CS 542 Project Description

Professor: Dr. Michael Y. Choi

TA: Viswatej Kasapu

Introduction

CS542 project is to simulate Link-state routing algorithm. Completing the project requires at least appropriate CS400 level C/C++, Python or JAVA programming skills under Windows 10/ 8 / 7 / XP OS GUI platform (like at Main/Rice campus PC lab) or command prompt mode (former MS-DOS or command line mode of Windows platform). If you want to work it under a different platform such as MAC, Linux OS or UNIX OS systems, please get permission from your TA before you start. If you want to choose different programming languages such as MS Visual Basic or others, please also get permission from your TA in advance before you start.

Your project package MUST provide the executable file in case of using a compiler. If you select to use Java, compiled Java bytecodes must be included in the submission package and specify the detail instructions how to run your project or include JAR executable file. TAs will not attempt to compile your program.

Basic requirement: You have to do your own work to prove what you have learned from the class.

Academic Honesty

Do not share your work with other peers. Do not attempt to use either old project or a similar one available from the Internet. Your TAs have special tools to cross-check potential cheating even with all history of similar works. If your TA finds a situation, both provider and requestor names or someone using online resource will be placed on department chair and dean’s desk. Once your name is sent, the professor cannot help the situation. The situation will be worse than the failure of this course.

** Warning: if you use any similar source from online, you must understand if anyone may use the same source.
** DO NOT use (or get help from) any resource (algorithm or source codes) from the Internet. There are sources you may get from the Internet but if you do, your work will be considered as a violation of academic honesty. **Your project must be done by using your brain, lecture notes and textbook only.**

This project can be done by an individual or a group of members up to 2 with different workload and requirements. A team project must be implemented by GUI. If any similar parts in the codes, customized program behavior, or any are founded, the work will be treated as cheating. It will cause failure of this course for both the helper (original work) and the requestor (duplicated or modified work). **No excuse, no exception.**

Objective

In this semester project, you are asked to develop a simulator to implement Link-State Routing Protocol. Your program should have two functions:

1) Simulate the process of generating forward table for each router in a given network,
2) Compute optimal path with least cost between any two specific routers.

   * A team project will require more functions from your TA when you report.

Problem Description

Suppose we have a network with an arbitrary number of routers. The network topology is given by a matrix, called the original topology (graph) matrix, which only indicates the costs of links between all directly connected routers. We assume each router only knows its own information and has no knowledge about others at the beginning.

In this project, to implement Link-State Routing Protocol, first, your program is required to create the state of the links by each router after the input file containing the network information been loaded. By reading the topology matrix file a network graph can be determined. A ***Dijkstra's algorithm*** could be applied to find the shortest path between two entities: source and destination nodes. Finally, your program should be able to output the connection table of any router and output the optimal path between any two selected routers.

Sample Input

1) Sample Network Topology Diagram:
   - Each link between the two nodes may have different costs by direction (means your project should work any different topology/matrix)

   e.g) Cost from R1 to R3 is 5 and from R3 to R1 is 3 respectively
   Cost between R1 and R4: both 1

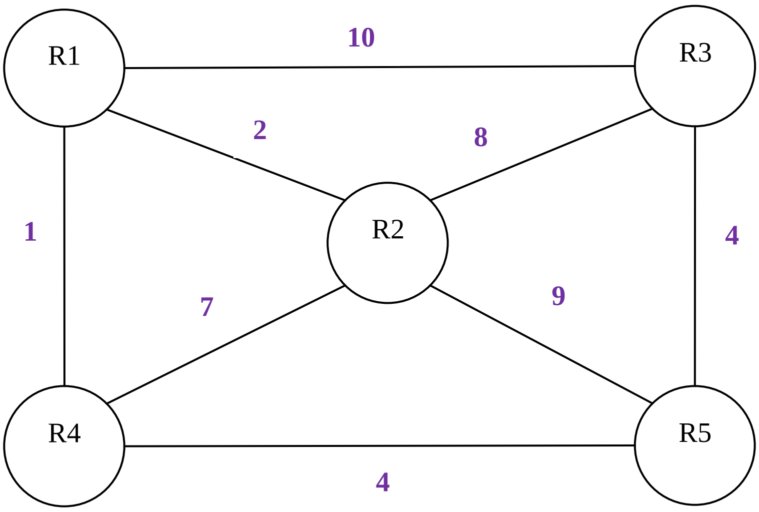

Figure: A sample network topology with costs on links

2) Sample Original Topology Matrix

Based on the above network topology diagram which is an undirected graph, we set the cost between a router and itself to 0, the cost between two indirectly connected routers to -1. Value -1 is equal meaning of infinite value (∞) as link state routing algorithm discussed in the class. Then we have the following original topology matrix:

	R1	R2	R3	R4	R5
R1	0	2	10	1	-1
R2	2	0	8	7	9
R3	10	8	0	-1	4
R4	1	7	-1	0	4
R5	-1	9	4	4	0

Table: A sample original topology matrix

3) Sample Input File (matrix)

The above table will be given as an input in a text file form named ***matrix (or other)***. Please notice that this is only a sample table and your program is required to be able to process a network with an arbitrary number of routers and costs.

Expected Sample Program Menu

The program prompts in blue. *User input in red.*

After a start to run your program, the following menu is expected to prompt (but the menu and user interface are free to design, no need to follow below way, even GUI is also welcome):
* A team project expects GUI design.

****Following will be a minimum requirement for all the students.**

```
CS542 Link State Routing Simulator

(1) Input a Network Topology
(2) Create a Forward Table
(3) Paths from Source to Destination
(4) Update Network Topology
(5) Best Router for Broadcast
(6) Exit
Command:
```

Master Command: ***1***
Prompt: Input original network topology matrix data file:
Input: ***matrix***
Prompt:

```
Review original topology matrix:
0    2    10   1    -1
2    0    8    7    9
10   8    0    -1   4
1    7    -1   0    4
-1   9    4    4    0
```

(Print matrix table and automatically return to the main menu)

(accept different input filename: matrix is a default name but also can accept other filename) – this part is new function from the past

Master Command: ***2***
Prompt: Select a source router:
Source router: ***1***
Prompt:

Router 1 Connection Table

```
Destination        Interface
======================
      1                    -
      2                    2
      3                    10
      4                    1
      5                    10000
```

(sample value so the interface selection may not be accurate)

(Print a connection table of router 1 and automatically return to main menu)

(if a source node is not selected but skip to choose menu 3, a default source node is 1) – this is new function from the past

Master Command: ***3***

Prompt: Select the destination router:

Destination Router? ***5***

Prompt: *Display all paths from the source to the destination.*   *And then* The shortest path from router 1 to router 5 is 1-?-…-?-5, the total cost is (value)?

// next selection router or other input to return to the main menu

Master Command: ***4***

Prompt: Add or Delete a router from this network topology.

Down router?   ***3***   *// means router 3 becomes down.   Adding a router is not listed*

(Result) Show an updated Connection table and a revised shortest path if the source and destination path was chosen by former last selection. Adding a router is not viewed but your project needs it.   If menu 3 has not done yet, create your own way how to handle the case

Master Command: ***5***

(Result) Show the router which has shortest paths to all the other routers in the network along with the cost. The sum of the costs from this router to all other routers should be minimum. If more than one router satisfies the criteria, choose one arbitrarily.

(Automatically return to the main menu)

Master Command: ***6***

Prompt: Exit CS542 2019 Spring project.   Good Bye!

(Terminates the program)

Deliverables and Grading Policy

1. Well commented source code (5%)
2. Design and test report. The design of your program, what you did to convince yourself that the program works correctly, briefly explain link-state routing

protocol. The test report for input files must be given in the project report. Also, you are required to specify at least five instances (5 routers at least, ×N times of routers for team project, N is number of members) you developed additional and report the corresponding results. TA will test your program with another file (in the same format as network.txt) which won't be released to you until the demo. A PPT (Power point) presentation file with screen shots how your project works and link state routing protocol. The maximum numbers of slides are 15 slides for an individual project. A team project PPT should be more. (20%)

3. Detailed description of your algorithm applied to find the shortest path between two selected routers. (10%)
4. Detailed instructions how to compile and run your program. (5%)
5. Results output by your program during demo. (30%)
6. TA will test your program with his own data file not released to the students. Your program should be able to process the input matrix regardless of its size when you show a demo. (30%)

- The above point allocation can be changed by the TA with his preference

Submission Guidelines

Due date: 4/28/2019 Sunday 23:59 US Central Time

- **NO** late submission will be accepted
- **If you have any unavoidable situation, provide proof of what happen to you and TA will give you up to 20% of late penalty. However, maximum late submission will be 48 hours from the deadline.**
- An individual or team demo is required to your TA (will be scheduled later)
- A maximum of **2** students per team, you may work individually if you like
- The team work should be superior to any individual work
- Your TA will ask to test more different scenarios for team project
- If you choose a team project, you must provide team member list to your TA no later than 3/31/19 and should receive a confirmation from your TA
- **A team project will get additional feature implementation requirements from your TA**
- Once you choose a team project, that team cannot change team members and must stay in the team -- **No return to individual project from a team.**
- Only one submission per team (or individual)
- The instructor and TA are not responsible for debugging your code
- Your submission should consist of four items only:

  - ➢ a project operations manual – how to use your project
  - ➢ the source code of your program – must contain detail comments
  - ➢ PPT file of your project screen shots and manual
  - ➢ executable program
- All items compress into one zip file format (not any other compression format at this time, no RAR format)
- Each team personal must contribute the work equally. Team project needs to create a project scheduling and plan of each individual work. Programming should also be done 50% each. If you think your other partner does not cooperate or progress well, you can report the status any time. Then the other lazy partner will be penalized and the team can be breakable to individual mode. The first reporter will be safe.

- filename format:

  CS542_2019SP_Project_{seat#}_Lastname_Firstname.zip

  - ➢ If a team project, use both team member's seat numbers and names as filename – team project requirement will be provided later
  - ➢ In a project operation manual, list who does what role
  - ➢ **E.g.**

    CS542_2019SP_Project_35_Kasapu_Viswatej.zip // individual proj

    CS542_2019SP_Project_99_Choi_Michael_35_Kasapu_Viswatej.zip

    // team project: Michael submits the whole package

    - o A head of the team (e.g. Michael Choi's seat number is 99) sends the whole package.

- Submission to BB only and **only one submission**
- The BB, at this purpose, will not accept multiple submissions and your TA will NOT accept more than a single submission -- So you should be careful to select your submission
- If you make any mistake, no bridge to return. However, if you have a technical issue, contact your TA but you must provide technical error you had. If you submit at the last minute such as 30 min before the submission due and cannot submit it, it will be considered as late submission
- Must avoid last minute submission to BB. With the excessive submissions from other courses, you may have long delay and it will cause late

submission – strongly suggest to submit at least six hours early

- Please put down your name (Last, First, Middle) student ID on the front page of the project report -- Your name must be the same as school registered name and if not, a penalty may be applied
- If you have any project related questions/doubts, please contact the TA (TA will consult your question with professor if he cannot decide by himself)
- You have to make sure your project MUST be executable at least on the PCs at the Main campus PC lab. Your TA will schedule everyone's project demo after all submissions. You have to be flexible to meet your TA's schedule.