

EXPERIMENT 2

SCENARIO -

The lab's floor plan has 16 network access points that must be connected to a central core switch and two backup distribution switches. The network engineer decides to prototype the layout in Scilab using the NL_G toolbox and implements the following logical design:

- Nodes 1–6 form a ring backbone connecting the main wiring closets on each side of the building.
- Nodes 7–9 are a star of wireless access points connected to node 2, which acts as a local distribution switch for one wing.
- Nodes 10–12 are another star of access points connected to node 5, serving the opposite wing.
- Nodes 13–16 are four core/distribution switches in the server room; they are connected in a partial mesh so that even if one link fails, traffic to the data center can still be rerouted.

The engineer uses the given Scilab script to:

- Plot the positions of all 16 nodes and visualize the hybrid Ring + Dual Star + Mesh topology.
- Display node and edge indices to document each physical cable run.
- Highlight the ring nodes (1–6) and mesh edges between the core switches (13–16) for analysis.
- Compute the total number of nodes and edges, and calculate every edge's physical length

based on its (X,Y) coordinates, which approximates cable length in meters

CODE -

```
clear;
```

```
NameOfNetwork = 'Hybrid_Ring_Star_Mesh';
```

```
NumberOfNodes = 16;
```

```
X = [400 500 600 600 500 400, ...  
     300 250 350, ...  
     700 750 650, ...  
     300 500 700 500];
```

```
Y = [500 600 500 400 300 400, ...  
     600 650 650, ...  
     600 650 650, ...  
     200 150 200 250];
```

```
head = [1 2 3 4 5 6, ...  
        2 2 2, ...  
        5 5 5, ...  
        13 13 14 15];
```

```
tail = [2 3 4 5 6 1, ...  
        7 8 9, ...  
        10 11 12, ...  
        14 15 16 16];
```

```
G = NL_G_MakeGraph(NameOfNetwork, NumberOfNodes, head, tail, X, Y);
```

```
NL_G_ShowGraph(G,1);
```

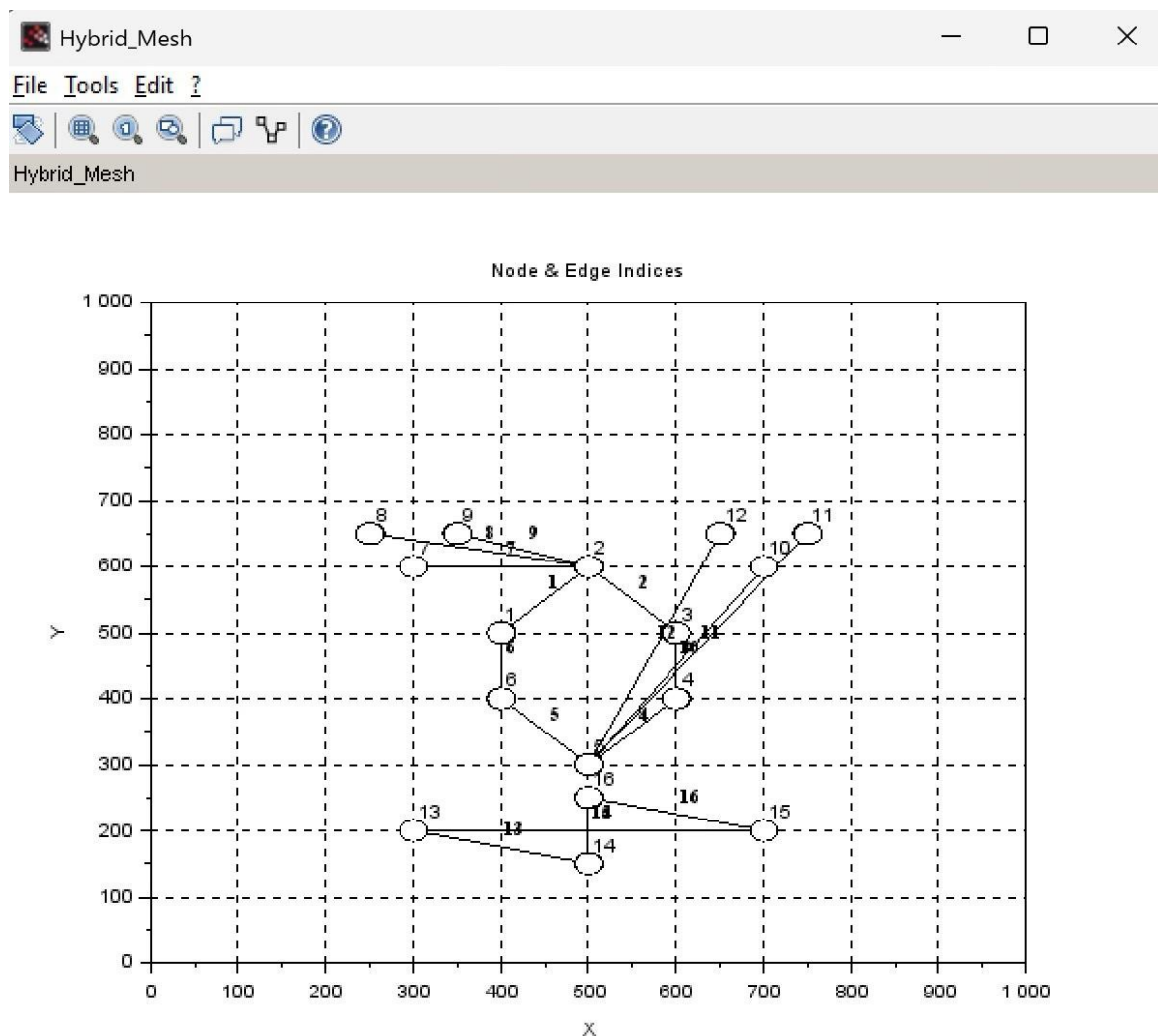
```
xtitle("Hybrid Ring + Dual Star + Mesh", "X", "Y");
```

```
NL_G_ShowGraphNE(G,2);
```

```
xtitle("Node & Edge Indices", "X", "Y");
```

```
NL_G_HighlightNodes(G,1:6,30,8,20,3);
```

```
meshEdges = [];  
meshEdges = [meshEdges NL_G_Nodes2Edge(G,13,14)];  
meshEdges = [meshEdges NL_G_Nodes2Edge(G,13,15)];  
meshEdges = [meshEdges NL_G_Nodes2Edge(G,14,16)];  
meshEdges = [meshEdges NL_G_Nodes2Edge(G,15,16)];  
  
NL_G_HighlightEdges(G, meshEdges, 5, 6, 4);  
  
[nnodes, nedges] = NL_G_GraphSize(G);  
disp("Number of nodes: "+string(nnodes));  
disp("Number of edges: "+string(nedges));  
  
len = NL_G_EdgesLength(G.node_x, G.node_y, G.head, G.tail);  
disp(len);
```



Output -

Github repo - <https://github.com/Dhruv-Kr-Mittal/CN-lab/tree/main/Lab2>