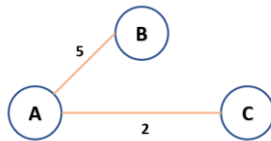


CSL317 Computer Networks Practical

Distance Vector Routing Protocol

In this assignment, you will be implementing the "Distance Vector Routing" protocol. Since it follows a decentralized approach, you will be implementing this algorithm via multithreading in Python. Each thread will represent a router instance. **The threads (routers) can communicate through a shared queue.**

For the following sample topology,



the program should parse the following input to understand the topology of the network:

```
3
A B C
A B 5
A C 2
EOF
```

where:

- The first line represents the number of routers present in the network.
- The second line represents the fixed name (interface) of the router.
- The third line like "A B 5" represents the cost of link connecting nodes A and B. Similar semantics is followed until the last line.
- The last line represents the end of file (denoted by "EOF").

Output and implementation:

The output should initially display the routing tables of each router.

After 2 seconds, each router should forward its routing table to the adjacent routers using a shared queue.

The routers should now update their table entries using the Bellman Ford equation.

The updated routing tables of each router should be displayed on the terminal.

Note that each time the routing tables are displayed, the iteration number should be also displayed along with it. **Also, those entries of the table that were updated in the latest computation round should be marked with an asterisk (*).**

After each computation, the routers should wait for **2 seconds** and then forward the tables.

Since each router knows the number of adjacent routers, each router will wait to receive the table from ALL the adjacent routers and only then perform computation.

The output to be displayed should be very intuitive and must clearly display the entries of each router after every iteration.

It should be noted that the computations should be performed in a decentralized manner in each individual thread. The code should be well documented.

Submission instructions:

Submit a `<roll no>_<name>.zip` file which consists of following files:

- *dvr.py* : that implements the dvr algo; **it should take inp.txt as input parameter that contains the topology information in the prescribed format.**
- README file that describes how to run the code with inp.txt file as input
- 5 minutes (approx.) video file clearly explaining the codes and demonstrating the various test cases that you have tried.