Contents

List of Experiments

List of Figures

Experiment: 1

Creation of the Bus topology

Scilab code Solution 1.1 Bus Topology Creation

```
//Experiment No.1
// This file must be used under the terms of the
   CeCILL.
// This source file is licensed as described in the
   file COPYING, which
// you should have received as part of this
   distribution.  The terms
// are also available at
// http://www.cecill.info/licences/Licence_CeCILL_V2
   -en.txt
//This Source file 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
//The Operating System used for writing the code
```

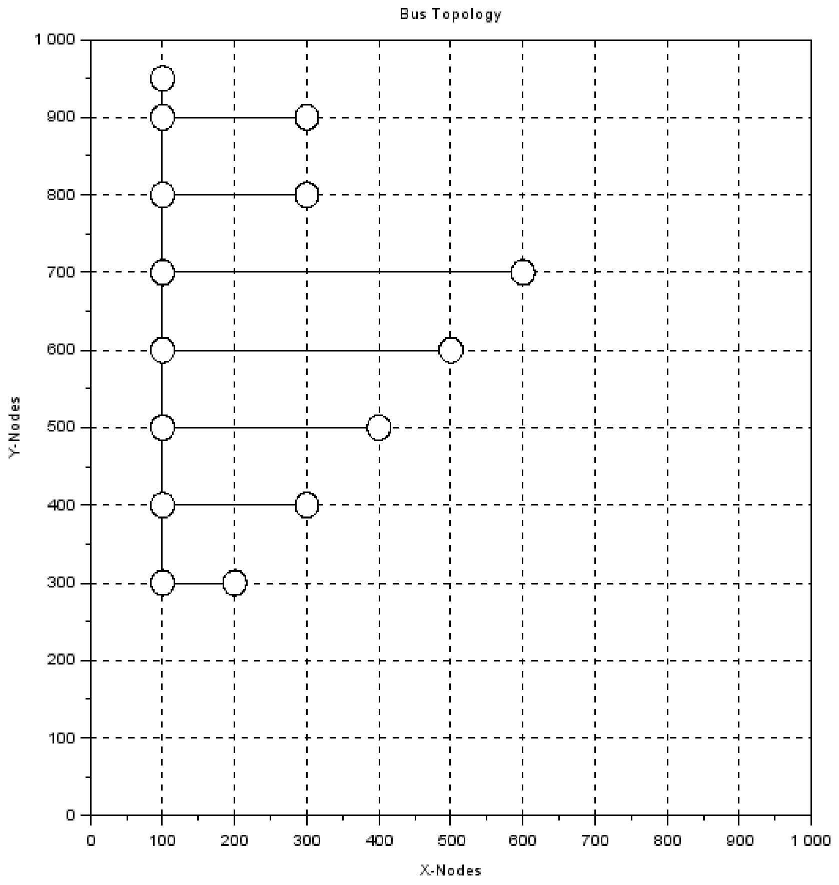

Figure 1.1: Bus Topology Creation

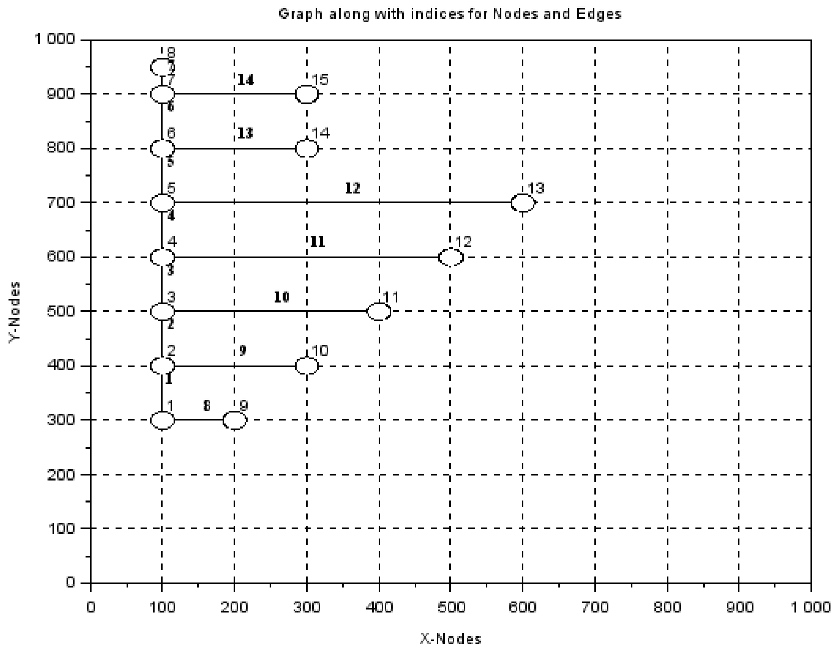

Figure 1.2: Bus Topology Creation

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

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

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

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

```
//3. Display the node number and edge number
WindowIndex=4; //Graph Window Number
[GraphVisualise] = NL_G_ShowGraphNE(TopologyGraph,
   WindowIndex);// Visualize the Graph along with
   indices for Nodes and Edges
xtitle("Graph along with indices for Nodes and Edges
   ","X-Nodes","Y-Nodes");
//4. Display the number of nodes and edges
[ExtractedNode,ExtractedEdge]=NL_G_GraphSize(
   TopologyGraph);//Extract the number of nodes and
   edges
disp('Number of nodes:',ExtractedNode); //display
   the number of nodes and edges
disp('Number of edges:',ExtractedEdge);

//5.Display the Edge Index between two nodes
StartingNode=4;//starting node
EndingNode=5;//ending node
[GetEdgeIndex]=NL_G_Nodes2Edge(TopologyGraph,
   StartingNode,EndingNode);//Get the Edge Index
disp('Index of Edge across nodes 4 and 5:',
   GetEdgeIndex);

//6. Display the length of every edge
[EdgeLength]=NL_G_EdgesLength(TopologyGraph.node_x,
   TopologyGraph.node_y,TopologyGraph.head,
   TopologyGraph.tail);//Display the length of every
    edge
disp('length of all edges',EdgeLength);

//7. Return the set of edges connected to a node
   inside a graph
NodeIndex=2;//node index
[GetEdgeIndex]=NL_G_EdgesOfNode(TopologyGraph,
   NodeIndex);//application of NL_G_EdgesOfNode
disp('Internal Edges of node 2:',GetEdgeIndex);

//8. Extract the data fields of the node
```

```
[node_x,node_y,node_border,node_diameter,node_color,
   node_number]=NL_G_NodeDataFields(TopologyGraph);
   // Extracting the node data fields
disp('Node x :',XCoordinatesOfNodes); // X
   Coordinates
disp('Node y :', YCoordinatesOfNodes);// Y
   Coordinates
disp('Node Borders :', node_border); //Node Border
disp('Node Diameter :', node_diameter); // Diameter
   of the node
disp('Node Colour :', node_color); // Node colour
disp('Node Number :', node_number); //Number of
   Nodes

//9.Extract the data fields of the Edge
[e_head,e_tail,e_color,e_width,e_length,e_weight,
   e_number]=NL_G_EdgeDataFields(TopologyGraph)//
   application of NL_G_EdgeDataFields
disp('Head Nodes',e_head); //Head node details
disp('Tail Nodes',e_tail); // Tail Node details
disp('Edge Colours',e_color); // Colour of edges
disp('Edge Widths',e_width); // Width of edges
disp('Edge Lengths',e_length); // Length of the
   edges
disp('Edge Weights',e_weight); //Weight of the edges
disp('Edge Numbers',e_number); // Edge Number
```

Experiment: 2

Creation of the Ring topology

Scilab code Solution 2.1 Ring Topology Creation

```


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

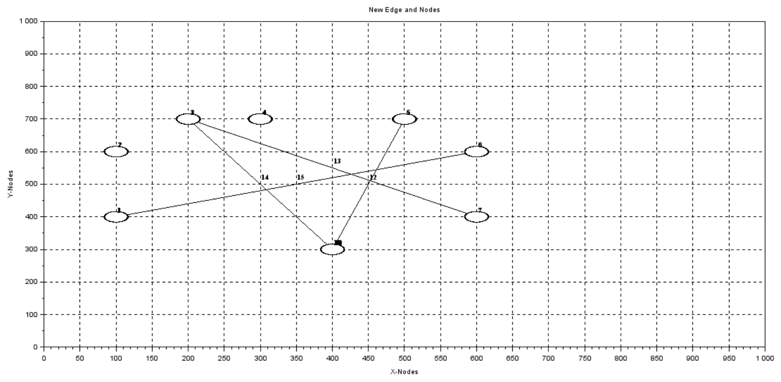

Figure 2.1: Ring Topology Creation

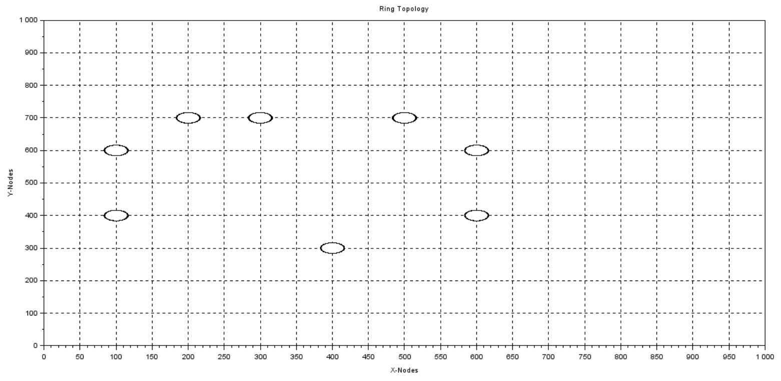

Figure 2.2: Ring Topology Creation

```
    Topology Creation
//The Operating System used for writing the code
    found in this file is Windows 8
//SCILAB version 5.5.2 and NARVAL toolbox version
    3.1
//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.

clear;
clc;

//1. create and Display Ring Topology
NameOfNetwork='Ring Topology';// Name of your
    network
NumberOfNodes=8;//Number of Nodes in the network
ConnectionEndingNode=[1 2 3 4 5 6 7 8]; //Ending
    Nodes of the connection lines
ConnectionStartingNode=[2 3 4 5 6 7 8 1];//Starting
    Nodes of the connection lines
XCoordinatedOfNodes=[100 100 200 300 500 600 600
    400]; // X-Coordinates of the nodes
YCoordinatedOfNodes=[400 600 700 700 700 600 400
    300]; // Y-Coordinates of the nodes
[TopologyGraph]=NL_G_MakeGraph(NameOfNetwork,
    NumberOfNodes,ConnectionEndingNode,
    ConnectionEndingNode,XCoordinatedOfNodes,
    YCoordinatedOfNodes)//Creates the Ring topoplogy

WindowIndex=1;//Graph Window Number

[VisualizeGraph1] = NL_G_ShowGraph(TopologyGraph,
    WindowIndex);// Visualize the Graph

```

```
//2. Colour the nodes
NodeColor=30; // Node Colour 2:[Blue],3:[Green], 5:[Red]
BorderThickness=10; // Node Border thickness
NodeDiameter=25; //Node diameter
WindowIndex=2;//window index
ListOfNodes=[1 3 5 7];//list of nodes
[NodeHighlight,VisualizeGraph1]=NL_G_HighlightNodes(TopologyGraph,ListOfNodes,NodeColor,BorderThickness,NodeDiameter,WindowIndex);//Highlight the specified nodes

WindoeIndex=3; //Graph Window Number
[VisualizeGraph2] = NL_G_ShowGraphNE(TopologyGraph,WindoeIndex);// Visualize the Graph along with indices for Nodes and Edges

// 4. Display the number of 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

//5. insert new nodes in the specified edge
EdgeIndex=8;//edge index
NewNodeQuantity=3;//quantity of new nodes
[go]=NL_G_SplitEdge(TopologyGraph,EdgeIndex,NewNodeQuantity);//application of NL_G_SplitEdge
WindowIndex=3;//window index
VisualizeGraph1=NL_G_ShowGraphNE(go,WindowIndex)

//6. insert new edges
NewEdgeHeadVector=[9 7 8 1];//head vector of new edges
NewEdgeTailVector=[5 3 3 6];//tail vector of new edges
NewEdgeNameVector=['e5' 'e6' 'e7' 'e8'];//name
```

```
    vector of new edges
[TopologyGraph]  = NL_G_AddEdges(go,
    NewEdgeHeadVector,NewEdgeTailVector,
    NewEdgeNameVector);
WindowIndex=4;
VisualizeGraph1=NL_G_ShowGraphNE(TopologyGraph,
    WindowIndex);

// 7. Display the number of nodes and edges again
    after adding the 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
```

Experiment: 3

Creation of the Star topology

Scilab code Solution 3.1 Star Topology Creation

```
//Experiment No.3
// This file must be used under the terms of the
   CeCILL.
// This source file is licensed as described in the
   file COPYING, which
// you should have received as part of this
   distribution.  The terms
// are also available at
// http://www.cecill.info/licences/Licence_CeCILL_V2
  -en.txt
//This Source file 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
//The Operating System used for writing the code
```

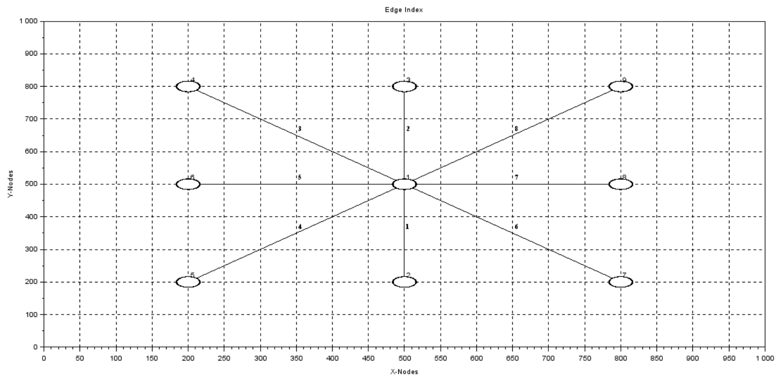

Figure 3.1: Star Topology Creation

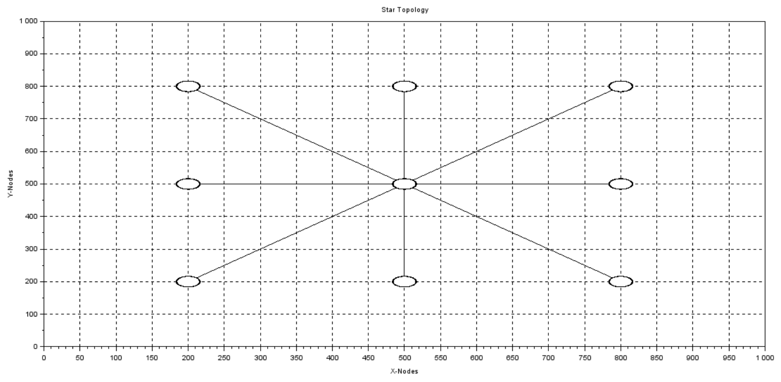

Figure 3.2: Star Topology Creation

```
    found in this file is Windows 8
//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

clear;
clc;

//1. Create and Visualize the Star Topology
NameOfNetwork='Star Topology';// Name of your
    network
NumberOfNodes=9;//Number of Nodes in the network
EndingNodesOfEdge=[1 1 1 1 1 1 1 1];//Ending Nodes
    of the connection lines
StartNodesOfEdge=[2 3 4 5 6 7 8 9];//Starting Nodes
    of the connection lines
XCoordinatesOfNodes=[500 500 500 200 200 200 800 800
     800];// X-Coordinates of the nodes
YCoordinatesOfNodes=[500 200 800 800 200 500 200 500
     800]; // Y-Coordinates of the nodes
[TopologyGraph]=NL_G_MakeGraph(NameOfNetwork,
    NumberOfNodes,EndingNodesOfEdge,StartNodesOfEdge,
    XCoordinatesOfNodes,YCoordinatesOfNodes)//Creates
     the Star topoplogy
WindowNumber=1;//Graph Window Number
[GraphVisualize] = NL_G_ShowGraph(TopologyGraph,
    WindowNumber); // Visualize the Graph
xtitle("Star Topology","X-Nodes","Y-Nodes");
// 2.Colour the nodes and visualize
NodeColor=30; // Node Colour 2:[Blue],3:[Green], 5:[
    Red]
BorderThickness=10; // Node Border thickness
NodeDiameter=25; //Node diameter
WindowNumber=2;//window index

[OutputGraph]=NL_G_GraphEdgesLength(TopologyGraph);
```

```

disp(OutputGraph);

disp('Edge Length',OutputGraph.edge_length);

ListOfNodes=[1 3 5 7];//list of nodes
[NodeHighlight,f2]=NL_G_HighlightNodes(TopologyGraph
    ,ListOfNodes,NodeColor,BorderThickness,
    NodeDiameter,WindowNumber);// Highlight the
    specified nodes
xtitle("Node Higlight","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

//4. Display the number of nodes and edges
[ExtractNode,ExtractEdge]=NL_G_GraphSize(
    TopologyGraph);//Extract the number of nodes and
    edges
disp('Number of nodes:',ExtractNode); //display the
    number of nodes and edges
disp('Number of edges:',ExtractEdge);
//5.Display the Edge Index between two nodes
StartingNode=4;//starting node
EndingNode=1;//ending node
xtitle("Edge Index","X-Nodes","Y-Nodes");
[EdgeIndex]=NL_G_Nodes2Edge(TopologyGraph,
    StartingNode,EndingNode);//Get the Edge Index
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

```
//Experiment No.4
// This file must be used under the terms of the
   CeCILL.
// This source file is licensed as described in the
   file COPYING, which
// you should have received as part of this
   distribution.  The terms
// are also available at
// http://www.cecill.info/licences/Licence_CeCILL_V2
  -en.txt
//This Source file 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
//The Operating System used for writing the code
```

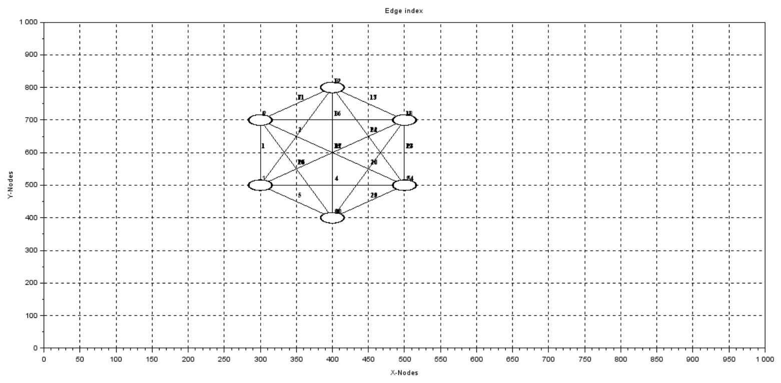

Figure 4.1: Mesh Topology Creation

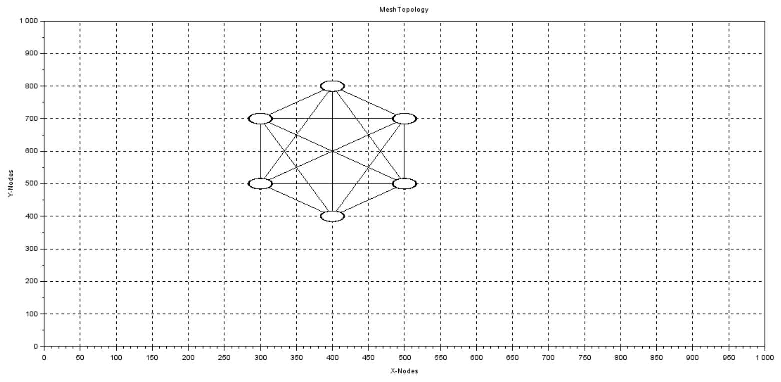

Figure 4.2: Mesh Topology Creation

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

clear;
clc;

//1. Create and visualize a graph with Mesh Topology
NameOfNetwork='Mesh Topology';// Name of your
    network
NumberOfNodes=6;// Number of Nodes in the network
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
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
XCoordinatesOfNodes=[300 300 400 500 500 400]; // X-
    Coordinates of the nodes
YCoordinatesOfNodes=[500 700 800 700 500 400]; // Y-
    Coordinates of the nodes
[TopologyGraph]=NL_G_MakeGraph(NameOfNetwork,
    NumberOfNodes,EndingNodesOfEdge,StartNodesOfEdge,
    XCoordinatesOfNodes,YCoordinatesOfNodes)//Creates
     the Mesh topoplogy
WindowNumber=1;//Graph Window Number
[GraphVisualize] = NL_G_ShowGraph(TopologyGraph,
    WindowNumber); // Visualize the Graph
xtitle("MeshTopology","X-Nodes","Y-Nodes");
//2. Colour the nodes
```

```
NodeColor=30; // Node Colour 2:[Blue],3:[Green], 5:[
    Red]
BorderThickness=10; // Node Border thickness
NodeDiameter=25; //Node diameter
WindowNumber=2;//window index
ListOfNodes=[1 3 5];//list of nodes

[NodeHighlight,Graph1]=NL_G_HighlightNodes(
    TopologyGraph,ListOfNodes,NodeColor,
    BorderThickness,NodeDiameter,WindowNumber);//
    Highlight the specified nodes and output a graph

//3. Visualize the Graph with node number and edge
    number
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
xtitle("Edge index","X-Nodes","Y-Nodes");
//4. Display the number of nodes and edges in scilab
     prompt
[ExtractNode,ExtractEdge]=NL_G_GraphSize(
    TopologyGraph);//Extract the number of nodes and
    edges
disp(ExtractNode,ExtractEdge); //display the number
    of nodes and edges

//5. Display the length of every edge in Scilab
    prompt
[ExtractEdge]=NL_G_EdgesLength(TopologyGraph.node_x,
    TopologyGraph.node_y,TopologyGraph.head,
    TopologyGraph.tail);//Display the length of every
     edge
disp('length of all edges',ExtractEdge);

//6. Display the set of edges connected to a node
    inside a graph in Scilab Prompt
```

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

Experiment: 5

Creation of Tree topology

Scilab code Solution 5.1 Treee Topology

```
//Experiment No.5
// This file must be used under the terms of the
   CeCILL.
// This source file is licensed as described in the
   file COPYING, which
// you should have received as part of this
   distribution.  The terms
// are also available at
// http://www.cecill.info/licences/Licence_CeCILL_V2
   -en.txt
//This Source file 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
//The Operating System used for writing the code
```

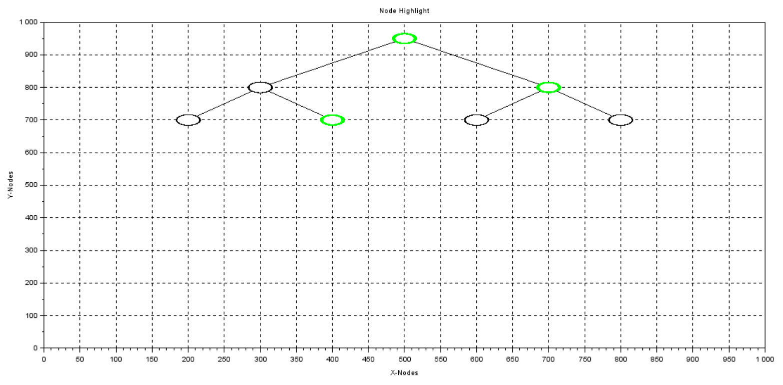

Figure 5.1: Treee Topology

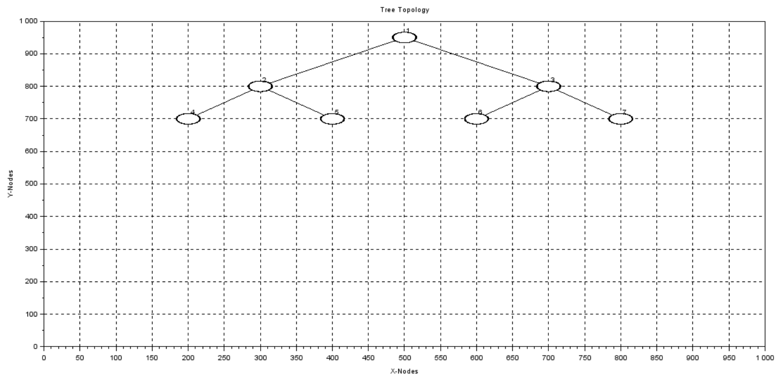

Figure 5.2: Treee Topology

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

clear;
clc;
NameOfNetwork='Tree Topology';// graph name
NumberOfNodes=7;//no. of nodes
//every edge has head and tail.
EndingNodesOfEdge=[1 1 2 2 3 3];// tail of eatch
   edge, this is a node from a edge or link is
   originating
StartNodesOfEdge=[2 3 4 5 6 7];//head of each edge,
   this is a node where edge or link is terminating
XCoordinatesOfNodes=[500 300 700 200 400 600 800];//
    x-coordinate of each node
YCoordinatesOfNodes=[950 800 800 700 700 700 700];//
   y-coordinate of each node
[TopologyGraph]=NL_G_MakeGraph(NameOfNetwork,
   NumberOfNodes,EndingNodesOfEdge,StartNodesOfEdge,
   XCoordinatesOfNodes,YCoordinatesOfNodes)//
   application of NL_G_MakeGraph
WindowNumber=1;//window index
GraphVisualize=NL_G_ShowGraphN(TopologyGraph,
   WindowNumber);//graph visualization
xtitle("Tree Topology","X-Nodes","Y-Nodes");
NodeColor=3; // Node Colour 2:[Blue],3:[Green], 5:[
   Red]
BorderThickness=10; // Node Border thickness
NodeDiameter=25; //Node diameter
WindowNumber=2;//window index
NumberOfNodes=[1 3 5];//list of nodes

[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

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

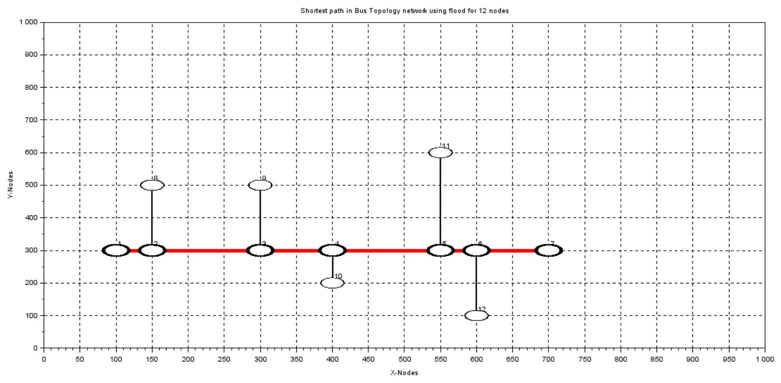

Figure 6.1: Shortest Path in Bus Topology

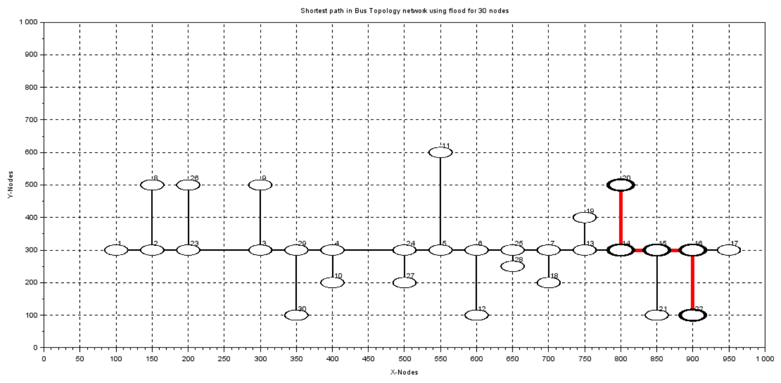

Figure 6.2: Shortest Path in Bus Topology

```
Topology Creation
//The Operating System used for writing the code found in this file is Windows 8
//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
//This is the scilab code to find the Shortest path from source to destination in bus topology network using flood for 12 nodes.
clear;
clc;
NameOfNetwork='Shortest path from s—>d in bus topology with 12 nodes using flood';// graph name
NumberOfNodes=12;//no. of nodes
//every edge has head and tail.
EndingNodesOfConnection=[1 2 3 4 5 6 2 3 4 5 6];// Ending Node of the connection
StartNodesOfConnection=[2 3 4 5 6 7 8 9 10 11 12];// Starting Nodes of the connection lines
// 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
XCoordinatesOfNodes=[100 150 300 400 550 600 700 150 300 400 550 600];// x—coordinate of each node
YCoordinatesOfNodes=[300 300 300 300 300 300 300 500 500 200 600 100];//y—coordinate of each node
[TopologyGraph]=NL_G_MakeGraph(NameOfNetwork,NumberOfNodes,EndingNodesOfConnection,StartNodesOfConnection,XCoordinatesOfNodes,YCoordinatesOfNodes)//Creates the Bus topoplogy
NetworkSize=length(TopologyGraph.node_x);//Real network size
```

```
NeworkLinkQuantity=length(TopologyGraph.head);//
    Quantity of network links
[ne,nr]=NL_F_RandIntNiNj(NetworkSize)//Random
    Selection of two distinct nodes
TimeToLive=15;//Time–To–Live update
[path]=NL_R_Flood(TopologyGraph,ne,nr,TimeToLive)//
    Application of NL_R_Flood
ShortestPathLinks=NL_G_Nodes2Path(path,TopologyGraph
    );//Links of the shortest path
EdgeColor=ones(1,NeworkLinkQuantity);//Display the
    path between i and j: edge color
EdgeBorder=1.5*ones(1,NeworkLinkQuantity);//Edge
    width
EdgeColor(ShortestPathLinks)=5;//Define path color
EdgeBorder(ShortestPathLinks)=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
TopologyGraph.edge_width=EdgeBorder;//Edge width
windownumber=1;//window index
GraphVisualize=NL_G_ShowGraphN(TopologyGraph,
    windownumber);//graph visualization
xtitle("Shortest path in Bus Topology network using
    flood for 12 nodes","X–Nodes","Y–Nodes");

//This is scilab code to find the Shortest path from
     source to destination in bus topology network
    using flood for 30 nodes.
NameofNetwork='Shortest path from s—>d in bus
    topology with 30 nodes using flood';// graph name
NumberOfNodes=30;//no. of nodes
//every edge has StartingNodesOfConnection and tail.
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
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
// 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
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
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
[TopologyGraph]=NL_G_MakeGraph(NameofNetwork,
    NumberOfNodes,EndingNodesOfConnection,
    StartingNodesOfConnection,XCoordinatesOfNodes,
    YCoordinatesOfNodes)//application of
    NL_G_MakeGraph
NetworkSize=length(TopologyGraph.node_x);//real
    network size
NeworkLinkQuantity=length(TopologyGraph.head);//
    quantity of network links
[Node1,Node2]=NL_F_RandIntNiNj(NetworkSize)//
    selection of two distinct nodes
TTL=15;//Time-To-Live update
[path]=NL_R_Flood(TopologyGraph,Node1,Node2,TTL)//
    application of NL_R_Flood
ShortestPathLinks=NL_G_Nodes2Path(path,TopologyGraph
    );//links of the shortest path
EdgeColor=ones(1,NeworkLinkQuantity);//display the
    path between i and j: edge color
EdgeBorder=1.5*ones(1,NeworkLinkQuantity);//edge
    width
```

```
EdgeColor(ShortestPathLinks)=5;//define path color
EdgeBorder(ShortestPathLinks)=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
TopologyGraph.edge_width=EdgeBorder;//edge width
WindowIndex=2;//window index
GraphVisualize=NL_G_ShowGraphN(TopologyGraph,
   WindowIndex);//graph visualization
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

```
//Experiment No.7
// This file must be used under the terms of the CeCILL.
// This source file is licensed as described in the file COPYING, which
// you should have received as part of this distribution.  The terms
// are also available at
// http://www.cecill.info/licences/Licence_CeCILL_V2-en.txt
//This Source file 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
//The Operating System used for writing the code found in this file is Windows 8
//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
//This is the scilab code to find the Shortest path
   from source to destination in ring topology
   network using flood for 10 nodes.
clear;
clc;

NameOfNetwork=' shortest path from source to
   destinetion in ring topology with 10 nodes using
   flood';// graph name
NumberOfNodes=10;//no. of nodes
//every edge has head and tail.
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
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
// 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
XCoordinatesOfNodes=[200 400 100 350 200 500 550 600
    500 550];// x-coordinate of each node
YCoordinatesOfNodes=[100 100 400 650 650 200 500 400
    600 300];//y-coordinate of each node
[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
TopologyGraph.edge_width=EdgeBorder;//edge width
WinodeNumber=1;//window index
GraphVisualize=NL_G_ShowGraphN(TopologyGraph,
   WinodeNumber);//graph visualization
xtitle("Shortest path in Ring Topology network using
    flood for 12 nodes","X-Nodes","Y-Nodes");
//This is the scilab code to find the Shortest path
   from source to destination in ring topology
   network using flood for 30 nodes.
NameOfNetwork=' shortest path from source to
   destinetion in ring topology with 30 nodes using
   flood';// graph name
NumberOfNodes=30;//no. of nodes
//every edge has head and tail.
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
```

```
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
// 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
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
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
[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
TTL=15;//Time-To-Live update
[path]=NL_R_Flood(TopologyGraph,Node1,Node2,TTL)//
   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
```

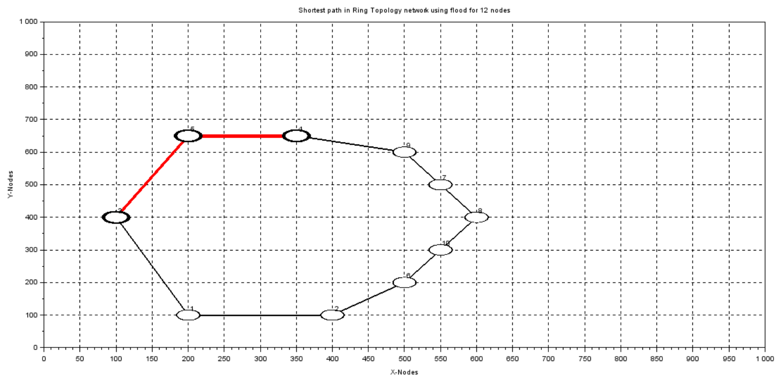

Figure 7.1: Shortest Path in Ring Topology

```
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
TopologyGraph.edge_width=EdgeBorder;//edge width
WindowIndex=2;//window index
GraphVisualize=NL_G_ShowGraphN(TopologyGraph, WindowIndex);//graph visualization
xtitle("Shortest path in Ring Topology network using flood for 30 nodes","X–Nodes","Y–Nodes");
```

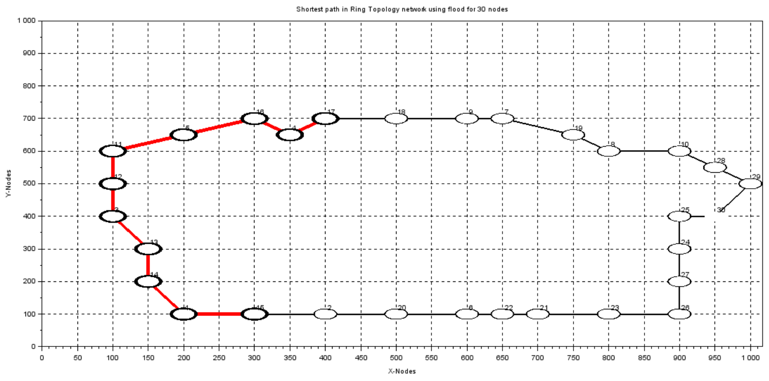

Figure 7.2: Shortest Path in Ring Topology

Experiment: 8

Finding the Shortest Path in Star topology

Scilab code Solution 8.1 Shortest Path in Star Topology

```
//Experiment No.8
// This file must be used under the terms of the
   CeCILL.
// This source file is licensed as described in the
   file COPYING, which
// you should have received as part of this
   distribution. The terms
// are also available at
// http://www.cecill.info/licences/Licence_CeCILL_V2
   -en.txt
//This Source file 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
//The Operating System used for writing the code
```

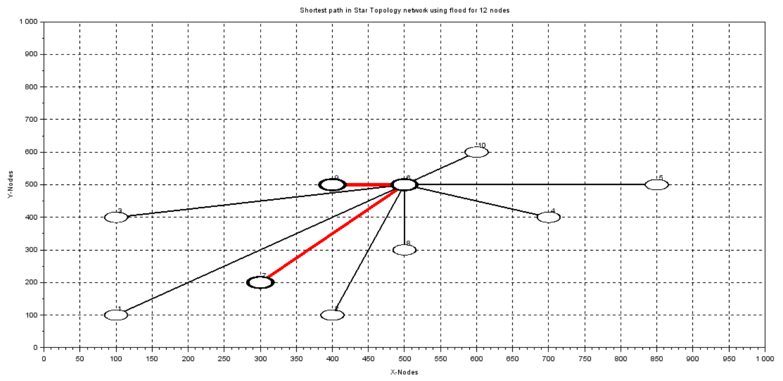
Shortest path in Star Topology network using flood for 12 nodes
Y-Nodes
X-Nodes

Figure 8.1: Shortest Path in Star Topology

```
    found in this file is Windows 8
//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
//This is the scilab code to find the Shortest path from source to destination in Star topology network using flood for 10 nodes.
clear;
clc;

NameOfNetwork='star network with 10 node';// graph name
NumberOfNodes=10;//no of nodes
//every edge has head and tail.
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
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
```

```
TopologyGraph.edge_width=EdgeBorder;//edge width
WindowNumber=1;//window index
GraphVisualize=NL_G_ShowGraphN(TopologyGraph,
   WindowNumber);//graph visualization

xtitle("Shortest path in Star Topology network using
    flood for 12 nodes","X-Nodes","Y-Nodes");

//This is the scilab code to find the Shortest path
   from source to destination in Star topology
   network using flood for 30 nodes.

NameOfNetwork='star network with 30 node';// graph
   name
NumberOfNodes=30;//no of nodes
//every edge has head and tail.
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 6];// tail of eatch
   edge, this is a node from a edge or link is
   originating
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
// 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
    400 600 300 200 500 400 700 800 900 850 950 450
   550 650 350 250 150 725 825 925 300 650];// x-
   coordinate of each node
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
//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
TTL=15;//Time-To-Live update
[path]=NL_R_Flood(TopologyGraph,Node1,Node2,TTL)//
    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
TopologyGraph.edge_width=EdgeBorder;//edge width
WindowNumber=2;//window index
GraphVisualize=NL_G_ShowGraphN(TopologyGraph,
    WindowNumber);//graph visualization
xtitle("Shortest path in star Topology network using
     flood for 30 nodes","X-Nodes","Y-Nodes");
```

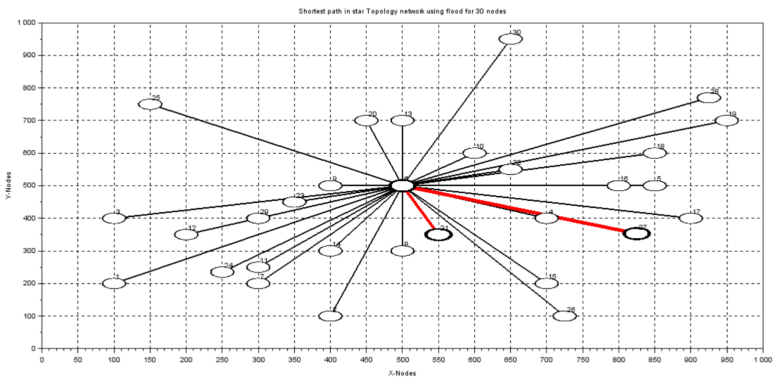

Figure 8.2: Shortest Path in Star Topology

Experiment: 9

Finding the Shortest Path in Mesh topology

Scilab code Solution 9.1 Shortest Path in Mesh Topology

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

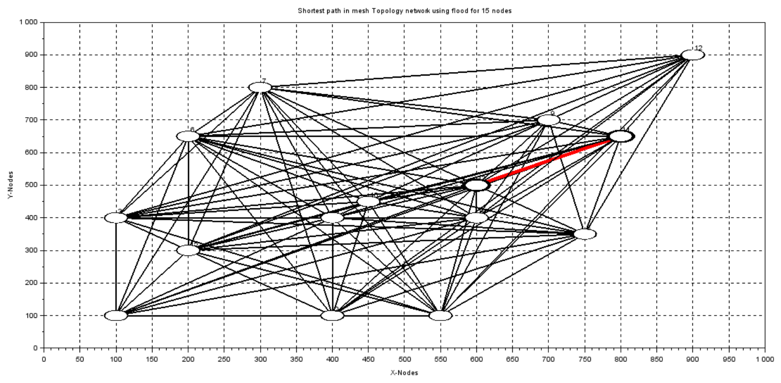

Figure 9.1: Shortest Path in Mesh Topology

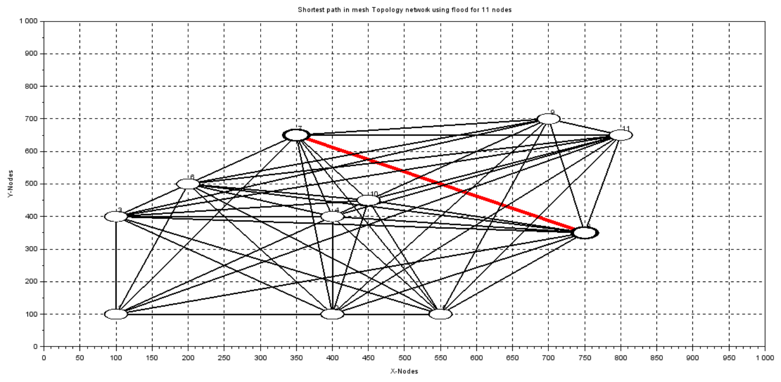

Figure 9.2: Shortest Path in Mesh Topology

```
   Topology Creation
//The Operating System used for writing the code
   found in this file is Windows 8
//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
//This is the scilab code to find the Shortest path
   from source to destination in Mesh topology
   network using flood for 11 nodes.
clear;
clc;

NameOfNetwork='mesh topology with 11 nodes';// graph
    name
NumberOfNodes=11;//no. of nodes
//every edge has head and tail.
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
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
// 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
XCoordinatesOfNodes=[100 400 100 400 550 200 350 750
    700 450 800];// x-coordinate of each node
YCoordinatesOfNodes=[100 100 400 400 100 500 650 350
```

```
    700 450 650];//y-coordinate of each node
[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=5;//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 color
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
TopologyGraph.edge_width=EdgeBorder;//edge width
WindowNumber=1;//window index
GraphVisualize=NL_G_ShowGraphN(TopologyGraph,
   WindowNumber);//graph visualization
xtitle("Shortest path in mesh Topology network using
    flood for 11 nodes","X-Nodes","Y-Nodes");
//This is the scilab code to find the Shortest path
   from source to destination in Mesh topology
   network using flood for 15 nodes.
NameOfNetwork='mesh topology with 15 nodes';// graph
```

```
   name
NumberOfNodes=15;//no. of nodes
//every edge has head and tail.
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 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
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
// 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
XCoordinatesOfNodes=[100 400 100 400 550 200 300 750
    700 450 800 900 600 600 200]; // x-coordinate of
    each node
YCoordinatesOfNodes=[100 100 400 400 100 650 800 350
    700 450 650 900 500 400 300];//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=5;//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 color
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
TopologyGraph.edge_width=EdgeBorder;//edge width
WindowNumber=2;//window index
GraphVisualize=NL_G_ShowGraphN(TopologyGraph,WindowNumber);//graph visualization
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

```
//Experiment No.10
// This file must be used under the terms of the
   CeCILL.
// This source file is licensed as described in the
   file COPYING, which
// you should have received as part of this
   distribution.  The terms
// are also available at
// http://www.cecill.info/licences/Licence_CeCILL_V2
  -en.txt
//This Source file 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
//The Operating System used for writing the code
```

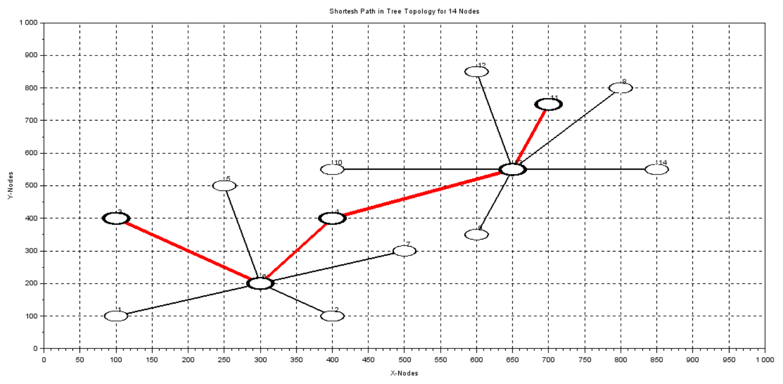

Figure 10.1: Shortest Path in Tree Topology

```
found in this file is Windows 8
//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
//This is the scilab code to find the Shortest path from source to destination in Tree topology network using flood for 14 nodes.
clear;
clc;

NameOfNetwork='shortest path from source to destinetion in star topology with 14 nodes using flood';// graph name
NumberOfNodes=14;//graph parameters
//every edge has head and tail.
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
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
// 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=[100 400 100 400 250 300 500 800
    600 400 700 600 650 850];// x-coordinate of each
    node
YCoordinatesOfNodes=[100 100 400 400 500 200 300 800
    350 550 750 850 550 550];//y-coordinate of each
   node
[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
TTL=15;//Time-To-Live update
[path]=NL_R_Flood(TopologyGraph,Node1,Node2,TTL)//
   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
TopologyGraph.edge_width=EdgeBorder;//edge width
WindowIndex=1;//window index
GraphVisualize=NL_G_ShowGraphN(TopologyGraph,
   WindowIndex);//graph visualization
xtitle("Shortesh Path in Tree Topology for 14 Nodes"
   ,"X—Nodes","Y—Nodes")

//This is the scilab code to find the Shortest path
   from source to destination in Tree topology
   network using flood for 14 nodes.
NameOfNetwork='shortest path from source to
   destinetion in tree topology with 30 nodes using
   flood';// graph name
NumberOfNodes=30;//no. of nodes
//every edge has head and tail.
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
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
// 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=[100 400 100 400 250 300 500 800
    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
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
```

```
[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
TTL=15;//Time-To-Live update
[path]=NL_R_Flood(TopologyGraph,Node1,Node2,TTL)//
   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
TopologyGraph.edge_width=EdgeBorder;//edge width
WindowNumber=2;//window index
GraphVisualize=NL_G_ShowGraphN(TopologyGraph,
   WindowNumber);//graph visualization

xtitle("Shortesh Path in Tree Topology for 30 Nodes"
   ,"X-Nodes","Y-Nodes")
```

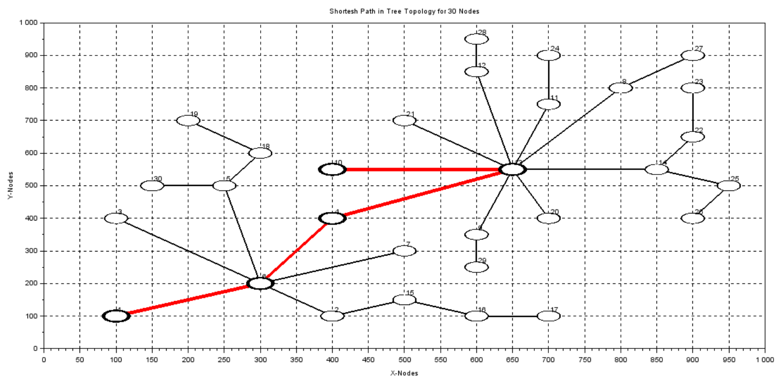

Figure 10.2: Shortest Path in Tree Topology

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

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

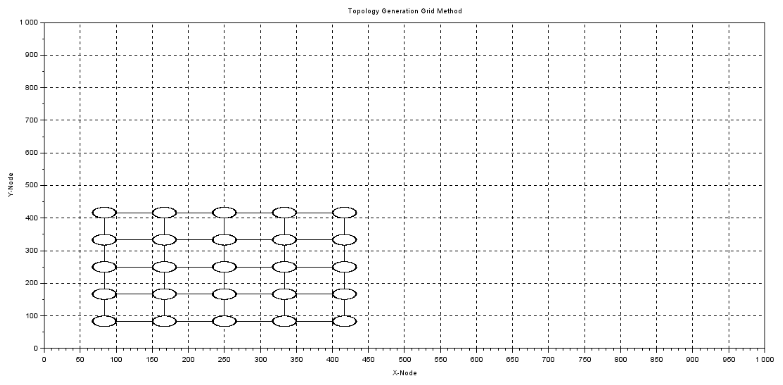

Figure 11.1: Creation of Network Square Area and Coloured Network Topology

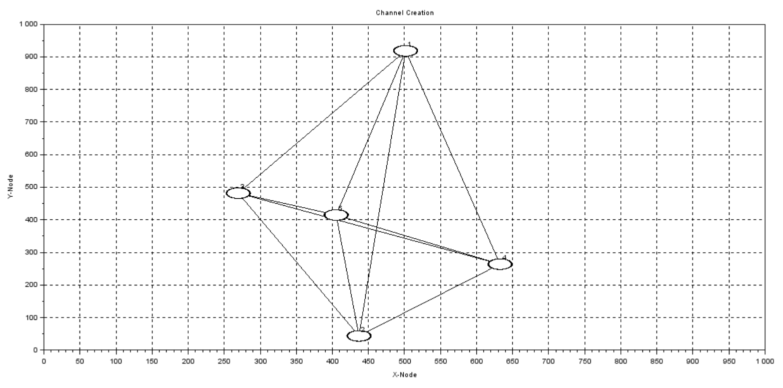

Figure 11.2: Creation of Network Square Area and Coloured Network Topology

```
//School of Computing Science and Engineering, VIT
    University Chennai
//using the NARVAL examples of Scilab for Network
    Topology Creation
//The Operating System used for writing the code
    found in this file is Windows 8
//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

//1.Topology creation and colouring using grid
    method 2. Topology creation and colouring using
    random method 3. Topology Creation using detailed
     method


//1.Topology creation and colouring using grid
    method
clear;
clf;

NumberOfRows=10;// number of rows
NumberOfColumns=10;//number of columns
XCoordinatesOfArea=1000;//Area of network x
    coordinates
YCoordinatesOfArea=1000;//Area of network y
    coordinates
radius = 100;
colour = 2;
[TopologyGraph]=NL_T_Grid(NumberOfRows,
    NumberOfColumns,XCoordinatesOfArea,
    YCoordinatesOfArea);// generation of a topology
    in respect with the grid method
WindowIndex=1;//window index
```

```
NameOfNetwork = "Network Created with Grid and
   Showgraph method";
InitialSquareArea=NL_M_Background(WindowIndex,
   NameOfNetwork);//initial square area

xlabel("x coordinates of node", "fontsize", 2)
ylabel("Y coordinates of node");

InitialSquareArea=NL_M_GraphDisplayUpdate(
   TopologyGraph,WindowIndex,radius,colour);

//2. Topology creation and colouring using random
   method
NetworkSize=5;//network size
NetworkAreaSide=1000;//network squared area side
LocalityRadius=1000;//Locality radius
[TopologyGraph]=NL_T_LocalityConnex(NetworkSize,
   NetworkAreaSide,LocalityRadius);//generation of a
    topology
WindowIndex=2;//window index
InitialSquareArea=NL_G_ShowGraphN(TopologyGraph,
   WindowIndex);//graph visualization
xtitle("Channel Creation","X-Node","Y-Node");
[go,n,e]=NL_G_WCDS(TopologyGraph)
ListOfChannels=[2];//list of chanels
[goc]=NL_G_WCDSChannel(go,NetworkSize,e,
   ListOfChannels);//application of NL_G_WCDSChannel
WindowIndex=3;//window index
InitialSquareArea=NL_G_ShowGraph(goc,WindowIndex);//
   graph visualization
xtitle("Channel Creation","X-Node","Y-Node");
disp(go,NetworkSize,e,ListOfChannels);


//3. Topology Creation using detailed method
NameOfNetwork='Detailed topology';// Name of your
   network
NetworkSize=8;//Number of Nodes in the network
```

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

```
    application of NL_G_ShowGraph
xtitle("Nodes mentioned in node vector","X-Node","Y-
    Node");

NumberOfLines=5;//number of lines
NumberOfColumns=5;//number of columns
XCoordinatesOfArea=500;//network area x-side
YCoordinatesOfArea=500;//network area x-side
[TopologyGraph]=NL_T_Grid(NumberOfLines,
    NumberOfColumns,XCoordinatesOfArea,
    YCoordinatesOfArea);//generation of a topology in
     respect with the grid method
WindowIndex=7;//window index
[f]=NL_G_ShowGraph(TopologyGraph,WindowIndex);//
    application of NL_G_ShowGraph
xtitle("Topology Generation Grid Method","X-Node","Y
    -Node");
//4. Display the number of nodes and edges
[ExtractNode,ExtractEdge]=NL_G_GraphSize(
    TopologyGraph);//Extract the number of nodes and
    edges
disp('Number of nodes:',ExtractNode); //display the
    number of nodes and edges
disp('Number of edges:',ExtractEdge);

//2. Colour the nodes and Edges
NodeColor=30; // Node Colour 2:[Blue],3:[Green], 5:[
    Red]
BorderThickness=10; // Node Border thickness
NodeDiameter=25; //Node diameter
WindowIndex=8;//window index
nodes=[1 2 4 6 8 10 12 14];//list of nodes
[GraphHighlight,NodeVector]=NL_G_HighlightNodes(
    TopologyGraph,nodes,NodeColor,BorderThickness,
    NodeDiameter,WindowIndex);//Highlight the
    specific nodes mentioned in the 'nodes' vector
xtitle("Nodes mentioned in node vector","X-Node","Y-
    Node");
```

```
EdgeColor=5;// Edge Colour
EdgeWidth=5;//Edge Width
WIndowIndex=9;//window index
edges=[1 5 7];//list of edges
[GraphHighlight,NodeVector]=NL_G_HighlightEdges(
   GraphHighlight,edges,EdgeColor,EdgeWidth,
   WIndowIndex);//Highlight the specific nodes
   mentioned in the 'edges' vector
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

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

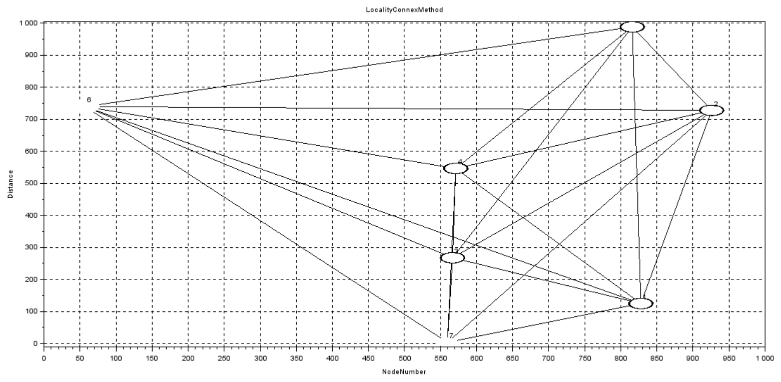

Figure 12.1: Shortest path using Dijikstras Algorithm

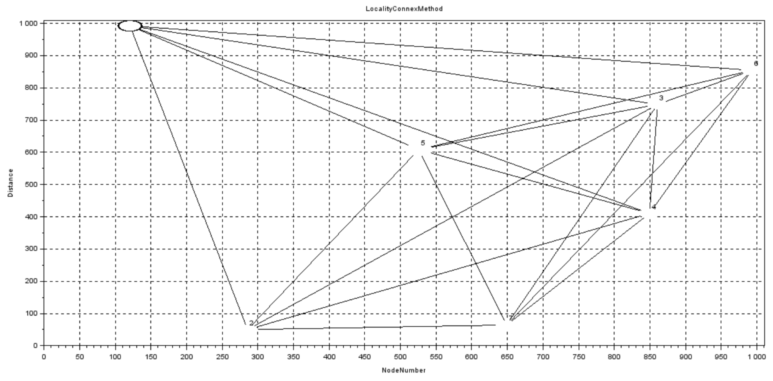

Figure 12.2: Shortest path using Dijikstras Algorithm

```
    Topology Creation
//The Operating System used for writing the code
    found in this file is Windows 8
//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

clear;
clc;

NetworkSize=7;//network size
NeworkSquareArea=1000;//network square area side
LocalityRadius=1000;//locality radius
[TopologyGraph]=NL_T_LocalityConnex(NetworkSize,
    NeworkSquareArea,LocalityRadius);//generation of
    a random topology in respect with the Locality
    method.
for i=1:1:NetworkSize
    disp(i,"node number :");
    [dist,pred]=NL_R_Dijkstra(TopologyGraph,i);//
        application of NL_R_Dijkstra
    j=dist;
    for x = j
    disp(x);
    plot(j);
    xlabel("Nodenumber", "fontsize", 2)
    ylabel("Distance");
    end
end
for i=1:1:NetworkSize
    [TopologyGraph]=NL_T_LocalityConnex(NetworkSize,
        NeworkSquareArea,LocalityRadius);//generation
         of a random topology in respect with the
        Locality method.
    [dist,pred]=NL_R_Dijkstra(TopologyGraph,i);//
```

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

Experiment: 13

Finding the shortest path by using Prim's Algorithm

Scilab code Solution 13.1 Shortest Path Using Prims Algorithm

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

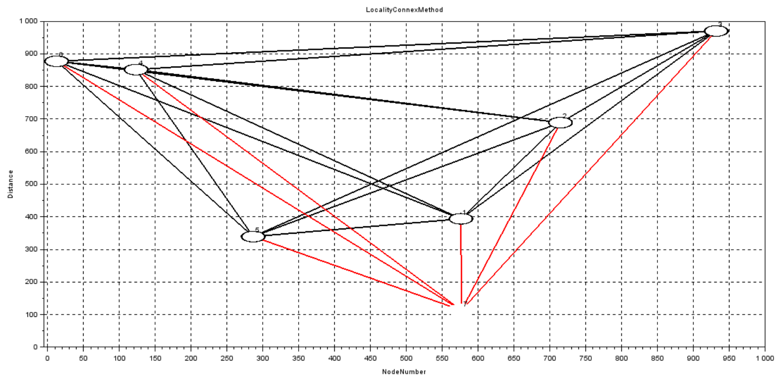

Figure 13.1: Shortest Path Using Prims Algorithm

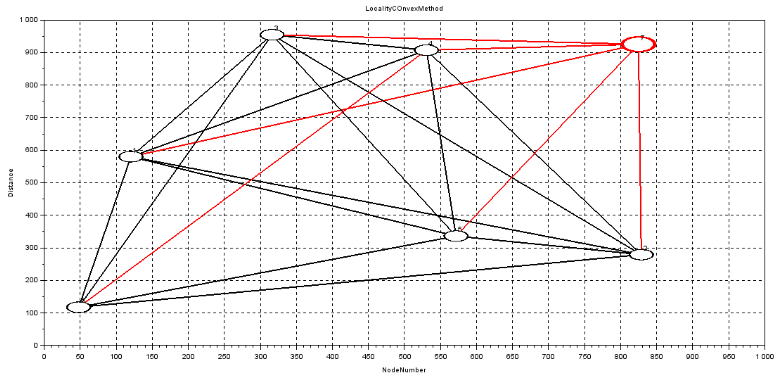

Figure 13.2: Shortest Path Using Prims Algorithm

```
   Topology Creation
//The Operating System used for writing the code
   found in this file is Windows 8
//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

clear;
clc;

NetworkSize=7;//network size
NeworkSquareArea=1000;//network square area side
LocalityRadius=1000;//locality radius
[ToplogyGraph]=NL_T_LocalityConnex(NetworkSize,
   NeworkSquareArea,LocalityRadius);//generation of
   a random topology in respect with the Locality
   method.
for SourceNode=1:1:NetworkSize
    disp(SourceNode,"node number :");
    dw = 2;
    WindowSize = SourceNode;
    [go,v,pred]=NL_R_Prim(ToplogyGraph,SourceNode,dw
       ,WindowSize)//application of NL_R_Prim
//      disp(go);
    disp(v);
    disp(pred);
    j = pred;
    for x = j
    disp(x);
    plot(j);
    xlabel("Nodenumber", "fontsize", 2)
    ylabel("Distance");
    end
end
for SourceNode=1:1:NetworkSize
```

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

Experiment: 14

Manet Simulation

Scilab code Solution 14.1 Manet Simulation

```
//Experiment No.14
// This file must be used under the terms of the
   CeCILL.
// This source file is licensed as described in the
   file COPYING, which
// you should have received as part of this
   distribution.  The terms
// are also available at
// http://www.cecill.info/licences/Licence_CeCILL_V2
   -en.txt
//This Source file 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
//The Operating System used for writing the code
```

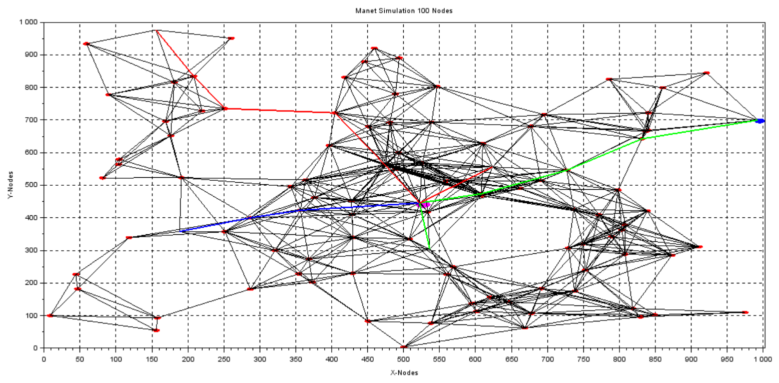

Figure 14.1: Manet Simulation

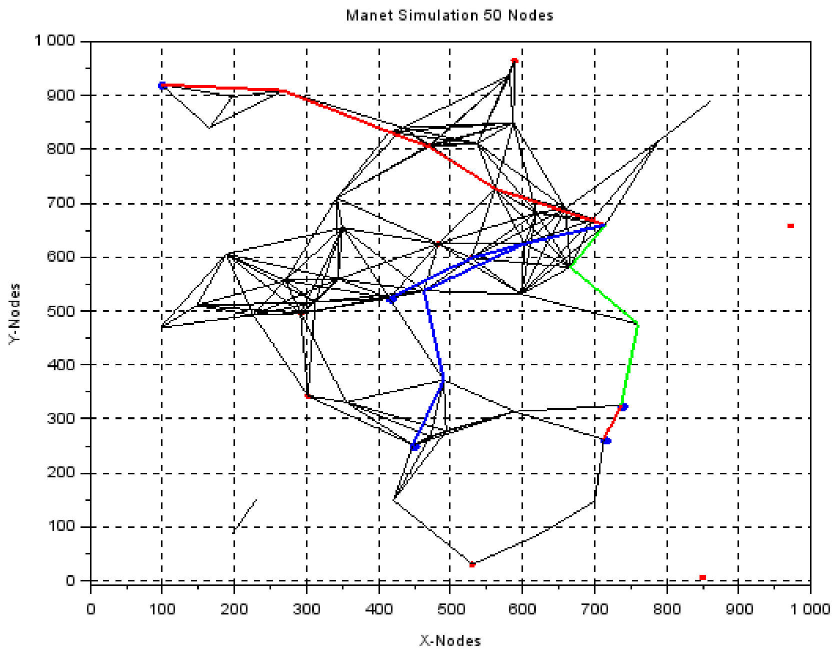

Figure 14.2: Manet Simulation

```
    found in this file is Windows 8
//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

clear;
clc;

RadiusMovingNode=10;//display radius of moving nodes
RadiusFixedNodes=15;//display radius of fixed nodes
MovingNodeConnection=20;//display radius of the
    moving nodes belonging to the connection under
    studies
NodeQuantity=50;//5quantity of moving nodes
FixedNodeQuantity=5;//quantity of fixed nodes
NetworkArea=1000;//network square area side
MaximumSpeed=20;//maximum speed
MinSimuDuration=100;//simulation duration
MaxSimuDuration=100;//maximal waiting time
RadiusForLinks=180;//Locality radius for the links
    attribution
WindowIndex=1;//window index
NL_M_Simulation1N2AllAP(RadiusMovingNode,
    RadiusFixedNodes,MovingNodeConnection,
    NodeQuantity,FixedNodeQuantity,NetworkArea,
    MaximumSpeed,MinSimuDuration,MaxSimuDuration,
    RadiusForLinks,WindowIndex);//application of
    NL_M_Simulation1N2AllAP
xtitle("Manet Simulation 50 Nodes","X-Nodes","Y-
    Nodes")

RadiusMovingNode=10;//display radius of moving nodes
RadiusFixedNodes=15;//display radius of fixed nodes
MovingNodeConnection=20;//display radius of the
    moving nodes belonging to the connection under
    studies
NodeQuantity=100;//quantity of moving nodes
FixedNodeQuantity=5;//quantity of fixed nodes
NetworkArea=1000;//network square area side
MaximumSpeed=50;//maximum speed
MinSimuDuration=25;//simulation duration in secs
MaxSimuDuration=10;//maximal waiting time in secs
RadiusForLinks=180;//Locality radius for the links
    attribution
WindowIndex=2;//window index
NL_M_Simulation1N2AllAP(RadiusMovingNode,
    RadiusFixedNodes,MovingNodeConnection,
    NodeQuantity,FixedNodeQuantity,NetworkArea,
    MaximumSpeed,MinSimuDuration,MaxSimuDuration,
    RadiusForLinks,WindowIndex);//application of
    NL_M_Simulation1N2AllAP
xtitle("Manet Simulation 100 Nodes","X-Nodes","Y-
    Nodes")

RadiusMovingNode=5;//display radius of moving nodes
RadiusFixedNodes=10;//display radius of fixed nodes
MovingNodeConnection=20;//display radius of the
    moving nodes belonging to the connection under
    studies
NodeQuantity=200;//quantity of moving nodes
FixedNodeQuantity=5;//quantity of fixed nodes
NetworkArea=1000;//network square area side
MaximumSpeed=20;//maximum speed
MinSimuDuration=100;//simulation duration
MaxSimuDuration=100;//maximal waiting time
RadiusForLinks=180;//Locality radius for the links
    attribution
WindowIndex=3;//window index
NL_M_Simulation1N2AllAP(RadiusMovingNode,
    RadiusFixedNodes,MovingNodeConnection,
    NodeQuantity,FixedNodeQuantity,NetworkArea,
    MaximumSpeed,MinSimuDuration,MaxSimuDuration,
    RadiusForLinks,WindowIndex);//application of
    NL_M_Simulation1N2AllAP
xtitle("Manet Simulation 200 Nodes","X-Nodes","Y-
    Nodes")
```

Experiment: 15

Congestion Control

Scilab code Solution 15.1 Congestion Control

```
//Experiment No.15
// This file must be used under the terms of the
   CeCILL.
// This source file is licensed as described in the
   file COPYING, which
// you should have received as part of this
   distribution. The terms
// are also available at
// http://www.cecill.info/licences/Licence_CeCILL_V2
   -en.txt
//This Source file 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
//The Operating System used for writing the code
```

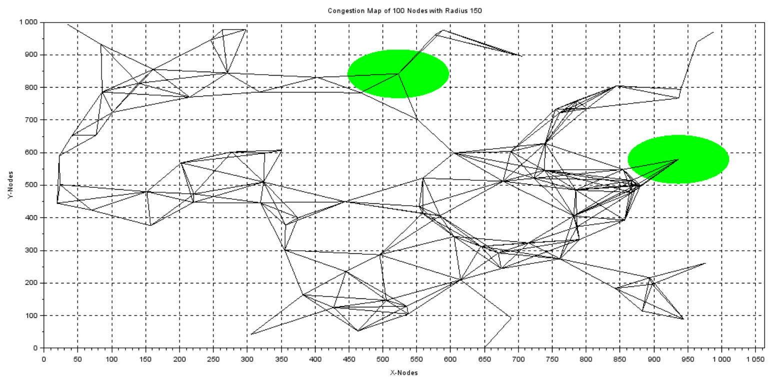

Figure 15.1: Congestion Control

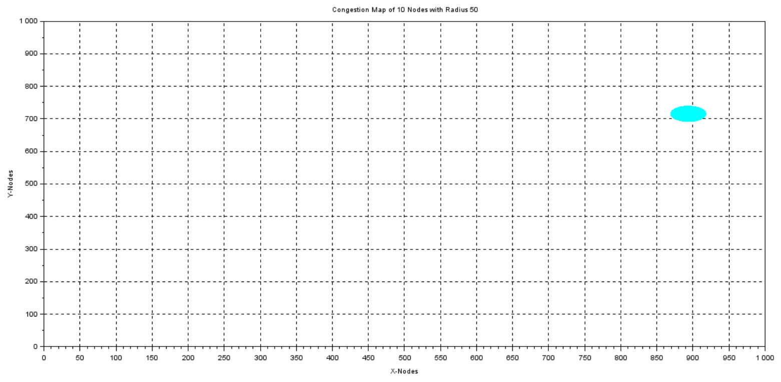

Figure 15.2: Congestion Control

```
    found in this file is Windows 8
//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

clear;
clc;

NetworkSize=100;//network size
NetworkSquareArea=1000;//network squared area side
LocalityRadius=50;//Locality radius
Color=1;//color
[TopologyGraph]=NL_T_LocalityConnex(NetworkSize,
    NetworkSquareArea,LocalityRadius);//generation of
     a topology
WindowIndex=1;//window index
NameOfNetwork="Congestion Map of 100 nodes with r=50
    "; // Name of the graph
InitialSquareArea=NL_M_Background(WindowIndex,
    NameOfNetwork);//initial square area
InitialSquareArea=NL_M_GraphDisplayUpdate(
    TopologyGraph,WindowIndex,LocalityRadius,Color);
//f=NL_G_ShowGraphN(g,w);//graph visualization
n=TopologyGraph.node_number;//graph size
sink=NL_F_RandInt1n(n);//selection of the sink
[cm,np,pred]=NL_R_CongestionSinkFlood(TopologyGraph,
    sink)//application of NL_R_CongestionSinkFlood
xtitle("Congestion Map of 100 Nodes with Radius 50",
     "X-Nodes","Y-Nodes");
NetworkSize=100;//network size
NetworkSquareArea=1000;//network squared area side
LocalityRadius=100;//Locality radius
Color=2;//color
[TopologyGraph]=NL_T_LocalityConnex(NetworkSize,
    NetworkSquareArea,LocalityRadius);//generation of
```

```
 a topology
WindowIndex=2;//window index
NameOfNetwork="Congestion Map of 100 nodes with r
   =100";
InitialSquareArea=NL_M_Background(WindowIndex,
   NameOfNetwork);//initial square area
InitialSquareArea=NL_M_GraphDisplayUpdate(
   TopologyGraph,WindowIndex,LocalityRadius,Color);
//f=NL_G_ShowGraphN(g,w);//graph visualization
n=TopologyGraph.node_number;//graph size
sink=NL_F_RandInt1n(n);//selection of the sink
[cm,np,pred]=NL_R_CongestionSinkFlood(TopologyGraph,
   sink)//application of NL_R_CongestionSinkFlood
xtitle("Congestion Map of 100 Nodes with Radius 100"
   , "X-Nodes","Y-Nodes");
NetworkSize=100;//network size
NetworkSquareArea=1000;//network squared area side
LocalityRadius=150;//Locality radius
Color=3;//color
[TopologyGraph]=NL_T_LocalityConnex(NetworkSize,
   NetworkSquareArea,LocalityRadius);//generation of
    a topology
WindowIndex=3;//window index
NameOfNetwork="Congestion Map of 100 nodes with r
   =150";
InitialSquareArea=NL_M_Background(WindowIndex,
   NameOfNetwork);//initial square area
InitialSquareArea=NL_M_GraphDisplayUpdate(
   TopologyGraph,WindowIndex,LocalityRadius,Color);
//f=NL_G_ShowGraphN(g,w);//graph visualization
n=TopologyGraph.node_number;//graph size
sink=NL_F_RandInt1n(n);//selection of the sink
[cm,np,pred]=NL_R_CongestionSinkFlood(TopologyGraph,
   sink)//application of NL_R_CongestionSinkFlood
xtitle("Congestion Map of 100 Nodes with Radius 150"
   , "X-Nodes","Y-Nodes");
NetworkSize=10;//network size
NetworkSquareArea=1000;//network squared area side
```

```
LocalityRadius=50;//Locality radius
Color=4;//color
[TopologyGraph]=NL_T_LocalityConnex(NetworkSize,
   NetworkSquareArea,LocalityRadius);//generation of
    a topology
WindowIndex=4;//window index
NameOfNetwork="Congestion Map of 10 nodes with r=50"
   ;
f=NL_M_Background(WindowIndex,NameOfNetwork);//
   initial square area
InitialSquareArea=NL_M_GraphDisplayUpdate(
   TopologyGraph,WindowIndex,LocalityRadius,Color);
//f=NL_G_ShowGraphN(g,w);//graph visualization
n=TopologyGraph.node_number;//graph size
sink=NL_F_RandInt1n(n);//selection of the sink
[cm,np,pred]=NL_R_CongestionSinkFlood(TopologyGraph,
   sink)//application of NL_R_CongestionSinkFlood
xtitle("Congestion Map of 10 Nodes with Radius 50",
   "X-Nodes","Y-Nodes");
NetworkSize=10;//network size
NetworkSquareArea=1000;//network squared area side
LocalityRadius=100;//Locality radius
Color=5;//color
[TopologyGraph]=NL_T_LocalityConnex(NetworkSize,
   NetworkSquareArea,LocalityRadius);//generation of
    a topology
WindowIndex=5;//window index
NameOfNetwork="Congestion Map of 10 nodes with r=100
   ";
InitialSquareArea=NL_M_Background(WindowIndex,
   NameOfNetwork);//initial square area
InitialSquareArea=NL_M_GraphDisplayUpdate(
   TopologyGraph,WindowIndex,LocalityRadius,Color);
//f=NL_G_ShowGraphN(g,w);//graph visualization
n=TopologyGraph.node_number;//graph size
sink=NL_F_RandInt1n(n);//selection of the sink
[cm,np,pred]=NL_R_CongestionSinkFlood(TopologyGraph,
   sink)//application of NL_R_CongestionSinkFlood
```

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