Advanced Network Communications H - Assessed Coursework Network Description & Walkthrough

Name: Dominic Houston-Watt **Matric:** 2253422h

This file will cover the structure of network description files, provide diagrammatic versions of the provided network description files and give a runthrough of using the application to simulate the provided networks.

Note: To run your own network, follow the file structure below in a *.txt* file and place the file in the same directory as the runnable jar file.

File structure

Provided in the submission documents are 2 example networks. When the application is run, network1.txt is preloaded, however you can change what network file is loaded by using the `load` function in the application. The structure of the network description files are as follows

LINE 1 - Number of nodes in the network
LINES 2->n - Bidirectional edges in the network, written as ***node1_label, nodel2_label, cost***

This structure should be used if the user is wishing to simulate their own network, with a network description file.

Diagrams

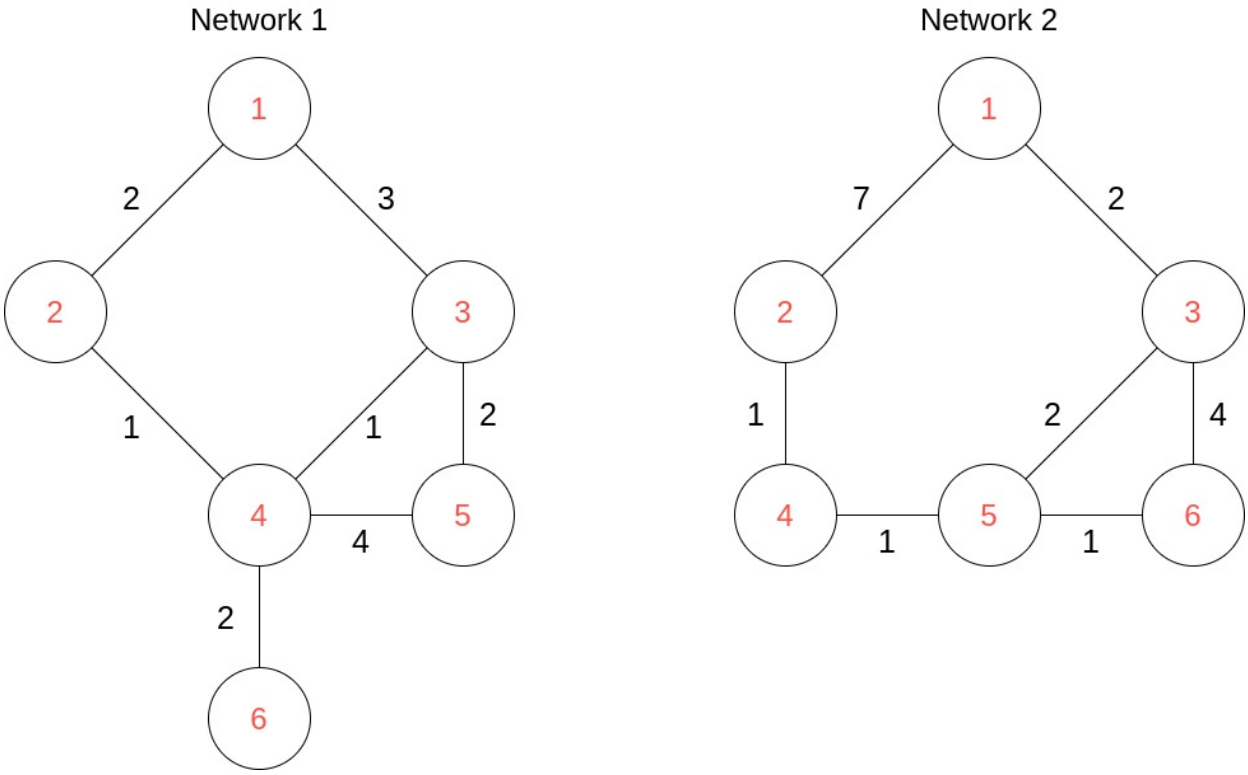

Fig 1: Diagrammic representations of the provided networks, with Nodes and links identified by labels and costs

Walkthrough

This section will give a brief runthrough to show the evolution of the distance vector table and network simulation for the networks provided.

Network 1

Upon running the application, network 1 is pre loaded. First, `printnet` is used to see the initial distance vector table

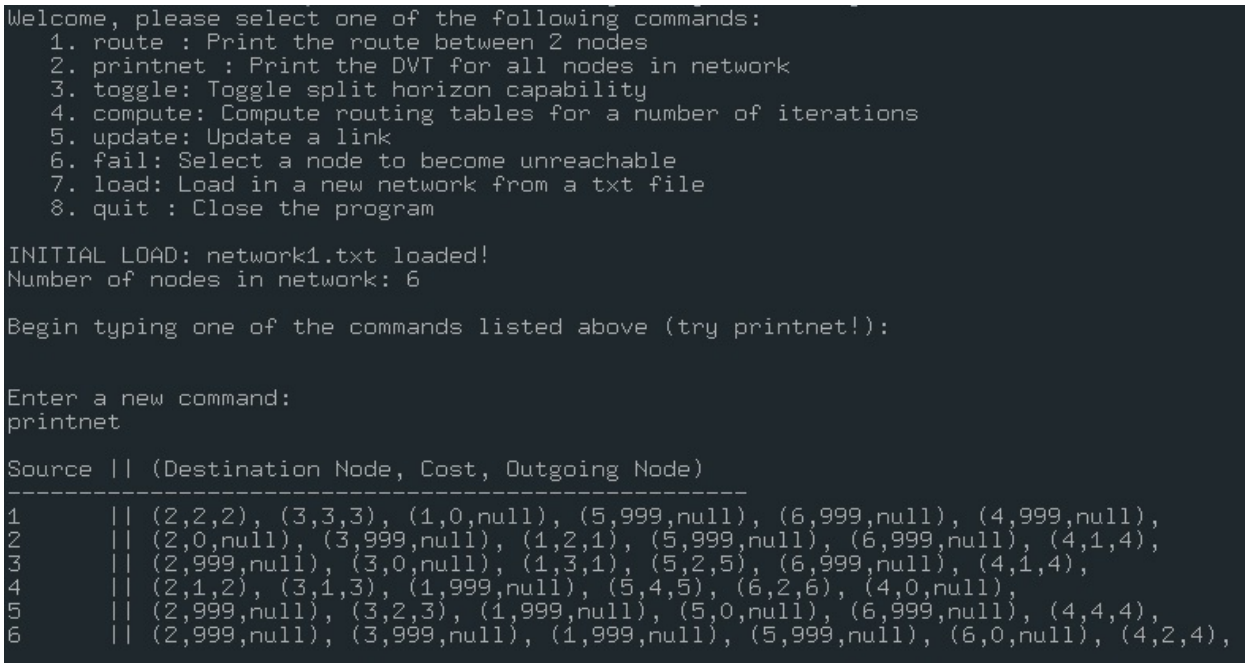

Fig 2: printnet command output in the application

You can see that prior to calculation of distances, nodes that don't know the location of others in the DVT have a total cost of 999 and an outgoing node of null. For example, node 2 does not directly connect to node 3, so the cost is set to 999.

Computing the DVT

Next, `compute` can be run to perform the exchanges for the nodes, providing the number of iterations the user wants to perform (each iteration is when the nodes perform the exchanges)

```
Enter a new command:
compute
Please enter the number of iterations you want to perform:
> 3
Initial Network DVT
Source || (Destination Node, Cost, Outgoing Node)
-------------------------------------------------
1      || (2,2,2), (3,3,3), (1,0,null), (5,999,null), (6,999,null), (4,999,null),
2      || (2,0,null), (3,999,null), (1,2,1), (5,999,null), (6,999,null), (4,1,4),
3      || (2,999,null), (3,0,null), (1,3,1), (5,2,5), (6,999,null), (4,1,4),
4      || (2,1,2), (3,1,3), (1,999,null), (5,4,5), (6,2,6), (4,0,null),
5      || (2,999,null), (3,2,3), (1,999,null), (5,0,null), (6,999,null), (4,4,4),
6      || (2,999,null), (3,999,null), (1,999,null), (5,999,null), (6,0,null), (4,2,4),


################################## Iteration: 1

Calculating...

Improved route identified from 1 to 4
       New Outgoing Node: 2
       New Cost: 3
Improved route identified from 1 to 5
       New Outgoing Node: 3
       New Cost: 5
Improved route identified from 2 to 3
       New Outgoing Node: 1
       New Cost: 5
Improved route identified from 2 to 5
       New Outgoing Node: 1
       New Cost: 7
Improved route identified from 2 to 3
       New Outgoing Node: 4
       New Cost: 2
Improved route identified from 2 to 5
       New Outgoing Node: 4
       New Cost: 5
Improved route identified from 2 to 6
       New Outgoing Node: 4
```

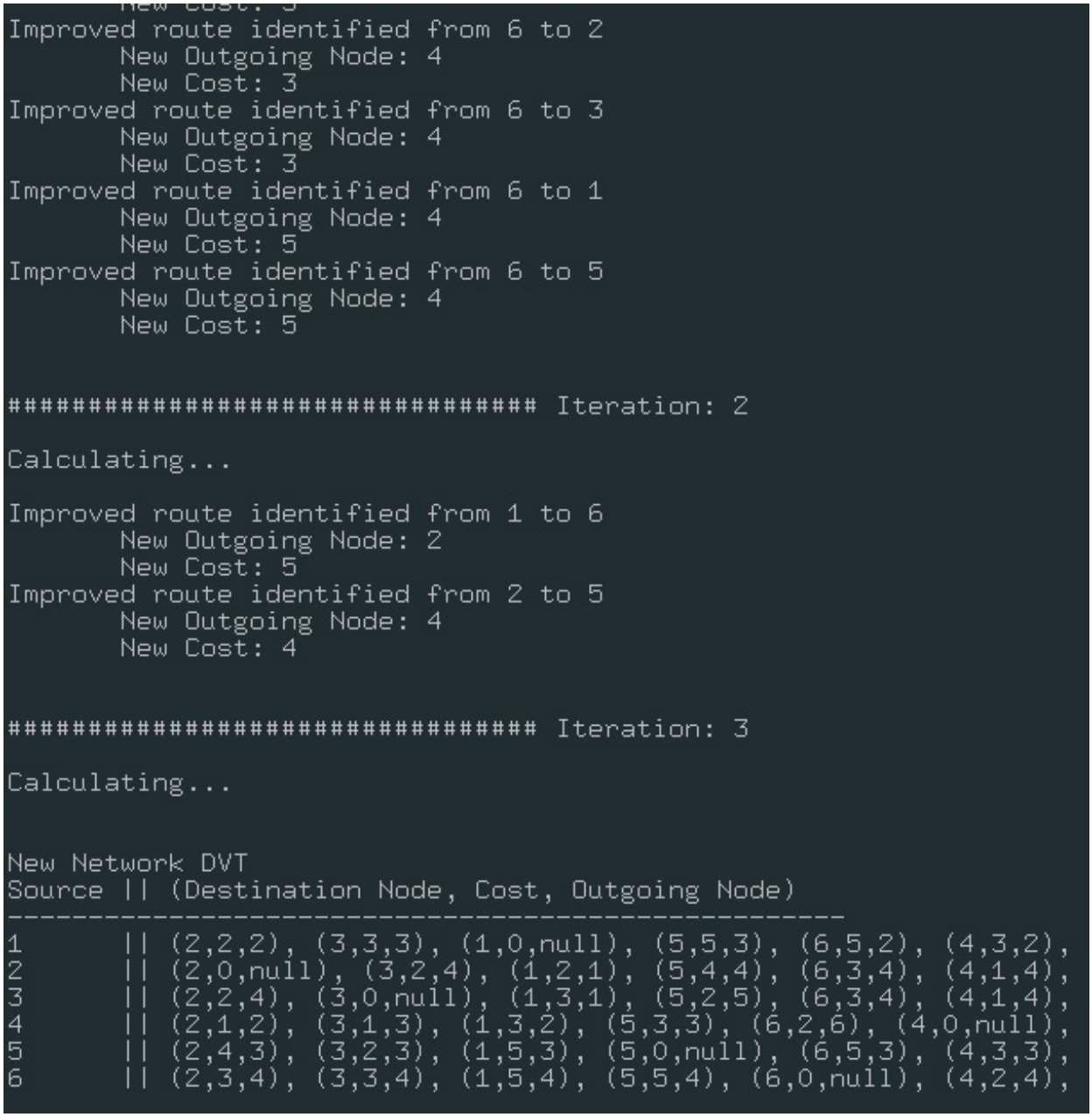

Fig 3: Output from the compute command, showing the progression of route calculation

When calculations are performed, by using the Bellman-Ford algorithm, if a better route from a source to a destination is found the new cost and outgoing node are printed, as the iterations continue you can see that less paths are being identified, as the network comes closer to stabalisation.

The networks DVT (showing the DVT for every node in the network) is printed after the iterations have completed.

Updating Links Between Nodes

If the user wishes to update a node, say changing the cost from node 2 to node 4, the `update` command can be called. The allows the user to enter a new cost associated with the link, however after this the DVT for all nodes now needs to be recalculated incase of any better routes now being available. In the screenshots, the cost between nodes 2 and 4 are changed to a value of 13.

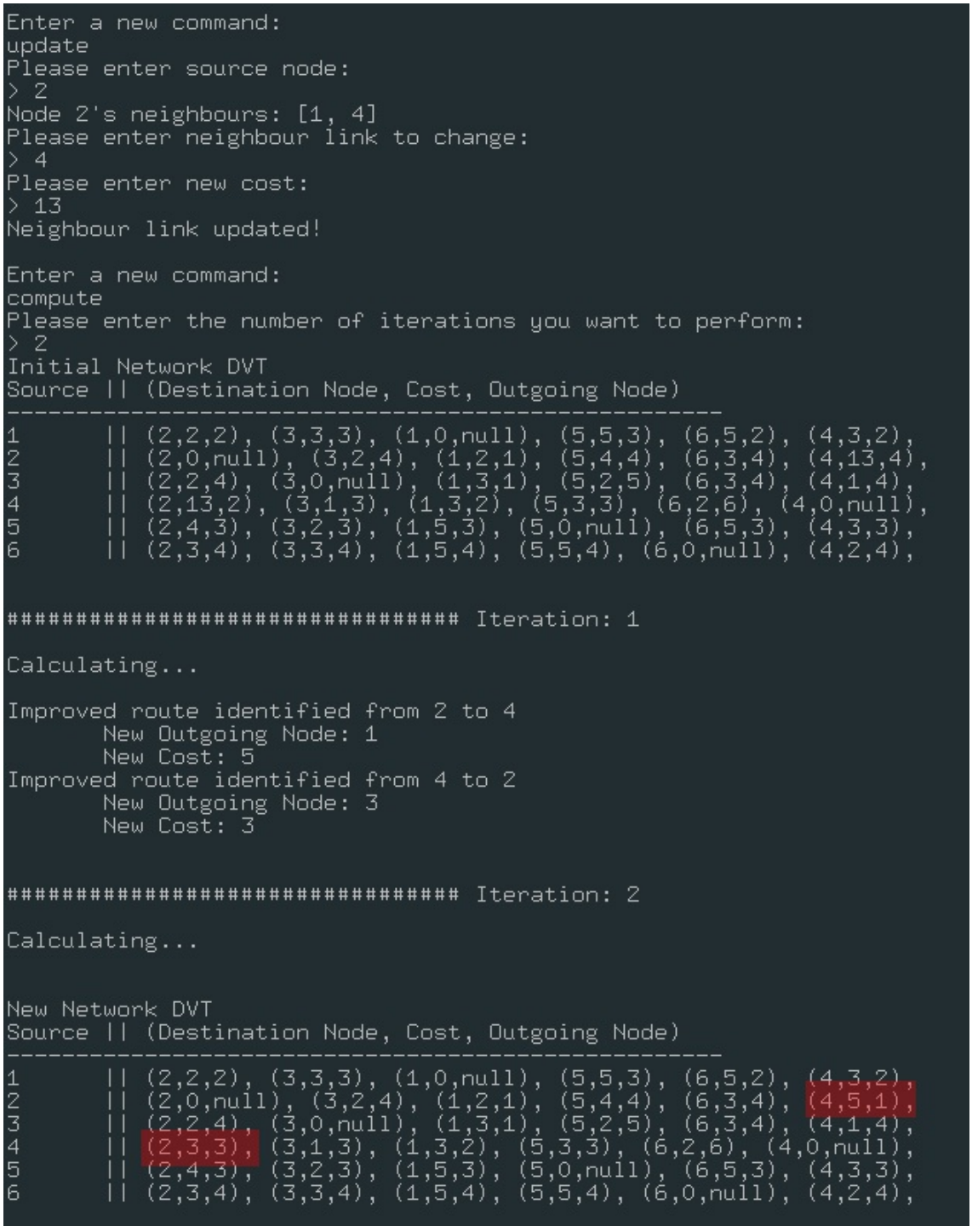

Fig 4: update command being used. Highlighted entries in the DVT show the changes made once links were updated and routes were recalculated