CS/CE/TE 6378: Advanced Operating Systems
Section 002
Project 1

Instructor: Neeraj Mittal

Assigned on: Wednesday, January 18, 2017
Due date: Wednesday February 8, 2017

This is an individual project. *Code sharing among students is strictly prohibited and will result in disciplinary action being taken.*

You can do this project in C, C++ or Java. Each student is expected to demonstrate the operation of this project to the instructor or the TA. Since the project involves socket programming, you can only use machines dcXX.utdallas.edu, where XX $\in$ {01, 02, .., 45}, for running the program. Although you may develop the project on any platform, the demonstration has to be on dcXX machines; otherwise, you will be assessed a penalty of 20%.

1 Project Description

Create a distributed system consisting of n nodes arranged in a certain topology specified in a configuration file. Design and implement a distributed algorithm that enables each node to discover the entire network and identify its k-hop neighbors for each k. Your algorithm should eventually terminate at each node after which a node will output k-hop neighbors for each k.

2 Submission Information

All the submission will be through eLearning. Submit all the source files necessary to compile the program and run it. Also, submit a README file that contains instructions to compile and run your program.

3 Configuration Format

Your program should run using a configuration file in the following format:

The configuration file will be a plain-text formatted file no more than 100KB in size. Only lines which begin with an unsigned integer are considered to be valid. Lines which are not valid should be ignored. The configuration file will contain $2n + 1$ valid lines.

The first valid line of the configuration file contains one token denoting the number of nodes in the system.

After the first valid line, the next n lines consist of three tokens each. The first token is the node ID. The second token is the host-name of the machine on which the node runs. The third token is the port on which the node listens for incoming connections.

Finally, the next n lines consist of the up to n tokens each. The first token is the node ID. The next up to $n-1$ tokens denote the IDs of the direct neighbors (or 1-hop neighbors) of the node referred to by the first token.

Your parser should be written so as to be robust concerning leading and trailing white space or extra lines at the beginning or end of file, as well as interleaved with valid lines. The `#` character will denote a comment. On any valid line, any characters after a `#` character should be ignored.

You are responsible for ensuring that your program runs correctly when given a valid configuration file. Make no additional assumptions concerning the configuration format. If you have any questions about the configuration format, please ask the TA.

Listing 1: Example configuration file

```
5

0 dc02.utdallas.edu 1234
1 dc03.utdallas.edu 1233
2 dc04.utdallas.edu 1233
3 dc05.utdallas.edu 1232
4 dc06.utdallas.edu 1233

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