Scilab Manual for Computer Networks by Dr Subbulakshmi T Computer Engineering VIT University Chennai[1]

Solutions provided by
Dr Subbulakshmi T
Computer Engineering
Vit University Chennai

May 19, 2022

[1]Funded by a grant from the National Mission on Education through ICT, http://spoken-tutorial.org/NMEICT-Intro. This Scilab Manual and Scilab codes written in it can be downloaded from the "Migrated Labs" section at the website http://scilab.in

Contents

List of Experiments

List of Figures

Experiment: 1

Creation of the Bus topology

Scilab code Solution 1.1 Bus Topology Creation

```
1 //Experiment No.1
2 // This file must be used under the terms of the
    CeCILL.
3 // This source file is licensed as described in the
    file COPYING, which
4 // you should have received as part of this
    distribution. The terms
5 // are also available at
6 // http://www.cecill.info/licences/Licence_CeCILL_V2
    -en.txt
7 //This Source file is Written by Dr. T. Subbulaskhmi
    , Professor,
8 //School of Computing Science and Engineering, VIT
    University Chennai
9 //using the NARVAL examples of Scilab for Network
    Topology Creation
10 //The Operating System used for writing the code
```

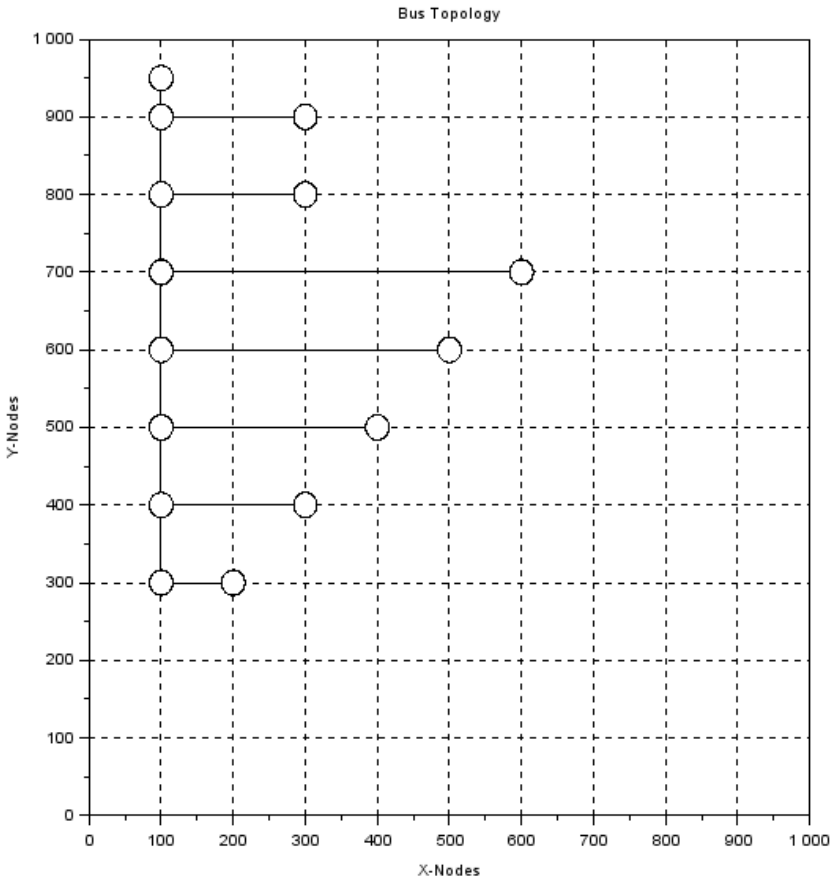

Figure 1.1: Bus Topology Creation

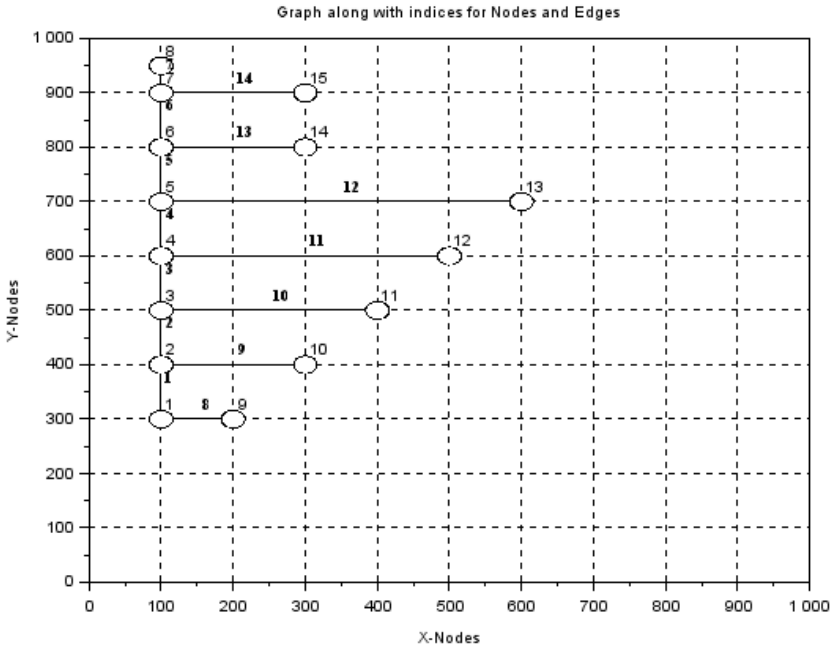

Figure 1.2: Bus Topology Creation

```
    found in this file is Windows 8
11 //SCILAB version 5.5.2 and NARVAL toolbox version
    3.1
12 //
   |-------------------------------------------------------------------------

13 //||This lab experiment will do the following
    operations                                |
14 //||1.create and Display Bus Topology
                                              |
15 //||2.Colour the nodes and Edges
                                                 |
16 //||3.Display the node number and edge number
                                             |
17 //||4.Display the number of nodes and edges
                                               |
18 //||5.Display the Edge Indexbetween two nodes
                                             |
19 //||6.Display the length of every edge
                                                |
20 //||7.Return the set of edges connected to a node
    inside a graph                      |
21 //||8.Extract the data fields of the node
                                              |
22 //||9.Extract the data fields of the Edge
                                              |
23 //
   |-------------------------------------------------------------------------

24 clear;
25 clc;
26 //1. create and Display Bus Topology
27 NameOfNetwork='Bus Topology';// Name of your network
28 NumberOfNodes=15;//Number of Nodes in the network
29 StartingNodesOfConnection=[1 2 3 4 5 6 7 1 2 3 4 5 6
    7]; //Starting Nodes of the connection lines
30 EndingNodesOfConnection=[2 3 4 5 6 7 8 9 10 11 12 13
    14 15]; //Ending Node of the connection
```

```
31 XCoordinatesOfNodes=[100 100 100 100 100 100 100 100
       200 300 400 500 600 300 300]; // X-Coordinates
      of the nodes
32 YCoordinatesOfNodes=[300 400 500 600 700 800 900 950
       300 400 500 600 700 800 900]; // Y-Coordinates
      of the nodes
33 [TopologyGraph]=NL_G_MakeGraph(NameOfNetwork,
      NumberOfNodes,StartingNodesOfConnection,
      EndingNodesOfConnection,XCoordinatesOfNodes,
      YCoordinatesOfNodes)//Creates the Bus topoplogy
34 WindowIndex=1; //Graph Window Number
35 [Graphparameters] = NL_G_ShowGraph(TopologyGraph,
      WindowIndex);// Visualize the Graph along with
      indices for Nodes and Edges
36 xtitle("Bus Topology","X-Nodes","Y-Nodes");
37 //2. Colour the nodes and Edges
38 NodeColor=30; // Node Colour 2:[Blue],3:[Green], 5:[
      Red]
39 BorderThickness=10; // Node Border thickness
40 NodeDiameter=25; //Node diameter
41 WindowIndex=2;//window index
42 ListOfNodes=[1 2 4 6 8 10 12 14];//list of nodes
43 [graph1,nodes]=NL_G_HighlightNodes(TopologyGraph,
      ListOfNodes,NodeColor,BorderThickness,
      NodeDiameter,WindowIndex);//Highlight the
      specific nodes mentioned in the 'nodes' vector
44 xtitle("Highlight the specific nodes mentioned in
      the nodes vector","X-Nodes","Y-Nodes");
45 NodeColor=5;// Edge Colour
46 EdgeWidth=5;//Edge Width
47 WindowIndex=3;//window index
48 ListOfEdges=[1 5 7];//list of edges
49 [graph2,nodes]=NL_G_HighlightEdges(TopologyGraph,
      ListOfEdges,NodeColor,EdgeWidth,WindowIndex);//
      Highlight the specific nodes mentioned in the '
      edges' vector
50 xtitle("Highlight the specific nodes mentioned in
      the  edges vector","X-Nodes","Y-Nodes");
```

```
51 //3. Display the node number and edge number
52 WindowIndex=4; //Graph Window Number
53 [GraphVisualise] = NL_G_ShowGraphNE(TopologyGraph,
      WindowIndex);// Visualize the Graph along with
      indices for Nodes and Edges
54 xtitle("Graph along with indices for Nodes and Edges
      ","X-Nodes","Y-Nodes");
55 //4. Display the number of nodes and edges
56 [ExtractedNode,ExtractedEdge]=NL_G_GraphSize(
      TopologyGraph);//Extract the number of nodes and
      edges
57 disp('Number of nodes:',ExtractedNode); //display
      the number of nodes and edges
58 disp('Number of edges:',ExtractedEdge);
59
60 //5.Display the Edge Index between two nodes
61 StartingNode=4;//starting node
62 EndingNode=5;//ending node
63 [GetEdgeIndex]=NL_G_Nodes2Edge(TopologyGraph,
      StartingNode,EndingNode);//Get the Edge Index
64 disp('Index of Edge across nodes 4 and 5:',
      GetEdgeIndex);
65
66 //6. Display the length of every edge
67 [EdgeLength]=NL_G_EdgesLength(TopologyGraph.node_x,
      TopologyGraph.node_y,TopologyGraph.head,
      TopologyGraph.tail);//Display the length of every
       edge
68 disp('length of all edges',EdgeLength);
69
70 //7. Return the set of edges connected to a node
      inside a graph
71 NodeIndex=2;//node index
72 [GetEdgeIndex]=NL_G_EdgesOfNode(TopologyGraph,
      NodeIndex);//application of NL_G_EdgesOfNode
73 disp('Internal Edges of node 2:',GetEdgeIndex);
74
75 //8. Extract the data fields of the node
```

```
76 [node_x,node_y,node_border,node_diameter,node_color,
     node_number]=NL_G_NodeDataFields(TopologyGraph);
     // Extracting the node data fields
77 disp('Node x :',XCoordinatesOfNodes); // X
     Coordinates
78 disp('Node y :', YCoordinatesOfNodes);// Y
     Coordinates
79 disp('Node Borders :', node_border); //Node Border
80 disp('Node Diameter :', node_diameter); // Diameter
     of the node
81 disp('Node Colour :', node_color); // Node colour
82 disp('Node Number :', node_number); //Number of
     Nodes
83
84 //9.Extract the data fields of the Edge
85 [e_head,e_tail,e_color,e_width,e_length,e_weight,
     e_number]=NL_G_EdgeDataFields(TopologyGraph)//
     application of NL_G_EdgeDataFields
86 disp('Head Nodes',e_head); //Head node details
87 disp('Tail Nodes',e_tail); // Tail Node details
88 disp('Edge Colours',e_color); // Colour of edges
89 disp('Edge Widths',e_width); // Width of edges
90 disp('Edge Lengths',e_length); // Length of the
     edges
91 disp('Edge Weights',e_weight); //Weight of the edges
92 disp('Edge Numbers',e_number); // Edge Number
```

Experiment: 2

Creation of the Ring topology

Scilab code Solution 2.1 Ring Topology Creation

```
1
2
3 //Experiment No.2
4 // This file must be used under the terms of the
    CeCILL.
5 // This source file is licensed as described in the
    file COPYING, which
6 // you should have received as part of this
    distribution. The terms
7 // are also available at
8 // http://www.cecill.info/licences/Licence_CeCILL_V2
    -en.txt
9 //This Source file is Written by Dr. T. Subbulaskhmi
    , Professor,
10 //School of Computing Science and Engineering, VIT
    University Chennai
11 //using the NARVAL examples of Scilab for Network
```

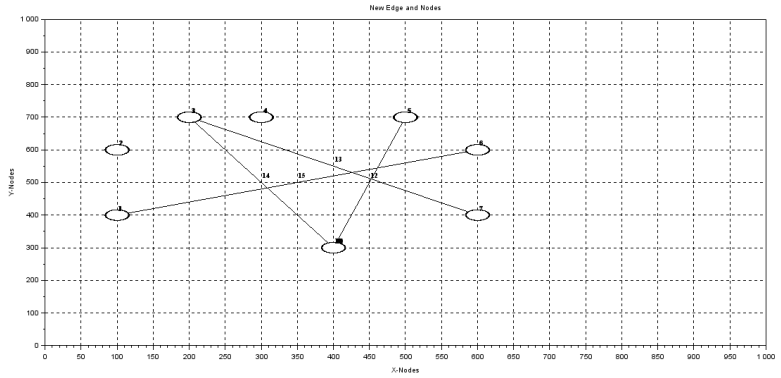

Figure 2.1: Ring Topology Creation

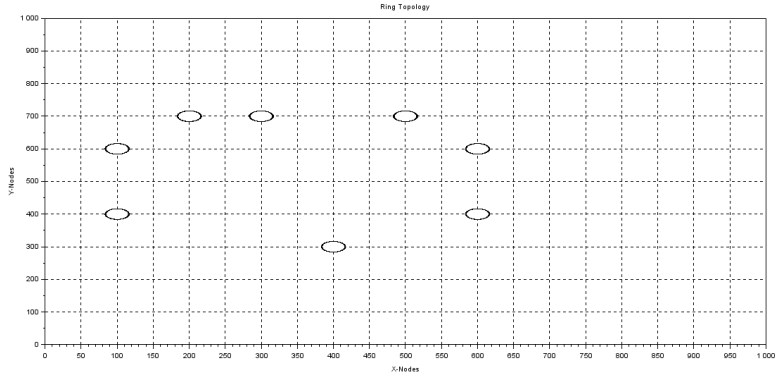

Figure 2.2: Ring Topology Creation

```
    Topology Creation
12 //The Operating System used for writing the code
    found in this file is Windows 8
13 //SCILAB version 5.5.2 and NARVAL toolbox version
    3.1
14 //Program to 1. create and Display Ring Topology 2.
    Colour the nodes 3. Display the node number and
    edge number 4. Display the number of nodes and
    edges 5. insert new nodes in the specified edge
    6. insert new edges 7. Display the number of
    nodes and edges again after adding the nodes and
    edges.
15
16 clear;
17 clc;
18
19 //1. create and Display Ring Topology
20 NameOfNetwork='Ring Topology';// Name of your
    network
21 NumberOfNodes=8;//Number of Nodes in the network
22 ConnectionEndingNode=[1 2 3 4 5 6 7 8]; //Ending
    Nodes of the connection lines
23 ConnectionStartingNode=[2 3 4 5 6 7 8 1];//Starting
    Nodes of the connection lines
24 XCoordinatedOfNodes=[100 100 200 300 500 600 600
    400]; // X-Coordinates of the nodes
25 YCoordinatedOfNodes=[400 600 700 700 700 600 400
    300]; // Y-Coordinates of the nodes
26 [TopologyGraph]=NL_G_MakeGraph(NameOfNetwork,
    NumberOfNodes,ConnectionEndingNode,
    ConnectionEndingNode,XCoordinatedOfNodes,
    YCoordinatedOfNodes)//Creates the Ring topoplogy
27
28 WindowIndex=1;//Graph Window Number
29
30 [VisualizeGraph1] = NL_G_ShowGraph(TopologyGraph,
    WindowIndex);// Visualize the Graph
31
```

```
32 //2. Colour the nodes
33 NodeColor=30; // Node Colour 2:[Blue],3:[Green], 5:[
     Red]
34 BorderThickness=10; // Node Border thickness
35 NodeDiameter=25; //Node diameter
36 WindowIndex=2;//window index
37 ListOfNodes=[1 3 5 7];//list of nodes
38 [NodeHighlight,VisualizeGraph1]=NL_G_HighlightNodes(
     TopologyGraph,ListOfNodes,NodeColor,
     BorderThickness,NodeDiameter,WindowIndex);//
     Highlight the specified nodes
39
40 WindoeIndex=3; //Graph Window Number
41 [VisualizeGraph2] = NL_G_ShowGraphNE(TopologyGraph,
     WindoeIndex);// Visualize the Graph along with
     indices for Nodes and Edges
42
43 // 4. Display the number of nodes and edges
44 [ExtractNode,ExtractEdge]=NL_G_GraphSize(
     TopologyGraph);//Extract the number of nodes and
     edges
45 disp(ExtractNode,ExtractEdge); //display the number
     of nodes and edges
46
47 //5. insert new nodes in the specified edge
48 EdgeIndex=8;//edge index
49 NewNodeQuantity=3;//quantity of new nodes
50 [go]=NL_G_SplitEdge(TopologyGraph,EdgeIndex,
     NewNodeQuantity);//application of NL_G_SplitEdge
51 WindowIndex=3;//window index
52 VisualizeGraph1=NL_G_ShowGraphNE(go,WindowIndex)
53
54 //6. insert new edges
55 NewEdgeHeadVector=[9 7 8 1];//head vector of new
     edges
56 NewEdgeTailVector=[5 3 3 6];//tail vector of new
     edges
57 NewEdgeNameVector=['e5' 'e6' 'e7' 'e8'];//name
```

```
    vector of new edges
58 [TopologyGraph]  = NL_G_AddEdges(go,
    NewEdgeHeadVector,NewEdgeTailVector,
    NewEdgeNameVector);
59 WindowIndex=4;
60 VisualizeGraph1=NL_G_ShowGraphNE(TopologyGraph,
    WindowIndex);
61
62 // 7. Display the number of nodes and edges again
    after adding the nodes and edges.
63 [ExtractNode,ExtractEdge]=NL_G_GraphSize(
    TopologyGraph);//Extract the number of nodes and
    edges
64 disp(ExtractNode,ExtractEdge); //display the number
    of nodes and edges
```

Experiment: 3

Creation of the Star topology

Scilab code Solution 3.1 Star Topology Creation

```
1 //Experiment No.3
2 // This file must be used under the terms of the
    CeCILL.
3 // This source file is licensed as described in the
    file COPYING, which
4 // you should have received as part of this
    distribution. The terms
5 // are also available at
6 // http://www.cecill.info/licences/Licence_CeCILL_V2
    -en.txt
7 //This Source file is Written by Dr. T. Subbulaskhmi
    , Professor,
8 //School of Computing Science and Engineering, VIT
    University Chennai
9 //using the NARVAL examples of Scilab for Network
    Topology Creation
10 //The Operating System used for writing the code
```

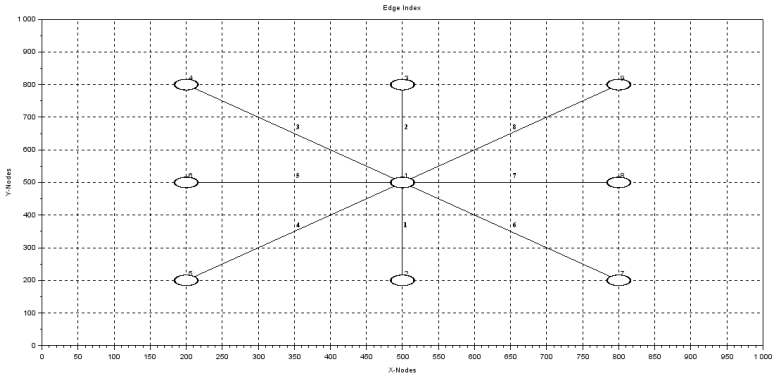

Figure 3.1: Star Topology Creation

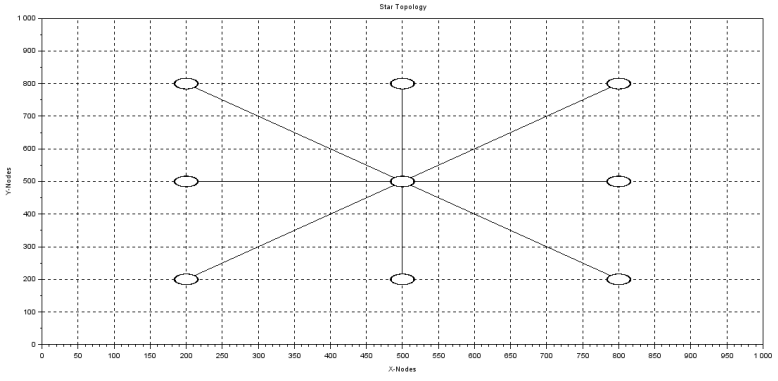

Figure 3.2: Star Topology Creation

```
    found in this file is Windows 8
11 //SCILAB version 5.5.2 and NARVAL toolbox version
    3.1//Program to 1. Create and Visualize the Star
    Topology // 2. Colour the nodes and visualize 3.
    Display the node number and edge number 4.
    Display the number of nodes and edges
12
13 clear;
14 clc;
15
16 //1. Create and Visualize the Star Topology
17 NameOfNetwork='Star Topology';// Name of your
    network
18 NumberOfNodes=9;//Number of Nodes in the network
19 EndingNodesOfEdge=[1 1 1 1 1 1 1 1];//Ending Nodes
    of the connection lines
20 StartNodesOfEdge=[2 3 4 5 6 7 8 9];//Starting Nodes
    of the connection lines
21 XCoordinatesOfNodes=[500 500 500 200 200 200 800 800
     800];// X-Coordinates of the nodes
22 YCoordinatesOfNodes=[500 200 800 800 200 500 200 500
     800]; // Y-Coordinates of the nodes
23 [TopologyGraph]=NL_G_MakeGraph(NameOfNetwork,
    NumberOfNodes,EndingNodesOfEdge,StartNodesOfEdge,
    XCoordinatesOfNodes,YCoordinatesOfNodes)//Creates
     the Star topoplogy
24 WindowNumber=1;//Graph Window Number
25 [GraphVisualize] = NL_G_ShowGraph(TopologyGraph,
    WindowNumber); // Visualize the Graph
26 xtitle("Star Topology","X-Nodes","Y-Nodes");
27 // 2.Colour the nodes and visualize
28 NodeColor=30; // Node Colour 2:[Blue],3:[Green], 5:[
    Red]
29 BorderThickness=10; // Node Border thickness
30 NodeDiameter=25; //Node diameter
31 WindowNumber=2;//window index
32
33 [OutputGraph]=NL_G_GraphEdgesLength(TopologyGraph);
```

```
34
35 disp(OutputGraph);
36
37 disp('Edge Length',OutputGraph.edge_length);
38
39 ListOfNodes=[1 3 5 7];//list of nodes
40 [NodeHighlight,f2]=NL_G_HighlightNodes(TopologyGraph
      ,ListOfNodes,NodeColor,BorderThickness,
      NodeDiameter,WindowNumber);// Highlight the
      specified nodes
41 xtitle("Node Higlight","X-Nodes","Y-Nodes");
42 WindowNumber=3; //Graph Window Number
43 [VisualizeNodesEdges] = NL_G_ShowGraphNE(
      TopologyGraph,WindowNumber);// Visualize the
      Graph along with indices for Nodes and Edges
44
45 [ExtractNode,ExtractEdge]=NL_G_GraphSize(
      TopologyGraph);//Extract the number of nodes and
      edges
46 disp(ExtractNode,ExtractEdge); //display the number
      of nodes and edges
47
48 //4. Display the number of nodes and edges
49 [ExtractNode,ExtractEdge]=NL_G_GraphSize(
      TopologyGraph);//Extract the number of nodes and
      edges
50 disp('Number of nodes:',ExtractNode); //display the
      number of nodes and edges
51 disp('Number of edges:',ExtractEdge);
52 //5.Display the Edge Index between two nodes
53 StartingNode=4;//starting node
54 EndingNode=1;//ending node
55 xtitle("Edge Index","X-Nodes","Y-Nodes");
56 [EdgeIndex]=NL_G_Nodes2Edge(TopologyGraph,
      StartingNode,EndingNode);//Get the Edge Index
57 disp('Index of Edge across nodes 4 and 1:',EdgeIndex
      );
```

Experiment: 4

Creation of the Mesh topology

Scilab code Solution 4.1 Mesh Topology Creation

```
1 //Experiment No.4
2 // This file must be used under the terms of the
    CeCILL.
3 // This source file is licensed as described in the
    file COPYING, which
4 // you should have received as part of this
    distribution. The terms
5 // are also available at
6 // http://www.cecill.info/licences/Licence_CeCILL_V2
    -en.txt
7 //This Source file is Written by Dr. T. Subbulaskhmi
    , Professor,
8 //School of Computing Science and Engineering, VIT
    University Chennai
9 //using the NARVAL examples of Scilab for Network
    Topology Creation
10 //The Operating System used for writing the code
```

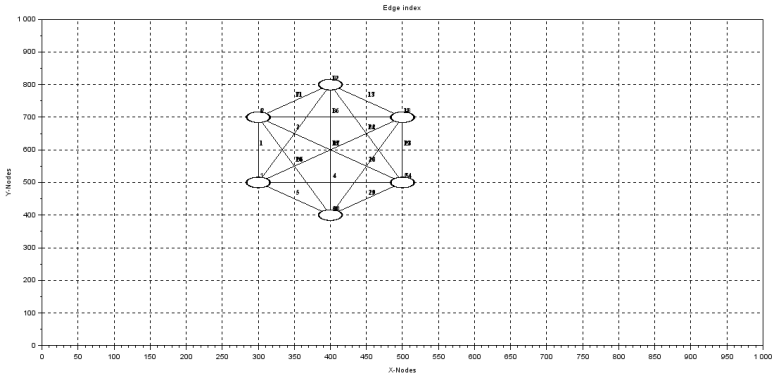

Figure 4.1: Mesh Topology Creation

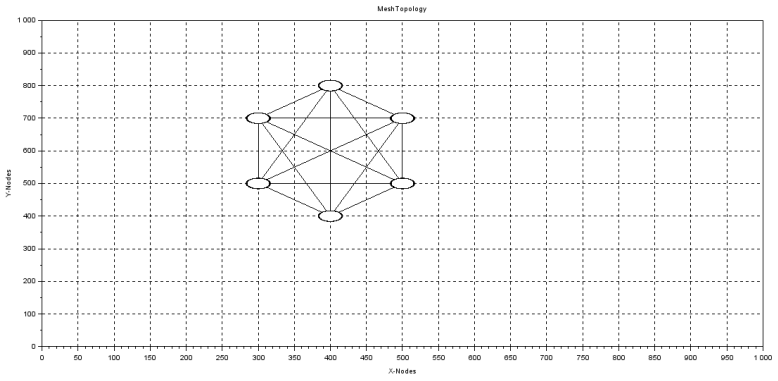

Figure 4.2: Mesh Topology Creation

```
    found in this file is Windows 8
11 //SCILAB version 5.5.2 and NARVAL toolbox version
    3.1
12 //Program to 1. Create and visualize a graph with
    Mesh Topology  2. Colour the nodes 3. Visualize
    the Graph with node number and edge number 4.
    Display the number of nodes and edges in scilab
    prompt 5. Display the length of every edge in
    Scilab prompt 6. Display the set of edges
    connected to a node inside a graph in Scilab
    Prompt
13
14 clear;
15 clc;
16
17 //1. Create and visualize a graph with Mesh Topology
18 NameOfNetwork='Mesh Topology';// Name of your
    network
19 NumberOfNodes=6;// Number of Nodes in the network
20 EndingNodesOfEdge=[1 1 1 1 1 2 2 2 2 2 3 3 3 3 3 4 4
     4 4 4 5 5 5 5 5 6 6 6 6 6]; //Ending Nodes of
    the connection lines
21 StartNodesOfEdge=[2 3 4 5 6 2 3 4 5 6 2 3 4 5 6 2 3
    4 5 6 2 3 4 5 6 2 3 4 5 6]; //Starting Nodes of
    the connection lines
22 XCoordinatesOfNodes=[300 300 400 500 500 400]; // X-
    Coordinates of the nodes
23 YCoordinatesOfNodes=[500 700 800 700 500 400]; // Y-
    Coordinates of the nodes
24 [TopologyGraph]=NL_G_MakeGraph(NameOfNetwork,
    NumberOfNodes,EndingNodesOfEdge,StartNodesOfEdge,
    XCoordinatesOfNodes,YCoordinatesOfNodes)//Creates
     the Mesh topoplogy
25 WindowNumber=1;//Graph Window Number
26 [GraphVisualize] = NL_G_ShowGraph(TopologyGraph,
    WindowNumber); // Visualize the Graph
27 xtitle("MeshTopology","X-Nodes","Y-Nodes");
28 //2. Colour the nodes
```

```
29 NodeColor=30; // Node Colour 2:[Blue],3:[Green], 5:[
     Red]
30 BorderThickness=10; // Node Border thickness
31 NodeDiameter=25; //Node diameter
32 WindowNumber=2;//window index
33 ListOfNodes=[1 3 5];//list of nodes
34
35 [NodeHighlight,Graph1]=NL_G_HighlightNodes(
     TopologyGraph,ListOfNodes,NodeColor,
     BorderThickness,NodeDiameter,WindowNumber);//
     Highlight the specified nodes and output a graph
36
37 //3. Visualize the Graph with node number and edge
     number
38 xtitle("Node Highlight","X-Nodes","Y-Nodes");
39 WindowNumber=3; //Graph Window Number
40 [VisualizeNodesEdges] = NL_G_ShowGraphNE(
     TopologyGraph,WindowNumber);// Visualize the
     Graph along with indices for Nodes and Edges
41 xtitle("Edge index","X-Nodes","Y-Nodes");
42 //4. Display the number of nodes and edges in scilab
      prompt
43 [ExtractNode,ExtractEdge]=NL_G_GraphSize(
     TopologyGraph);//Extract the number of nodes and
     edges
44 disp(ExtractNode,ExtractEdge); //display the number
     of nodes and edges
45
46 //5. Display the length of every edge in Scilab
     prompt
47 [ExtractEdge]=NL_G_EdgesLength(TopologyGraph.node_x,
     TopologyGraph.node_y,TopologyGraph.head,
     TopologyGraph.tail);//Display the length of every
      edge
48 disp('length of all edges',ExtractEdge);
49
50 //6. Display the set of edges connected to a node
     inside a graph in Scilab Prompt
```

```
51 NodeIndex=2;//node index
52 [EdgeApplication]=NL_G_EdgesOfNode(TopologyGraph,
     NodeIndex);//application of NL_G_EdgesOfNode
53 disp('Internal Edges of node 2:',EdgeApplication);
```

Experiment: 5

Creation of Tree topology

Scilab code Solution 5.1 Treee Topology

```
1 //Experiment No.5
2 // This file must be used under the terms of the
    CeCILL.
3 // This source file is licensed as described in the
    file COPYING, which
4 // you should have received as part of this
    distribution. The terms
5 // are also available at
6 // http://www.cecill.info/licences/Licence_CeCILL_V2
    -en.txt
7 //This Source file is Written by Dr. T. Subbulaskhmi
    , Professor,
8 //School of Computing Science and Engineering, VIT
    University Chennai
9 //using the NARVAL examples of Scilab for Network
    Topology Creation
10 //The Operating System used for writing the code
```

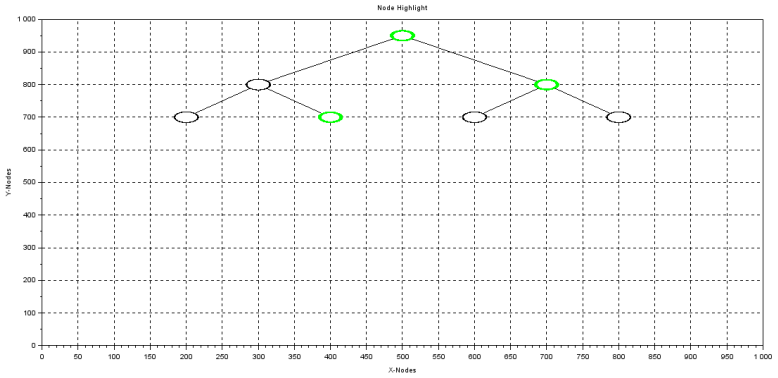

Figure 5.1: Treee Topology

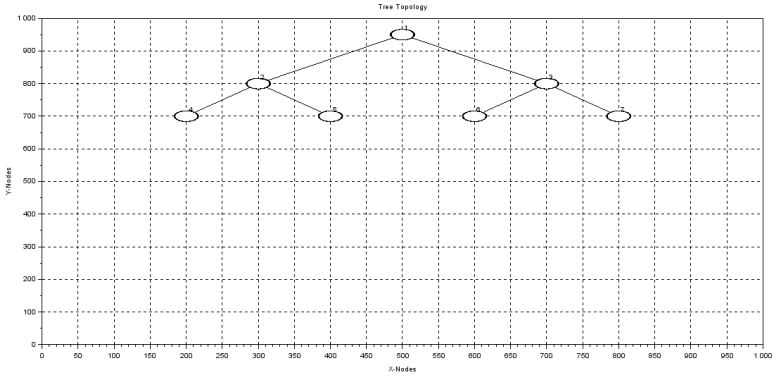

Figure 5.2: Treee Topology

```
    found in this file is Windows 8
11 //SCILAB version 5.5.2 and NARVAL toolbox version
    3.1
12 //Program to 1. create and Display Tree Topology 2.
    Colour the nodes 3. Display the node number and
    edge number 4. Display the number of nodes and
    edges
13
14 clear;
15 clc;
16 NameOfNetwork='Tree Topology';// graph name
17 NumberOfNodes=7;//no. of nodes
18 //every edge has head and tail.
19 EndingNodesOfEdge=[1 1 2 2 3 3];// tail of eatch
    edge, this is a node from a edge or link is
    originating
20 StartNodesOfEdge=[2 3 4 5 6 7];//head of each edge,
    this is a node where edge or link is terminating
21 XCoordinatesOfNodes=[500 300 700 200 400 600 800];//
     x-coordinate of each node
22 YCoordinatesOfNodes=[950 800 800 700 700 700 700];//
    y-coordinate of each node
23 [TopologyGraph]=NL_G_MakeGraph(NameOfNetwork,
    NumberOfNodes,EndingNodesOfEdge,StartNodesOfEdge,
    XCoordinatesOfNodes,YCoordinatesOfNodes)//
    application of NL_G_MakeGraph
24 WindowNumber=1;//window index
25 GraphVisualize=NL_G_ShowGraphN(TopologyGraph,
    WindowNumber);//graph visualization
26 xtitle("Tree Topology","X-Nodes","Y-Nodes");
27 NodeColor=3; // Node Colour 2:[Blue],3:[Green], 5:[
    Red]
28 BorderThickness=10; // Node Border thickness
29 NodeDiameter=25; //Node diameter
30 WindowNumber=2;//window index
31 NumberOfNodes=[1 3 5];//list of nodes
32
33 [NodeHighlight,Nodes]=NL_G_HighlightNodes(
```

```
    TopologyGraph,NumberOfNodes,NodeColor,
    BorderThickness,NodeDiameter,WindowNumber);//
    Highlight the specified nodes
xtitle("Node Highlight","X-Nodes","Y-Nodes");
WindowNumber=3; //Graph Window Number
[VisualizeNodesEdges] = NL_G_ShowGraphNE(
    TopologyGraph,WindowNumber);// Visualize the
    Graph along with indices for Nodes and Edges
[ExtractNode,ExtractEdge]=NL_G_GraphSize(
    TopologyGraph);//Extract the number of nodes and
    edges
disp(ExtractNode,ExtractEdge); //display the number
    of nodes and edges
xtitle("Indices for Nodes and Edges","X-Nodes","Y-
    Nodes");
```

Experiment: 6

Finding the Shortest Path in Bus topology

Scilab code Solution 6.1 Shortest Path in Bus Topology

```
1 //Experiment No.6
2 // This file must be used under the terms of the
    CeCILL.
3 // This source file is licensed as described in the
    file COPYING, which
4 // you should have received as part of this
    distribution.  The terms
5 // are also available at
6 // http://www.cecill.info/licences/Licence_CeCILL_V2
    -en.txt
7 //This Source file is Written by Dr. T. Subbulaskhmi
    , Professor,
8 //School of Computing Science and Engineering, VIT
    University Chennai
9 //using the NARVAL examples of Scilab for Network
```

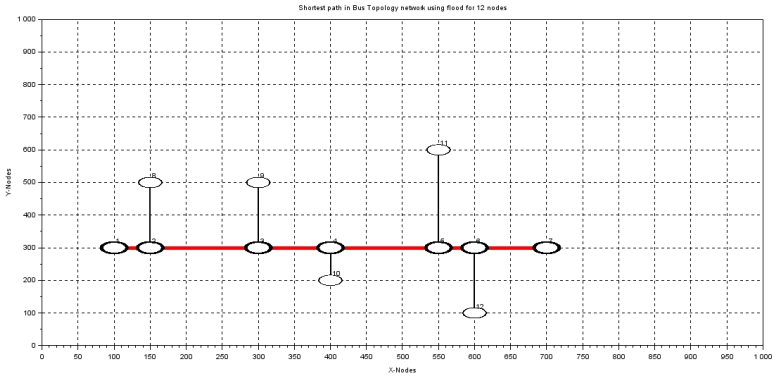

Figure 6.1: Shortest Path in Bus Topology

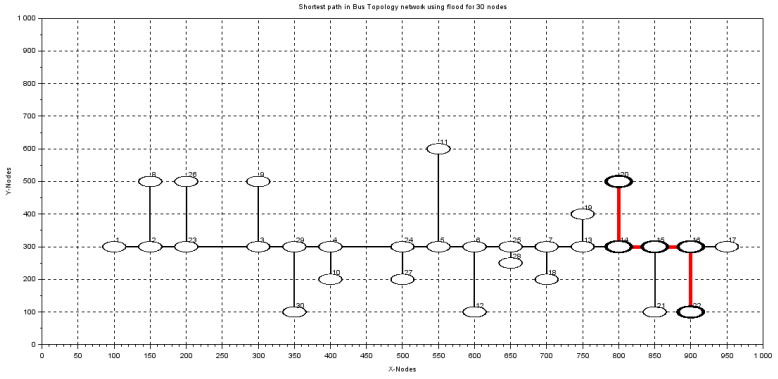

Figure 6.2: Shortest Path in Bus Topology

```
    Topology Creation
10 //The Operating System used for writing the code
    found in this file is Windows 8
11 //SCILAB version 5.5.2 and NARVAL toolbox version
    3.1//This Program is Written by Souarv kumar
    Surya(15bce1364),Prayag Bhatia (15bce1363),Maaz
    Ahmed (15bce1261), School of Computing Science
    and Engineering, VIT University Chennai using the
     NARVAL examples of Scilab for finding the
    shortest path
12 //This is the scilab code to find the Shortest path
    from source to destination in bus topology
    network using flood for 12 nodes.
13 clear;
14 clc;
15 NameOfNetwork='Shortest path from s-->d in bus
    topology with 12 nodes using flood';// graph name
16 NumberOfNodes=12;//no. of nodes
17 //every edge has head and tail.
18 EndingNodesOfConnection=[1 2 3 4 5 6 2 3 4 5 6];//
    Ending Node of the connection
19 StartNodesOfConnection=[2 3 4 5 6 7 8 9 10 11 12];//
    Starting Nodes of the connection lines
20 // so, an edge can be represent as (tail,head),tail
    and head both are node no. eg.:   tail[1]=2,head
    [1]=1 means edge 1 is originating from 2 and
    terminating on 1 that implies there is an edge
    betweem node 1 and   node 2
21 XCoordinatesOfNodes=[100 150 300 400 550 600 700 150
     300 400 550 600];// x-coordinate of each node
22 YCoordinatesOfNodes=[300 300 300 300 300 300 300 500
     500 200 600 100];//y-coordinate of each node
23 [TopologyGraph]=NL_G_MakeGraph(NameOfNetwork,
    NumberOfNodes,EndingNodesOfConnection,
    StartNodesOfConnection,XCoordinatesOfNodes,
    YCoordinatesOfNodes)//Creates the Bus topoplogy
24 NetworkSize=length(TopologyGraph.node_x);//Real
    network size
```

```
25 NeworkLinkQuantity=length(TopologyGraph.head);//
     Quantity of network links
26 [ne,nr]=NL_F_RandIntNiNj(NetworkSize)//Random
     Selection of two distinct nodes
27 TimeToLive=15;//Time-To-Live update
28 [path]=NL_R_Flood(TopologyGraph,ne,nr,TimeToLive)//
     Application of NL_R_Flood
29 ShortestPathLinks=NL_G_Nodes2Path(path,TopologyGraph
     );//Links of the shortest path
30 EdgeColor=ones(1,NeworkLinkQuantity);//Display the
     path between i and j: edge color
31 EdgeBorder=1.5*ones(1,NeworkLinkQuantity);//Edge
     width
32 EdgeColor(ShortestPathLinks)=5;//Define path color
33 EdgeBorder(ShortestPathLinks)=5;//Define width
34 NodeBorder=4*ones(1,NetworkSize);//Node border
35 NodeBorder(path)=10;//Node border for source to
     destination path
36 TopologyGraph.node_border=NodeBorder;//Node border
37 TopologyGraph.edge_color=EdgeColor;//Define edge
     color
38 TopologyGraph.edge_width=EdgeBorder;//Edge width
39 windownumber=1;//window index
40 GraphVisualize=NL_G_ShowGraphN(TopologyGraph,
     windownumber);//graph visualization
41 xtitle("Shortest path in Bus Topology network using
     flood for 12 nodes","X-Nodes","Y-Nodes");
42
43 //This is scilab code to find the Shortest path from
      source to destination in bus topology network
     using flood for 30 nodes.
44 NameofNetwork='Shortest path from s-->d in bus
     topology with 30 nodes using flood';// graph name
45 NumberOfNodes=30;//no. of nodes
46 //every edge has StartingNodesOfConnection and tail.
47 EndingNodesOfConnection=[1 2 23 3 29 4 24 5 6 25 7
     13 14 15 16 2 23 3 29 4 24 5 6 25 7 13 14 15 16];
     // tail of eatch edge, this is a node from a edge
```

```
    or link is originating
48 StartingNodesOfConnection=[2 23 3 29 4 24 5 6 25 7
    13 14 15 16 17 8 26 9 30 10 27 11 12 28 18 19 20
    21 22];//head of each edge,this is a node where
    edge or link is terminating
49 // so, an edge can be represent as (tail,head),tail
    and head both are node no. eg.:   tail[1]=2,head
    [1]=1 means edge 1 is originating from 2 and
    terminating on 1 that implies there is an edge
    betweem node 1 and   node 2
50 XCoordinatesOfNodes=[100 150 300 400 550 600 700 150
     300 400 550 600 750 800 850 900 950 700 750 800
    850 900 200 500 650 200 500 650 350 350];// x-
    coordinate of each node
51 YCoordinatesOfNodes=[300 300 300 300 300 300 300 500
     500 200 600 100 300 300 300 300 300 200 400 500
    100 100 300 300 300 500 200 250 300 100];//y-
    coordinate of each node
52 [TopologyGraph]=NL_G_MakeGraph(NameofNetwork,
    NumberOfNodes,EndingNodesOfConnection,
    StartingNodesOfConnection,XCoordinatesOfNodes,
    YCoordinatesOfNodes)//application of
    NL_G_MakeGraph
53 NetworkSize=length(TopologyGraph.node_x);//real
    network size
54 NeworkLinkQuantity=length(TopologyGraph.head);//
    quantity of network links
55 [Node1,Node2]=NL_F_RandIntNiNj(NetworkSize)//
    selection of two distinct nodes
56 TTL=15;//Time-To-Live update
57 [path]=NL_R_Flood(TopologyGraph,Node1,Node2,TTL)//
    application of NL_R_Flood
58 ShortestPathLinks=NL_G_Nodes2Path(path,TopologyGraph
    );//links of the shortest path
59 EdgeColor=ones(1,NeworkLinkQuantity);//display the
    path between i and j: edge color
60 EdgeBorder=1.5*ones(1,NeworkLinkQuantity);//edge
    width
```

```
61 EdgeColor(ShortestPathLinks)=5;//define path color
62 EdgeBorder(ShortestPathLinks)=5;//define width
63 NodeBorder=4*ones(1,NetworkSize);//node border
64 NodeBorder(path)=10;//node border for source to
      destination path
65 TopologyGraph.node_border=NodeBorder;//node border
66 TopologyGraph.edge_color=EdgeColor;//define edge
      color
67 TopologyGraph.edge_width=EdgeBorder;//edge width
68 WindowIndex=2;//window index
69 GraphVisualize=NL_G_ShowGraphN(TopologyGraph,
      WindowIndex);//graph visualization
70 xtitle("Shortest path in Bus Topology network using
      flood for 30 nodes","X-Nodes","Y-Nodes");
```

Experiment: 7

Finding the Shortest Path in Ring topology

Scilab code Solution 7.1 Shortest Path in Ring Topology

```
1  //Experiment No.7
2  // This file must be used under the terms of the
       CeCILL.
3  // This source file is licensed as described in the
       file COPYING, which
4  // you should have received as part of this
       distribution.  The terms
5  // are also available at
6  // http://www.cecill.info/licences/Licence_CeCILL_V2
       -en.txt
7  //This Source file is Written by Dr. T. Subbulaskhmi
       , Professor,
8  //School of Computing Science and Engineering, VIT
       University Chennai
9  //using the NARVAL examples of Scilab for Network
       Topology Creation
10 //The Operating System used for writing the code
       found in this file is Windows 8
11 //SCILAB version 5.5.2 and NARVAL toolbox version
```

```
    3.1//This Program is Written by Souarv kumar
    Surya(15bce1364),Prayag Bhatia (15bce1363),Maaz
    Ahmed (15bce1261), School of Computing Science
    and Engineering, VIT University Chennai using the
     NARVAL examples of Scilab
12 //This is the scilab code to find the Shortest path
    from source to destination in ring topology
    network using flood for 10 nodes.
13 clear;
14 clc;
15
16 NameOfNetwork=' shortest path from source to
    destinetion in ring topology with 10 nodes using
    flood';// graph name
17 NumberOfNodes=10;//no. of nodes
18 //every edge has head and tail.
19 EndingNodesOfConnection=[2 3 5 5 6 4 9 7 8 6];//
    tail of eatch edge, this is a node from a edge or
     link is originating
20 StartNodesOfConnection=[1 1 3 4 2 9 7 8 10 10];//
    head of each edge,this is a node where edge or
    link is terminating
21 // so, an edge can be represent as (tail,head),tail
    and head both are node no. eg.:   tail[1]=2,head
    [1]=1 means edge 1 is originating from 2 and
    terminating on 1 that implies there is an edge
    betweem node 1 and   node 2
22 XCoordinatesOfNodes=[200 400 100 350 200 500 550 600
     500 550];// x-coordinate of each node
23 YCoordinatesOfNodes=[100 100 400 650 650 200 500 400
     600 300];//y-coordinate of each node
24 [TopologyGraph]=NL_G_MakeGraph(NameOfNetwork,
    NumberOfNodes,EndingNodesOfConnection,
    StartNodesOfConnection,XCoordinatesOfNodes,
    YCoordinatesOfNodes)//application of
    NL_G_MakeGraph
25 NetworkSize=length(TopologyGraph.node_x);//real
    network size
```

```
26 NetworkQuantity=length(TopologyGraph.head);//
     quantity of network links
27 [Node1,Node2]=NL_F_RandIntNiNj(NetworkSize)//
     selection of two distinct nodes
28 TimeToLive=15;//Time-To-Live update
29 [path]=NL_R_Flood(TopologyGraph,Node1,Node2,
     TimeToLive)//application of NL_R_Flood
30 ShortestPath=NL_G_Nodes2Path(path,TopologyGraph);//
     links of the shortest path
31 EdgeColor=ones(1,NetworkQuantity);//display the path
      between i and j: edge color
32 EdgeBorder=1.5*ones(1,NetworkQuantity);//edge width
33 EdgeColor(ShortestPath)=5;//define path color
34 EdgeBorder(ShortestPath)=5;//define width
35 NodeBorder=4*ones(1,NetworkSize);//node border
36 NodeBorder(path)=10;//node border for source to
     destination path
37 TopologyGraph.node_border=NodeBorder;//node border
38 TopologyGraph.edge_color=EdgeColor;//define edge
     color
39 TopologyGraph.edge_width=EdgeBorder;//edge width
40 WinodeNumber=1;//window index
41 GraphVisualize=NL_G_ShowGraphN(TopologyGraph,
     WinodeNumber);//graph visualization
42 xtitle("Shortest path in Ring Topology network using
      flood for 12 nodes","X-Nodes","Y-Nodes");
43 //This is the scilab code to find the Shortest path
     from source to destination in ring topology
     network using flood for 30 nodes.
44 NameOfNetwork=' shortest path from source to
     destinetion in ring topology with 30 nodes using
     flood';// graph name
45 NumberOfNodes=30;//no. of nodes
46 //every edge has head and tail.
47 EndingNodesOfConnection=[2 20 4 9 7 8 11 5 16 11 12
     3 13 1 1 15 17 18 19 20 6 22 21 23 26 27 24 30 29
      28 30];// tail of eatch edge, this is a node
     from a edge or link is originating
```

```
48 StartNodesOfConnection=[1 2 17 7 19 10 5 16 4 12 3
     13 14 14 15 2 18 9 8 6 22 21 23 26 27 24 25 29 28
      10 25];//head of each edge,this is a node where
     edge or link is terminating
49 // so, an edge can be represent as (tail,head),tail
     and head both are node no. eg.:   tail[1]=2,head
     [1]=1 means edge 1 is originating from 2 and
     terminating on 1 that implies there is an edge
     betweem node 1 and   node 2
50 XCoordinatesOfNodes=[200 400 100 350 200 600 650 800
      600 900 100 100 150 150 300 300 400 500 750 500
     700 650 800 900 900 900 900 950 1000 950];// x-
     coordinate of each node
51 YCoordinatesOfNodes=[100 100 400 650 650 100 700 600
      700 600 600 500 300 200 100 700 700 700 650 100
     100 100 100 300 400 100 200 550 500 400];//y-
     coordinate of each node
52 [TopologyGraph]=NL_G_MakeGraph(NameOfNetwork,
     NumberOfNodes,EndingNodesOfConnection,
     StartNodesOfConnection,XCoordinatesOfNodes,
     YCoordinatesOfNodes)//application of
     NL_G_MakeGraph
53 NetworkSize=length(TopologyGraph.node_x);//real
     network size
54 NetworkQuantity=length(TopologyGraph.head);//
     quantity of network links
55 [Node1,Node2]=NL_F_RandIntNiNj(NetworkSize)//
     selection of two distinct nodes
56 TTL=15;//Time-To-Live update
57 [path]=NL_R_Flood(TopologyGraph,Node1,Node2,TTL)//
     application of NL_R_Flood
58 ShortestPath=NL_G_Nodes2Path(path,TopologyGraph);//
     links of the shortest path
59 EdgeColor=ones(1,NetworkQuantity);//display the path
      between i and j: edge color
60 EdgeBorder=1.5*ones(1,NetworkQuantity);//edge width
61 EdgeColor(ShortestPath)=5;//define path color
62 EdgeBorder(ShortestPath)=5;//define width
```

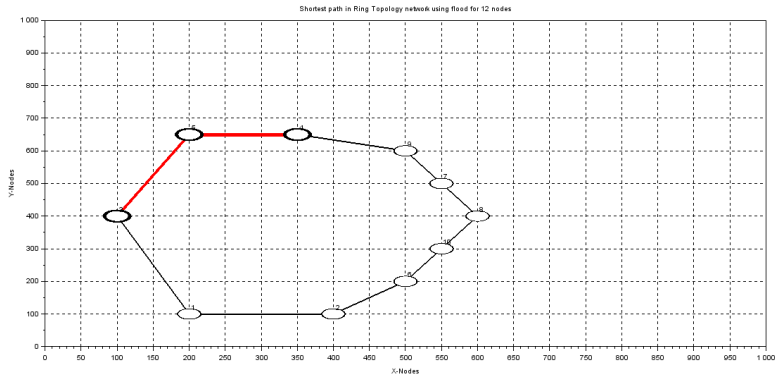

Figure 7.1: Shortest Path in Ring Topology

```
63 NodeBorder=4*ones(1,NetworkSize);//node border
64 NodeBorder(path)=10;//node border for source to
     destination path
65 TopologyGraph.node_border=NodeBorder;//node border
66 TopologyGraph.edge_color=EdgeColor;//define edge
     color
67 TopologyGraph.edge_width=EdgeBorder;//edge width
68 WindowIndex=2;//window index
69 GraphVisualize=NL_G_ShowGraphN(TopologyGraph,
     WindowIndex);//graph visualization
70 xtitle("Shortest path in Ring Topology network using
      flood for 30 nodes","X-Nodes","Y-Nodes");
```

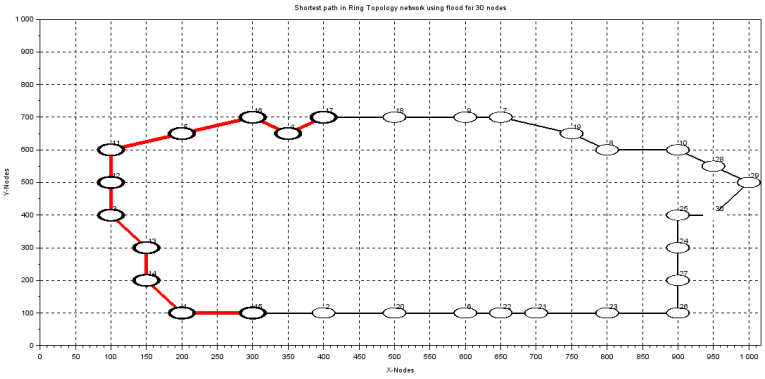

Figure 7.2: Shortest Path in Ring Topology

Experiment: 8

Finding the Shortest Path in Star topology

Scilab code Solution 8.1 Shortest Path in Star Topology

```
1  //Experiment No.8
2  // This file must be used under the terms of the
       CeCILL.
3  // This source file is licensed as described in the
       file COPYING, which
4  // you should have received as part of this
       distribution.  The terms
5  // are also available at
6  // http://www.cecill.info/licences/Licence_CeCILL_V2
       -en.txt
7  //This Source file is Written by Dr. T. Subbulaskhmi
       , Professor,
8  //School of Computing Science and Engineering, VIT
       University Chennai
9  //using the NARVAL examples of Scilab for Network
       Topology Creation
10 //The Operating System used for writing the code
```

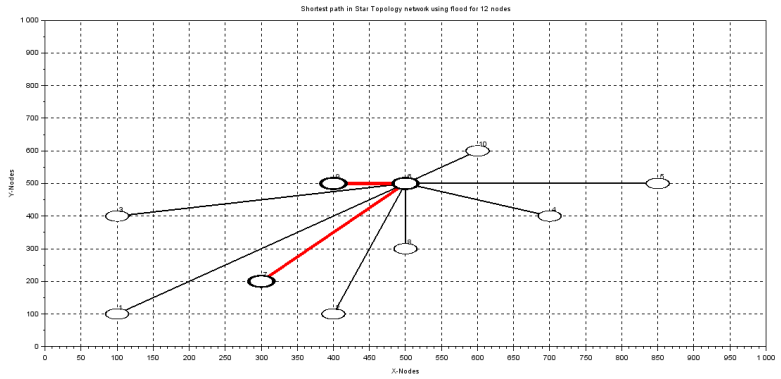

Figure 8.1: Shortest Path in Star Topology

```
    found in this file is Windows 8
11 //SCILAB version 5.5.2 and NARVAL toolbox version
    3.1//This Program is Written by Souarv kumar
    Surya(15bce1364),Prayag Bhatia (15bce1363),Maaz
    Ahmed (15bce1261), School of Computing Science
    and Engineering, VIT University Chennai using the
     NARVAL examples of Scilab for finding the
    shortest path
12 //This is the scilab code to find the Shortest path
    from source to destination in Star topology
    network using flood for 10 nodes.
13 clear;
14 clc;
15
16 NameOfNetwork='star network with 10 node';// graph
    name
17 NumberOfNodes=10;//no of nodes
18 //every edge has head and tail.
19 EndingNodesOfConnection=[6 6 6 6 6 6 6 6 6 ];// tail
     of eatch edge, this is a node from a edge or
    link is originating
20 StartNodesOfConnection=[1 2 3 4 5 7 8 9 10];//head
    of each edge,this is a node where edge or link is
```

```
    terminating
// so, an edge can be represent as (tail,head),tail
   and head both are node no. eg.:   tail[1]=6,head
   [1]=1 means edge 1 is originating from 6 and
   terminating on 1 that implies there is an edge
   betweem node 1 and   node 2
XCoordinatesOfNodes=[100 400 100 700 850 500 300 500
    400 600];// x-coordinate of each node
YCoordinatesOfNodes=[100 100 400 400 500 500 200 300
    500 600];//y-coordinate of each node
//node i can represent as node_x[i],node_y[i]
[TopologyGraph]=NL_G_MakeGraph(NameOfNetwork,
   NumberOfNodes,EndingNodesOfConnection,
   StartNodesOfConnection,XCoordinatesOfNodes,
   YCoordinatesOfNodes)//application of
   NL_G_MakeGraph
NetworkSize=length(TopologyGraph.node_x);//real
   network size
NetworkQuantity=length(TopologyGraph.head);//
   quantity of network links
[Node1,Node2]=NL_F_RandIntNiNj(NetworkSize)//
   selection of two distinct nodes
TimeToLive=15;//Time-To-Live update
[path]=NL_R_Flood(TopologyGraph,Node1,Node2,
   TimeToLive)//application of NL_R_Flood
ShortestPath=NL_G_Nodes2Path(path,TopologyGraph);//
   links of the shortest path
EdgeColor=ones(1,NetworkQuantity);//display the path
    between i and j: edge color
EdgeBorder=1.5*ones(1,NetworkQuantity);//edge width
EdgeColor(ShortestPath)=5;//define path color
EdgeBorder(ShortestPath)=5;//define width
NodeBorder=4*ones(1,NetworkSize);//node border
NodeBorder(path)=10;//node border for source to
   destination path
TopologyGraph.node_border=NodeBorder;//node border
TopologyGraph.edge_color=EdgeColor;//define edge
   color
```

```
40 TopologyGraph.edge_width=EdgeBorder;//edge width
41 WindowNumber=1;//window index
42 GraphVisualize=NL_G_ShowGraphN(TopologyGraph,
      WindowNumber);//graph visualization
43
44 xtitle("Shortest path in Star Topology network using
       flood for 12 nodes","X-Nodes","Y-Nodes");
45
46 //This is the scilab code to find the Shortest path
      from source to destination in Star topology
      network using flood for 30 nodes.
47
48 NameOfNetwork='star network with 30 node';// graph
      name
49 NumberOfNodes=30;//no of nodes
50 //every edge has head and tail.
51 EndingNodesOfConnection=[6 6 6 6 6 6 6 6 6 6 6 6 6 6
       6 6 6 6 6 6 6 6 6 6 6 6 6 6];// tail of eatch
      edge, this is a node from a edge or link is
      originating
52 StartNodesOfConnection=[1 2 3 4 5 7 8 9 10 11 12 13
      14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29
      30];//head of each edge,this is a node where edge
       or link is terminating
53 // so, an edge can be represent as (tail,head),tail
      and head both are node no. eg.:   tail[1]=6,head
      [1]=1 means edge 1 is originating from 6 and
      terminating on 1 that implies there is an edge
      betweem node 1 and   node 2
54 XCoordinatesOfNodes=[100 400 100 700 850 500 300 500
       400 600 300 200 500 400 700 800 900 850 950 450
      550 650 350 250 150 725 825 925 300 650];// x-
      coordinate of each node
55 YCoordinatesOfNodes=[200 100 400 400 500 500 200 300
       500 600 250 350 700 300 200 500 400 600 700 700
      350 550 450 235 750 100 353 770 400 950];//y-
      coordinate of each node
56 //node i can represent as node_x[i],node_y[i]
```

```
57 [TopologyGraph]=NL_G_MakeGraph(NameOfNetwork,
      NumberOfNodes,EndingNodesOfConnection,
      StartNodesOfConnection,XCoordinatesOfNodes,
      YCoordinatesOfNodes)//application of
      NL_G_MakeGraph
58 NetworkSize=length(TopologyGraph.node_x);//real
      network size
59 NetworkQuantity=length(TopologyGraph.head);//
      quantity of network links
60 [Node1,Node2]=NL_F_RandIntNiNj(NetworkSize)//
      selection of two distinct nodes
61 TTL=15;//Time-To-Live update
62 [path]=NL_R_Flood(TopologyGraph,Node1,Node2,TTL)//
      application of NL_R_Flood
63 ShortestPath=NL_G_Nodes2Path(path,TopologyGraph);//
      links of the shortest path
64 EdgeColor=ones(1,NetworkQuantity);//display the path
       between i and j: edge color
65 EdgeBorder=1.5*ones(1,NetworkQuantity);//edge width
66 EdgeColor(ShortestPath)=5;//define path color
67 EdgeBorder(ShortestPath)=5;//define width
68 NodeBorder=4*ones(1,NetworkSize);//node border
69 NodeBorder(path)=10;//node border for source to
      destination path
70 TopologyGraph.node_border=NodeBorder;//node border
71 TopologyGraph.edge_color=EdgeColor;//define edge
      color
72 TopologyGraph.edge_width=EdgeBorder;//edge width
73 WindowNumber=2;//window index
74 GraphVisualize=NL_G_ShowGraphN(TopologyGraph,
      WindowNumber);//graph visualization
75 xtitle("Shortest path in star Topology network using
       flood for 30 nodes","X-Nodes","Y-Nodes");
```

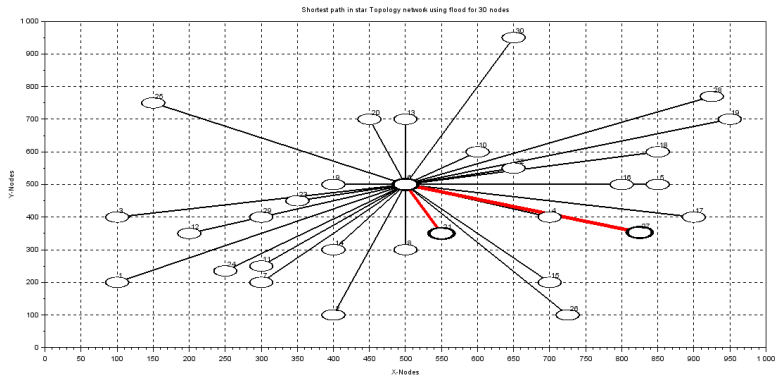

Figure 8.2: Shortest Path in Star Topology

Experiment: 9

Finding the Shortest Path in Mesh topology

Scilab code Solution 9.1 Shortest Path in Mesh Topology

```
1 //Experiment No.9
2 // This file must be used under the terms of the
    CeCILL.
3 // This source file is licensed as described in the
    file COPYING, which
4 // you should have received as part of this
    distribution.  The terms
5 // are also available at
6 // http://www.cecill.info/licences/Licence_CeCILL_V2
    -en.txt
7 //This Source file is Written by Dr. T. Subbulaskhmi
    , Professor,
8 //School of Computing Science and Engineering, VIT
    University Chennai
9 //using the NARVAL examples of Scilab for Network
```

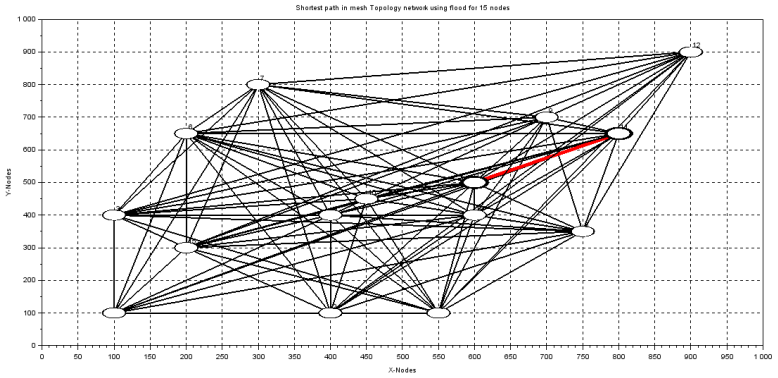

Figure 9.1: Shortest Path in Mesh Topology

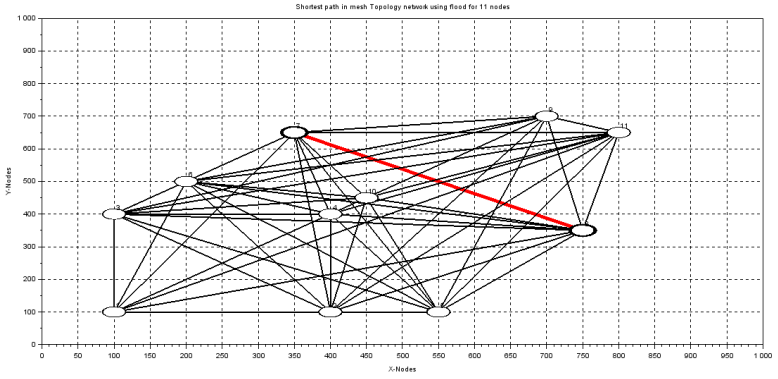

Figure 9.2: Shortest Path in Mesh Topology

```
    Topology Creation
10 //The Operating System used for writing the code
    found in this file is Windows 8
11 //SCILAB version 5.5.2 and NARVAL toolbox version
    3.1//This Program is Written by Souarv kumar
    Surya(15bce1364),Prayag Bhatia (15bce1363),Maaz
    Ahmed (15bce1261), School of Computing Science
    and Engineering, VIT University Chennai using the
     NARVAL examples of Scilab for finding the
    shortest path
12 //This is the scilab code to find the Shortest path
    from source to destination in Mesh topology
    network using flood for 11 nodes.
13 clear;
14 clc;
15
16 NameOfNetwork='mesh topology with 11 nodes';// graph
     name
17 NumberOfNodes=11;//no. of nodes
18 //every edge has head and tail.
19 EndingNodesOfConnection=[2 3 4 3 4 5 5 5 5 6 6 6 6 6
     7 7 7 7 7 7 8 8 8 8 8 8 8 9 9 9 9 9 9 9 9 10 10
    10 10 10 10 10 10 10 11 11 11 11 11 11 11 11 11
    11];// tail of eatch edge, this is a node from a
    edge or link is originating
20 StartNodesOfConnection=[1 2 3 1 2 1 2 3 4 1 2 3 4 5
    1 2 3 4 5 6 1 2 3 4 5 6 7 1 2 3 4 5 6 7 8 1 2 3 4
     5 6 7 8 9 1 2 3 4 5 6 7 8 9 10];//head of each
    edge,this is a node where edge or link is
    terminating
21 // so, an edge can be represent as (tail,head),tail
    and head both are node no. eg.:   tail[1]=2,head
    [1]=1 means edge 1 is originating from 2 and
    terminating on 1 that implies there is an edge
    betweem node 1 and   node 2
22 XCoordinatesOfNodes=[100 400 100 400 550 200 350 750
     700 450 800];// x-coordinate of each node
23 YCoordinatesOfNodes=[100 100 400 400 100 500 650 350
```

```
      700 450 650];//y-coordinate of each node
24 [TopologyGraph]=NL_G_MakeGraph(NameOfNetwork,
     NumberOfNodes,EndingNodesOfConnection,
     StartNodesOfConnection,XCoordinatesOfNodes,
     YCoordinatesOfNodes)//application of
     NL_G_MakeGraph
25 NetworkSize=length(TopologyGraph.node_x);//real
     network size
26 NetworkQuantity=length(TopologyGraph.head);//
     quantity of network links
27 [Node1,Node2]=NL_F_RandIntNiNj(NetworkSize)//
     selection of two distinct nodes
28 TimeToLive=5;//Time-To-Live update
29 [path]=NL_R_Flood(TopologyGraph,Node1,Node2,
     TimeToLive)//application of NL_R_Flood
30 ShortestPath=NL_G_Nodes2Path(path,TopologyGraph);//
     links of the shortest path
31 EdgeColor=ones(1,NetworkQuantity);//display the path
      between i and j: edge color
32 EdgeBorder=1.5*ones(1,NetworkQuantity);//edge width
33 EdgeColor(ShortestPath)=5;//define path color
34 EdgeBorder(ShortestPath)=5;//define width color
35 NodeBorder=4*ones(1,NetworkSize);//node border
36 NodeBorder(path)=10;//node border for source to
     destination path
37 TopologyGraph.node_border=NodeBorder;//node border
38 TopologyGraph.edge_color=EdgeColor;//define edge
     color
39 TopologyGraph.edge_width=EdgeBorder;//edge width
40 WindowNumber=1;//window index
41 GraphVisualize=NL_G_ShowGraphN(TopologyGraph,
     WindowNumber);//graph visualization
42 xtitle("Shortest path in mesh Topology network using
      flood for 11 nodes","X-Nodes","Y-Nodes");
43 //This is the scilab code to find the Shortest path
     from source to destination in Mesh topology
     network using flood for 15 nodes.
44 NameOfNetwork='mesh topology with 15 nodes';// graph
```

```
    name
45 NumberOfNodes=15;//no. of nodes
46 //every edge has head and tail.
47 EndingNodesOfConnection=[2 3 4 3 4 5 5 5 5 6 6 6 6 6
    7 7 7 7 7 7 8 8 8 8 8 8 9 9 9 9 9 9 9 9 10 10
   10 10 10 10 10 10 10 11 11 11 11 11 11 11 11 11
   11 12 12 12 12 12 12 12 12 12 12 12 13 13 13 13
   13 13 13 13 13 13 13 13 14 14 14 14 14 14 14 14
   14 14 14 14 14 15 15 15 15 15 15 15 15 15 15 15
   15 15 15];// tail of eatch edge, this is a node
   from a edge or link is originating
48 StartNodesOfConnection=[1 2 3 1 2 1 2 3 4 1 2 3 4 5
   1 2 3 4 5 6 1 2 3 4 5 6 7 1 2 3 4 5 6 7 8 1 2 3 4
    5 6 7 8 9 1 2 3 4 5 6 7 8 9 10 1 2 3 4 5 6 7 8 9
    10 11 1 2 3 4 5 6 7 8 9 10 11 12 1 2 3 4 5 6 7 8
    9 10 11 12 13 1 2 3 4 5 6 7 8 9 10 11 12 13 14];
   //head of each edge,this is a node where edge or
   link is terminating
49 // so, an edge can be represent as (tail,head),tail
   and head both are node no. eg.:   tail[1]=2,head
   [1]=1 means edge 1 is originating from 2 and
   terminating on 1 that implies there is an edge
   betweem node 1 and   node 2
50 XCoordinatesOfNodes=[100 400 100 400 550 200 300 750
    700 450 800 900 600 600 200]; // x-coordinate of
    each node
51 YCoordinatesOfNodes=[100 100 400 400 100 650 800 350
    700 450 650 900 500 400 300];//y-coordinate of
   each node
52 //node i can represent as node_x[i],node_y[i]
53 [TopologyGraph]=NL_G_MakeGraph(NameOfNetwork,
   NumberOfNodes,EndingNodesOfConnection,
   StartNodesOfConnection,XCoordinatesOfNodes,
   YCoordinatesOfNodes)//application of
   NL_G_MakeGraph
54 NetworkSize=length(TopologyGraph.node_x);//real
   network size
55 NetworkQuantity=length(TopologyGraph.head);//
```

```
    quantity of network links
56 [Node1,Node2]=NL_F_RandIntNiNj(NetworkSize)//
    selection of two distinct nodes
57 TimeToLive=5;//Time-To-Live update
58 [path]=NL_R_Flood(TopologyGraph,Node1,Node2,
    TimeToLive)//application of NL_R_Flood
59 ShortestPath=NL_G_Nodes2Path(path,TopologyGraph);//
    links of the shortest path
60 EdgeColor=ones(1,NetworkQuantity);//display the path
    between i and j: edge color
61 EdgeBorder=1.5*ones(1,NetworkQuantity);//edge width
62 EdgeColor(ShortestPath)=5;//define path color
63 EdgeBorder(ShortestPath)=5;//define width color
64 NodeBorder=4*ones(1,NetworkSize);//node border
65 NodeBorder(path)=10;//node border for source to
    destination path
66 TopologyGraph.node_border=NodeBorder;//node border
67 TopologyGraph.edge_color=EdgeColor;//define edge
    color
68 TopologyGraph.edge_width=EdgeBorder;//edge width
69 WindowNumber=2;//window index
70 GraphVisualize=NL_G_ShowGraphN(TopologyGraph,
    WindowNumber);//graph visualization
71 xtitle("Shortest path in mesh Topology network using
    flood for 15 nodes","X-Nodes","Y-Nodes");
```

Experiment: 10

Finding the Shortest Path in Tree topology

Scilab code Solution 10.1 Shortest Path in Tree Topology

```
1  //Experiment No.10
2  // This file must be used under the terms of the
       CeCILL.
3  // This source file is licensed as described in the
       file COPYING, which
4  // you should have received as part of this
       distribution.  The terms
5  // are also available at
6  // http://www.cecill.info/licences/Licence_CeCILL_V2
       -en.txt
7  //This Source file is Written by Dr. T. Subbulaskhmi
       , Professor,
8  //School of Computing Science and Engineering, VIT
       University Chennai
9  //using the NARVAL examples of Scilab for Network
       Topology Creation
10 //The Operating System used for writing the code
```

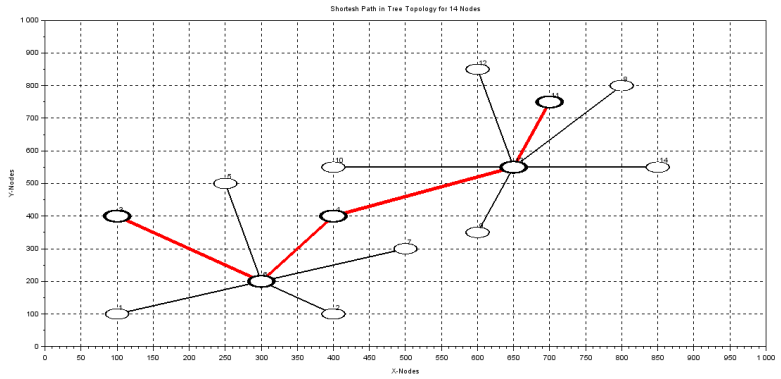

Figure 10.1: Shortest Path in Tree Topology

```
    found in this file is Windows 8
11 //SCILAB version 5.5.2 and NARVAL toolbox version
    3.1//This Program is Written by Souarv kumar
    Surya(15bce1364),Prayag Bhatia (15bce1363),Maaz
    Ahmed (15bce1261), School of Computing Science
    and Engineering, VIT University Chennai using the
     NARVAL examples of Scilab for finding the
    shortest path
12 //This is the scilab code to find the Shortest path
    from source to destination in Tree topology
    network using flood for 14 nodes.
13 clear;
14 clc;
15
16 NameOfNetwork='shortest path from source to
    destinetion in star topology with 14 nodes using
    flood';// graph name
17 NumberOfNodes=14;//graph parameters
18 //every edge has head and tail.
19 EndingNodesOfConnection=[6 6 6 6 6 6 13 13 13 13 13
    13 13];// tail of eatch edge, this is a node from
     a edge or link is originating
20 StartNodesOfConnection=[1 2 3 4 5 7 8 9 10 11 12 14
```

```
    4];//head of each edge,this is a node where edge
    or link is terminating
21 // so, an edge can be represent as (tail,head),tail
    and head both are node no. eg.:   tail[1]=6,head
    [1]=1 means edge 1 is originating from 6 and
    terminating on 1 that implies there is an edge
    betweem node 1 and   node 2
22 XCoordinatesOfNodes=[100 400 100 400 250 300 500 800
     600 400 700 600 650 850];// x-coordinate of each
     node
23 YCoordinatesOfNodes=[100 100 400 400 500 200 300 800
     350 550 750 850 550 550];//y-coordinate of each
    node
24 [TopologyGraph]=NL_G_MakeGraph(NameOfNetwork,
    NumberOfNodes,EndingNodesOfConnection,
    StartNodesOfConnection,XCoordinatesOfNodes,
    YCoordinatesOfNodes)//application of
    NL_G_MakeGraph
25 NetworkSize=length(TopologyGraph.node_x);//real
    network size
26 NetworkQuantity=length(TopologyGraph.head);//
    quantity of network links
27 [Node1,Node2]=NL_F_RandIntNiNj(NetworkSize)//
    selection of two distinct nodes
28 TTL=15;//Time-To-Live update
29 [path]=NL_R_Flood(TopologyGraph,Node1,Node2,TTL)//
    application of NL_R_Flood
30 ShortestPath=NL_G_Nodes2Path(path,TopologyGraph);//
    links of the shortest path
31 EdgeColor=ones(1,NetworkQuantity);//display the path
     between i and j: edge color
32 EdgeBorder=1.5*ones(1,NetworkQuantity);//edge width
33 EdgeColor(ShortestPath)=5;//define path color
34 EdgeBorder(ShortestPath)=5;//define width
35 NodeBorder=4*ones(1,NetworkSize);//node border
36 NodeBorder(path)=10;//node border for source to
    destination path
37 TopologyGraph.node_border=NodeBorder;//node border
```

```
38 TopologyGraph.edge_color=EdgeColor;//define edge
     color
39 TopologyGraph.edge_width=EdgeBorder;//edge width
40 WindowIndex=1;//window index
41 GraphVisualize=NL_G_ShowGraphN(TopologyGraph,
     WindowIndex);//graph visualization
42 xtitle("Shortesh Path in Tree Topology for 14 Nodes"
     ,"X-Nodes","Y-Nodes")
43
44 //This is the scilab code to find the Shortest path
     from source to destination in Tree topology
     network using flood for 14 nodes.
45 NameOfNetwork='shortest path from source to
     destinetion in tree topology with 30 nodes using
     flood';// graph name
46 NumberOfNodes=30;//no. of nodes
47 //every edge has head and tail.
48 EndingNodesOfConnection=[6 6 6 6 6 6 13 13 13 13 13
     13 13 15 16 17 18 19 20 21 22 23 24 25 26 27 28 9
      5];// tail of eatch edge, this is a node from a
     edge or link is originating
49 StartNodesOfConnection=[1 2 3 4 5 7 8 9 10 11 12 14
     4 2 15 16 5 18 13 13 14 22 11 14 25 8 12 29 30];
     //head of each edge,this is a node where edge or
     link is terminating
50 // so, an edge can be represent as (tail,head),tail
     and head both are node no. eg.:   tail[1]=6,head
     [1]=1 means edge 1 is originating from 6 and
     terminating on 1 that implies there is an edge
     betweem node 1 and node 2
51 XCoordinatesOfNodes=[100 400 100 400 250 300 500 800
      600 400 700 600 650 850 500 600 700 300 200 700
     500 900 900 700 950 900 900 600 600 150];// x-
     coordinate of each node
52 YCoordinatesOfNodes=[100 100 400 400 500 200 300 800
      350 550 750 850 550 550 150 100 100 600 700 400
     700 650 800 900 500 400 900 950 250 500];//y-
     coordinate of each node
```

```
53 [TopologyGraph]=NL_G_MakeGraph(NameOfNetwork,
      NumberOfNodes,EndingNodesOfConnection,
      StartNodesOfConnection,XCoordinatesOfNodes,
      YCoordinatesOfNodes)//application of
      NL_G_MakeGraph
54
55 NetworkSize=length(TopologyGraph.node_x);//real
      network size
56 NetworkQuantity=length(TopologyGraph.head);//
      quantity of network links
57 [Node1,Node2]=NL_F_RandIntNiNj(NetworkSize)//
      selection of two distinct nodes
58 TTL=15;//Time-To-Live update
59 [path]=NL_R_Flood(TopologyGraph,Node1,Node2,TTL)//
      application of NL_R_Flood
60 ShortestPath=NL_G_Nodes2Path(path,TopologyGraph);//
      links of the shortest path
61 EdgeColor=ones(1,NetworkQuantity);//display the path
       between i and j: edge color
62 EdgeBorder=1.5*ones(1,NetworkQuantity);//edge width
63 EdgeColor(ShortestPath)=5;//define path color
64 EdgeBorder(ShortestPath)=5;//define width
65 NodeBorder=4*ones(1,NetworkSize);//node border
66 NodeBorder(path)=10;//node border for source to
      destination path
67 TopologyGraph.node_border=NodeBorder;//node border
68 TopologyGraph.edge_color=EdgeColor;//define edge
      color
69 TopologyGraph.edge_width=EdgeBorder;//edge width
70 WindowNumber=2;//window index
71 GraphVisualize=NL_G_ShowGraphN(TopologyGraph,
      WindowNumber);//graph visualization
72
73 xtitle("Shortesh Path in Tree Topology for 30 Nodes"
      ,"X-Nodes","Y-Nodes")
```

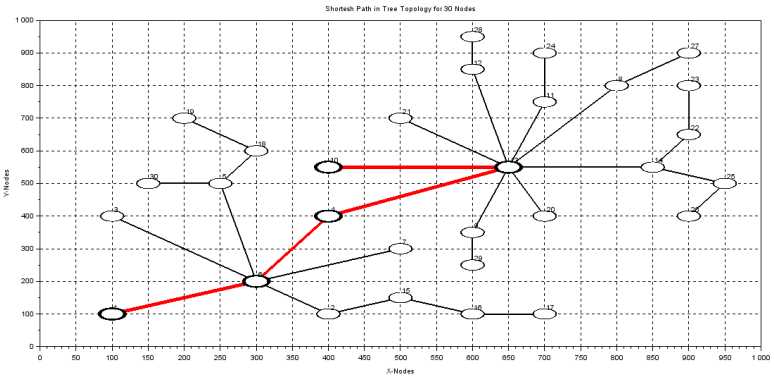

Figure 10.2: Shortest Path in Tree Topology

Experiment: 11

Creating a Network Square Area & Coloured Network Topology Creation

Scilab code Solution 11.1 Creation of Network Square Area and Coloured Network Topology

```
1 //Experiment No.11
2 // This file must be used under the terms of the
    CeCILL.
3 // This source file is licensed as described in the
    file COPYING, which
4 // you should have received as part of this
    distribution. The terms
5 // are also available at
6 // http://www.cecill.info/licences/Licence_CeCILL_V2
    -en.txt
7 //This Source file is Written by Dr. T. Subbulaskhmi
    , Professor,
```

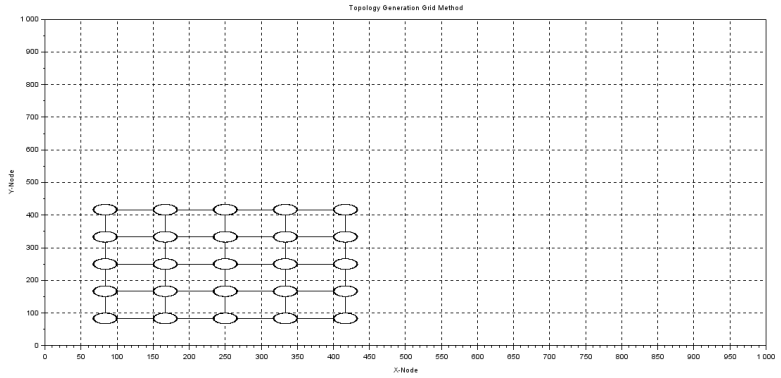

Figure 11.1: Creation of Network Square Area and Coloured Network Topology

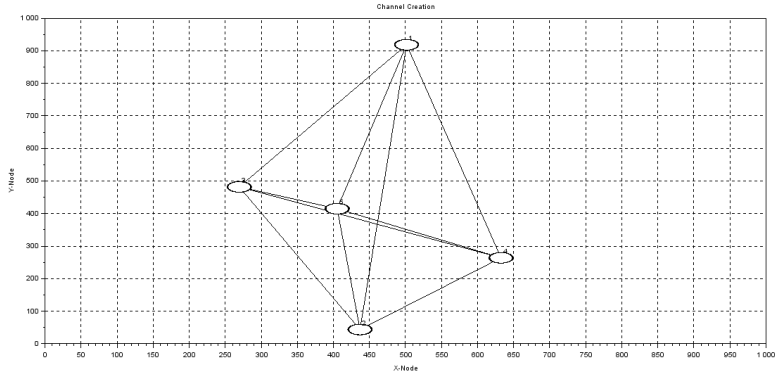

Figure 11.2: Creation of Network Square Area and Coloured Network Topology

```
8  //School of Computing Science and Engineering, VIT
       University Chennai
9  //using the NARVAL examples of Scilab for Network
       Topology Creation
10 //The Operating System used for writing the code
       found in this file is Windows 8
11 //SCILAB version 5.5.2 and NARVAL toolbox version
       3.1//This Program is Written by Dr. T.
       Subbulaskhmi, Professor, School of Computing
       Science and Engineering, VIT University Chennai
       using the NARVAL examples of Scilab for Network
       Topology Creation and display using various
       methods
12
13 //1.Topology creation and colouring using grid
       method 2. Topology creation and colouring using
       random method 3. Topology Creation using detailed
        method
14
15
16 //1.Topology creation and colouring using grid
       method
17 clear;
18 clf;
19
20 NumberOfRows=10;// number of rows
21 NumberOfColumns=10;//number of columns
22 XCoordinatesOfArea=1000;//Area of network x
       coordinates
23 YCoordinatesOfArea=1000;//Area of network y
       coordinates
24 radius = 100;
25 colour = 2;
26 [TopologyGraph]=NL_T_Grid(NumberOfRows,
       NumberOfColumns,XCoordinatesOfArea,
       YCoordinatesOfArea);// generation of a topology
       in respect with the grid method
27 WindowIndex=1;//window index
```

```
28 NameOfNetwork = "Network Created with Grid and
      Showgraph method";
29 InitialSquareArea=NL_M_Background(WindowIndex,
      NameOfNetwork);//initial square area
30
31 xlabel("x coordinates of node", "fontsize", 2)
32 ylabel("Y coordinates of node");
33
34 InitialSquareArea=NL_M_GraphDisplayUpdate(
      TopologyGraph,WindowIndex,radius,colour);
35
36 //2. Topology creation and colouring using random
      method
37 NetworkSize=5;//network size
38 NetworkAreaSide=1000;//network squared area side
39 LocalityRadius=1000;//Locality radius
40 [TopologyGraph]=NL_T_LocalityConnex(NetworkSize,
      NetworkAreaSide,LocalityRadius);//generation of a
       topology
41 WindowIndex=2;//window index
42 InitialSquareArea=NL_G_ShowGraphN(TopologyGraph,
      WindowIndex);//graph visualization
43 xtitle("Channel Creation","X-Node","Y-Node");
44 [go,n,e]=NL_G_WCDS(TopologyGraph)
45 ListOfChannels=[2];//list of chanels
46 [goc]=NL_G_WCDSChannel(go,NetworkSize,e,
      ListOfChannels);//application of NL_G_WCDSChannel
47 WindowIndex=3;//window index
48 InitialSquareArea=NL_G_ShowGraph(goc,WindowIndex);//
      graph visualization
49 xtitle("Channel Creation","X-Node","Y-Node");
50 disp(go,NetworkSize,e,ListOfChannels);
51
52
53 //3. Topology Creation using detailed method
54 NameOfNetwork='Detailed topology';// Name of your
      network
55 NetworkSize=8;//Number of Nodes in the network
```

```
56 StartingNodes=[1 2 3 4 1 2 3 4 6 7 8 6 7 8]; //
     Starting Nodes of the connection lines
57 EndingNodes=[2 3 4 5 3 6 7 8 2 4 5 1 2 3]; //Ending
     Node of the connection
58 XCoordinatesOfNode=[100 200 500 300 400 600 700
     400]; // X-Coordinates of the nodes
59 YCoordinatesOfNode=[300 400 500 600 700 800 900
     950]; // Y-Coordinates of the nodes
60 [g1]=NL_G_MakeGraph(NameOfNetwork,NetworkSize,
     StartingNodes,EndingNodes,XCoordinatesOfNode,
     YCoordinatesOfNode)//Creates the Bus topoplogy
61 WindowIndex=4; //Graph Window Number
62 [f1] = NL_G_ShowGraph(g1,WindowIndex);// Visualize
     the Graph along with indices for Nodes and Edges
63 xtitle("Bus Topology Creation","X-Node","Y-Node");
64 NumberOfLines=5;//number of lines
65 NumberOfColumns=5;//number of columns
66 XCoordinatesOfArea=1000;//network area x-side
67 YCoordinatesOfArea=1000;//network area x-side
68 [TopologyGraph]=NL_T_Grid(NumberOfLines,
     NumberOfColumns,XCoordinatesOfArea,
     YCoordinatesOfArea);//generation of a topology in
      respect with the grid method
69 WIndowIndex=5;//window index
70 [f]=NL_G_ShowGraph(TopologyGraph,WIndowIndex);//
     application of NL_G_ShowGraph
71 xtitle("Nodes mentioned in node vector area squae
     1000","X-Node","Y-Node");
72 NumberOfLines=5;//number of lines
73 NumberOfColumns=7;//number of columns
74 XCoordinatesOfArea=500;//network area x-side
75 YCoordinatesOfArea=500;//network area x-side
76 [TopologyGraph]=NL_T_Grid(NumberOfLines,
     NumberOfColumns,XCoordinatesOfArea,
     YCoordinatesOfArea);//generation of a topology in
      respect with the grid method
77 WindowIndex=6;//window index
78 [f]=NL_G_ShowGraph(TopologyGraph,WindowIndex);//
```

```
    application of NL_G_ShowGraph
79  xtitle("Nodes mentioned in node vector","X-Node","Y-
    Node");
80
81  NumberOfLines=5;//number of lines
82  NumberOfColumns=5;//number of columns
83  XCoordinatesOfArea=500;//network area x-side
84  YCoordinatesOfArea=500;//network area x-side
85  [TopologyGraph]=NL_T_Grid(NumberOfLines,
    NumberOfColumns,XCoordinatesOfArea,
    YCoordinatesOfArea);//generation of a topology in
     respect with the grid method
86  WindowIndex=7;//window index
87  [f]=NL_G_ShowGraph(TopologyGraph,WindowIndex);//
    application of NL_G_ShowGraph
88  xtitle("Topology Generation Grid Method","X-Node","Y
    -Node");
89  //4. Display the number of nodes and edges
90  [ExtractNode,ExtractEdge]=NL_G_GraphSize(
    TopologyGraph);//Extract the number of nodes and
    edges
91  disp('Number of nodes:',ExtractNode); //display the
    number of nodes and edges
92  disp('Number of edges:',ExtractEdge);
93
94  //2. Colour the nodes and Edges
95  NodeColor=30; // Node Colour 2:[Blue],3:[Green], 5:[
    Red]
96  BorderThickness=10; // Node Border thickness
97  NodeDiameter=25; //Node diameter
98  WindowIndex=8;//window index
99  nodes=[1 2 4 6 8 10 12 14];//list of nodes
100 [GraphHighlight,NodeVector]=NL_G_HighlightNodes(
    TopologyGraph,nodes,NodeColor,BorderThickness,
    NodeDiameter,WindowIndex);//Highlight the
    specific nodes mentioned in the 'nodes' vector
101 xtitle("Nodes mentioned in node vector","X-Node","Y-
    Node");
```

```
102 EdgeColor=5;// Edge Colour
103 EdgeWidth=5;//Edge Width
104 WIndowIndex=9;//window index
105 edges=[1 5 7];//list of edges
106 [GraphHighlight,NodeVector]=NL_G_HighlightEdges(
      GraphHighlight,edges,EdgeColor,EdgeWidth,
      WIndowIndex);//Highlight the specific nodes
      mentioned in the 'edges' vector
107 xtitle("Nodes mentioned in edge vector","X-Node","Y-
      Node");
```

Experiment: 12

Finding the shortest path by using Dijikstra's Algorithm

Scilab code Solution 12.1 Shortest path using Dijikstras Algorithm

```
1 //Experiment No.12
2 // This file must be used under the terms of the
    CeCILL.
3 // This source file is licensed as described in the
    file COPYING, which
4 // you should have received as part of this
    distribution. The terms
5 // are also available at
6 // http://www.cecill.info/licences/Licence_CeCILL_V2
    -en.txt
7 //This Source file is Written by Dr. T. Subbulaskhmi
    , Professor,
8 //School of Computing Science and Engineering, VIT
    University Chennai
9 //using the NARVAL examples of Scilab for Network
```

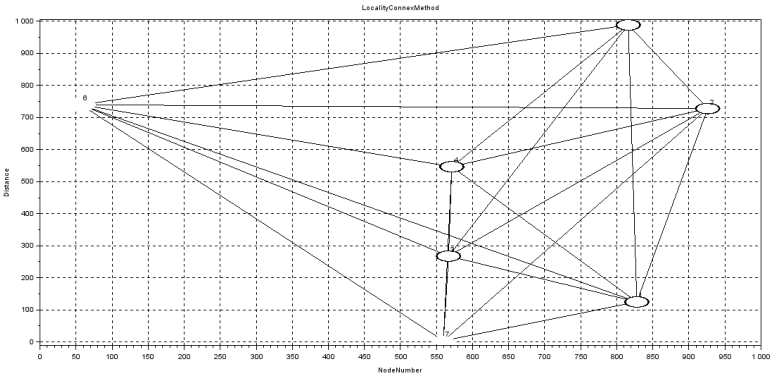

Figure 12.1: Shortest path using Dijikstras Algorithm

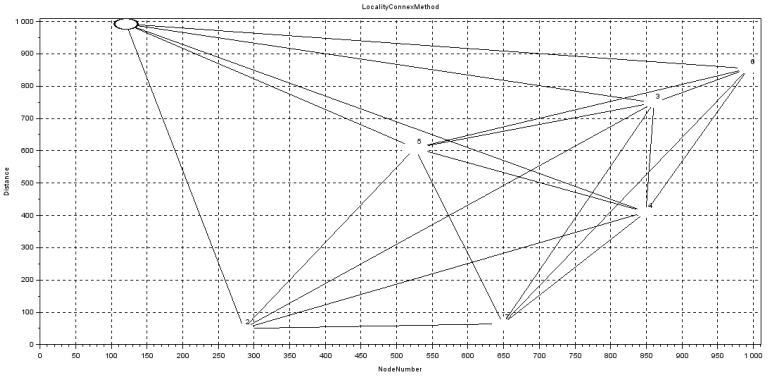

Figure 12.2: Shortest path using Dijikstras Algorithm

```
    Topology Creation
10 //The Operating System used for writing the code
    found in this file is Windows 8
11 //SCILAB version 5.5.2 and NARVAL toolbox version
    3.1//This Program is Written by Dr. T.
    Subbulaskhmi, Professor, School of Computing
    Science and Engineering, VIT University Chennai
    using the NARVAL examples of Scilab for Network
    Topology Creation
12
13 clear;
14 clc;
15
16 NetworkSize=7;//network size
17 NeworkSquareArea=1000;//network square area side
18 LocalityRadius=1000;//locality radius
19 [TopologyGraph]=NL_T_LocalityConnex(NetworkSize,
    NeworkSquareArea,LocalityRadius);//generation of
    a random topology in respect with the Locality
    method.
20 for i=1:1:NetworkSize
21     disp(i,"node number :");
22     [dist,pred]=NL_R_Dijkstra(TopologyGraph,i);//
        application of NL_R_Dijkstra
23     j=dist;
24     for x = j
25     disp(x);
26     plot(j);
27     xlabel("Nodenumber", "fontsize", 2)
28     ylabel("Distance");
29     end
30 end
31 for i=1:1:NetworkSize
32     [TopologyGraph]=NL_T_LocalityConnex(NetworkSize,
        NeworkSquareArea,LocalityRadius);//generation
         of a random topology in respect with the
        Locality method.
33     [dist,pred]=NL_R_Dijkstra(TopologyGraph,i);//
```

```
        application of NL_R_Dijkstra
34      TopologyGraph.node_diam(i)=40;//node diameter
35      TopologyGraph.node_border(i)=10;//node border
36      TopologyGraph.node_color(i)=5;//node color
37     [GraphVisualize]=NL_G_ShowGraphN(TopologyGraph,i)
          ;//graph visualization
38  xtitle("LocalityConnexMethod","NodeNumber","Distance
     ");
39  end
```

Experiment: 13

Finding the shortest path by using Prim's Algorithm

Scilab code Solution 13.1 Shortest Path Using Prims Algorithm

```
1 //Experiment No.13
2 // This file must be used under the terms of the
    CeCILL.
3 // This source file is licensed as described in the
    file COPYING, which
4 // you should have received as part of this
    distribution. The terms
5 // are also available at
6 // http://www.cecill.info/licences/Licence_CeCILL_V2
    -en.txt
7 //This Source file is Written by Dr. T. Subbulaskhmi
    , Professor,
8 //School of Computing Science and Engineering, VIT
    University Chennai
9 //using the NARVAL examples of Scilab for Network
```

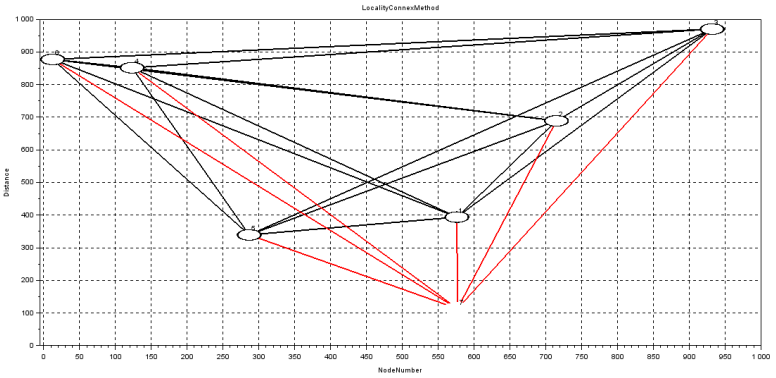

Figure 13.1: Shortest Path Using Prims Algorithm

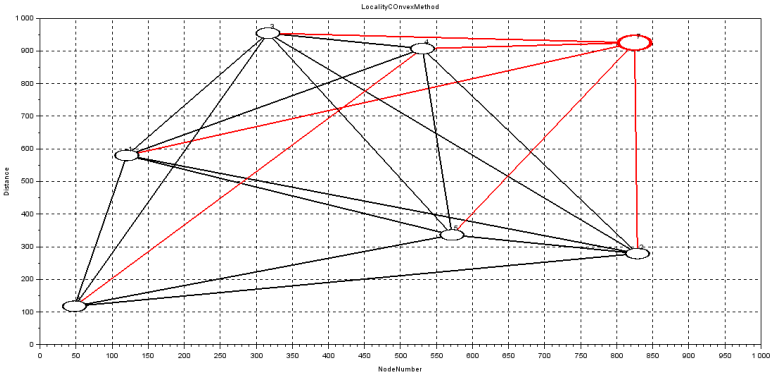

Figure 13.2: Shortest Path Using Prims Algorithm

```
    Topology Creation
10 //The Operating System used for writing the code
    found in this file is Windows 8
11 //SCILAB version 5.5.2 and NARVAL toolbox version
    3.1//This Program is Written by Dr. T.
    Subbulaskhmi, Professor, School of Computing
    Science and Engineering, VIT University Chennai
    using the NARVAL examples of Scilab for Network
    Topology Creation
12
13 clear;
14 clc;
15
16 NetworkSize=7;//network size
17 NeworkSquareArea=1000;//network square area side
18 LocalityRadius=1000;//locality radius
19 [ToplogyGraph]=NL_T_LocalityConnex(NetworkSize,
    NeworkSquareArea,LocalityRadius);//generation of
    a random topology in respect with the Locality
    method.
20 for SourceNode=1:1:NetworkSize
21     disp(SourceNode,"node number :");
22     dw = 2;
23     WindowSize = SourceNode;
24     [go,v,pred]=NL_R_Prim(ToplogyGraph,SourceNode,dw
        ,WindowSize)//application of NL_R_Prim
25 //     disp(go);
26     disp(v);
27     disp(pred);
28     j = pred;
29     for x = j
30     disp(x);
31     plot(j);
32     xlabel("Nodenumber", "fontsize", 2)
33     ylabel("Distance");
34     end
35 end
36 for SourceNode=1:1:NetworkSize
```

```
37      [ToplogyGraph]=NL_T_LocalityConnex(NetworkSize,
           NeworkSquareArea,LocalityRadius);//generation
            of a random topology in respect with the
           Locality method.
38      dw = 2;
39      WindowSize = SourceNode;
40      [go,v,pred]=NL_R_Prim(ToplogyGraph,SourceNode,dw
           ,WindowSize)//application of NL_R_Prim
41  //      [dist,pred]=NL_R_Dijkstra(g,i);//application
        of NL_R_Dijkstra
42      go.node_diam(SourceNode)=40;//node diameter
43      go.node_border(SourceNode)=10;//node border
44      go.node_color(SourceNode)=5;//node color
45      [GraphVisualize]=NL_G_ShowGraphN(go,SourceNode);
           //graph visualization
46  xtitle("LocalityConnexMethod","NodeNumber","Distance
       ")
47  end
48  5
```

Experiment: 14

Manet Simulation

Scilab code Solution 14.1 Manet Simulation

```
1 //Experiment No.14
2 // This file must be used under the terms of the
    CeCILL.
3 // This source file is licensed as described in the
    file COPYING, which
4 // you should have received as part of this
    distribution. The terms
5 // are also available at
6 // http://www.cecill.info/licences/Licence_CeCILL_V2
    -en.txt
7 //This Source file is Written by Dr. T. Subbulaskhmi
    , Professor,
8 //School of Computing Science and Engineering, VIT
    University Chennai
9 //using the NARVAL examples of Scilab for Network
    Topology Creation
10 //The Operating System used for writing the code
```

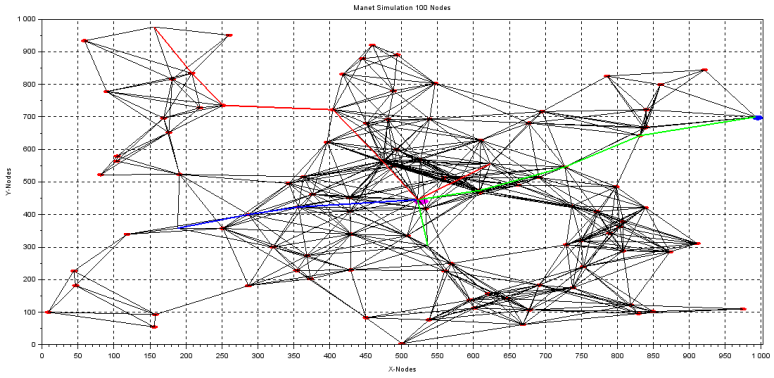

Figure 14.1: Manet Simulation

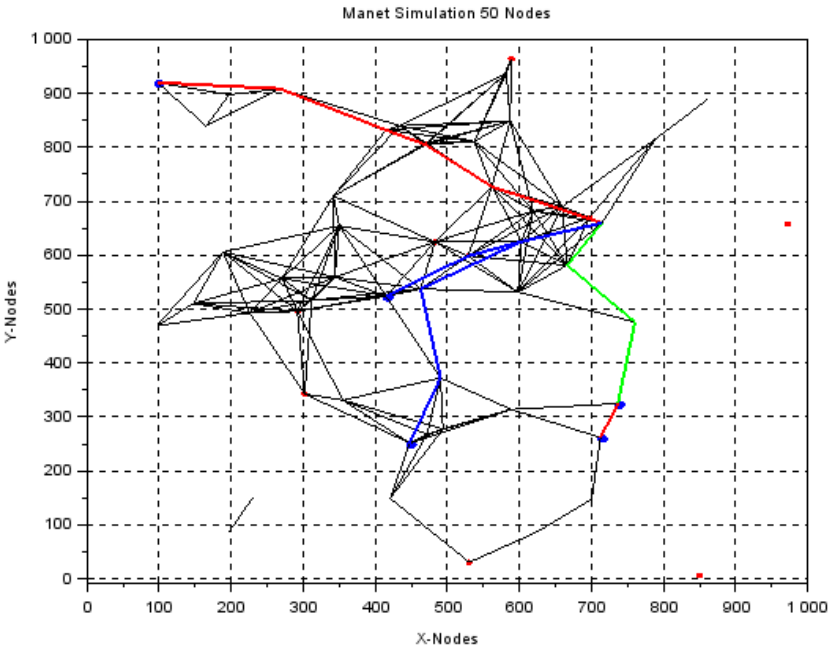

Figure 14.2: Manet Simulation

```
    found in this file is Windows 8
11 //SCILAB version 5.5.2 and NARVAL toolbox version
    3.1//This Program is Written by Dr. T.
    Subbulaskhmi, Professor, School of Computing
    Science and Engineering, VIT University Chennai
    using the NARVAL examples of Scilab for Network
    Topology Creation
12
13 clear;
14 clc;
15
16 RadiusMovingNode=10;//display radius of moving nodes
17 RadiusFixedNodes=15;//display radius of fixed nodes
18 MovingNodeConnection=20;//display radius of the
    moving nodes belonging to the connection under
    studies
19 NodeQuantity=50;//5quantity of moving nodes
20 FixedNodeQuantity=5;//quantity of fixed nodes
21 NetworkArea=1000;//network square area side
22 MaximumSpeed=20;//maximum speed
23 MinSimuDuration=100;//simulation duration
24 MaxSimuDuration=100;//maximal waiting time
25 RadiusForLinks=180;//Locality radius for the links
    attribution
26 WindowIndex=1;//window index
27 NL_M_Simulation1N2AllAP(RadiusMovingNode,
    RadiusFixedNodes,MovingNodeConnection,
    NodeQuantity,FixedNodeQuantity,NetworkArea,
    MaximumSpeed,MinSimuDuration,MaxSimuDuration,
    RadiusForLinks,WindowIndex);//application of
    NL_M_Simulation1N2AllAP
28 xtitle("Manet Simulation 50 Nodes","X-Nodes","Y-
    Nodes")
29
30 RadiusMovingNode=10;//display radius of moving nodes
31 RadiusFixedNodes=15;//display radius of fixed nodes
32 MovingNodeConnection=20;//display radius of the
    moving nodes belonging to the connection under
```

```
    studies
33 NodeQuantity=100;//quantity of moving nodes
34 FixedNodeQuantity=5;//quantity of fixed nodes
35 NetworkArea=1000;//network square area side
36 MaximumSpeed=50;//maximum speed
37 MinSimuDuration=25;//simulation duration in secs
38 MaxSimuDuration=10;//maximal waiting time in secs
39 RadiusForLinks=180;//Locality radius for the links
    attribution
40 WindowIndex=2;//window index
41 NL_M_Simulation1N2AllAP(RadiusMovingNode,
    RadiusFixedNodes,MovingNodeConnection,
    NodeQuantity,FixedNodeQuantity,NetworkArea,
    MaximumSpeed,MinSimuDuration,MaxSimuDuration,
    RadiusForLinks,WindowIndex);//application of
    NL_M_Simulation1N2AllAP
42 xtitle("Manet Simulation 100 Nodes","X-Nodes","Y-
    Nodes")
43
44 RadiusMovingNode=5;//display radius of moving nodes
45 RadiusFixedNodes=10;//display radius of fixed nodes
46 MovingNodeConnection=20;//display radius of the
    moving nodes belonging to the connection under
    studies
47 NodeQuantity=200;//quantity of moving nodes
48 FixedNodeQuantity=5;//quantity of fixed nodes
49 NetworkArea=1000;//network square area side
50 MaximumSpeed=20;//maximum speed
51 MinSimuDuration=100;//simulation duration
52 MaxSimuDuration=100;//maximal waiting time
53 RadiusForLinks=180;//Locality radius for the links
    attribution
54 WindowIndex=3;//window index
55 NL_M_Simulation1N2AllAP(RadiusMovingNode,
    RadiusFixedNodes,MovingNodeConnection,
    NodeQuantity,FixedNodeQuantity,NetworkArea,
    MaximumSpeed,MinSimuDuration,MaxSimuDuration,
    RadiusForLinks,WindowIndex);//application of
```

```
    NL_M_Simulation1N2AllAP
56 xtitle("Manet Simulation 200 Nodes","X-Nodes","Y-
    Nodes")
```

Experiment: 15

Congestion Control

Scilab code Solution 15.1 Congestion Control

```
1 //Experiment No.15
2 // This file must be used under the terms of the
    CeCILL.
3 // This source file is licensed as described in the
    file COPYING, which
4 // you should have received as part of this
    distribution. The terms
5 // are also available at
6 // http://www.cecill.info/licences/Licence_CeCILL_V2
    -en.txt
7 //This Source file is Written by Dr. T. Subbulaskhmi
    , Professor,
8 //School of Computing Science and Engineering, VIT
    University Chennai
9 //using the NARVAL examples of Scilab for Network
    Topology Creation
10 //The Operating System used for writing the code
```

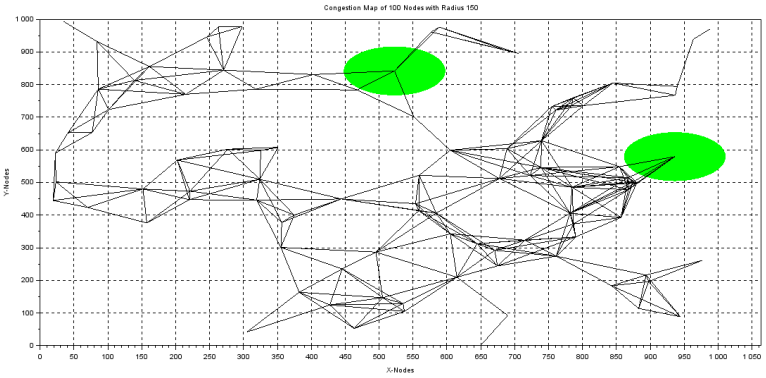

Figure 15.1: Congestion Control

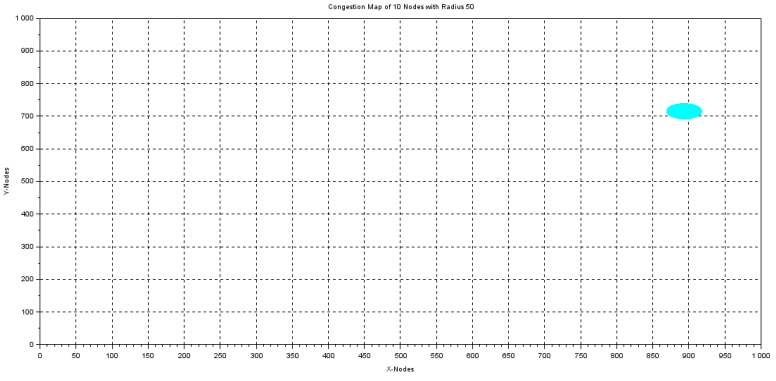

Figure 15.2: Congestion Control

```
    found in this file is Windows 8
11 //SCILAB version 5.5.2 and NARVAL toolbox version
    3.1//This Program is Written by Dr. T.
    Subbulaskhmi, Professor, School of Computing
    Science and Engineering, VIT University Chennai
    using the NARVAL examples of Scilab for Network
    Topology Creation
12
13 clear;
14 clc;
15
16 NetworkSize=100;//network size
17 NetworkSquareArea=1000;//network squared area side
18 LocalityRadius=50;//Locality radius
19 Color=1;//color
20 [TopologyGraph]=NL_T_LocalityConnex(NetworkSize,
    NetworkSquareArea,LocalityRadius);//generation of
     a topology
21 WindowIndex=1;//window index
22 NameOfNetwork="Congestion Map of 100 nodes with r=50
    "; // Name of the graph
23 InitialSquareArea=NL_M_Background(WindowIndex,
    NameOfNetwork);//initial square area
24 InitialSquareArea=NL_M_GraphDisplayUpdate(
    TopologyGraph,WindowIndex,LocalityRadius,Color);
25 //f=NL_G_ShowGraphN(g,w);//graph visualization
26 n=TopologyGraph.node_number;//graph size
27 sink=NL_F_RandInt1n(n);//selection of the sink
28 [cm,np,pred]=NL_R_CongestionSinkFlood(TopologyGraph,
    sink)//application of NL_R_CongestionSinkFlood
29 xtitle("Congestion Map of 100 Nodes with Radius 50",
     "X-Nodes","Y-Nodes");
30 NetworkSize=100;//network size
31 NetworkSquareArea=1000;//network squared area side
32 LocalityRadius=100;//Locality radius
33 Color=2;//color
34 [TopologyGraph]=NL_T_LocalityConnex(NetworkSize,
    NetworkSquareArea,LocalityRadius);//generation of
```

```
    a topology
35 WindowIndex=2;//window index
36 NameOfNetwork="Congestion Map of 100 nodes with r
   =100";
37 InitialSquareArea=NL_M_Background(WindowIndex,
   NameOfNetwork);//initial square area
38 InitialSquareArea=NL_M_GraphDisplayUpdate(
   TopologyGraph,WindowIndex,LocalityRadius,Color);
39 //f=NL_G_ShowGraphN(g,w);//graph visualization
40 n=TopologyGraph.node_number;//graph size
41 sink=NL_F_RandInt1n(n);//selection of the sink
42 [cm,np,pred]=NL_R_CongestionSinkFlood(TopologyGraph,
   sink)//application of NL_R_CongestionSinkFlood
43 xtitle("Congestion Map of 100 Nodes with Radius 100"
   , "X-Nodes","Y-Nodes");
44 NetworkSize=100;//network size
45 NetworkSquareArea=1000;//network squared area side
46 LocalityRadius=150;//Locality radius
47 Color=3;//color
48 [TopologyGraph]=NL_T_LocalityConnex(NetworkSize,
   NetworkSquareArea,LocalityRadius);//generation of
    a topology
49 WindowIndex=3;//window index
50 NameOfNetwork="Congestion Map of 100 nodes with r
   =150";
51 InitialSquareArea=NL_M_Background(WindowIndex,
   NameOfNetwork);//initial square area
52 InitialSquareArea=NL_M_GraphDisplayUpdate(
   TopologyGraph,WindowIndex,LocalityRadius,Color);
53 //f=NL_G_ShowGraphN(g,w);//graph visualization
54 n=TopologyGraph.node_number;//graph size
55 sink=NL_F_RandInt1n(n);//selection of the sink
56 [cm,np,pred]=NL_R_CongestionSinkFlood(TopologyGraph,
   sink)//application of NL_R_CongestionSinkFlood
57 xtitle("Congestion Map of 100 Nodes with Radius 150"
   , "X-Nodes","Y-Nodes");
58 NetworkSize=10;//network size
59 NetworkSquareArea=1000;//network squared area side
```

```
60 LocalityRadius=50;//Locality radius
61 Color=4;//color
62 [TopologyGraph]=NL_T_LocalityConnex(NetworkSize,
      NetworkSquareArea,LocalityRadius);//generation of
       a topology
63 WindowIndex=4;//window index
64 NameOfNetwork="Congestion Map of 10 nodes with r=50"
      ;
65 f=NL_M_Background(WindowIndex,NameOfNetwork);//
      initial square area
66 InitialSquareArea=NL_M_GraphDisplayUpdate(
      TopologyGraph,WindowIndex,LocalityRadius,Color);
67 //f=NL_G_ShowGraphN(g,w);//graph visualization
68 n=TopologyGraph.node_number;//graph size
69 sink=NL_F_RandInt1n(n);//selection of the sink
70 [cm,np,pred]=NL_R_CongestionSinkFlood(TopologyGraph,
      sink)//application of NL_R_CongestionSinkFlood
71 xtitle("Congestion Map of 10 Nodes with Radius 50",
      "X-Nodes","Y-Nodes");
72 NetworkSize=10;//network size
73 NetworkSquareArea=1000;//network squared area side
74 LocalityRadius=100;//Locality radius
75 Color=5;//color
76 [TopologyGraph]=NL_T_LocalityConnex(NetworkSize,
      NetworkSquareArea,LocalityRadius);//generation of
       a topology
77 WindowIndex=5;//window index
78 NameOfNetwork="Congestion Map of 10 nodes with r=100
      ";
79 InitialSquareArea=NL_M_Background(WindowIndex,
      NameOfNetwork);//initial square area
80 InitialSquareArea=NL_M_GraphDisplayUpdate(
      TopologyGraph,WindowIndex,LocalityRadius,Color);
81 //f=NL_G_ShowGraphN(g,w);//graph visualization
82 n=TopologyGraph.node_number;//graph size
83 sink=NL_F_RandInt1n(n);//selection of the sink
84 [cm,np,pred]=NL_R_CongestionSinkFlood(TopologyGraph,
      sink)//application of NL_R_CongestionSinkFlood
```

```
85 xtitle("Congestion Map of 10 Nodes with Radius 100",
       "X-Nodes","Y-Nodes");
86 NetworkSize=10;//network size
87 NetworkSquareArea=1000;//network squared area side
88 LocalityRadius=150;//Locality radius
89 Color=6;//color
90 [TopologyGraph]=NL_T_LocalityConnex(NetworkSize,
       NetworkSquareArea,LocalityRadius);//generation of
       a topology
91 WindowIndex=6;//window index
92 NameOfNetwork="Congestion Map of 10 nodes with r=150
       ";
93 InitialSquareArea=NL_M_Background(WindowIndex,
       NameOfNetwork);//initial square area
94 InitialSquareArea=NL_M_GraphDisplayUpdate(
       TopologyGraph,WindowIndex,LocalityRadius,Color);
95 //f=NL_G_ShowGraphN(g,w);//graph visualization
96 n=TopologyGraph.node_number;//graph size
97 sink=NL_F_RandInt1n(n);//selection of the sink
98 [cm,np,pred]=NL_R_CongestionSinkFlood(TopologyGraph,
       sink)//application of NL_R_CongestionSinkFlood
99 xtitle("Congestion Map of 10 Nodes with Radius 150",
       "X-Nodes","Y-Nodes");
```