Project 2

Due: 9 & 23 November 2021

For this project, you will implement a simulation and perform an analysis on parts of the paper

> Ratnasamy et al. A scalable content-addressable network. In *Proceedings of SIGCOMM'01*, 2001.

as discussed in class.

Objectives

In the course of this project, the successful student will:

- Read and distill the ideas from a published conference paper
- Write a program implementing an algorithm system from an academic specification
- Analyse the results and performance of the program

Requirements and restrictions

This work can be somewhat collaborative in that you can talk to each other about design and help with debugging, but the program you write should be your own. (See my collaboration policy for more details.)

The program you write for the simulation can be in any language(s) but does need to run on the machines in the lab.

The space you will design for should be 2-dimensional with integer coordinates in the range $[0, 2^{16})$.

Because we're simulating the network-related aspects of this, you don't need to model the *values* that are stored, just their key (or address), which will be a string.

In the course of its run, your simulation will generate "enough" nodes in the system and "enough" data to store at them to effectively demonstrate the

working of the system. Some components of your program will represent nodes in the network that only know certain things; at least one component of your system should have a sort of gods-eye view so that you can report on and analyse the performance.

Design work

You should have a draft idea of what your design looks like, in time for class on Tuesday the 9th. Some things you should be thinking about:

- what the data structures look like, both in terms of what they store/"know" and how they talk to each other
- how the controller program holds everything (i.e. what-all data will there be?)
- what the controller program will do (and what output it will generate) in order to illustrate the system's performance

Bring that with you (either on paper or in a document on a laptop you have with you). We'll be talking about the project in class that day.

Parts to implement

The paper has a lot of details and ideas in it. You won't be implementing all of them! As of right now, the following areas seem most important to implement in order to illustrate and analyse the paper, particularly with respect to the network-theory ideas we've been discussing this semester:

Building the network: nodes join

Routing: someone accesses data (lookup or update) in the network

Robustness: nodes intentionally leave, and nodes unintentionally disappear; and queries occur after this point

Those are all in Section 2 of the paper. Some additional things (like the bootstrapping stuff) there and in section 3 also seem interesting and maybe important, though. We might renegotiate the above list after we've talked about the paper in class on the 4th and 9th.

Receivables

In the end, you'll produce and hand in at least four things:

- A transcript/output/results from one or more successful program runs. This might be in multiple files, and if so, they should be named and organised effectively so I can look at them.
- An analysis of what worked well and didn't work about the paper's described system. This analysis can incidentally mention problems *you* had with implementing *but that is not the point of this analysis*. You should be talking about the paper's ideas here.
- The program.
- Instructions on how to build and run the program in such a way as to generate something similar to the submitted results. ("Similar to", because randomness in the program will mean that each run is presumably somewhat different from the next.)

Hand all of that in with the handin script, as `proj2`, by 23 November.