Distance Vector Routing Protocol

BT21CSE073 Mahika

Instructions to run the code

- Download the zip folder of the codebase and unzip.
- Open terminal and change directory to that of the codebase folder.
- type *python dvr.py <input_filename.txt>* in the terminal to run the code.
- There are three input files, namely input.txt, input1.txt and input2.txt

Working of the code

- Designed and implemented a Router class which contains attributes and methods like -
    - Name of the Router
    - List of Neighbouring Routers
    - Routing Table: The Routing Table of each router contains all the destination routers, the minimum cost required to route to the destination routers and the path via which it routes to the destination routers.
    - Shared queue (containing routing tables from other routers)
    - *queue_lock* to acquire and release the lock for the shared queue

- The *input_parser* function parses the input text file and creates the Router instances phase and assigns cost to the links between the routers. It implements the Initialization phase of the DV (Distance Vector) Algorithm.
- The *threaded* method is the function which is invoked when a thread is created for each instance of a router. This method invokes the *add_to_queue* method which shares the current router's routing table to all other routers. The threaded function then waits for the current router to receive the routing tables from other routers.
- After storing the router tables of other routers in Router.queue , we will apply **Bellman-Ford's** equation to calculate the least-cost path from the current router to all other routers in the network.
- We then update the current routing table according to **Bellman-Ford's** equation and wait for 2 seconds before running the next iteration of the algorithm on each router
- Note: Each iteration of the algorithm is run on each router concurrently (parallelly) using multiple threads.
- Finally, over multiple iterations (in our case 4), the least-cost required to route to all the other routers is calculated and stored in the Routing Table.

Testing

- The DV Algorithm was tested on three input files, namely input.txt, input1.txt and input2.txt
- input.txt contains 4 routers with 5 links and is a fairly simple network.This was the example network provided to us in the Problem Statement. The algorithm calculates the least-cost path for all routers to all other routers in two iterations