## High level overview

The program has two versions, the library and the application. The application uses the library in it's implementation

The application is a text based program, which provides the user with a multitude of options to run their network simulations. Networks can be loaded from file and stored to files. The full explanation of how to use the application is in the user manual

There is two classes used in the library, *Node* and *NodeHandler:*

- Node describes the individual nodes of the network, Each node maintains a list of the nodes they are directly connected to and their current distance-vector table
- NodeHandler controls the network as a whole. NodeHandler tells the nodes when to send updates and when to apply their updates. If a connection is changed or downed or a node is removed from the network, the handler notifies the relevant node(s)

In order to ensure each node is updated synchronous, the updates happen in two parts. First, each node sends a copy of their distance vector table to each of its neighbour. Every node in the network does this in turn. Once all the updates have been sent and gathered, the nodes then begin to update their tables. New updates will not be sent out until all of the nodes have applied their updates. The NodeHandler enforces this behaviour

For updating the tables, each node first checks if any route in the received tables does not exist in theirs. If so, the new route is added. Following this, all the routes from all the nodes are compared, including the newly learned ones. The minimum distance for all the nodes is calculated and found from this. This is compared with the node's known direct connection neighbours and their distance. The shortest connection is  thenchosen. If this matches the node's distance-vector table, no changes are made and the node is considered stable. Otherwise, the distance vector table is updated to match this new route.

The node handler on each set of exchanges checks if every node is stable, i.e., their tables have remained the same. If any node in the network is unstable, the network is still considered unstable. This is used to decide if another series of exchanges should be sent. However, the node handler may also be given a limit for the number of exchanges to be sent, which stops exchanges being sent, even if the network has not converged. A value for infinity has also been implemented to help with this and can be set or changed to any value

The node handler can also print nodes routing tables, either during an update request or at the end. The node handler also traces the route from one node to another, using the successive distance vector tables of each node

The NodeHandler can be configured for split horizon. With this, when requesting the nodes to update their neighbours, they will remove any values in their distance vector table, where the link is their neighbour, before sending it to a neighbour.

2236693B – Conor Begley