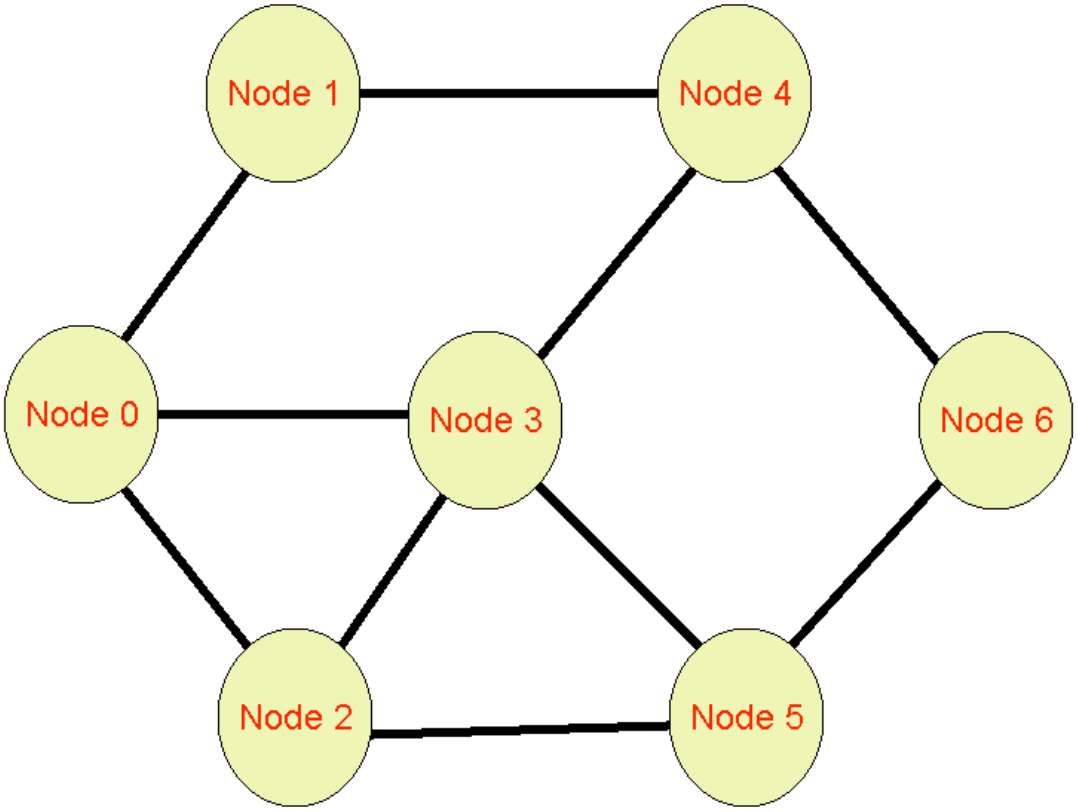
JNW
USERS GUIDE



Version 1.202
of January 22, 2012
FOR
JAVA NETWORK WORKBENCH (JNW)

NETLAB.GMU.EDU
Networking and Simulation Lab George Mason University

Java Network Workbench Version 1.106
Copyright (c) 2007 by J. Mark Pullen
George Mason University, Fairfax, Virginia

Permission to use, copy, modify, and distribute this software and its documentation for academic purposes is hereby granted without fee, provided that the above copyright notice and this permission appear in all copies and in supporting documentation, and that the name of George Mason University not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. GMU makes no representations about the suitability of this software for any purposes. It is provided "AS IS" without express or implied warranties. All risk associated with use of this software is expressly assumed by the user. GMU does not give its permission to sell this software, or include it within commercial products, without separate written agreement.

INTRODUCTION

This guide is provided to help students use JNW, the Java version of Network Workbench. JNW is compatible with Netbeans 6.9.1 using Java 1.5 or later. This guide is oriented toward students using JNW for network simulation. A "Software Developers Guide" provides more information for software developers.

NetBeans ENVIRONMENT INSTALLATION

You can download the NetBeans IDE environment from http://netbeans.org/downloads/. Probably JNW will work with the latest version. However, the latest version tested against is 6.9.1.

INSTALLING JAVA NETWORK WORKBENCH SOURCE CODE IN NetBeans

In NetBeans, create a new project by entering Ctrl+Shift+N. In the New Project window choose "Java Application" or "general application," depending on the NetBeans version. In the next window, change the default Project Name from "JavaApplication"N to "JNWS." In the same window, you can choose the Project Location, where the project will be stored: C:\JNWS is strongly recommended. When the folder C:\JNWS is created, delete the folder C:\JNWS\src\JNWS, where NetBeans stores the source code. Copy the Java Network Workbench code into C:\JNWS\src. You should now be ready to compile and run JNW.

INSTALLING JAVA NETWORK WORKBENCH JAR IN NetBeans

Put the .jar file wherever you want. If you are running the student version of JNW, JNWS, put the .jar file into the folder where you are installing the student exercises folder as described above. In the NetBeans Projects window, right-click on the project JNWS. Choose "Properties" at the bottom of the list. In the Project Properties window choose the Libraries category. Choose the Compile tab and click the Add JAR/Folder button. Navigate to the JNW.jar file and click OK. The JNW.jar file is now in the classpath of the compiler.

Now when you compile, all the classes of JNW are available.

Some students have successfully run JNW in Eclipse or other envrionments. However, this guide provides documentation only for using NetBeans.

JNW DEVELOPMENT STATUS

JNW, the Java version of Network Workbench is in development. Some of the features of the C++ version of Network Workbench are not yet implemented. Presently, a general router network and an Ethernet bus are implemented; a token ring is not. There is no multicasting.

Some of the buttons on the Main GUI are inoperable. They are simply placeholders for later additions.

JAVADOC

JNW has a JavaDoc, which you can create by clicking on "Build" on the tool bar and then "Generate JavaDoc for 'JNWS'." Once generated, the JavaDoc is stored in C:\JNW\dist\javadoc.

Other than the overall description, the JavaDoc is complete, and it defines all class variables down to the private level. This is how you should learn about the classes and methods that are available for your use.

You may choose to display the version number with the JavaDoc. If the version number is not displayed, right-click on the project JNW(S), select Properties at the bottom, select Build Documenting and check the Document Additional Tags: @Author and @Version.

The JavaDoc for JNW has several links to external sites. For example the class ModelNetwork.Routing has fours links to sites that explain Dijkstra's Algorithm and have interactive implementation of the algorithm.

SETTING THE MAIN METHOD

If you run the student version of JNW and get an error message

```
java.lang.NoClassDefFoundError: jnws/Main
Exception in thread "main"
Java Result: 1
```

this means that NetBeans is trying to find the incorrect main() method. There are at least 5 main() methods in JNW, most of which are for testing classes. To correct this

1. Right-click on the JNWS project in the Projects window of NetBeans.

2. Click on the "Properties", the last entry on the list.

3. Click on the "Run" category in the Project Properties window.

4. In the " Main Class:" entry, use the "Browse" button to choose **StartJNW** or another main if you specifically need it.

JNWS will now run.

MAIN GUI

The main GUI,

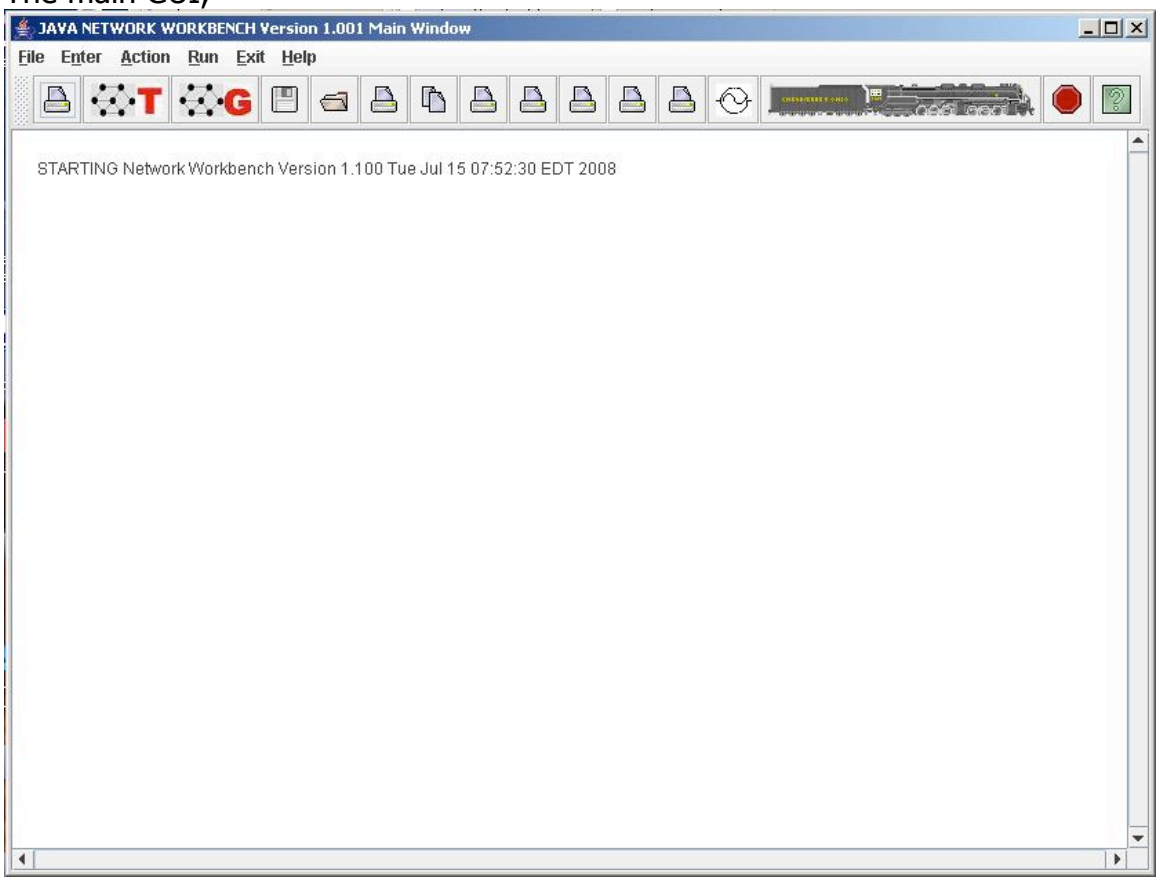


Figure 1 MainGUI

contains the controls to input the data. The output data is written to the center panel of the Main GUI. Please note that the sixth through eleventh buttons on the toolbar are inoperable placeholders for later additions.

NETWORK TYPES

Network Workbench can represent three different network types, a Router Network, an Ethernet bus and a token ring as shown in figure 2 below:

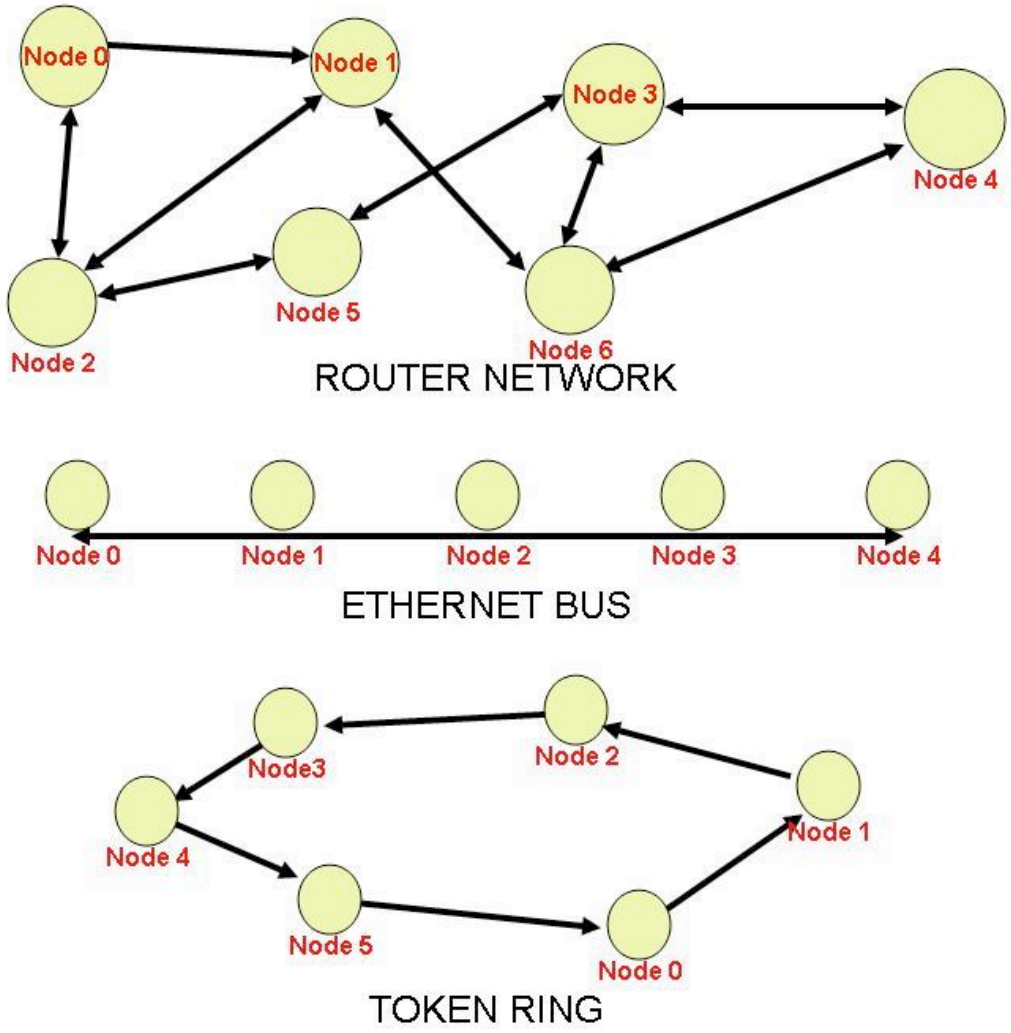


Figure 2 Network Types in Network Workbench

Presently only the Router Network is properly implemented. The other network types will be implemented as time and resources allow.

NODES, LINKS, AND INTERFACES

In JNW, a Node is a computer which can send or receive messages or route messages. It can be the message source, the messages destination, or it may be only an intermediate stop between the source and destination. JNW does not differentiate between a node and a router. In JNW they are identical.

Links are the physical layer of the network model. A link only introduces a delay in frame propagation and int introduces bit errors.

Interfaces operated the data link layer. The interface wraps an input UDP or TCP packets into a frame fro transmission. The interface handles error detection and retransmission.

In JNW, Nodes, Links, and Interfaces are stored in java.util.Vectors. In this way, the user does not need to specify a priori, the number of nodes, etc. All the information is contained in the network topology file. The nodes in the network topology file must be numbered 0 to N-1. If the nodes are not numbered this way, an error message is generated. JNW will automatically correctly number the links and interfaces.

READING NETWORK TOPOLOGY FROM A TEXT FILE

Regardless of the network type, Router Network, Ethernet bus, or token ring, the network topology is specified by a comma-separated text file, which can be stored anywhere. The type of network is specified by a line

NETWORK-TYPE:Router_Network,
NETWORK-TYPE:Token_Ring, or
NETWORK-TYPE:Ethernet_Bus

Each other line of the input represents one link between nodes. The nodes should be numbered 0 to N-1. The input data in order is for each line in the text file is:

1. SOURCE NODE NUMBER *(0 to* N-1*)* ,
2. DESTINATION NODE NUMBER *(0 to* N-1*)* ,
3. CAPACITY *(in kbps),*
4. LINK TYPE *("HDLC", "Ethernet", "Token_Ring"),*
5. BIT ERROR RATE *(for example 0.0000001)* , and
6. TRANSMISSION DELAY *(in microseconds)*

In the example below

```
// JAVA NETWORK WORKBENCH TOPOLOGY INPUT FILE


// FORMAT: SOURCE NODE NUMBER, DESTINATION NODE NUMBER, CAPACITY IN KBPS
// FORMAT: INTERFACE TYPE, BIT ERROR RATE, TRANSMISSION DELAY

// Figure 8.8 of Pullen
// The delay time is 10 times the number in Figure 8.8 of Pullen
// NOTE: The error rates are insignificant, essentially 0.!

// Network type is a router network
   NETWORK-TYPE:Router_Network


// MANDATORY LINK CHARACTERISTICS
// LINK FORMAT: SOURCE NODE NUMBER, DESTINATION NODE NUMBER, CAPACITY IN KBPS,
// INTERFACE TYPE, BIT ERROR RATE, TRANSMISSION DELAY IN MICROSECONDS

//Duplex link between nodes 1 and 2, by the numbering in the book
0,1,10000,Ethernet, 0.000000001 , 60

//Duplex link between nodes 1 and 3, by the numbering in the book
0,2,10000,Ethernet, 0.000000001 , 90

//Duplex link between nodes 1 and 4, by the numbering in the book
0,3,10000,Ethernet, 0.000000001 , 20

//Duplex link between nodes 2 and 5, by the numbering in the book
1,4,10000,Ethernet, 0.000000001 , 30

//Duplex link between nodes 3 and 6, by the numbering in the book
2,5,10000,Ethernet, 0.000000001 , 20

//Duplex link between nodes 4 and 5, by the numbering in the book
3,4,10000,Ethernet, 0.000000001 , 30

//Duplex link between nodes 4 and 6, by the numbering in the book
3,5,10000,Ethernet, 0.000000001 , 40

//Duplex link between nodes 5 and 7, by the numbering in the book
4,6,10000,Ethernet, 0.000000001 , 40

//Duplex link between nodes 6 and 7, by the numbering in the book
5,6,10000,Ethernet, 0.000000001 , 50
```

there are seven nodes connected via 10 megabit per second, Ethernet, point-to-point links, with essentially error-free transmissions. The delay between nodes 0 and 2 is 60 microseconds, 1 and 4 30 microseconds, etc. This represents the router network shown below in Figure 8.8 of the textbook, except that the node numbering in the model begins at 0, not 1 as in the figure.
Blank lines and lines beginning with "//' are ignored.

The order of entry of the links is irrelevant, for example

```
NETWORK-TYPE:Router_Network
4,6,10000,Ethernet, 0.000000001 , 40
5,6,10000,Ethernet, 0.000000001 , 50
```

is the same as

```
5,6,10000,Ethernet, 0.000000001 , 50
NETWORK-TYPE:Router_Network
4,6,10000,Ethernet, 0.000000001 , 40
```

because NetworkWorkbench, after reading the whole network topology, reorders the nodes in numerical order. The position of NETWORK-TYPE in the file is also irrelevant.

Optinally, the use can enter the nods X and Y coordinates. The coordinates are used only the establish a position for graphical display. The input below adds the node positions to the input file above.

```
// JAVA NETWORK WORKBENCH TOPOLOGY INPUT FILE


// FORMAT: SOURCE NODE NUMBER, DESTINATION NODE NUMBER, CAPACITY IN KBPS
// FORMAT: INTERFACE TYPE, BIT ERROR RATE, TRANSMISSION DELAY

// Figure 8.8 of Pullen
// The delay time is 10 times the number in Figure 8.8 of Pullen
// NOTE: The error rates are insignificant, essentially 0.!

// Network type is a router network
   NETWORK-TYPE:Router_Network


// MANDATORY LINK CHARACTERISTICS
// LINK FORMAT: SOURCE NODE NUMBER, DESTINATION NODE NUMBER, CAPACITY IN KBPS,
// INTERFACE TYPE, BIT ERROR RATE, TRANSMISSION DELAY IN MICROSECONDS

//Duplex link between nodes 1 and 2, by the numbering in the book
0,1,10000,Ethernet, 0.000000001 , 60

//Duplex link between nodes 1 and 3, by the numbering in the book
0,2,10000,Ethernet, 0.000000001 , 90

//Duplex link between nodes 1 and 4, by the numbering in the book
0,3,10000,Ethernet, 0.000000001 , 20

//Duplex link between nodes 2 and 5, by the numbering in the book
1,4,10000,Ethernet, 0.000000001 , 30

//Duplex link between nodes 3 and 6, by the numbering in the book
2,5,10000,Ethernet, 0.000000001 , 20

//Duplex link between nodes 4 and 5, by the numbering in the book
3,4,10000,Ethernet, 0.000000001 , 30

//Duplex link between nodes 4 and 6, by the numbering in the book
3,5,10000,Ethernet, 0.000000001 , 40

//Duplex link between nodes 5 and 7, by the numbering in the book
4,6,10000,Ethernet, 0.000000001 , 40

//Duplex link between nodes 6 and 7, by the numbering in the book
5,6,10000,Ethernet, 0.000000001 , 50


// NODE POSITIONS
```

```
// IF NOT READ-IN HERE, THE POSITIONS ARE RANDOMLY ASSIGNED BY THE MODEL

Position of Node 0 = ( 65 , 240 )
Position of Node 1 = ( 200 , 100 )
Position of Node 2 = ( 200 , 420 )
Position of Node 3 = ( 450 , 240 )
Position of Node 4 = ( 600 , 100 )
Position of Node 5 = ( 600 , 420 )
Position of Node 6 = ( 750 , 240 )
```

READING NETWORK TOPOLOGY GRAPHICALLY

Network workbench also has the ability to enter and display topologies graphically. This part of JNW is beta software. Clicking on the button will cause the topology display window to open with the current topology displayed. If no topology has been read into JNW, then the display is blank.

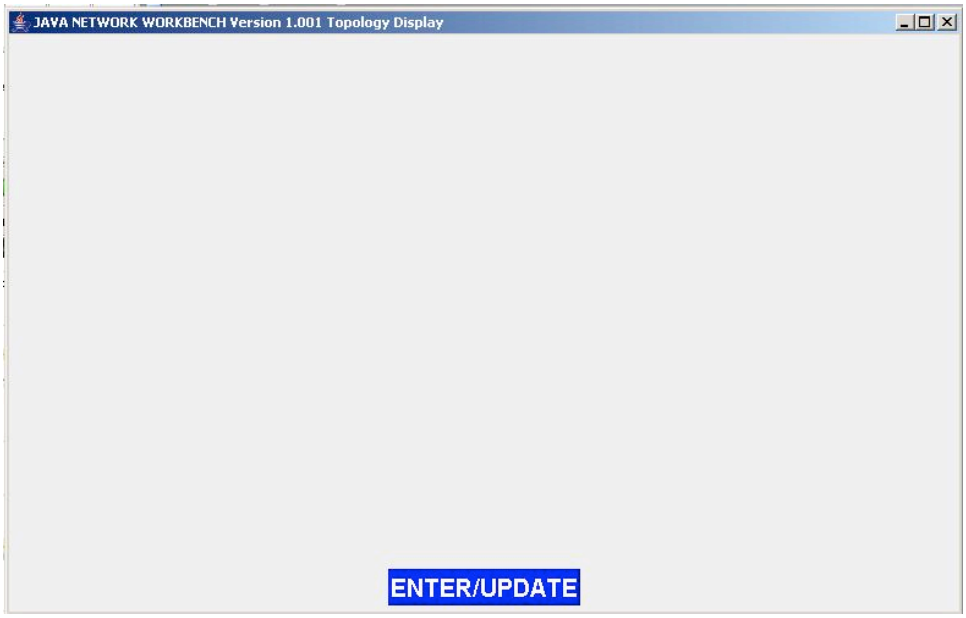


Figure 3 Topology Display Window (with blank topology)

The user can enter Nodes and Links graphically. Clicking on the window will place a Node

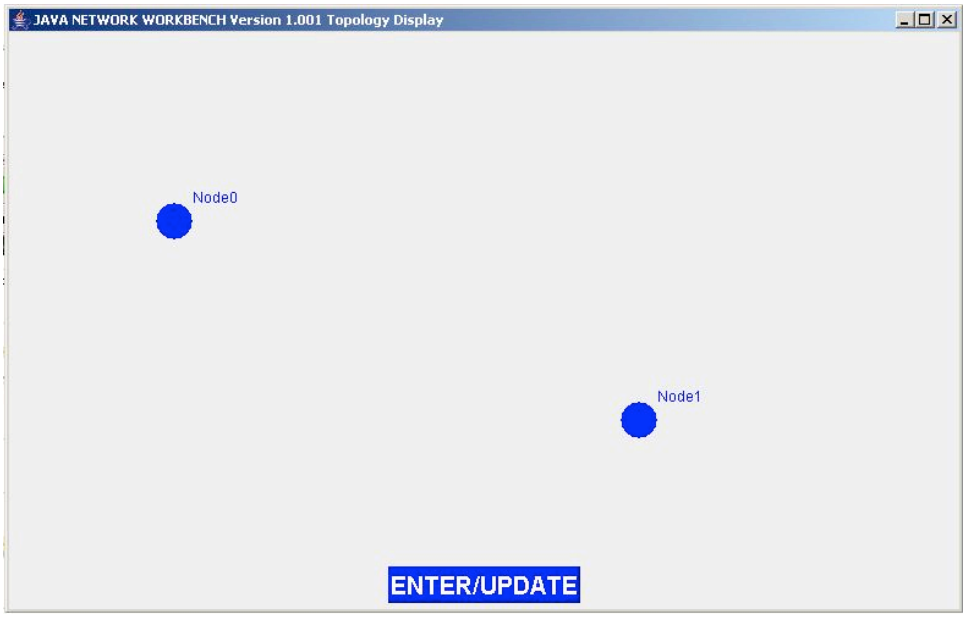


Figure 4 Topology Display Window With Nodes (After Two Separate Clicks)

Links can be added between Nodes by right-clicking on one Node and dragging to another Node.

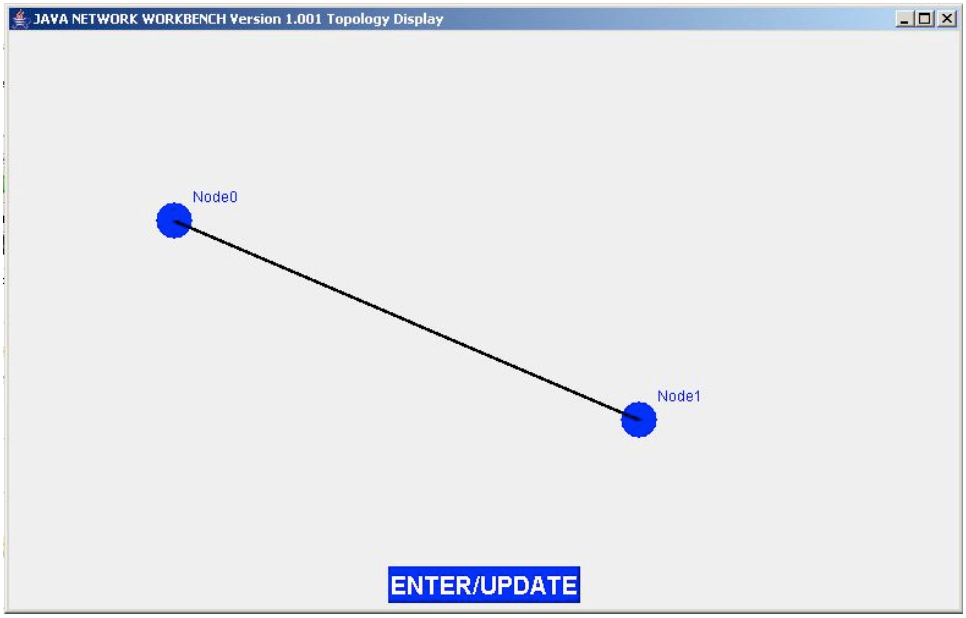


Figure 5 Creating Link By Right-Clicking And Dragging Between Nodes

You can continue to enter a complete network.

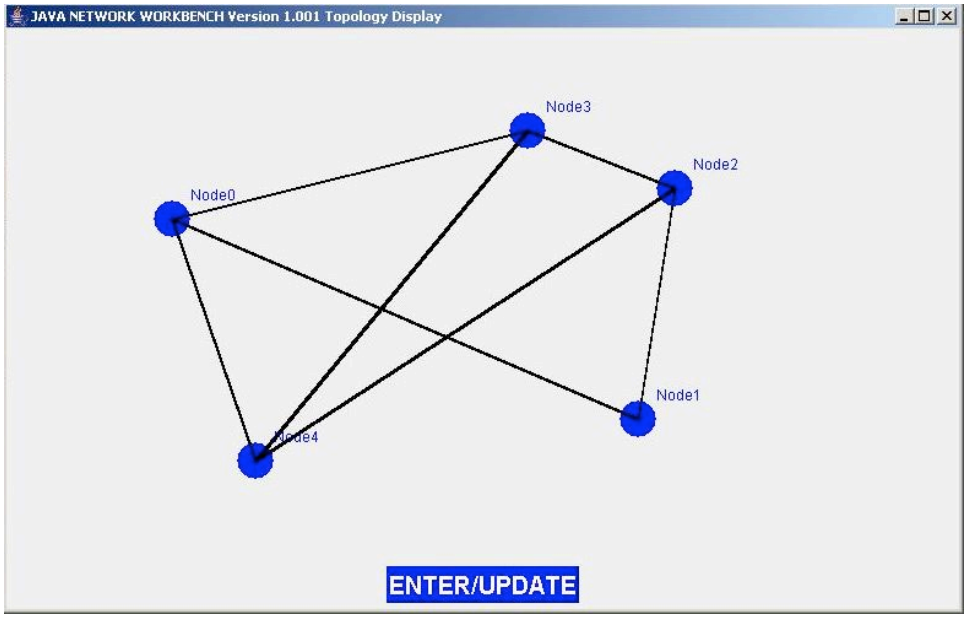


Figure 6 A Complete Network

Nodes can be dragged by clicking and dragging.

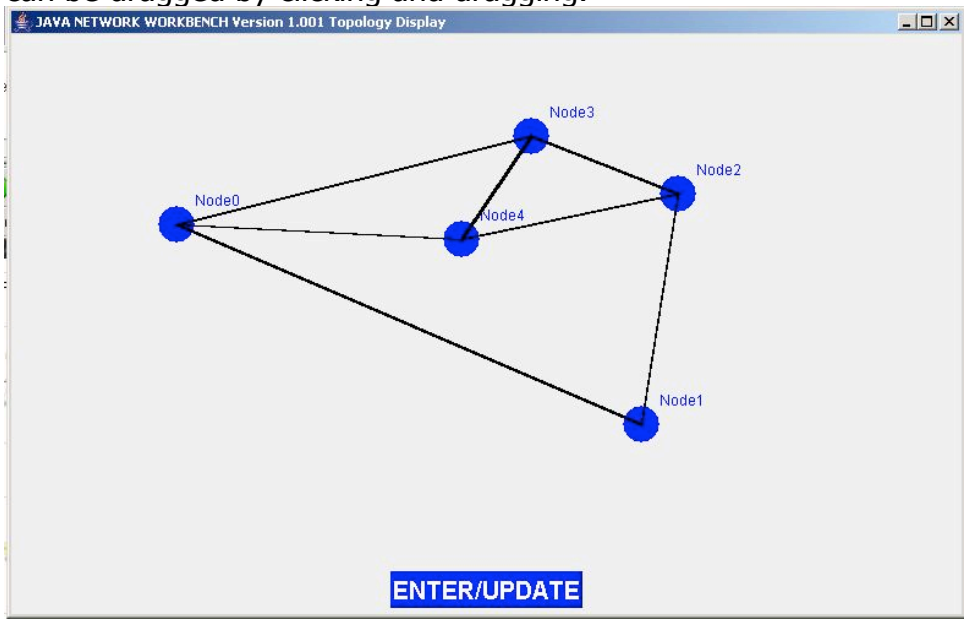


Figure 7 Network After Node 4 Is Dragged

Links can be deleted by double clicking on a Link.

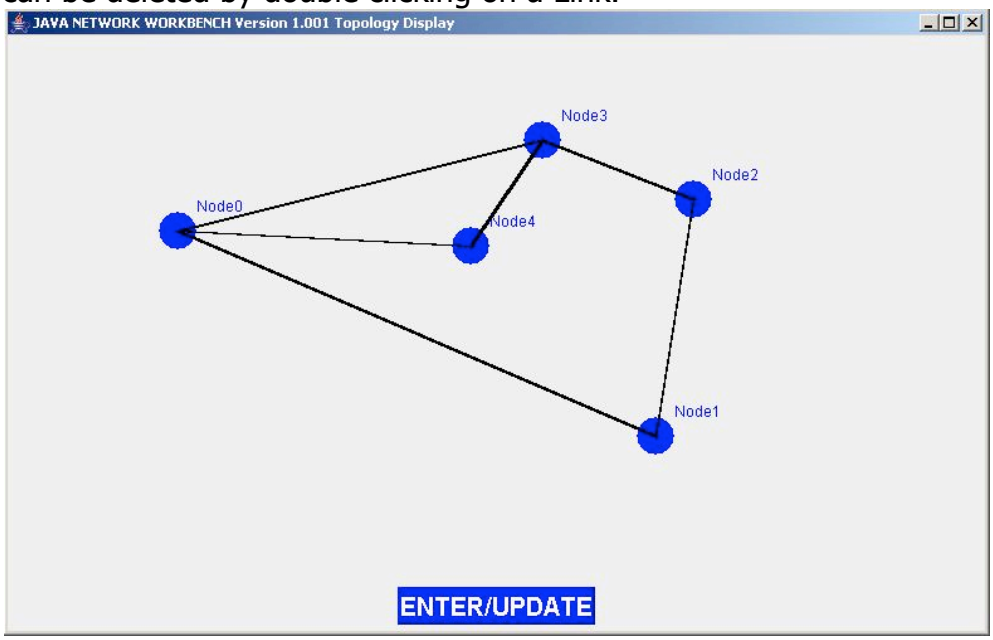


Figure 8 Network After Deleting Link with Double-Click on Link

Nodes and their attached Links can be deleted with a double-click on the Node. The nodes and the Links are automatically renumbered.

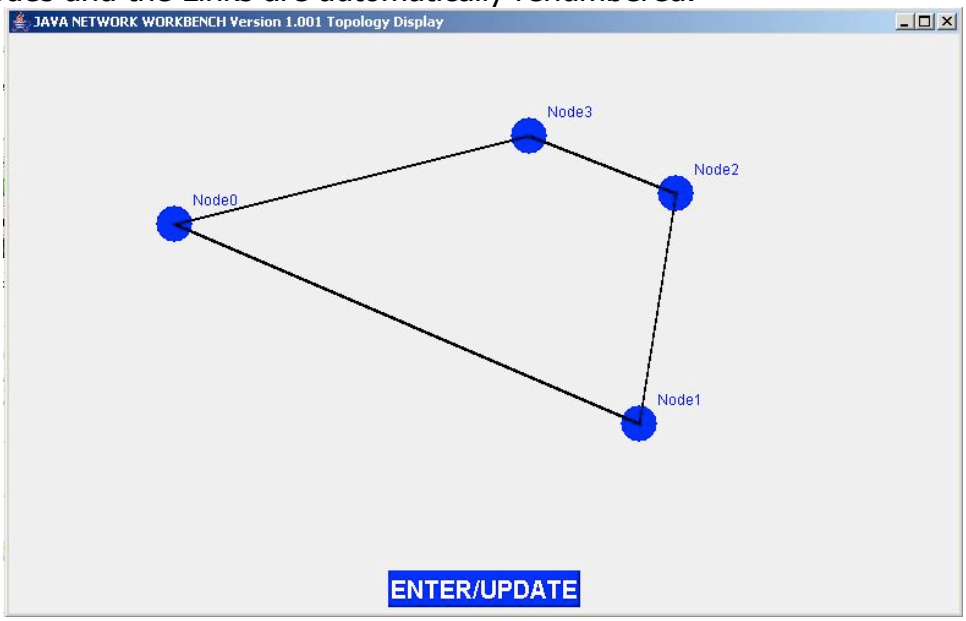


Figure 9 Network After Deleting Node with Double-Click on Link

Link characteristics can be displayed and edited in the input GUI window by right-clicking on a Link

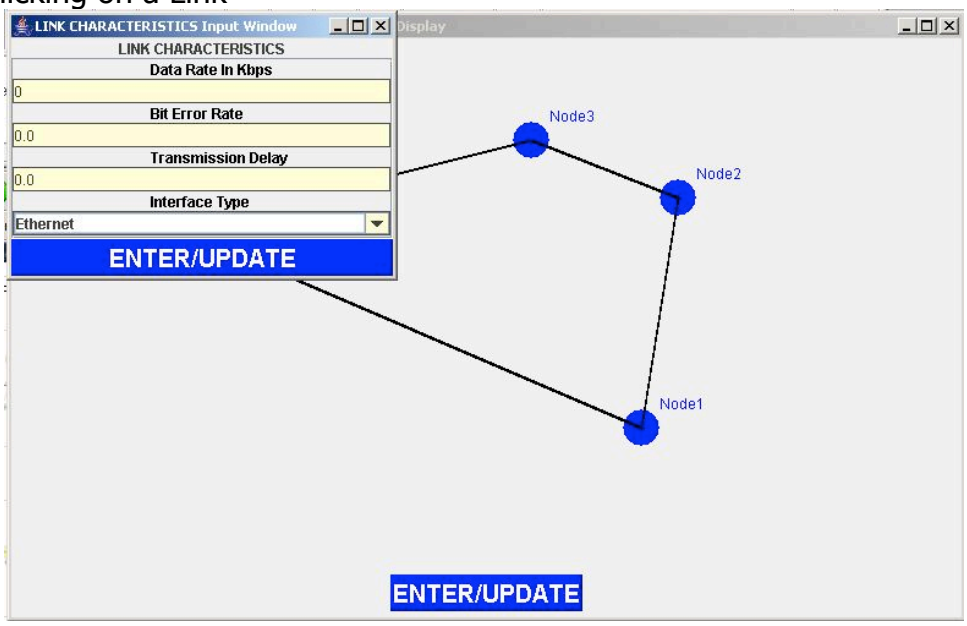


Figure 10 Reviewing Link Characteristics with a Double-Click on Link

Clicking the "ENTER/UPDATE" button of the Input Window will save the Link characteristics. Clicking the "ENTER/UPDATE" button of the Topology Display window will save the complete topology.

Graphical Command Summary

The table below summarizes the graphical commands.

Desired Action	Procedure
Create Node	Click on graphical input
Delete Node (and attached Links)	Double-click on an existing Node
Move a Node (and its attached Links)	Click on an existing Node, and drag it to the new position
Create Link	Right-click and drag between existing Nodes
Delete Link	Double-click on an existing Link
View Link characteristics	Right-click on an existing Link
Change Link characteristics	Right-click on an existing Link, make changes in the input GUI, and press the "ENTER/UPDATE" button
Save the topology	Click the "ENTER/UPDATE" button on the Topology Display window

Table 1 Graphical Input Display Commands

Desired Action	Procedure
Move a Node (and its attached Links)	Click on an existing Node, and drag it to the new position
View Node characteristics (buffer status)	Right-click on an existing Node
View Link characteristics	Right-click on an existing Link
View Interface characteristics	Right-click on an existing Interface

Table 2 Graphical Output Display Commands

SAVE TOPOLOGY AS TEXT FILE

JNW can write the topology to a text file. Click on the button on the MainGUI. This will open a JChooserPanel which will permit the user to decide where to save the topology. For example, the topology below

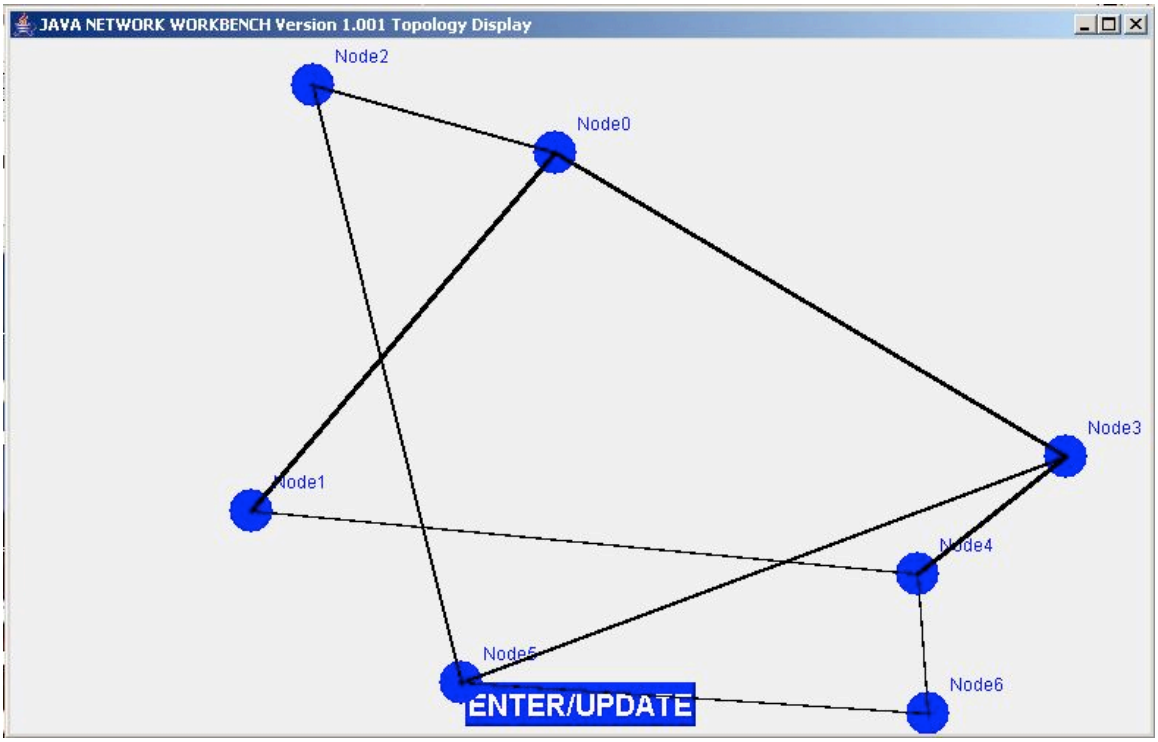


Figure 11 Sample Topology

saves as

```
// FORMAT: SOURCE NODE NUMBER, DESTINATION NODE NUMBER, CAPACITY IN
KBPS
```

```
// FORMAT: INTERFACE TYPE, BIT ERROR RATE, TRANSMISSION DELAY
// Network type is
NETWORK-TYPE:Router_Network
0,1,10000,Ethernet,1.0E-9,60.0
0,2,10000,Ethernet,1.0E-9,90.0
0,3,10000,Ethernet,1.0E-9,20.0
1,4,10000,Ethernet,1.0E-9,30.0
2,5,10000,Ethernet,1.0E-9,20.0
3,4,10000,Ethernet,1.0E-9,30.0
3,5,10000,Ethernet,1.0E-9,40.0
4,6,10000,Ethernet,1.0E-9,40.0
5,6,10000,Ethernet,1.0E-9,50.0
```

which is the topology from Figure 8.8 of the textbook.

ROUTER NETWORK TOPOLOGY

In Network Workbench a router network is a multiple-node network connected by any combination of links.

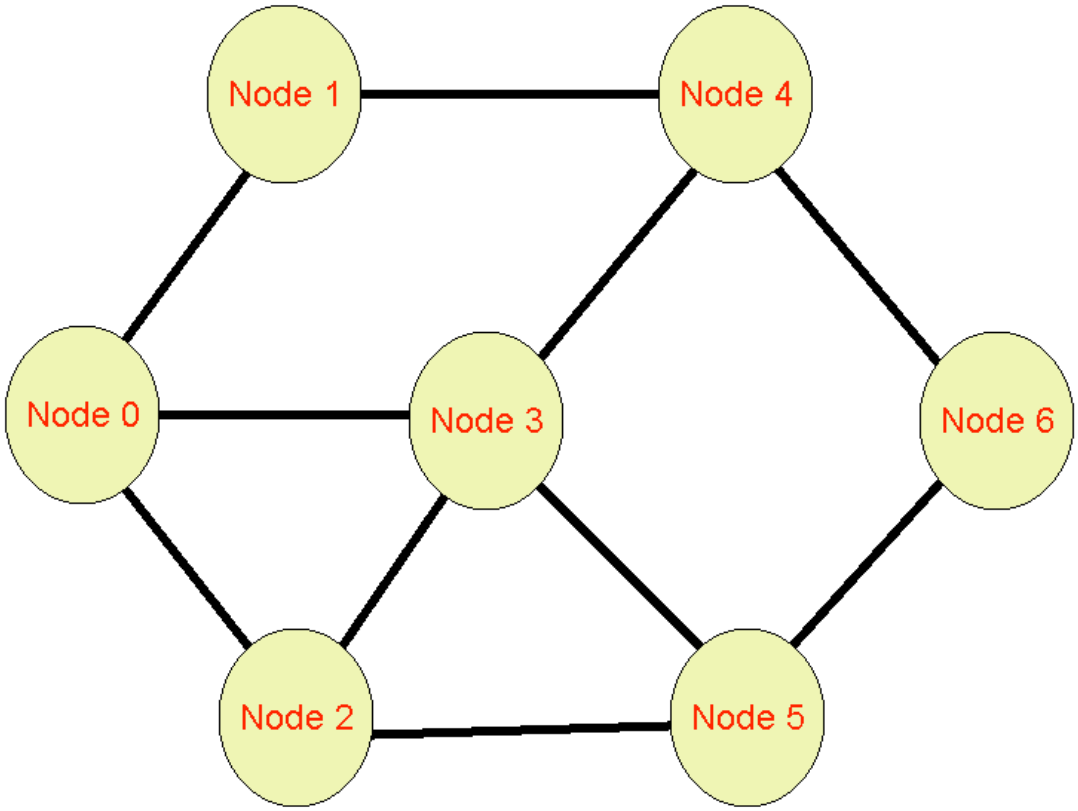


Figure 12 Example Router Network Topology (Figure 8.8 of Pullen)

This example can be found in Figure 8.8 of "Understanding Internet Protocols Through Hands-On Programming," J. Mark Pullen, Wiley Computer Publishing, 2000.

The topology input file can be read using the chooser below by using the button. Use the chooser to find the text file containing the topology definition.

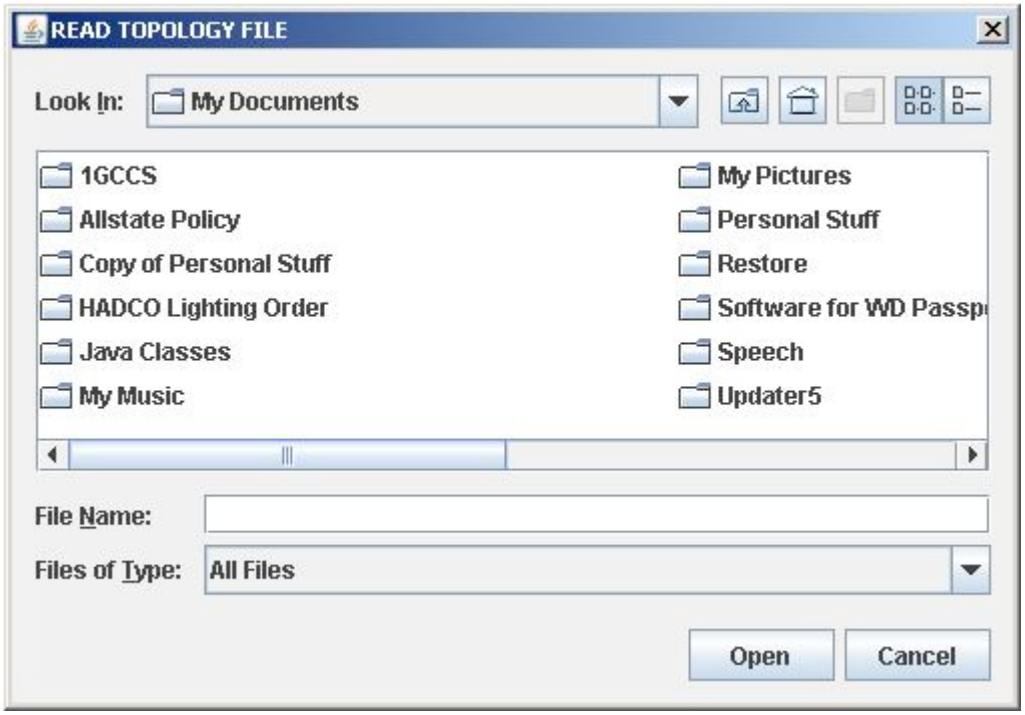


Figure 13 Chooser Panel for Reading Topology File

Just click on the text file, and click the open button. When the topology file is read, it creates the topology, for example the figure below

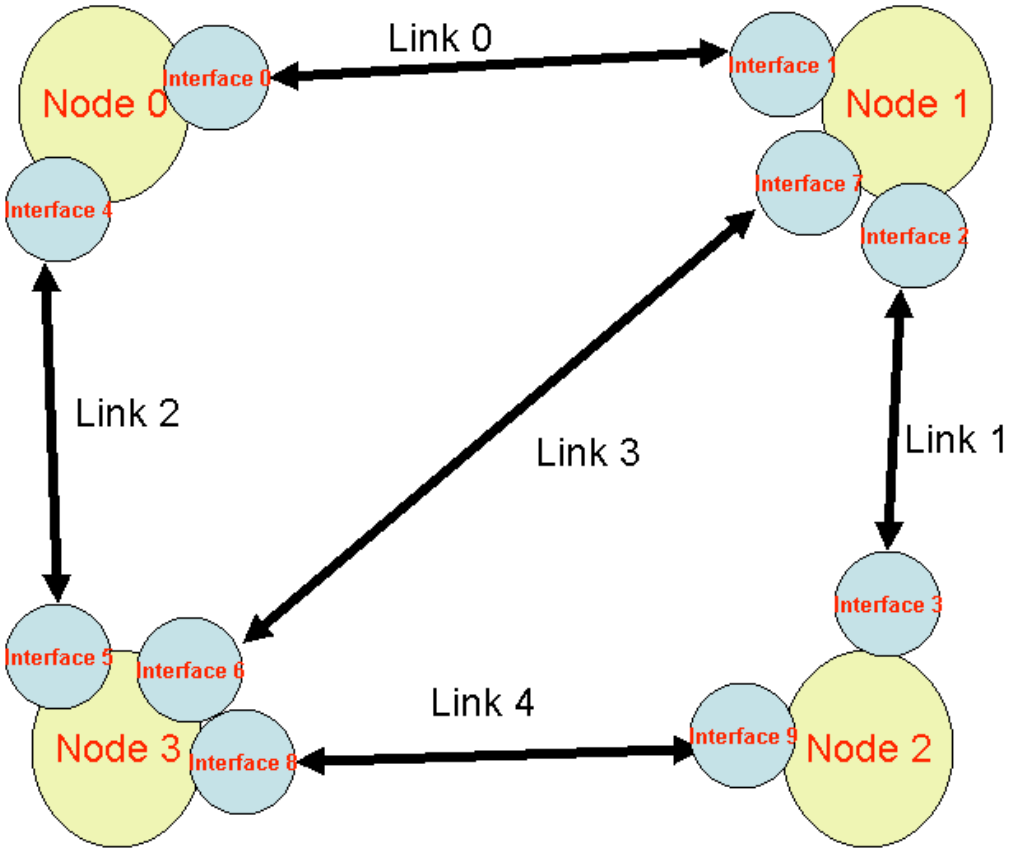


Figure 14 Network Workbench Representation of Router Network

JNW creates Java vectors of nodes, interfaces and links. It links them together as shown in the figure 15 below. This way each node, interface, and link has a unique address.

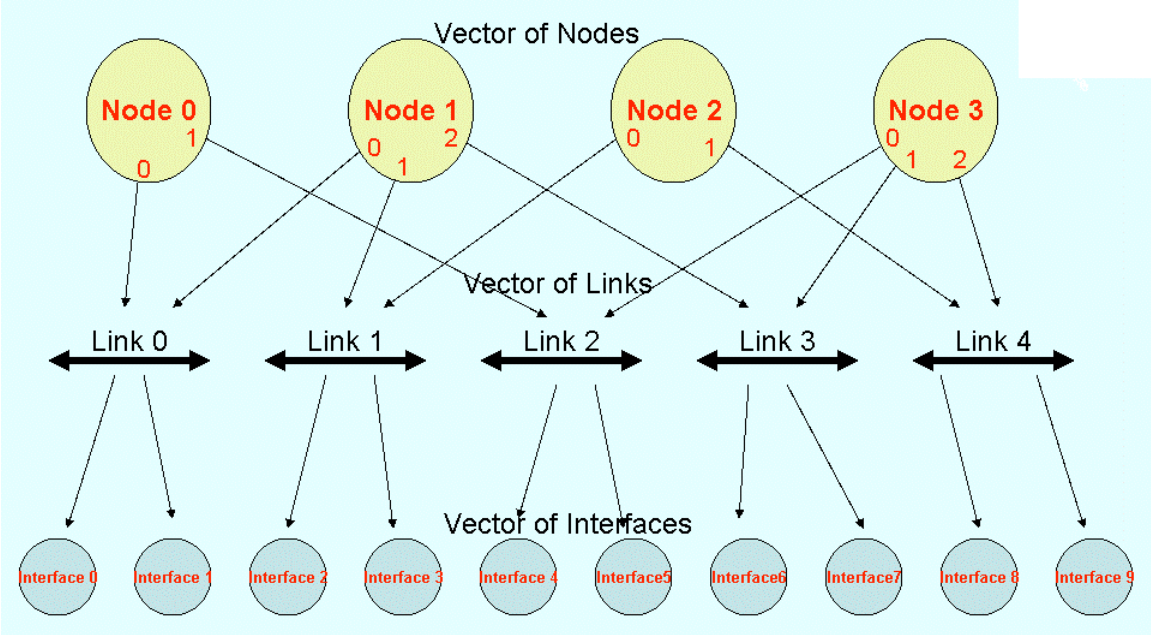


Figure 15 Data Structures in JNW

ROUTER NETWORK

After Network Workbench reads the topology file, it calls ModelNetworking.Routing.ComputeCostMatrix to compute the cost matrix, cost [i] [j] [k], of moving a frame from node j to node k as seen from node i. Presently the cost is just the transmission delay time between nodes j and k. Later the student could dynamically update the cost matrix to include additional time-varying costs.

After computing the costMatrix, Network Workbench executes ModelNetworking.Routing.ComputeRoutingMatrices which uses Dijkstra's algorithm to compute the best routing between node i and all the other nodes. The result is stored in the matrix routingMatrix. It then copies the ith row of routingMatrix into the routingTable, for each node i.

One could dynamically update the routing by calling Networking.Routing.ComputeCostMatrix and Networking.Routing.ComputeRoutingMatrices during the execution of JNW. Presently a call to Networking.Routing.ComputeRoutingMatrices automatically calls Networking.Routing.ComputeCostMatrix, so a separate call to Networking.Routing.ComputeCostMatrix in unnecessary. This may be separated in the future.

You can re-enter a different topology at any time. Each time you re-enter the topology file, the old topology is deleted and the costMatrix and the routingTable are recomputed for each node.

ETHERNET BUS TOPOLOGY

An Ethernet bus has N nodes but only one link, the Ethernet bus.

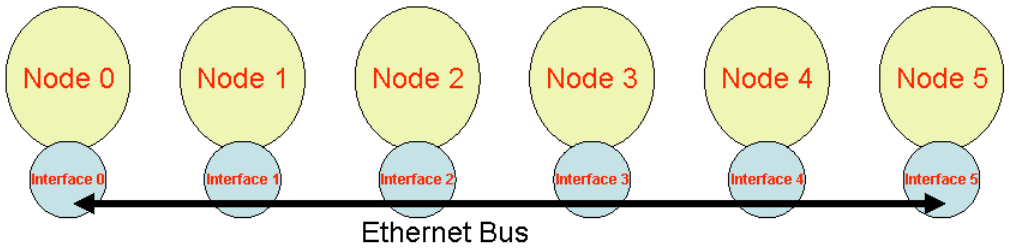


Figure 16 Representation of Ethernet Bus in Java Network Workbench

An example Ethernet topology input file is

```
// FORMAT: SOURCE NODE NUMBER, DESTINATION NODE NUMBER, BANDWIDTH
// FORMAT: LINK TYPE, BIT ERROR RATE, TRANSMISSION DELAY, INTERFACE TYPE
//
0,1,10000,Ethernet, 0.0000001 , 5
//
1,2,10000,Ethernet, 0.0000001 , 10
//
2,3,10000,Ethernet, 0.0000001 , 15
//
3,4,10000,Ethernet, 0.0000001 , 20
//
4,5,10000,Ethernet, 0.0000001 , 25
//
5,6,10000,Ethernet, 0.0000001 , 30
NETWORK-TYPE:Ethernet_Bus
```

This defines a six-node, 10-megabit Ethernet bus with distance of 6 kilometers to 21 kilometers between adjacent nodes. Please note that in a router network it is OK to enter

```
4, 6, 10000, Ethernet, 0.0, 100
5, 6, 10000, Ethernet, 0.0, 70
```

This defines the delay times between nodes 4 and 6 as 100 milliseconds and the time delay betweens nodes 5 and 6 as 70 milliseconds. But in an Ethernet bus it only makes sense to include the delay between adjacent nodes such as

5, 6, 10000, Ethernet, 0.0, 70

The Ethernet bus would not know what do with

4, 6, 10000, Ethernet, 0.0, 100
5, 6, 10000, Ethernet, 0.0, 70

BASIC MODEL OPERATION

Each node in a router network can have any number of interfaces. Each Ethernet bus or token ring node has only one interface. Within one node the interfaces are numbered 0 to $N_{Interface}$(node), for 0 <= node < N. Additionally, each interface has a unique sequence number, shown in figure 7 above, 0 to $\sum_{node=0}^{node=N} (N_{Interface}(node)+1)$.

Each node has a buffer for storing packets. All buffers in JNW will overflow when they exceed their programmed limit. The message, packet, or frame will be lost.

In a router network, the node has a buffer for performing packet routing. Each interface has transmit and receive buffers. In there buffers, the IP (UDP or TCP) is enclosed in the appropriate data-link-level wrapper (for example the Ethernet header and checksum) , before frame transmission or the wrapper is removed. After removing the wrapper, the packet is sent to the node buffer where it is routed using the routingTable.

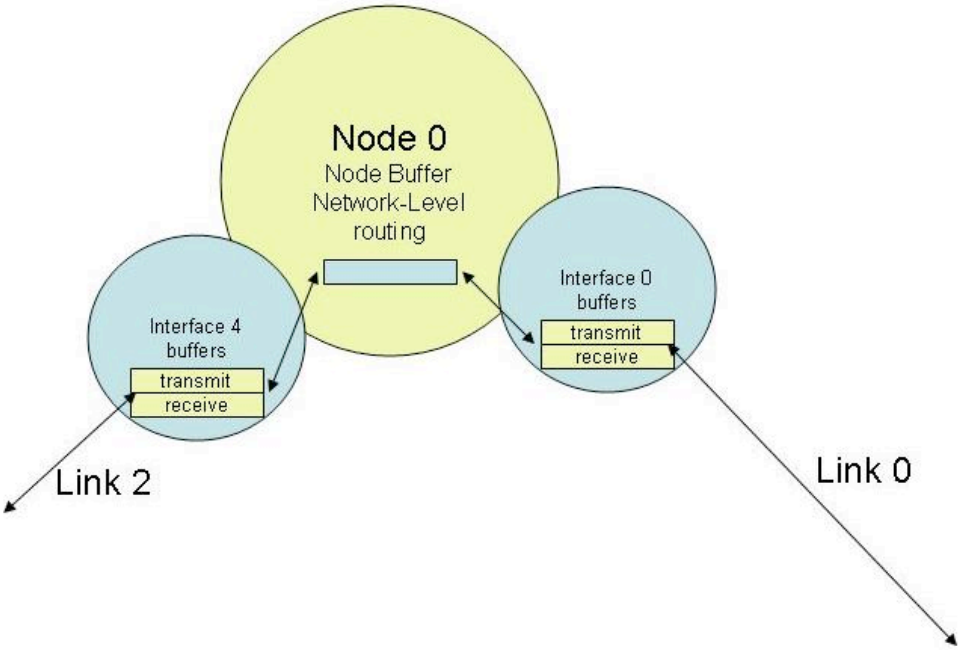


Figure 17 Node Structure for Router Network

In Ethernet bus, each node has only one interface.

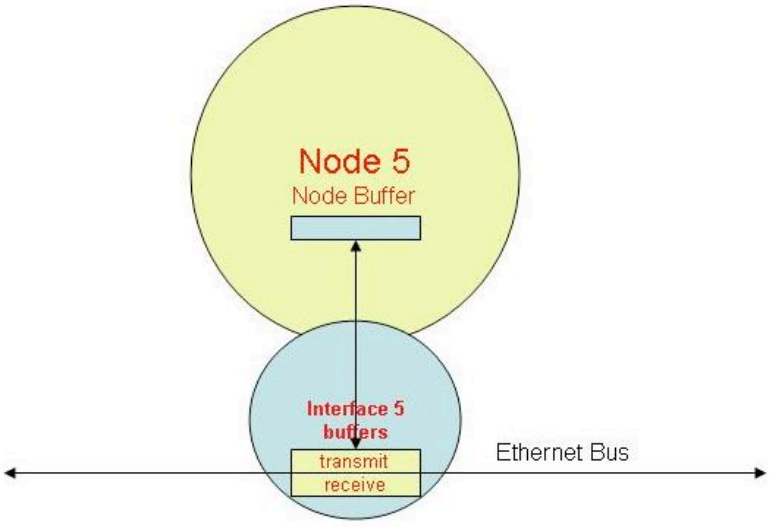


Figure 18 Node Structure for Ethernet Bus

When the Ethernet frame is received, the interface reads the destination address. If the node is not the destination, the frame is discarded. If the node is the destination, the frame is forwarded to the node buffer where the e-mail is removed from the packets for display.

READING E-MAIL

E-mail messages are read using another chooser panel from text files by clicking on the button. For example the file EMAIL_UDP_2.txt contains

```
FROM:0
TO:6
PROTOCOL:UDP
/FROM:0
/TO:6
/PROTOCOL:UDP
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Characters:  0400
MESSSAGE-END:


FROM:6
TO:0
PROTOCOL:UDP
/FROM:6
/TO:0
/PROTOCOL:UDP
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Characters:  0
```

An e-mail can be of any length, and you may have a many emails in the file as you wish.

JNWS now reads the source node and destination nodes from the email file. If you fail to provide both TO and FROM entries, an error messages is printed. The email ends with "MESSSAGE-END:". Further you cannot open the email chooser panel until you have read in the topology. Otherwise, JNW will issue an error message.

Each time you read in an e-mail text file, the file is queued in the buffer of the source node. Therefore, you can queue several e-mail messages before starting to execute the model. You can put the messages in different nodes if the enter different source nodes in FROM: field. For example, you could send a message from node 0 to node 1 and simultaneously a message from node 1 to node 0.

UDP MULTICASTING

UDP multicasting is not yet implemented.

MESSAGE LOAD GENERATOR

JNW provides the capability to create a sequence of blank UDP messages. This may be useful for demonstrating buffer overflow or buffer performance.

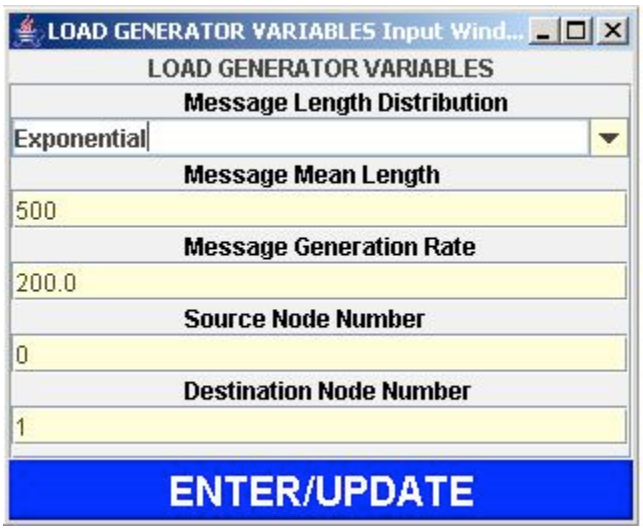


Figure 19 Load Generator Input GUI

The user specifies the message length distribution, fixed or exponential, the means message length, the rate of messages creation in messages per second, and the source and destination nodes. When the use executes the model, a message generation events executes at time 0. It uses the random number generator to define the message length and the time that the next message will be generated. If the load generator output is printed the output for an exponentially-distributed length with mean length 5000 bytes, and generated at 500 messages a second the output might be:

```
At time 0 a message of length 3887 bytes is sent
At time 8188 a message of length 535 bytes is sent
At time 26994 a message of length 5087 bytes is sent
At time 32757 a message of length 3567 bytes is sent
At time 47630 a message of length 6926 bytes is sent
At time 90938 a message of length 2835 bytes is sent
At time 116804 a message of length 3489 bytes is sent
At time 149847 a message of length 1988 bytes is sent
At time 152895 a message of length 7288 bytes is sent
At time 163320 a message of length 3314 bytes is sent
At time 165939 a message of length 1639 bytes is sent
At time 168415 a message of length 4011 bytes is sent
At time 178948 a message of length 2716 bytes is sent
At time 185659 a message of length 3859 bytes is sent
At time 209338 a message of length 4196 bytes is sent
At time 228727 a message of length 11928 bytes is sent
```

The load generator wraps the blank messages in UDP and IP headers and put the packet into the source node buffer for transmission. The message is filled with "00000000"s. The message length includes the UDP and IP headers but not the Ethernet headers and trailer.

Only one GenerateMessageLoad event can exist in JNW. JNW is designed to create one GenerateMessageLoad event only when the ENTER/UPDATE button is pushed on the load generator variables GUI. The load generator variables can be updated; when JNW executes it will execute with the latest data entered.

JNW EXECUTION

Use the button to start the simulation. Each time you click the button, the random number generator is reinitialized to the random number generator seed you entered on the Global Variable input window. So, each time you execute JNW with the same random number seed, you will get the same results.

Output is written to the center panel, but it can be directed to standard output using the boolean Utility.Print.Out.sendOutputToStandardOutput = true ;

IMPLEMENTING LOOK AND FEEL

JNW implements different look-and-feels. If for example, you are running UNIX, you may wish to change the look and feel to OpenLook as shown in the figure below.

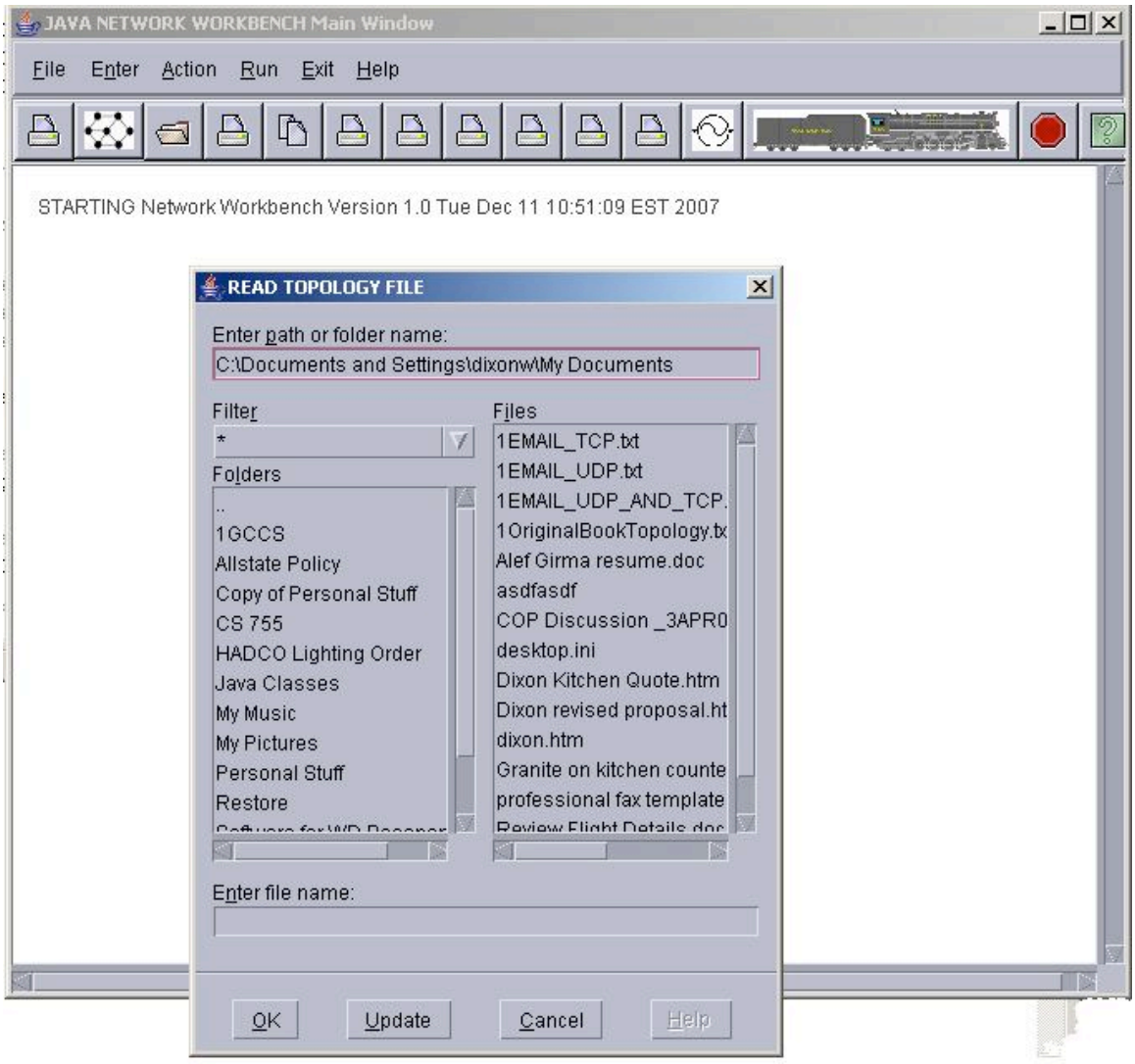


Figure 20 OpenLook Look and Feel in JNW with LOOK_AND_FEEL_CONSTANT = 1

The looks and feels can be changed by setting GUIConstants.General. lookAndFeelConstant. The possible settings are lookAndFeelConstant = 0 for "Metal," 1 for "Openlook," and 2 for "Windows."

SWING CONCURRENCY PROGRAMMING

JNW is a model employing active entry of data and controls from the input GUIs and extensive computation. This requires that the programmer carefully separate the GUI from the computations. Read http://java.sun.com/docs/books/tutorial/uiswing/concurrency/index.html for a brief description of the problem and the solution. As suggested in the web page http://java.sun.com/docs/books/tutorial/uiswing/concurrency/initial.html , the MainGUI is started with

```
javax.swing.SwingUtilities.invokeLater(new Runnable() {
public void run() {
GUIs.MainGUI maiGUI = new GUIs.MainGUI ( ) ;  }  } ) ;
```

not just

```
new GUIs.MainGUI ( ) ;
```

In class GUIActions. RunModelAction and the two other Actions supporting model execution, the model is run by executing

```
GUIGraphical.SwingWorker < java.lang.Void , java.lang.Void > swingWorker
  = new GUIGraphical.SwingWorker < java.lang.Void , java.lang.Void > ( ) {
  public java.lang.Void doInBackground ( )
  {
   ModelDES.DiscreteEventSimulator.getInstance ().StartModel ();
   return null ;
  }
} // end of anonymous inner class
;
swingWorker . execute ( ) ;
```

The SwingWorker class is not available until Java 1.6. I therefore took backport of SwingWorker, available from https://swingworker.dev.java.net/ and put the classes into package GUIGraphical by changing the package name. This cause the discrete event simulator to run in background. This permits the GUI to respond rapidly to user input and the update outputs.

DEMONSTRATIONS

JNW has several demonstrations provided as .jar files. Open a command prompt window. Go to the folder containing the .jar files and type

```
java –jar BitStuffing_0s_and_1s.jar
```

or

```
java –jar BitStuffing_EMail.jar
```

or

```
java –jar CRC_Computation.jar
```

ERROR MESSAGES

JNW has extensive internal error testing. Error messages, staring "ERROR" are written to the standard error output. If Print Errors on MainGUI is checked (default setting) on the global input variables, the errors will also be printed on the Main GUI in red, as shown in the following example.

```
ERROR IN TCPDestinationBuffer.RemoveBottomNRow attempting to
remove bottom 5 rows from dataArray of length 50
```

JNW also issues "WARNING" and "NOTIFICATION" messages. These are warnings respectively that something may be wrong and that something unusual is happening. A NOTIFICATION is not an error. An array overflow is an error. If the array is automatically adjusting its size to avoid an overflow it will issue a NOTIFICATION. Warnings and notifications are written to the standard error output. If Print Warnings on MainGUI is checked (default setting) on the global input variables, the warnings and notifications will also be printed on the Main GUI in blue.

```
NOTIFICATION in Utility.TCPDestinationBuffer EnlargeDataArray :
dataArray overflow.
More than 20 packets. Increasing dataArray size by 10.
```

FURTHER READING

More material for JNW is contained in the slide presentation Class_Presentation_20070903.ppt.

AUTHORS

JNW was written by Dr. J. Mark Pullen,

Professor of Computer Science, George Mason University and Director, Center of Excellence in Command, Control, Communications, Computing and Intelligence (C4I Center), and Bill Dixon,

a graduate student. (Now you know why the colorful Run button on JNWS!)

APPENDIX A: Utility.JNWBitSet DESCRIPTION

Utility.JNWBitSet, a subclass of java.util.BitSet, is a class of static methods to support bit processing in Java Network Workbench.

java.util.BitSet represents bits as a vector of boolean values (true and false). It is a very flexible class since it allows bit sets to grow or shrink to any size. It has many methods to support bit operations. You can find a JavaDoc for java.util.BitSet at http://java.sun.com/j2se/1.5.0/docs/api/java/util/BitSet.html .

Unfortunately, from our point of view java.util.BitSet has two errors. In java.util.BitSet the bit set 00000000 00000000 has length 0. java.util.BitSet uses the position of the first 1 as its `length()`. We want 00000000 00000000 to have length 16. Therefore, a JNWBitSet has two lengths, the `length()` inherited from java.util.BitSet and the JNWLength, which is the total length of the bit set including all the leading zeros. I have not overridden `length()` because JNWBitSet relies on java.util.BitSet and the performance of java.util.BitSet depends on `length()`. So there are two lengths: Be sure to use the correct one, JNWLength.

java.util.BitSet also has a very strange behavior. java.util.BitSet is supposed to automatically expand if you enter a bit outside the current length. That is, data.set (16*1024, true), will automatically expand data to 16K length, but data.set (16*1024, false) will not! java.util.BitSet is only interested in the first true and all the bits below that. Be careful with java.util.BitSet. JNWBitSet solves both these problems. The JNWLength is the total length of the bit set, and data.set (16*1024, false) will expand data to 16K length. JNWBitSet also provides a very complete set of utilities for your use.

Therefore, it was necessary to create the subclass Utility.JNWBitSet which corrects these errors. The correct length, for our purposes, is usually JNWLength. Be very careful when using any of the methods of class java.util.BitSet since they may produce unexpected results.

There are several things you should know about JNWBitSet. java.util.BitSet and Utility.JNWBitSet represent bits as booleans (trues and falses) not 0s and 1s. So, if you went to set the indexValue bit to 0, you must use

```
set (indexValue, false);
```

and

```
set (indexValue, true);
```

to set it to 1.

There are two methods in Utility.JNWBitSet, getBitValue (bitIndex) and setBitValue (bitIndex , 0_or_1) , which get and set bits as 0s and 1s. The

methods internally convert 0 to false and 1 to true. If you want boolean input and output, **get** (bitIndex) and **set** (bitIndex), and **set**(bitIndex, true_or_false) from java.util.BitSet work fine.

An Utility.JNWBitSet with length JNWLength has an index from 0 to JNWLength-1 as shown in the figure below, with the convention that **THE 0 INDEX BIT IS THE RIGHTMOST BIT AS SHOWN BELOW. THIS IS AN UNAVOIDABLE REVERSE OF THE C++ VERSION, BECAUSE THE SUPERCLASS java.util.BitSet FOLLOWS THIS CONVERTION.**

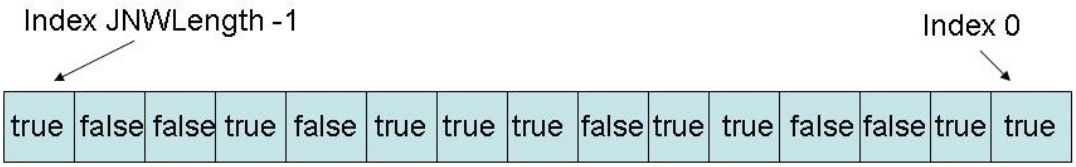


Figure 1 Indices of a JNWBitSet

JNWBitSet has several useful methods

- setJNWLength (int length) sets the length of the JNWBitSet
- getJNWLength ( ) gets the length of the JNWBitSet
- setBitValue (int bitindex , int value) sets the bit at index bitIndex to 0 or 1
- getBitValue (int bitindex) gets the bit at index bitIndex as a 0 or 1
- shift (int numberOfBitsToShift) shifts the BitSet left (positive) or right (negative) by numberOfBitsToShift bits
- setJNWBitSetToString (String string) converts a String into a JNWBitSet
- toString ( ) converts a JNWBitSet to a formatted String of 0s and 1s, 40 bits per row
- toString (int numberOfBits) converts a JNWBitSet to a formatted String of 0s and 1s, numberOfBits bits per row

Notice that a shift with a positive index causes the bits to shift to the left and a shift with a negative shift causes the bits to shift to the right. The effects of a shift are shown in the figure below.

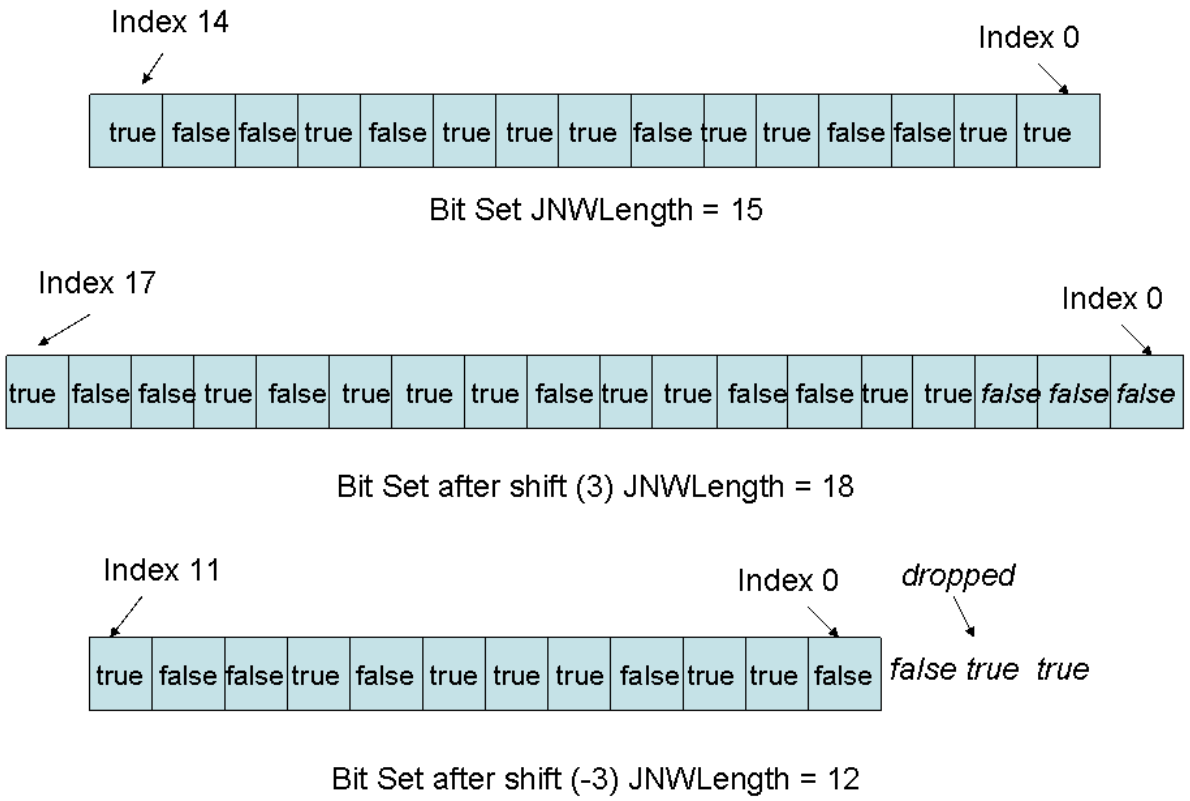


Figure 2 Effects of shift ( ) Methods on a JNWBitSet

There are several methods to convert between Strings, integers, etc. and JNWBitSets. They include

- setJNWBitSetToLongAndTruncate (long inputIn , int numberOfBits) converts a long, int, short, or byte to a bit set and truncates *this* to a specified number of bits
- setJNWBitSetToString (String string) converts a String to a bit set.
- setJNWBitSetToZerosAndOnes (String stringOfZerosAndOnes) converts a String of 0s and 1s, such as "01010011 10101010" to a bit set. The acceptable characters are "0", "1", and " ", so you can enter two octets as "00000111 11111100".
- convertJMUBitSetToLong ( ) converts a bit set to a long. You can then cast it to an int, short, or byte, as, for example;
  int portNumber = (int) dataBitSet. convertJMUBitSetToLong ( ) ;
- convertJMUBitSetToString ( ) converts a bit set to a String

There are several methods for appending and prepending bit sets. For example,

- data.append ("01010011 10101010") appends 0101001110101010 to

the end of data
- data.prepend ("01010011 10101010") prepends 0101001110101010 as the header of data

as do

- data.append (bitSet) appends bitSet to the end of data
- data.prepend (bitSet) prepends bitSet as the header of data

There are methods for getting and setting subsets of a bit set.

- getSubset (int fromIndex, int toIndex) returns a clone of the subset from fromIndex (inclusive) to toIndex (exclusive). For example, data.getSubset (data.getJNWLength ( ) -32, data.getJNWLength ( ) – 24) will return the fourth octet of data.
- setSubset (int fromIndex, JNWBitSet data) sets the subset of *this* from fromIndex (inclusive) to data.

Constructors
Utility.JNWBitSet also has several useful constructors

- JNWBitSet(int nbits) creates a JNWBitSet of length nbits with all the bits false (0)
- JNWBitSet (String stringOfZerosAndOnes) creates a bit set of 0s and 1s. For example,
  JNWBitSet data = new JNWBitSet ("00001111 1111 0000 10") ;
  creates the bit set 000011111111000010.
- JNWBitSet (long inputIn , int numberOfBits) converts a long into a bit set of length numberOfBits.
  For example
  JNWBitSet data = new JNWBitSet (765 , 16) ;
  creates the bit set 0000001011111101.
- JNWBitSet (JNWBitSet JNWBitSetIn) is a cloning constructor which creates an identical copy of JNWBitSetIn. The clone () method does not work correctly with java.util.BitSet-which is not unusual. Remember,

  JNWBitSet clone = dataBitSet ;

  will result in clone and dataBitSet pointing to the same data. If you manipulate clone, you are also manipulating dataBitSet. But

  JNWBitSet clone = new JNWBitSet (dataBitSet) ;

creates two identical copies. Later a clone method was added to Utility.JNWBitSet, so you can now say

```
JNWBitSet clone = dataBitSet. clone ( )  ;
```

One useful method from class java.util.BitSet is

```
data. xor ( argument )
```

which performs a logical bit-by-bit XOR of the bit set data with the bit set argument. data and bitSet need not be the same length. XOR starts with the rightmost (index 0) bit and performs the xor function until it reaches the end of either bitSet or data.

Another useful method in Utility.JNWBitSet is

```
data.xorAndShift (JNWBitSet xorIn )
```

This method computes the XOR of data and xorIn starting at bit 0. It then recursively computes the XOR at position xorIn.JNWLength, 2 * xorIn.JNWLength, etc. until it has computed the XOR for all data. This method is useful for computing XORs over the IP header, UCP packet, and TCP packet.

The toString ( ) method is overridden in JNWBitSet, but a better substitute is

```
toString ( int columnWidth )
```

or

```
toString ( )
```

this writes the JNWBitSet as octets. The first method writes columnWidth bits per row and the second writes 40 bits per row.

IMPORTANT: though JNWBitSet, is a subclass of java.util.BitSet, it does not necessarily override the methods in java.util.BitSet. Some of these methods are still used internally in Utility.JNWBitSet. But the use of the java.util.BitSet methods directly may give unexpected results.

But most students need use **ONLY** the methods of JNWBitSet or the get and set methods from java.util.BitSet. Do not use the methods of java.util.BitSet unless you are positive of what you are doing. Unexpected results may occur.

If you need a new method in JNWBitSet, contact the authors.

APPENDIX B: RANDOM NUMBER GENERATOR

To ensure repeatability, JNWS has only one random number generator. You instantiate that random number generator as

```
Utility.JNWRandom randomNumberGenerator
  = Utility.JNWRandom.getInstance ( ) ;
```

This will give you access to all the methods of java.util.Random plus an Exponential number generator and a Poisson distribution:

double	**nextExponential**(double mu) Method to return exponentially distributed number with mean 1 / mu.
int	**nextPoisson**(double lambda) Method to return a Poisson distributed number with mean lambda.

java.util.Random is described in http://java.sun.com/j2se/1.5.0/docs/api/java/util/Random.html. Class Utility.JNWRandom also has a JavaDoc which you can create (in NetBeans) by clicking on Build on the menubar and then clicking on Generate Javadoc for "JNW".

Each time you run the model with identical input you should get identical results. The default random number generator seed is 123456789. You can change the random number seed at the General Variables pull-down on the MainGUI. The possible seeds are -9223372036854775808 to 9223372036854775807. That should give you enough to choose from!

CLASS Utility.JNWRandom_Non_Singleton

The class Utility.JNWRandom is a singleton. Later, it was decided to put separate random number generators on each Link. Therefore, a second class, Utility.JNWRandom_Non_Singleton was created. It is identical to class Utility.JNWRandom except that it is NOT a singleton. It can be instantiated in the normal way

```
Utility.JNWRandom_Non_Singleton randomNumberGenerator
      = new Utility.JNWRandom_Non_Singleton ( ) ;
```

APPENDIX C: INPUT AND OUTPUT

This appendix describes input and output to JNW. Input to JNW is either by reading text files, such as the topology, or though GUIs such as

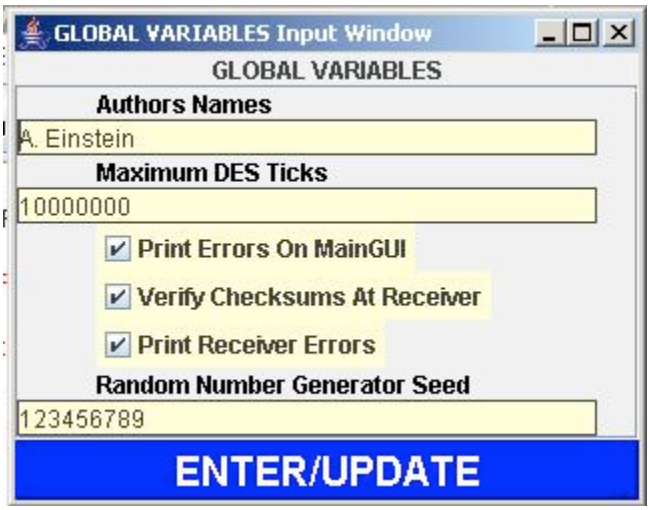


Figure 1 Example Input GUI

The GUIs control the entry of incorrect data. All data entered in the GUI is entered as a String that is converted to the correct data type, int, or JNWBitSet, etc. Anything that will throw an exception when the input String is converted to the desired data type, for example an integer, will cause the background int offending field to turn red. Bit sets are entered as Strings of zeros, ones, and blanks: Any other entry causes a red background.

Data that does not throw an exception IS updated. Only the incorrectly-entered data has background turned red. The GUI will not close until all the fields have been entered correctly. In the example below the authors name, interactive mode of operation, destination node number, network type, transmission mode, and random number generator seed have been updated. The other three fields are waiting for correct entries.

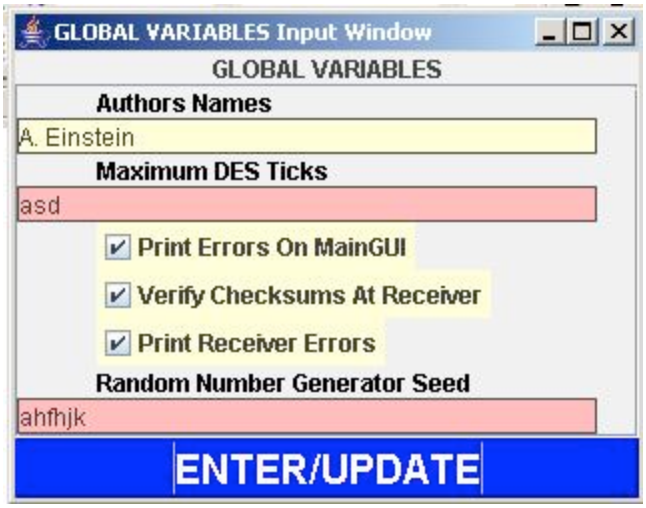


Figure 2 Example Input GUI With Errors

Java is extremely flexible in input and output. Unfortunately, this means that simple operations can become programming nightmares. Since the student will often have to read lines of input, a method for reading from standard input is provide in Utility.IO.ReadSingleLineFromStandardIO ( ) which returns a String.

A student who wants to write output can use System.out.println (String string) which sends the output to the standard output. But it is expected that the student will want to write output or errors to the center panel of the Main GUI. For this purpose use Utility.Out.print (String string), Utility.Out.println (String string), Utility.Err.print (String string), or Utility.Err.println (String string).

In Utility.Out.print or Utility.Out.println the output is always sent to the center panel of the MainGUI. Error message are always sent to System.err.println. But class Utility.Out and Utility.Err have three booleans

```
/**Boolean telling whether to print output sent to the Main GUI central
  *panel also to the standard output*/
   static boolean sendOutputToStandardOutput = false ;

  /**Boolean telling whether to print error messages also to the Main GUI.*/
   static boolean sendErrorsToMainGUI = true ;

 /**Boolean telling whether to print error messages also to
  * the standard output.*/
   static boolean sendErrorsToStandardOutput = false ;
```

by which the student can direct the output or errors to additional locations.

Saving Input or Output

The student can save the output in the center panel of the MainGUI by either (a) cutting and pasting the center panel into a text file or (b) under "File" on the menubar choose "Write Output Data as Text."

The input data can also be saved by under "Action" on the menubar choose "Copy Input Data to Output." This will cause JNW to write all the input data to the center panel of the MainGUI. The student can then save the center panel as described in the previous paragraph. Clicking the button on the toolbar will also copy the input to the center panel of the MainGUI.

Choosing "File" on the menubar choose "Write Input Data as Text" will write all the input data directly to a file. Choose .rtf as the file name extension.

APPENDIX D: DATA FORMATS

This appendix explains the data formats used throughout the model.

Ethernet Frame Format

The Ethernet Frame is a full Ethernet frame as shown in Figure 4-17 (b) of *Computer Networks*, Fourth Edition by Andrew S. Tananbaum. See also http://en.wikipedia.org/wiki/Ethernet .The frame includes an 8 byte preamble including a SOF (start of frame), two 6-byte addresses, the 2-byte length of the frame, data, a pad (if needed) and a full CRC-32 checksum as defined in http://en.wikipedia.org/wiki/Cyclic_redundancy_check . Ethernet headers and redundancy codes are added in class ModelDataLink.EthernetUtilities. AddEthernetHeaderAndTrailer. Data is padded into the data field if the data is too short, but JNWBitSet does not limit the frame length to 1500 bytes.

IPv4 Header Format

The IPv4 header is defined in Figure 5-53 of *Computer Networks*, Fourth Edition by Andrew S. Tananbaum. The version is 4, the IHL is 5, the type of service is "00000110," which is low priority for everything, the identification is 0, the DF flag is set, but not the MF flag, the fragment offset is 0, the time to live is set to 255, the protocol is 7 for UDP and 6 for TCP. The header checksum is performed over the header only. See www.freesoft.org/CIE/Course/Section3/7.htm for a detailed description of the format. The IP header is created in ModelTransport.IPUtilities. ConstructIPHeader.

UDP Header Format

The UDP header is defined in Figure 6-23 of *Computer Networks*, Fourth Edition by Andrew S. Tananbaum. The source and destination ports are set in ModelTransport.UDPState to 23, the TELNET port. The checksum is performed over the data, the header and the pseudoheader. The pseudoheader is shown in Figure 6-30 of *Computer Networks*, Fourth Edition by Andrew S. Tananbaum, except that protocol = 17. See www.protocols.com/pbook/tcpip2.htm for a detailed description of the format. The IP header is created in ModelTransport.UDPUtilities ConstructUDPHeader.

TCP Header Format

The TCP header is defined in Figure 6-29 of *Computer Networks*, Fourth Edition by Andrew S. Tananbaum. The source and destination ports are set in ModelTransport.TCPState to 23, the TELNET port. The sequence number and acknowledgement numbers change as messages are sent and received. All the flags are set to 0, except the FIN bit which is set to 1 for the last TCP packet of a message. The TCP header length is 5, the checksum is performed over the data, the header and the pseudoheader. The pseudoheader is shown in Figure 6-30 of *Computer Networks*, Fourth Edition by Andrew S. Tananbaum. The urgent pointer is 0. See www.freesoft.org/CIE/Course/Section4/8.htm for a detailed description of the format. The TCP header is created in ModelTransport.TCPUtilities ConstructTCPHeader.

APPENDIX E: NOMENCLATURE

This software was developed over a long time. In order to ensure understanding some common nomenclature is defined. You may find some exceptions in the code, but this nomenclature is intended to be used in naming methods and variables.

Frame, Packet, and Data Nomenclature

The figure below defines the frame, packet and data nomenclature. The "data" can be either an email messages, or, when using the load generator, a string of 0s.

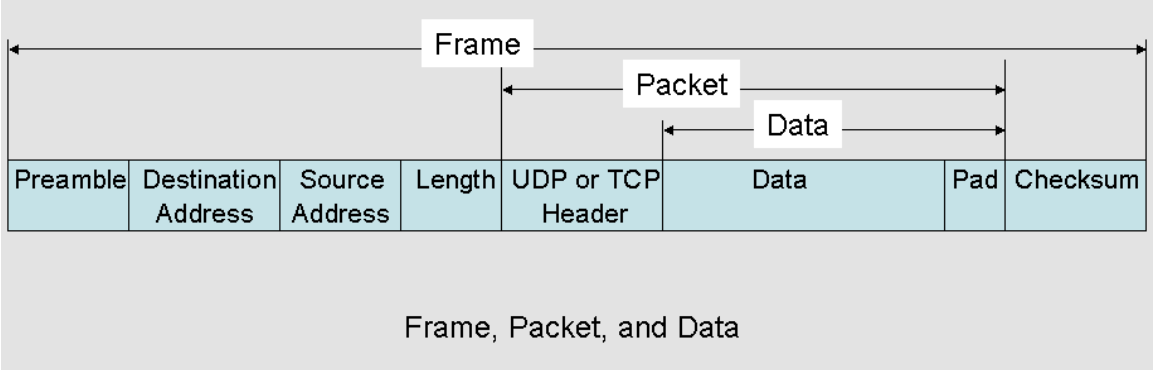


Figure1 Definitions of Frame, Packet, and Data for an Ethernet Frame

As much as possible throughout the model, email Strings are called data, IP packets are called IPPacket, and UDP packets are "UDPPacket." There is also UDP_IPPacket and TCP_IPPacket. Since presently only an Ethernet interface is implemented, all frames are called "frame."

an Ethernet interface is implemented, all frames are called "frame."

APPENDIX F: DATA FORMATS AND CONVENTIONS FOR COMMONLY-USED VARIABLES

Generally the Code Conventions for the Java™ Programming Language of http://java.sun.com/docs/codeconv/html/CodeConvTOC.doc.html are followed. Variable names follow the convention of Deitel & Deitel, Java How to Program, Prentice Hall.

This model was developed over a long time. In order to ensure consistency, commonly-used variables must use the same data formats and they are defined below.

VARIABLE	FORMAT	RANGE
sourceNodeAddress	long	0 (000.000.000.000) to (2**32) -1 (255.255.255.255)
destinationNodeAddress	long	0 (000.000.000.000) to (2**32) -1 (255.255.255.255)
sequenceNumber	long	0 to (2**32) -1
acknowledgementNumber	long	0 to (2**32) -1
sourceNodeNumber	int	0 to numberOfNodes (for example < 255)
destinationNodeNumber	int	0 to numberOfNodes (for example < 255)

Since IP addresses are 32 bits long it is not always possible to represent an IP address using type *int*. All class B, C, D, etc. addresses have their highest bit set. If the address is represented as an int or Integer, the number will be negative which can lead to problems. An address must be able to correctly represent any address in an IPV4 address. A Number is intended for addressing the node in the vectorOfNodes. Therefore, it will normally represent a small number, say < 255

So in most situations a node number is adequate, but sometimes a node address is necessary. Vectors are addresses as ints as

```
node = vectgorOfNodes . get ( sourceNodeNumber ) ';
```

or with longs as

```
node = vectorOfNodes . get ( ( int ) sourceNodeAddress ) ';
```

APPENDIX G: VERSION CONTROL

Each class and document has a version control attached. The version numbers begin at 1.000. As the class or document is changed the last two decimals are incremented as 1.001, 1.002, 1.003, etc. But there is no effort to maintain consistent version numbers between the different documents and classes. For example, version 1.010 of a class may be described in version 1.025 of the Users Guide.

When a new version is released, the latest JNW version number is applied.

A general description of all the major changes will be found on the JavaDoc overview page.

APPENDIX H: DESCRIPTION OF THE TCP IMPLEMENTATION

Appendix H describes the implementation of TCP within JNW.

SYSTEM DESCRIPTION

The TCP implementation consists of several software components:

1. TCP utilities (ModelTransport.TCPUtilities) to break a message into segments of maximum length maximumTCPPayloadInBytes, add TCP headers, and compute the TCP header checksum.

2. A class ModelTransport.TCPConnection and an instance tcpConnections, on each ModelPhysical.Node, which holds the one allowed TCPConnection per node (NOTE: this class extends TCPUtilities, which extends TCPState; thus, it can refer to variables in TCPUtilities and TCPState)

3. A method, GUIActions. ReadEMailTextFilesAndPutIntoSourceNodeBufferAction. CreateTCPConection to instantiate the connections if they do not already exist,

4. A method ModelTransport.TCPUtilities. PrependTCPHeadersToData, which adds the TCP headers to the message segments

5. An array saved_segments in TCPState describing the messages sent but not yet acknowledged,

6. A method, ModelTransport.TCPUtilities.ComposeACKPacket, which composes the ACK message for every TCP packet successfully received,

7. A TCPDestinationBuffer for the destination that
    a. stores the received TCP packets by sequence number,
    b. deletes duplicate packets, and
    c. composes the complete message.

The user sets the TCP transmission mode by choosing the TCP transmission protocol in the input text file as:

```
FROM: 0
TO:6
PROTOCOL:TCP
```

```
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
character count  319
MESSSAGE-END:


FROM: 6
TO:0
PROTOCOL:UDP
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
character count  319
MESSSAGE-END:
```

The PROTOCOL: field determines the transmission protocol. Only UDP and TCP are permitted. Any other protocol will result in an error message. Both TCP and UDP messages can be sent.

When the method GUIActions.
ReadEMailTextFilesAndPutIntoSourceNodeBufferAction reads the input emails, it converts the emails to JNWBitSets and passes the bit sets to ModelTransport.TCPUtilities.

APPENDIX I: TOPOLOGIES AND MESSAGES TO CUT AND PASTE

Some email messages and topologies are provided in Data_For_JNWS_Students. This appendix has more such data that you can cut and paste to build files.

The topology below defines Figure 8.8 of "Understanding Internet Protocols Through Hands-On Programming," J. Mark Pullen, Wiley Computer Publishing, 2000.

```
// FORMAT: SOURCE NODE NUMBER, DESTINATION NODE NUMBER, CAPACITY IN KBPS
// FORMAT: INTERFACE TYPE, BIT ERROR RATE, TRANSMISSION DELAY

// Figure 8.8 of Pullen
// The delay time is 10 times the number in Figure 8.8 of Pullen

// Network type is a router network
   NETWORK-TYPE:Router_Network


//Duplex link between nodes 1 and 2, by the numbering in the book
0,1,10000,Ethernet, 0.000000001 , 60

//Duplex link between nodes 1 and 3, by the numbering in the book
0,2,10000,Ethernet, 0.000000001 , 90

//Duplex link between nodes 1 and 4, by the numbering in the book
0,3,10000,Ethernet, 0.000000001 , 20

//Duplex link between nodes 2 and 5, by the numbering in the book
1,4,10000,Ethernet, 0.000000001 , 30

//Duplex link between nodes 3 and 6, by the numbering in the book
2,5,10000,Ethernet, 0.000000001 , 20

//Duplex link between nodes 4 and 5, by the numbering in the book
3,4,10000,Ethernet, 0.000000001 , 30

//Duplex link between nodes 4 and 6, by the numbering in the book
3,5,10000,Ethernet, 0.000000001 , 40

//Duplex link between nodes 5 and 7, by the numbering in the book
4,6,10000,Ethernet, 0.000000001 , 40

//Duplex link between nodes 6 and 7, by the numbering in the book
5,6,10000,Ethernet, 0.000000001 , 50
```

The file below defines two TCP messages between nodes 0 and 1.

```
FROM:0
TO:1
PROTOCOL:TCP
/FROM:0
/TO:1
/PROTOCOL:TCP
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
```

```
Now is the time for all good men
 to come to the aid of their country.
Characters:  0400
MESSSAGE-END:


FROM:1
TO:0
PROTOCOL:TCP
/FROM:1
/TO:0
/PROTOCOL:TCP
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Characters:  0400
MESSSAGE-END:
```

You can replace "TCP" with "UDP" to pass UDP messages.

The file below passes one TCP message between every pair of nodes in Figure 8.8 of Pullen. You can replace "TCP" with "UDP" to pass UDP messages.

```
FROM:0
TO:1
PROTOCOL:TCP
/FROM:0
/TO:1
/PROTOCOL:TCP
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Characters:  0400
MESSSAGE-END:


FROM:0
TO:2
PROTOCOL:TCP
/FROM:0
/TO:2
/PROTOCOL:TCP
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
```

Now is the time for all good men
 to come to the aid of their country.
Characters: 0400
MESSSAGE-END:

FROM:0
TO:3
PROTOCOL:TCP
/FROM:0
/TO:3
/PROTOCOL:TCP
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Characters: 0400
MESSSAGE-END:

FROM:0
TO:4
PROTOCOL:TCP
/FROM:0
/TO:4
/PROTOCOL:TCP
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Characters: 0400
MESSSAGE-END:

FROM:0
TO:5
PROTOCOL:TCP
/FROM:0
/TO:5
/PROTOCOL:TCP
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Characters: 0400
MESSSAGE-END:

FROM:0
TO:6
PROTOCOL:TCP
/FROM:0
/TO:6
/PROTOCOL:TCP
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Characters: 0400
MESSSAGE-END:

FROM:1
TO:0
PROTOCOL:TCP
/FROM:1
/TO:0
/PROTOCOL:TCP
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Characters: 0400
MESSSAGE-END:

FROM:1
TO:2
PROTOCOL:TCP
/FROM:1
/TO:2
/PROTOCOL:TCP
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Characters: 0400
MESSSAGE-END:

FROM:1
TO:3
PROTOCOL:TCP
/FROM:1
/TO:3
/PROTOCOL:TCP

Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Characters: 0400
MESSSAGE-END:

FROM:1
TO:4
PROTOCOL:TCP
/FROM:1
/TO:4
/PROTOCOL:TCP
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Characters: 0400
MESSSAGE-END:

FROM:1
TO:5
PROTOCOL:TCP
/FROM:1
/TO:5
/PROTOCOL:TCP
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Characters: 0400
MESSSAGE-END:

FROM:1
TO:6
PROTOCOL:TCP
/FROM:1
/TO:6
/PROTOCOL:TCP
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.

Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Characters: 0400
MESSSAGE-END:

FROM:2
TO:1
PROTOCOL:TCP
/FROM:2
/TO:1
/PROTOCOL:TCP
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Characters: 0400
MESSSAGE-END:

FROM:2
TO:0
PROTOCOL:TCP
/FROM:2
/TO:2
/PROTOCOL:TCP
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Characters: 0400
MESSSAGE-END:

FROM:2
TO:3
PROTOCOL:TCP
/FROM:2
/TO:3
/PROTOCOL:TCP
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Characters: 0400
MESSSAGE-END:

FROM:2
TO:4
PROTOCOL:TCP
/FROM:2
/TO:4
/PROTOCOL:TCP
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Characters: 0400
MESSSAGE-END:

FROM:2
TO:5
PROTOCOL:TCP
/FROM:2
/TO:5
/PROTOCOL:TCP
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Characters: 0400
MESSSAGE-END:

FROM:2
TO:6
PROTOCOL:TCP
/FROM:2
/TO:6
/PROTOCOL:TCP
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Characters: 0400
MESSSAGE-END:

FROM:3
TO:1
PROTOCOL:TCP
/FROM:3

/TO:1
/PROTOCOL:TCP
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Characters: 0400
MESSSAGE-END:

FROM:3
TO:2
PROTOCOL:TCP
/FROM:3
/TO:2
/PROTOCOL:TCP
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Characters: 0400
MESSSAGE-END:

FROM:3
TO:0
PROTOCOL:TCP
/FROM:3
/TO:0
/PROTOCOL:TCP
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Characters: 0400
MESSSAGE-END:

FROM:3
TO:4
PROTOCOL:TCP
/FROM:3
/TO:4
/PROTOCOL:TCP
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.

Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Characters: 0400
MESSSAGE-END:

FROM:3
TO:5
PROTOCOL:TCP
/FROM:3
/TO:5
/PROTOCOL:TCP
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Characters: 0400
MESSSAGE-END:

FROM:3
TO:6
PROTOCOL:TCP
/FROM:3
/TO:6
/PROTOCOL:TCP
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Characters: 0400
MESSSAGE-END:

FROM:4
TO:1
PROTOCOL:TCP
/FROM:4
/TO:1
/PROTOCOL:TCP
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.

Characters: 0400
MESSSAGE-END:

FROM:4
TO:2
PROTOCOL:TCP
/FROM:4
/TO:2
/PROTOCOL:TCP
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Characters: 0400
MESSSAGE-END:

FROM:4
TO:3
PROTOCOL:TCP
/FROM:4
/TO:3
/PROTOCOL:TCP
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Characters: 0400
MESSSAGE-END:

FROM:4
TO:0
PROTOCOL:TCP
/FROM:4
/TO:0
/PROTOCOL:TCP
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Characters: 0400
MESSSAGE-END:

FROM:4
TO:5

PROTOCOL:TCP
/FROM:4
/TO:5
/PROTOCOL:TCP
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Characters: 0400
MESSSAGE-END:

FROM:4
TO:6
PROTOCOL:TCP
/FROM:4
/TO:6
/PROTOCOL:TCP
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Characters: 0400
MESSSAGE-END:

FROM:5
TO:1
PROTOCOL:TCP
/FROM:5
/TO:1
/PROTOCOL:TCP
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Characters: 0400
MESSSAGE-END:

FROM:5
TO:2
PROTOCOL:TCP
/FROM:5
/TO:2
/PROTOCOL:TCP
Now is the time for all good men
 to come to the aid of their country.

Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Characters: 0400
MESSSAGE-END:

FROM:5
TO:3
PROTOCOL:TCP
/FROM:5
/TO:3
/PROTOCOL:TCP
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Characters: 0400
MESSSAGE-END:

FROM:5
TO:4
PROTOCOL:TCP
/FROM:5
/TO:4
/PROTOCOL:TCP
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Characters: 0400
MESSSAGE-END:

FROM:5
TO:0
PROTOCOL:TCP
/FROM:5
/TO:0
/PROTOCOL:TCP
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.

Now is the time for all good men
 to come to the aid of their country.
Characters: 0400
MESSSAGE-END:

FROM:5
TO:6
PROTOCOL:TCP
/FROM:5
/TO:6
/PROTOCOL:TCP
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Characters: 0400
MESSSAGE-END:

FROM:6
TO:1
PROTOCOL:TCP
/FROM:6
/TO:1
/PROTOCOL:TCP
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Characters: 0400
MESSSAGE-END:

FROM:6
TO:2
PROTOCOL:TCP
/FROM:6
/TO:2
/PROTOCOL:TCP
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Characters: 0400
MESSSAGE-END:

FROM:6
TO:3
PROTOCOL:TCP
/FROM:6
/TO:3
/PROTOCOL:TCP
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Characters: 0400
MESSSAGE-END:

FROM:6
TO:4
PROTOCOL:TCP
/FROM:6
/TO:4
/PROTOCOL:TCP
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Characters: 0400
MESSSAGE-END:

FROM:6
TO:5
PROTOCOL:TCP
/FROM:6
/TO:5
/PROTOCOL:TCP
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Characters: 0400
MESSSAGE-END:

FROM:6
TO:0
PROTOCOL:TCP
/FROM:6
/TO:0
/PROTOCOL:TCP

Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Now is the time for all good men
 to come to the aid of their country.
Characters: 0400
MESSSAGE-END: