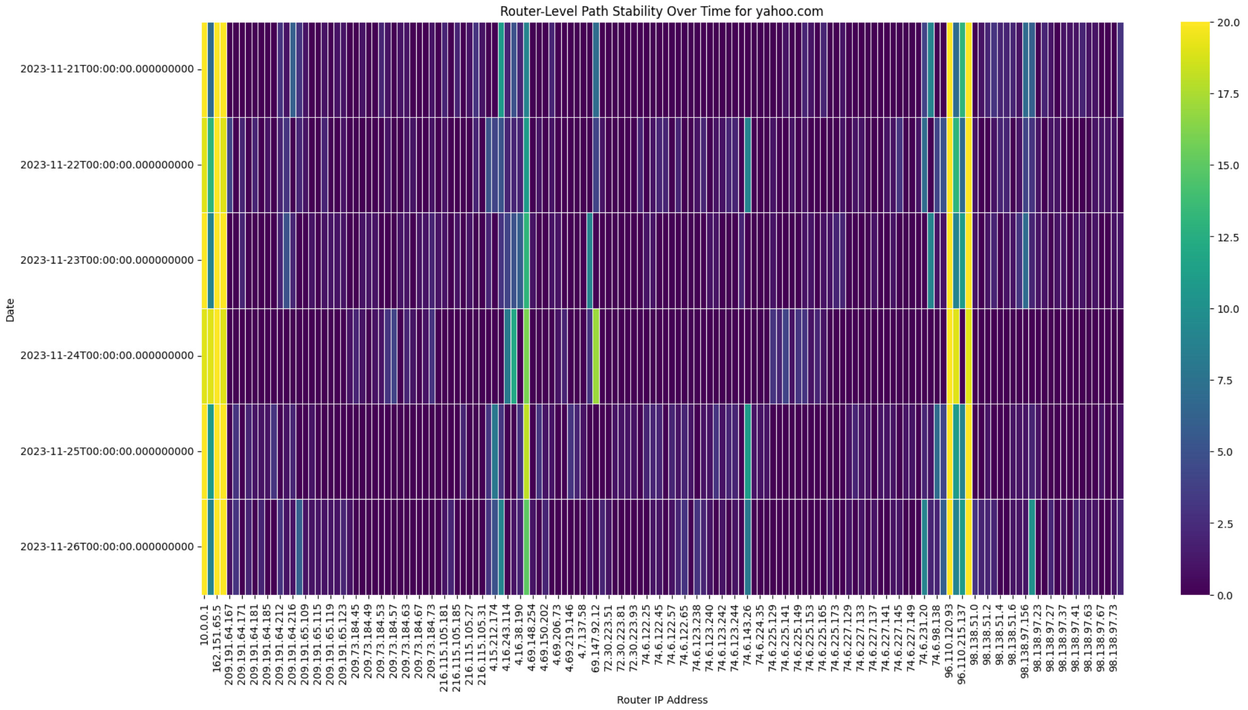
Similar to the heatmap for Google, there is a mix of colors, indicating path changes over time. However, there are some routers that appear to be consistently part of the path, as indicated by vertical lines of darker colors.

www.pitt.edu



The pattern here is similar to the one for GitHub, with some routers appearing consistently and others less so. This could reflect a relatively stable path to this destination, with occasional changes.

www.yahoo.com



There's a high degree of variability in the routers encountered in the traceroutes to Yahoo. This suggests that the path to Yahoo might be dynamic, possibly due to routing policies, load balancing, or the distributed nature of Yahoo's infrastructure.

AS Level:
Most dominant paths and their counts :

Note: The "nan" values in the paths represent hops where the AS number could not be determined, possibly due to private IP addresses.

URL	Path	Count
www.github.com	nan nan 7922.0 7922.0 7922.0 7922.0 7922.0 7922.0 7922.0	72

	7922.0 7922.0 7922.0 7922.0 7922.0 3257.0	
ww.google.com	nan nan 7922.0 7922.0 7922.0 7922.0 7922.0 7922.0 7922.0 7922.0 7922.0 7922.0 7922.0 15169.0 15169.0 15169.0 15169.0 15169.0 15169.0 15169.0	37
www.microsoft.com	nan nan 7922.0 7922.0 7922.0 7922.0 7922.0 7922.0	77
www.pitt.edu	nan nan 7922.0 7922.0 7922.0 7922.0 7922.0 7922.0 7922.0 7922.0 7922.0	81
www.yahoo.com	nan nan 7922.0 7922.0 7922.0 7922.0 3356.0 3356.0 10310.0 10310.0 36646.0 36646.0 36646.0 36646.0	29

1. **www.github.com**: The most frequent AS path to GitHub involved mainly AS 7922, followed by a transition to AS 3257. This path was observed 72 times, indicating that AS 7922 plays a significant role in the routing to GitHub.

2. **www.google.com**: The path to Google predominantly stayed within AS 7922 before moving to AS 15169. This path was observed 37 times. AS 15169 is known to be associated with Google, so it's common to see this AS in the path towards Google services.

3. **www.microsoft.com**: The path to Microsoft consistently involved AS 7922, observed 77 times, showing a high level of path stability within this AS.

4. **www.pitt.edu**: The path to the University of Pittsburgh stayed within AS 7922 for all observed hops and was seen 81 times, indicating a very stable route.

5. **www.yahoo.com**: The path to Yahoo transitioned from AS 7922 to AS 3356 and then to AS 10310 and AS 36646. This path was observed 29 times, suggesting a more complex route with multiple AS transitions.

Path Change Frequencies:

URL	Change Frequency
www.github.com	0.83%
www.google.com	6.67%
www.microsoft.com	3.36%
www.pitt.edu	0.83%
www.yahoo.com	11.02%

1. **www.github.com** (0.83%): The path to GitHub changed in 0.83% of measurements, showing a high level of stability in AS-level paths.

2. **www.google.com** (6.67%): The path to Google changed in 6.67% of measurements, indicating more variability in the AS-level routing to Google compared to GitHub.

3. **www.microsoft.com** (3.36%): The path to Microsoft changed in 3.36% of measurements, showing moderate stability.

4. **www.pitt.edu** (0.83%): Similar to GitHub, the path to the University of Pittsburgh showed high stability with changes in only 0.83% of measurements.

5. **www.yahoo.com** (11.02%): Yahoo had the highest path change frequency at 11.02%, indicating that the AS-level path to Yahoo is more variable than the others.

Most variable hop:

URL	Most Variable Hop	Changes
www.github.com	13	32
www.google.com	13	17
www.microsoft.com	1	0
www.pitt.edu	1	0
www.yahoo.com	10	47

1. **www.github.com**: Hop 13 had the most changes (32 times) for GitHub, which could be a key junction point in the network where routing decisions lead to variability.

2. **www.google.com**: Hop 13 also had the most changes (17 times) for Google, which may indicate a dynamic routing environment at this point in the network, possibly due to traffic management or load balancing.

3. **www.microsoft.com** and **www.pitt.edu**: There were no changes observed for the most variable hop, indicating that the AS-level path was completely stable during the measurement period for the first hop.

4. **www.yahoo.com**: Hop 10 saw the most changes (47 times), suggesting this is a critical point for routing variability in the path to Yahoo.

General Observation:

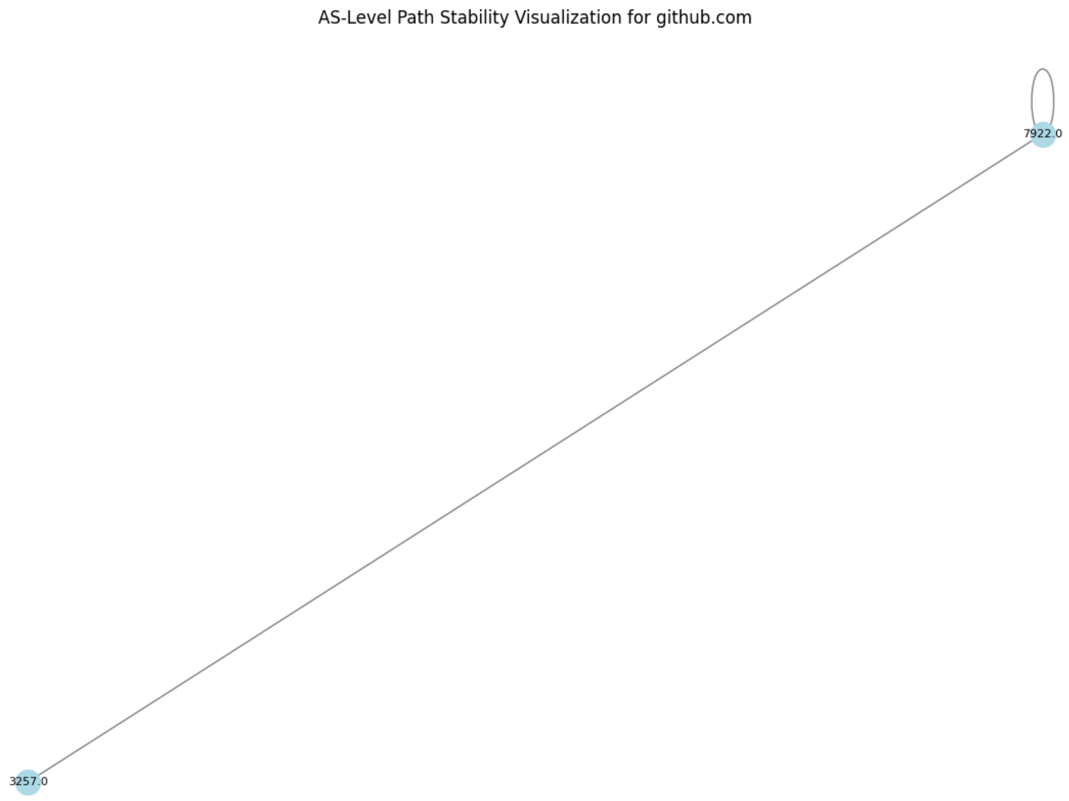
The AS-level data indicate that for some services like GitHub and Pitt.edu, the AS paths are quite stable, while for others like Google and especially Yahoo, there is more variability. The most variable hop data specifically point out which hop in the path tends to change the most, offering insights into where the network might be experiencing the most dynamic routing or traffic distribution changes.

Visualization-AS Level Stability

The graphs below are network graphs that represent the AS-Level Path Stability for different websites: GitHub, Google, Microsoft, the University of Pittsburgh (pitt.edu), and Yahoo. Each node in the graph represents an Autonomous System (AS) identified by its number, and each edge represents a connection that was observed between two ASes in the traceroute data.

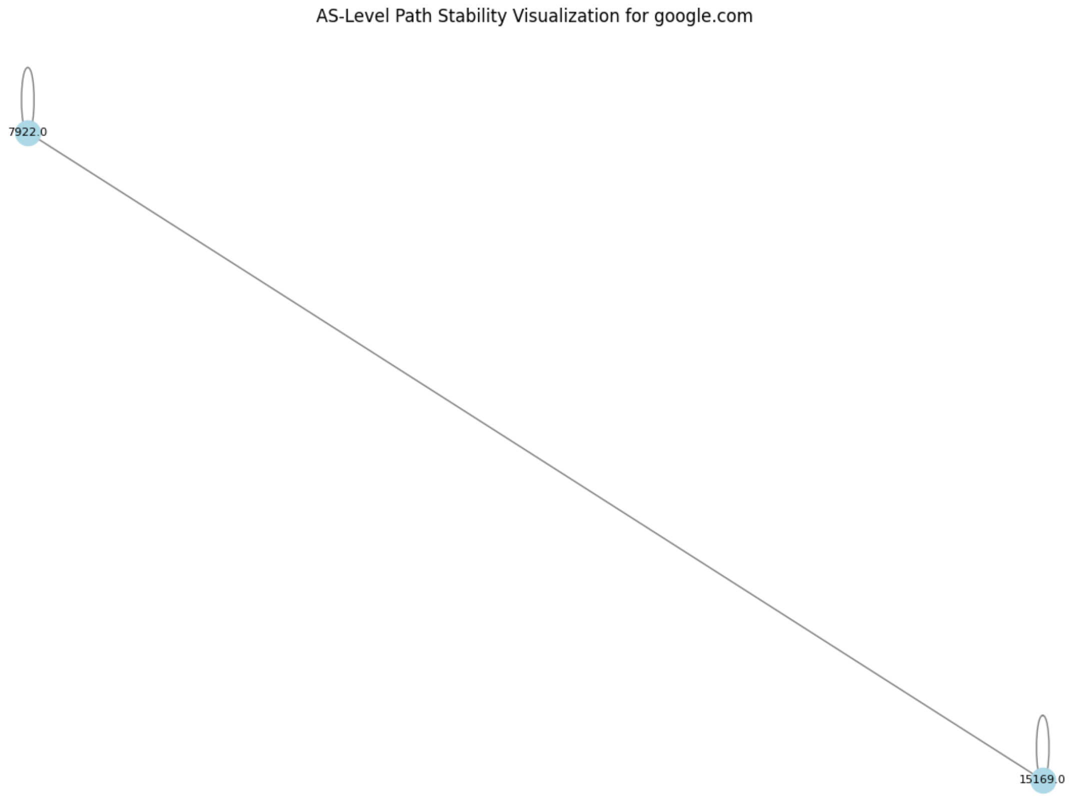
The structure of these graphs can give insights into the routing policies and interconnectivity of different ASes. For instance, a graph with few nodes and edges would indicate a simpler and potentially more stable routing path, while a graph with many nodes and edges might indicate a more complex and dynamic routing environment.

www.github.com



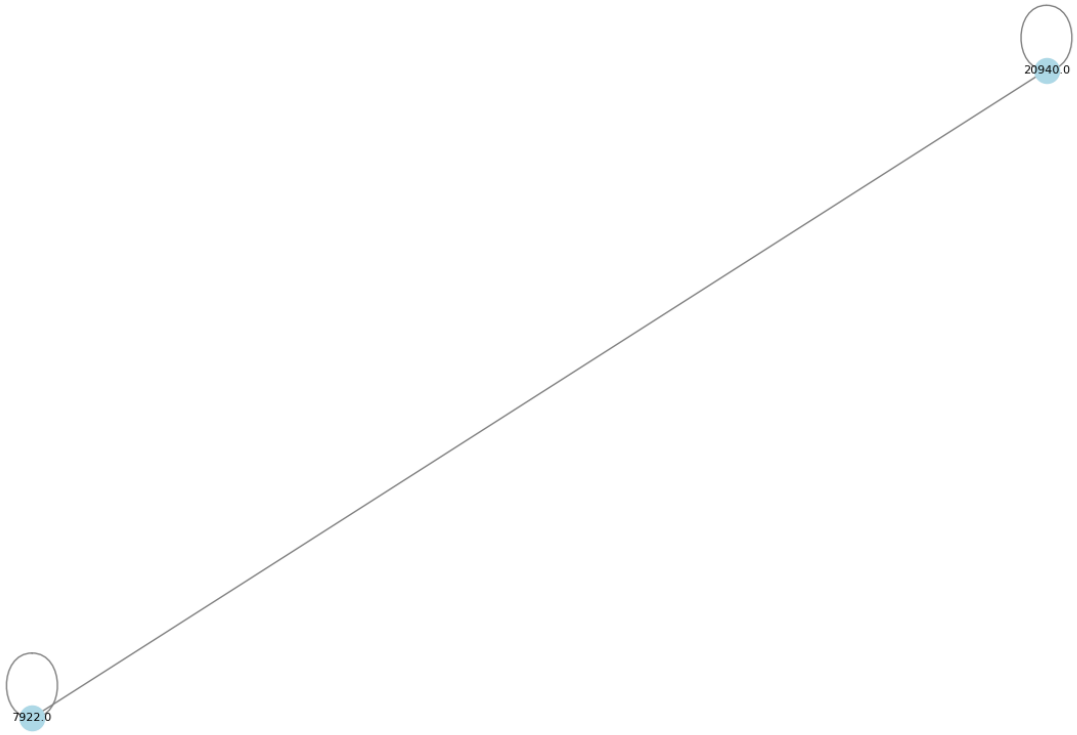
This graph shows a linear path mostly within AS 7922 before reaching AS 3257. The simplicity of the graph suggests a consistent and straightforward path with few AS transitions, which indicates a stable route.

www.google.com



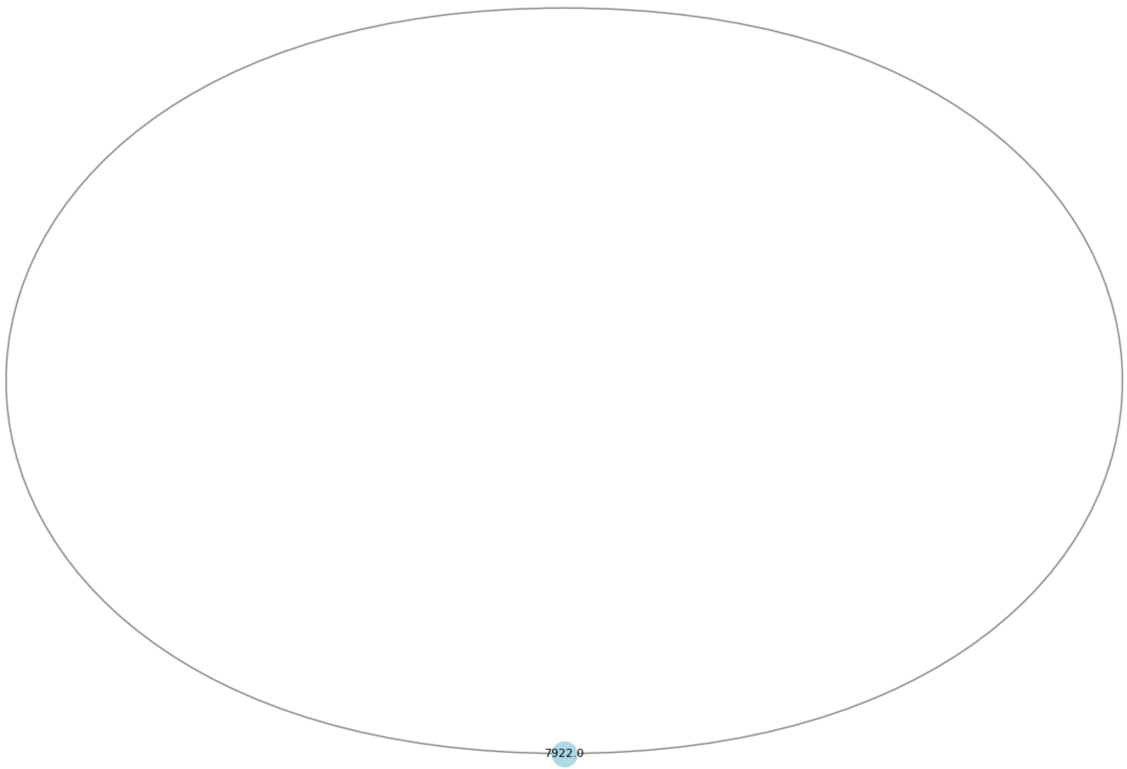
The graph starts at AS 7922 and transitions into AS 15169. The direct path between these two ASes suggests that once traceroute traffic reaches AS 7922, it is then handed off to AS 15169, which is associated with Google. It demonstrates a consistent routing pattern with a transition from the originating AS to Google's network.

www.microsoft.com


AS-Level Path Stability Visualization for microsoft.com

This graph displays a path beginning at AS 7922 and moving to AS 20940. There is only one transition between these ASes, indicating that the route to Microsoft services was quite stable over the observed period.

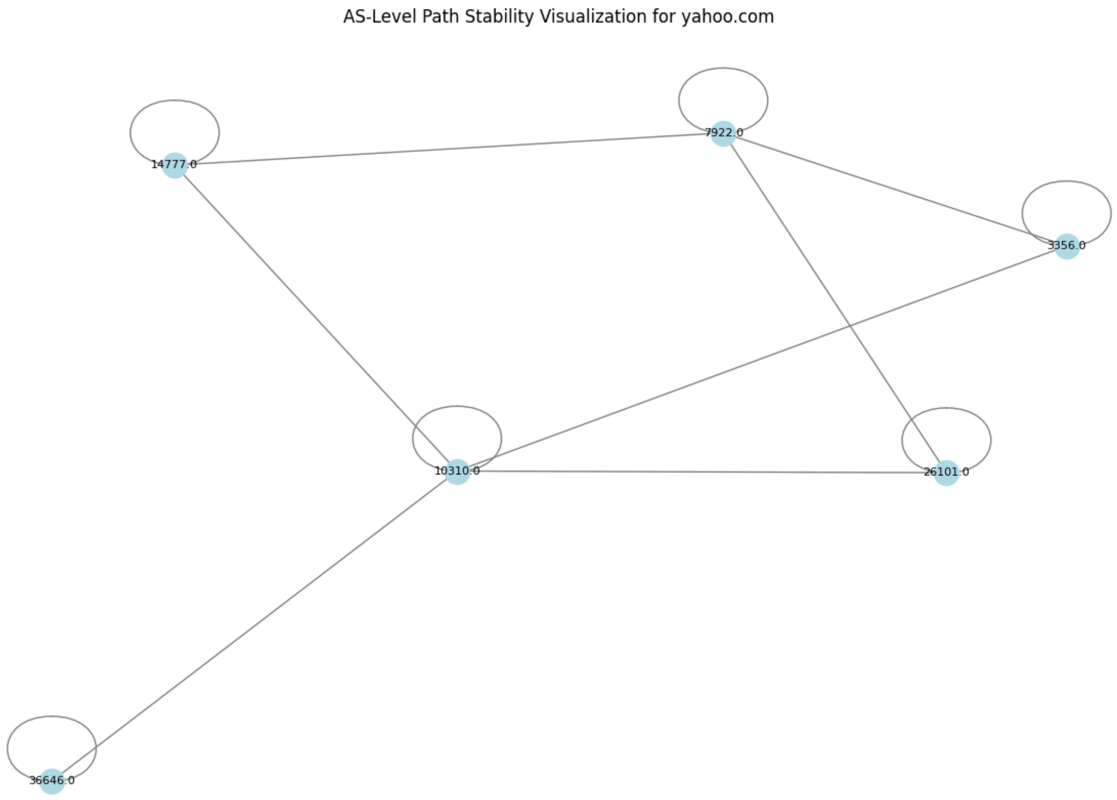
www.pitt.edu



AS-Level Path Stability Visualization for pitt.edu

The graph for Pitt shows a circular path entirely within AS 7922, indicating that all traceroutes to pitt.edu remained within the same AS. This implies that the university's network is either self-contained within a single AS or that the observed paths did not transition to other ASes.

www.yahoo.com



The graph for Yahoo is more complex, with multiple ASes involved (7922, 3356, 10310, 26101, and 36646). This complexity reflects multiple transitions and a more dynamic routing environment, which is typical for a global service provider like Yahoo.

C. Stability of the Delay

To analyze the stability of the delay, the following analysis has been performed.

1. **Per-Hop Delay Analysis**: For each URL and hop, the mean and standard deviation of the RTTs are calculated. The standard deviation here serves as a measure of delay variability: a higher standard deviation indicates more variability in delay for that hop.
2. **End-to-End Delay Analysis**: The end-to-end delay is calculated by summing the RTTs across all hops for each traceroute

measurement, then calculating the mean and standard deviation of these sums for each URL. This gives an overall view of the network's delay stability to each destination.

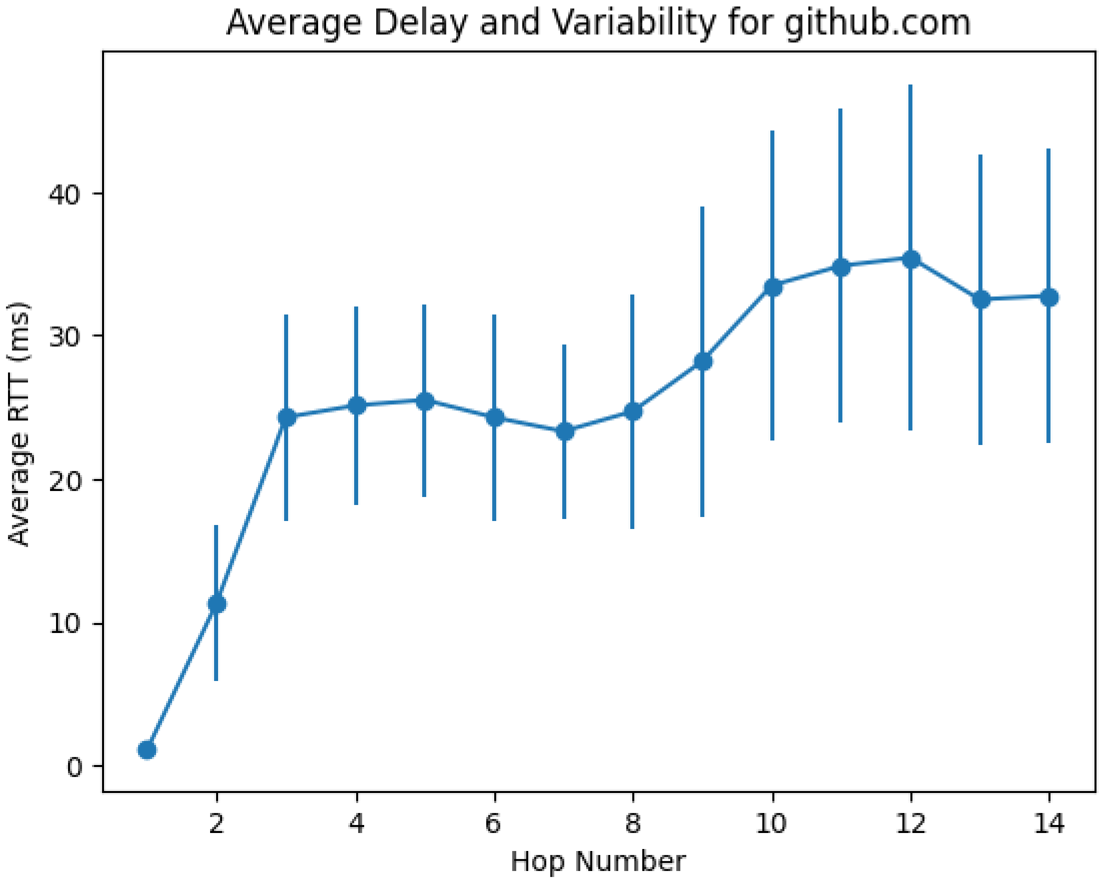
For each URL, plots have been plotted for the delay variability for each hop and the overall end-to-end delay stability. These visualizations will help identify which hops are more variable in terms of delay and how stable the end-to-end delay is to each destination.

Per-Hop Delay Analysis

The graphs below show the average delay and its variability for each hop along the traceroute path to different websites: GitHub, Google, Microsoft, Pitt, and Yahoo. Each graph displays the average round-trip time (RTT) for each hop along the path from the source to the destination (URL), with error bars representing the standard deviation of the RTT measurements at each hop.

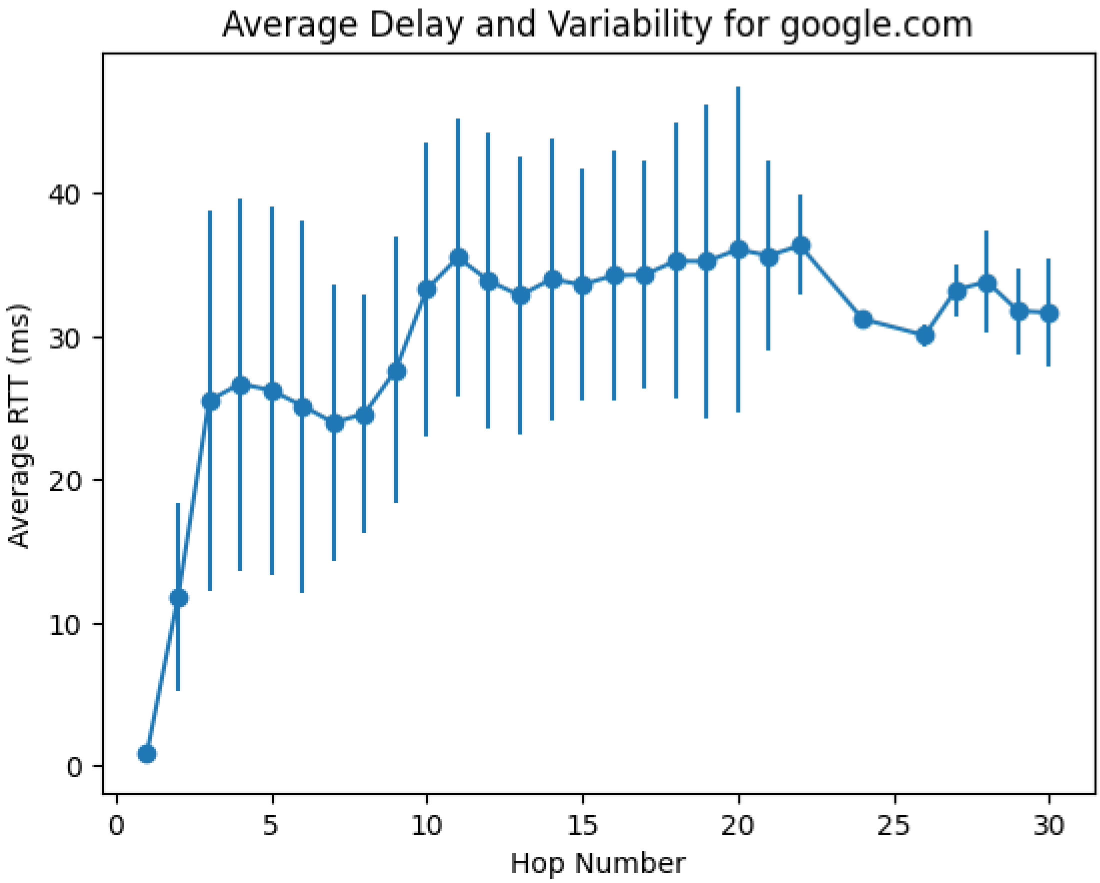
These graphs are useful for identifying potential issues in network performance and can help network engineers target specific hops for further investigation and optimization.

www.github.com



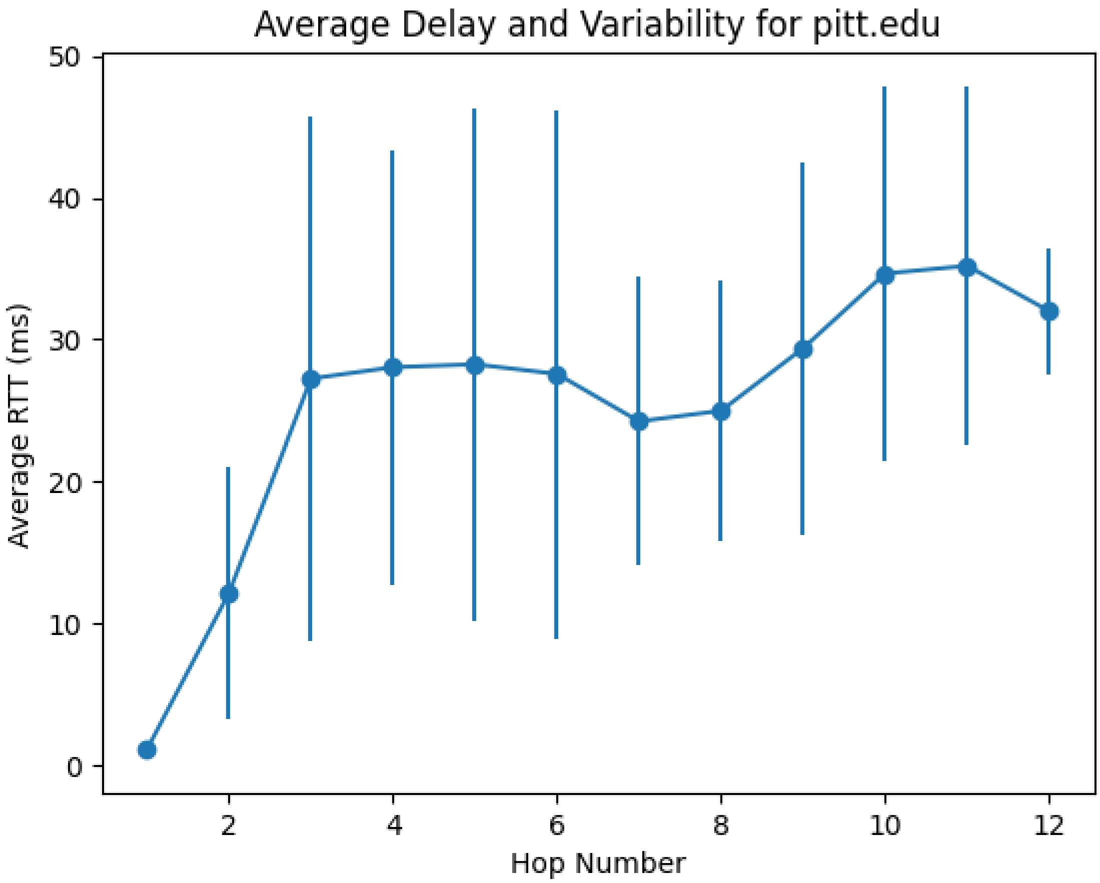
The graph for GitHub shows increasing RTT values as the hop number increases, which is expected as the packets travel further from the source. The variability (as indicated by the length of the error bars) also increases with the hop number, suggesting that later hops have more variation in delay times, possibly due to the increased complexity and number of network segments the packets traverse.

www.google.com



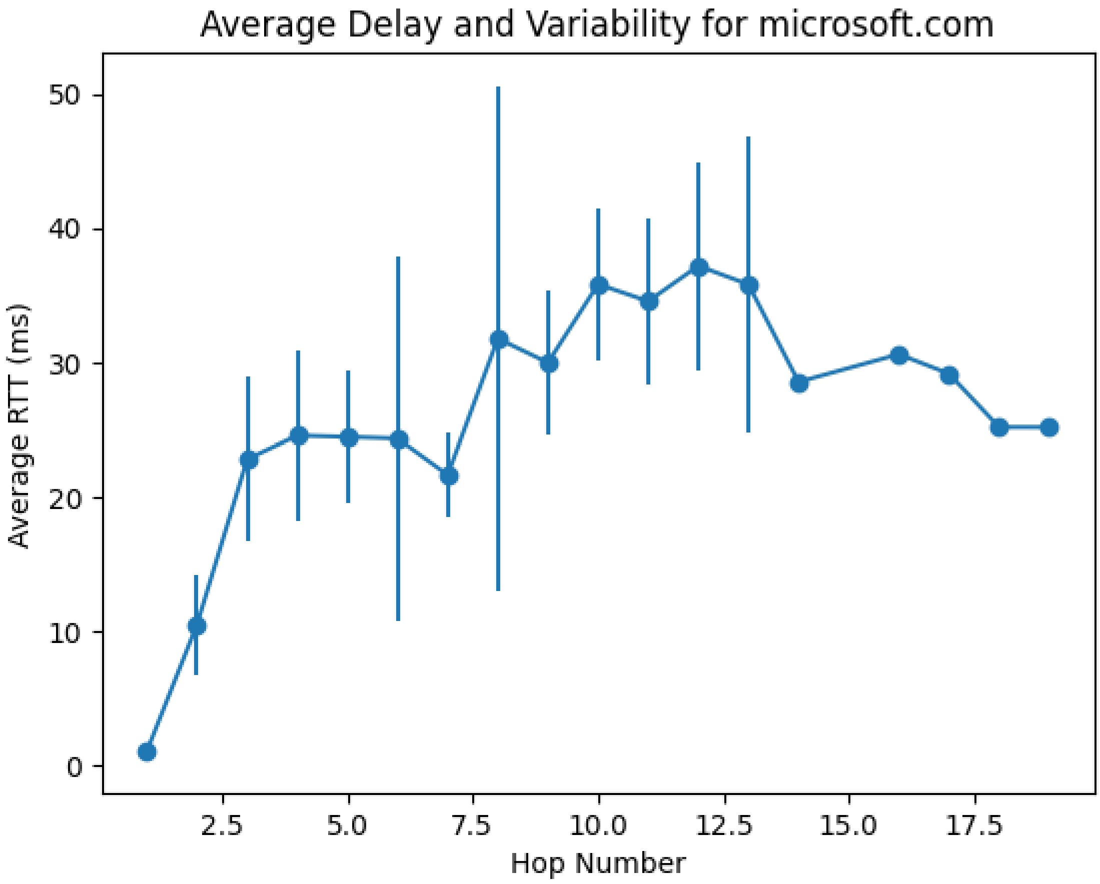
Google's graph has a similar pattern with RTT values increasing with hop numbers. However, the RTT appears to stabilize and slightly decrease after the initial increase. This could suggest efficient routing within Google's network. The variability is significant throughout, indicating that the path experiences fluctuations in delay, which could be due to routing changes or network congestion.

www.pitt.edu



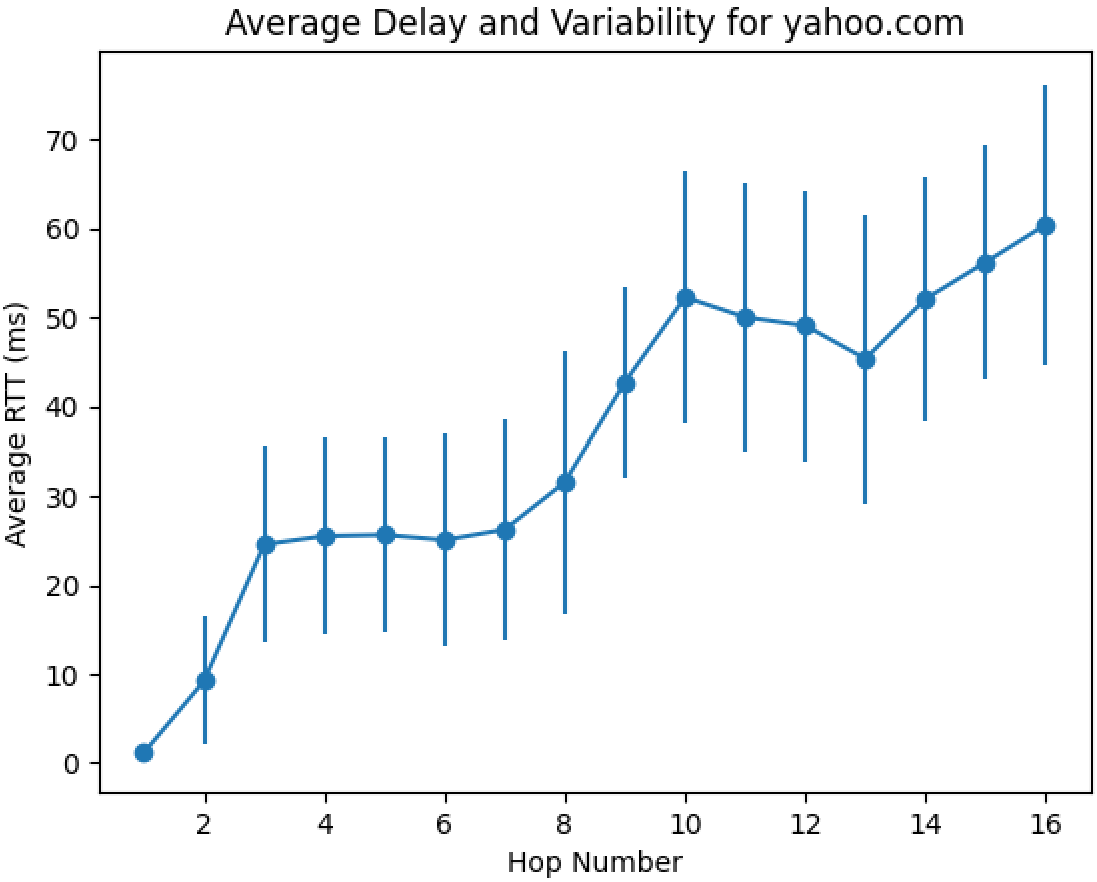
The graph for Pitt displays a gradual increase in RTT, followed by a plateau. The variability is somewhat consistent across hops but shows some spikes, indicating specific hops with more delay variation. This could be due to varying traffic conditions at those network points.

www.microsoft.com



Microsoft's graph shows a peak in RTT around the middle hops, followed by a general decrease. This could mean that the packets encounter a significant network delay or bottleneck at these middle hops, but then they reach a faster network segment or a direct peering link that reduces the overall RTT. Variability is somewhat consistent across hops, with some hops showing more variability than others.

www.yahoo.com



Yahoo's graph shows a steady increase in RTT across hops, with significant variability in later hops. This suggests that the path to Yahoo may go through multiple network transitions or peering points that introduce variability in delay.

General Observations

Increasing RTT with Hop Count: As packets travel further, the RTT generally increases due to the additional distance and number of devices they must traverse.

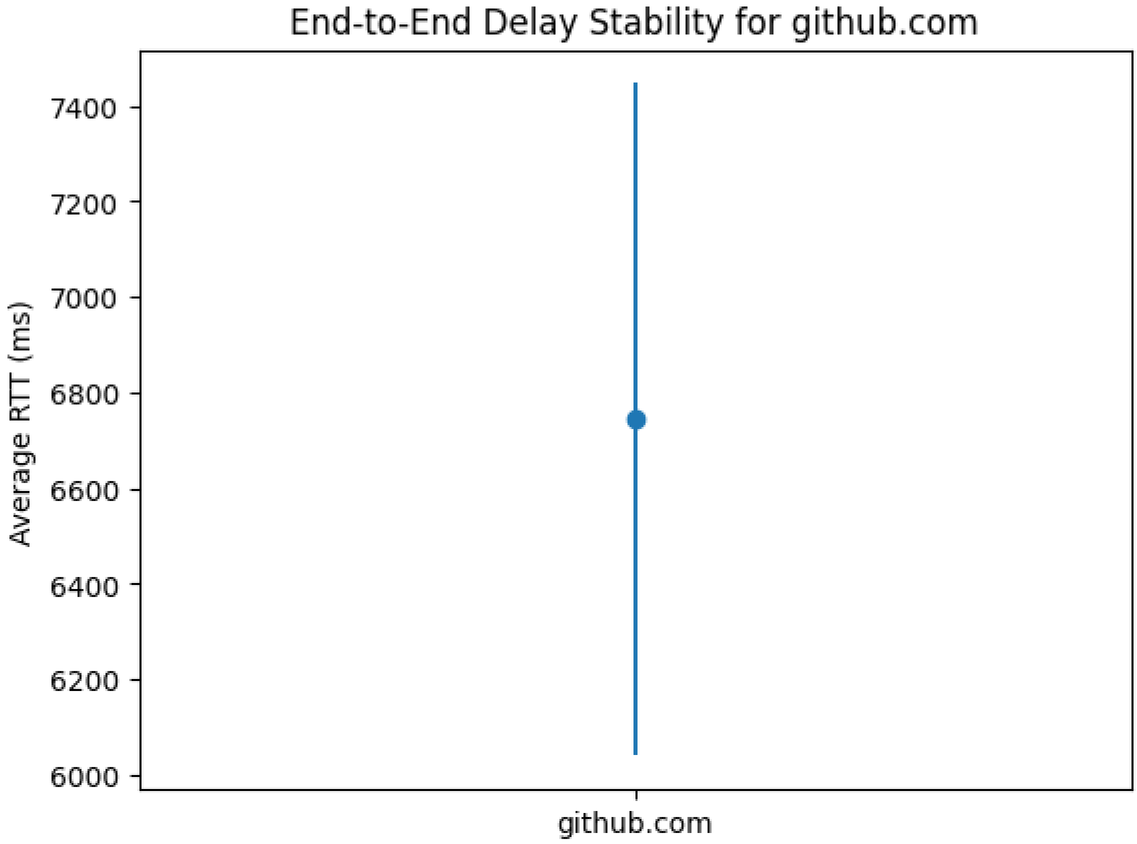
Variability in RTT (Standard Deviation): The length of the error bars indicates how much variation there is in the RTT measurements for each hop. Larger bars mean more variability, which could be caused by network congestion, dynamic routing, or changes in the path taken by packets.

Potential Bottlenecks: Hops with a significant increase in RTT could be points of congestion or bottlenecks in the network.
Network Efficiency: Stabilizing or decreasing RTT values in later hops could indicate efficient routing within a well-connected network segment or data center.

End-to-End Delay Analysis

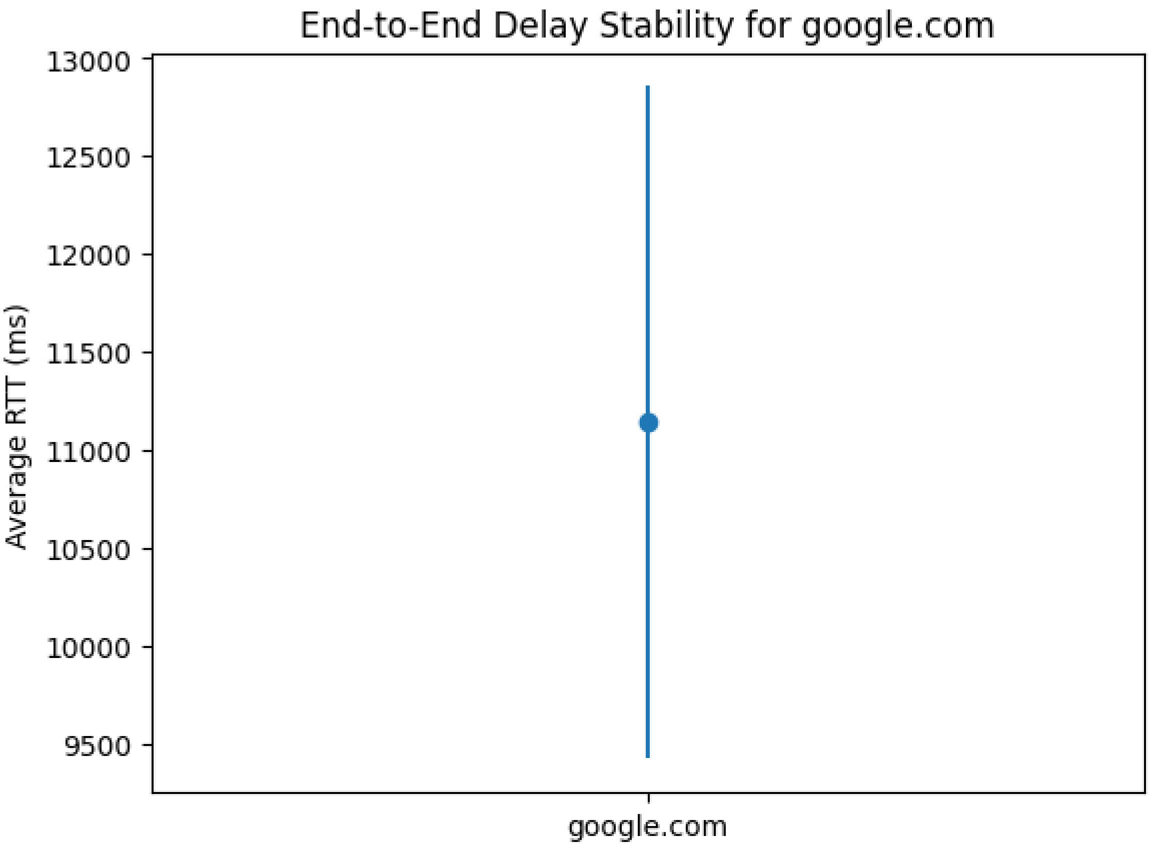
The plots plotted below represent the end-to-end delay stability for different websites. In these graphs, the blue dot indicates the mean (average) round-trip time (RTT) of the end-to-end path to the specified website, and the vertical line represents the range of RTT values, showing the variability of the measurements.

github.com



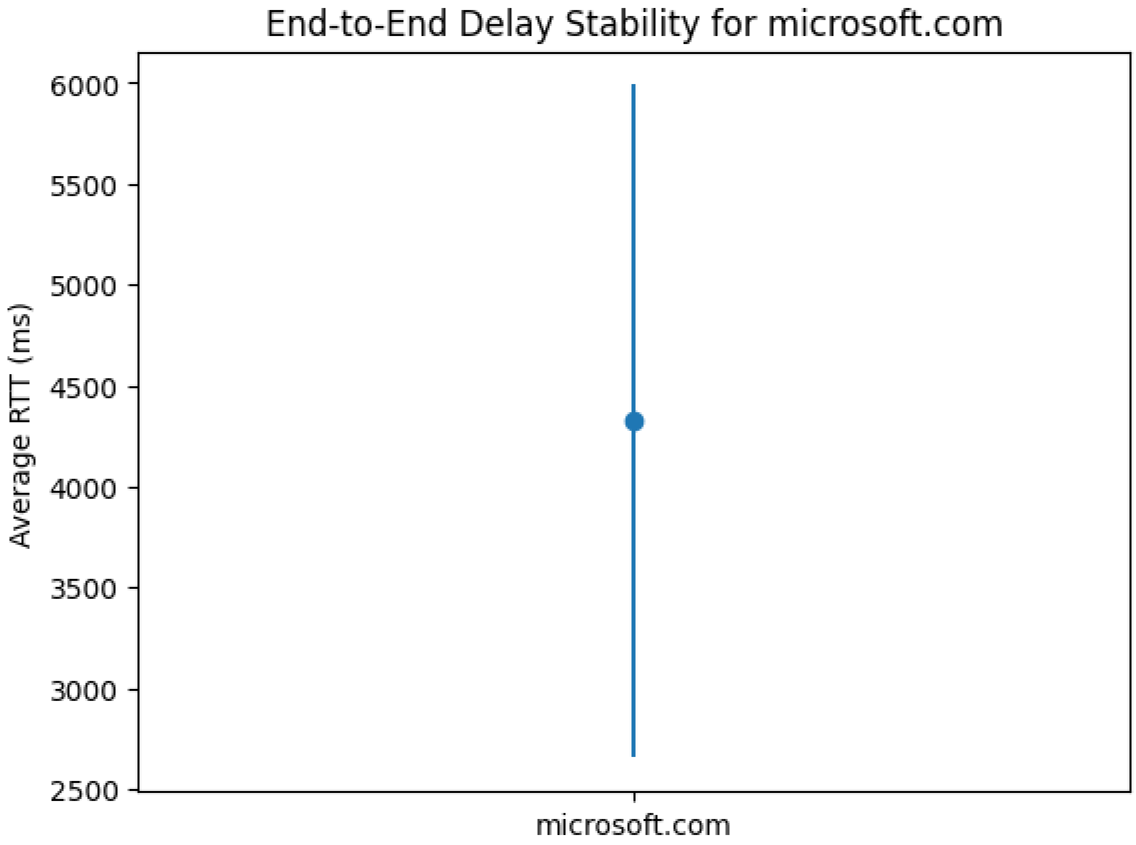
The average RTT is about 6700 ms, but the variability is high, with some measurements going as high as 7400 ms or as low as 6000 ms. This suggests that while the average is relatively stable, there are instances when the network path experiences significant delays or, conversely, lower than average delays.

www.google.com



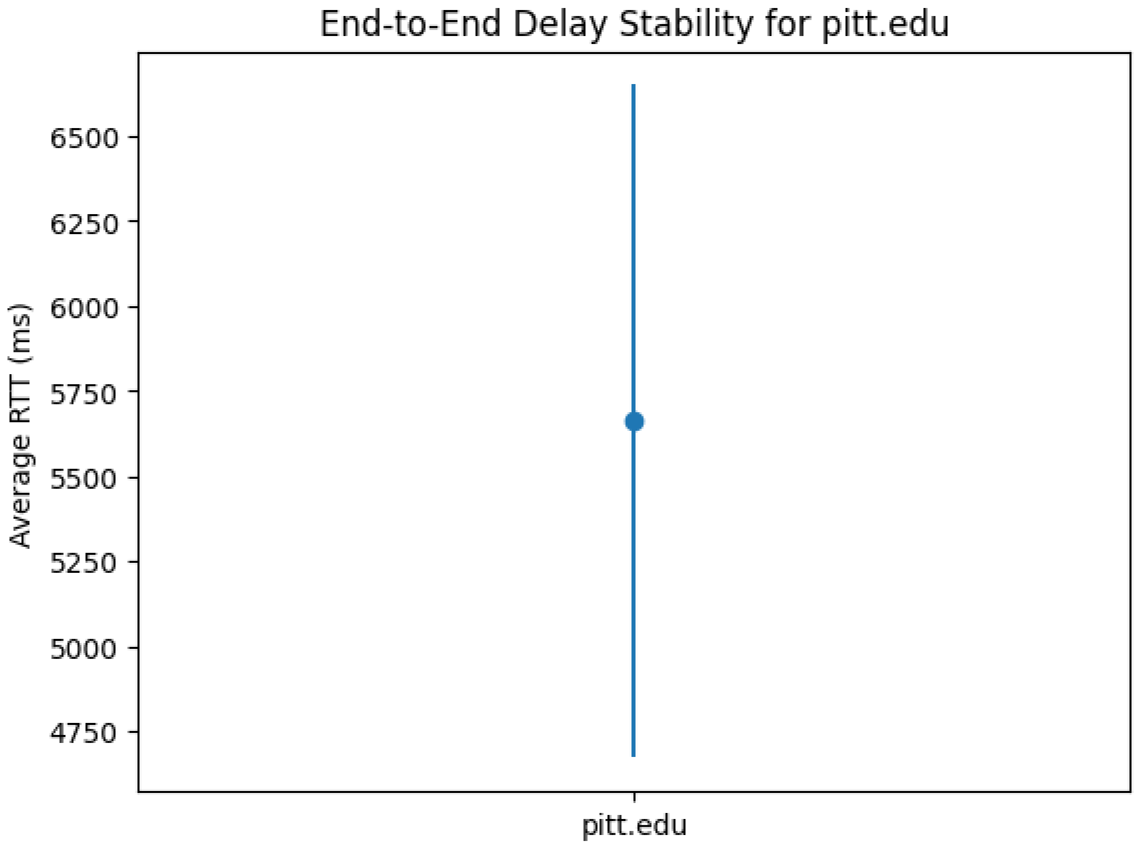
The average end-to-end RTT to Google is around 10500 ms with a wide variability, indicating that the network path to Google experiences fluctuating delays, which could be due to changing network conditions, routing changes, or other factors that affect network latency.

www.microsoft.com



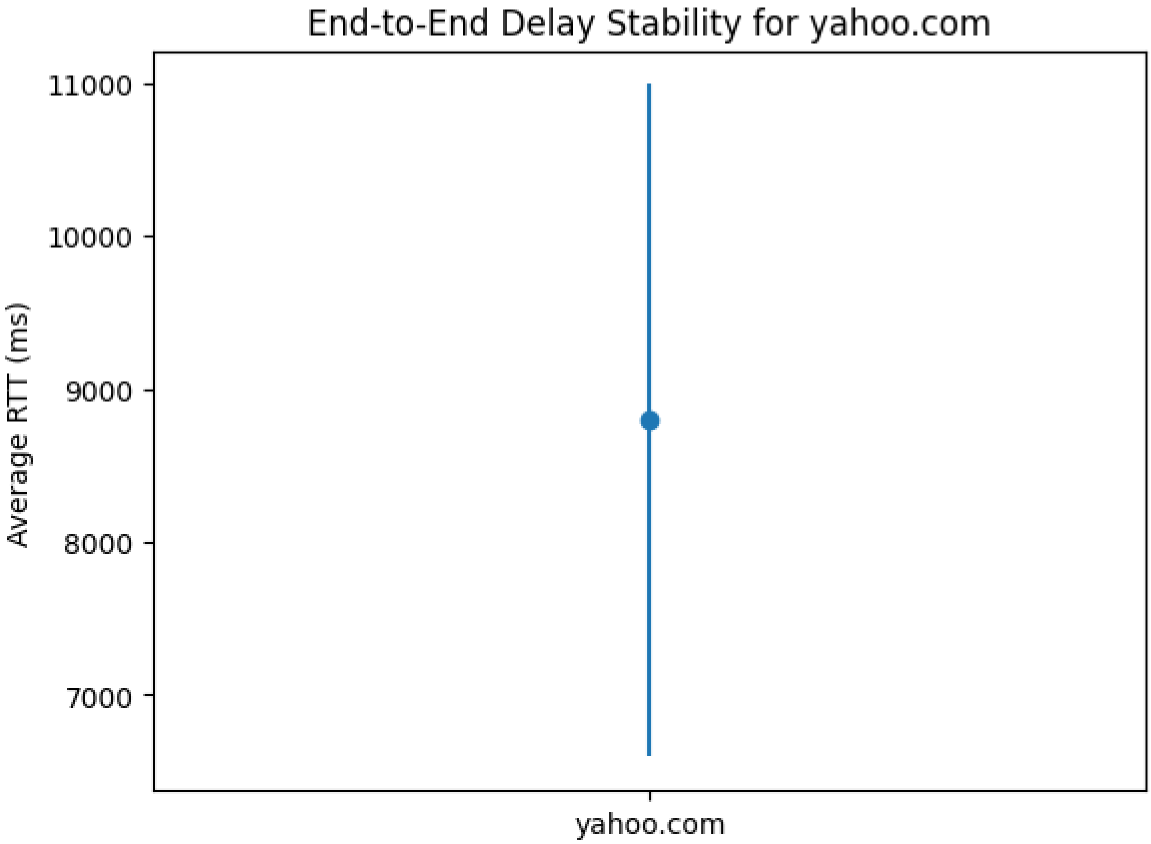
The average RTT appears to be around 4000 ms, with a variability indicating some fluctuations in the network path. The range is not as wide as seen with Google, suggesting a somewhat more stable connection but still with notable delay variations.

www.pitt.edu



The average RTT is around 5500 ms, with a large range of variability similar to that of Google. This high variability suggests that the path to Pitt.edu experiences significant changes in network delay.

www.yahoo.com



The average RTT for Yahoo is around 9000 ms, and the variability is quite extensive, implying that the network path to Yahoo can be highly inconsistent with how data is routed or how traffic is managed.

General Observation:

The presence of variability (as shown by the length of the lines) indicates that the network paths are not consistently providing the same performance, which could be influenced by multiple factors like traffic load, routing policies, or network configurations. These measurements are critical for network engineers and system administrators as they indicate not just how fast the connections are on average, but also how reliable and predictable the network performance is over time. It's important to aim for not only a low average delay but also a low variability to ensure a consistently good user experience.