Life on the Digital Highway

Collin DeKorne, Darnel Cristobal, Lewis Shine, Tannar Mandak

Why This Project?

Description

Life on the Digital Highway will provide users with an inside look on how many network hops your packets take to get to their destination and use that data to relate it on a map using Google Maps

Motivation

We wanted to build a traceroute from scratch to provide us with a better understanding about network protocols and packets.

Problem

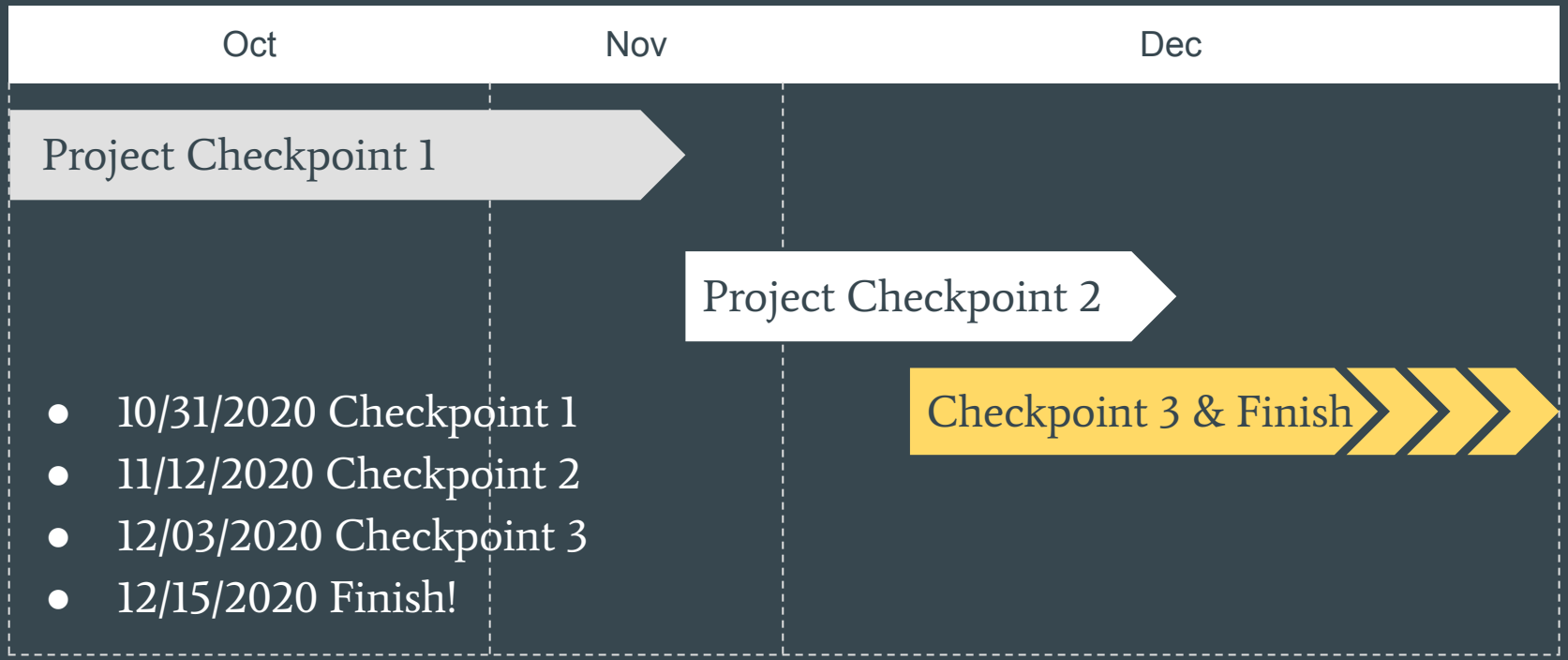
There are many traceroutes out there, however what if we wanted to know where are packets are bouncing around to?

Overview

For this program we used a variety of tools and modules. Our program is built off of the Python programming language and below are the modules we used.

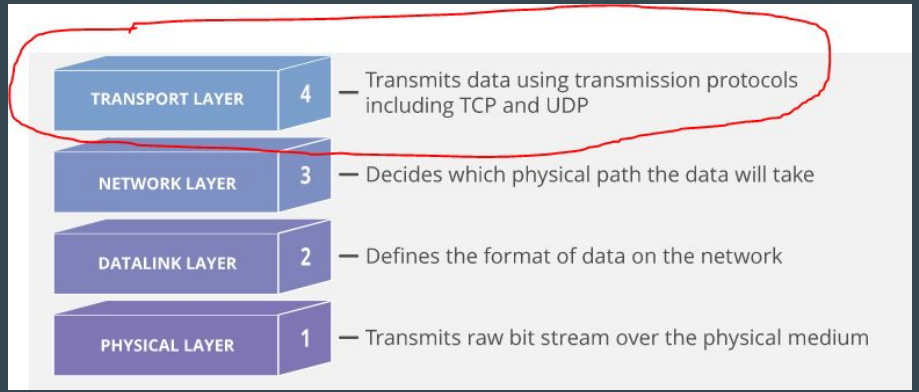
Project objective:
To design a traceroute and have each point mapped out on a map using Google Maps

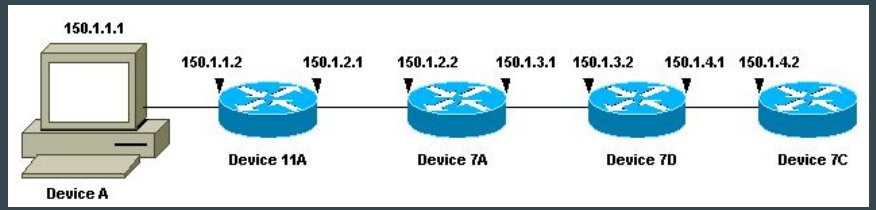
Timeline



- 10/31/2020 Checkpoint 1
- 11/12/2020 Checkpoint 2
- 12/03/2020 Checkpoint 3
- 12/15/2020 Finish!

Implementations





Traceroute

This part of the program is consolidated into 1 function. A request for a certain IP address/website is processed here and output for hops is provided.

Traceroute

- We used the Scapy module for python to implement the traceroute section
- Scapy is a packet creation tool where you can create packets and send it wherever you command it to go
- In our case, were creating packets, then sending them to the destination and then waiting to see if we get a response

```
pkt = IP(dst=hostname, ttl=i)/ ICMP()
```

```
Enter a website or IP: amazon.com
Destination: amazon.com
1 hops away:  192.168.0.1
2 hops away:  10.134.168.1
3 hops away:  100.120.109.34
4 hops away:  100.120.108.32
5 hops away:  68.1.2.121
6 hops away:  99.82.176.102
7 hops away:  150.222.206.207
8 hops away:  15.230.48.26
11 hops away:  150.222.243.116
16 hops away:  150.222.245.155
26 hops away:  52.93.28.78
```

Traceroute - Issues

- One of the issues we ran into was using the TCP protocol to ping servers leading us to have no response for some traceroutes
- After some more research ICMP packets are designed specifically for this purpose and fixed our issues
- We were also tweaking the timeout and retry values of the packets

```
Destination: google.com
1 hops away:  192.168.0.1
2 hops away:  10.145.0.1
3 hops away:  68.6.10.138
4 hops away:  100.120.108.2
5 hops away:  68.1.1.13
6 hops away:  72.215.224.173
```

Early prototype

```
reply = sr1(pkt, verbose=0, timeout=1, retry =-3)
```

Timeout times, retry times

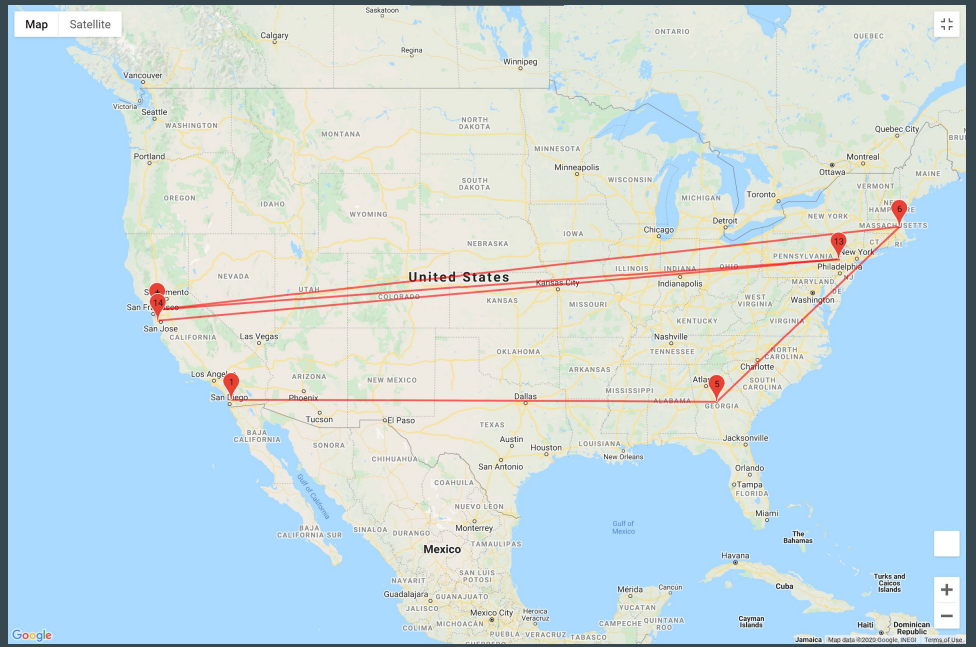
```
if(i > 28):
    print("Request timed out.")
```

Mapping

Mapping

APIs Used:

- Gmaps for Jupyter Notebook
  - Display the map with information
- IPstack
  - Checking validity of IP
- IPInfo
  - Gathering information from valid IP
  - More accurate when compared to IPStack when looking at locations



Connecting to google.com

Mapping - Issues

- Trying to display the map after compilation
- Missing information errors
- Compiling off of Jupyter Notebook (getting all information displayed)
- Wanted to make it look “prettier”

The END

Thank you for listening!