

CSEE W4119 Computer Networks - PA2

Li Yan (ly2278) ly2278@columbia.edu

Distance-Vector Routing Algorithm

1 Building instruction

```
cd src
make
```

Then run java program.

2 Document detail

```
src/
```

- `Accessory.java`
- `Constant.java`
- `DV.java`
- `dvnode.java`
- `Receiver.java`
- `Router.java`
- `Transmission.java`
- `makefile`

The the main function of the project is `dvnode.java`.

3 Project feature

1. Applying Bellman Ford algorithm to the Distance-Vector Routing algorithm.
2. Print node table when ever it is updated.
3. Print the message sending and receiving condition.

4 Data structure

In this project, each node is identified by its port number. However, port number is not very suitable as the index of the routing table. Thereby, each node have a map mapping other nodes' port number to a **router ID**. Router ID is an integer start from 0 (0 represents itself). Then using `double[ID1][ID2]` to represent the routing table from router ID1 to router ID2.

4.1 Port to ID map

This is instantiated by HashMap in Java, which is `HashMap<Integer, Integer> port2routerIDHashMap`. Its key is the port number, its value is relative router's ID.

4.2 ID to Port map

This is instantiated by `int routerID2port[]`. `router2port[i]` gives the port of the router whose ID is `i`.

4.3 Routing Table

This is the most important data structure, the routing table is stored in `double DV.data[][]`.

4.4 Link information

Another important data structure is for link information, which stores all routers that are neighbours and also stores link weight to this neighbour router. This is instantiated by `HashMap<Integer, Double> linkHashMap`. Its key is peer router's port, this value is the link weight to this router.

5 Algorithm

The main flowchart for DV is given in Fig 5-1:

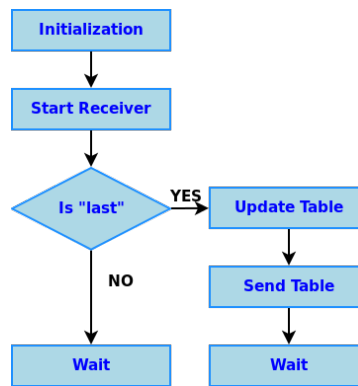


Fig 5-1

The receiver have the flowchart of Fig 5-2:

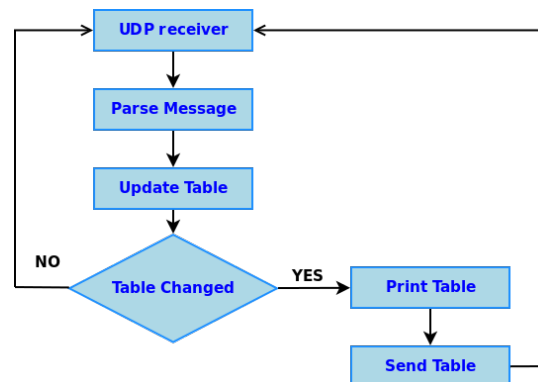


Fig 5-2

The receiver is run as a thread, the reason for this is that in the combination, the main process will do many other jobs.

6 Usage scenario

With proper run command, all dvnodes that is not the “last” listen to updates from their neighbours. The node that is the “last” will update the table from its link condition and send out the first updates. The each node will update tables and send out updates to its neighbours until the table does not change any more.

At last, the convergent condition is that each node have a table will have the shorstest path and no more message sent.

7 Test

A test output is in folder “[Test/](#)”, this use the model given in the homework example.

In detail:

- file “[1111](#)”: the output of node whose port is 1111.
- file “[2222](#)”: the output of node whose port is 2222.
- file “[3333](#)”: the output of node whose port is 3333.
- file “[4444](#)”: the output of node whose port is 4444.