Overview

This documents covers the things that don’t deserve their own document

Network simulator’s main components are its classes, derived from abstract parent class Node and class Packet, services main_map and DNS and also the graphical user interface. The network is downloaded from two text files that parse the data to the correct form

Packet objects are transmitted between these links following (some of) the rules of real networks, such as resolving “real” addresses from server names, finding fastest paths between nodes, and dropping packets when nodes are full

Installing / user guide

See project README.md

Scope

- All messages are sent in one packet, i.e. no limit on packet size
- No TCP-style congestion control or reliability
- No TCP/IP-style data encapsulation

Features

- Load a network from text files
- Establish routing tables to find lowest cost routes
- Send packets between nodes
- Packets might be dropped by nodes if maximum capacity has/would be reached
- Change node capacity mid-execution
- Waiting propagation and transmission delays
- Print useful information about nodes to observe network behaviour
- QT GUI with animations

Services

See short_technical_pp.pptx last two slides

Classes

See classes.pdf

testnodes and testlinks.txt

The network is downloaded from two files after selecting them in the GUI (see user guide), from where the objects are created. The files are parsed following this format:

testnodes.txt:
node_type;id;max_capacity;connection_cost_bandwidth;conncection_cost_latency;name;content_type;content_size;
Router does not have the last three members, and the client the last two, while the server has all of them, but thanks to the node_type in the beginning, the parsing function knows what to do. The objects are created based on these values, and if you make a mistake in writing these files, unexpected program behaviour can occur

testnodes.txt:
first_id;bandwidth;latency;second:id;
This defines the link speeds between routers, and they are updated as they are parsed. All links are undirected

General thing about the source code

- All data concerning bits and bytes is stored in Kb/B, but sometimes in printing they are converted to Mb/B when it is more appropriate (e.g. links bandwidth)
- There are differences in the src/ and gui/ code since some things can't be done the same way in QT as is done in vanilla C++
- The final version of source code is in gui/ directory