Austin Feydt

EECS 425

4/21/2016

Project 2 Report

**\*\*Preface\*\***

Most of my original 10 sites did not yield results, so most of these IPs are ones that were consistently successful

**Measurement Results:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **IP ADDRESS** | **RTT (ms)** | **# Hops** | **Geographical Distance (km)** | **Bytes of original datagram in ICMP response:** |
| 119.254.11.66 | 0.24730 | 26 | 10,780 | 548 |
| 104.124.54.16 | 0.28092 | 20 | 8,297 | 548 |
| 210.242.196.205 | 0.21801 | 20 | 13,143 | 28 |
| 125.212.216.123 | 0.26474 | 24 | 11,650 | 548 |
| 112.244.202 | 0.22699 | 21 | 12,382 | 548 |
| 219.238.238.114 | 0.24105 | 26 | 12,223 | 548 |
| 218.60.4542 | 0.29565 | 18 | 12,704 | 548 |
| 104.124.54.17 | 0.29309 | 22 | 8,467 | 548 |
| 125.6.149.67 | 0.17616 | 14 | 14,263 | 28 |
| 220.130.123.170 | 0.22939 | 20 | 13,142 | 28 |

**Most of the ICMP responses contained 548 bytes of the original datagram, while 3 of them contained the expected, standard 28 bytes.**

**Graphs:**

Correlation (via excel data analysis): **0.351657884663172**

Correlation**: -0.723759214584322**

Correlation**: -0.394725714949146**

Based on the graphs above, there seems to be a general positive correlation between RTT and Hops. In other words, you can expect a larger RTT if there are more hops (which logically makes sense, because more hops = more routers the packet has to go through = more times going through perhaps congested routers).

When looking at geographical distance, it does not seem to have as prevalent of a correlation with Hops or RTT. Both correlations are negative, which doesn’t seem quite right. In fact, the left graph suggests that RTT decreases when the geographical distance between two endpoints increases. There is not a very strong correlation between geographical distance and Hops either, and it is again negative, suggesting that as the distance increase, you should expect *less* hops, which again seems counterintuitive.

**Bold question answers:**

**How to match ICMP responses with probes sent out**:

To do this, I simply looked at the address from which the ICMP response came from, and compared it to the address I sent my UDP datagram to. If they were the same, I assumed that the response correctly matched the probe. This may not be a sure-fire way to determine it, but for a simplified version such as this script, it suffices.

**Possible reasons for not getting answer when probing arbitrary host:**

There are a bunch of reasons for this happening. For starters, I noticed that any big website, like Facebook, Google, Target, etc, never sent an ICMP response. Perhaps these large servers do not want to risk any sort of attack on the server, so they close off any port that isn’t port 80, thus ignoring an arbitrary packet sent to an arbitrary port. Therefore, I would never receive a response. On the other side, it is also possible that my machine blocked a certain port that got the ICMP response, so again, I would never receive it.

There is also the rare possibility that the datagram happened to get lost/ timed out every time I sent it. It could very well have been routed in an inefficient way, such that the TTL runs out too soon every time. Also, the datagram could get really corrupted, leading to it being unrecognizable by the time it reaches a server.