

# **Scilab – Routing and Congestion Control algorithms**

## **Task 1**

- Create a network with 100 nodes.
- Increment from 5 nodes to 100 nodes in the interval of 5 nodes .
- Visualize the created networks using graph
- Perform routing using Bellman Ford Algorithm for the created networks.
- Perform routing using Dijkstra's algorithm for the created networks.
- Calculate the duration required for routing for both the algorithms for both the algorithms for all the node increments.
- Plot both the durations in a graph.

Ref : Appendix 1 - Model Scilab Program for routing

## **Task 2**

- Create a network with 200 nodes
- Increase the number of nodes of the above network to 300 nodes
- Visualize both the networks
- Perform Congestion control using ARC algorithm or any other ARC based algorithm for both the networks
- Calculate the duration required for congestion control for both the topologies
- Plot both the durations in a graph.
- Generate a network of 500 nodes using any five methods .
- Perform congestion map with in the nodes of the network with 500 nodes.
- Reduce the nodes to 400, 300, 200 and 100
- Perform congestion map with in the nodes of the network
- Calculate the duration of all the five types of networks and for all the types of methods
- Plot it in a graph.

Ref : Scilab – NARVAL help pages

## Appendix 1 - Model Scilab Program for routing

```
//Author:
//Dr. T. Subbulakshmi, Professor
//School of Computing Science and Engineering
//VIT Chennai
//Code to compute the CPU time taken by the FloydWarshall routing algorithm to return the
//Vector of the total distance between each network node and the source node and
//Vector composed by the predecessor of each node in order to reach the source
//node in respect with the shortest path for various network sizes.

//Clear display and environment variables

clc;
clear all;

//Creating a random topology which is tiny in size
k = 1;
j = 1;
for n=10:10:160
    b(k) = n;
    L=1000;           //network square area side is 1000
    dmax=100;         //locality radius is 100
    // networkname = "Routing using Floyd Warshall Algorithm";
    [g]=NL_T_LocalityConnex(n,L,dmax); //generation of a random topology in respect with the Locality
    method.
    i=NL_F_RandInt1n(length(g.node_x)); //selection of the source node
    ind=1;           //window index
    g.node_diam(i)=40; //node diameter
    g.node_border(i)=10; //node border
    g.node_color(i)=5; //node color
    [f]=NL_G_ShowGraphN(g,ind); //graph visualization
    //Application of the FloydWarshall algorithm
    for i = 1:5      //Run 5 iterations
        timer();    //Initialize timer
        [dist,pred]=NL_R_FloydWarshall(g); //application of NL_R_FloydWarshall
        A(i) = timer() //Store timer value in array
    end
    c(j)=mean(A);    //Calculate average time taken
    disp(c(j),n,"Time of finding the shortest path for nodes",) //Display average time
    j = j+1;
    k = k+1;
end
clf();
for x = c
    for y = b
        disp(x,y);
        plot(b,c,'--mo');
        xtitle('Time of Computation for Floyd Warshall Algorithm', 'numberofnodes', 'time', boxed = %t );
    end
end
end
```