ANC4(H) Assessed Exercise
Ibrahim Javed 2265799J

Overview:

Figure 1 below shows the application.

- The application has 4 text fields and 4 drop-down menus at the bottom, the menus allow you to select nodes and the text fields then display the routing table for that node. The tables show the best route found so far.
- The “Single Iteration” button at the top left runs the routing algorithm for 1 exchange of tables.
- The “Iterations” button to the right of the previous button, runs the algorithm for how ever many iterations specified in the text field next to it.
- The “Until Convergence” button runs the algorithm until convergence.
- The “Update LINK” button on the second row allows the user to update the cost of any link and allows the deletion of links.
- The toggle above the table’s named “Split-Horizon” allows the user to switch on the split horizon facility.
- The app was developed in Java using object-oriented programming.

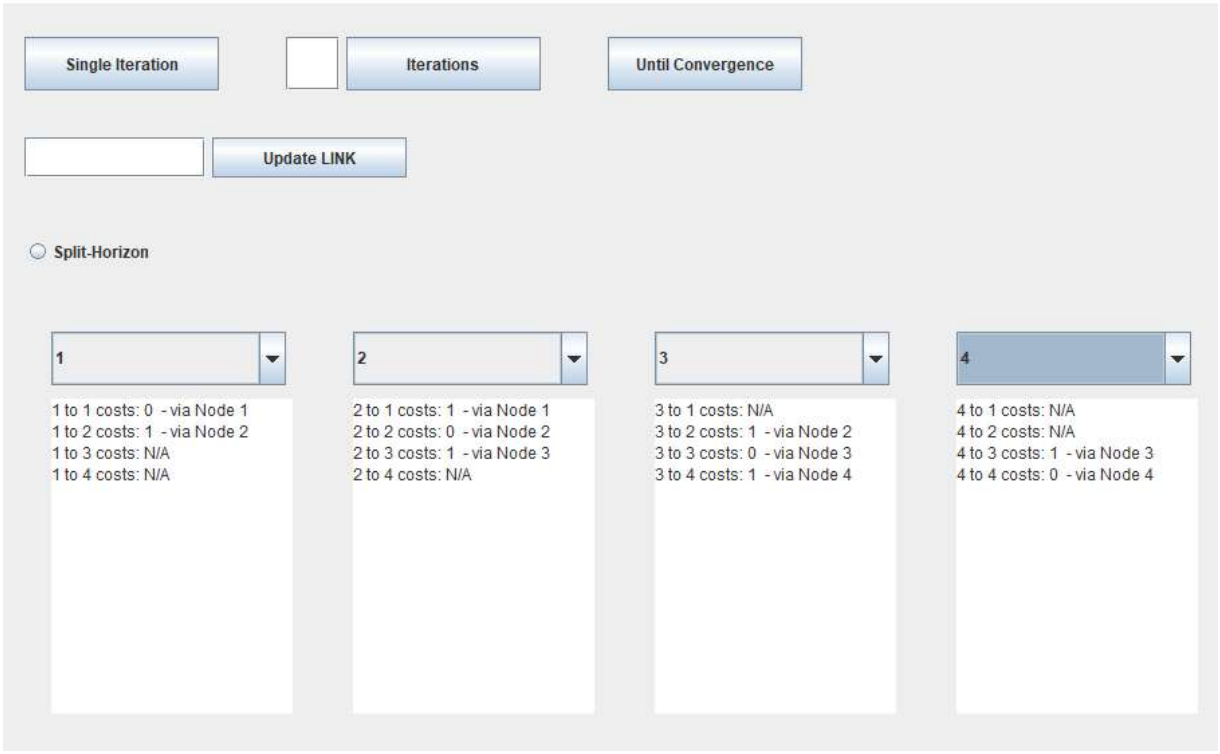


Figure 1

User Manual:

Install:

- Import the project file into Eclipse, on the top menu of Eclipse select the 'Run' menu and select 'Run Configurations', Select 'ANC4' as the project and 'network.App' as the main class. Apply and close the pop-up.

- Select from the top menu 'File', select 'Export', select 'Runnable JAR file', under 'Launch Configuration' select 'App-ANC4' choose any destination to save the runnable.

- Place the runnable inside a new file and inside the file create a file called "network" the file will contain the network description read by the app. The text file must be called "network". This file should be edited uses a text editor such as Notepad.

Instructions:

- **The first line in the network file describes all the nodes and subsequent lines describe the links**. For example, figure 2 below shows a network file, this network has 4 nodes: 1,2,3,4 as described by the first line. The second line describes a link between nodes 1 and 2 which has a cost of 1. Links are bidirectional and only need to be described once.

- Once the network file is completed and saved, double click to run the executable, select from the drop-downs what routing tables you want to monitor, these will update in real time, you can only see up to 4 at one time however you can change selection at any point.

- To change the cost of any link simply select the field next to the "Update Link" button and provide a new costing. For example, to change the cost between nodes 1 and 2 to a cost of 5, write "1,2,5" and press the "Update Link" button. To completely delete a link, write "1,2,1000" this will delete the link between nodes 1 and 2. "1000" tells the app to delete the link.

- Note, the app doesn't allow adding links meaning once a link is deleted it is gone, however links can be added in the network file.

- When using the split-horizon toggle, select or de-select the button **prior** to altering or deleting links.

- The "Until Converges" button uses a checksum however it is limited to a maximum of twenty iterations in case the checksum is never reached – for very big networks it may have to be pressed twice.

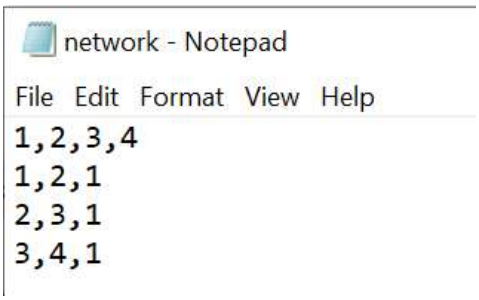


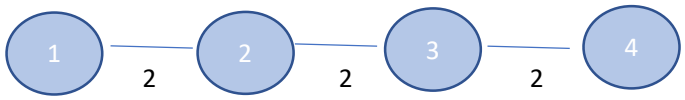
Figure 2

ANC4(H) Assessed Exercise
Ibrahim Javed 2265799J

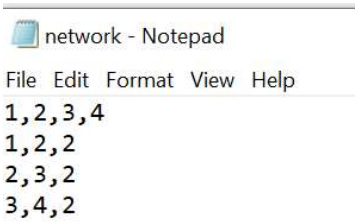
Examples:

Network 1:

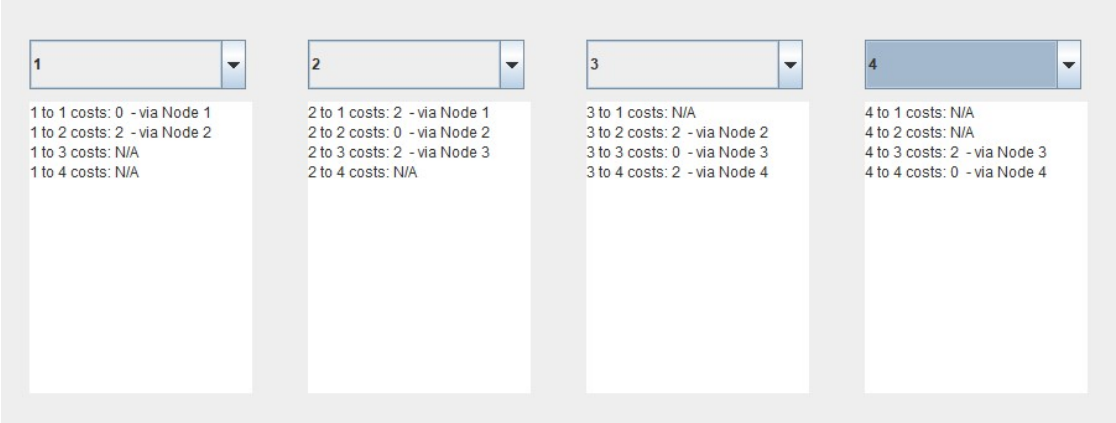
The first example is very simple, this network is used to test basic functionality and to test slow convergence. The network can be described graphically as follows:



The network can also be described to the application in the "network" file as follows:

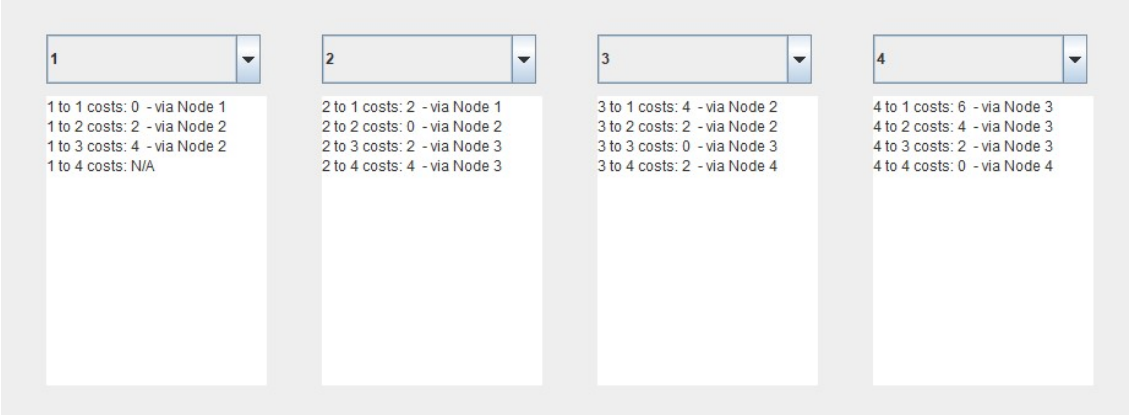


Upon opening the application and selecting the routing tables to be shown, since this network has 4 nodes, we can view all 4 tables:



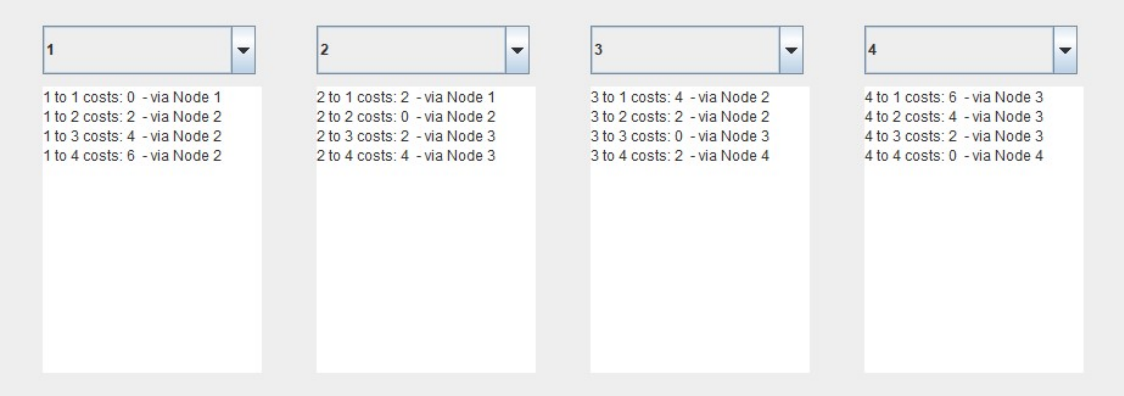
We can see the initial routing tables before any exchanges occur, the tables only show the bidirectional links described in the file. The routing tables are as expected.

Below we can see the updated routing tables after pressing the "Single Iteration" button:



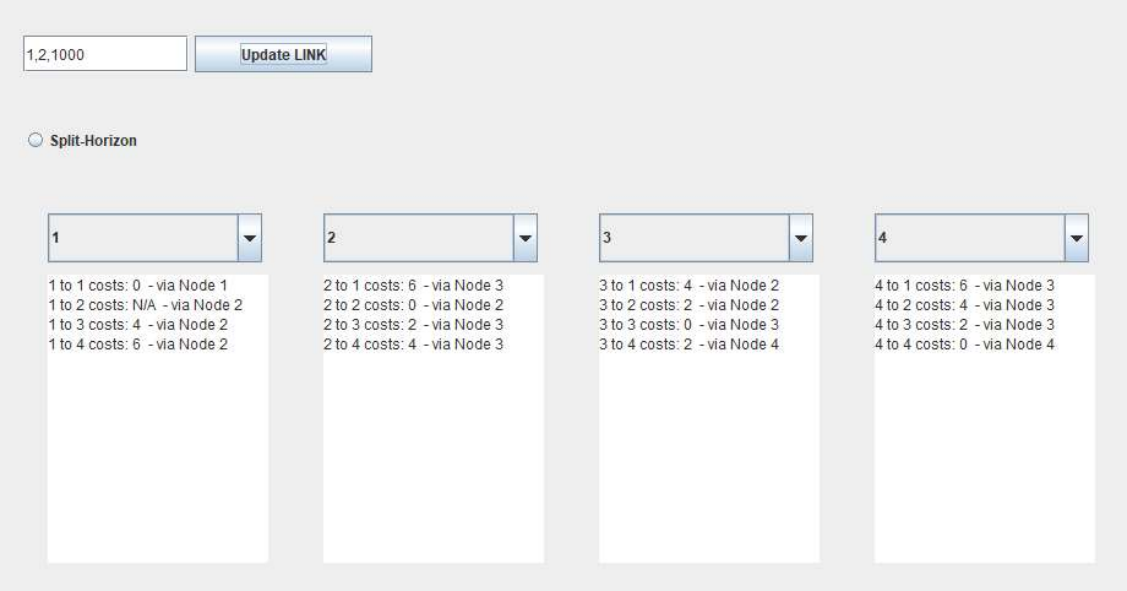
After 1 iteration we can see that **Node 1 now has a path to Node 3 via Node 2** as they exchanged tables, **Node 2 now has a path to Node 4 via Node 3** and, **Node 4 now has paths to Nodes 1 and 2 via Node 3.** Node 1 still doesn't have a path to Node 4, this is because Node 1 and 2 exchanged tables first and when they exchanged Node 2 didn't have a route to Node 4.

After pressing "Single Iteration" again:



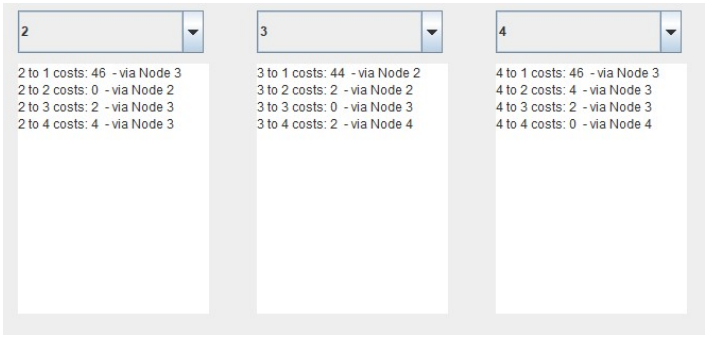
We can now see after this iteration **Node 1 has a path to Node 4 via Node 2**. It took 2 iterations for the network to converge.

Now the network has converged the link between Nodes 1 and 2 is going to be deleted, this is done using the "Update LINK" button as follows:



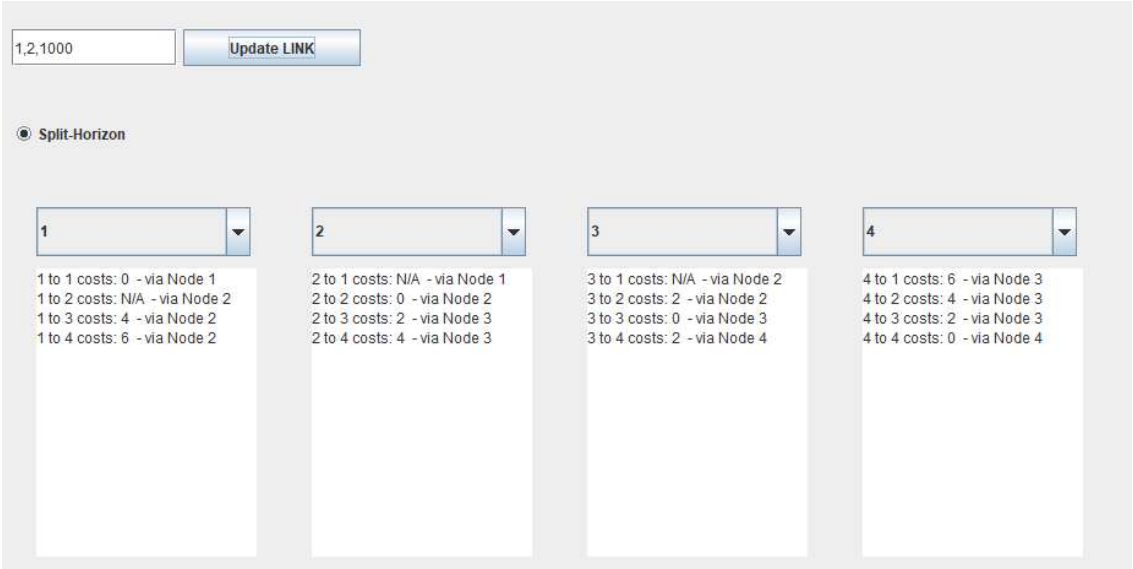
On Node 1's routing table we can see the link between Node 1 and 2 has changed to "N/A" as expected however Nodes 2 and 4 think they can reach Node 1 via Node 3 and, Node 3 thinks it can reach Node 1 via Node 2. This is an example of slow convergence.

We now run 10 iterations using the "Iterations" button:



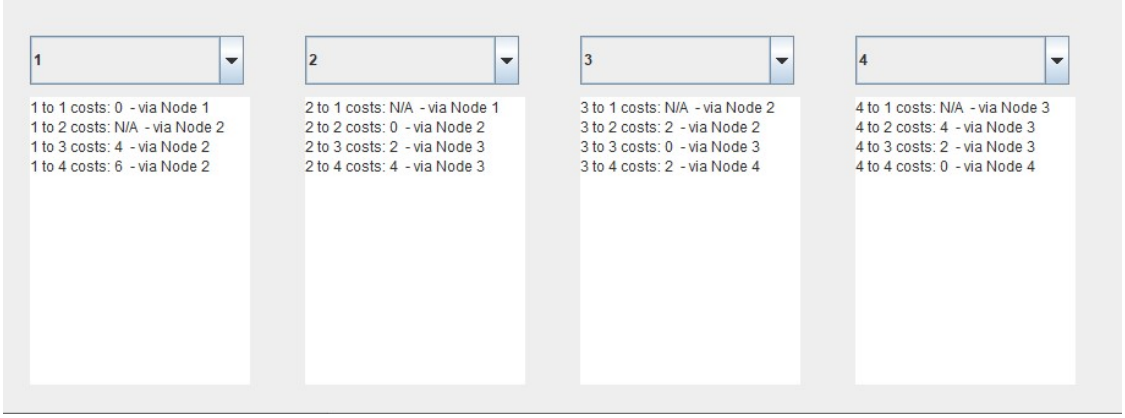
We can see the full effect of the count to infinity problem.

The app is now restarted, the “Until Convergence” button is pressed to skip to convergence, we are again going to delete the link between Nodes 1 and 2 this time with the “Split-Horizon” toggle selected. Below shows the routing tables after deletion:



This time, the link to Node 1 is “N/A” on all routing tables except for Node 4’s. Node 3 is showing “N/A” as the start and end nodes of the link have alerted their neighbours of the change. Node 4 is still unaware as no exchange of tables has occurred.

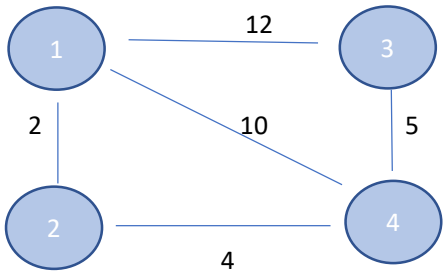
Below show the updated routing table after another iteration:
Node 4 is now aware of the broken link and shows “N/A”.



ANC4(H) Assessed Exercise
Ibrahim Javed 2265799J

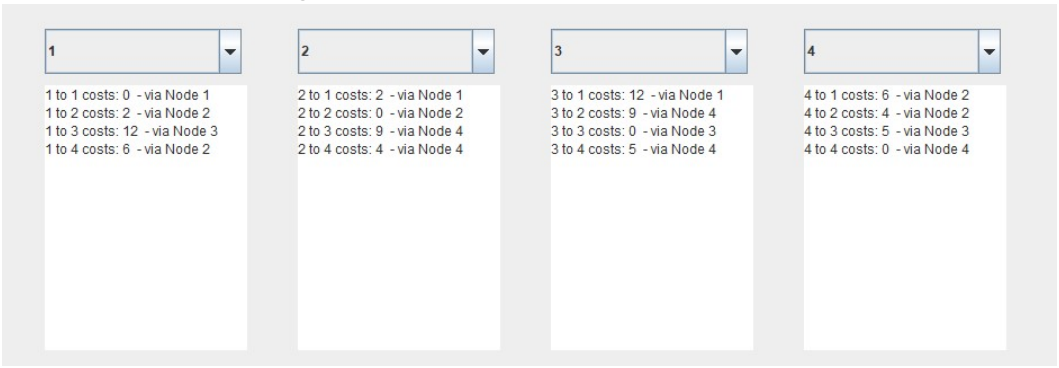
Network 2:

The network can be described graphically as follows:



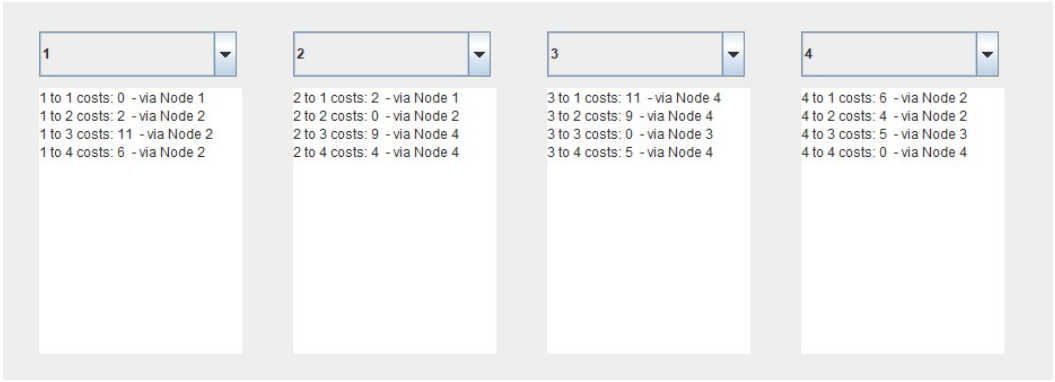
The network can also be described to the application in the “network” file as follows:

After 1 iteration the routing tables are as follows:

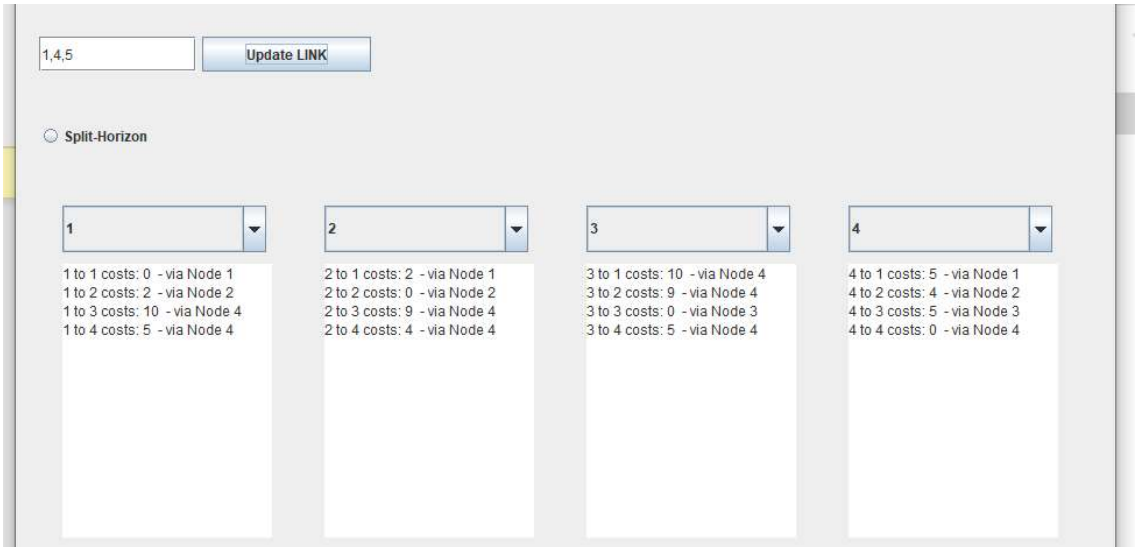


Node 1 to Node 3 costs 12 and they travel via each other. This isn’t the cheapest path. It takes another iteration for Node 1 and 3 to find the quicker path.

Routing tables after another iteration:
We can now see Node 1 travels to Node 3 via Node 2. And Node 3 also now takes the quicker path via Node 4



Node 1 has a direct link to Node 4, however after first iteration it goes via Node 2 as this is cheaper, lets lower the cost of the direct link to 5 and see what happens.



Node 1 and 4 immediately update and now communicate to each other via the direct link. As Node 1 and Node 4 have notified their neighbours of the cost change, Node 3 and 1 now also communicate via Node 4 at the cheaper cost. There hasn't been an iteration to cause all these changes, however since nodes tell neighbours of cost updates these updates have occurred.

Limitations and Improvements:

- Given more time, I would test the application as it hasn't been thoroughly tested.
- The program doesn't communicate to the user, this would be added, for example once convergence is reached the user should be notified, ideally the user should be notified of the status of any command.
- I would investigate the split-horizon implementation as currently you can't turn it on and off on the fly, it must be chosen before links are deleted or changed to work properly.
- I would have a reset button within the application, as currently say you want to re-simulate a network you have to open and close the application which isn't ideal.
- Potentially could show the network graphically.
- Use a character set that allows the symbol for infinity as to "N/A".
- The application would have a better way of reading in files, a set file name isn't ideal especially with multiple networks.
- Could be improved by implementing a different algorithm and comparing convergence times and routing tables.

END OF REPORT