

Network Simulator 2

Max Mussalo

Christina Zhou

Kanerva Kotilainen

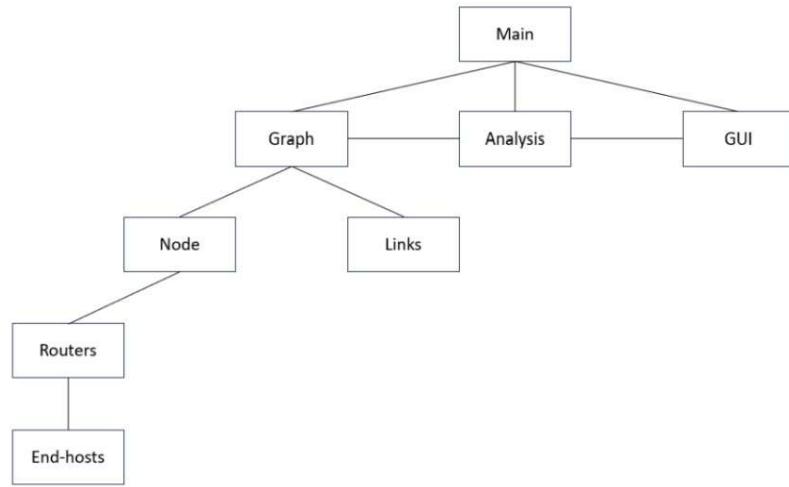
Jesper Riihola

Overview

The theme of the project is network simulators. We have implemented a discrete event simulator that models and analyses the data communication behavior in arbitrary complex networks which consists of nodes and links. Nodes can be either routers or end-hosts. Links are the connections between nodes and can have different transportation speeds. The maximum number of edges (links) in a network is $n(n-1)$ where n is the number of nodes. The user can create their own network by adding an arbitrary number of nodes and links to a file which is then read into the simulator. The simulator keeps track and calculates the number of sent, delivered, and dropped packets. In addition, it calculates statistical numbers such as mean and variance.

Software structure

The architecture and class diagram has stayed the same from the plan phase. We have a graph class which models the network and simulation of the network. The graph class consists of an arbitrary number of nodes and links between them. The nodes can be routers or endhosts. The analysis class acts as a middleman between the graph and GUI. The analysis class takes a graph as a parameter and from that class gathers and calculates the relevant data which the GUI then displays. The GUI class, as can be guessed, handles the graphical user interface and user interaction. As stated in the plan there is not any main class. The main box in the picture is more of an abstract concept of glue that ties everything together. The GUI class has instances of graph and analysis which it then uses so there does not exist any main class or box.



The GUI is made so that it subclasses Qt's QMainWindow and QWidget and adds relevant functionality to these. So, it is hard to define an interface between the GUI and the external library Qt.

Instructions for building and using the software

After the build files have been written by CMake the program can be compiled by running 'make' from the build directory. **Qt5 and Qt5Charts must be installed locally for the compilation to work.** To install the external dependencies in the terminal run the following command

```
'sudo apt-get install qtbase5-dev libqt5charts5-dev'.
```

To run the program the network must be loaded into the program first. This can be done from the menu (top left) "File" and "Import File". This prompts a window where the file can be selected. When the file has been read the simulation can be started, paused, and stopped from the toolbar (default location right under menu bar) and the GUI updates in real-time. In the center there are charts for the total utilization of the

network and the amount of sent vs dropped packets per node. Under that there are some calculated values which can be viewed.

Testing

Testing of the GUI was done “manually” i.e. by clicking and seeing if the correct stuff happens. No tests were written for the GUI. Nodes should always be initialized first, then packets and lastly the links. Testing of the simulation was done with a second main function main2 and using the vs code debugging tool.

Format for the read file

The files first line should be “#Network simulator parameters”. Otherwise, it is not recognized as the right file type. After that, create the nodes, then the packets and lastly the links. Nodes should be in the format of:

```
<Node type> <ID> <address> <maximum load> <behavior>
```

Similarly, links should be implemented in format:

```
<Link> <ID of the source node> <ID of the destination node> <speed> <maximum load> <delay>
```