# COMP3221 Assignment 1

## Network Topology: Briefly describe how you generate the network topology, using Visual aids or diagrams for clarity.

* Firstly, we add a specific number of nodes to the graph with the first node starting at the letter A.
* Secondly, since all the nodes are disconnected, we pick a random node in the connected graph (that has a degree of less than 3) which will, at first, be the first node in the disconnected graph. We then pick another random node in the disconnected graph and add an edge between the two chosen nodes. Finally, the node chosen from the disconnected graph is removed and this process is repeated until all nodes are connected.
* If the number of edges wanted is greater than the number of nodes, a second loop will happen to add edges between nodes at random.
* After all edges are assigned, they are then given a random weight from 0.1 to 9.00.
* A configuration file is then written for all nodes where it contains the degree of their node, it’s neighbours and the ports and weights associated with those nodes.

## Routing Algorithm: Outline the routing algorithm(s), explaining your selection and providing an overview of its functionality. Highlight any modifications or optimizations made to standard algorithms.

* Through implementing Djikstra’s algorithm, we were able to create a shortest path routing algorithm.
* Firstly, we set a node dictionary where it would start at a given node and will hold the least cost path to every other node as well as the nodes that are visited on the path.
* Secondly, the algorithm will find all the neighbours of the current node

## Implementation Methodology: Describe the programming approaches and tools used, with a focus on specific techniques applied to implement the routing protocols, handle link cost changes and node failures, and ensure continuous operation of the network.

## Simulation Results: Summarize key findings from your simulations, stating what you have and haven’t completed. Provide specific examples of how effective your system is in routing and adapting to link-cost changes and failures.