# **Lab 2 - Task 4**

In this task, we will learn how Dijkstra's algorithm used in the link-state routing can be used to determine the shortest path between any two nodes.

**INSTRUCTIONS**

Routing using the link-state algorithm is different than routing using the distance vector algorithm, because the link-state algorithm first estimates the graph of the complete network at every node. The information about the complete network is obtained by message flooding, where each node broadcasts information about its connections to its neighbors, and these neighbors then forward that information on and so on. Thus, information is propagated globally over the entire network, allowing each node to construct an estimate of the complete network. Given the graph of the complete network, each node then computes the shortest path to every other node in the network using Dijkstra's algorithm. It then uses this information to construct its routing/fowarding table. On the other hand, nodes in the distance vector algorithm never estimate the graph of the complete network. Although nodes know the cost of the route to get to each of the nodes through each of their neighbors, they do not know anything about the path taken by the packets after they arrive at their neighbors. The distance vector algorithm is a decentralized algorithm where each node broadcasts and updates its own routing table iteratively. However, this information is not re-broadcast explicitly, but rather only implicitly through changes in its neighbors' routing tables.

The MATLAB code in the window below uses the Dijkstra algorithm to compute the shortest path between two nodes. The implementation of the Dijkstra's algorithm is too complicated to be covered in this course. In this lab exercise, your task is to learn how to represent the network connection by a matrix and input the matrix to a function implementing Dijkstra's algorithm (**getPathDijkstra**).

In order to run Dijkstra's algorithm and get credit for this task, you must set the value of the variable **graph**, which is initialized to represent the networks used in Task 1 through 3, so that it represents the network shown in the figure below. We have already discussed the matrix representation of a graph in Task 2 and in the lab video. Once you have revised the current matrix, try to change the source and the destination nodes, and check whether the retrieved path is the one with the smallest cost.