Computer Networks CSE 4344/5344

Project 3
Implementation of Routing Protocol

Instructor: Dr. Sajib Datta

GTA (4344/5344): Srinivasan Murali & Abhishek Ambarla
Spring 2023

"What I cannot create, I do not understand." Richard P Feynman

Objectives

(A) To understand Routing protocols via sockets

(B) To gain exposure to a practical implementation of Distance Vector Protocol

(C) To simulate a router using software defined routing

Due Date

April 30, 2023 11:59 PM[1]

Project Description

(A) You will be developing an application which simulates a router/router protocol using sockets

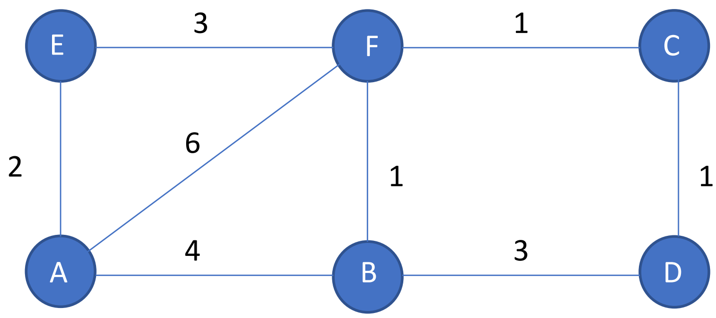
(B) You will attempt to write a set of distributed procedures for implementing a software defined routing simulation of the Distance vector routing protocol.

(C) Display the necessary contents on all routers for the different conditions as stated in the "Specifications" of this document. Please read it very carefully.

[1] All Submissions must be completed through Canvas

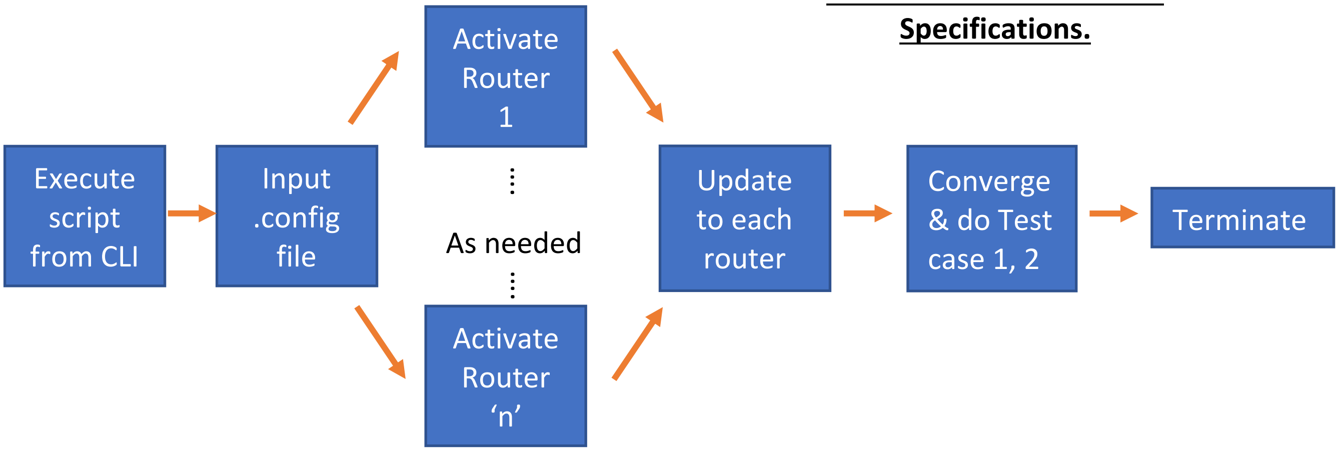
Specification

Network Topology



1. Construct the topology table, with the costs for the above figure, until convergence
2. This should be supplied as an input as a configuration file(.config) to an application (script) which would read in the router's configuration table and displays all its directly connected neighbors and their respective costs.
   a. Use a local host IP approach (127.0.0.x) to create the router configuration
3. The script should then start listening to desired port on UDP
   a. This port number should be entered via Command Line interface (CLI)
   b. Execute the script from CLI
4. The script should be able to connect to the specified IP's in .config.
   a. Print the configuration topology on the command line interface (on router 1)
   b. Incoming IP connections from neighbors should be accepted.
   c. Print router IP for socket creation and binding to port number, connection type (for each router)
5. Once the configuration file has been read by the script, this information must be sent to all neighbors and receive the corresponding updates from them
   a. Continue listening for any updates
6. If Update affects the concerned router's vector, then update the routing table, until convergence
   a. Print the updates on CLI for each router as follows:
   b. Source IP (Host)
   c. Current Cost (updated value)
   d. Previous Cost
7. Test Case 1: The script terminates all connections and shuts down the whole system after 'n' consecutive updates after the following information is printed (broadcasted across all routers) after convergence:
   a. Message from "Router Name, IP address, Port No."
   b. Your UTA-ID number(s)
   c. the date and timestamp in UTC
   d. Total number of updates value ('n')
   e. Payload size exclusively for this last broadcast
8. Test Case 2: Add the last 2 digits of all team members' UTA-ID in your team. If your result is:
   a. Even: Assume link between A/ B is broken.
   b. Odd: Assume link between B/ D is broken.
      i. Provide the explanation on what would happen and why.

Overview



Show/ Print all required contents as mentioned in Specifications.

Notes

(a) This is a group project.(refer to assigned sheet/ announcement)

(b) You can use the programming language of your choice 8.

(c) You may also want to refer to the text-book, section 2 and section 2.7, for socket programming, section 4,5 for routing protocols.

(d) The source codes should be well documented to make it easier for the TA to follow:

- ***To include: Function's parameters, input, output, purpose***

(e) The routers can all be started from the same computer using different CLI windows.

Submission Guidelines

- Submit a single zipped file with the naming convention,

  < your_U TA_id > _ < your_name > :zip

- Your submission should have the following items to be considered for evaluation,

  (a) All relevant source codes

  (b) Any additional files required to run your codes

  (c) –Very Important– readme.txt file with instructions on how to compile and run your codes. You must mention the IDE as well as any packages that are required to run the codes.

  (d) –Very Important– Provide ample amount of comments in the code to make it more readable and sustainable.

- Do NOT include any executable (binaries) program in your submission. Canvas has different interpretations for various executable and varied lev-els of security preventing TAs from accessing the executable.
- Make sure your name and your UTA ID are also listed in the readme file and in comments at the beginning of your source files.
- Make sure that submissions of the zipped file is through UTA Canvas.[2]
- Late submission will be NOT accepted by any means.

Additional Requirements/Instructions

(a) Please email your TA for any doubts and clarifications regarding the project 3.

(b) Complete documentation and instructions for running the codes are recommended. ***You are also requested to demonstrate the working of your code to the TA[3].***

(c) If you are using any code from some external source or book, you MUST mention it explicitly in the codes as well as the readme file. Otherwise, it will be considered plagiarism and your project will not be evaluated.

(d) You can discuss with other classmates on steps/algorithms to implement the project. However, the source codes must be written by your-self .

Grading Rubric (100 points)

(i) Construct correct routing table (10 points)

(ii) Read and print .config file (5 points)

(iii) Update until convergence correctly with all necessary specifications printed (30 points)

(iv) Test Cases passed (30 points)

(v) Coding quality (5 points)

(vi) Proper closing of the ports with exception handling. (5 points)

(vii) Display/log of proper messages on all nodes (5 points)

(viii) Code documentation and Readme file (10 points)

[2] Please strictly follow the naming convention of the zipped file

[3] TA hours will be the best time to do this. We will try open some slots in google doc, so that you can come in specified time slots to give a demo without much trouble