EXP2: Network Topology Creation In Scilab

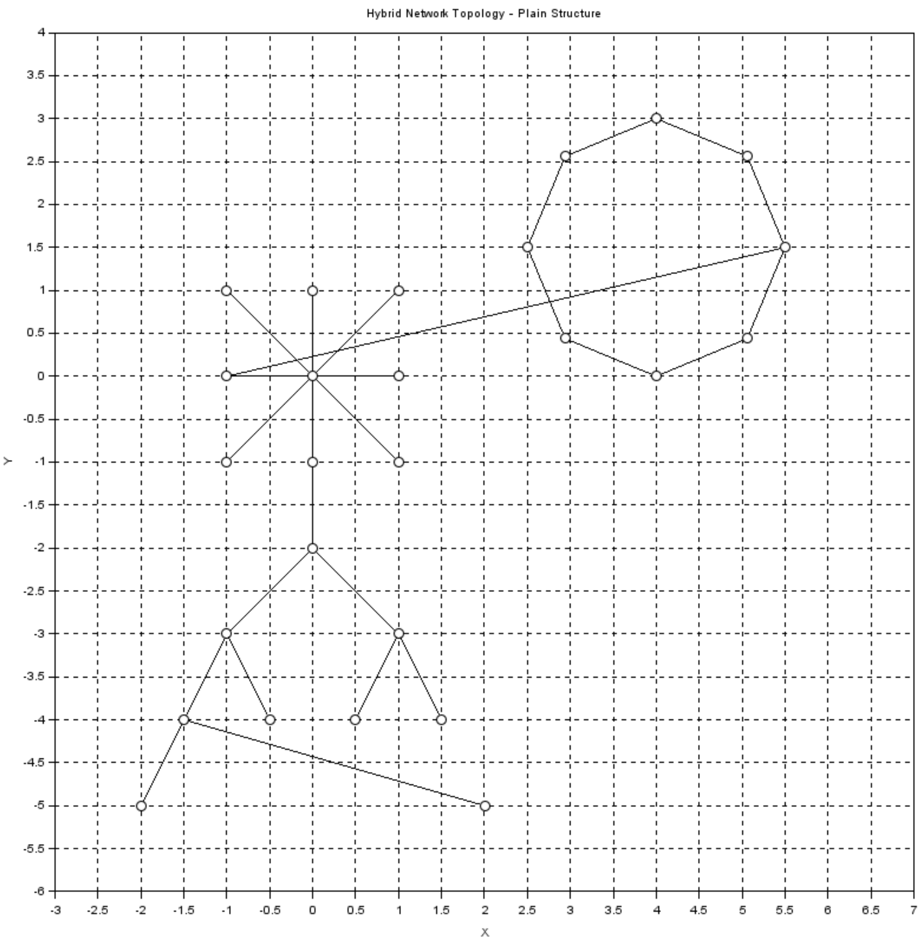
NAME: SRIVARSHINI S
ROLL NUMBER: 24BCE1929

Aim: To design and visualize a hybrid network topology using Scilab by combining Star, Tree, and Ring topologies, and to analyze the network by displaying node and edge details, colouring components, and computing network parameters such as node degrees and total edges.

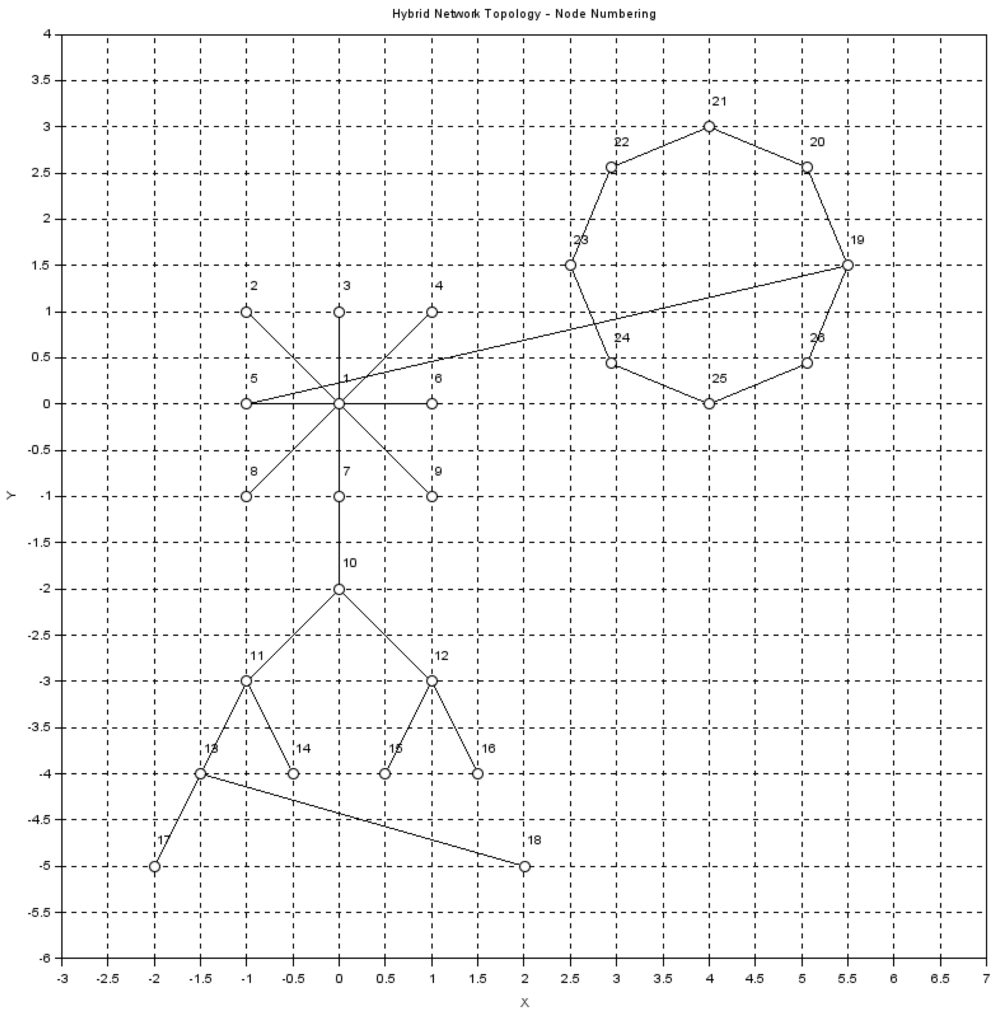
Tools Used: Scilab (Version 6.x), Scilab Graphics Library, Windows Operating System, Basic concepts of Graph Theory, Adjacency Matrix representation.

PART 1

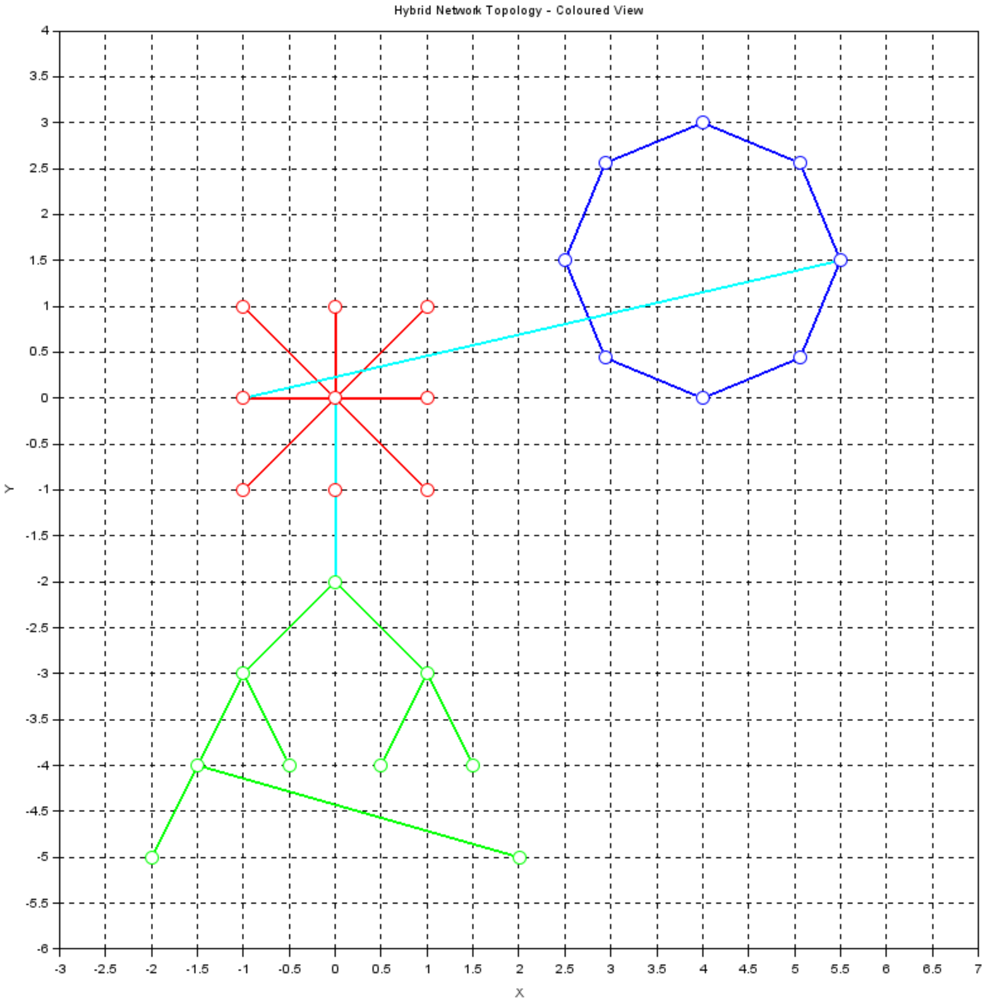
A. Display the created topology



B. Number the nodes and edges



C. Colour the nodes and edges.



D. Print the number of edges every node has along with the node with maximum edges.

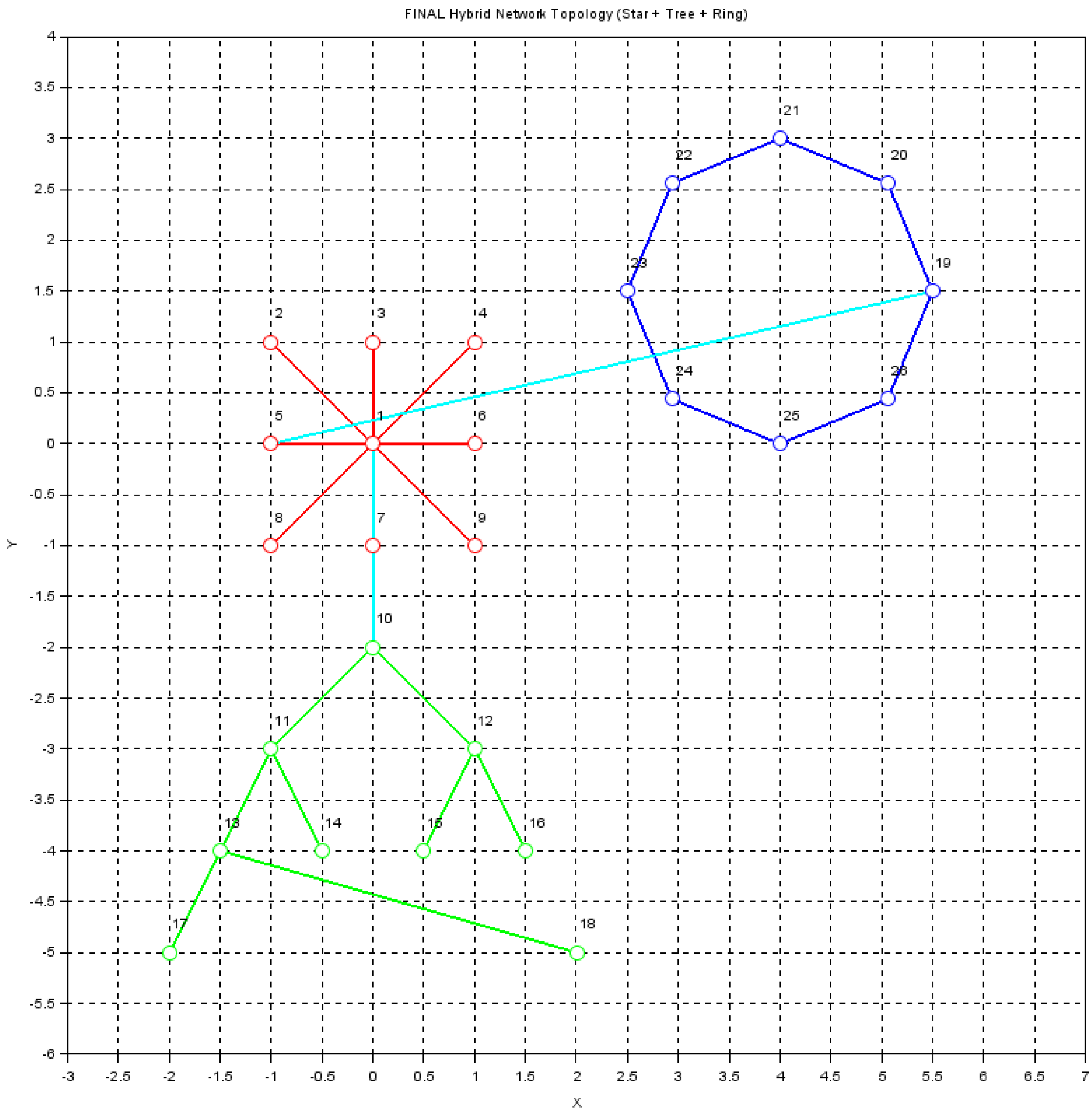
```
 ===== NODE DEGREES =====
Node  1 :  9 edges
Node  2 :  1 edges
Node  3 :  1 edges
Node  4 :  1 edges
Node  5 :  2 edges
Node  6 :  1 edges
Node  7 :  1 edges
Node  8 :  1 edges
Node  9 :  1 edges
Node 10 :  3 edges
Node 11 :  3 edges
Node 12 :  3 edges
Node 13 :  3 edges
Node 14 :  1 edges
Node 15 :  1 edges
Node 16 :  1 edges
Node 17 :  1 edges
Node 18 :  1 edges
Node 19 :  3 edges
Node 20 :  2 edges
Node 21 :  2 edges
Node 22 :  2 edges
Node 23 :  2 edges
Node 24 :  2 edges
Node 25 :  2 edges
Node 26 :  2 edges

Node with maximum edges: Node 1 with 9 edges
```

E. Print the total number of nodes and edges.

```
Total number of nodes : 26
Total number of edges : 26
```

FINAL STRUCTURE OF MY HYBRID NETWORK TOPOLOGY



THE CODE

```
clear;
clc;

// ----------------------------------------------
// HYBRID NETWORK : STAR + TREE + RING (26 NODES)
// ----------------------------------------------

n = 26;
```

```
adj = zeros(n, n);

// ---------------- STAR (1–9) ----------------
for i = 2:9
    adj(1, i) = 1;
    adj(i, 1) = 1;
end

// ---------------- TREE (10–18) ----------------
adj(10,11)=1; adj(11,10)=1;
adj(10,12)=1; adj(12,10)=1;

adj(11,13)=1; adj(13,11)=1;
adj(11,14)=1; adj(14,11)=1;

adj(12,15)=1; adj(15,12)=1;
adj(12,16)=1; adj(16,12)=1;

adj(13,17)=1; adj(17,13)=1;
adj(13,18)=1; adj(18,13)=1;

// ---------------- RING (19–26) ----------------
for i = 19:25
    adj(i, i+1) = 1;
    adj(i+1, i) = 1;
end
adj(26,19) = 1;
adj(19,26) = 1;

// ---------------- HYBRID LINKS ----------------
adj(1,10) = 1; adj(10,1) = 1;
adj(5,19) = 1; adj(19,5) = 1;

// ------------------------------------------------
// NODE COORDINATES
// ------------------------------------------------

x = [];
y = [];

// STAR
x = [x 0 -1 0 1 -1 1 0 -1 1];
y = [y 0  1 1 1  0 0 -1 -1 -1];

// TREE
x = [x 0 -1 1 -1.5 -0.5 0.5 1.5 -2 2];
y = [y -2 -3 -3 -4 -4 -4 -4 -5 -5];
```

```
// RING
for k = 0:7
    angle = k * 2 * %pi / 8;
    x = [x 4 + 1.5*cos(angle)];
    y = [y 1.5 + 1.5*sin(angle)];
end

// ============================================================
// WINDOW 1 : PLAIN TOPOLOGY
// ============================================================
scf(1); clf;

for i = 1:n
    for j = i+1:n
        if adj(i,j) == 1 then
            plot([x(i) x(j)], [y(i) y(j)], 'k-', 'thickness', 1);
        end
    end
end

for i = 1:n
    plot(x(i), y(i), 'ko', 'marksize', 8);
end

title("Hybrid Network Topology – Plain Structure");
xlabel("X"); ylabel("Y");

a = gca();
a.isoview = "on";
a.data_bounds = [-2.5 -5.5; 6.5 3.5];
xgrid();

// ============================================================
// WINDOW 2 : NODE NUMBERING
// ============================================================
scf(2); clf;

for i = 1:n
    for j = i+1:n
        if adj(i,j) == 1 then
            plot([x(i) x(j)], [y(i) y(j)], 'k-', 'thickness', 1);
        end
    end
end

for i = 1:n
    plot(x(i), y(i), 'ko', 'marksize', 8);
    xstring(x(i), y(i)+0.15, string(i));
```

```
end

title("Hybrid Network Topology – Node Numbering");
xlabel("X"); ylabel("Y");

a = gca();
a.isoview = "on";
a.data_bounds = [-2.5 -5.5; 6.5 3.5];
xgrid();

// ============================================================
// WINDOW 3 : COLOURED TOPOLOGY
// ============================================================
scf(3); clf;

// STAR
for i = 2:9
    plot([x(1) x(i)], [y(1) y(i)], 'r-', 'thickness', 2);
end

// TREE
plot([x(10) x(11)], [y(10) y(11)], 'g-', 'thickness', 2);
plot([x(10) x(12)], [y(10) y(12)], 'g-', 'thickness', 2);
plot([x(11) x(13)], [y(11) y(13)], 'g-', 'thickness', 2);
plot([x(11) x(14)], [y(11) y(14)], 'g-', 'thickness', 2);
plot([x(12) x(15)], [y(12) y(15)], 'g-', 'thickness', 2);
plot([x(12) x(16)], [y(12) y(16)], 'g-', 'thickness', 2);
plot([x(13) x(17)], [y(13) y(17)], 'g-', 'thickness', 2);
plot([x(13) x(18)], [y(13) y(18)], 'g-', 'thickness', 2);

// RING
for i = 19:25
    plot([x(i) x(i+1)], [y(i) y(i+1)], 'b-', 'thickness', 2);
end
plot([x(26) x(19)], [y(26) y(19)], 'b-', 'thickness', 2);

// HYBRID
plot([x(1) x(10)], [y(1) y(10)], 'c-', 'thickness', 2);
plot([x(5) x(19)], [y(5) y(19)], 'c-', 'thickness', 2);

// nodes
for i = 1:9
    plot(x(i), y(i), 'ro', 'marksize', 10);
end
for i = 10:18
    plot(x(i), y(i), 'go', 'marksize', 10);
end
for i = 19:26
```

```
    plot(x(i), y(i), 'bo', 'marksize', 10);
end

title("Hybrid Network Topology – Coloured View");
xlabel("X"); ylabel("Y");

a = gca();
a.isoview = "on";
a.data_bounds = [-2.5 -5.5; 6.5 3.5];
xgrid();

// ==========================================================
// FINAL WINDOW : COLOURED + NUMBERED HYBRID TOPOLOGY
// ==========================================================
scf(4);
clf;

// ---------------- STAR EDGES (RED) ----------------
for i = 2:9
    plot([x(1) x(i)], [y(1) y(i)], 'r-', 'thickness', 2);
end

// ---------------- TREE EDGES (GREEN) ----------------
plot([x(10) x(11)], [y(10) y(11)], 'g-', 'thickness', 2);
plot([x(10) x(12)], [y(10) y(12)], 'g-', 'thickness', 2);
plot([x(11) x(13)], [y(11) y(13)], 'g-', 'thickness', 2);
plot([x(11) x(14)], [y(11) y(14)], 'g-', 'thickness', 2);
plot([x(12) x(15)], [y(12) y(15)], 'g-', 'thickness', 2);
plot([x(12) x(16)], [y(12) y(16)], 'g-', 'thickness', 2);
plot([x(13) x(17)], [y(13) y(17)], 'g-', 'thickness', 2);
plot([x(13) x(18)], [y(13) y(18)], 'g-', 'thickness', 2);

// ---------------- RING EDGES (BLUE) ----------------
for i = 19:25
    plot([x(i) x(i+1)], [y(i) y(i+1)], 'b-', 'thickness', 2);
end
plot([x(26) x(19)], [y(26) y(19)], 'b-', 'thickness', 2);

// ---------------- HYBRID LINKS (CYAN) ----------------
plot([x(1) x(10)], [y(1) y(10)], 'c-', 'thickness', 2);
plot([x(5) x(19)], [y(5) y(19)], 'c-', 'thickness', 2);

// ---------------- NODES ----------------
// Star nodes
for i = 1:9
    plot(x(i), y(i), 'ro', 'marksize', 10);
end
// Tree nodes
```

```
for i = 10:18
    plot(x(i), y(i), 'go', 'marksize', 10);
end
// Ring nodes
for i = 19:26
    plot(x(i), y(i), 'bo', 'marksize', 10);
end

// ---------------- NODE NUMBERS ----------------
for i = 1:n
    xstring(x(i), y(i)+0.15, string(i));
end

title("FINAL Hybrid Network Topology (Star + Tree + Ring)");
xlabel("X");
ylabel("Y");

// ---------------- AXIS & GRID ----------------
a = gca();
a.isoview = "on";
a.data_bounds = [-2.5 -5.5; 6.5 3.5];
xgrid();

// ==================================================
// DEGREE + SUMMARY
// ==================================================
degrees = sum(adj, 2);
[max_degree, max_node] = max(degrees);

disp("===== NODE DEGREES =====");
for i = 1:n
    mprintf("Node %2d : %2d edges\n", i, degrees(i));
end

mprintf("\nNode with maximum edges: Node %d with %d edges\n", max_node,
max_degree);

disp(" ");
mprintf("Total number of nodes : %d\n", n);
mprintf("Total number of edges : %d\n", sum(adj)/2);
```

PART 2 – SCENARIO BASED QUESTIONS

SCENARIO

A university campus network uses a hybrid topology to manage communication efficiently. The administrative block follows a Star topology for centralized control, the academic departments use a Tree topology for hierarchical communication, and the student hostels are connected using a Ring topology for equal access and reliability. Hybrid links connect all these topologies to form a single integrated network.

QUESTIONS

1. Explain how data is transmitted from a student hostel node to a department laboratory node.

Data from the hostel node first travels through the Ring topology to the hybrid connection, then passes through the central Star node, and finally moves through the Tree topology hierarchy to reach the department laboratory node.

---

2. Identify the node with the maximum number of connections and justify its importance.

The central node of the Star topology has the maximum number of connections. It is important because it acts as the main communication hub, enabling data exchange between all parts of the network.

---

3. State one advantage of using a hybrid topology for a campus network.

A hybrid topology provides scalability and reliability by combining the advantages of different topologies, making it suitable for large and complex networks like university campuses.

GITHUB REPOSITORY LINK

All source code related to the hybrid network topology implementation has been uploaded to GitHub.

📎 **GitHub Repository:** GITHUB LINK

(The repository contains the Scilab source code and supporting files used for creating and visualizing the hybrid topology.)

CONCLUSION

In this experiment, a hybrid network topology combining **Star, Tree, and Ring** topologies was successfully created and visualized using Scilab. The use of an adjacency matrix allowed efficient representation and analysis of network connections. Separate visualizations were generated to display the plain structure, node numbering, coloured topology, and a final combined view.