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

Internet

2.3 Gmap Module
- Input Data

2.4 Traffic Request
- Generate Bandwidth
- Request Method

2.1 Main Controller
- Node List
- Store Data
- Retrieve Data
- Local Storage

2.2 User Interface (GUI)

2.5 Routing Module
- Input Data
- Routing Method
- Simulation Result

2.6 Simulator Module
- Input Data
Simulator

3 Inputs:

- Topology
- Traffic request method
- Routing method
Topology

- Adjacency matrix
Traffic Request

Sample Request:
Starting Node: 1  
Destination Node: 2  
Bandwidth: 57
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
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
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

Depth of Earthquakes in US (9/2012-9/2017)

Distribution

Depth

Interest Distribution
Failure Model

Time of Earthquakes in US
(9/2012-9/2017)

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?