Investigating Factors in Request Delay

Ben Smith

December 2024

Outline

Investigating Factors in Request Delay
Ben Smith (bxs566)

1 Introduction

The delay between a client and a server is affected by many factors, including physical distance, time of day, and network settings. The goal of my project is to measure how these factors impact the round-trip time (RTT) of a request to a server. In order to obtain a variety of measurements, so that the different factors can be accurately assessed, I measured differences in delay between the intranet and the Internet, and between local, regional, and international hosts, and measured at different times of day. Then I used tools to visualize the path each packet took to the server to determine what transpired between request and response.

Understanding the factors that impact delay in a network is important because it provides information on how requests propagate through networks to reach their destination servers. If a packet has a long or short RTT, it is indicative that something interesting could be happening within the network. If the RTT is long, it could mean that there is significant physical distance between the client and the server or that the server is poorly configured. If the RTT is short, it could mean that the network uses crafty tricks to speed up requests or that the server is very close. As a result, determining the cause of atypical request delays will reveal the inner workings of the networks through which a request traveled.

2 Procedure

2.1 Data Collection

My data collection process was fairly simple. I wrote a Python script to run a set number of ping requests to a specified destination and send the unprocessed output to a text file. This script simply runs the command

```
ping -c 1 [destination]
```

n times to send n single packets to the destination. After each ping, the script sleeps for the remaining time needed to satisfy the rate limit of five requests per second, taking into account the time the request took. I chose to only send one packet at a time because the existence of the rate limit ultimately determines the speed at which the test can run, so running 1000 requests will take 200 seconds regardless of how many packets I send at once.

2.2 Data Analysis Methods

To process the data from my ping script, I created another Python script that transforms the text output to comma-separated values format, with columns for domain, IP, bytes transmitted, packets transmitted, and RTT for each ping. After the fact, I edited the CSV files in Excel to determine total packets transmitted and average, maximum, minimum, and median RTT. I also used Excel for the creation of graphs and figures used in the report.

The domains I chose to ping were eecslab-10.case.edu, cwru.edu, google.com, and mail.ru. I chose these domains for variety of location, with eecslab-10 and cwru.edu being local, google.com being regional, and mail.ru being international. I also chose to measure during the middle of the day at around noon and later at night at close to midnight to capture delay under different network load levels since a

network should be more active during the day, which may have implications on request delay.
Other tools I used include traceroute, dig, iplocation.net, and curl. Traceroute was used to follow the path a packet takes to a server, dig was used to perform DNS lookups, iplocation.net was used to determine the physical location of IPs, and curl was used to view HTTP headers.

3 Results

3.1 Requests to Self

By far the lowest average delay in any set of requests were the requests made from eecslab-10 to itself. There was an average RTT of 0.0377 ms, a median of 0.038 ms, a maximum of 0.233 ms, and a minimum of 0.0130 ms. This is to be expected since the request does not have to travel anywhere, so any delay only comes from the resolution of the domain name and the server processing the request. This can be confirmed by viewing the route trace from eecslab-10 to itself, with there being only one hop: eecslab-10.EECS.CWRU.Edu. Although the domain name is slightly different than eecslab-10.case.edu, which is what I pinged, it shares the same IP address, indicating that the server is taking requests from both domains. Upon running dig on eecslab-10.case.edu, we find that this domain name is a canonical name record that points to eecslab-10.eecs.cwru.edu, so any requests to the former simply resolve to the latter.

3.2 Requests at Different Times of Day

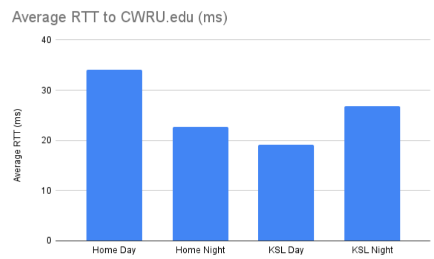

Figure 1: Comparison of RTTs

During the day, when server loads are higher, it is to be expected that a request will take longer to fulfill because a server can only process a limited number of requests at a time. To test this theory, I ran requests during the day and at night to get an idea of how RTT is affected by server load. Looking at my requests to cwru.edu from my house, I do observe increased RTT during the day. The daytime requests had an average RTT of 34.0 ms, while the nighttime requests had an average RTT of 22.7 ms. Interestingly, the minimum RTT for both times of day was roughly 14.5 ms, but there was a large difference in maximum RTT with 514 ms during the day and 179 ms at night.
My results also had counterexamples to the theory that RTT will be higher during the day. By observing my requests to cwru.edu from Kelvin Smith Library, we find that average RTT is significantly higher during the night. During the day average RTT was 19.2 ms, and at night average RTT was 26.8 ms. Although we know the CWRU server load is lower during the day from the requests from my house, that does not take into account load on the network. Since it is the week before finals,

KSL is overrun at night with people cramming for finals and rushing to finish projects, so load on the KSL network is much higher at night, explaining the higher RTT.

3.3 Requests to Different Geographical Locations

A major factor in RTT of a packet is the physical distance it has to travel to reach the target server. A packet traveling over a long distance must travel through many miles of wire and many intermediate networks, causing significant delay. By comparing differences in RTT of requests to geographical locations of varying distance, we can better understand these factors.

Looking at the requests to eecslab-10.case.edu, Google.com, and mail.ru from my house, we should see increasing average RTTs since each server is hosted farther from my house than the last. To determine server location, I input the IPs that returned pings into iplocation.net, which informed me that eecslab is hosted in Cleveland, Google is hosted in Chicago, and mail.ru is hosted in Moscow. The average RTT for eecslab was 46.5 ms, for Google was 34.7 ms, and for mail.ru was 178 ms. Mail.ru has a far higher average RTT than the other two, as expected, but Google had a lower average RTT than eecslab. The reason for this is because for 64 bytes of data, the few hundred extra miles to travel back and forth from Chicago to Cleveland does not make much of a difference, and with enough optimization that difference can be accounted for. A company like Google undoubtedly has made incredible optimizations to their servers to be able to handle millions of requests per second with the speed they do.

We can visualize the extensive path that packets take to mail.ru by inputting IPs from a route trace into iplocation.net. We see that the packet originates in Cleveland before bouncing around the US then passing by Amsterdam before stopping in Moscow. This path spans thousands of miles, which even with fiber-optic cables will take more than a few milliseconds to travel across. Compared to the path packets must take to the Google server in Chicago, the path to mail.ru is an order of magnitude longer, so it is no wonder the average RTT is significantly higher.

3.4 Requests Using a VPN

A virtual private network establishes a connection between a user and a server that allows the user to remotely make requests from the server using the server's IP address. Because the use of a VPN adds at least one stop in a request's path, it should increase RTT. To test this theory, I made requests to eecslab10 from KSL with and without connecting to a VPN. While connected to a VPN, average RTT was 10.8 ms, and while not connected it was 6.62 ms. I unfortunately did not capture a route trace from KSL to eecslab via VPN, but I was able to do so from my house. Rather than the request going straight from my house to the eecslab server, it takes two additional hops to intermediate IP addresses located in Columbus indicating a pass through a VPN, which both add delay to the request.

3.5 Redirected Requests

When collecting data, I did not realize that cwru.edu actually redirects to case.edu, but this provided me with an opportunity to observe the impact of redirects on RTT. I did notice while gathering data that although the average RTT for pinging cwru.edu from KSL was 19.2 ms, each ping command would take over a second to execute, which was intriguing. Using curl allows us to view the redirect happening. When cwru.edu receives a request, it returns a response header with code 301, meaning that the resource has been permanently moved to a new location, and there is a location entry with value https://case.edu/, meaning the resource has been moved to that location.

We can view effect of the redirect on RTT by comparing the average RTT of the requests from my hotspot to cwru.edu to that of case.edu. The requests to cwru.edu had an average RTT of 98.7 ms, while the requests to case.edu had an average RTT of 67.6 ms. Thus, we can tell that a redirect significantly increases RTT because the round-trip of the packet includes multiple requests and responses.