NOTE: The nodes in the files are indexed starting from 0.

n.net – network topology file, n being the name of the network

File format:

Number of nodes

Number of links

Node-to-node distance matrix in km.

n.pat – routing paths file, n being the name of the network

For each pair of nodes, there are k=30 candidate paths. But in the experiments, different values of k (candidate paths, k=2, k=3, etc.), can be considered so, for example, for k=2 out of 30 paths, the first 2 paths are selected, and so on.

For a network with *n* nodes and *e* links (numbered according to the order in the network topology file), each path is encoded as a sequence of binary values, where 1 means that the link belongs to the path and 0 means that the link does not belong to the path.

File format:

Number of all paths

Path 1 for node pair (0,1)

Path 2 for node pair (0,1)

…

Path 30 for node pair (0,1)

Path 1 for node pair (0,2)

Path 2 for node pair (0,2)

…

Path 30 for node pair (0,2)

…

Path 1 for node pair (2,0)

…

Path 30 for node pair (*n, n-1*)

…

xxx.txt – unicast demand file, dynamic time-varying traffic

Each file is one connection request (demand) to be provisioned in the network

File format:

Source node

Destination node

Traffic class (in this project to be ignored)

Bitrate in successive iterations

Please investigate the performance of the proposed algorithm for different traffic volumes. The easiest way to do this is by running simulations for different numbers of requests. That is, run, for example, 5 simulations in a row, each time loading more requests, e.g., for the first simulation 100, for the second 125, for the third 150, and so on. Each simulation run gives a numerical result and at the end interesting trends can be observed for different algorithms.

The experiments should be repeated for the two given topologies. The files for each topology are in a separate folder. For each topology, please run the experiments for all given sets of requests and report the averaged results.