

Network Simulator (NS)

Advisor/Liaison: Dr. Ye

Team Leader: Andy Do

Team Members: Dibakar Barua,
William Fong, Y Hoang, Daniel Romo,
Zifan (Francis) Yang



Project Overview

- NS is a real world network simulation to evaluate the best strategy for resource allocation accounting for the failure rate in various network scenarios.

Missions:

- Back-End – run algorithm to evaluate resource
- Front-End – gui and user friendly and visualize everything

Team

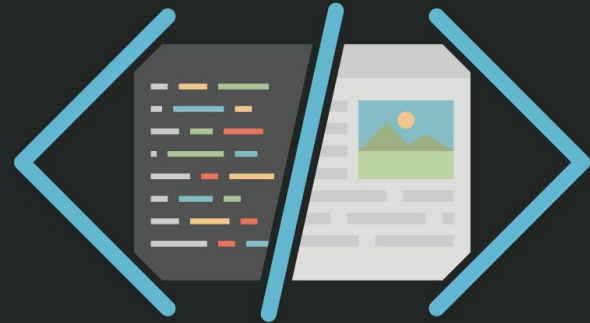
Back-End: Andy, William

- Implementing Various Algorithms
- Simulate Topology
- Implement Routing Methods
- Failure Rate Data



Front-End: Dibakar, Daniel, Y, Francis

- Build User Friendly GUI
- Node, Edge Implementation
- Google Maps Integration

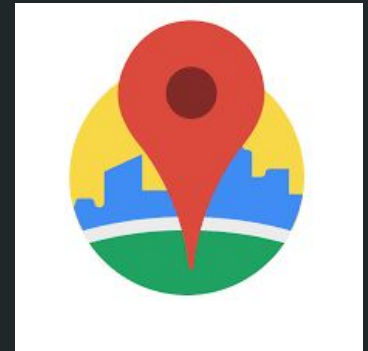


Development Challenges

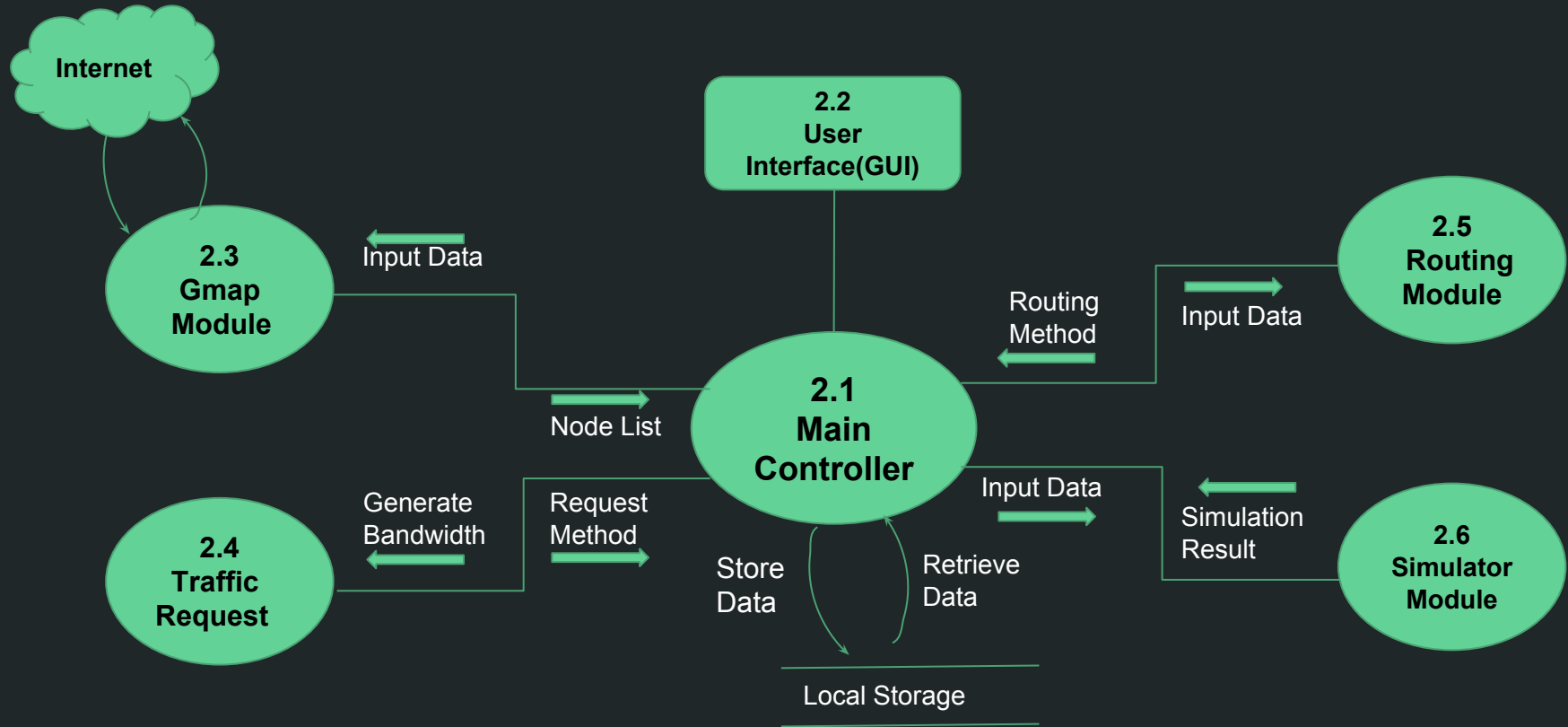
- For most of us, it is our first time working in a large group
- Merging complications with Github
- GmapsFX was complicated to get it working with the existing system
- Learning new API or Software was somewhat difficult for some of us
- Development hardware conflicts

Requirements/Technology

- JavaFX
- Scene Builder
- Fxml
- GmapsFX
- Github



Level 1 DFD Network Simulator



Simulator

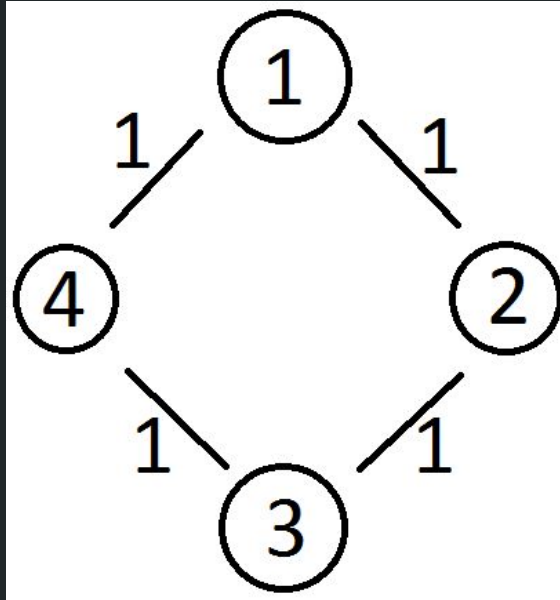
3 Inputs:

- Topology
- Traffic request method
- Routing method

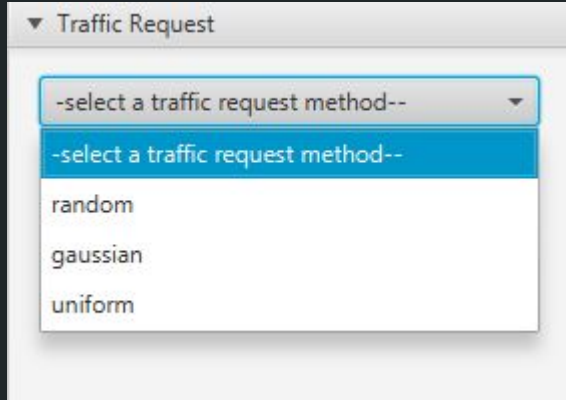
Topology

- Adjacency matrix

	1	2	3	4
1	0	1	0	1
2	1	0	1	0
3	0	1	0	1
4	1	0	1	0

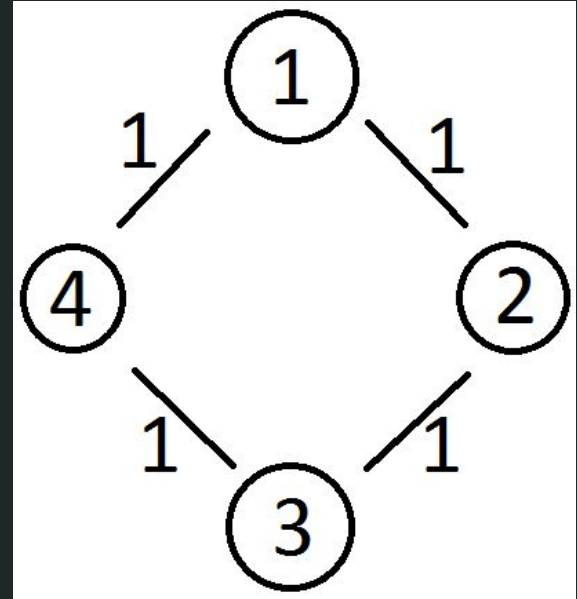


Traffic Request



Sample Request:

Starting Node:	Destination Node:	Bandwidth:
1	2	57



Routing Method

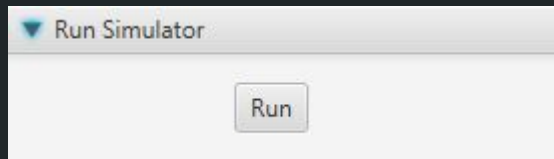
▼ Routing

--select a routing method-- ▼

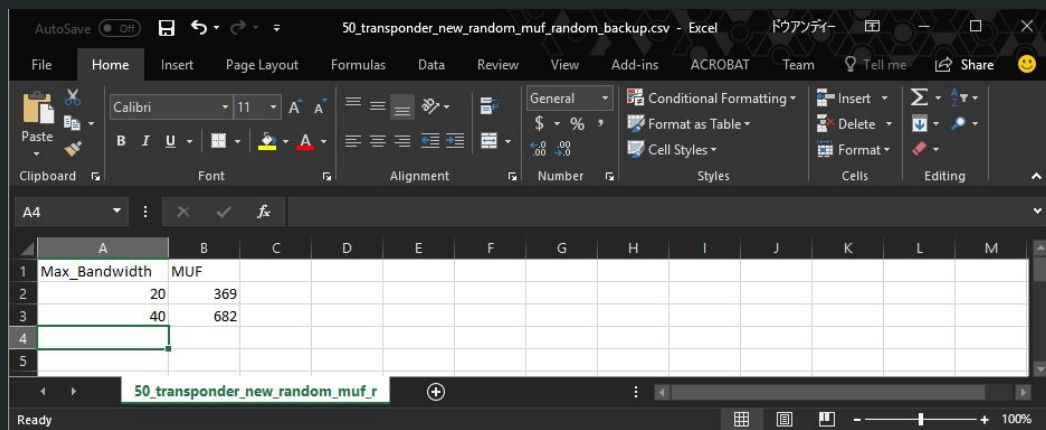
--select a routing method--

- SPF
- LUF
- MUF
- OPT
- MUX
- Hybrid

Running the Simulator



- Requests randomly generated
 - Random starting and ending node
- Output:
 - .csv file



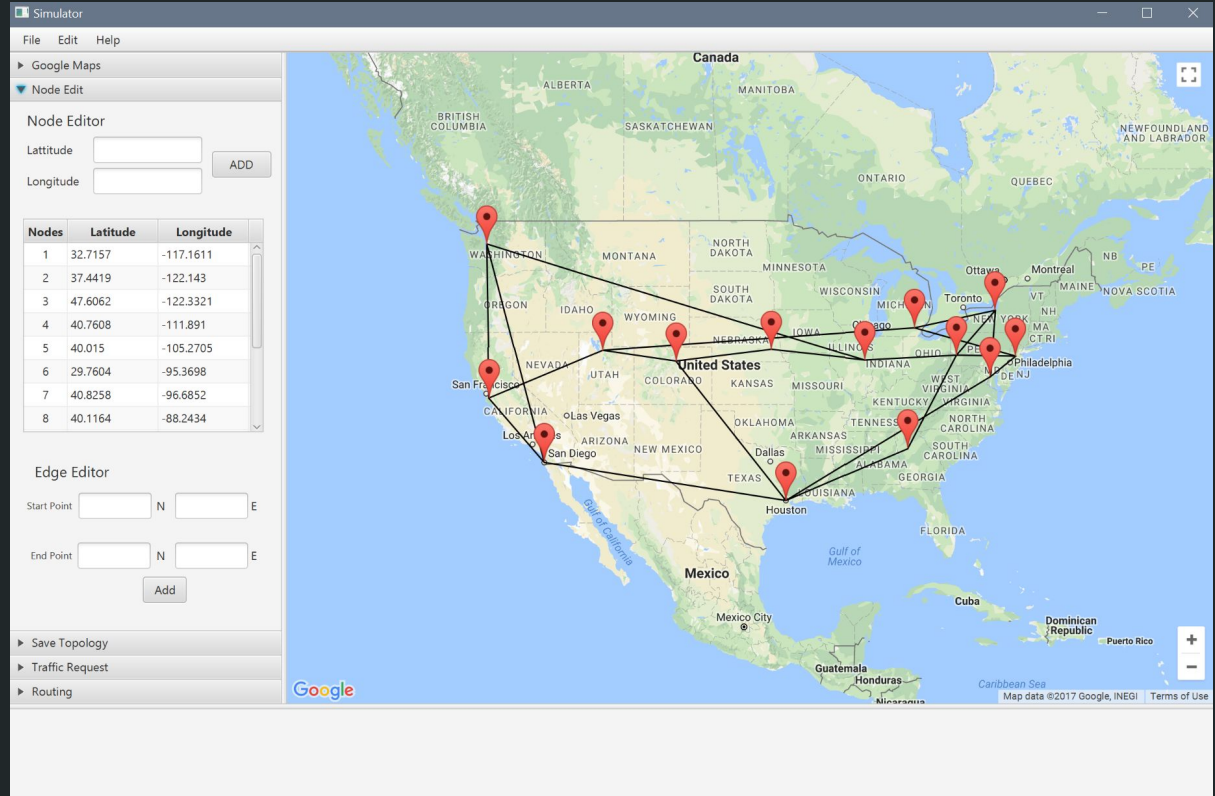
The image shows an Excel spreadsheet with the following data:

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Max_Bandwidth	MUF											
2		20	369										
3		40	682										
4													
5													

The spreadsheet is titled "50_transponder_new_random_muf_random_backup.csv" and the active cell is A4. The status bar at the bottom shows "Ready" and a zoom level of 100%.

GMAPSFX

- GmapsFX is the API we are using to show the Google Maps.
- Using NSFNET topology as default
- Use listeners to get user inputs
- Display and save data into files in order to read and write



Save and Add

- Using the left side of the window physically display the position of the node on the node edit title pin
 - Can add nodes and links manually
- In the save topology title pin user can see all user inputs and links between nodes

Simulator

File Edit Help

► Google Maps

▼ Node Edit

Node Editor

Latitude

Longitude

ADD

Nodes	Latitude	Longitude
1	32.7157	-117.1611
2	37.4419	-122.143
3	47.6062	-122.3321
4	40.7608	-111.891
5	40.015	-105.2705
6	29.7604	-95.3698
7	40.8258	-96.6852
8	40.1164	-88.2434

Edge Editor

Start Point N E

End Point N E

Add

► Save Topology

► Traffic Request

► Routing

Simulator

File Edit Help

► Google Maps

► Node Edit

▼ Save Topology

Node List

Nodes	Latitude	Longitude
1	32.7157	-117.1611
2	37.4419	-122.143
3	47.6062	-122.3321
4	40.7608	-111.891
5	40.015	-105.2705
6	29.7604	-95.3698
7	40.8258	-96.6852

Node Links

LatLng	Latlng	Distance
lat: 32.71570...	lat: 37.44190...	694.5841
lat: 32.71570...	lat: 47.60620...	1713.9693
lat: 47.60620...	lat: 37.44190...	1131.5901
lat: 40.76080...	lat: 37.44190...	958.8414
lat: 47.60620...	lat: 40.11640...	2834.3889
lat: 32.71570...	lat: 29.76040...	2096.1993
lat: 40.76080...	lat: 40.01500...	567.3115

Save

► Traffic Request

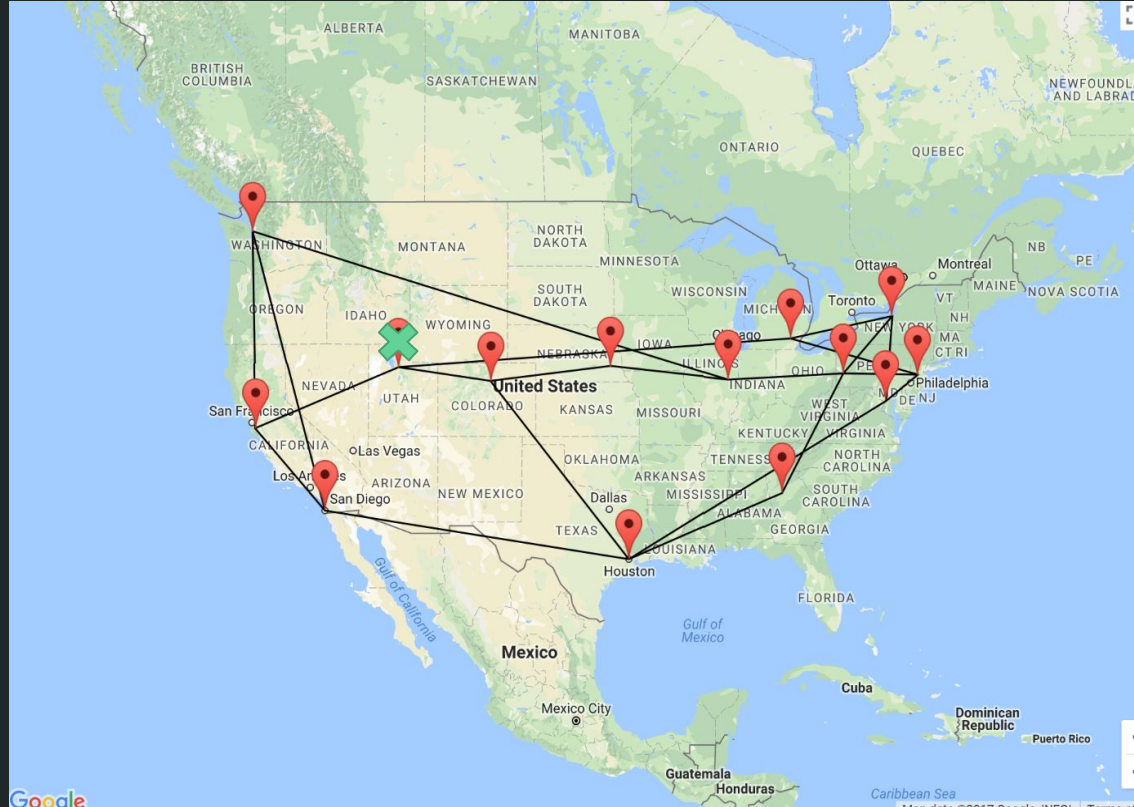
► Routing

Demo

Failure Model



Failure Model

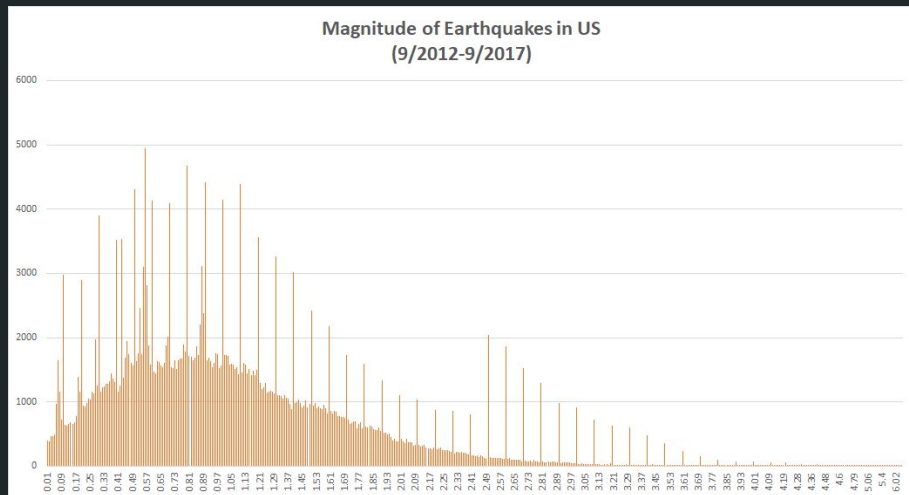


Failure Model

- Automatically reroute the traffic around downed node
- Data analysis on earthquakes
- Based off of 5 years of data in US
- Use past history to predict future

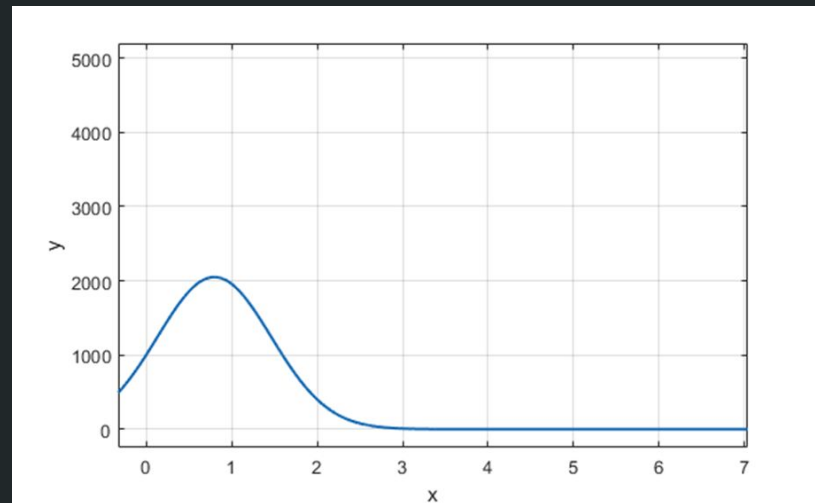


Failure Model



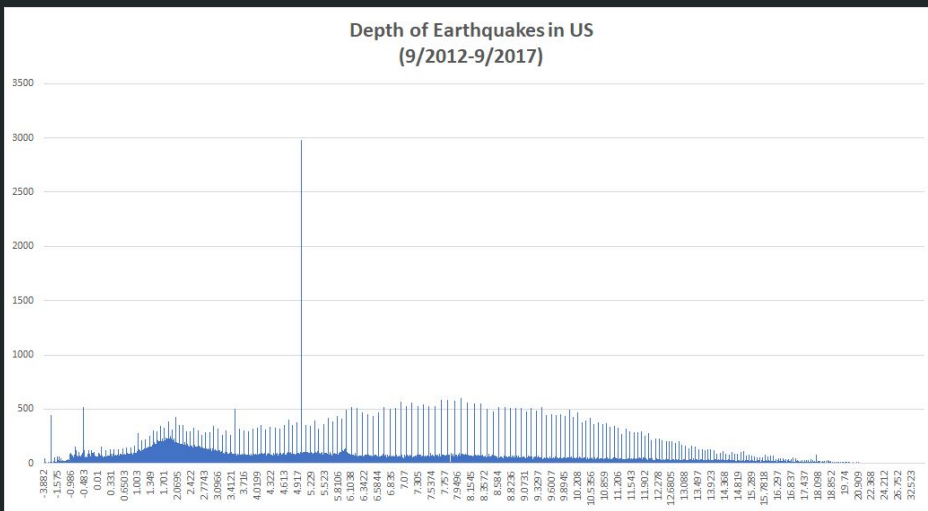
Distribution

Magnitude

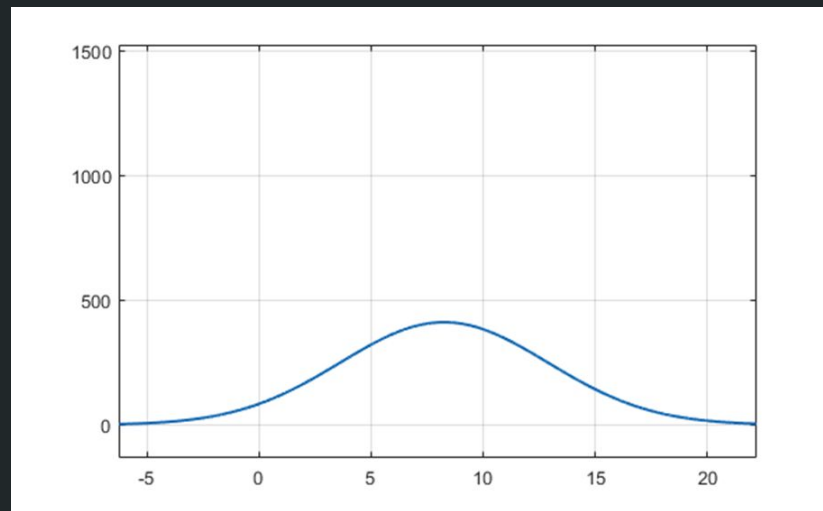


Interest Distribution

Failure Model



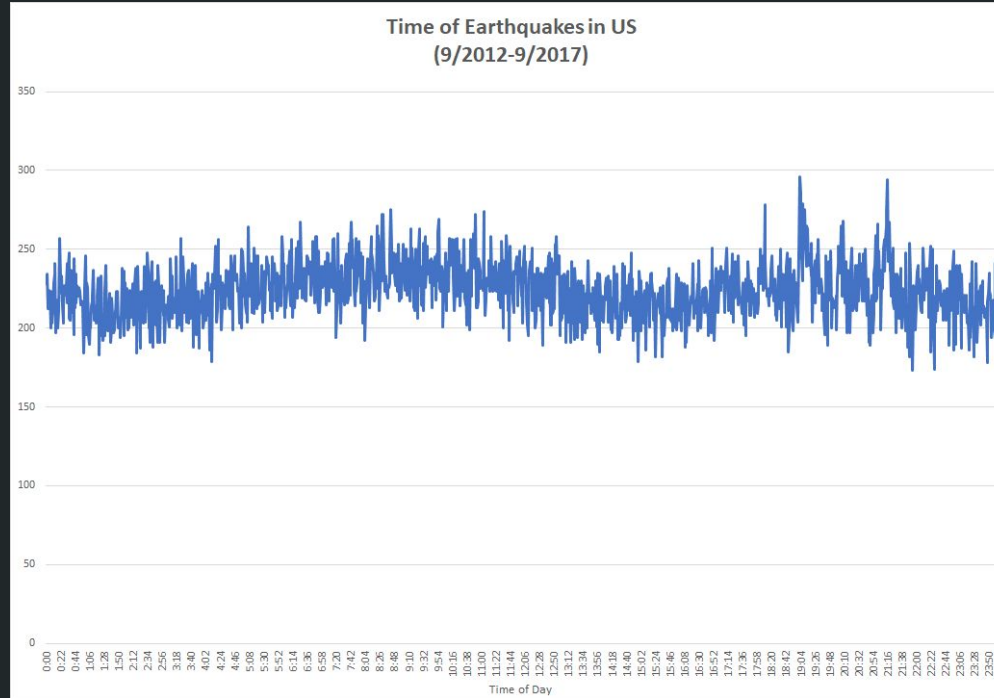
Distribution



Interest Distribution

Depth

Failure Model



Time
Distribution

What We Have Accomplished

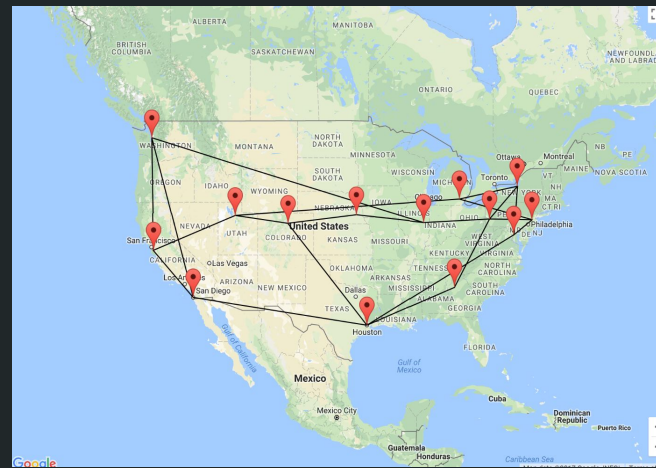
- Implemented routing algorithms
- Implemented a GUI
- Implemented Google Maps
 - API Calls
- Adding custom nodes

Hopes for the Future

- Complete the GUI for NSFNET
- Implement failure model
- Implement custom request

Custom Request

- Import a text file with all the start and end nodes
- Automatically process and display on map



Thank You

Any questions?