COSC 350 Assignment 3 Spring 2020

Due Tuesday May 12 11:59 pm.

Upload to BB a single zip file GroupNameA3 for your group that has the 2 Java source code files for the server and client named as GroupNameA3Server.java and GroupNameA3Client.java.

Write a TCP socket program that allows the server (router R1, node ID 7) to receive a DVR message from the client (router R2, node ID 6) and update R1’s routing table using the DVR algorithm. The server and client must do the following (values are given below just as an example to show how a user enters the input).

Server

1. R1 (server: node 7) first reads a file rt.txt containing R1’s neighbor distances and R1’s routing table stored in the following form (the meaning of each line in the file is given as text after the //, but neither the // nor the text is stored in the file):

1//number of neighbors (only one in this case but can be more)

6//neighbor IDs (separate by a single space if more)

5//neighbor distances (same as edge weights or costs in the network graph; separate by a single space if more)

6//number of destinations in routing table

7 4 6 3 2 5//IDs of destinations in routing table

0 4 6 4 6 6//next hop to reach each destination

0 3 5 14 13 19//distance to each destination

So R1’s neighbor distance pair is (6, 5); and R1’s routing table is as follows:

|  |  |  |
| --- | --- | --- |
| Destination | Next Hop | Distance |
| 7 | 0 | 0 |
| 4 | 4 | 3 |
| 6 | 6 | 5 |
| 3 | 4 | 14 |
| 2 | 6 | 13 |
| 5 | 6 | 19 |

2. R1 (server: node 7) then waits for a TCP socket connection on port 6789 from R2 (client: node 6) to receive the DVR message from R2.

3. R1 receives the DVR message and uses the DVR algorithm (Algorithm 18.3) to update its routing table using the (destination, distance) values in the DVR message.

4. R1 prints its updated routing table showing values of (destination, next hop, distance).

Client

1. R2 (client: node 6) first reads a file dvr.txt containing the information in the DVR message to be sent to R1 (server: node 7) stored in the following form (the meaning of each line in the file is given as text after the //, but neither the // nor the text is stored in the file)

4//number of entries in DVR message: 4

6 3 2 5//destinations in DVR message separated by a single space:

0 2 8 28//distances to each destination in DVR message separated by a single space:

2. R2 (client: node 6) uses the preceding information to form the DVR message, which consists of the pairs of (destination, distance) values, which is as follows:

(6, 0), (3, 2), (2, 8), (5, 28)

3. R2 (client: node 6) makes a TCP socket connection to R1 (server: node 7) on port 6789 and sends a message M containing the DVR message to R1. The message M consists of the number of entries in the DVR message followed by the (destination, distance) pairs in the DVR message above.