Lab 3 Routing Protocols

Overview

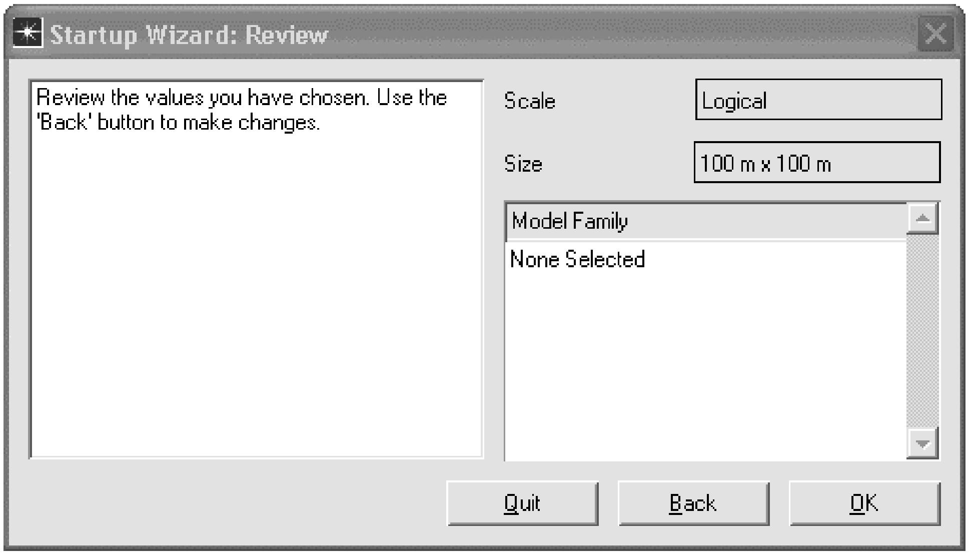
The Routing Information Protocol (RIP) is an example of a dynamic distance vector routing algorithm. This protocol chooses routes within a network that are of minimum distance. Routers adapt to changes in network topology (link or router failures) by exchanging information with their directly connected neighbors. Under normal conditions, routing table information is exchanged periodically (typically every 30 seconds). When changes in topology occur, however, the *triggered update* mechanism comes into play. If a router receives new routing information, it will wait 1-5 seconds (randomly determined) before passing that information on to its neighbors, rather than waiting the full 30 seconds. The triggered update mechanism causes topology information to propagate through the network much faster than it would using the standard timeout.

Objective

To simulate the behavior of several routers running the RIP routing protocol and to learn how to use routing tables to find paths in a network.

Build the Simulation Model

Start up OPNET IT Guru Academic Edition.
Select the **File** menu => **New...**
Choose **Project** and click on **OK**.
Change the **Project Name** to **xx_RIP_Network** (where **xx** are your initials) and click on **OK**.
In the **Initial Topology** window, select **Create Empty Scenario** and click on **Next**.
In the **Choose Network Scale** window, select **Logical** and click on **Next**.
In the **Select Technologies** window, click on **Next**.
In the **Review** window, click on **OK**.



Startup Wizard: Review

Review the values you have chosen. Use the 'Back' button to make changes.

Scale: Logical

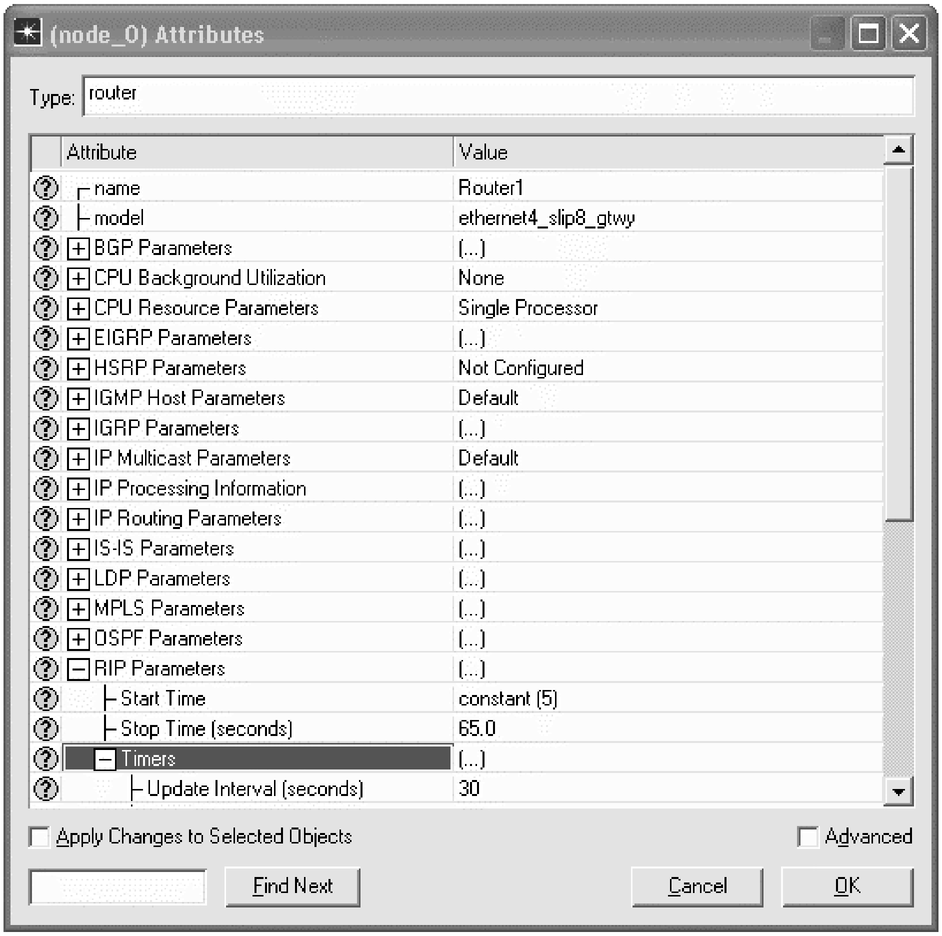
Size: 100 m x 100 m

Model Family
None Selected

First, you may read the model usage guide for the RIP protocol. This guide explains how to configure nodes to use RIP and defines the meaning of each parameter.
Click on the **Protocols** tab => **RIP** => **Model Usage Guide**.
When you have finished reading the guide, click on the close window icon of the PDF viewer.

We will now build a small network of routers and set up the RIP parameters.

Select an **ethernet4_slip8_gtwy** object from the Object Palette and place it in the project workspace. Right click on the object and choose **View Node Description**. Note that this gateway is equipped with four Ethernet interfaces and eight SLIP interfaces. Click on the close window icon to close the window. Right click on the router, choose **Edit Attributes**, and set the **name** attribute to **Router1**. Expand the **RIP Parameters** attribute and the **Timers** attribute. Note that the **Update Interval (seconds)** is set to 30 seconds. This means that a router will exchange its routing table with each of its neighbors every 30 seconds, even if new information is not learned. Click on **OK** to close the window.



(node_0) Attributes

Type: router

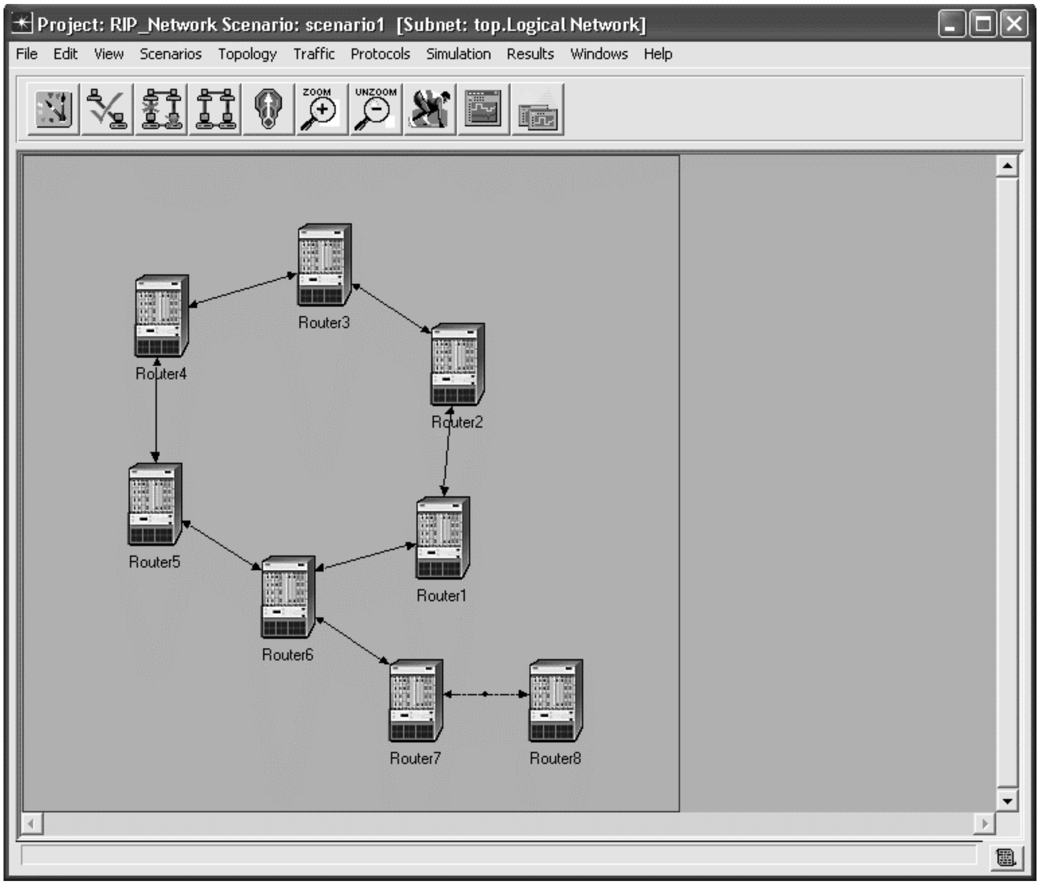
Attribute	Value
name	Router1
model	ethernet4_slip8_gtwy
⊞ BGP Parameters	(...)
⊞ CPU Background Utilization	None
⊞ CPU Resource Parameters	Single Processor
⊞ EIGRP Parameters	(...)
⊞ HSRP Parameters	Not Configured
⊞ IGMP Host Parameters	Default
⊞ IGRP Parameters	(...)
⊞ IP Multicast Parameters	Default
⊞ IP Processing Information	(...)
⊞ IP Routing Parameters	(...)
⊞ IS-IS Parameters	(...)
⊞ LDP Parameters	(...)
⊞ MPLS Parameters	(...)
⊞ OSPF Parameters	(...)
⊟ RIP Parameters	(...)
Start Time	constant (5)
Stop Time (seconds)	65.0
⊟ Timers	(...)
Update Interval (seconds)	30

☐ Apply Changes to Selected Objects

☐ Advanced

Left click on the router and choose **Copy** from the **Edit** menu at the top of the window. Choose **Paste** from the **Edit** menu five times to generate five copies of the first router in the project workspace. Arrange them in a circle and then connect them together into a ring using six **PPP_DS1** lines from the Object Palette. Each router should be connected to two other routers.

Paste two more routers on the map. Using 2 more **PPP_DS1** lines, make a tail off of Router6. Your final network should look like a ring of six routers with a tail of two routers:

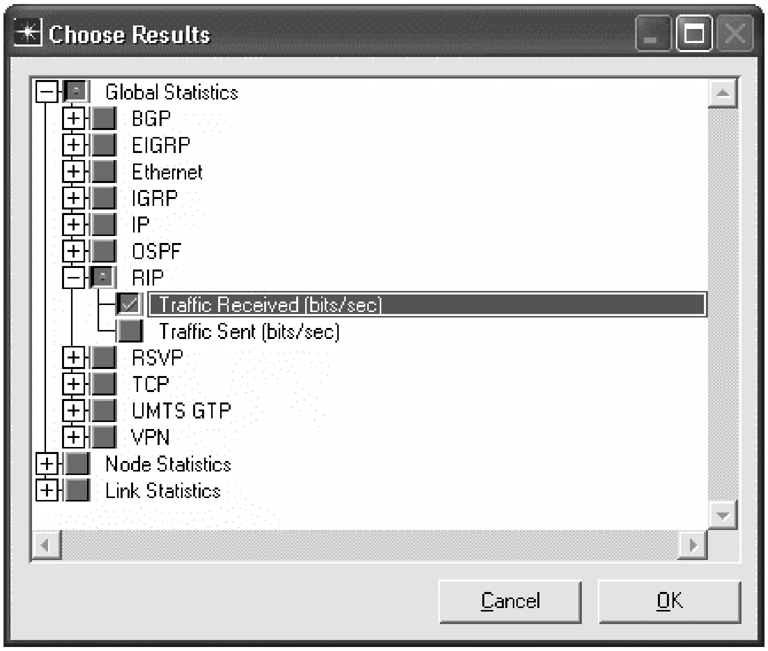

Project: RIP_Network Scenario: scenario1 [Subnet: top.Logical Network]
File Edit View Scenarios Topology Traffic Protocols Simulation Results Windows Help
ZOOM
UNZOOM
Router3
Router4
Router2
Router5
Router1
Router6
Router7
Router8

Select the **Protocols** menu => **IP** => **Routing** => **Configure Routing Protocols…** Check the boxes next to **RIP**, **Apply the above selection to subinterfaces**, and **Visualize Routing Domains.** Select the radio button next to **All interfaces (including loopback).** Click on **O K** to close the window. By doing so, you have ensured that the RIP routing protocol will be used to route packets across all the interfaces on all the routers. Note that the legend shows that RIP is used on all links.

Now we are going to configure one of the PPP links to fail part of the way through the simulation. Select **utilities** from the pull-down menu in the Object Palette in order to display all utility objects. Place a **Failure Recovery** object in the project workspace. Right click on the failure object and **Edit Attributes**. Set the **name** to **Link Failure**. Expand the **Link Failure/Recovery Specification** attribute and set **rows** to **1**. Expand **row 0**, and set the **Name** to **Logical Network.Router 1<->Router 6**. Set the **Time** to **300**. Note that the **Status** is set to **Fail**. This failure node will cause the link between router 1 and router 6 to fail 300 seconds into the simulation. Click on **OK** to close the window.

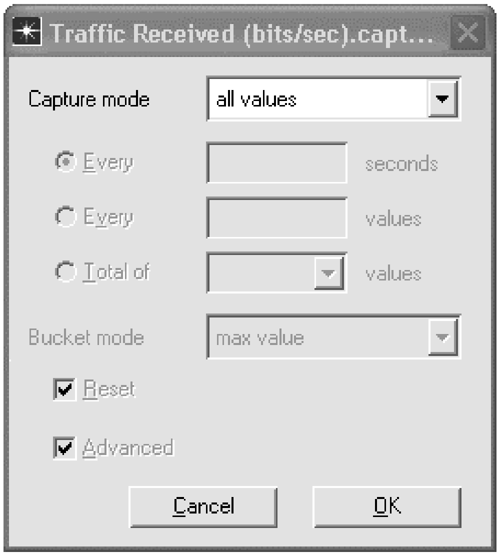
Configure and Run the Simulation

Select the **Simulation** tab => **Choose Individual Statistics…**
Expand the **Global Statistics** item, and the **RIP** item and select the **Traffic Received (bits/sec)** statistic. Right click on the statistic and choose **Change Collection Mode**. Check the box next to **Advanced**, and change the **Capture Mode** to **all values**. This will ensure that OPNET will provide a more detailed graph.
Click on **OK** twice to close the windows.



Choose Results

- Global Statistics
  - BGP
  - EIGRP
  - Ethernet
  - IGRP
  - IP
  - OSPF
  - RIP
    - ☑ Traffic Received (bits/sec)
    - Traffic Sent (bits/sec)
  - RSVP
  - TCP
  - UMTS GTP
  - VPN
- Node Statistics
- Link Statistics



Traffic Received (bits/sec).capt...

Capture mode: all values

● Every ___ seconds

○ Every ___ values

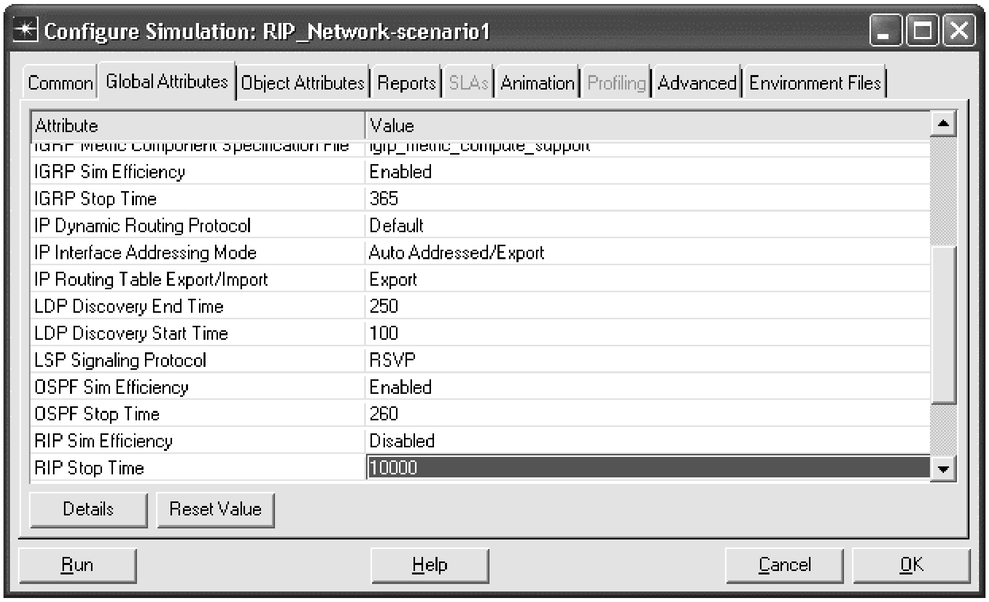
○ Total of ___ values

Bucket mode: max value

☑ Reset

☑ Advanced

Select **Simulation => Configure Discrete Event Simulation…**
Under the **Common** tab, set the **Duration** to **10**, and the unit to **minute(s)**.
Click on the **Global Attributes** tab. Set the **IP Interface Addressing Mode** to **Auto Address/Export**. Set the **IP Routing Table Export/Import** field to **Export**. This will cause the routing tables of all the routers to be written to a text file when the simulation completes. Set the **RIP Sim Efficiency** to **disabled**. Set the **RIP Stop Time** to **10000**. This will ensure that the routers keep exchanging routing information even when their routing tables have stabilized.
Click on **Run** to run the simulation.
When the simulation has completed, click on **Close** to close the window.



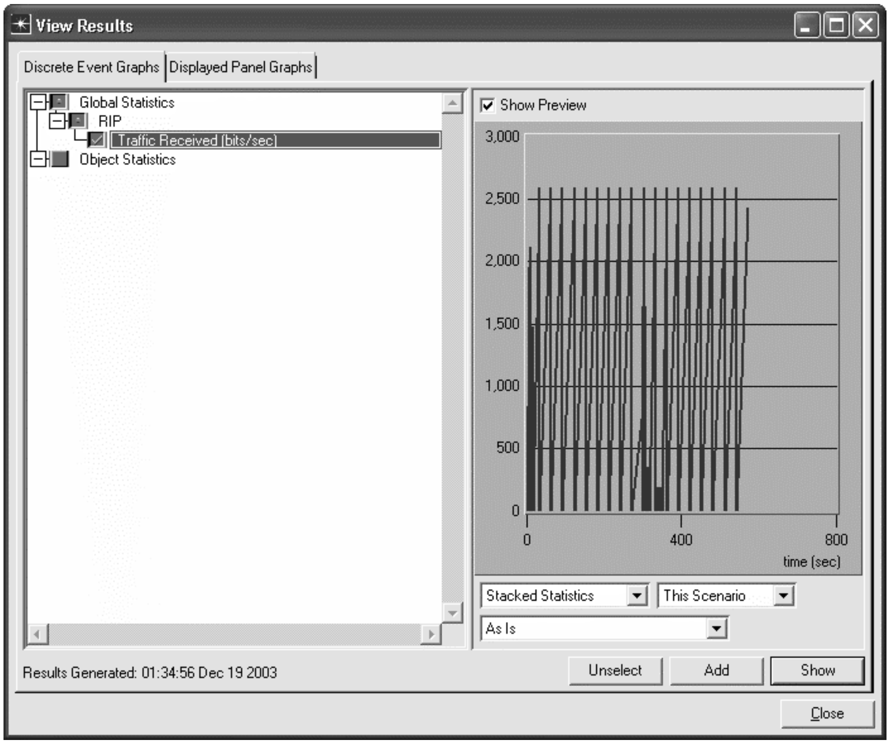
Configure Simulation: RIP_Network-scenario1

Attribute	Value
IGRP Metric Component Specification File	igrp_metric_compute_support
IGRP Sim Efficiency	Enabled
IGRP Stop Time	365
IP Dynamic Routing Protocol	Default
IP Interface Addressing Mode	Auto Addressed/Export
IP Routing Table Export/Import	Export
LDP Discovery End Time	250
LDP Discovery Start Time	100
LSP Signaling Protocol	RSVP
OSPF Sim Efficiency	Enabled
OSPF Stop Time	260
RIP Sim Efficiency	Disabled
RIP Stop Time	10000

Inspect and Analyze Results

Select **Results** => **View Results…**

Select and expand the **Global Statistics** item and the **RIP** item, and select the **Traffic Received (bits/sec)** statistic. This statistic shows how much traffic was generated by the routing algorithm. Click on **Show** to generate a separate window for the statistics graph. You should see nice, regular peaks in routing protocol traffic every 30 seconds except for two periods, one at the beginning of the simulation and one half way through. Select the first 2 minutes of the graph to zoom in. You should now be able to see that many irregularly timed updates were done very quickly at the beginning of the simulation. These updates are due to the triggered update mechanism in RIP. The other period of irregular updates is due to the link failure that occurred after 300 seconds. Click on the close window icon and **Delete** the panel. Click on **Close** to close the **View Results** window.



View Results

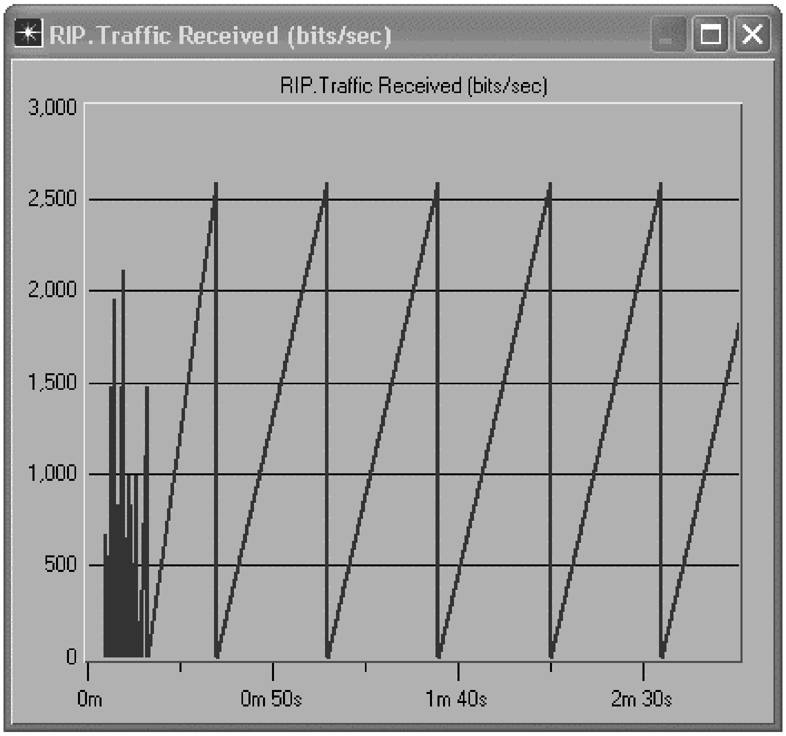
- Global Statistics
  - RIP
    - ☑ Traffic Received (bits/sec)
- Object Statistics

☑ Show Preview

Stacked Statistics | This Scenario

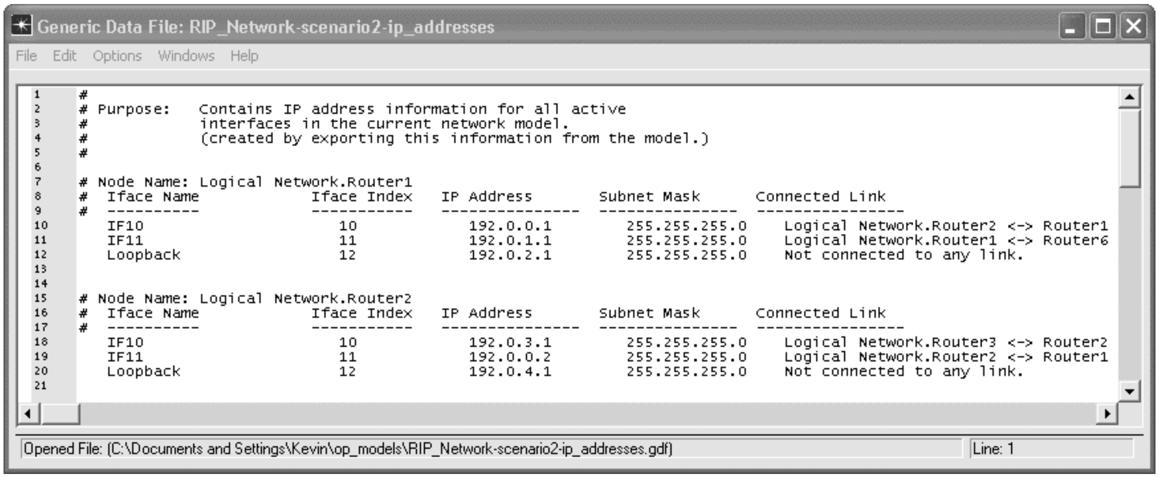
As Is

Results Generated: 01:34:56 Dec 19 2003



Select **Scenarios** => **Duplicate Scenario…** Accept the default scenario name of **scenario2** by clicking on the **OK** button. Left click on the Failure node to select it and choose **Edit** => **Cut** to remove the Failure node. Choose **Simulation** => **Run Discrete Event Simulation** to generate results for this scenario in which no link failures occur. When the simulation has completed, click on **Close** to close the window.

Select **File** => **Model Files** => **Refresh Model Directories.** Select **File** => **Open…** Choose to open a **Generic Data File**, and from the list displayed, choose **xx_RIP_Network-scenario2-ip_addresses**. This file shows the IP addresses and subnetwork masks which have been assigned to each interface by OPNET. Note that each router has one interface called the *loopback* interface that is not attached to any link. These logical interfaces are commonly used for testing or to allow two socket programs to communicate even if they are running on the same machine. Note that the figure below shows only part of the entire file. Click on the close window icon to close the addresses window.

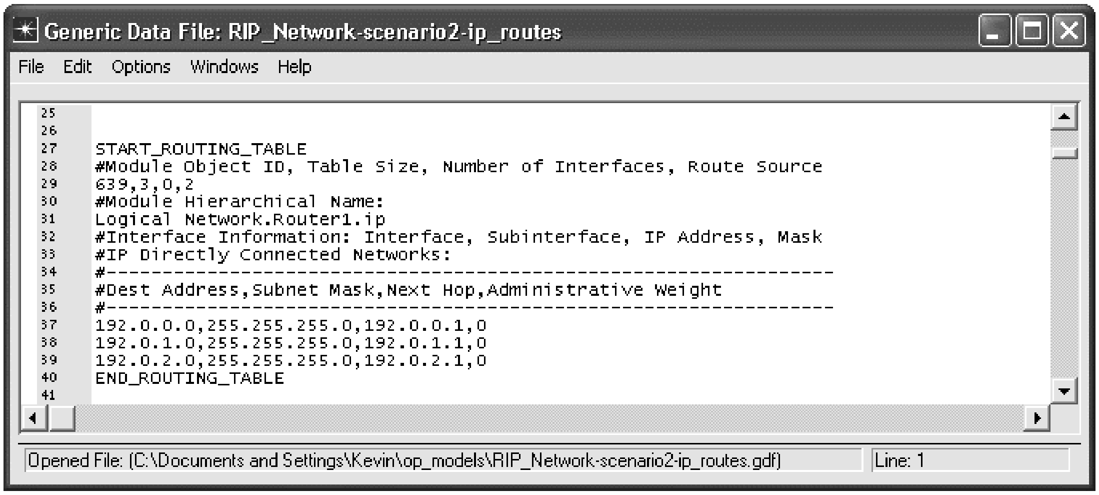


```
#
# Purpose:   Contains IP address information for all active
#            interfaces in the current network model.
#            (created by exporting this information from the model.)
#

# Node Name: Logical Network.Router1
#  Iface Name           Iface Index   IP Address       Subnet Mask      Connected Link
#  ----------           -----------   ---------------  ---------------  ----------------
   IF10                    10            192.0.0.1        255.255.255.0    Logical Network.Router2 <-> Router1
   IF11                    11            192.0.1.1        255.255.255.0    Logical Network.Router1 <-> Router6
   Loopback                12            192.0.2.1        255.255.255.0    Not connected to any link.


# Node Name: Logical Network.Router2
#  Iface Name           Iface Index   IP Address       Subnet Mask      Connected Link
#  ----------           -----------   ---------------  ---------------  ----------------
   IF10                    10            192.0.3.1        255.255.255.0    Logical Network.Router3 <-> Router2
   IF11                    11            192.0.0.2        255.255.255.0    Logical Network.Router2 <-> Router1
   Loopback                12            192.0.4.1        255.255.255.0    Not connected to any link.

```

Opened File: (C:\Documents and Settings\Kevin\op_models\RIP_Network-scenario2-ip_addresses.gdf)

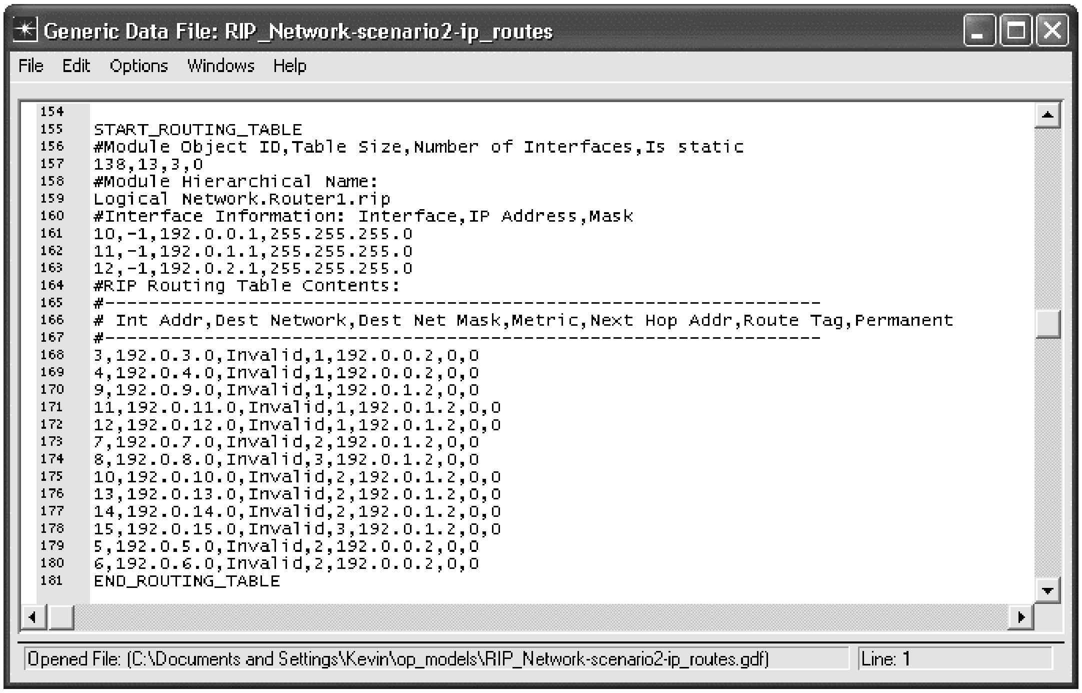
Line: 1

Now we will inspect the routing tables of the nodes. Select **File** => **Open…** Choose to open a **Generic Data File** again, and from the list displayed, choose **xx_RIP_Network-scenario2-ip_routes**. This file is a dump of the routing tables maintained at each router in the state they were in at the end of the simulation run. There are two parts to the file. In the first part, the directly connected networks for each router are listed. A network is directly connected to a router if the router has an interface which matches that network address up to the length of the subnetwork mask. For example, Router1 has an interface of 192.0.2.1 with a subnetwork mask of 255.255.255.0. A subnetwork mask of 255.255.255.0 requires that the first 24 bits of the IP address match the network address. So the 192.0.2.0 network matches the interface address of 192.0.2.1 and Router1 is directly connected to the 192.0.2.0 network.


Generic Data File: RIP_Network-scenario2-ip_routes
File Edit Options Windows Help
Opened File: (C:\Documents and Settings\Kevin\op_models\RIP_Network-scenario2-ip_routes.gdf)
Line: 1

```


START_ROUTING_TABLE
#Module Object ID, Table Size, Number of Interfaces, Route Source
639,3,0,2
#Module Hierarchical Name:
Logical Network.Router1.ip
#Interface Information: Interface, Subinterface, IP Address, Mask
#IP Directly Connected Networks:
#-----------------------------------------------------------------
#Dest Address,Subnet Mask,Next Hop,Administrative Weight
#-----------------------------------------------------------------
192.0.0.0,255.255.255.0,192.0.0.1,0
192.0.1.0,255.255.255.0,192.0.1.1,0
192.0.2.0,255.255.255.0,192.0.2.1,0
END_ROUTING_TABLE

```

The second part of the ip_routes file shows routes to networks which are *not* directly connected. Note that the next figure below shows just the part of the routing table file which applies to Router1. Your file will include one routing table for each router. Note that your address assignments may be different, depending on the order in which you placed the routers in the workspace.


Generic Data File: RIP_Network-scenario2-ip_routes
File Edit Options Windows Help
Opened File: (C:\Documents and Settings\Kevin\op_models\RIP_Network-scenario2-ip_routes.gdf)
Line: 1

```

START_ROUTING_TABLE
#Module Object ID,Table Size,Number of Interfaces,Is static
138,13,3,0
#Module Hierarchical Name:
Logical Network.Router1.rip
#Interface Information: Interface,IP Address,Mask
10,-1,192.0.0.1,255.255.255.0
11,-1,192.0.1.1,255.255.255.0
12,-1,192.0.2.1,255.255.255.0
#RIP Routing Table Contents:
#------------------------------------------------------------------
# Int Addr,Dest Network,Dest Net Mask,Metric,Next Hop Addr,Route Tag,Permanent
#------------------------------------------------------------------
3,192.0.3.0,Invalid,1,192.0.0.2,0,0
4,192.0.4.0,Invalid,1,192.0.0.2,0,0
9,192.0.9.0,Invalid,1,192.0.1.2,0,0
11,192.0.11.0,Invalid,1,192.0.1.2,0,0
12,192.0.12.0,Invalid,1,192.0.1.2,0,0
7,192.0.7.0,Invalid,2,192.0.1.2,0,0
8,192.0.8.0,Invalid,3,192.0.1.2,0,0
10,192.0.10.0,Invalid,2,192.0.1.2,0,0
13,192.0.13.0,Invalid,2,192.0.1.2,0,0
14,192.0.14.0,Invalid,2,192.0.1.2,0,0
15,192.0.15.0,Invalid,3,192.0.1.2,0,0
5,192.0.5.0,Invalid,2,192.0.0.2,0,0
6,192.0.6.0,Invalid,2,192.0.0.2,0,0
END_ROUTING_TABLE
```

Using the **ip_routes** file, we can determine the path that a packet would take through our network. Let's say that we would like to find the path from **Router1** to **Router8**. Using the **xx_RIP_Network-scenario1-ip_addresses** file, let's pick an interface on each router (other than the loopback interfaces).

Router1: 192.0.1.1
Router8: 192.0.13.2

192.0.13.2 is on the 192.0.13.0 network so that is what we'll be looking for in the routing tables. Note that, in order to reduce the size of the routing tables, routing is done based on network addresses rather than individual IP addresses.

Starting with Router1's routing table, we look for the destination network (**Dest Network** field in the table), 192.0.13.0, and pull out the **Next Hop Addr** field. In this case, the **Next Hop Addr** is 192.0.1.2. We now look at the **Interface Information** portion of this file or in the **xx_RIP_Network-scenario1-ip_addresses** file to see which router has an interface with this address. In this case, it is Router6. We then repeat our steps, first looking for the destination network, 192.0.13.0, in Router6's routing table, pulling out the next hop address (192.0.11.2), and finding the router that owns this address (Router7). We continue until we reach a router that is directly connected to the destination network. Router 7 has an interface (192.0.13.1) on the destination network, meaning that it has a direct connection to Router8 (192.0.13.2). You can see that Router7 and Router8 are on the same network because their IP addresses are the same up to the length of the subnetwork mask (24 bits).

In our example, the path is

Router1 (192.0.1.1) => Router6 (192.0.1.2) => Router7 (192.0.11.2) => Router8 (192.0.13.2) Success!

Click on the close window icon to close the **ip_routes** window.

You may also access the routing table dump file directly. Using Windows Explorer, look in your home directory for a directory called **op_models**. Change to this directory and look for the generic data file called **xx_RIP_Network-scenario1-ip_routes.gdf**. This is the file you just inspected from within OPNET. If you have trouble finding the directory, select **Edit** => **Preferences**, and look for the **mod_dirs** attribute. The associated value is the path to your **op_models** directory.

Save your model and close all windows.

Questions

1. Look at the **ip_addresses** and **ip_routes** files associated with the first scenario (scenario1), in which a failure occurred on the line which connects Router1 and Router6. You can switch between scenarios by choosing **Scenarios** => **Switch to Scenario**. Find the path from Router1 to Router8 as shown above.

2. Write a program which takes three parameters, a source IP address, a destination IP address, and the **ip_routes** file from the model you created above. Your program must automate the process of finding the path from the source to the destination. You must print out output as shown above. For example:

RouterA (interface x.x.x.x) => RouterB (interface y.y.y.y) =>… Success/Failure

3. Duplicate scenario1 and add another row to the Failure Recovery object so that the link between Router6 and Router5 fails at time = 300 seconds. Rerun the simulation and examine the **ip_routes** file. Is there a path from Router1 to Router8? If so, what is it? If not, why not?