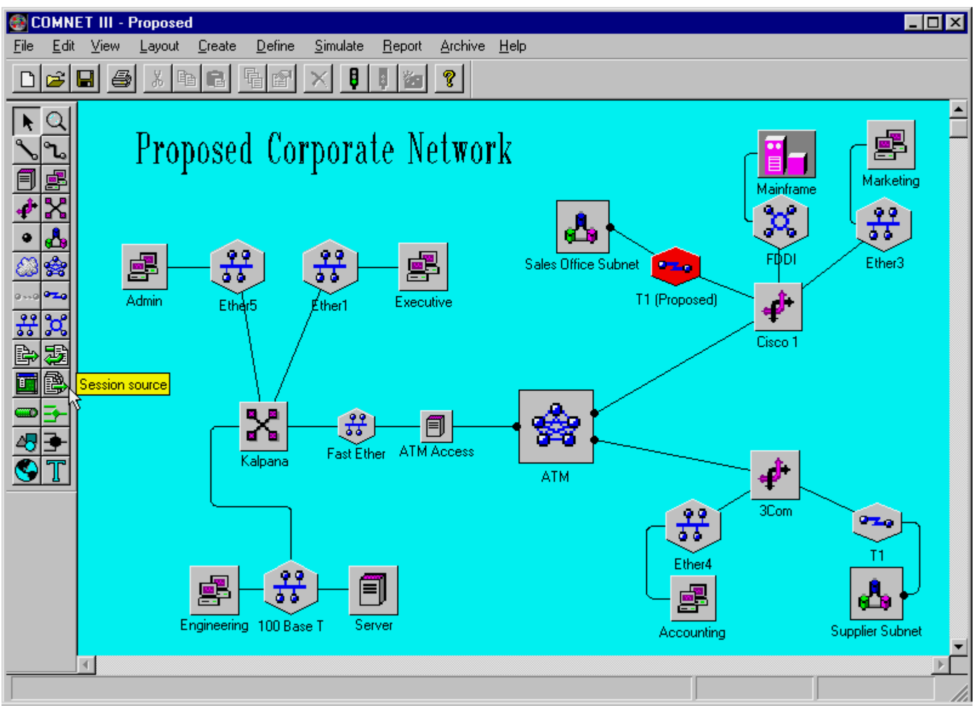
COMNET III™ Tutorial

A Detailed Guide to Modeling Networks with COMNET III



CACI Products Company

COMNET III™ Tutorial

A Detailed Guide to Modeling Networks with COMNET III

CACI

Call of Fax for an Immediate Response

Worldwide
CACI Products
3333 N Torrey Pines Ct
Third Floor
La Jolla, California
92037 USA
Tel (619) 457-9681
Fax (619) 457-1184

Wash., DC Area
CACI Products
1600 Wilson Blvd
Thirteenth Floor
Arlington, Virginia
22209 USA
Tel (703) 875-2900
Fax (703) 875-2904

UK
CACI Products
Watchmoor Park
Riverside Way
Camberley, Surrey
GU15 3YL UK
Tel +44 (0) 1276 671 671
Fax +44 (0) 1276 670 677

For product information contact:

CACI Products Company	CACI Products Division
3333 North Torrey Pines Ct	Watchmoor Park, Riverside Way
La Jolla, California 92037	Camberley, Surrey, GU15 3YL, UK
Telephone: (619) 457-9681	Telephone: +44 (0) 1276 671 671
Fax: (619) 457-1184	Fax: +44 (0) 1276 670 677

This manual was originally written by Jeffrey Edward Sullivan while a student at the Naval Post Graduate School. The author has granted CACI Products Company the right to update, modify and distribute the manual. CACI Products Company has modified the original manual in order to update it with COMNET III version 1.3. If you desire an original manual, please write to CACI Products Company which will provide a mailing address to reach the author.

ACKNOWLEDGMENT

The author would like to express his appreciation to Professor Sridhar and Professor Boger for their support and guidance. In addition the author would like to thank Terry Williams, James Schmidt, Milena Cochran, and Don McGreggor for there assistance in the collection of data used for this study and for their patience in answering the author’s endless stream of questions.

ABSTRACT

This thesis provides a tutorial to explain the theory used in the application for the modeling and simulation of networks. Each chapter presents the theory of several objects which may be used in the application, states a network problem which is to be analyzed, provides step-by-step instructions to build a model to analyze the network problem, and presents the results of the network simulation.

TABLE OF CONTENTS

I. INTRODUCTION

A. THE NEED FOR NETWORK SIMULATION

Network simulation can be a valuable tool as it provides a manner of modeling a network to determine its performance characteristics. Often, due to the critical nature of a network, the ability to disconnect the network for testing and evaluation or upgrades is not an option or must be scheduled during periods of minimal usage. Network simulation provides a means of testing proposed changes prior to placing them into effect, performing what-if analysis concerning the reliability of key components in a network and the effects of losing a component, planning for future growth, and initial design of a proposed network. The costs associated with the building and operating of a network make simulation a viable option in making choices in the building, modification, and performance analysis of a network.

B. THESIS SCOPE

There are many commercial off-the-shelf network simulation applications available on the market today. The focus of this thesis is on one of these applications. COMNET III release 1.1n is a network simulation tool develop by CACI Products Inc. which may be used in the simulation of both voice and data networks. The scope of this thesis will be the development of a tutorial which deals only in the area of using this application in the modeling of data networks. Specific areas which will be looked at are the modeling of local area networks and packet-switched wide area networks, and the methods of generating traffic in this application.

The objective of the tutorial is to use a building block method where each chapter of the tutorial focuses on a particular aspect or modeling construct which the application provides. Each chapter in the tutorial will use the following format:

1. The objective of the chapter will be stated.
2. COMNET III constructs and theory which will be used in the chapter will be presented.

3. A network problem description will then be stated.
4. Analysis which was performed in determining characteristics of network equipment or traffic sources will be covered.
5. Step-by-step instructions for building a model to analyze the problem statement will be given.

C. COMNET III OVERVIEW

COMNET III is a commercial off-the-shelf application whose function is to allow users to estimate the performance characteristics of computer based networks. A network description is created graphically using a window interface, and no actual programming is required of the user. The application was formulated primarily for the modeling of both Wide Area Networks (WANs) and Local Area Networks (LANs). The recommended usage of the application include:

- Peak loading studies
- Network sizing at the design stage
- Resilience and contingency planning
- Introduction of new users/applications
- Evaluating performance improvement options
- Evaluating grade of service contracts.

The COMNET III application was written in the programming language MODSIM II using an object-oriented design. As such, objects are created within the application which represent various pieces of hardware that may be found in a network. In writing the program using an object-oriented framework, creating representations of all types of equipment which could be present in a network would be very difficult. The program instead was written with several objects or basic building blocks whose characteristics may be edited to match those of equipment found in a real world network. These building blocks which include computer and communication nodes, router nodes, ATM nodes, and links may all be edited to define the characteristics of the piece of hardware which is desired to be modeled.

The basic steps to build a model using the application are to first define a network topology using the various nodes and links available in the application. The traffic loading and computer workload are then established by creating application sources, traffic sources, or by the direct input of traffic gathered from an actual network using a protocol analyzer. The model is then verified for its correctness and a simulation may be run. At the completion of a simulation, reports are generated which describe the performance of the modeled network.

II. COMNET III BASICS

A. INTRODUCTION

The purpose of this chapter is to provide an introduction to the COMNET III graphical user interface and the methods for creating, editing, and manipulating objects using this interface. Also, aspects concerning the running of a simulation and the reports and statistics which may be obtained from running a network simulation will be covered.

B. COMNET III GRAPHICAL USER INTERFACE

COMNET III uses a standard Windows[tm] interface for the creation of networks models. Figure 2.1 displays the view which appears when initially starting the application. The *standard menu format* of most windowed applications across the top of the window activates pull down menus. The *toolbar* to the left hand side of the screen allows for a simple method of creating objects in a model. The area to the right of the

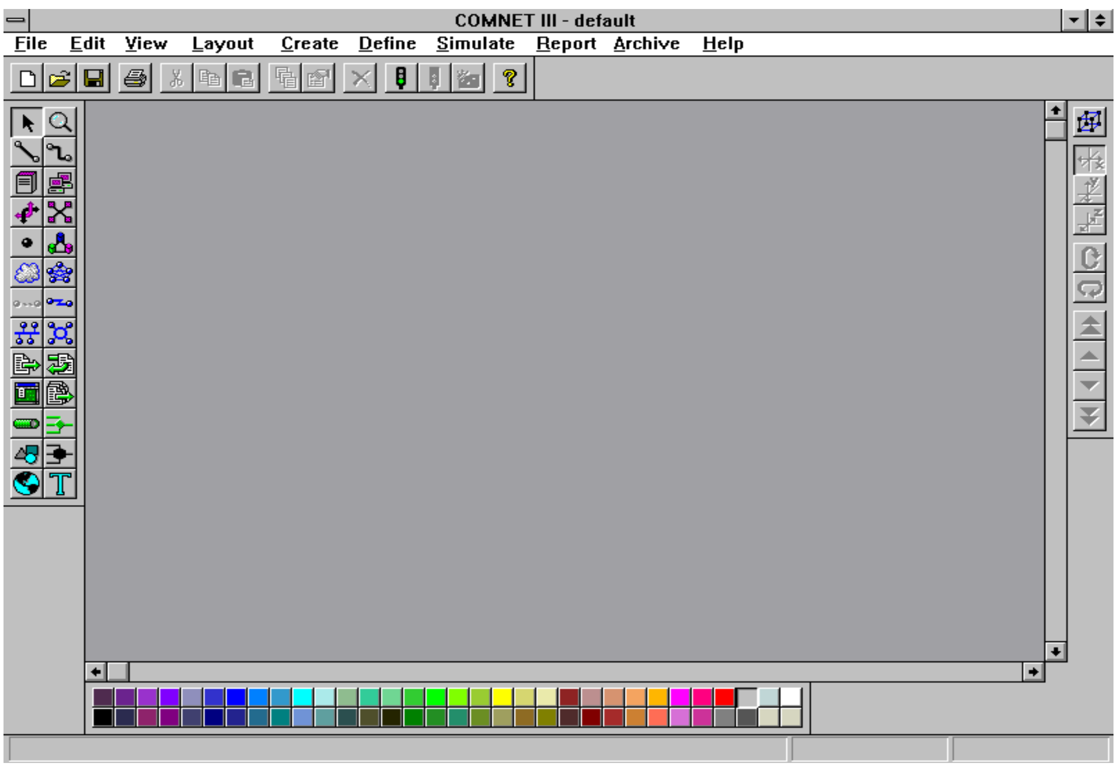


Figure 2.1 View of COMNET III Graphical User Interface

toolbar is the work area where models are built. The small strip below the toolbar and the workspace is used to display *messages* to the user concerning actions performed while using the application.

C. COMNET III MENUS

The menu bar across the top of the window activates drop down lists which present the options available for a menu option. The following subsections present brief descriptions of each menu followed by a list of commands available in the menu option which may be used for the manipulation of objects within a model.

1. File Menu

This menu is used primarily for the manipulation of files which may be created and imported within the application. The menu options are as follows
[Ref. 2 p. 210, 211]:

- ***New***: Clears the current model and creates a new clean workspace for the creation of a new model. If a model is in the workspace, the application prompts the user to cancel this operation before it is carried out.
- ***Open***: Used to open an existing model. All COMNET III model file have a filename followed by a **.c3** extension.
- ***Save***: Used to save the current model.
- ***Save As***: Saves the current model to the path and directory specified by the user.
- ***Import***: Allows importing external data into COMNET III.
    - ***External Model (*.c3e)***. A model file created outside of the COMNET III design process. The external model file must conform to the specifications defined in the COMNET III API External Model File Format document. This document can be obtained from CACI Products Company by request.

- ***NMS topology (*.top)***. Currently, the application allows importing network topology files from HP OpenView, Cabletron Spectrum, Digital Equipment Corporation, and IBM Netview 6000.
- ***External Traffic***. Currently the application allows importing of trace files gathered from various network analyzers. The supported analyzers include Network General Corporations Sniffer, HP's Netmetrix, and Wandel & Goltermann's Domino.
- ***Bitmaps***. This feature allows you to add a bitmap image to be mapped onto an object to the existing image library.

- ***Export***: Used primarily to export simulation statistics from the COMNET III format to a tab-delimited format which can be recognized by most spreadsheet applications. When the simulation statistics option from the export option is selected a tab-delimited file called **statfile.xpt** is created.
- ***Print***: Prints a picture of the current model layout.
- ***Exit***: Exits the application.

2. Edit Menu

The options available from this menu are used for the manipulation of objects created in a model and are as follows [Ref. 2 p. 211, 212]:

- ***Cut***: Deletes the selected object and places it on the clipboard.
- ***Copy***: Places a copy of the selected object on the clipboard so that it can be used later in a ***Paste*** operation.
- ***Paste***: Places a copy of an object which has been placed on the clipboard in the location specified by the user.
- ***Copy Paste***: The same function as a ***Copy*** followed by a ***Paste*** operation.
- ***Clear***: Deletes the selected object.
- ***Clone***: Provides a method of creating duplicate copies of an object. When selected a window will appear as shown in Figure 2.2. Enter the number of clones desired and the X and Y offset and press the *OK* button. Duplicates of

the original will then be made positioned according to the X and Y offset from the original.

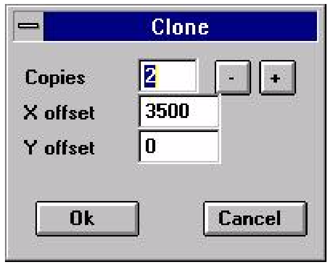


Figure 2.2 Clone Window

- ***Select All***: Selects every object in the workspace.
- ***Scale***: Used to scale the size of all icons selected when this option is chosen. The user may specify the scaling factor based on the needs of the model.
- ***Move***: Allows several objects which have been selected to be moved to a new position in the work area while maintaining there relative position to each other.
- ***Detail***: Used to bring up the dialog window which allows editing of an object. Double clicking on the object will bring up the same window.
- ***Parent***: Used to define the routing method of the highest level in a network.
- ***Enter***: Used to display the screen of a sub-network for building the sub-network. Double clicking on the sub-network icon will accomplish the same function.
- ***Leave***: Returns to a the main model layout from a subnetwork or a WAN cloud object.

3. View Menu

The view menu is primarily used to edit the look and size of the work area. The options which are available include [Ref. 2 p. 212]:

- ***Zoom In***: Allows the user to zoom in on a given selected portion of the model. After selecting this option, single click to the upper left of the area to zoom in on. Then, drag the mouse to define the area. A rectangular box will appear on the screen showing the area which will be zoomed into. When the area is defined, single click again and the screen will zoom into the selected area.

- ***Zoom Out***: Expands the view to take in the entire work area.
- ***Zoom Reset***: Sets the zoom to the size of the original work area.
- ***Set Work Area Size***: The default work area size is a square consisting of the size of the screen of the computer which the application is run on. Selecting this option brings up a window as shown in Figure 2.3 which allows scaling the work area to meet the needs of the model being built. A scale factor of two will double the size of the X and Y axis. The adjust content box, if selected, will stretch the objects to fit the size of the new work space if selected.

Figure 2.3 Work Area Scaling Window

- ***Set Color***: Selection of this option brings up a window such as in Figure 2.4 which allows setting the color of the background and foreground in the workspace. Foreground sets the color of the text in the work area. Background sets the color of the background in the work area. Different color schemes may be used within the same model between the backbone network view and subnetwork views.

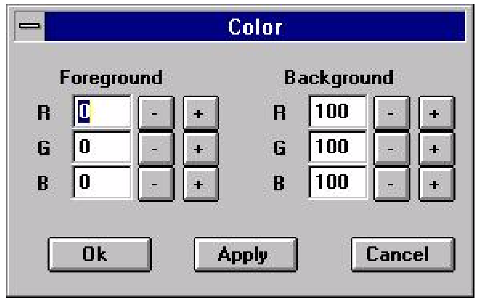


Figure 2.4 Work Area Color Window

- ***Toggle Names***: Turns the names of all icons within the model on or off in the work area.

4. Create Menu

The primary objective of this menu is the creation of objects to build a model. The drop down menu contains a list of all objects. When an object is selected, the outline of the object appears by the mouse cursor. By moving the cursor to the desired position on the workspace and single clicking, the icon of that object appears. All objects which may be created using this menu may also be more easily created using the toolbar which is discussed in Section D of this chapter.

5. Define Menu

This menu is used to define global parameters or characteristics of any object which may be created within the application. The selection of any of the options within this menu brings up a window which allows editing of the parameters for a given type of object. These parameters will then be available to any object for selection when later editing the object.

6. Simulation Menu

The simulation menu provides options which allow setting the parameters which govern the running of a simulation. Aspects of simulation and the options available in this menu are covered in detail in Section I of this chapter.

7. Report Menu

This menu allows the selection of those objects within a model for which statistic are desired to be gathered when running a simulation. The method of selecting the reports chosen and the options available are covered in Section G of this chapter.

8. Archive Menu

COMNET III maintains a list of all objects and parameter sets with there default values. When an object is initially created, these default values are used as the parameters which describe any object, and only the parameters which are maintained in the archive list may be selected. Objects or parameters entered into a model are not automatically archived for use in subsequent models. The archive menu allows the user to manipulate these default values or to enter user-defined parameters for commonly-used objects. Objects and parameters which are explicitly entered via the archive menu will be available to any other model which the user may build.

9. Help Menu

The COMNET III help menu is limited in scope. It contains useful information about each of the menu options, which is useful as an online reference, and information concerning the manipulation and creation of objects. Detailed information on the objects and their parameters is not included within the *Help* menu.

D. COMNET III TOOLBAR

The toolbar, located to the left hand side of the program window, facilitates the creation of the network model topology, traffic sources, and application sources. It offers the same function as the *Create* menu but is often easier to use as it provides a simple one button point and click creation. Figure 2.5 displays the toolbar along with the name of each button on the toolbar.

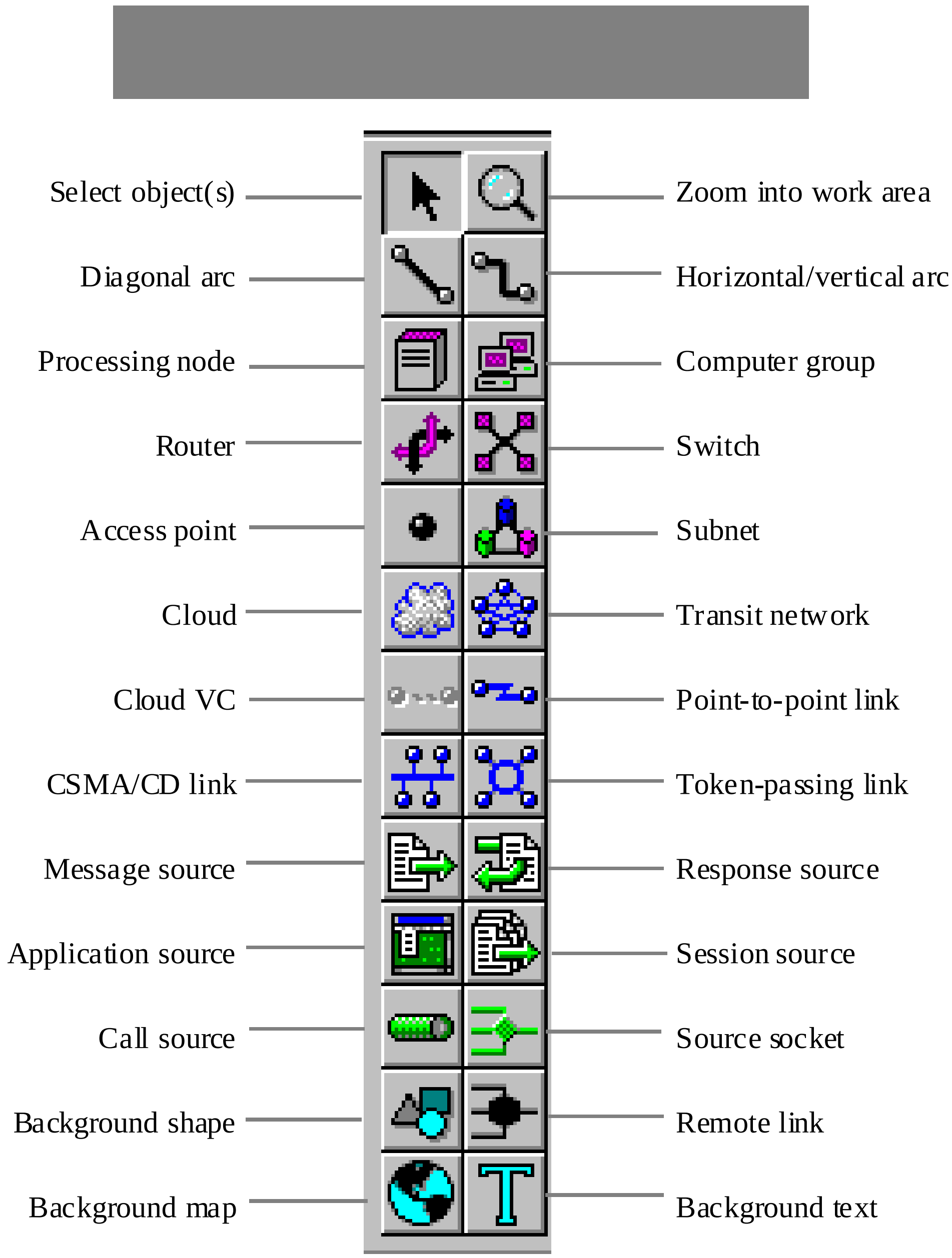


Figure 2.5 COMNET III Toolbar

The toolbar has two modes of operation, standard and extended. Standard mode is the default mode where a single click on a button activates that button. The object may then be placed in the work area by either depressing the mouse button and dragging the object to its desired position or moving the mouse to the desired position and single clicking again. Extended mode is activated by double clicking on a button. The button will then appear to be further depressed to indicate that extended mode is in use. Multiple objects can now be created by positioning the mouse where the object is desired to be placed and single clicking. To place the toolbar back into standard mode when finished creating the multiple objects, simply depress the *Selection Tool* button on the toolbar.

E. TERMINOLOGY

COMNET III uses the following list of terminology as defined in reference to the parameter fields available to define objects within a model [Ref. 2 p. 21]:

- Byte: 8 bits
- Kilobyte (kB): 1024 bytes
- Kilobit (kb): 1000 bits
- Kilobits per second (kbps): 1000 bits per second
- Megabyte (MB): 1024 Kilobytes = 1,048,576 Bytes
- Megabits per second: 1,000,000 bits per second

F. STATISTICAL DISTRIBUTIONS AND TABLES

The major concept which must be understood to model a network within the COMNET III application is that the majority of all the characteristics of any of the nodes, traffic sources, or applications may be described either by a constant or a statistical distribution. The method by which COMNET III picks values from a distribution when running a simulation is based on the generation of random numbers. The application generates random numbers based on a multiplicative congruential pseudo random number generator. A starting seed is used to generate a random number ranging from 0 to 1 and the next seed. The new seed produced is used to generate the next random number and the next seed. To provide multiple independent random number generators, each distribution has the ability to set a stream number. Up to 99 different streams may be used within a model. Each stream has its own starting seed which causes it to produce different random numbers from any other stream. Weighting functions are then used, which manipulate the random numbers pulled from the uniform (0,1) distribution, to generate the desired distribution. The following list displays the distributions which are built into the application and the parameters which are used to set the desired shape of the distribution.

- Beta: Shape 1, Shape 2, Min, Max, Stream
- Erlang: Mean, Shape, Stream
- Exponential: Mean, Stream
- Gamma: Mean, Shape, Stream
- Geometric: Min, Mean, Stream
- Hyperexponential: Mean 1, Mean 2, Prob Mean 1, Stream
- Integer: Min, Max, Stream
- Lognormal: Mean, Standard Deviation, Stream
- Normal: Mean, Standard Deviation, Stream
- Poisson: Mean, Stream
- Triangular: Min, Max, Mode, Stream
- Uniform: Min, Max, Stream
- Weibull: Shape, Scale, Stream

Often times when building a model, the same distribution may be needed to set parameters on different nodes or traffic sources. The application gives the user the option to create user-defined distributions from any of the distributions in the previous list. This is accomplished by selecting the *User Distribution* option from the *Define* menu. A window will then appear as shown in Figure 2.6. Select the *Add* button and a window will appear such as shown in Figure 2.7. A unique name for the distribution must then be entered, and a distribution selected and edited to the desired parameters. After a user-defined distribution is created, it will appear in the list of distributions which may be selected to set the parameters of any object created within the model.

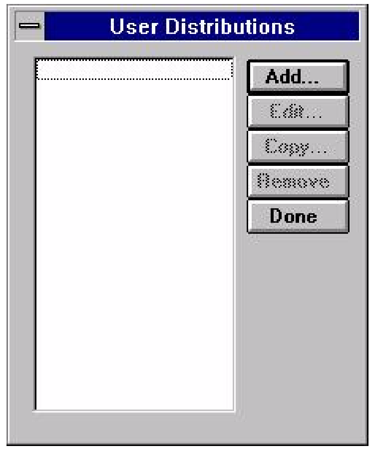


Figure 2.6 User Distribution Window

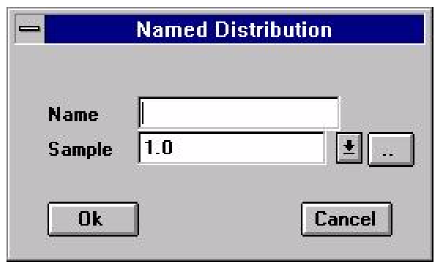


Figure 2.7 User Defined Distribution Window

Some data cannot be described in a useful manner using a statistical distribution, but can be described using a probability table. COMNET III allows creation of these tables through the selection of the *Table* option from the *Define* menu which will open a window as shown in Figure 2.8. The table must be given a unique name, and the type set to either *discrete* or *continuous*. Selecting the *discrete* option implies values entered are taken as observation points whose likelihood's of being chosen are based on the probabilities assigned. The *continuous* option implies the values entered describe the envelope of a curve. Values which will be used when running a simulation are then interpolated from those entered in the table. For both cases, the probability values entered must be in the form of a cumulative distribution function. The last entry in any table must have a probability of 1 assigned. To enter values in a table, first select the cell for the entry to be made by single clicking on it. Then, in the field above the table enter the value which is to be assigned to that cell. Unfortunately the application does not allow entering the probability and its associated value at the same time. Each cell in a table must be edited individually.

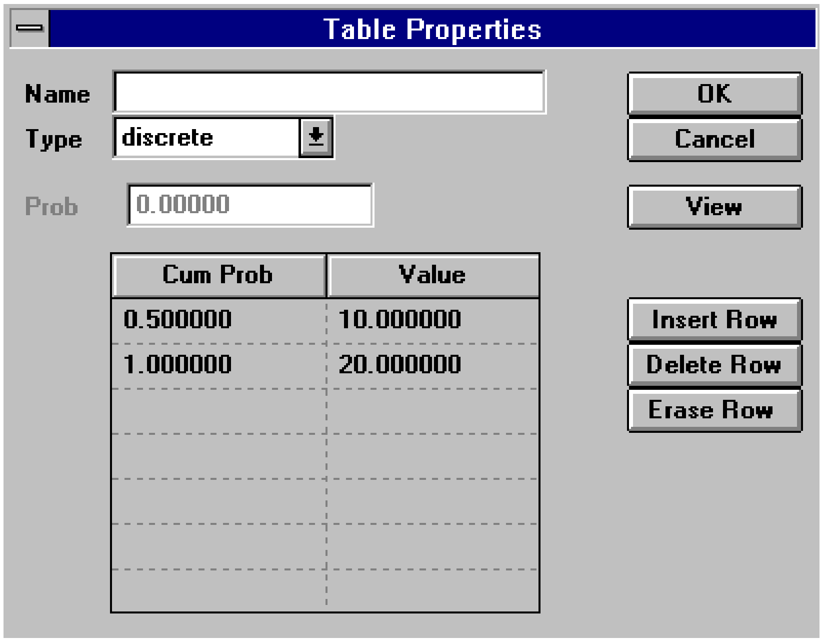


Figure 2.8 User defined Tables Window

G. REPORTS

The main goal of creating any model is the results which are obtained from running a simulation of that model. Reports are produced automatically at the end of running a simulation or if the simulation is halted prior to completion of the simulation run time. Each replication in a simulation will append the results to the bottom of the chosen report. The default report that will be generated is an ASCII file labeled **Stat1.rpt**. Statistics are only gathered for those reports which are selected prior to running the simulation. The number of reports and the objects which are included in these reports will affect the simulation run time. A greater number of reports requires more processing which will increase the actual length of time necessary to run a simulation. Reports are viewed at the end of a simulation by selecting the *Browse Reports* option from the *Report* menu. A window will appear which allows selecting which report to view by entering the replication number.

Reports are selected from the *Define* menu which displays a list of all the types of objects which may appear in a model. By selecting one of these objects a flyaway menu appears which displays the report options available for that object. By selecting a report option a window appears, such as in Figure 2.9, which displays all of the objects within a model for which the report may collect data on. The objects are selected by single clicking on those desired. Those objects selected will be apparent as they become highlighted in blue. If all objects are desired, the *All* box may be checked. The *Show Group Node Detail* box may be checked if it is desired to see a report on each individual node in a computer group. If the box is not selected the computer group is treated as a whole. After all of the objects desired are selected click the *On* box to set the report and press the *OK* button.

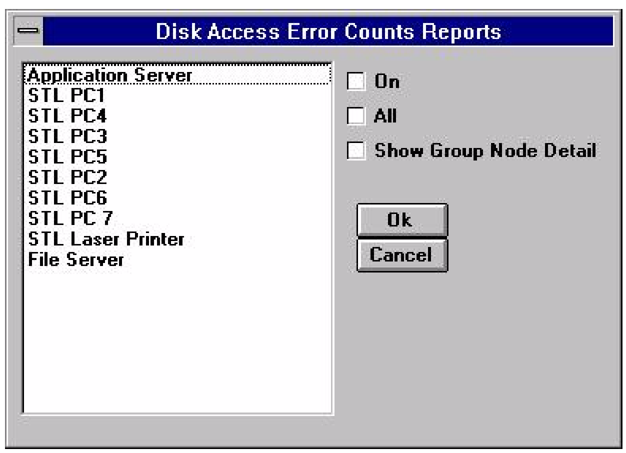


Figure 2.9 Report Window

The following lists the types of reports available within the application which may be used to gather information concerning packet networks by object type:

- ***Node Reports***: Processor and Disk Utilization, Input Buffer Use by Port, Input Buffer Totals, Output Buffer Use by Port, Output Buffer Totals, Received Message Counts, Disk Access Error Counts, and Session Level
- ***Link Reports***: Channel Utilization, Collision Statistics, and Session Level
- ***WAN Cloud Reports***: Frame delay by Virtual Circuit, Frame Counts by Virtual Circuit, and Access Link Statistics

- ***Application Source Reports***: Application Run Length
- ***Message and Response Source Reports***: Message Delay and Packet Delay
- ***Session Source Reports***: Message delay, Packet Delay, Setup delay, Session Length, and Setup Counts
- ***Transport and Command Reports***: Message Delay and Packet Delay
- ***Setup Command Reports***: Message Delay, Packet Delay, Setup Delay, Session Length, and Setup Commands

H. PLOTS AND PERCENTILES

In addition to the reports available within COMNET III, the application also allows gathering of further data on any type of link defined in a model, all types of traffic sources, and WAN cloud virtual circuits and access links. As with any report, the option to collect statistics from any of these objects must be selected prior to running a simulation. Setting the collection of statistics is accomplished by pressing the *Statistics* button at the bottom center of the window shown in Figure 2.10 for a point-to-point link. After pressing the *Statistics* button, a window will appear, such as Figure 2.11, which

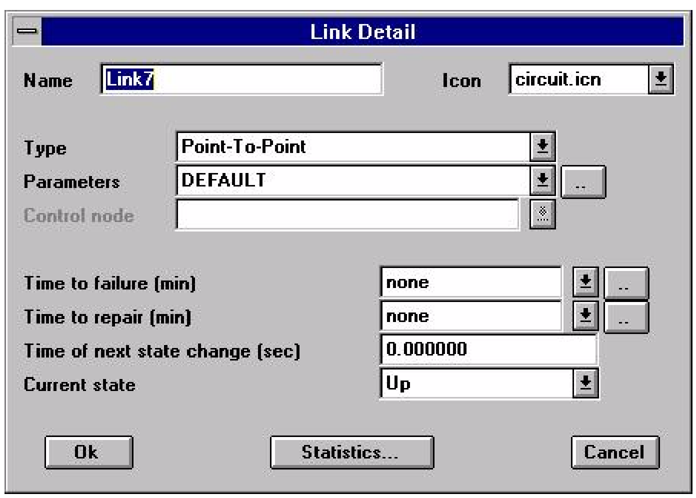


Figure 2.10 Point-to-Point Detail Window Showing Statistics Button

displays the statistics which may be gathered for the object. The options for gathering statistics are selected with a single click on the option which will highlight that option. By then depressing the *Edit* button, a window will appear such as shown in Figure 2.12. Selecting the *Collect stats* and *Save observations* boxes and then pressing the *OK* button will set this option for gathering data during a simulation.

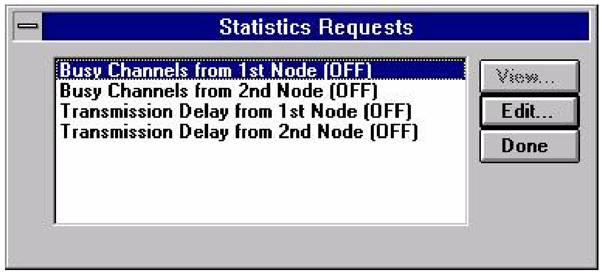


Figure 2.11 Statistics Options Window

Figure 2.12 Editing Statistics Options Window

After the completion of a simulation, the statistics gathered and plots of the data gathered may be viewed. This is accomplished by using the *Statistics* button again to bring up the same window as in Figure 2.11. By depressing the *View* button the statistics gathered will be shown. In certain cases, the option will then be presented to plot the data saved. By selecting this option a window will appear which contains no plots, but has the menu options of *File* and *Plot* at the top. By selecting the *Plot* menu a drop down menu will appear giving the option of plotting smooth data, histograms, or percentiles. After selecting the desired option the window which appears allows setting the parameters for the type of plot selected.

I. RUNNING A SIMULATION

COMNET III uses a discrete event simulation methodology when running the simulation of a network model. This means the application looks for the first event which is to occur in the simulation based on the traffic which is described in the model and executes this event. After completion of this event the simulation then skips to the next event which is to occur. This process repeats until the length of time the simulation is set to run is completed.

The first step in running any simulation is to first verify the network model which has been built. This is accomplished by selecting the *Verify* option from the *Simulate* menu. When this option is selected the application performs a logical check of the model which has been built. If no errors are detected, the message *No verification errors detected* will appear in the message window at the bottom of the screen. If errors are detected, a window will appear displaying all errors in the model which must be corrected before a simulation may be run.

The next step in running a simulation is selecting the parameters which will govern how the simulation is to be run. These parameters are set by using the *Run Parameters* option from the *Simulate* menu which will bring up a window as shown in Figure 2.13. The main parameters which are entered via this window are the replication length, the warmup length, and the number of replications. Replication length is the length of time in seconds that a simulation will run. Warmup length is used to specify the time in the simulation when the application will begin to collect data. This feature is important as the statistics gathered are based on the entire simulation time after the warmup length. At the initiation of a simulation, traffic levels are often low and are building up to the steady state level. If a warmup length is not used, the results of reports and statistics collected may be erroneously low. The number of replications fields sets the

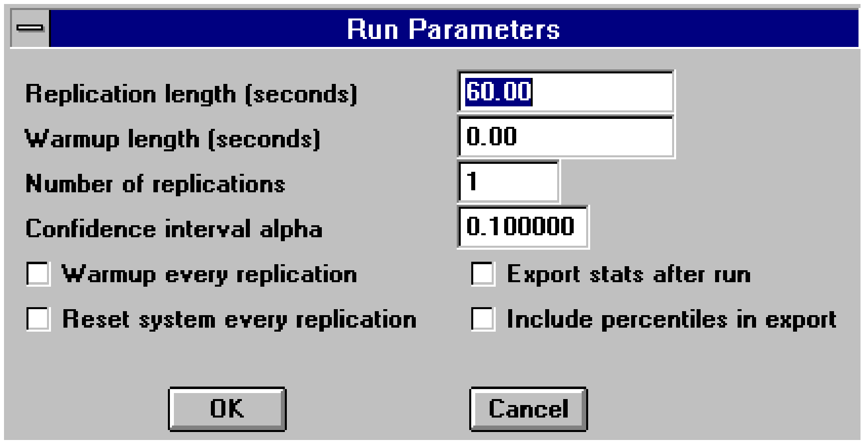


Figure 2.13 Simulation Run Parameters Window

number of replications for a simulation. This can be used to get multiple shorter runs for the collection of data. The *Warmup every replication* option tells the program to wait the warmup length for each replication before collecting data. If a warmup length has been specified, but this box is not set, only the first replication will use the warmup length. The *Reset system every replication* box, if set, will clear all traffic from the previous replication prior to starting the next. If it is not set, all traffic from the previous replication is saved and the next replication begins where the previous left off. The *Export Stats after run*, if set, will output the statistics to an ascii file. The *Include percentiles in export*, if set, will include the percentile figures for the simulation in the statistics export.

The application also gives the option to view the movement of packets on screen during a simulation. This option is set via the *Animate* option on the *Simulate* menu which brings up a window as shown in Figure 2.14. The *Animate* box, if checked, will allow the packets to be viewed as they traverse the network.. This option is useful when initially running a simulation as a visual check to see if the network is performing as expected. However, the *Animate* option greatly increases the actual simulation run length and is not recommended for use when simulation runs are set for gathering data. The *Next on/off time* field allows toggling the animation to either on or off at the simulation time set in this field. The *Step size* field sets the speed at which packets will traverse the network topology. The slowest setting is 10, and the fastest is 1000. Any integer value between these numbers may be entered. The animate option may be toggled on or off at any point when running a simulation.

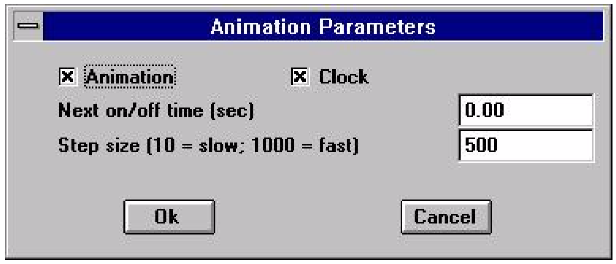


Figure 2.14 Simulation Animation Parameters Window

After the model has been verified, run parameters set, and animate options set, the simulation may be run by selecting the *Start Simulation* option on the *Simulate* menu. The program will then perform another verification of the model and prompt the user to save the model if a the model has not been immediately saved prior to starting the simulation. The simulation will run for the length of time specified. Upon completion of the simulation, the application will generate any reports which were specified. A simulation may be stopped at any time when running by selecting the *Halt Simulation* option. When selected, the simulation will stop and a report will be generated for all data gathered up to the point of stopping.

The program also includes two options which may be used to troubleshoot the running of a simulation. The first is the trace option which is invoked by selecting the *Trace* option from the *Simulate* menu. This brings up a window as shown in Figure 2.15. The trace option allows the user to record messages either to a file or to the screen. Either option will cause the step-by-step logic followed by the application when running a simulation to be recorded which is useful in troubleshooting traffic and application sources. If the trace is set to appear on the screen, the user is given two separate options. The first option is for display of the message to the screen to appear in a single step fashion where the next step will not occur until prompted by the user. The second option is to set an amount of time the message will appear on the screen before the application goes on to the next step. If the trace to file option is set, a text file is created that records each action performed in the simulation. This file may be viewed after completion of the simulation.

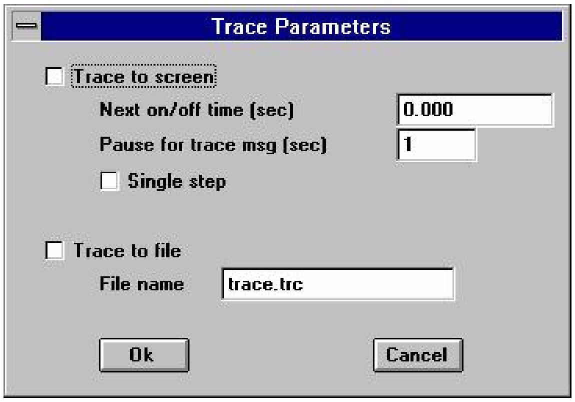


Figure 2.15 Simulation Trace Parameters Window

III. MESSAGE TRAFFIC GENERATION

A. INTRODUCTION

The goal of this chapter is to describe the methods available in the COMNET III application to build simple traffic sources and the characteristics which are used to describe these sources. The other major goals are to introduce the steps required to build a simple model of a local area network and to familiarize the user with running a simulation.

B. MESSAGE GENERATING SOURCES

COMNET III offers several methods for modeling message traffic in a simulation. One method is through the building of application sources using global and local commands which will be covered in detail in Chapter V. The other method is through the use of traffic sources. Traffic sources are simplifications of application commands built directly into COMNET III for ease of generating message traffic. The common thread in defining all traffic sources is having the ability to describe the characteristics of the traffic being generated by using statistical distributions. The three types of traffic generating sources within COMNET III are message sources, response sources, and session sources.

1. Message Source

The message source is a message generator which is capable of sending messages to one or more destinations. It is useful at modeling many forms of data transport including file transfers and e-mail. A message source may be created using either the toolbar or by using the *Create* menu. Editing of a message source by double clicking on the icon will cause the message source window shown in Figure 3.1 to appear. The use of the fields to describe the characteristics of the message source are discussed in Section C.

2. Response Source

The response source is a message generator used to send message replies upon receipt of a message. This source is useful in modeling database queries, e-mail replies,

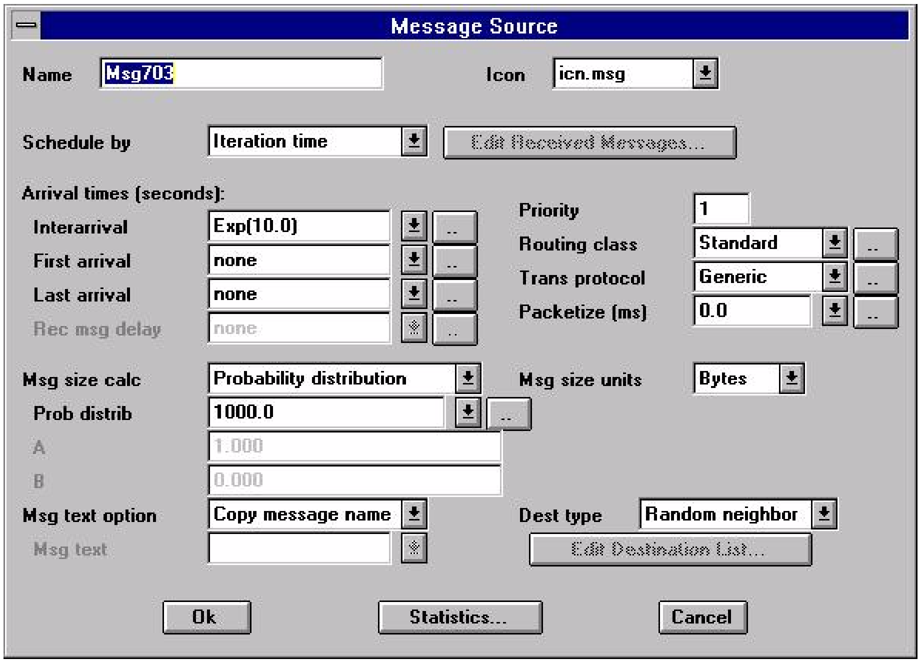


Figure 3.1 Message Source Window

and any type of message traffic which would be triggered by the receipt of a message. The message which is generated by a response source is always sent to the node which generated the message which triggered the response source. Response sources may be created by either using the toolbar, or the *Create* menu. Editing of a response source by double clicking on the icon will cause the message source window shown in Figure 3.2 to appear. The use of the fields to describe the characteristics of the message source are discussed in Section C.

3. Session Source

The session source is message generator which first sets up a session with another node, and then sends the message traffic. It is useful at modeling message sources which have a bursty message arrival process as several messages may be transmitted within one session. The session source is also used to model connection-oriented traffic. Session sources may be created by either using the toolbar, or the *Create* menu. Editing of a session source by double clicking on the icon will cause the message source window shown in Figure 3.3 to appear. The use of the fields other than those which apply

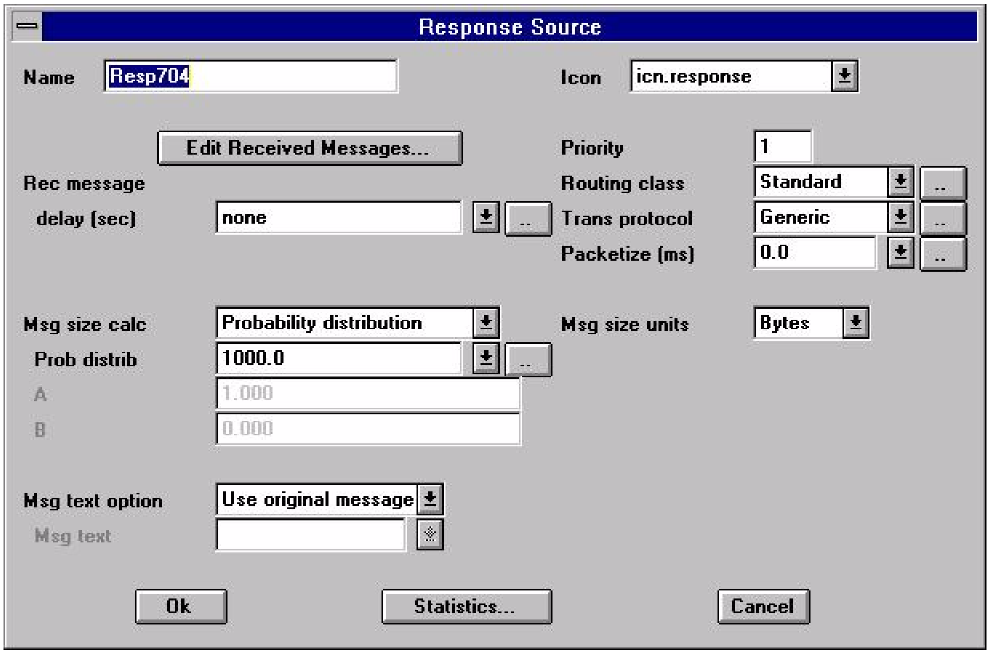


Figure 3.2 Response Source Window

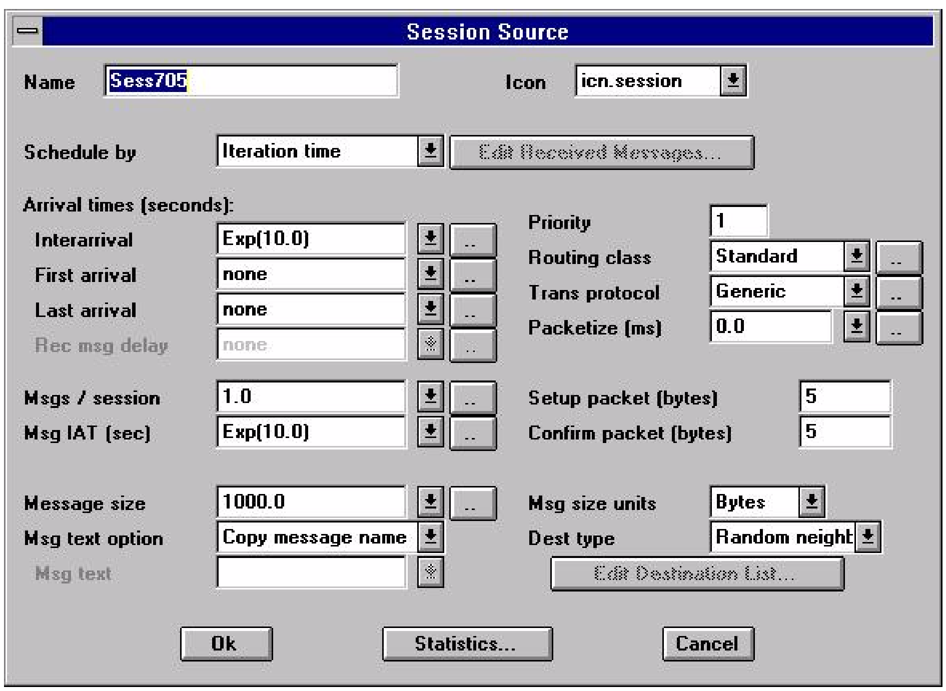


Figure 3.3 Session Source Window

specifically to the session source are discussed in Section C. The session source has four fields which may be used to describe the characteristics of the source which other traffic sources do not have. Explanations of these unique fields are as follows:

- Messages/session: Specifies the number of messages which will be transmitted during one session. This field may be set to a constant or described by a statistical distribution.
- Message IAT: Specifies the interarrival time between messages within one session. This field may be set to a constant or described by a statistical distribution.
- Setup packet: Specifies the number of bytes in the setup packet.
- Confirm packet: Specifies the number of bytes in the session confirmation packet.

C. MESSAGE CHARACTERISTICS

The characteristics or parameters of all message sources within COMNET III are basically the same. There is some variation based upon the type of message source being used. The common features of all sources include specification of a unique name to the source; scheduling method, message priority, routing class, selection of a transport protocol, setting of a packetizing delay, selection of message size and text, and the choice of the traffic destination.

1. Message Name

The message name is a unique identifier given to the source for identification purposes. This is also the name which will appear in the workspace when building a model.

2. Message Scheduling

The two methods available for the scheduling of message or session sources of traffic are by *iteration time* or by *received message*. Response sources may only be scheduled by the receipt of a received message. Scheduling message traffic by iteration time means an interarrival time is used to determine the time of the start of the creation of one message to the start of the creation of the next message. The interarrival time may be set to a fixed value, a user-defined distribution, or any of the distributions supported in COMNET III. The option is also given, when iteration time is specified, to set the simulation time when the first and last messages are sent. The first and last arrival may be specified in the same manner that the interarrival time may be specified. Received message scheduling implies that the traffic source will only become active if triggered by the receipt of a message of the required text. When received message scheduling is specified, a received message list must be created by the user to trigger the traffic source. This is done by pressing the *Edit Received Message* button on any of the traffic sources. A list of all messages defined in the model will appear. By selecting the messages which are desired to trigger the source, a list is generated. Received message scheduling also allows the option of specifying a delay time which occurs before the response is sent. This delay may be modeled by setting a specified time or any statistical distribution supported by the application.

3. Message Priority

The priority field is used to specify the order of packets in the input and output buffers of the various nodes. The priority field may be set to an integer value between 1 and 99 where larger values have priority over smaller values. Packets with a higher priority are placed at the head of the buffer's queue. Packets of the same priority are placed in the queue based on a first-in-first-out (FIFO) algorithm. The priority field does not imply priority between several traffic generating sources on the same node [Ref. 2 p. 112]. The scheduling of different sources on the same node which are in contention is done similar to a round-robin fashion.

4. Message Routing Class

The routing class is a label which can be applied to any traffic source. A routing class called standard is included as a part of any model made and is the default setting unless changed. The routing is an additional description of the message which adds parameters which affect the routing of a message when either IGRP, minimum penalty, or a user-defined routing tables are specified for a model. The use of routing classes will be discussed more in Chapter VIII's discussion of routing message across a wide area network.

5. Message Transport Protocol

The transport protocol defines the characteristics of data transfer from the origin of the message to its destination across the network. Different protocols may be applied to different traffic sources within a model. Currently, COMNET III has protocol parameters for ATP, TCP/IP - Microsoft, TCP/IP - Sun, UDP/IP - Sun, NCP/IPX, and NCP/IPX Burst Mode built into the library for use in a model. The program also allows the user to define a protocol for use in a model . This may be performed by either selecting *Transport Protocols* in the *Define* menu or by clicking the ".." button next to the transport protocol field in any traffic source. When either of these options is performed a window such as the one shown in Figure 3.4 will appear.

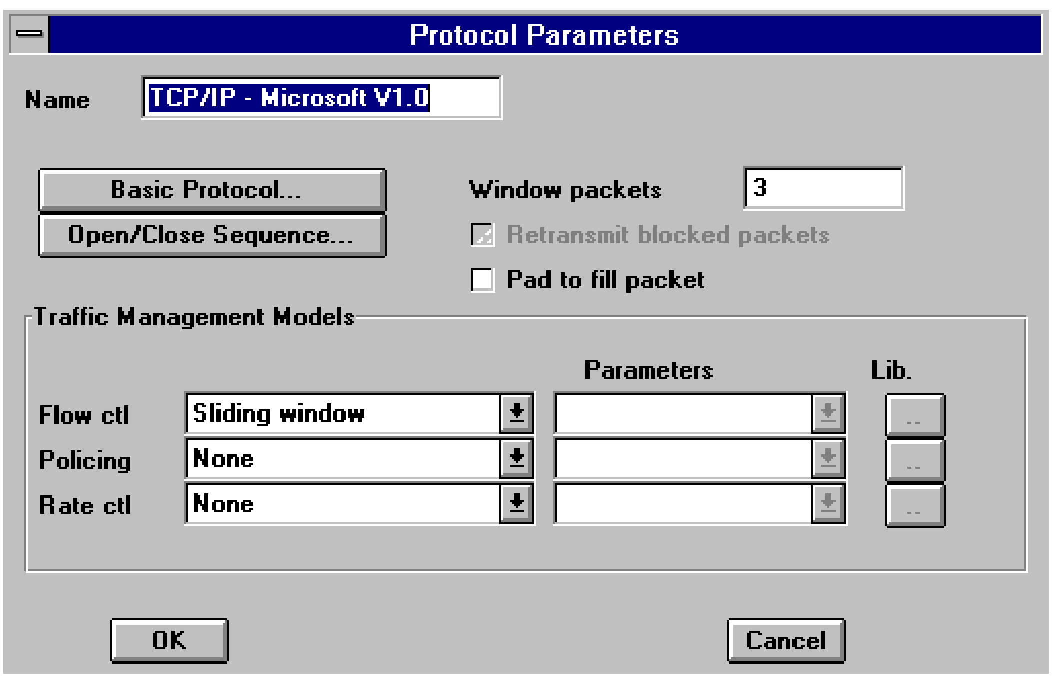


Figure 3.4 Transport Protocol Parameters

When defining a protocol, a unique name must be set and a routed protocol identification selected from the choices given. COMNET III does not allow the user to define the routed protocol identification. These are built into the program, and the user may choose from Appletalk, IP, OSI, Source Routing, IPX, bridge, and DECnet protocols. The packet data bytes field allows setting the maximum packet size. This number is made up of both the payload capacity of the packet and the overhead. The packet overhead bytes field sets the number of overhead bytes in the creation of a packet. If a method of flow control is used, the window size field must have an integer number placed in this field and a retransmit time must be specified. Also, if flow control is used, the number of bytes of the acknowledgment packet must be set. If no flow control is needed in the protocol, the flow control field should be set to none. COMNET III has flow control algorithms for fixed window, sliding window, and SNA pacing available within the application.

a. Fixed Window Flow Control

The originating node creates and transmits packets until the number of packets equals the window size. The destination only acknowledges the *last* packet to reach the destination. When the acknowledgment for the last packet is received at the origin, another set of packets (#packets = window size) may be created and transmitted.

b. Sliding Window Flow Control

The originating node creates and transmits packets until the number of packets equals the window size. The destination acknowledges *every* packet. When the acknowledgments for all packets are received at the origin another set of packets (#packets = window size) may be created and transmitted.

c. SNA Pacing Flow Control

In this mode of flow control, a pacing counter is initialized to the window size specified. This counter decrements each time a packet is created, and packet creation stops when the counter reaches zero. The destination only acknowledges the first packet of the window of packets which is sent. When the acknowledgment is received by the originating node, the counter is incremented and another window of packets may be created and transmitted.

6. Packetizing Delay

Packetizing delay is the amount of processing time in milliseconds to make a packet at the originating node. The COMNET III Users Manual specifies that this delay could be used to model transaction processing times at a node vice building detailed applications, or to model the transport layer overhead processing [Ref. 2 p. 110].

7. Message Size

Message size may be selected to be in units of either bytes or packets. This option is set by setting the message size units field in any of the traffic sources to the desired option. The message size calculation has two options available for determining the size of the message to be calculated. Any of the statistical distributions supported in COMNET III may be used to model the size of the message generated. In addition, if received message scheduling is used, the size of the message may be based on the size of the message which triggered the traffic source. Only a linear model is available if the option is selected, and the size calculation takes the form:

$$\text{Message Size} = \text{Multiplier} * \text{Received Message Size} + \text{Offset}$$

Thus, if the multiplier is set to 1 and the offset to 100, a message of 100 bytes greater in length than the message received will be generated.

8. Message Text

When a message is created, an arbitrary text label associated with the message is also created. This text label can then be used to trigger an application or traffic source at the receiving node. Different applications or traffic sources may use the same message text. This can often be a very useful feature in simplifying the creation of a model as various nodes having different parameters for their traffic’s size and interarrival time may be desired to trigger a response source on another node. The message text for these applications or traffic sources may all be set the same in order to simplify remembering which messages are used to trigger which applications.

There are three methods for which the text of the message may be set. The *Use original message* option can be used for applications or traffic sources which were scheduled using the received message option. In this case the text of the message which triggered the source is set as the text of the message to be generated. The *Copy message name* option is the default setting which uses the name of the source generating the traffic as the text. The final option, *Set message text*, allows the user to explicitly set the message text by typing in the desired text or choosing the text of any traffic source that has been created in a model.

9. Message Destination

The final attribute of all message sources is the destination field. The option of setting a destination is available to all traffic sources except the response source where, by default, the destination will be to the node that triggered the source generating the packet. The four options available for defining message destinations are to a random neighbor, random list, weighted list or a multicast list.

a. Random Neighbor

When the random neighbor option is chosen, COMNET III builds a list of all other nodes which are one hop away from the node generating the packet. These nodes are each given an equal probability of receiving the packet and picked randomly as the destination when a packet is created.

b. Random List

A random list allows the user to select a random group of nodes which will each be assigned an equal probability of being randomly selected as the destination. The list of possible nodes is defined by the user by selecting the *Edit Destination List* button. This will bring up a list of all nodes in the model from which the nodes desired may be selected.

c. Weighted List

The weighted list is very similar to the random list with exception that the probability of a given node being chosen as the destination may be weighted as needed to model where the traffic will go. Building the list for the destination is done in the same manner as in building the random list. There is, however, an additional field for setting the probability for a message being selected to go to a given node. This probability may be set by either double clicking on a given destination in the list or by selecting a destination with a single click and then depressing the edit button. In both cases a window will appear which allows setting the probability. The most important part of building this list and setting the probability values is that the sum of the probabilities of all nodes in the list must be exactly equal to one. If this is not true, an error message will appear when the model is verified prior to running a simulation.

d. Multicast List

The multicast list is used to send a message to more than one destination at a single time. The list creation is done in the same manner as that of a random list. A restriction placed on this option is that it is only available for use with a message source. The COMNET III Users Manual states that no flow control is used when generating packets to a multicast list irrespective of the transport protocol chosen to deliver the message [Ref. 2 p. 65]. In working with the program, however, this was not found to be true. To use this type of list the transport protocol which is selected must not use any flow control or attempt of packet retransmission. The only protocol which is directly built into the application which has this feature is UDP/IP.

D. MODELING OF MESSAGE TRAFFIC

All traffic sources within COMNET III follow a similar logic in the creation of message traffic when running a simulation. The following list describes the key steps which occur in the simulation:

1. A message of the required size, priority, routing class, transport protocol, and packetizing delay is created.
2. The message text is set.
3. The destination of the message is set. In the case of the response source, the message destination is restricted to only being to the node which triggered the response source.
4. If required acknowledgments (ACK) packets from the flow control windowing have been received, a packet is created. If ACK packets have not been received, the packet creation will not occur until receipt of the outstanding ACK packets. The first packet of any message will never have to wait for an ACK packet.
5. A routing decision is then made to determine the buffer to which the packet has to be sent. The node which is generating the packets is then made busy by the packet switching delay defined for the node. If the routing algorithm cannot find a route, then the packet is blocked. The packet may then be optionally retried depending on the characteristics of the transport protocol.
6. If no space is available in the output buffer, the packet is dropped. If the transport protocol specifies retries, the retry time elapses and the packet creation is attempted again.
7. The packet is then transported across each link to arrive at its destination node. Again, if a block occurs at any point the packet is dropped and retry occurs depending on the transport protocol used.
8. When the packet reaches its destination, if flow control is in operation, and if receipt of the packet requires an acknowledgment, then an ACK packet will be

created at the destination and routed back to the origin using the same routing class, transport protocol, and routing algorithm as the received packet.

This previous list of steps is valid for all traffic sources, but the following logic is applied prior to the creation of the message packets when setting up a session:

1. The session is created at the node where the session source is located, and the destination for the session is made.
2. A session setup packet is created. Unlike message packets, the setup packet does not incur a packetizing delay. It does, however, incur a packet processing delay which is defined on the node creating the packet.
3. A routing decision is then made and the packet switched to the appropriate output buffer.
4. Session packets may be blocked in several manners including insufficient space in the output buffer or reaching the session limit for a node or a link.
5. When the setup packet reaches the destination node, the destination creates a confirmation packet following the reverse route of the setup packet.
6. On receipt of the confirmation packet at the originating node, message packets may then begin to be created.

When running a simulation, there may be more than one application or traffic source scheduled or in progress of generating packets at the same time on a node. Each node maintains a list of applications and message sources currently scheduled. The logic in deciding who gets to send the next packet is that the application or traffic source at the top of the list is selected and a packet is created. After execution, the application or traffic source is then placed at the bottom of the list and the next application runs. This multitasking of applications or traffic sources is similar to a round robin tasking with the exception that a message which is waiting for flow control authorization will not block another application.

E. PROBLEM STATEMENT

A portion of a mythical company's LAN consists of two networks. Each network services one department. One network is set up in accordance with the IEEE 802.3 CSMA/CD 10BaseT Ethernet network. The other network is set up in accordance with the IEEE 802.5 16Mbps Token Ring standard. The two networks are connected together via a Proteon DNX300, V12.0 router. The ethernet LAN supports 10 computers, one of which is designated as the e-mail server for both departments. The token ring LAN also supports 10 computers where one is designated as the file server for both departments.

The company is currently considering the addition of personnel to both departments. However, a concern is that the current network configuration will not be able to support additional personnel as the company currently has no method of measuring network utilization or delay. The company wishes to estimate these current baseline levels prior to hiring any new personnel. Also, complaints have recently been made by the employees in both departments concerning network delays when trying to receive a file from the file server.

An estimate of the common traffic which flows across the two LANs indicates that the majority of traffic is from e-mail, file transfers from various applications, and a LAN voice mail system which allows supervisors to send voice mail message to personnel in there own department over the LAN. Interviews with the employees in each department, and estimates of the common size of messages sent over the LAN were used to statistically describe the message characteristics.

E-mail is used by all personnel in both departments. The interview of all personnel indicated the sending of an email message occurs at an average interarrival rate which can be described by an exponential distribution which has a mean of 900 seconds. The size of email messages can be described by a uniform distribution where the size is evenly dispersed over the range of 500 to 2000 bytes. All email messages are sent to the email server located on the ethernet LAN where they are saved in each employee's account. In order to read the email the employee must send a request to the e-mail server for downloading the messages to their computer. The interarrival time for the checking of email can be described by a Poisson distribution with a mean of 900 seconds. Each request has a message size set at 60 bytes. Upon receipt of a request to download e-mail, the e-mail server reads the employee's file and transmits the messages to the employee's computer. The amount of time necessary to read the file and to process the messages on the server can be described by a uniform distribution which ranges from 3 to 5 seconds. The size of the e-mail messages transmitted by the server may be described by a Normal distribution which has a mean of 40000 bytes and a standard deviation of 10000 bytes.

There are eight employees in each department who each have their own computer and also generate traffic due to requests for files to the file server. The file requests can be described by an exponential distribution with a mean of 900 seconds. The size of the file requests vary in accordance with a uniform distribution which ranges from 10 to 20 bytes in length. All file requests are sent exclusively to the file server located on the token ring network. Upon receipt of a file request, the file server reads the file and transmits it to the computer which made the request. There is little delay in this process. The size of the files to be transferred may be described by a Normal distribution which has a mean of 100000 bytes and a standard deviation of 25000 bytes.

The LAN voice mail application is used solely by the supervisors of each department. The supervisors generally send voice mail only to personnel within their own department. This application first sets up a session with the computer belonging to the person the message is to be sent to. Upon receipt of the acknowledgment that a session has been set up, the voice mail message is sent. The size of these messages can be characterized by a Normal distribution with a mean of 50000 bytes and a standard deviation of 1200 bytes. The interarrival time of these message can also be described by a normal distribution with a mean of 1000 seconds and a standard deviation of 10 seconds. As a final note, all of the message sources which are to be defined use TCP/IP as the transport protocol and have been estimated at having a packetizing delay of 0.01 milliseconds.

F. BUILDING THE NETWORK MODEL

The steps to build the model are presented in a method such that the pieces of the model are constructed object-by-object. In general, models are normally constructed by first constructing the network topology. This implies constructing all computers, routers, switches, and links through which messages are created and transported. The second step consists of defining the traffic sources which will generate the load on the network. Upon completion of building the network model, the network topology should appear similar to the topology shown in Figure 3.5.

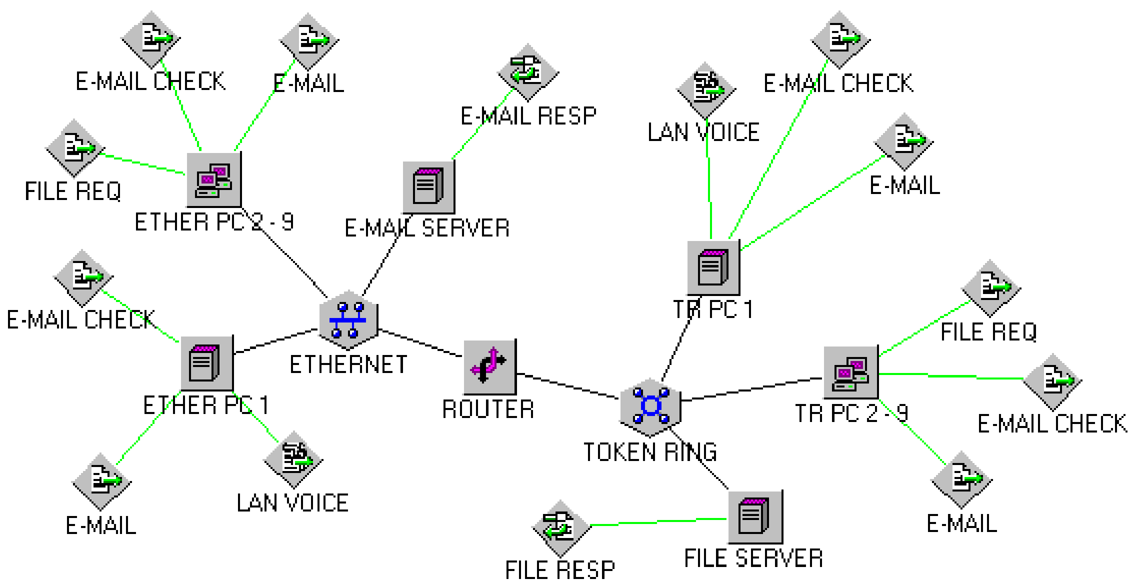


Figure 3.5 Network Model Topology

1. Creating the Network Topology

The following defines the steps necessary to create the nodes and links which will be used in the model:

a) Create a collision link and a token passing link in the work area.

b) Edit the fields of the collision link detail window to those values shown in the following list. Figure 3.6 depicts the link detail window upon completion of this step.

- Name: ETHERNET
- Type: CSMA/CD
- Parameters: 802.3 CSMA/CD 10BASET

c) Edit the fields of the token passing link detail window to those values shown in the following list.

- Name: TOKEN RING

- Type: Token passing
- Parameters: 802.5 16 Mbps

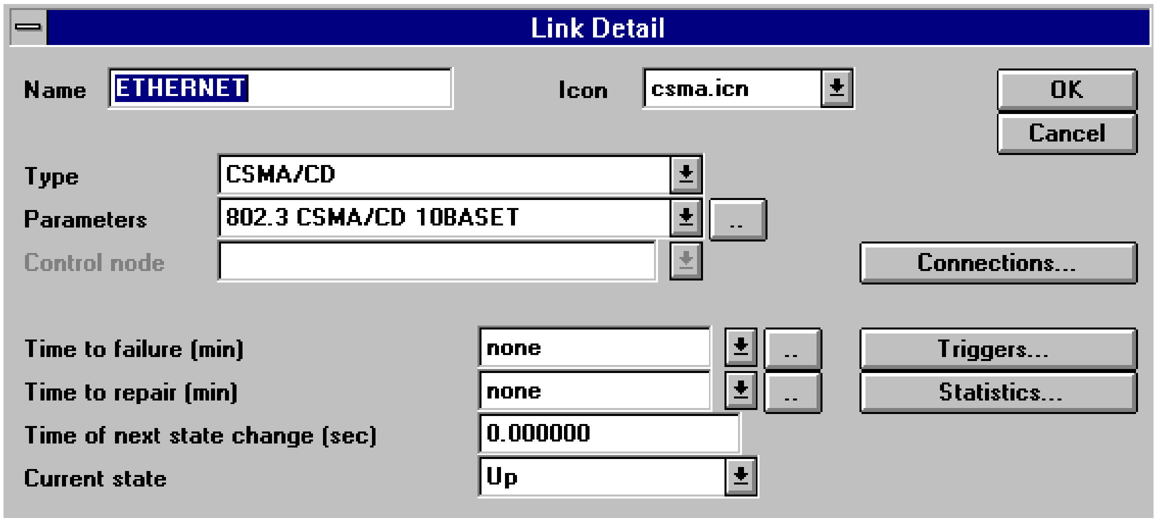


Figure 3.6 Ethernet Link Detail Window

d) Create a router node and place it between the two links in the work area. Connect the router node to both links using either connection tool on the toolbar. Edit the fields of the node detail window for this node to those of the following list and shown in Figure 3.7.

- Name: ROUTER
- Type: Router
- Parameters: Proteon DNX 300, V12.0

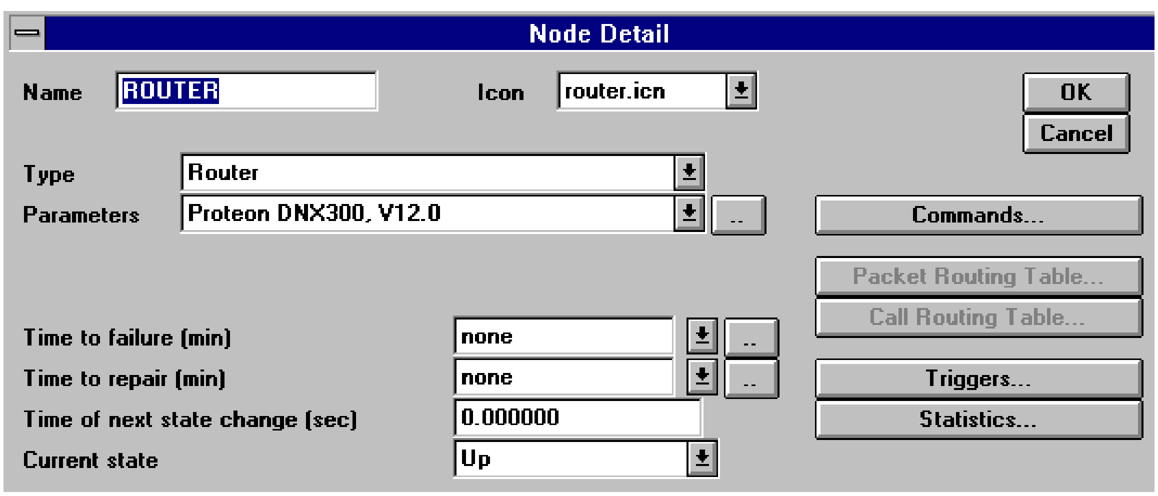


Figure 3.7 Router Node Detail Window

e) Create four computer and communication (C&C) nodes. Edit the fields of the node detail window for each node to the values shown in Table 3.1. Attach each node to the link specified in Table 3.1.

Name	Type	Parameters	Link
ETHER PC 1	C&C Node	Default	ETHERNET
E-MAIL SERVER	C&C Node	Default	ETHERNET
TR PC 1	C&C Node	Default	TOKEN RING
FILE SERVER	C&C Node	Default	TOKEN RING

Table 3.1 Computer and Communication Node Detail Values

f) Choose the *Node Parameters/Computer Group* option from the *Define* menu. In the first computer group parameters window which appears press the *Add* button. In the second computer group parameters window which appears, edit only the following fields:

- Name: PC
- Number in group: 8

g) Create two computer group nodes in the work area. Edit the fields of the node detail window for each node to the values shown in Table 3.2. Attach each node to the link specified in Table 3.2.

Name	Type	Parameters	Link
ETHER PC 2 - 9	Computer Group	PC	ETHERNET
TR PC 2 -9	Computer Group	PC	TOKEN RING

Table 3.2 Computer Group Node Detail Values

2. Creating the E-mail Message Source

The e-mail message source is used to model the transport of e-mail message from nodes in the network to the e-mail server.

a) Create a message source and attach it to the ETHER PC 1 node.

b) Edit the message source such that the fields of the message source window are set to the values of the following list. Figure 3.8 depicts the message source window upon completion of this step.

- Name: E-MAIL
- Schedule by: Iteration time
- Interarrival: Set to an exponential distribution with a mean of 900 seconds. The stream value may be set to any integer value from 0 to 99.
- Priority: 1
- Routing class: Standard
- Transport protocol: TCP/IP - Microsoft
- Packetize: 0.01 milliseconds
- Message size calculation: Probability distribution
- Probability distribution: Set to a uniform distribution with a minimum of 500 bytes and a maximum of 2000 bytes. The stream value may be set to any integer value from 0 to 99.
- Message text option: Copy message name
- Destination type: Set to weighted list. Create a list containing only the E-MAIL SERVER node.

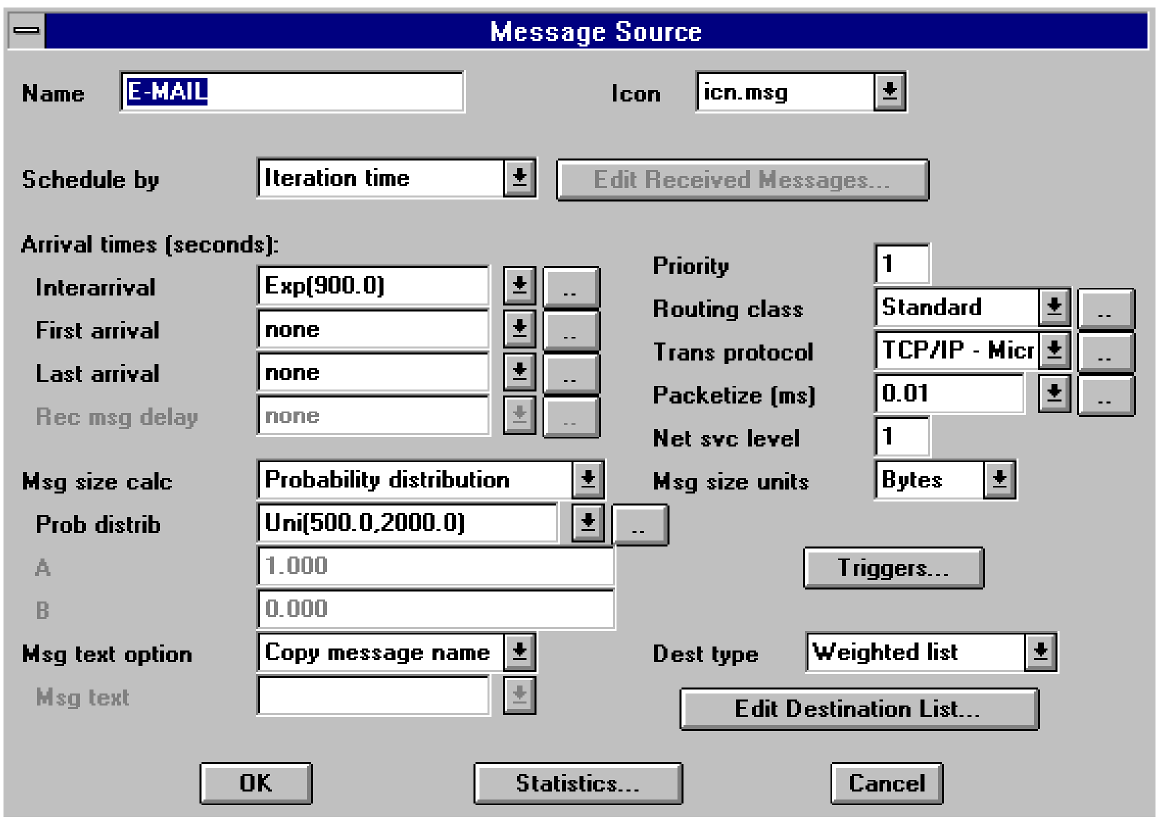


Figure 3.8 E-MAIL Message Source

c) Create three clones of the E-MAIL message source. It is important to clone the message source vice copy and pasting. Copying a message source will cause the destination information to be lost in the copies made whereas cloning will keep the destination information.

d) Attach one copy of the E-MAIL message source to the ETHER PC 2 - 9, TR PC 1, and TR PC 2 - 9 nodes.

3. Creating the E-mail Checking Message Source

The e-mail check message source is used to model periodic checks by users on the network to the e-mail server to download their mail.

a) Create a message source and attach it to the ETHER PC 1 node.

b) Edit the message source such that the fields of the message source window are set to the values of the following list. Figure 3.9 depicts the message source window upon completion of this step.

- Name: E-MAIL CHECK
- Schedule by: Iteration time
- Interarrival: Set to a Poisson distribution with a mean of 900 seconds. The stream value may be set to any integer value from 0 to 99.
- Priority: 1
- Routing class: Standard
- Transport protocol: TCP/IP - Microsoft
- Packetize: 0.01 milliseconds
- Message size calculation: Probability distribution
- Probability distribution: Set to a constant value of 60 bytes.
- Message text option: Copy message name
- Destination type: Set to weighted list. Create a list containing only the E-MAIL SERVER node.

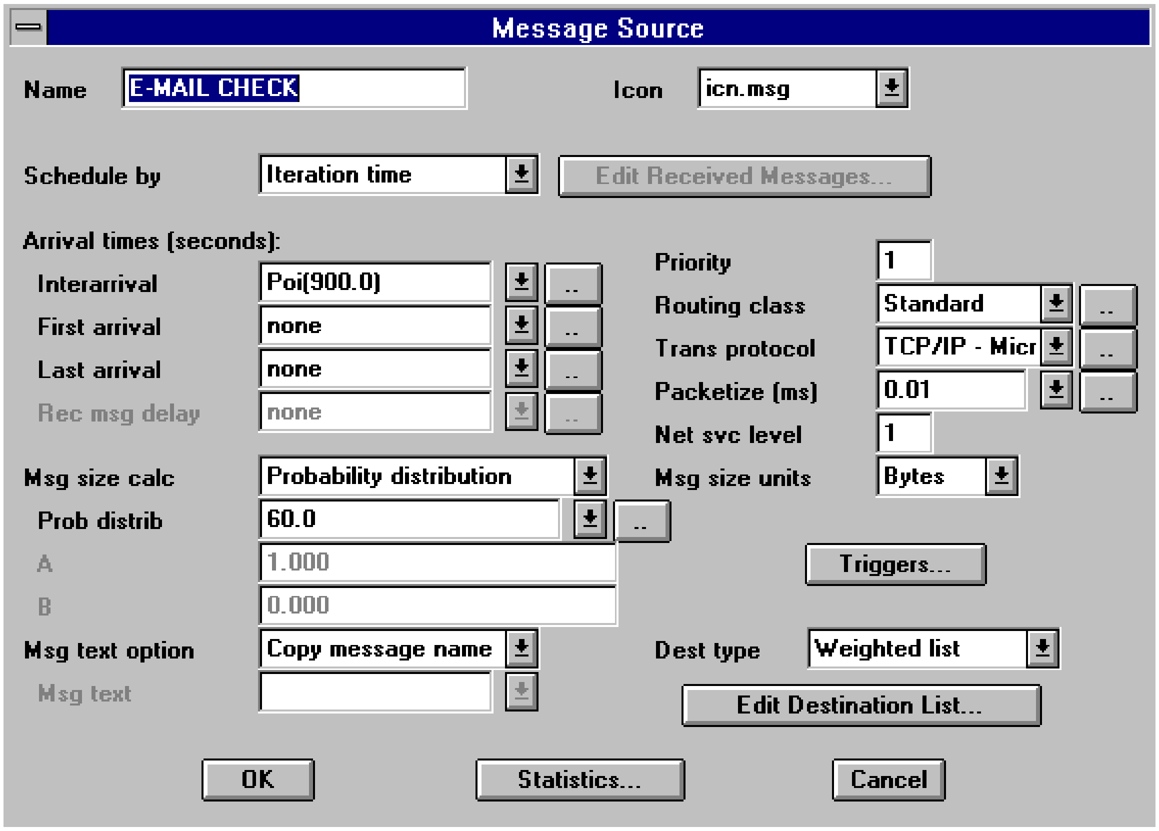


Figure 3.9 E-MAIL CHECK Message Source

c) Create three clones of the E-MAIL CHECK message source.

d) Attach one copy of the E-MAIL message source to the ETHER PC 2 - 9, TR PC 1, and TR PC 2 - 9 nodes.

4. Creating the E-mail Response Source

The e-mail response source is used to model the downloading and transport of e-mail traffic from the e-mail server.

a) Create a response source and attach it to the E-MAIL SERVER node.

b) Edit the fields of the response source window to the values shown in the following list and depicted in Figure 3.10.

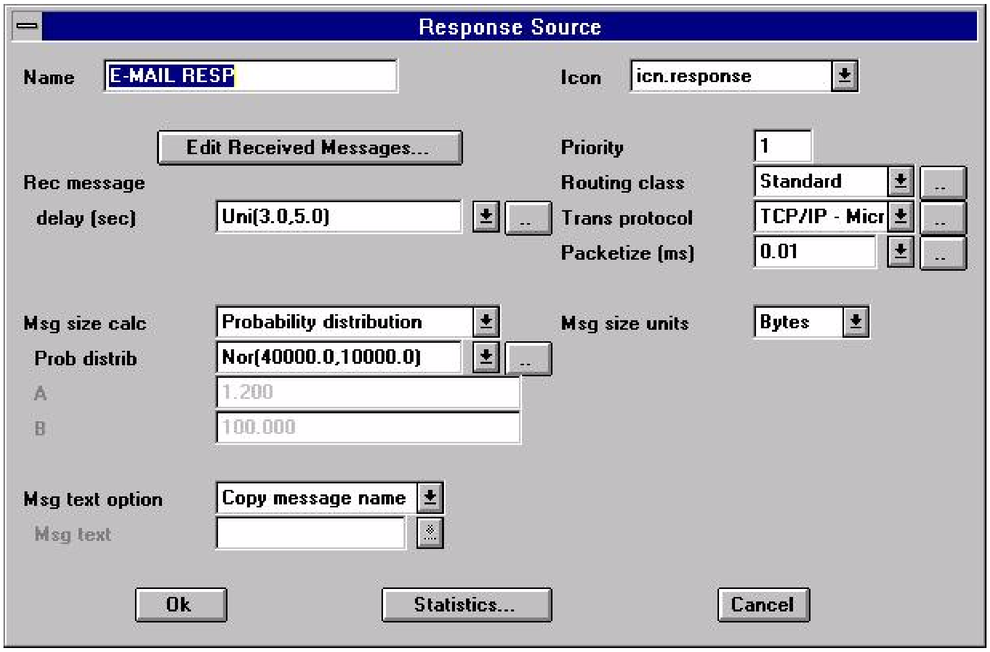


Figure 3.10 E-MAIL RESP Response Source

- Name: E-MAIL RESP
- *Edit Received Message* button: Press and create a list such that receipt of one message with the text E-MAIL CHECK will trigger the source.
- Received message delay: Set to a uniform distribution with a minimum of 3 seconds and a maximum of 5 seconds. The stream value may be set to any integer from 0 to 99.
- Priority: 1
- Routing class: Standard
- Transport protocol: TCP/IP - Microsoft
- Packetize: 0.01 milliseconds
- Message size calculation: Probability distribution
- Probability distribution: Set to a normal distribution with a mean of 40000 bytes and a standard deviation of 10000 bytes. The stream value may be set to any integer from 0 to 99.
- Message text option: Use original message

5. Creating the File Request Message Source

The file request message source is used to model requests to the file server to download files.

a) Create a message source and attach it to the ETHER PC 2 - 9 node.

b) Edit the message source such that the fields of the message source window are set to the values of the following list. Figure 3.11 depicts the message source window upon completion of this step.

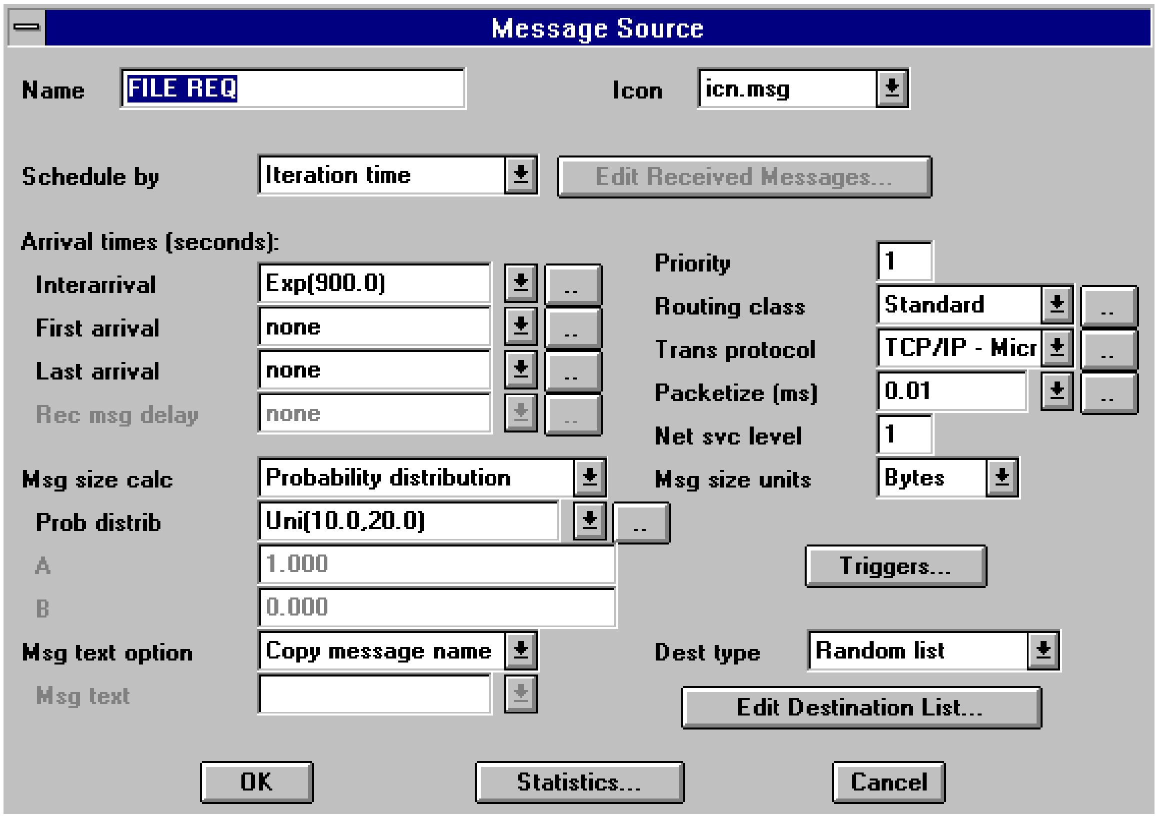


Figure 3.11 FILE REQ Message Source

- Name: FILE REQ
- Schedule by: Iteration time
- Interarrival: Set to an exponential distribution with a mean of 900 seconds. The stream value may be set to any integer value from 0 to 99.
- Priority: 1
- Routing class: Standard
- Transport protocol: TCP/IP - Microsoft
- Packetize: 0.01 milliseconds
- Message size calculation: Probability distribution
- Probability distribution: Set to a uniform distribution with a minimum of 10 bytes and a maximum of 20 bytes. The stream value may be set to any integer value from 0 to 99.
- Message text option: Copy message name

- Destination type: Set to random list. Create a list containing only the FILE SERVER node.

c) Create one clone of the FILE REQ message source and attach the clone to the TR PC 2- 9 node.

6. Creating the File Server Response Source

The response source which will be connected to the file server is used to model the transmission of files across the network upon receipt of a file request.

a) Create a response source and attach it to the FILE SERVER node.

b) Edit the fields of the response source window to the values shown in the following list and depicted in Figure 3.12.

- Name: FILE RESP
- *Edit Received Message* button: Press and create a list such that receipt of one message with the text FILE REQ will trigger the source.
- Priority: 1
- Routing class: Standard
- Transport protocol: TCP/IP - Microsoft
- Packetize: 0.01 milliseconds
- Message size calculation: Probability distribution
- Probability distribution: Set to a normal distribution with a mean of 100000 bytes and a standard deviation of 25000 bytes. The stream value may be set to any integer from 0 to 99.
- Message text option: Use original message

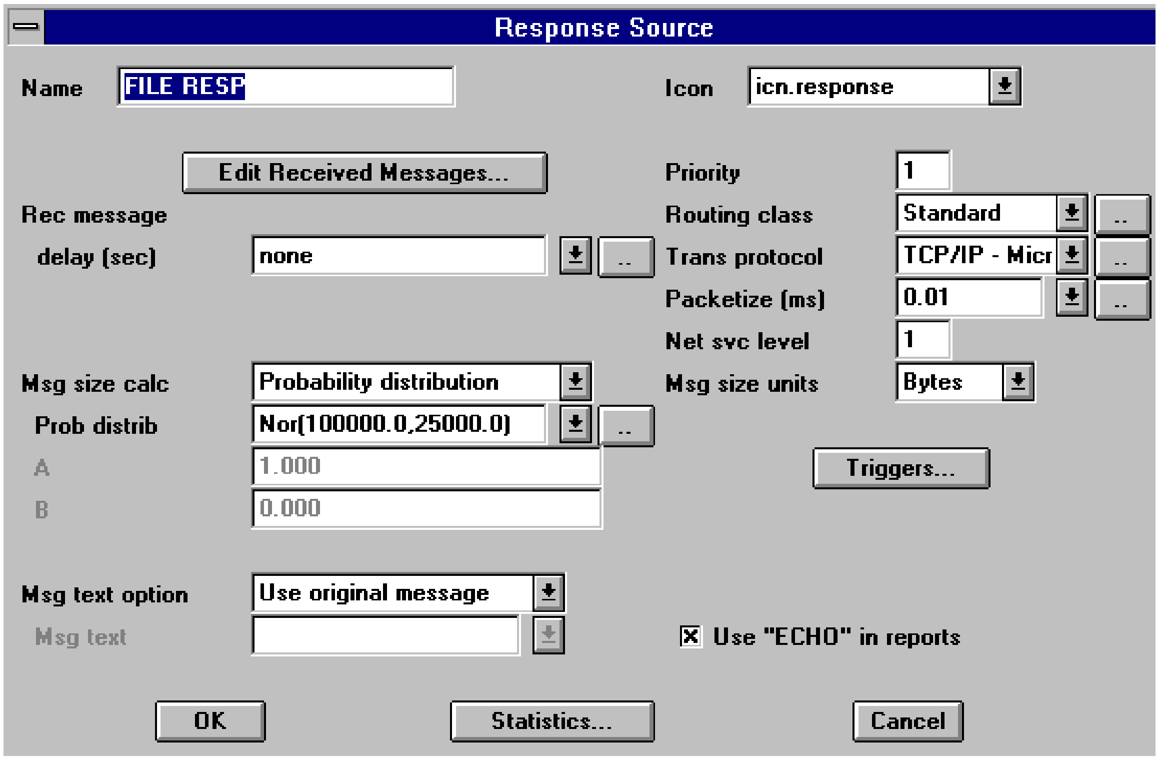


Figure 3.12 FILE RESP Response Source

7. Creating the LAN VOICE Session Source

The LAN VOICE session source is used to model the transmission of voice mail across the network.

a) Create a session source and attach it to the ETHER PC 1 node.

b) Edit the fields in the session source window to those of the following list and which are depicted in Figure 3.13:

- Name: LAN VOICE
- Schedule by: Iteration time
- Interarrival: Set to a normal distribution with a mean of 1000 seconds and a standard deviation of 10 seconds. The stream value may be set to any integer value from 0 to 99.
- Priority: 1
- Routing class: Standard
- Transport protocol: TCP/IP - Microsoft
- Packetize: 0.01 milliseconds
- Messages/session: 1
- Message IAT: none
- Setup packet: 40 bytes
- Confirm packet: 40 bytes

- Message size: Set to a normal distribution with a mean of 50000 bytes and a standard deviation of 1200 bytes. The stream value may be set to any integer value from 0 to 99.
- Message text option: Copy message name
- Destination type: Set to random list and create a list containing only the ETHER PC 2 - 9 node.

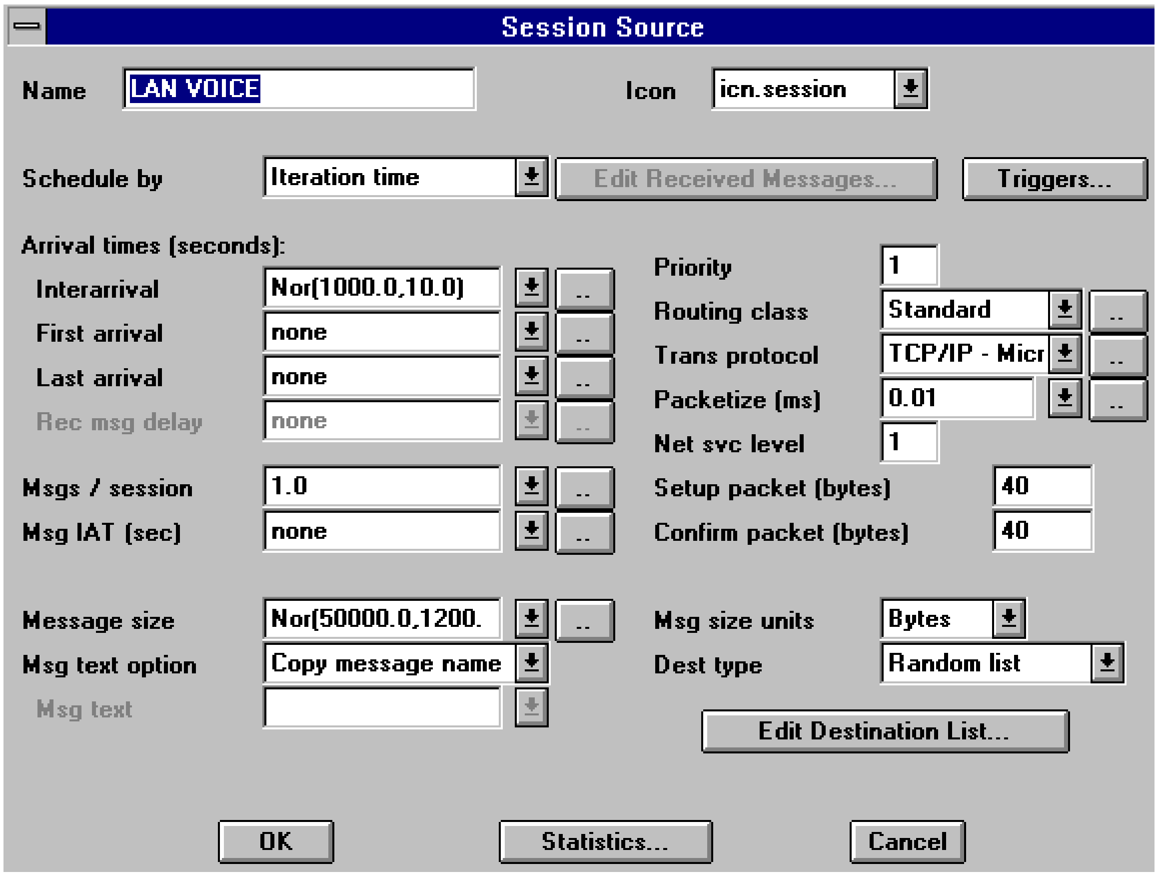


Figure 3.13 LAN VOICE Session Source

c) Create a copy of the LAN VOICE session source and attach it to the TR PC 1 node. Edit the destination list of this copy so that traffic from this source only goes to the TR PC 2 - 9 node.

d) The model is now complete. Use the *Verify Model* option from the *Simulate* menu to check the model. Fix any errors which may have occurred while building the model and save the model.

G. SIMULATION AND RESULTS

The simulation used to analyze the problem was set to run for 3600 seconds to simulate the traffic which would occur in one hour. A warmup length of 120 seconds was used to allow message traffic to build up prior to beginning to collect statistics. For this simulation only one replication was be run. Prior to running the simulation the reports which will collect statistics need to be set. The following reports were used for this simulation:

- Link Reports: Channel Utilization and Collision Statistics for all links.
- Node Reports: Received Message Count for all nodes.
- Message and Response Reports: Message Delay for all nodes.
- Session Source Reports: Message Delay for all nodes.

After running the simulation, the results may be viewed by selecting the *Browse Reports* option on the *Report* menu. The following shows a portion of the report which is generated by COMNET III after the completion of a simulation.

```
                             LINK DELAYS AND UTILIZATION

                     REPLICATION 1 FROM 120.0 TO 3720.0 SECONDS

                               FRAMES          TRANSMISSION DELAY (MS)        %
LINK                       DELIVERED RESENT    AVERAGE     STD DEV     MAXIMUM   UTIL
_____________________      _________ ______   _________   _________   _________  _____

ETHERNET                        9483      0       0.763       0.883      31.002   0.11
TOKEN RING                      7386      0       0.381       0.363       0.763   0.08

ORIGIN / MSG SRC NAME:  MESSAGES          MESSAGE DELAY (MILLISECONDS)
  DESTINATION LIST      ASSEMBLED      AVERAGE        STD DEV       MAXIMUM
______________________  _________   ____________   ____________   ____________

E-MAIL SERVER / src E-MAIL RESP:
  ECHO                         73         42.503         11.426         67.976
FILE SERVER / src FILE RESP:
  ECHO                         31      80186.450      82589.636     434398.450


LINK NAME                  ETHERNET


ACCESS PROTOCOL              CSMA/CD

COLLISION EPISODES              4830

COLLIDED FRAMES                 9665

NBR OF TRIES TO RESOLVE
  AVERAGE                       1.63
  STANDARD DEVIATION            0.96
  MAXIMUM                          9

NBR OF DEFERRALS                3161

DEFERRAL DELAY  (MS)
```

```
  AVERAGE                       0.62
  STANDARD DEVIATION            0.52
  MAXIMUM                       1.22

DEFERRAL QUEUE SIZE  (FRAMES)
  AVERAGE                       0.00
  STANDARD DEVIATION            0.02
  MAXIMUM                          2

MULTIPLE COLLISION EPISODES
  NBR EPISODES                     5
  AVG PER EPISODE               3.00
  MAX PER EPISODE                  3
```

The results show that neither network is utilized to a great extent and collisions on the ethernet network are not severe. The interesting note is that the simulation does show that long delays do occur when requesting files from the file server, but that the delay is not consistent as the standard deviation is greater than the average delay. One possibility to remedy this situation would be to add another file server to the ethernet network which would serve only users on that network.

The final question to be asked is “How accurate is this simulation ?” The answer is probably not as good as needed to make any major changes to the network, but it provides a decent first look at estimating the performance of the network. The model which was developed in this chapter was not based on any actual network, but was for demonstration purposes only to give a first look at building a model within COMNET III with the emphasis on familiarization with the tools available to simulate network traffic.

IV. THE COMPUTER AND COMMUNICATION NODE

A. INTRODUCTION

The goal of this chapter is to familiarize users with the computer and communications node (C&C node) object and the characteristics of this object.
The computer group node will also be covered as it is very similar to the computer and communication node, and the port object will also be covered.

B. COMPUTER AND COMMUNICATIONS NODE THEORY

The computer and communication (C&C) node is a generic node that is used for the modeling of end systems such as computers, printers, facsimile machines, or any general piece of network hardware. The C&C node may act as the origin or destination for message traffic, run applications, or act as a switching point within a network. Any type of link, and as many links as needed, may connect to this type of node for modeling a network. The C&C node is represented in a model as having the following attributes:

- Input buffers for each link connected to the node for accepting packets transmitted to the node.
- A processor for command execution and packet processing.
- Output buffers for each link connected to the node through which the node may route packets.
- Local disk storage for modeling disk read and write processes.
- A command list which defines how application commands are to be executed on the node.
- A pending application list of all applications and traffic sources currently scheduled to run.
- A prototype application list of all available applications and traffic sources for the node.
- A received message list for saving received messages used to trigger an application or traffic source.
- A list of files which reside in storage on the local disk.

1. Computer and Communications Node Processor Scheduling

The processor in the computer and communications node is used for the processing of application and traffic sources, and for the processing of packets which the node receives. When the processor becomes idle a choice must be made as to whether to run an application from the pending application list or to process a packet waiting in an input buffer queue. The choice is made by comparing the wait times of the packet at the head of each input buffer to the wait time of the application at the head of the pending application list. The packet or application which has the longest wait time is chosen to execute next.

Each node maintains a list of all applications and traffic sources which may run on that node. When scheduled to run in an application, a prototype of the application or traffic source is created and added to the pending application list. The list is maintained such that the application or traffic source which has waited the longest will be at the top of the list. When the choice is made for the node to process a packet or run an application, normally the application at the top of the pending application list is chosen. The exception to the rule occurs when the application at the top of the list is waiting for flow control clearance to create packets or for a file to be unlocked for a read command. These exceptions make the scheduling of applications on the list be described as a semi round robin process.

The node processor is not modeled by a unique value such as the processor speed. Instead, various attributes of the node are set to model the use of the processor in performing different functions. Parameters for a local hard disk are used to describe the processing required when reading or writing a file. Packet processing attributes model the manner in which the processor on the node processes packets. A processing time per cycle attribute is used to describe the manner of executing processing commands in an application.

In the scheduling of all these types of functions the option is given to the user to model the node as a single function computer which completes the entirety of one task before going on to the next or as a multitasking type computer which uses time slicing to alternate between various applications.

a. No Time Slice Execution

A node in this mode of operation will pick an application to run or a packet to process as in the manner previously described. The processor will then execute every command within an application or complete the processing of a packet. When the present task is completed a scheduling decision will be made for the next task to accomplish. The exception to this rule occurs in the generation of messages on a node by either an application or a traffic source. When a message is created on a node, it must have flow control clearance to generate the packets to be sent across the network. Instead of waiting for flow control to complete the sending of all packets of one message at one time, the processor will either process incoming packets or generate packets for another application in the pending application list. In this manner packets coming into the node will be interleaved with those packets being created by the node.

b. Time Slice Execution

When time slice execution is chosen the method of choosing which application to run or packet to process does not change. The difference is that each process is given a portion of the processor time, and at the end of this time the application is interrupted and put at the bottom of the pending application list. When the application is rescheduled later the interrupted command resumes where it last left off.

2. Buffer Processing

Buffer processing is used to model the delays within the input/output ports of the simulated computer due to the processing of packets at the port buffers. This processing delay is not a portion of the main processor simulated on a C&C node. A packet arrives at a node when all of the frames which make up the packet reach the node. The packet is then attempted to be placed into the input buffer of the node. If either the port input buffer is full or the total buffer space for the entire C&C node is full the packet is blocked. If the packet is accepted into the port buffer, its position in the buffer's queue is based on decreasing priority and first-in-first-out order. The packet will then incur the buffer processing delay. After incurring this delay, the packet is then made available for switching by the main C&C node processor.

C. COMPUTER AND COMMUNICATION NODE PARAMETERS

The computer and communications node may be created from either the toolbar or by selecting the *Nodes/Computer and Communications* option from the *Create* menu. After creation of the node, editing it will bring up a window as shown in Figure 4.1. The

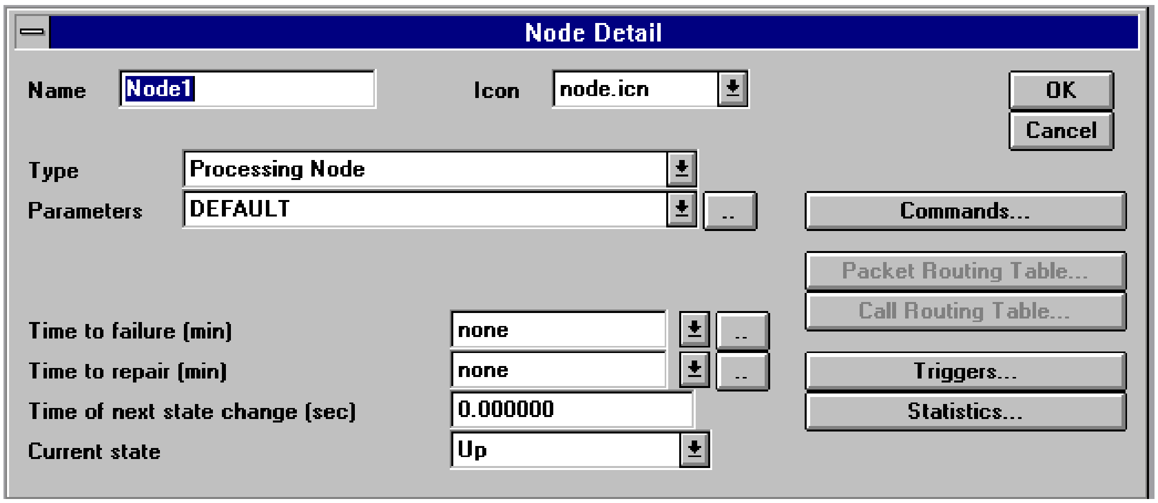


Figure 4.1 Computer and Communications Node Dialog Box

creation of any type of node will bring up this window. Each node created should be assigned a unique name to be able to identify it in a model. The window also allows entering information concerning the time to failure and time to repair which may be modeled as a constant or using a statistical distribution. The current state of the node may be set to either up or down, and a simulation time to toggle the state of the node from its current setting may be entered. The *Command* button allows for the generation of application commands which will be available only to the node they are created on. The parameter set which will define the attributes of the node may be set by either pressing the button with double dots on it at the end of the parameter field or by defining a computer and communications node parameter set from the define menu. When either of these option is selected a window will appear listing all of the parameter sets available for the node. By depressing the *Add* button in this window, the node parameter window as shown in Figure 4.2 will appear. Discussion of the parameters which may be entered are covered in the following subsections.

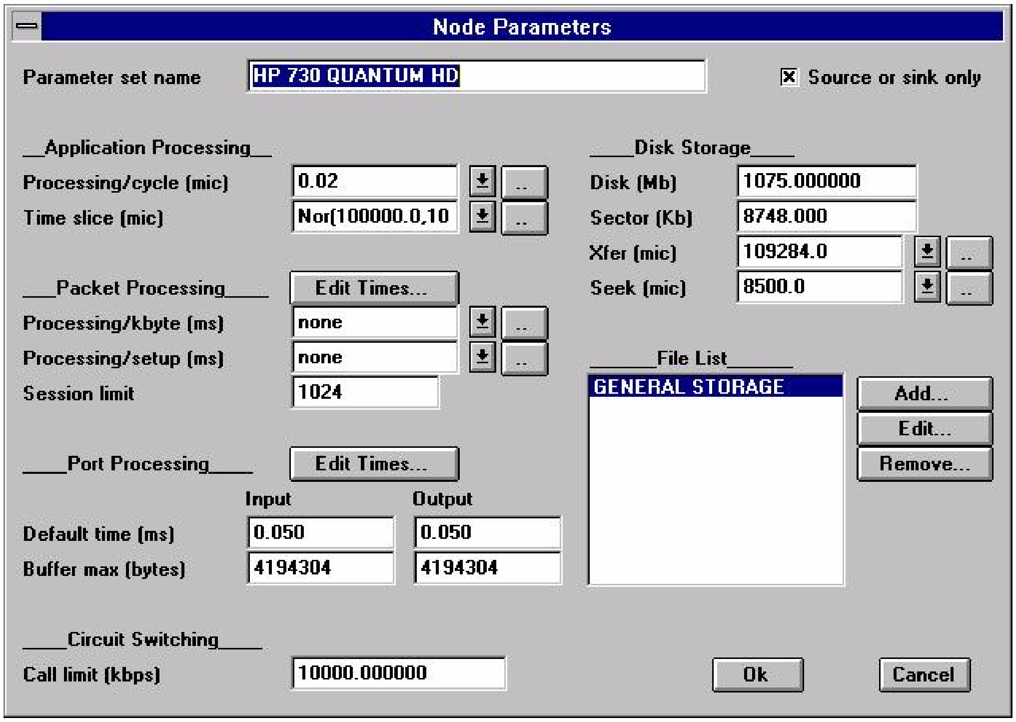


Figure 4.2 Computer and Communication Node Parameter Window

1. Parameter Set Name

A unique name should be assigned to any parameter set made. Once a parameter set is entered into a model it is available for use on all subsequent C&C nodes created. The parameter set on subsequent nodes is selected using the name entered in this field.

2. Source or Sink Only

A check in this box designates that the node for which the parameter set is applied may only be used to run applications or for the receipt or generation of message traffic. The node may not be used to switch traffic.

3. Application Processing

The two fields of *Processing/cycle* and *Time slice* set the parameters which describe the basic operation in modeling a computer. They are also two of the most difficult parameters to model. According to Mr. Kit Jones, a CACI Products Inc. test engineer, "COMNET III does not provide a robust capability for the modeling of computers using the C&C node."

The *Processing/cycle* field may be set as a constant or by a statistical distribution and is entered in units of microseconds. This parameter setting this field is used only for modeling the speed at which processing commands in an application occur in a simulation. The choice of scaling was used as a processing command is entered in units of cycles. Thus, the actual amount of time a processing command will execute on a node is the processing/cycle field multiplied by the number of cycles to execute in a process.

The *Time slice* field may be set to none to model a computer for which multitasking of applications is not desired, or to a constant or any statistical distribution to model computers that do multitask. Most operating systems perform the time slicing of applications based on the number of applications currently running on a computer at a given time and the state of processes which are running. COMNET III currently does not support scheduling of applications in this manner. Correspondence with CACI Products Inc., Mr. Kit Jones, indicated that using a short time slice is the best method to approximate a multitasking environment as no process would wait an excessive amount of time to be serviced, and all processes would be serviced in a round-robin order.

4. Packet Processing

Packet processing is a delay applied to each packet as it is switched through a node which is used to model the processing time which would occur by the main node processor. This delay is in addition to the buffer processor delay which occurs in the port buffer. Packets created on a node first undergo a packetizing delay from the generating source to model the time needed to create a packet. The packet processing delay then occurs to model the processing delay of switching the packet to the desired output buffer.

The packet finally undergoes the buffer processing delay. The order of these delays is reversed for an incoming packet. packet processing may be modeled in three different ways in the model.

a. Packet Processing by Routed Protocol Identification

The packet processing in this mode is modeled based on the routed protocol identification of the packet being created or received. These values are set by pressing the *Edit Times...* button next to the packet processing as shown in Figure 4.2 which will a list of packet processing times to appear in a window such as shown in Figure 4.3. In this window press the *Add* button and the packet processing time window shown in Figure 4.4 will appear. In this window select the desired routed protocol identification from the drop down list. The *Processing/packet* field is set in units of milliseconds and may be set to a constant value or modeled using a statistical distribution. When all values are entered depress the *OK* button as shown in Figure 4.4, and then the *Done* button as shown in Figure 4.3 to set the entered values.

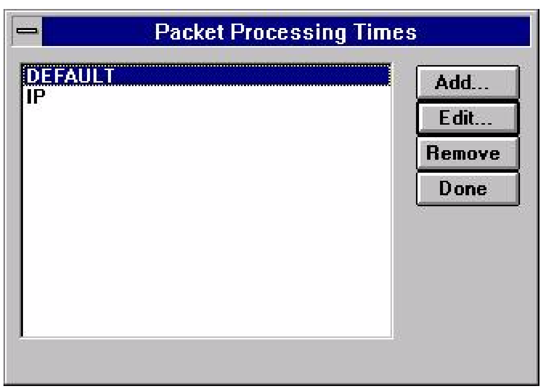


Figure 4.3 Packet Processing Times List Window

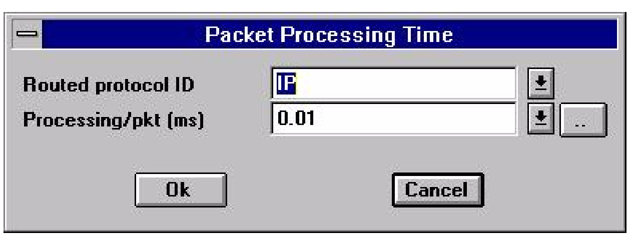


Figure 4.4 Packet Processing Times Editing Window

b. Packet Processing by Packet Size

A device may behave such that the switching of larger packets takes longer than the switching of a shorter packet. The *Processing/Kbytes* field allows for the modeling of this behavior using the linear relationship [Ref. 2 p. 130]:

$$\text{Switching Time} = Ax + B$$

- A = processing/Kbytes
- x = packet size in kilobytes
- B = the processing/packet.

The *Processing/Kbytes* field may be set to a constant or modeled using any statistical distribution.

c. Packet Processing for Setup Packets

Session sources and setup commands may be used to establish connection oriented routing across a network. All subsequent data packets which follow the setup packet will follow the same route as the setup packet. In establishing the route, the setup packet may be modeled as having a longer delay. The *Processing/setup* field only affects setup packets and may be used to model a longer delay at each node in setting up a route. The field may be modeled as a constant or any statistical distribution and is scaled in units of milliseconds.

5. Port Processing

The processing time of a packet in the port buffers may vary as a function of the routed protocol identification and the type of link the node is connected to. When a packet arrives at a node the model will first look for any special case processing times. If no match is found, the default value is used.

The port processing times are entered by selecting the *Edit Times...* button next to the port processing field as shown in Figure 4.2 which will cause a window such as shown in Figure 4.5 to appear which contains the list of all special port processing times entered. By pressing the *Add* button in this window the port processing editing window will appear as shown in Figure 4.6. In this window select the desired type of link and routed protocol identification from the drop down list. The port input and output delays may only be entered as constant values and are scaled in units of milliseconds. When all values have been entered, press the *OK* button as shown in Figure 4.6 and then the *Done* button as shown in Figure 4.5 to set the port processing times for the model.

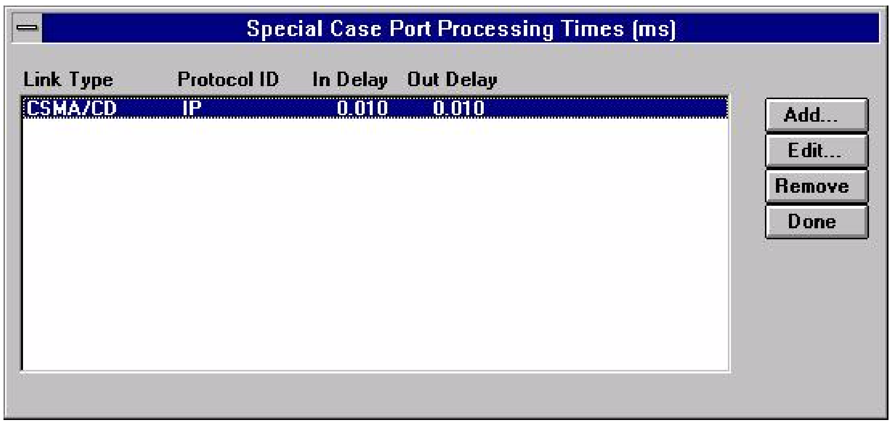


Figure 4.5 Special Port Processing Times Window

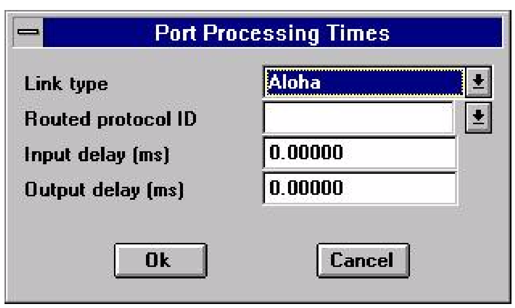


Figure 4.6 Port Processing Times Editing Window

The port processing default input and output delays are set in the default times field. These values may be set only as constants in units of milliseconds. The total input and output buffer size for the entire node are set in the *buffer max* field in units of bytes. These values are the maximum allowable buffer usage across all ports connected to the node. The individual port buffers are set by editing the ports, and will be discussed further in Section E of this chapter.

6. Disk Storage

The fields which affect disk storage are used to model a local hard drive on the node and the processing which occurs in using this disk. If any read or write commands are used in an application on the node these fields must be set. The *Disk* field sets the size of the hard drive to be modeled in units of megabytes. The *Sector* field sets the size of the sectors on the hard drive in units if kilobytes. The *Xfer* field sets the amount of time requires to read one sector of the disk in units of microseconds and may be modeled as a constant or using a statistical distribution. The *Seek* field sets the amount of time needed for the disk to find a file in a sector in units of microseconds and may also be modeled as either a constant or using a statistical distribution.

7. File List

The file list is a list of files that are to be created on the hard drive prior to the beginning of a simulation. All nodes have a file created called GENERAL STORAGE at the beginning of every simulation. This file may be used to read or write an arbitrary set of bytes to or from the hard disk without the necessity of modeling a complete file structure. Files may be created in the file list by pressing the *Add* button as shown in Figure 4.2 which will cause a file detail window to appear as shown in Figure 4.7. In this window a unique file name must be entered, the file size in bytes set, and the option is given to make the file read only. When all entries are made, press the *OK* button.

Figure 4.7 File Detail Window

D. COMPUTER GROUP NODES

While modeling a network, it is common that several nodes will have the same parameters and the same traffic generation patterns. The computer group node may be used to model several computers in this instance and will reduce the clutter in the work area. When running a simulation, each computer defined in the group will be created as an instance of that node. Any application or traffic source connected to the node will be modeled as available on each computer instance. Messages which are destined for the computer group will be parsed out evenly between all computer instances in the group. The computer group node is essentially the same as a computer and communications node with the following exception. It, by default, may only be used as a source or sink for the generation or receipt of message traffic. Also, it may be connected to only one of the multi-access type links in a model and may never be connected to a point-to-point type link.

Computer group nodes may be created from either the toolbar or the *Create* menu. When editing the computer group the computer group parameter window will appear such as shown in Figure 4.8. All of the fields for the computer group node parameter sets are exactly the same as for the computer and communication node with one exception. The computer group node has a *Number-in-group* field which may be set to an integer value to set the total number of computers this node is to model.

E. PORTS

Ports are the arcs or lines created in a model which are used to connect the various nodes to the various types of links. Ports may be created by using either of the connection tools on the toolbar. Connections are made by first selecting a connection tool from the toolbar. After a selection is made, single click on the node for which the port is to be created. Then drag the mouse to the link the node is to be created on and single click again. If the connection is a logical connection, the port will be made. If the connection is not logical, such as connecting a node to a node, nothing will happen.

Ports may be thought of as modeling a portion of the network interface cards within all types of nodes. They have the characteristics of having buffer capacity and are

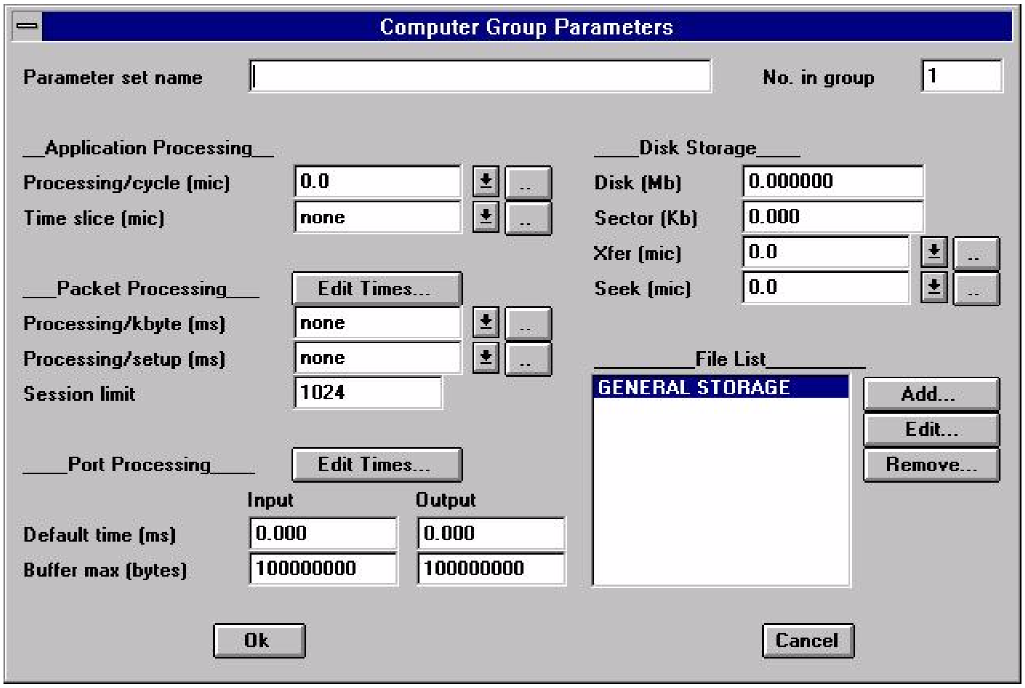


Figure 4.8 Computer Group Parameters Window

used in modeling the port processing delays. Editing of the port is accomplished by either double clicking on the arc or by selecting the arc and then choosing the detail option from the edit menu. When this is done the port parameter window as shown in Figure 4.9 will appear. The buffer capacity for both the input and output buffers is the number of bytes available to hold packets in the port while waiting to be switched on a node. The delay is

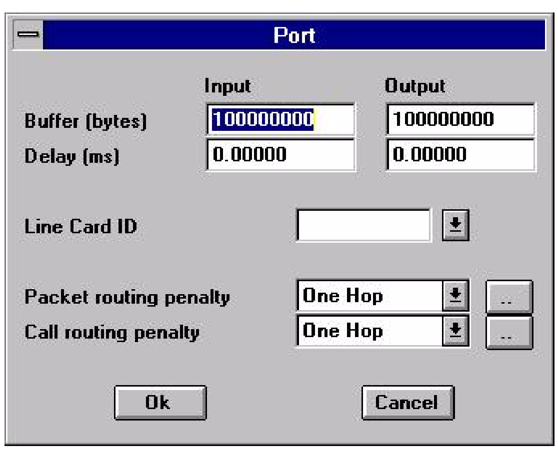


Figure 4.9 Port Parameter Window

by default set to the default port processing time of the node it is attached to or may be edited in this window. The *Line Card Identification* field is only applicable when the port is attached to a router node. This field is used to differentiate between different ports on the router node for simulated switching across a router bus. The packet routing penalty may be defined using a penalty table. This feature will be discussed in more detail in Chapter VIII.

F. PROBLEM STATEMENT

The Naval Postgraduate School's Scientific and Visualization Lab (VisLab) contains a HP 730 workstation that is used as a HTTP server. Due to the increasing popularity of the World Wide Web, there is a concern that increases in the number of HTTP requests to this server will adversely affect the 10Base2 Ethernet network the server resides on. It is estimated that requests to this server may increase by 60 percent within the next year. Additionally, it is desired to determine the average amount of time necessary to process a HTTP request to the server and the effects of upgrading the 1 gigabyte Connor SCSI hard drive installed to a 1 gigabyte Quantum SCSI hard drive.

G. ANALYSIS AND OBJECT MAPPING

The model which will be built to analyze the problem deals directly with the characteristics of the HP 730 and the processes the server performs in responding to HTTP requests. The basic characteristics of the HP 730 were obtained from Introduction to the Visualization Lab home page [Ref. 3]. Data concerning server requests and the file sizes of the home pages most requested was obtained from the Statistics for the VisLab HTTP Server home page [Ref. 4] which covers a time period of 120 days from 09/06/95 to 01/03/96.

1. Modeling of the HP 730 Workstation

The parameters which are used to model the server when building the model are all estimates with the exception of the disk storage values. Hard drive specifications for Connor and Quantum 1 gigabyte SCSI hard drives were obtained from the *Hard Drive Spec* home page [Ref. 5]. Both hard drives were assumed to have 120 sectors. Manipulation of the specifications to map them into C&C node parameters yields the following parameters as shown in Table 4.1.

Manufacturer:	Connor	Quantum
Model:	CFP1080	Atlas
Disk Size (Mb):	1080	1075
Sector Size (Kb):	8789	8748
Data Transfer Time (microseconds):	136636	109248
Seek Time (microseconds):	11000	8500

Table 4.1 Hard Drive Parameters

2. Modeling of HTTP Requests

Figure 4.10 displays in a graphical form the cumulative number of requests made over the 120 day period by hour. The time period of concern is during normal working hours from 0800 to 1600. In order to use this data it must be converted to a form which may be used by a message source to model the interarrival time of HTTP requests. The assumption is made that the requests occur uniformly over the hourly time period as no

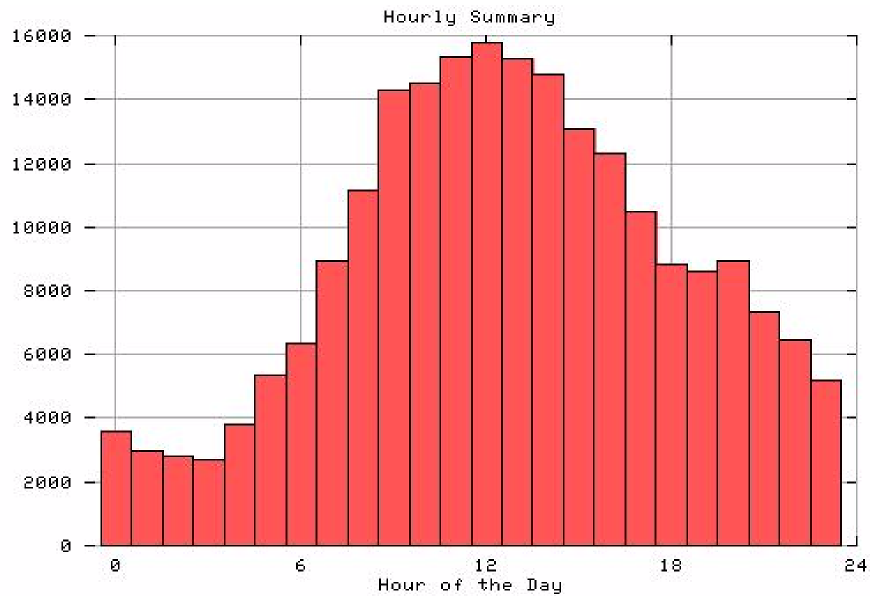


Figure 4.10 Cumulative Hourly HTTP Server Requests

information was available concerning the burstiness of the requests. The average time, in seconds, between requests was calculated, and a normal probability plot of these average times was constructed as shown in Figure 4.11. Based on this plot the average interarrival times for HTTP requests was chosen to be modeled as a normal distribution with a mean of 31.8 seconds and a standard deviation of 3.8 seconds. To model a 60 percent increase in HTTP requests, the number of requests per hour were scaled up by a factor of 60 percent. The average interarrival time for each hour was then calculated. A 60 percent increase in requests was chosen to be modeled as a normal distribution with a mean of 19.4 seconds and a standard deviation of 3.8 seconds. Table 4.2 displays the raw data used and the average interarrival times of packets calculated for each hour.

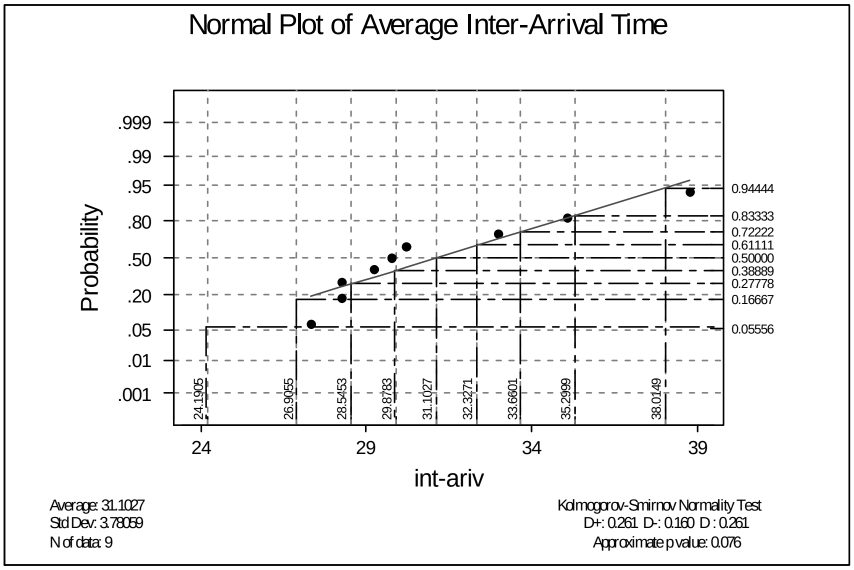


Figure 4.11 Normal Probability Plot of Hourly Average Interarrival Times of HTTP Requests

Data to describe the size of the HTTP requests was not collected, but was estimated to be a uniform distribution with a minimum of 10 bytes and a maximum of 100 bytes. This estimate is based on the number of characters in a typical web server request.

Time	Requests over 120 days	Avg. Interarrival Time (sec)	60% Increase in Requests	Avg. Interarrival Time (sec)
8	11143	38.8	17828.8	24.2
9	14306	30.2	22889.6	18.9
10	14516	29.8	23225.6	18.6
11	15350	28.1	24560.0	17.6
12	15796	27.3	25273.6	17.1
13	15282	28.3	24451.2	17.7
14	14771	29.2	23633.6	18.3
15	13095	33.0	20952.0	20.6
16	12316	35.1	19705.6	21.9

Table 4.2 Hourly Web Server Requests

3. Modeling of the Web Server Application

As stated previously, the process that occurs at the server upon receipt of a HTTP request may be modeled by an application source which reads a file and then creates a message to send back to requesting node based on the size of the file read. The modeling problem which occurs is how to estimate the number of bytes to read for each request. The file request statistics obtained from the Statistics for the VisLab HTTP Server home page [Ref. 4] contained information concerning the number of requests of the most frequently requested home pages. The individual portions of each home page which was still active were downloaded and summed to estimate the total number of bytes to be read. Table 4.3 contains a list of all pages which were active along with their file size and cumulative probability. This information may be used to create a distribution table to model the number of bytes to read upon receipt of a HTTP request.

File Name	Requests over 120 Days	Probability of Request	Cumulative Probability	File Size (Bytes)
photo	1280	0.03	0.03	1295
netiquette	1265	0.03	0.06	2154
pirates	2011	0.04	0.10	6618
npscompu	1045	0.02	0.12	8673
origami	1249	0.03	0.15	9975
edo	2603	0.05	0.20	14494
sfgram	849	0.02	0.22	14857
helos	850	0.02	0.24	16650
cfbush	2165	0.04	0.28	17040
thesis	710	0.01	0.29	17053
mclinks	1337	0.03	0.32	18197
nsgdhome	1566	0.03	0.35	21797
bkandber	1041	0.02	0.37	25875
psu	1936	0.04	0.41	28188
photo_gallery	1378	0.03	0.44	30167
ch1	847	0.02	0.46	30365
direc	1990	0.04	0.50	30420
c4i-pro	820	0.02	0.52	36514
library	1409	0.03	0.55	39625
marine	6579	0.14	0.69	41829
fapapoul	1667	0.03	0.72	62230
opnsrsch	850	0.02	0.74	69609
habitatmc	1090	0.02	0.76	72317
jcking	1028	0.02	0.78	96015
oa2200	724	0.01	0.79	127800
braccio	1326	0.03	0.82	134370
pelopon	908	0.02	0.84	136681
history	989	0.02	0.86	136681
macedon	3887	0.08	0.94	136681
ece	1128	0.02	0.96	149812
maps	1122	0.02	0.98	296816
pictures	1039	0.02	1.00	526065

Table 4.3 File Size and Cumulative Probability Table

H. BUILDING THE MODEL

The building of the model described in the following subsections is done by creating objects or parameter sets one step at a time with detailed instructions on how to complete the step. The completed model should look similar to that shown in Figure 4.12.

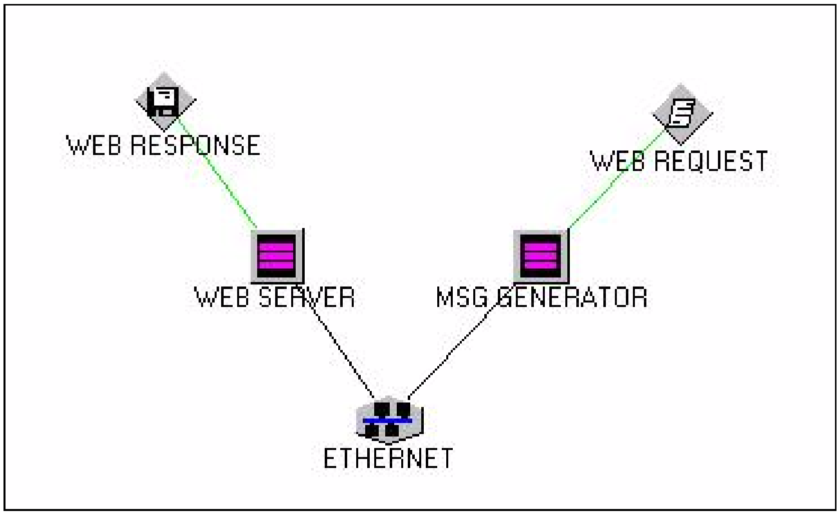


Figure 4.12 Completed Model View

1. Defining Computer and Communications Node Parameters

The parameter sets for the computer and communications nodes are defined first so that they may be applied later when the nodes are created.

a) From the *Define* menu select the *Nodes/Computer and Communications* option.

b) Edit the node parameters to the following parameters:

- Name: HP 730 CONNOR HD
- Source or sink box: checked
- Processing/cycle: 0.02 microseconds
- Time slice: Set to a normal distribution with a mean of 100000 microseconds and a standard deviation of 10000 microseconds. The stream value may be any value from 0 to 99.
- Packet processing: Edit to model a 0.01 millisecond delay for the IP protocol.
- Port Processing: Edit to model a delay of 0.01 milliseconds for the CSMA/CD link and the IP protocol for both input and output delays.
- Default port processing: Set to a delay of 0.05 milliseconds for input and output Buffer Max: Set to 4194304 bytes for both the input and output to model 4Mb total available for each.
- Disk: 1080 Mb
- Sector: 8789 Kb
- Transfer rate: 163636 microseconds
- Seek: 11000 microseconds

c) When all the values have been entered, press the *Done* button at the bottom of the window.

d) Create a second parameter set using the same values as in step b with the following exceptions:

- Name: HP 730 QUANTUM HD
- Disk: 1075 Mb
- Sector: 8748 Kb
- Transfer rate: 109248 microseconds
- Seek: 8500 microseconds

e) When all the values have been entered, press the *Done* button at the bottom of the window.

2. Creating the Ethernet Link

The following steps describe the creation and editing of the Ethernet link:

a) Create a collision link using either the toolbar or the *Create* menu.

b) Edit the link to set the following parameters such as shown in Figure 4.13:

- Name: ETHERNET
- Type: CSMA/CD
- Parameters: 802.3 CSMA/CD 10BASE2

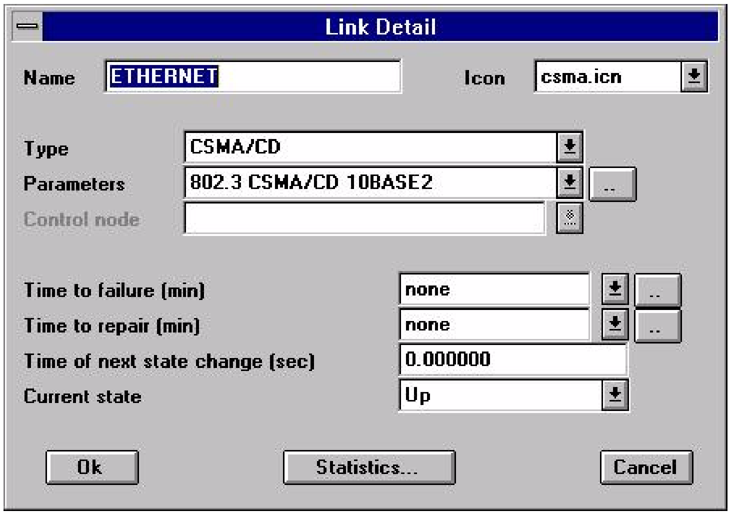


Figure 4.13 Link Detail Window

c) When all values have been entered press the *OK* button.

3. Creating the Web Server and Message Generator Nodes

The following steps detail the methods of creating the C&C nodes to model the web server and a message generator:

a) Create two C&C nodes using either the toolbar or the *Create* menu.

b) Edit one of the nodes to the following parameters:

- Name: WEB SERVER
- Type: Computer and Communications Node
- Parameters: HP 730 CONNOR HD

c) Edit the second node to the following parameters:

- Name: MSG GENERATOR
- Type: Computer and Communications Node
- Parameters: DEFAULT

d) Create the ports to connect the two nodes to the ethernet link using either connection tool on the toolbar, and edit the port attached to the WEB SERVER node such that both the input and output buffers are set to 262144 bytes to model a 256 Kb buffer on the port.

4. Creating Message Source Distributions

Prior to creating the message source for the generation of the HTTP requests, the two distributions which will be used to model the interarrival time of current HTTP requests and a 60 percent increase in requests is created using the following steps:

a) Select the *User Distribution* option on the *Define* menu.

b) In the window which appears press the *Add* button.

c) Edit the distribution to the following parameters as shown in Figure 4.14:

- Name: 0_PERCENT
- Sample: Set to a normal distribution with a mean of 31.8 seconds and a standard deviation of 3.8 seconds. The stream value may be set to any value between 0 and 99.

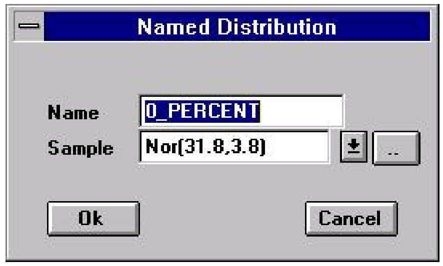


Figure 4.14 O_PERCENT User Defined Distribution

d) When all values are entered, press the *OK* button as shown in Figure 4.14. Then, press the *Add* button again to create a second distribution.

e) Edit the second distribution to the following parameters:

- Name: 60_PERCENT
- Sample: Set to a normal distribution with a mean of 19.4 seconds and a standard deviation of 3.8 seconds. The stream value may be set to any value between 0 and 99.

f) When all values are entered, press the *OK* button to clear this window, and the *Done* button to clear the second window.

5. Creating the HTTP Requests Message Source

The following defines the steps necessary to create the message source used to model the HTTP requests which will be attached to the MSG GENERATOR node in the model:

a) Create a message source using either the toolbar or the *Create* menu and place it near the MSG GENERATOR node.

b) Use either of the connection tools to connect the message source to the MSG GENERATOR node.

c) Edit the message source entering the parameters shown in the following list and also in Figure 4.15:

- Name: WEB REQUEST
- Schedule by: Iteration Time
- Interarrival: Set to the O_PERCENT distribution
- Priority: 1
- Routing Class: Standard

- Transport Protocol: TCP/IP-Sun
- Packetize: 0.01 milliseconds
- Message Size Calculation: Probability Distribution
- Probability Distribution: Set to a uniform distribution with a minimum of 10 bytes and a maximum of 100 bytes. The stream value may be set to any value between 0 and 99.
- Message text option: Copy Message Name
- Destination Type: Set to weighted list. Create a list adding only the WEB SERVER.

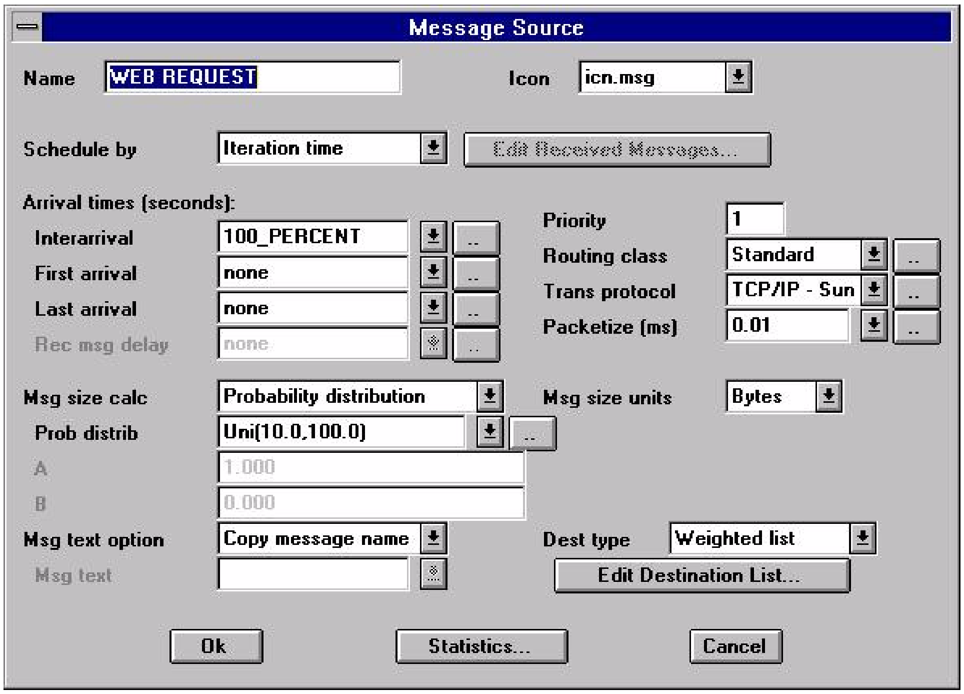


Figure 4.15 Web Request Message Source Parameters

d) When all parameters have been set, press the *OK* button at the bottom of Figure 4.15.

6. Creating a Distribution Table

The distribution table will be used in a read command to model the number of bytes read upon the receipt of the WEB REQUEST message.

a) From the *Define* menu select the *Tables* option.

b) In the tabular distribution window which appears select the *Add* button.

c) Edit the fields in the table detail window to the following values as shown in Figure 4.16:

- Name: WEB_READ
- Table type: discrete
- Probability and Values: Edit the probability and values pairs such that the probability values are set to the cumulative column and the values are set to the file size column of Table 4.3.

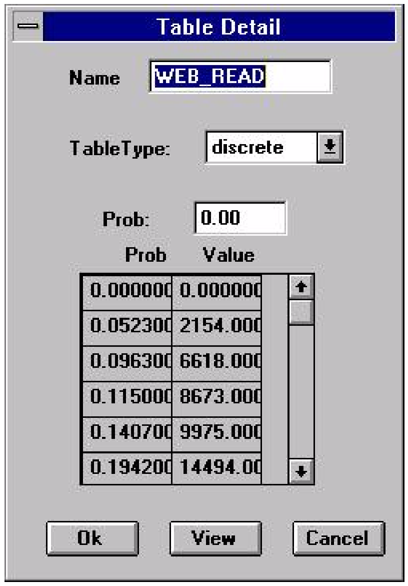


Figure 4.16 WEB_READ User Defined Table

d) When all values are entered in the table detail window, press the *View* button at the bottom of the window. A plot of the distribution entered as shown in Figure 4.17 will appear.

e) Clear the plot of the distribution; press the *OK* button at the bottom of the table detail window; and press the *Done* button at the bottom of the tabular distribution window.

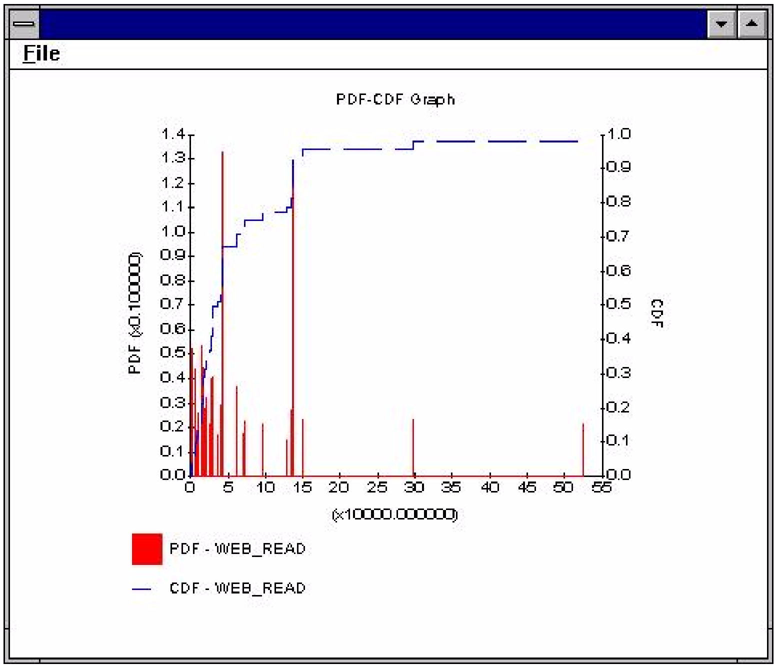


Figure 4.17 WEB_READ Discrete Distribution

7. Creating the Application Read Command

The read command will be used in a application source to execute a read to the local disk of the WEB SERVER node.

a) Double click on the WEB SERVER node icon to bring up the node detail window and press the *Command* button in this window.

b) In the command repertoire window which appears press the *Add* button.

c) In the library selection window that appears select the *Read File* option.

d) Set the parameters of the read command window to the values shown in the following list and also shown in Figure 4.18.

- Read command name: WEB PAGE READ
- File to access: GENERAL STORAGE
- File modification method: Do not modify file
- Bytes read calculation: Probability distribution
- Probability distribution: WEB_READ

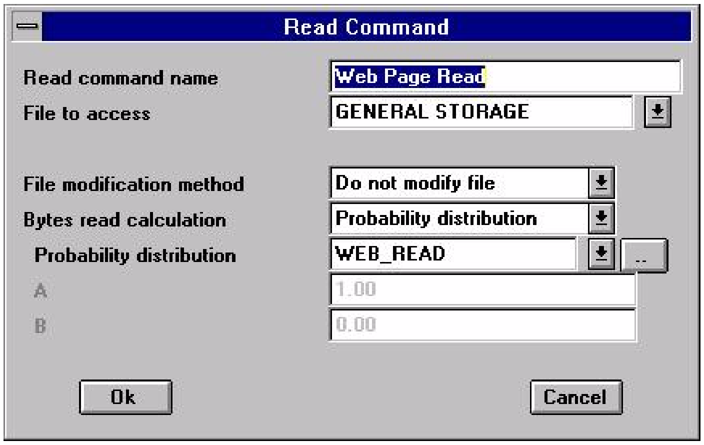


Figure 4.18 WEB PAGE READ Command

e) When all values have been entered, press the *OK* button in the read command window; press the *Done* button in the command repertoire window; and press the *OK* button in the node detail window to clear these windows.

8. Creating the Application Answer Command

The answer command will be used in an application to model the generation of the message to be sent based on the size of the file read using the WEB PAGE READ command.

a) From the *Define* menu choose the *Global Commands* option.

b) In the command repertoire window which appears select the *Add* button.

c) In the library selection window select the *Answer Message* option from the list.

d) Set the answer command fields to the following values as shown in Figure 4.19:

- Name: WEB ANSWER
- Priority: 1
- Routing class: Standard
- Transport protocol: TCP/IP - Sun
- Packetize: 0.01 milliseconds
- Message size calculation: (A x File Bytes) + B
- A: 1.000
- B: 0.000
- Message text option: Copy message name

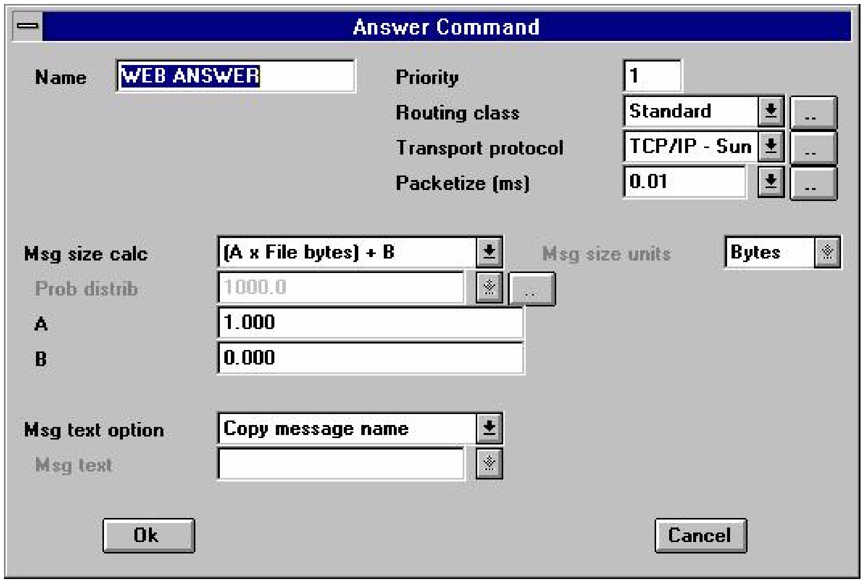


Figure 4.19 WEB ANSWER Command

e) When all values are entered, press the *OK* button at the bottom of the answer command window, and the *Done* button at the bottom of the command repertoire window to clear these windows.

9. Creating the Web Server Application Source

The application source will utilize the commands previously created to model the processes that will occur on the WEB SERVER node upon receipt of a HTTP request.

a) Create an application source using either the toolbar or the *Create* menu and place the source near the WEB SERVER node.

b) Using either connection tool on the toolbar connect the application source to the WEB SERVER node.

c) Edit the application source fields to the following values as shown in Figure 4.20:

- Name: WEB RESPONSE
- Schedule by: Received message
- Received message delay: none

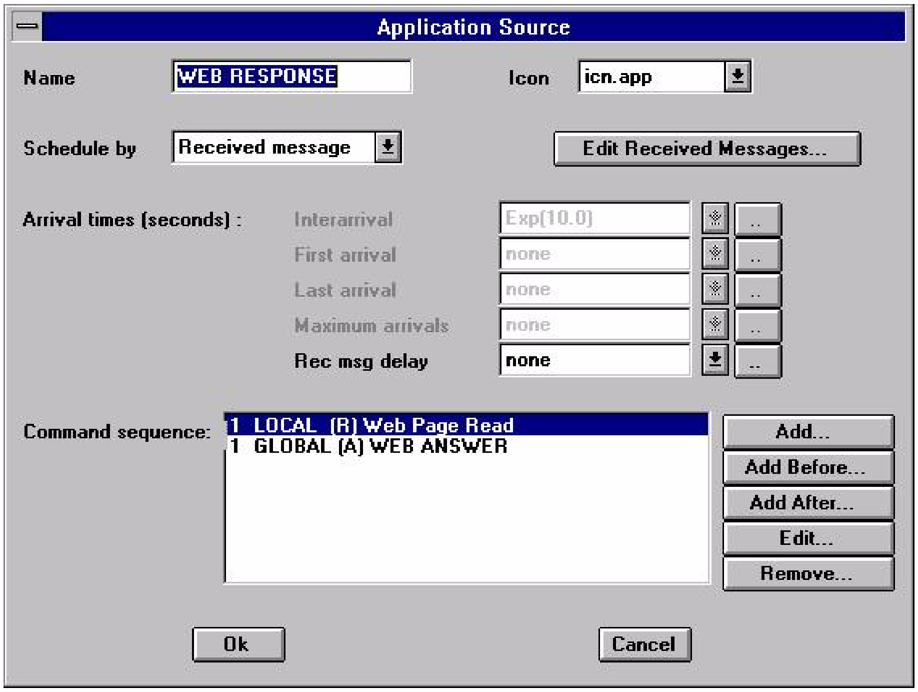


Figure 4.20 WEB RESPONSE Application

d) Using the *Edit Received Messages...* button add the WEB REQUEST message to trigger this application upon the receipt of only one message.

e) In the command sequence portion as shown in Figure 4.20, press the *Add* button. A window as shown in Figure 4.21 will appear. Using the drop down list, select the WEB PAGE READ command and set the number of executions to 1.

Figure 4.21 Command Name Detail Window

f) Press the *OK* button in the command name detail window, and then press the *Add* button again. Select the WEB ANSWER command, set the number of executions to 1, and press the *OK* button.

g) Press the *OK* button at the bottom of the application source window to complete building the application.

10. Verifying and Saving the Model

The final step in completing the model is to select the *Verify Model* option on the *Simulate* menu to check for any errors. The message bar should indicate *No verification errors detected*. If errors in the model are present a window will appear showing the errors. Any errors must be corrected prior to running a simulation. Finally, save the model selecting the *Save* or *Save As* option from the *File* menu.

I. SIMULATION AND RESULTS

Three simulation runs will be used to collect the data needed to analyze the problem. The following reports need to be selected prior to running the simulation:

- Links: Channel Utilization for the ETHERNET link
- Application Sources: Application Run Length for the WEB RESPONSE application.

The length of each run in the simulation should be set to 3600 seconds to simulate an hour of simulated requests to the server. After completion of the first run, a text file named report.1 will be created. The name of this file should be changed to prevent overwriting the file during the subsequent simulations. Before starting the second simulation, edit the MSG GENERATOR message source so that the interarrival time is set to the 60_PERCENT user-defined distribution. Run the second simulation, and again change the name of the report generated at the end of the simulation so that it will not be overwritten by the report generated by the third simulation. Before running the third simulation, edit the MSG GENERATOR message source again to set the interarrival distribution back to the 0_PERCENT user-defined distribution, and edit the WEB SERVER node to change the parameters field to HP 730 QUANTUM HD. After these changes have been made run the third simulation.

The simulation results for the current level of HTTP requests to the server showed that these requests and web pages being sent represent very little utilization of the network (0.19%). An increase of 60 percent in HTTP requests will increase the utilization of this network minimally (0.31%). The average delay in processing a HTTP request in the simulation was 256 milliseconds for simulations run modeling the Connor hard drive irrespective of the traffic level. The simulation run with the Quantum hard drive showed a decrease in processing the requests to 187 milliseconds. This decrease in processing times was expected due to the faster seek time and transfer rate of the Quantum hard drive over the Connor hard drive.

The modeling of the problem was chosen to be done to isolate the effects of HTTP requests alone on the network and server. Other traffic on the network which was not modeled would add greater validity to the model if other parameters such as the latency for packets across the network were desired to be determined.

V. APPLICATION SOURCES AND COMMANDS

A. INTRODUCTION

The goal of this chapter is to introduce the user to the application source and the commands available in COMNET III to script the processes which may occur in an application source. The use of application sources will then be shown in a model which deals with changing a peer-to-peer network into a client-server architecture.

B. APPLICATION SOURCES

Application sources are used in COMNET III to model workload on a node and to create more complex processes on a node than can be defined using message, response, and session sources. The processes which occur in an application source are scripted using commands. When an application source is scheduled to run on a node, the node will sequentially execute each command defined until the entire application has been completed. Application sources may only be used with computer and communications nodes, computer group nodes, and router nodes.

The scheduling of an application source may occur in three different ways. Iteration time scheduling is one method of time-based scheduling which sets the time from the start of one instance of an application to the start of the second instance. The use of iteration time scheduling may cause the multiple overlapping application to be present on a node at the same time. If only one application source is needed to be modeled on a node at a time, then delay time scheduling should be used. In this second method of time-based scheduling, the interarrival time is used to set the time between the end of one instance of the application and the start of the next instance. The final method of scheduling is by received message. In this case the application will trigger upon receipt of a message of the required text at the node.

Application sources may be created either by using the toolbar or the *Create* menu. Editing of the application source will cause the application source window shown in Figure 5.1 to appear. The parameters fields shown are used to set the scheduling of the application source, and to script the commands which are to be executed within the application source. The fields shown in Figure 5.1 have the following uses:

- Name: Used to set a unique name to the application source to distinguish it from other application sources in the model.
- Schedule by: Used to set the method of scheduling to either delay time, iteration time, or received message.
- *Edit Received Messages* ... Button: The button becomes active when received message scheduling is used. It allows creating a list of the number of messages required and there associated text which may be used to trigger the application.
- Interarrival: Used with both time-based scheduling to set the interarrival time between application instances. This field may be set to a constant value or to any statistical distribution.
- First arrival: Used with time-based scheduling to set in simulation time when the first instance of the application is to occur. This field may be set to none, a constant value, or to any statistical distribution.
- Last arrival: Used with time-based scheduling to set in simulation time when the last instance of an application will occur. This field may be set to none, a constant value, or to any statistical distribution.
- Maximum arrivals: Used in time-based scheduling to limit the number of instances of an application source which may occur in one simulation repetition. This field may be set to none, a constant value, or to any statistical distribution.
- Received message delay: Used with received message scheduling to set a delay in the start of the application source after the receipt of the message which triggers the source. This field may be set to none, a constant value, or to any statistical distribution.
- Command Sequence: This field is used to script the commands which will be executed in the application source. The buttons to the right of this field allow editing the order in which commands appear and the number of times each

command is to execute. Pressing the *Add*, *Add Before*, or *Add After* buttons will bring up the window shown in Figure 5.2 used to enter commands.

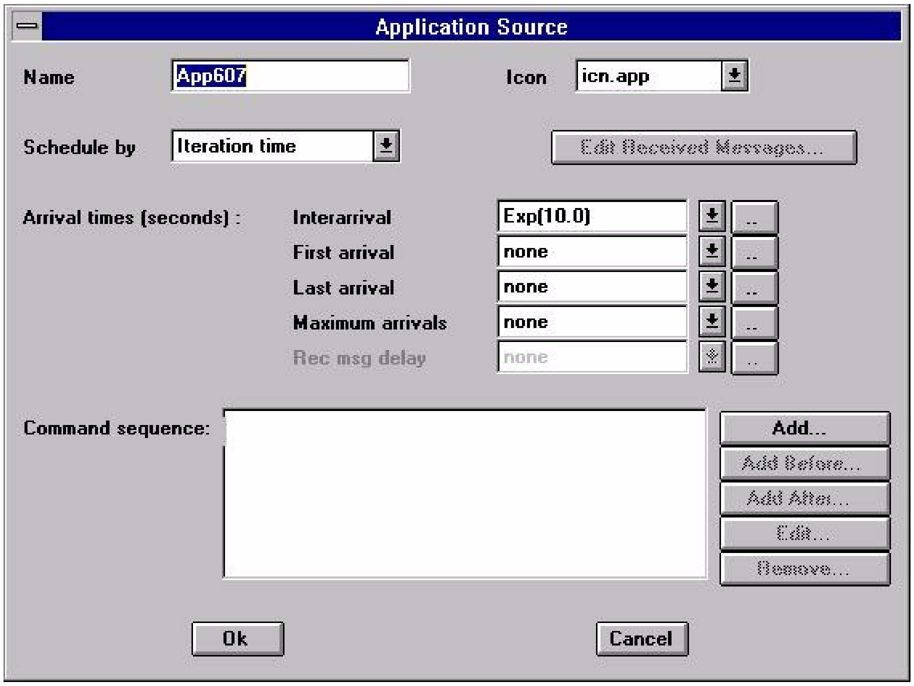


Figure 5.1 Application Source Window

Figure 5.2 Application Command Name Detail Window

C. APPLICATION COMMANDS

Application commands are used to script the processes which may occur within an application source. They may be defined globally within a model by using the *Global Commands* option in the *Define* menu, or locally by using the *Command* button in the dialog box of the computer and communications node, computer group node, or router node. Global commands are available for use by any application source created in a model. Local commands reside only on the node they are created on and become available for use in an application source only after the source is connected to the node. A global and local command within the same model may be given the same name. When executing the application's command sequence during a simulation, the application source first checks the list of local commands available on the node. If the command is not found, it will then check the global command list. Thus, a local command takes precedence over a global command with the same name.

When creating a command, whether local or global, the first window that will appear is the command repertoire window shown in Figure 5.3. Commands are created by selecting the *Add* button in this window which will activate the library selection window shown in Figure 5.4. The library selection window contains a list of all the commands which may be constructed in COMNET III. By selecting a command from this window with a single click of the mouse, a window will appear which enables the editing the parameters of the command.

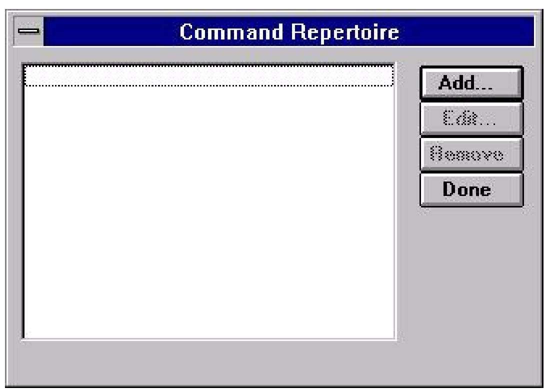


Figure 5.3 Command Repertoire Window

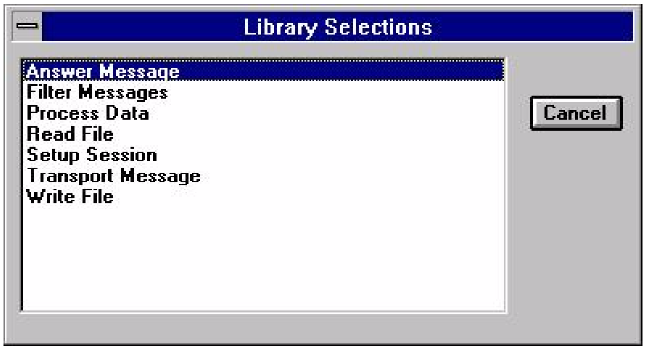


Figure 5.4 Library Selection Window

1. Answer Message Command

The answer message command is used to create a message which is to be sent back to node that triggered the application source. The logic which occurs within the COMNET III application upon the execution of this command is the same as for a response source which was covered in Chapter III. When the answer message command is selected, the window shown in Figure 5.5 will appear. The fields which may be edited in this window have the same function as the fields for the response source with the following exception. The message size of a answer message command may also be based upon the size of the first file read within an application source by a read command.

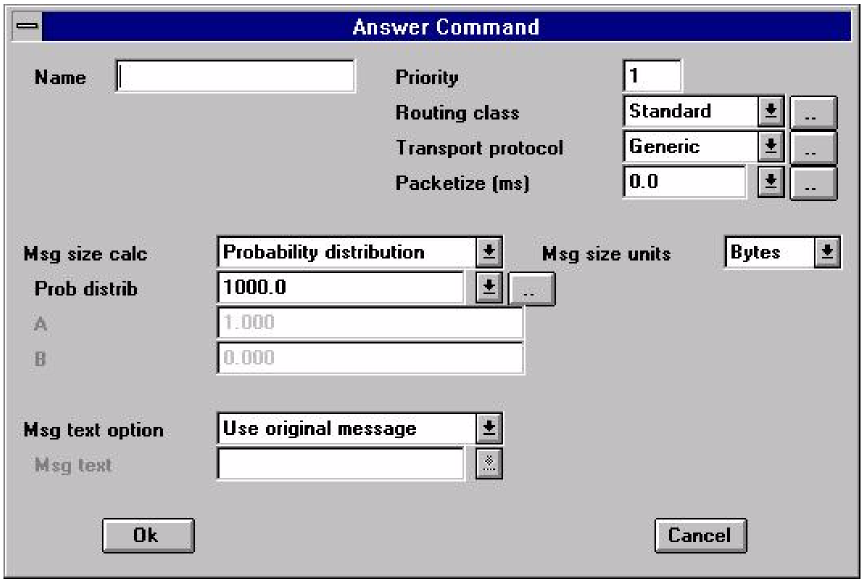


Figure 5.5 Answer Command Window

2. Filter Message Command

The filter message command is used to suspend the execution of an application source until a message is received which fulfills the requirements of the filter [Ref. 2 p. 46]. If a message received at a node may be used to fulfill the requirements of the filter, or trigger an application source, the filter command takes priority over the creation of the new application instance. This command is very useful in modeling of a client-server architecture where an application source will send a request to a node and must then wait for the response before the process may continue. Additionally, if situations arise in the modeling of an application where a message to be sent needs to be based on the size of a message received and the application source is not scheduled by received message, a filter command must be used to capture the received message for later use by an answer or transport command. Errors will occur when verifying the model if the filter command is not used in this situation.

When a filter command is created, the window shown in Figure 5.6 will appear. The window allows entering a unique name to the command and setting the text of the received message the filter is waiting for. The latter may be accomplished by choosing the text from the drop down list or by explicitly typing in the required text. The “Retain previous received message (if any)” option is useful for calculating the round-trip time for a filtered message. To better illustrate this, assume that a node sends a message request to a server requesting a database file located on another host system. In this scenario, the server is the connecting link between the workstation and the host (a typical client/server application). The application source then waits for the database file to be returned. When the database file returns to the server, the application source answers the request using the original workstation message request, and send the file to the workstation. It is important to note that when the application source sends the file to the workstation the message text option needs to be set to “use original message in order to calculate and track the total round-trip time for the original workstation request.

The “Only accept msg responding to this app” option restricts the filter command to respond to messages only from specific application instances that were triggered by earlier messages from the same application that has this filter command. The purpose of this switch is to bind two application instances to interact in a sequential way.

Figure 5.6 Filter Command Window

3. Process Data Command

The process data command may be used in an application source to simulate workload on a node, or as a spacer to separate the execution of commands by a certain amount of time. This command is scaled in units of number of cycles. The amount of time necessary to complete the cycles is based on the processing/cycle value set on the node the application source is connected.

When a process data command is created the window shown in Figure 5.7 will appear. The fields of this window allow entering a unique name for the command and offer three different methods to determine the number of cycles the command is to execute shown in the following list:

- Based on a probability distribution
- Based on message size: This method uses the linear relationship

  Number of Cycles = A * Message Size + B

  to determine the number of cycles. To use this option the application must either use received message scheduling or the processing command must be preceded by a filter command.
- Based on file size: This method uses the linear relationship

  Number of cycles = A * Number of Bytes Read + B to determine the number of processing cycles. To use this option the process command must be preceded by a read command.

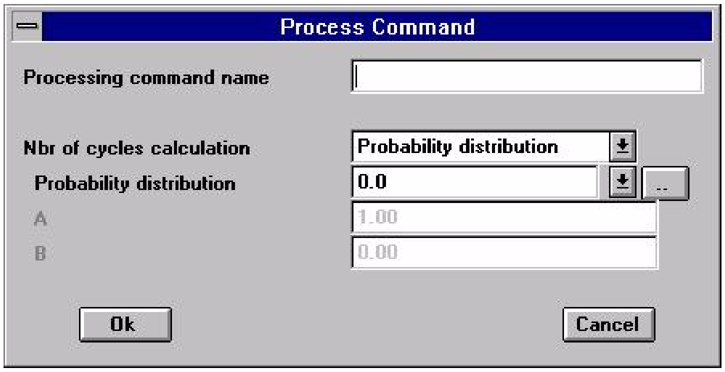


Figure 5.7 Process Command Window

4. Read File Command

The read file command is used in an application source to model the time delay associated with reading a file from a node's local disk. When a read command is invoked, the following logic is applied to the execution of the command [Ref. 2, p. 50, 51]:

1. The file to be read is locked for exclusive use by the read command.
2. The time duration of the command is calculated based upon the disk parameters of the node.
3. The nodes processor is made busy for the time calculated in Step (2).
4. After completion of the delay the file information is updated, and the file is unlocked.

Creation of a read command causes the window shown in Figure 5.8 to appear. The fields of this window allow specifying a distinct name to identify the command, and the file to access on the node's disk. The file modification field allows the following options:

- Do not modify: File size is unchanged after a read command

- Decrement file: File decrements in size by the number of bytes read.
- Delete file: The file is deleted after reading the required number of bytes.

The bytes read calculation may be based on any of the following options:

- Entire file: The number of bytes the file has defined are read.
- Probability distribution
- Based on message size: This option may only be selected if the application uses received message scheduling. The size calculation is based on the liner relation ship

Bytes Read = A * Message Size + B.

- Based on file size: If a previous read command has been executed in an application source, subsequent read commands may be based on the number of bytes read by the first read command using the linear relationship

Bytes Read = A * First Number of Bytes Read + B.

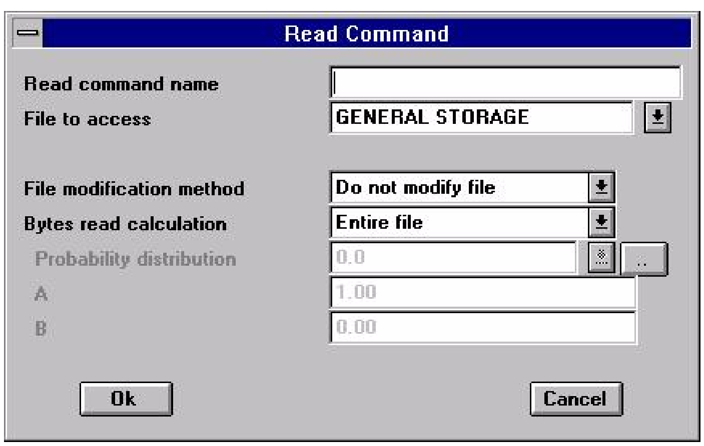


Figure 5.8 Read Command Window

5. Setup Session Command

The setup session command is used to model the establishment of switched or permanent virtual circuits or for modeling any connection oriented messages. The logic applied when a setup session command executed is the same as for a session source which was described in Chapter III.

Creation of a setup session command will cause the window shown in Figure 5.9 to appear. The parameter fields associated with the setup command are the same as those of a session source.

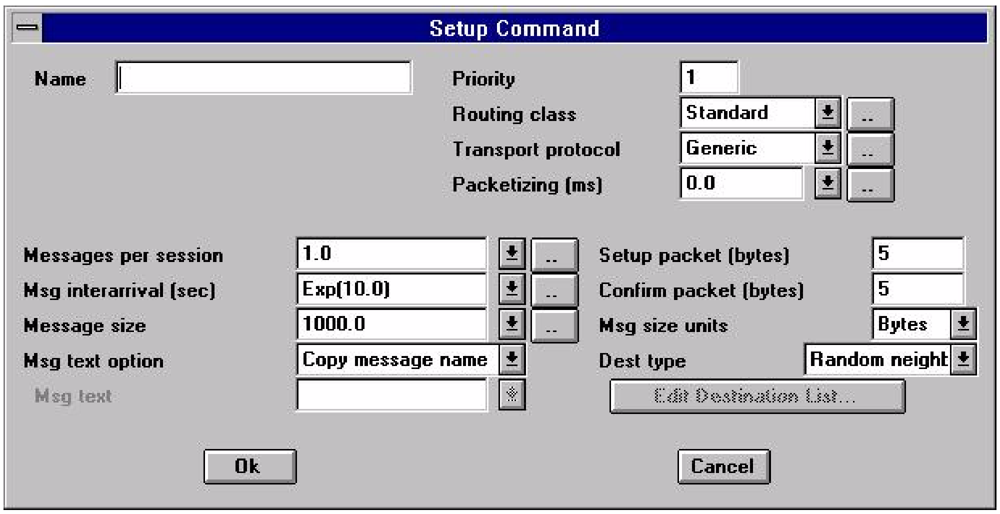


Figure 5.9 Setup Command Window

6. Transport Message Command

The transport command is used to model connectionless data transport across a network. The logic which applies upon the execution of a transport command is the same as applied to a message source which was described in Chapter III.

Creation of a transport command will cause the window shown in Figure 5.10 to appear. The parameters associated with the transport command are the same as for a message source with the exceptions that the scheduling method is not included, as this is done by the application source, and the message size calculation may also be based on the size of a file previously read.

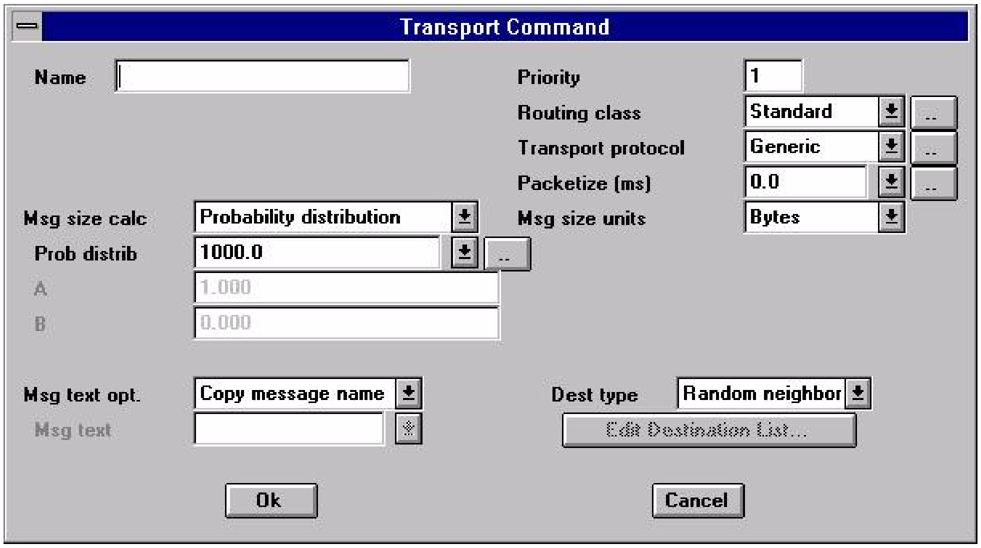


Figure 5.10 Transport Command Window

7. Write File Command

The write file command is used to model the time delay associated with writing a file to a node's local disk. The logic applied by the program when a write command is executed is the same as for a read command.

The creation of a write command will cause the window shown in Figure 5.11 to appear. The fields in this window allow specifying a unique name for the command, and the file to access on the local disk. The three file modification methods available are as follows:

- Append: The file size is increased by the number of bytes wrote to the file.
- Replace: The file is replaced by the number of bytes written.
- Update: The file size is not changed, but the delay used to write the file is incurred on the node.

The three options available to determine the bytes to write are by probability distribution, based on message size, or based on file size. These three options are executed in the same manner as for a read command.

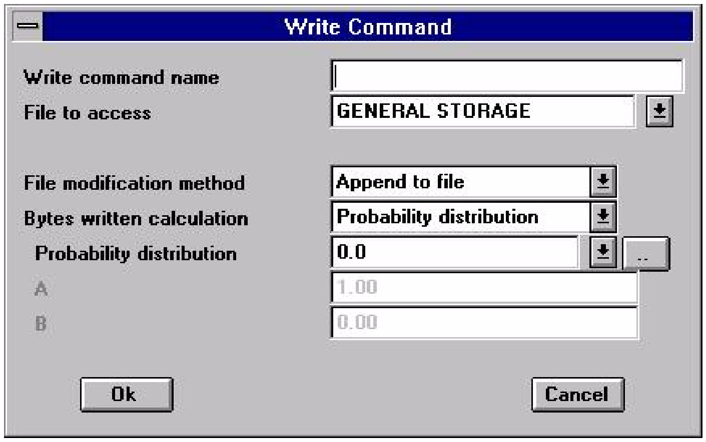


Figure 5.11 Write Command Window

D. PROBLEM STATEMENT

The Systems Technology Lab has seven Pentium 90 MHz personnel computers connected to a 10Base2 ethernet network in a peer-to-peer arrangement. These computers are commonly used by students for thesis writing and for preparing presentations. The major problem with this network configuration is the time spent in configuration management of the applications which reside on each computer. This has required the reloading of all applications on each computer at the end of every quarter. Students have also voiced complaints about the lack of disk storage space available in the lab for maintaining their files.

To alleviate configuration management problems and to create disk storage space the lab has purchased a Pentium 90 MHz computer which has four processors and multiple 1 Gbyte hard drives, and is planning to run this computer using the Windows NT^{tm} operating system. This computer will be used as both an application and file server. The lab desires an estimate of the increase in network loading and collisions during peak usage hours, and estimates of the application delays on the application and file server.

E. ANALYSIS AND OBJECT MAPPING

The modeling of a shift of a peer-to-peer network to a client-server architecture requires not only the modeling of the hardware which makes up the network, but also estimates of the most commonly used programs used in the lab which will now generate additional traffic due to file requests and the transfer of applications across the network. Four applications can be created to model these changes to a client-server architecture.

- An application server application which will read a program from the application server's hard drive and transfer a copy of the program to the requesting computer.
- A file request application which reads files from the file server and transfers the file to the requesting computer.
- A save application which models the saving of files to the file server hard drive.
- A user application which models the behavior of users in requesting programs, requesting files, saving files, and printing documents.

1. Modeling of the Application Server's Application Source

The application server's application source may be constructed by creating a simple model of a read command followed by an answer command. Upon receipt of a request for a program at the application server, the file requested is read from the server's hard drive, and a copy of the program is sent across the network to the computer where the request was generated.

Instead of modeling the entire file structure of the application server, the program to be read upon receipt of a request was chosen to be modeled using a probability table. This choice was made to simplify the modeling for the entire problem as the requests for a program cannot be modeled to specify which file to read. The specification of reading a file may only be done in an application source which implies that an application source for every program accessible on the server would need to be created. Estimates for the most commonly used programs in the lab indicated that the Microsoft Office Suite were used to the greatest extent. To estimate the file size which would need to be read by the application server the programs were individually launched and the amount of memory the program occupied was determined using the System Information Utility of the Norton Utilities for Windows 95[tm] application. The amount of memory required for each application was then used to estimate the file size to read. The individual application file sizes along with the estimated probabilities of the request being for that application are summarized in Table 5.1. From the file size read, an answer command may be created to transport the program to the requesting node using the TCP/IP-Microsoft transport protocol with a 1 millisecond packetizing delay.

Application	Probability	Cumulative Probability	File Size (bytes)
Excel	0.2	0.2	637000
PowerPoint	0.3	0.5	1370000
Word	0.5	1.0	1440000

Table 5.1 Program Cumulative Probabilities and File Size

2. Modeling of the File Server’s File Request Application Source

The process that occurs on the file server upon receipt of a file request may be modeled using a processing command, a read command, and an answer command. The processing command models the processing necessary to find a file in the file allocation table based on the size of the file request message with no scaling of the message size. The read command models the reading of the requested file from the file server’s hard drive. Again, an entire file structure of the server was not constructed to simplify the modeling of the problem. File sizes to be read are estimated to follow a normal distribution with a mean of 150000 bytes and a standard deviation of 50000 bytes. Upon completion reading the file, an answer command is used to transport the file to the requesting computer using the TCP/IP-Microsoft transport protocol and a packetizing delay of 1 millisecond.

3. Modeling of the File Server’s Save Application Source

The modeling of the save application source may be performed by the use of a write command. Processing of the packets which make up the message is performed by the node’s processor upon receipt of the packet. A write command based on the size of the message will model the writing of this message to the file server’s hard drive.

4. Modeling of the Personal Computer’s Application Source

To model the behavior of the users on the personal computers during peak usage, it was estimated that a single computer remained vacant for a time period which followed a uniform distribution ranging from 5 to 15 minutes after a user completed work. Also, the assumption was made that only one program was in use on the computer at a time. In order to randomize the time that the first instance of an application occurred on each computer, the first arrival was estimated to follow a uniform distribution and occur in the range of 0 to 10 minutes of simulation time.

The first operation expected of the computer users is to request the launch of an program from the application server. This action is modeled as a transport command where the message size is estimated to be a constant of 125 bytes in length. The computer must then wait for the application server's response which implies a filter command be used to suspend the application source. Upon receipt of the copy of the program the computer processes for a period of time based on the size of the message which is modeled using a processing command.

To model the worst case in traffic loading on the network, the assumption is made that the computer user then requests a file from the file server. This action may also be modeled using a transport command with the message size estimated to be 100 bytes in length. Again, the application source must wait for the file to be transferred which again implies the use of a filter command to suspend the application source.

The application source resumes executing upon receipt of the file requested. It was assumed that the computer user would then work for 15 to 20 minutes and save their work. The 15 to 20 minutes of processing may be modeled using a processing command set to a uniform distribution scaled to the processing/cycle value of the node to achieve the time delay needed. The save action may be modeled by a transport command to the file server which transfers a file scaled which is scaled to be 10 percent greater than the file earlier retrieved.

Upon completion of saving the file, it was again assumed the computer user would again process for 15 to 20 minutes which may be modeled in the same fashion as before. The user then requests printing of the document and saves the document. The printing of the document to the laser printer located in the lab may be modeled by a transport command with the message size scaled to a factor of 20 percent greater than the original file size. The save command may be modeled as the previous save but with the message to be transferred also scaled 20 percent greater than the original message received from the file transfer.

All messages which will be created by this application use the TCP/IP-Microsoft transport protocol with the exception of print requests to the laser printer which use the NCP/IPX transport protocol. All messages are also assumed to have a packetizing delay of 1 millisecond.

5. Modeling of the Network Computers

The computer purchased by the Systems Technology Lab to be used as both the application and file server has multiple processors. However, COMNET III currently does not have an object available to model a multiprocessor computer. The decision was made to model the application and file server as two separate machines using computer and communications nodes. The values of the parameter set used to describe the servers are given in Section F of this chapter.

The seven personal computers used in the lab could be modeled separately as seven computer and communications nodes or by using a single computer group node. The latter option was chosen to minimize clutter in the work area. The values of the parameter set used to describe the personal computers are given in Section F of this chapter.

The laser printer in the lab may also be modeled using a computer and communications node. Since the laser printer is used in the model only as a destination

F. BUILDING THE MODEL

This section provides step by step instructions for each object which needs to be created to model the problem. Figure 5.12 shows a view of the completed model topology.

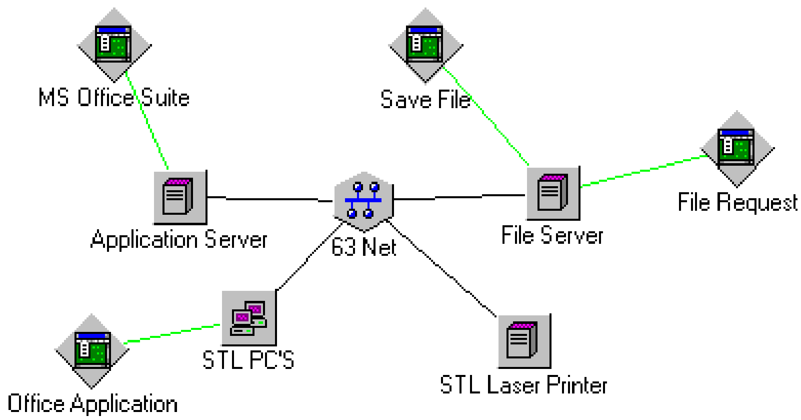


Figure 5.12 Model Topology

1. Defining the Application and File Servers Parameter Sets

The following instruction define the steps for building the parameter set to be used in modeling the application and filer server:

a) From the *Define* menu select the *Nodes Parameters/Comp and Comm Option.*

b) In the computer and communications node parameters window which appears select the *Add* button.

c) Edit the fields of the node parameter window to the following:

- Parameter set name: NT Server
- Processing /cycle: .0111 microseconds
- Time slice: Set to a uniform distribution with a minimum value of 100 and a maximum value of 1000000 microseconds. The streams value may be set to any integer value between 0 and 99.
- Packet Processing: Use the Edit Times... button to create a delay of 0.01 milliseconds for the IP protocol.

- Port Processing: Use the Edit Times... button to create an input and output delay of 0.01 milliseconds for the IP protocol on a CSMA/CD link.
- Default time: Set to 0.05 milliseconds for both input and output buffers.
- Buffer max: Set to 4194304 bytes for both the input and output buffers.
- Disk: 1024 Mb
- Sector: 0.5 KB
- Xfer: 13 microseconds
- Seek: 12000 microseconds

d) When all values have been entered, press the *OK* button at the bottom of the node parameter window, and then press the *Done* button at the bottom of the computer and communication node parameter window.

2. Defining the Systems Technology Lab Computer Parameter Sets

The following steps define the creation of the parameter set which will be used to model the seven personal computers located in the Systems Technology Lab:

a) From the *Define* menu select *Node Parameters* and highlight the *Node Components* and click on the *Edit* button. Highlight the *Computer Group Parameters* option and click the *Edit* button.

b) In the computer group parameters window which appears select the *Add* button.

c) Edit the fields of the computer group parameters window that appears to the following:

- Parameter set name: STL PC’s
- Processing/cycle: 0.0111 microseconds
- Time slice: none
- Packet processing: Use the Edit Times button to set a 0.01 millisecond delay for IP packets and a 0.05 millisecond delay for IPX packets.
- Port Processing: Use the Edit Times button to create for the CSMA/CD link type a 0.01 millisecond input and output delay for IP packets, and a 0.05 millisecond input and output delay for IPX packets.

- Default time: Set to 0.05 milliseconds for both the input and output buffers
- Buffer max: Set to 4194304 bytes for both the input and output buffers.
- Disk: 540 MB
- Sector: 0.5 KB
- Xfer 13 microseconds
- Seek: 12000 microseconds

d) When all values have been entered, press the OK button at the bottom of the computer group parameter window, and then press the *Done* button to clear the remaining window.

3. Building the Network Topology

The following section defines the steps necessary to create each object needed to build the model topology and the parameters associated with these objects:

a) Create a collision link icon using either the toolbar or the Create menu. Edit the link's parameters to the following:
   - Name: 63 Net
   - Type: CSMA/CD
   - Parameters: 802.3 CSMA/CD 10BASE2

b) Create three computer and communication nodes which will be used to represent the application server, file server, and the laser printer.

c) Edit the node to be used as the application server to the following parameters:
   - Name: Application Server
   - Type: Computer and Communications Node
   - Parameters: NT server

d) Edit the node to be used as the application server to the following parameters:
   - Name: File Server
   - Type: Computer and Communications Node
   - Parameters: NT server

e) Edit the node to be used as the application server to the following parameters:

- Name: STL Laser Printer
- Type: Computer and Communications Node
- Parameters: Default

f) Create one computer group node and edit the node to the following parameters:

- Name: STL PC's
- Type: Computer Group
- Parameters: STL PC's

g) Using either of the connection tools on the toolbar, connect all of the nodes created to the 63 Net link. Edit the port extending to each node so that the input and output port values are set to 262144 bytes.

4. Creating the Personal Computer Application Source

The application source which will run on the personal computers will run on the computers will be built by first defining the commands which will be used in this application, and then creating the application source.

a) Define a global transport command and edit the parameters of this field to those shown in the following list. Figure 5.13 displays the completed command.

- Name: MS Office Request
- Message Size Calculation: probability Distribution
- Probability distribution: Set to a constant value of 125 bytes
- Message text option: Copy message name
- Priority: 1
- Routing Class: Standard
- Transport protocol: TCP/IP-Microsoft
- Packetize: 1.0 milliseconds
- Destination Type: Set to a weighted list and create a list containing only the Application Server node.

b) When all values have been entered press the *OK* button as shown in Figure 5.13

and then define a global filter command.

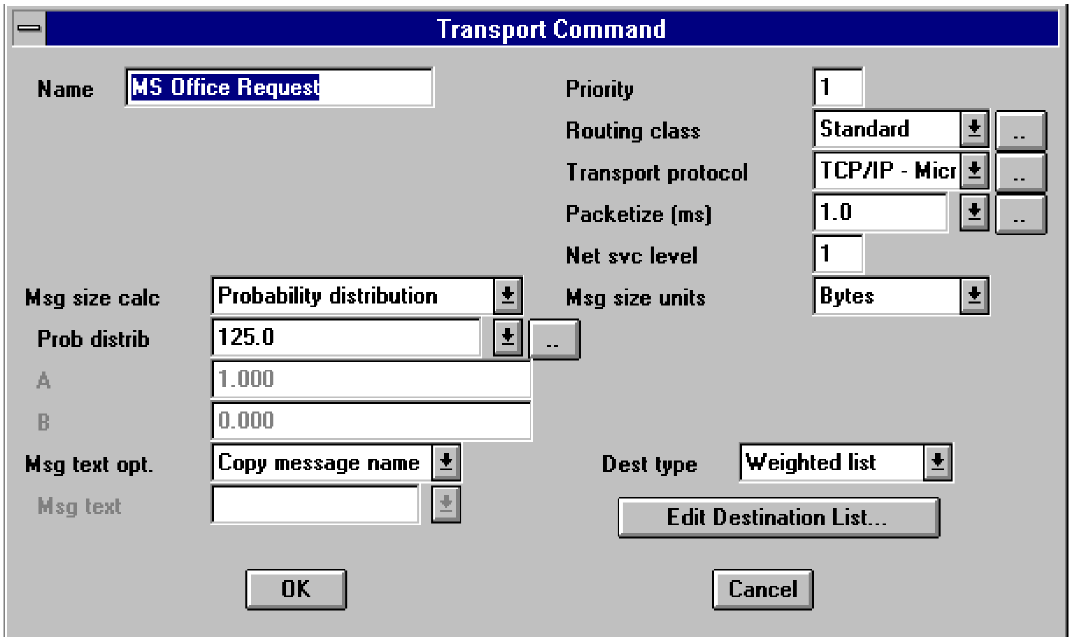


Figure 5.13 MS Office Request Transport Command

c) Edit the filter command parameters to those of the following list and shown in Figure 5.14

- Filter Command Name: MS Office Wait
- Required message text: Set to MS Office Answer

Figure 5.14 MS Office Wait Filter Command

d) When all values have been entered press the *OK* button as shown in Figure 5.14 and then define a global processing command.

e) Edit the processing command to the value defined in the following list and also shown in Figure 5.15.

- Processing Command Name: MS Office Processing
- Number of cycles calculated: (A x Msg Bytes) + B
- A: 1.00
- B: 0.00

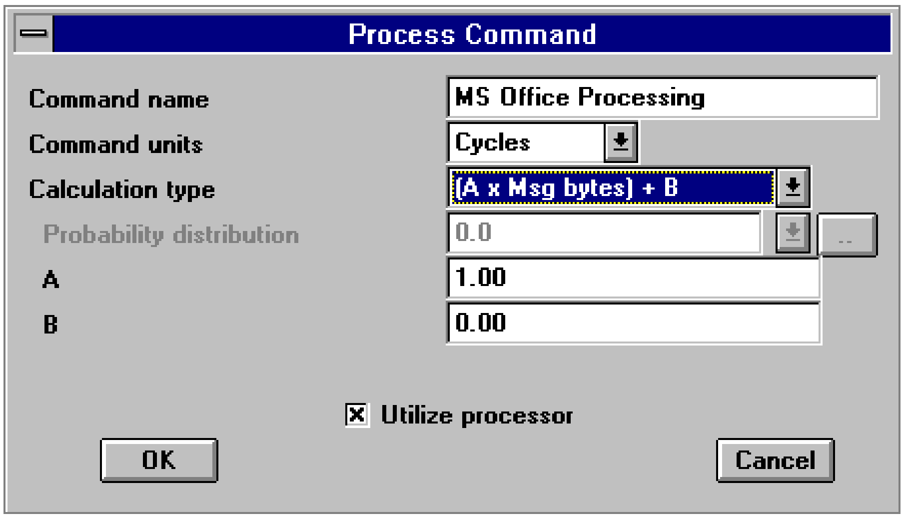


Figure 5.15 MS Office Processing Command

f) When all values have been entered press the *OK* button as shown in Figure 5.15 and then define a global transport command.

g) Edit the transport command parameters to those of the following list and shown in Figure 5.16.

- Name: File Request
- Message Size Calculation: Probability Distribution
- Probability distribution: Set to a constant value of 100 bytes
- Message test option: Copy message name
- Priority: 1
- Routing Class: Standard

- Transport protocol: TCP/IP-Microsoft
- Packetize: 1.0 milliseconds
- Destination Type: Set to a weighted list and create a list containing only the File Server node.

h) When all values have been entered, press the *Ok* button as shown in Figure 5.16 and then define a global filter command.

i) Edit the parameter of the filter command to the following values:

- Filter Command Name: File Request Wait
- Required message text: Set to File Request Answer

j) When all values have been entered, press the *OK* button in the filter command window, and then define a global processing command.

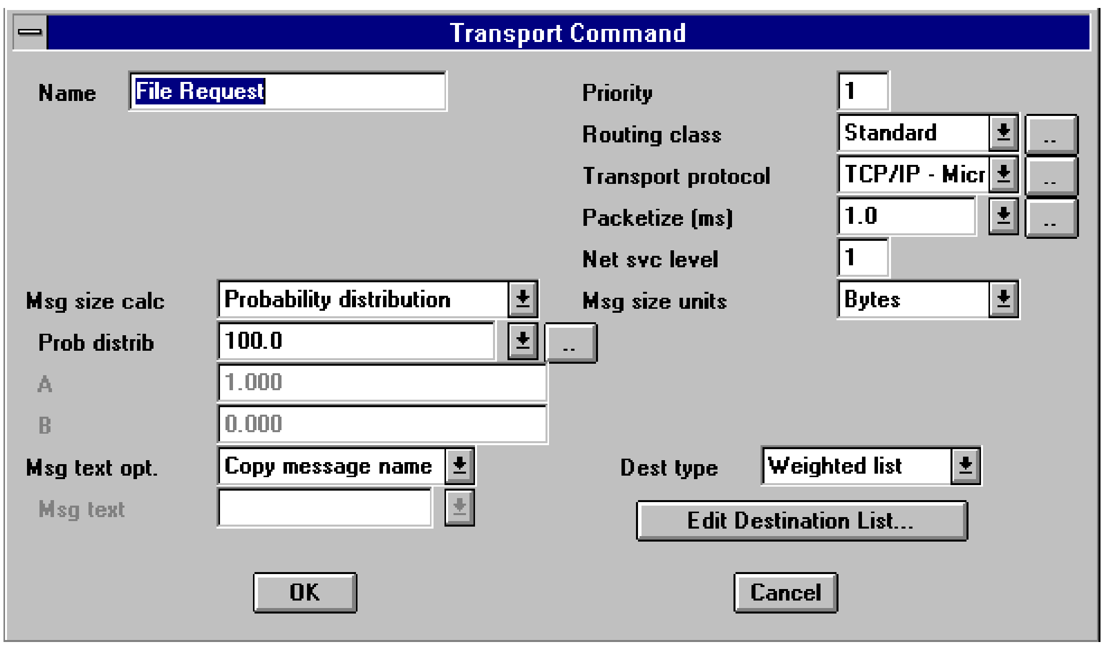


Figure 5.16 File Request Transport Command

k) Edit the parameters of the processing of the processing command to the following values and as shown in Figure 5.17.

- Processing command Name: Application Processing
- Number of cycles calculated: Probability Distribution

- Probability Distribution: Set to a uniform distribution with a minimum value of 81,000,000,000 cycles and a maximum value of 108,000,000,000 cycles.

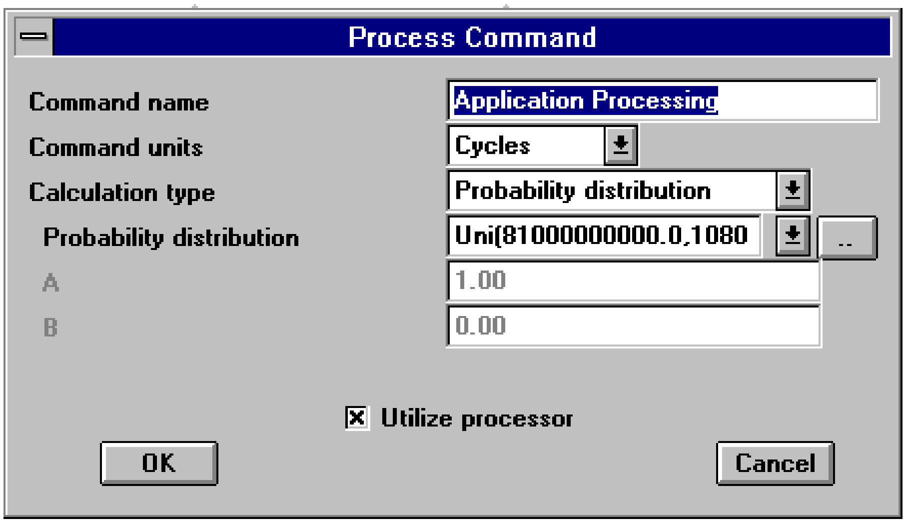


Figure 5.17 Application Processing Command

l) When all values have been entered, press the *OK* button as shown in Figure 5.17, and then define a global transport command.

m) Edit the transport command parameters to those of the follwoing list and shown in Figure 5.18.

- Name: Save Request 1
- Message Size Calculation: (A x Msg Bytes) + B
- A: 1.100
- B: 0.000
- Message text option: Copy message name
- Priority: 1
- Routing Class: Standard
- Transport Protocol: TCP/IP-Microsoft
- Packetize: 1.0 millisecond
- Destination Type: Set to a weighted list and create a list containing only the File Server node.

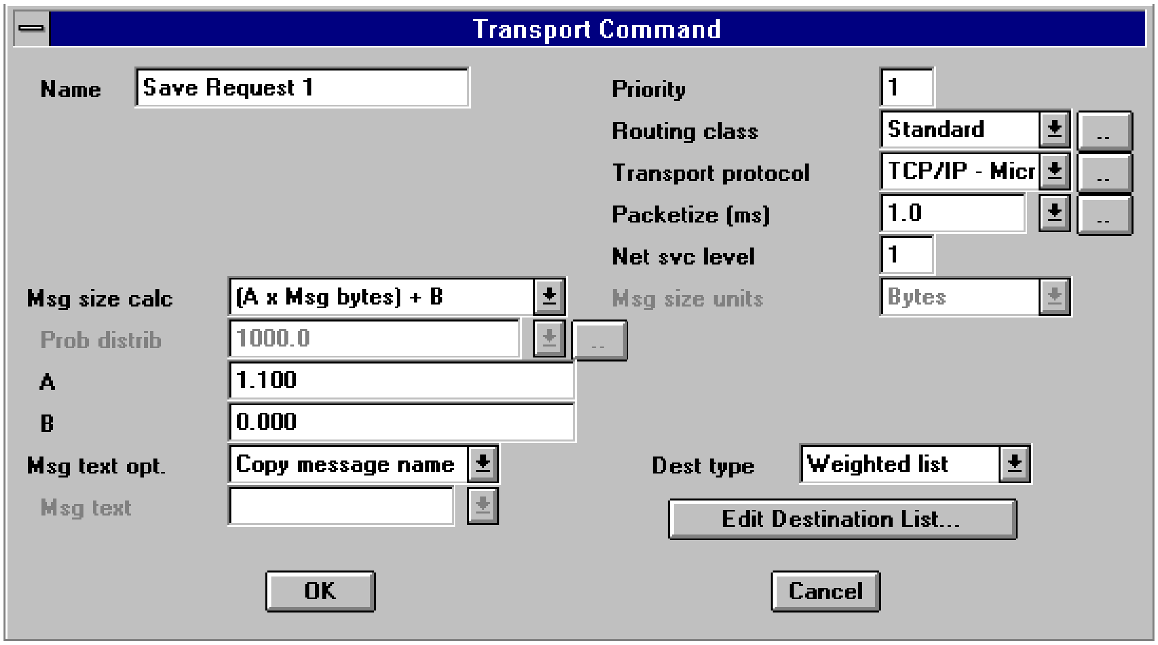


Figure 5.18 Save Request 1 Transport Command

n) When all values have been entered, press the *OK* button as shown in Figure 5.18, and then define another global transport command.

o) Edit the transport values to those of the previous transport command defined with the following exceptions:

- Name: Save Request 2
- A: 1.200

p) When all values have been entered, press the *OK* button and then define another global transport command.

q) Edit the transport command parameters to those shown in the following list:

- Name: STL PC Print Request
- Message Size Calculation: (A x Msg Bytes) + B
- A: 1.200
- B: 0.000
- Message text option: Copy message name
- Priority: 1
- Routing Class: Standard

- Transport protocol: NCP
- Packetize: 1.0 milliseconds
- Destination Type: Set to a weighted list and create a list containing only the STL Laser Printer node.

r) When all values have been entered, press the *OK* button in the transport command window, and then press the *Done* button in the command repertoire window.

s) Create an application source and place it next to the STL PC's node.

t) Using either connection tool connect the application source to the STL PC's node.

u) Edit the application source fields to the following list and as shown in Figure 5.19:

- Name: Office Application
- Schedule by: Delay Time
- Interarrival: Set to a uniform distribution with a minimum value of 300 seconds and a maximum value of 900 seconds. The stream value may be set to any integer value from 0 to 99.
- First arrival: Set to a uniform distribution with a minimum value of 0 seconds and a maximum value of 600 seconds. The stream value may be set to any integer value from 0 to 99.
- Last Arrival: none
- Maximum Arrivals: none

v) In the command sequence box shown in Figure 5.19, use the *Add* button to script the commands in the following order where each command executes only one time.

1. MS Office Request
2. MS Office Wait
3. MS Office Processing
4. File Request
5. File Request Wait
6. Application Processing
7. Save Request 1
8. Application Processing
9. STL PC Print Request

10. Save Request 2

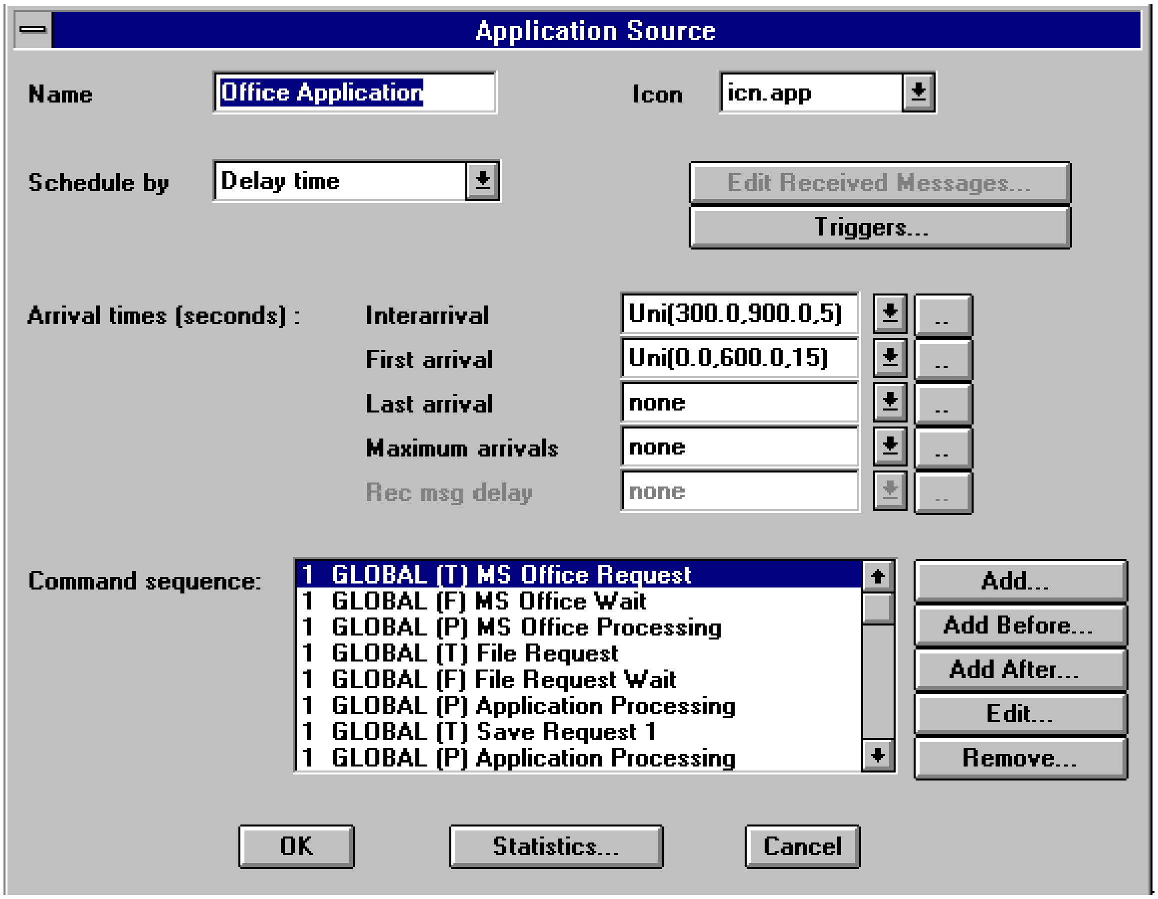


Figure 5.19 Office Application Source

w) When all values have been entered and the command sequence scripted, press the *OK* button as shown in Figure 5.19.

5. Creating a Tabular Distribution

The tabular distribution created will be used to model the selection of the file size to read on the application server:

a) Select the *Table* option from the *Define* menu, and in the tabular distribution window which appears press the *Add* button.

b) Edit the fields of the table detail window to the following values. Figure 5.20 depicts the table upon completion.

- Name: Office
- Table Type: Discrete
- Prob and Values: Set to the cumulative probabilities and file size values of Table 5.1.

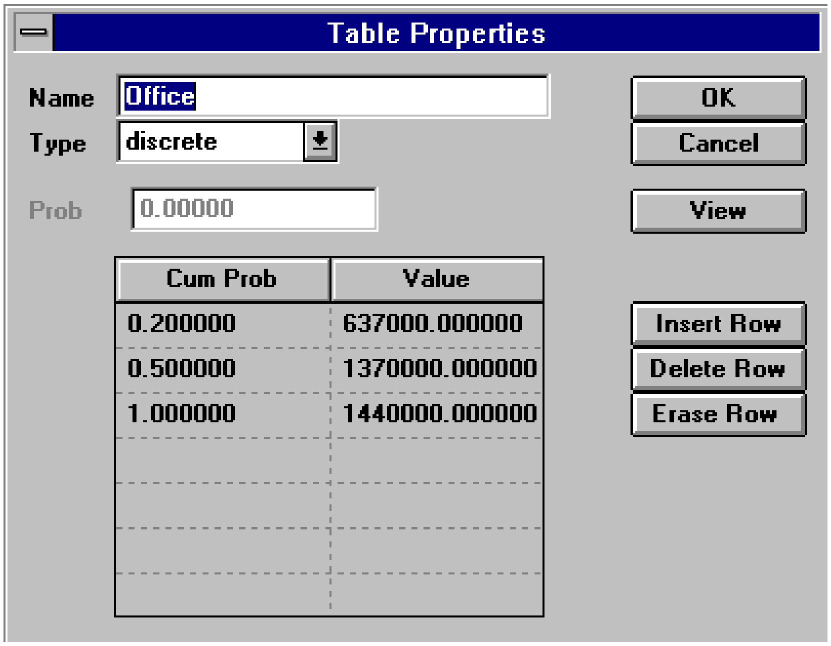


Figure 5.20 Office User Defined Distributions

c) When all values have been entered press the *OK* button as shown in Figure 5.20 and then press the *Done* button at the bottom of the tabular distribution window.

6. Creating the Application Server's Application Source

The application source which will run on the application server will be built by first defining the commands which will be used in this application, and then creating the application.

- Name: MS Office Read
- File to Access: GENERAL STORAGE
- File modification method: Do not modify file
- Bytes read calculation: Probability distribution
- Probability Distribution: Office

c) When all fields have been edited, press the *OK* button at the bottom of the read command window; press the *Done* button at the bottom of the command repertoire window; and press the *OK* button at the bottom of the node detail window.

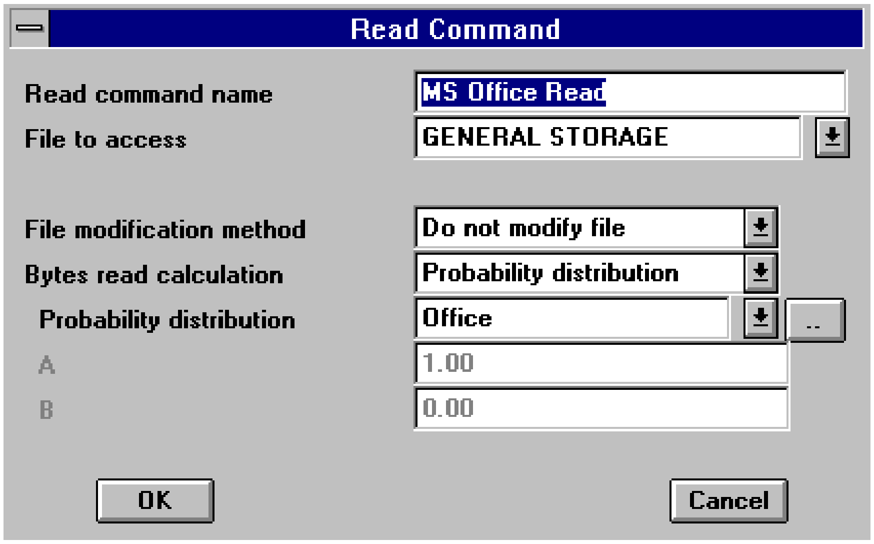


Figure 5.21 MS Office Read Command

d) Define a global answer command and edit the field of the answer command window tot the following list. Figure 5.22 depicts the command upon completion.

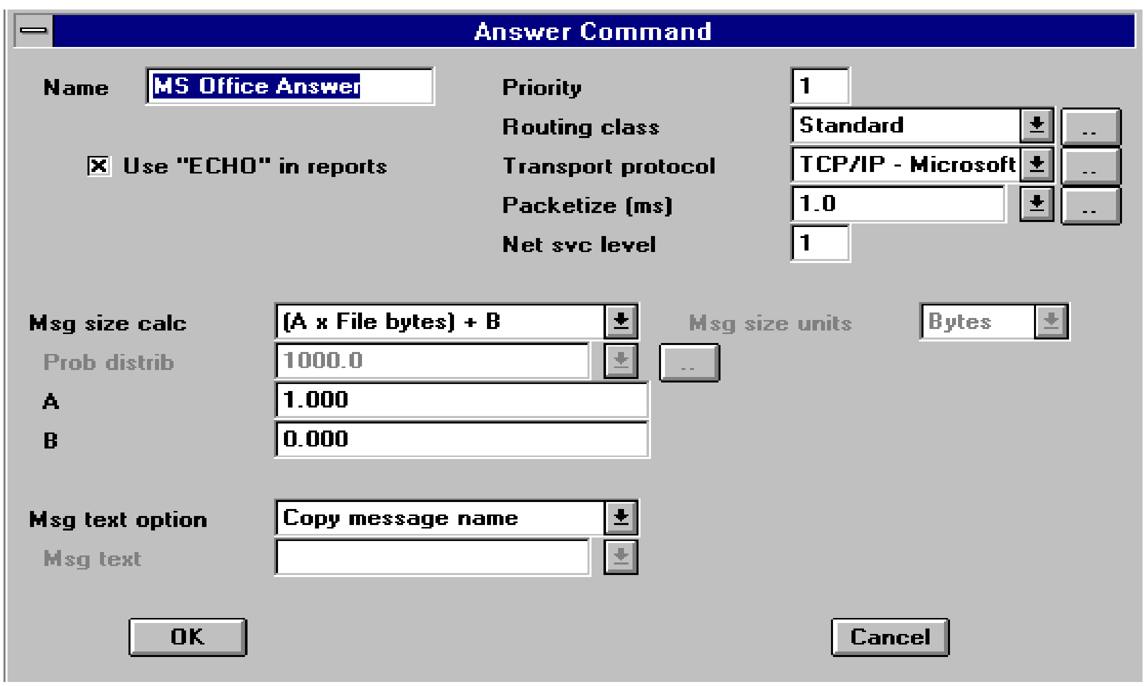


Figure 5.22 MS Office Answer Command

- Name: MS Office Answer
- Priority: 1
- Routing Class: Standard
- Transport Protocol: TCP/IP-Microsoft
- Packetize: 1.0 milliseconds
- Message Size Calculation: (A x File Bytes) + B
- A: 1.000
- B: 0.000
- Message Text Option: Copy message name

e) When all fields are edited, press the *OK* button at the bottom of the answer command window, and the *Done* button at the bottom of the command repertoire window.

f) Create an application source and place it next to the Application Source node.

g) Connect the application source to the Application Server node using either connection tool.

h) Edit the application source fields to the following parameters. Figure 5.23 depicts the application source upon completion.

    - Name: MS Office Suite
    - Schedule by: Set to the received message option, and create a received message list which requires one message with the text "MS Office Request" to trigger the application source

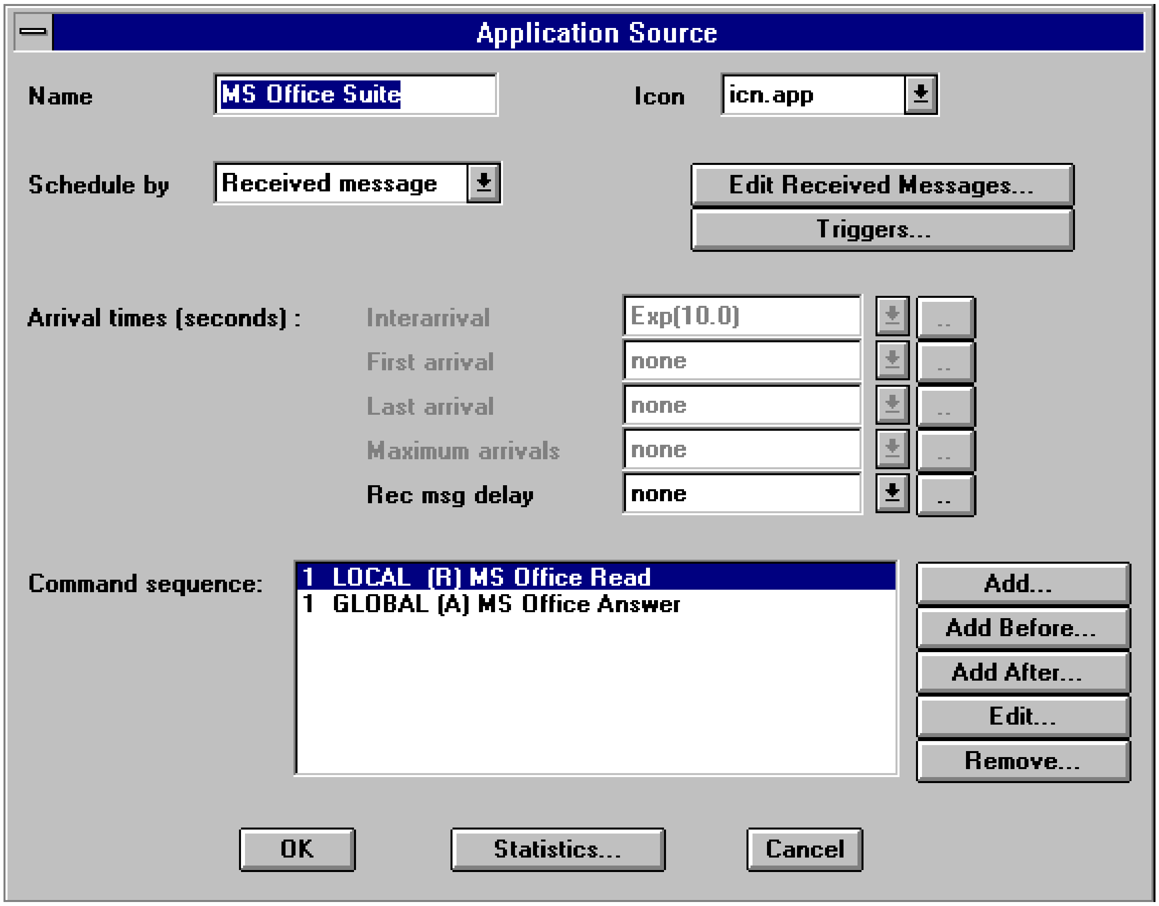


Figure 5.23 MS Office Suite Application Source

i) In the command sequence box shown in Figure 5.23 add the following commands set to execute a single time:

1. MS Office Read
2. MS Office Answer

j) When all values and the command sequence have been entered, press the *OK* button at the bottom of the application source window.

7. Crating the File Server's File Request Application Source

The file request application source which will run on the file server will be built by first defining the commands which will be used in this application, and then creating the application source.

a) Create a global process command and edit the fields of the process command window to the following values:
   - Processing command name: File Request Processing
   - Number of cycles calculated: (A x Msg Bytes) + B
   - A: 1.500
   - B: 0.000
b) When all values are entered, press the *OK* button at the bottom of the process command window.
c) Define a global read command and edit the parameters of the read command window to the following:
   - Name: File Request Read
   - File to Access: GENERAL STORAGE by default
   - File modification method: Do not modify file
   - Bytes read calculation: Probability distribution
   - Probability distribution: Set to a normal distribution with a mean of 150,000 bytes and a standard deviation of 50,000 bytes. The stream value may be set to any integer value from 0 to 99.
d) When all values have been entered, press the *OK* button at the bottom of the read command window.
e) Define a global answer command and edit the fields of the answer command window to the following:
   - Name: File Request Answer
   - Priority: 1
   - Routing Class: Standard
   - Transport protocol: TCP/IP-Microsoft
   - Packetize: 1.0 milliseconds
   - Message Size Calculation: (A x File Bytes) + B
   - A: 1.000
   - B: 0.000

- Message text option: Copy message name

f) When all values have been entered, press the *OK* button at the bottom of the answer command window, and press the *Done* button at the bottom of the command repertoire window.

g) Create an application source and place it by the File Server node.

h) Connect the application source to the File server node using either connection tool.

i) Edit the application source parameters to the values shown in the following list. Figure 5.24 depicts the application source upon completion.

  - Name: File Request
  - Schedule by: Set to the received message option, and create a received message list for which requires one message with the text “File Request” to trigger the application source.

j) Edit the command sequence

  1. File Request Processing
  2. File Request Read
  3. File Request Answer

8. Creating the Save File Application Source

The save file application source which will run on the file server will be built by first defining the commands which will be used in this application, and then creating the application source:

a) Define a global write file command and edit the command to the values shown in the following list. Figure 5.25 depicts the write command upon completion

  - Write command name: Save Write
  - File modification method: Update data in file
  - Bytes written calculation: (A x Msg Bytes) + B
  - A: 1.000
  - B: 0.000

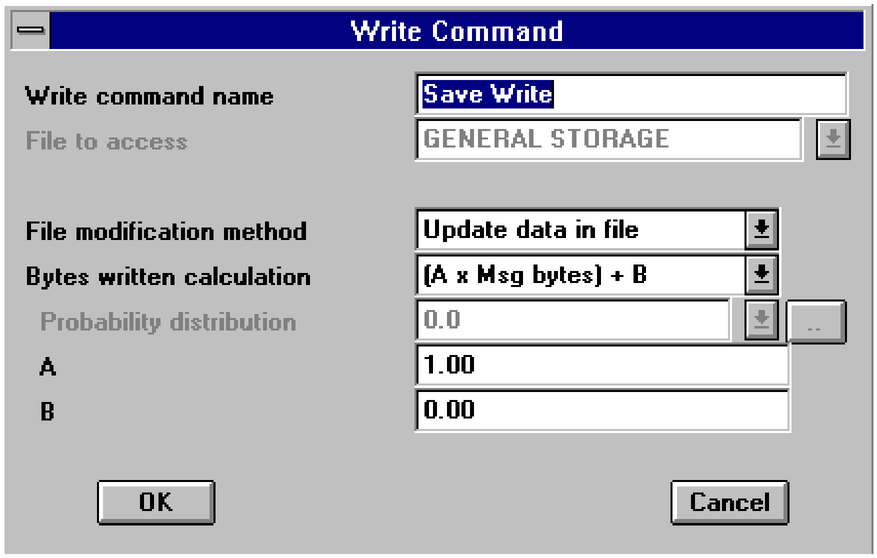


Figure 5.25 Save Write Command

b) When all fields are set, press the *OK* button at the bottom of the write command window, and then press the *Done* button at the bottom of the command repertoire window.

c) Create an application source and place it next to the File Server node.

d) Connect the application source to the File Server node using either connection tool.

e) Edit the application source fields to the values shown in the following list. Figure 5.26 depicts the application source upon completion.

- Name: Save File
- Schedule by: Set to the received message option, and create a received message list for which requires one message with the text "Save Request 1" or "Save Request 2" to trigger the application source.
- Command sequence: Add only the Save Write command with a single execution.

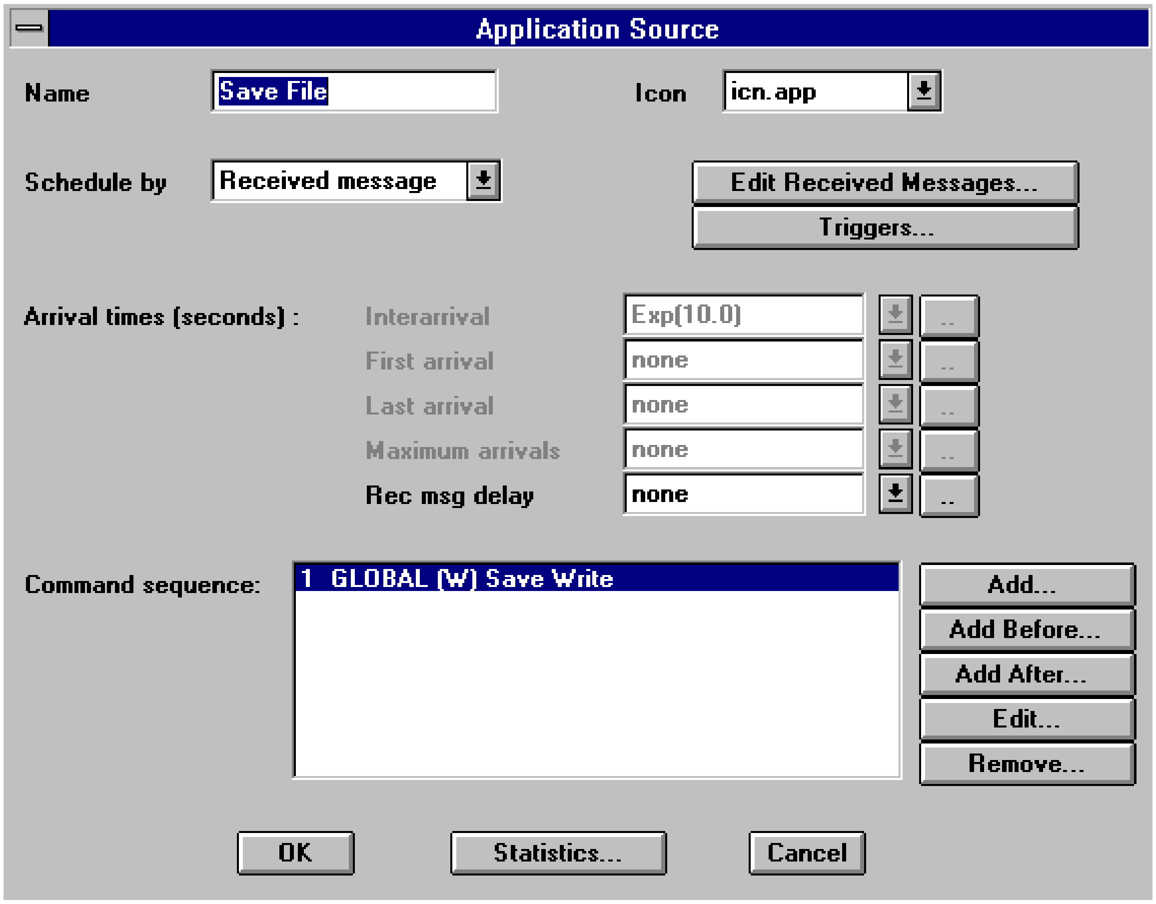


Figure 5.26 Save File Application Source

f) When all fields are set, press the *OK* button at the bottom of the application source window.

9. Verifying and Saving the Model

The final step in building the model is verifying the logical connections. This is accomplished by selecting the *Verify Model* option form the *Simulate* menu. Any errors in the model must be corrected. Finally, save the model.

G. SIMULATION AND CONCLUSIONS

Three replications within a simulation will be run to analyze the problem. Each replication will be 3600 seconds in length to model one hour of network loading. The first replication was set to a warmup length of 30 seconds to allow the application in the model to begin generating traffic before statistics were started to be gathered.

Prior to running the simulation, the following reports were set to gather the necessary information on network loading, application run length, and disk information on the file server.

- Link Reports: Channel Utilization and Collision Stats for the 63 Net.
- Application Run Length: Selected for the Application and File Server nodes.

After running the simulation, the results of each replication are stored in the files report.1, report.2, and report.3 where the number indicates the replication number. The results showed that the average network utilization varied from 0.49 percent in the first replication, dropped to 0.35 percent in the second replication, and increased again to 0.51 percent in the final replication. After looking at the collision statistics and transmission delays statistics gathered during the simulation, it was decided that the results of the first simulation are biased by the manner of modeling the behavior of the users. The average transmission delay did not vary greatly between replication (0.977, 0.800, and 0.786 milliseconds), but the standard deviations did vary greatly (12.620, 1.370, and 0,800 milliseconds). Thus, in the first simulation, the start of several applications at the same time caused multiple collision episodes and increased the delay on the network. Better results might be obtained by increasing the number of replications as the randomness built into the modeling of the user behavior would better model the use of the lab.

The file and application serves application run lengths showed similar behavior of longer delays in the first replication, followed by a decrease in application delays in the second replication, but then followed by increased delays in the third replication. Due to the effects of greater randomness in the third replication, the results of this replication are deemed to be the best estimates. The application server took an average of 30 seconds with a standard deviation of 7 seconds to read and transmit the program onto the network. The file server took on the average 4 seconds with a standard deviation of 1 second to either save or retrieve a file.

Overall, better results could be obtained by increasing the number of replications which would allow the randomness of the users' behavior to spread out the request over time.

VI. LINKS AND COMNET Baseliner

A. INTRODUCTION

The goal of this chapter is to introduce the various link objects which are available in COMNET III to model the physical medium over which messages may be transported. In addition, the COMNET Baseliner utility within COMNET III will be covered. This utility allows importing traffic files captured from an actual network into a model.

B. LINK OBJECTS

Link objects are used to model both the physical media over which packets are transmitted and the framing characteristics which model the data link layer of the OSI Reference Model. COMNET III currently has the capability to model point-to-point, polling, Aloha, carrier sense multiple access (CSMA), carrier sense multiple access with collision detection (CSMA/CD), token ring, FDDI with Target Token Rotation Time, Priority FDDI with Target Token Rotation Time, Priority Token Ring with parameters for 100VG-AnyLAN, IEEE 802.11 CSMA/CA Wireless LAN, and Demand-Assigned Multiple Access links for TDMA, FDMA, CDMA.

The general logic that COMNET III uses to model message traffic on a link is performed in the following manner for all types of links. When a message packet on a node is ready to be transmitted, the node eventually gains access to the link. The packets are then broken down into frames according to the framing characteristics set for the link. The model then calculates the transmission delay for the frame across the link based on the size of the frame and the transmission rate of the link. If a frame error probability is set for the link, a random pull from the distribution used to describe this error is then made to determine if the frame will arrive in error and require retransmission. The link is then made busy for the transmission delay calculated and a propagation delay which may be set for the link. After the frame has incurred these delays, it is considered to have arrived at either its destination, if the destination is on the same link, or the next node along the frame's routed path.

Links may be created by choosing any of the three link types available on the toolbar or by selecting the *Link* option from the *Create* menu. Editing of a link of any type will cause the link detail window shown in Figure 6.1 to appear. The window contains fields which allow setting a unique name for the link and specifying the type of link which is to be modeled. Any link created may be switched to a different type of link by changing the type field. The parameter field allows selecting any of the predefined links available in COMNET III or creating a user defined parameter set by pressing the ".." button to the right of the parameter field. The control node field may be used to specify a controlling node in the model for contention links such as CSMA, CSMA/CD, or aloha link types, and is required to be set for a polling link. As with all objects in COMNET III, information in the form of a constant or statistical distribution may be entered concerning the time to failure and time to repair for the link. The current state of the link, and a simulation toggle time to change this state, may also be entered.

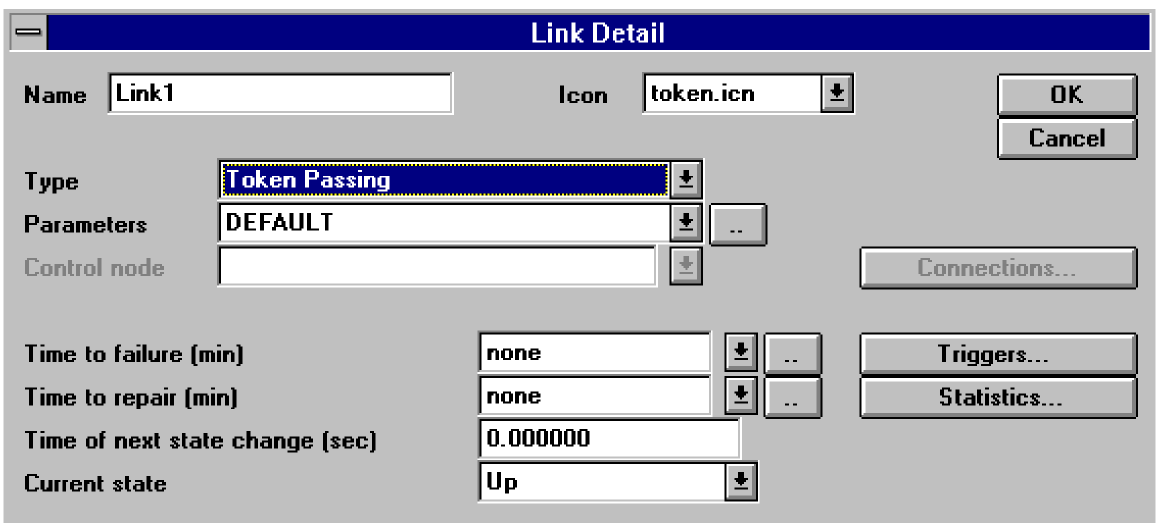


Figure 6.1 Link Detail Window

1. Common Link Parameters

As with any object in COMNET III, parameter sets may be defined for each type of link. The common fields that apply when defining any link are the ability to assign a unique name to the link parameter set, specifying the link's bandwidth, setting of the link framing characteristics, setting a propagation delay, defining a frame error probability, and setting a link session limit.

Framing characteristics are used to model the data link protocols in a network. The modeling is not detailed in that the common functions which apply to the data link layer such as time outs, link level acknowledgments; and cyclic redundancy checks (CRC) are not modeled [Ref. 2 p. 80]. The framing characteristics are used solely to model the frame payload capability and overhead which are used in link delay and utilization calculations. The fields that define the framing characteristics are as follows:

- Frame Minimum: Sets the smallest data payload that a frame may carry. Frames which are smaller than this value are padded to the minimum size. This value includes the frame overhead bits.
- Frame Maximum: Sets the upper limit of the data payload of a frame. The exception is setting this value to zero which implies there is no upper limit. This value includes the frame overhead bits.
- Frame Overhead: Used to specify the overhead bytes associated with the link level protocol.
- Frame Error Probability: Used to set the probability of a frame arriving in error on a link which will require retransmission of the frame. This value may be described by a constant or any statistical distribution.

The propagation delay field which is common to all links is used to model the time required for a packet to move between two nodes on a link based on distance. The COMNET III User's Manual states that this feature should be most commonly used to model delays associated with transmissions over long distances such as when modeling geographically disbursed nodes in a wide area network or satellite links [Ref. 2 p. 98].

The bandwidth field associated with each type of link is used to specify the transmission rate of the link in kilobits per second. The session limit is used to specify the maximum number of connection-oriented sessions which the link may support at a given time. When the session limit for the link is reached, additional setup packets generated by session sources or application sources will be blocked.

In addition to the link parameters common to every type of link, the contention type links of CSMA, CSMA/CD, and Aloha all allow specifying a retry interval when a collision occurs. The retry interval may be set to any of the statistical distributions available in COMNET III or to a binary exponential backoff distribution. The binary exponential backoff algorithm is specified by the IEEE for determining the retry times for nodes connected to a collision link [Ref. 2 p. 69]. The algorithm doubles the retry interval of a frame every time a collision occurs, up to a maximum of 10 times. After the tenth collision the retry time remains constant, and frame transmissions are attempted up to the retry limit. If the retry limit is exceeded the retry interval used is set by the limit delay.

2. Aloha Link Characteristics and Parameters

The Aloha link is used for the modeling of random access radio links where random transmission may occur, collisions are detected, and retransmission of collided frames occurs. The link allows selecting one of the nodes as a control node to which all communications on the link may be directed. The control node is allocated a contention-free channel for transmitting packets to all other nodes in the link. All other nodes on the link compete over a single channel for transmission.

The logic applied when running a simulation for frames on an aloha link is as previously described for the general case for all links with the exception that collisions may occur on this type of link. Collisions occur when two nodes transmit at the same time on the link. As there is no control on the link to prevent this from occurring, upon detection of a collision the node waits to attempt retransmission based on the retry interval specified for the link.

The transmission of packets on an Aloha link may be modeled using either unslotted or slotted operation. Unslotted operation implies that any node may transmit packet frames the instant the frames are ready to be transmitted. Slotted operation is similar to a time division multiple access (TDMA) scheme where the link is divided into time slots of equal duration. The difference between Aloha and TDMA is that instead of each node having a dedicated time slot to transmit in, as in TDMA, there is contention for each time slot by all nodes on the link. Packets may only be transmitted at the beginning of the time slot. Thus, if only one node transmits, a collision will not occur. If multiple nodes transmit in the same time slot, a collision will occur and retransmission is rescheduled based on the retry interval.

Editing the parameter set of an Aloha link will cause the Aloha parameter window shown in Figure 6.2 to appear. In addition to the common characteristics of all links, the following fields may be specified to describe the operation of the link:

- Slot width: used to set the duration of each slot in units of milliseconds. A slot width of zero is used to model unspotted operation.
- Control channel: Checking this box allows a control node to be specified in the link detail window for the link as shown in Figure 6.1.
- Transmission Return Rate: Used to set the transmission rate in kilobits per second of the control channel.

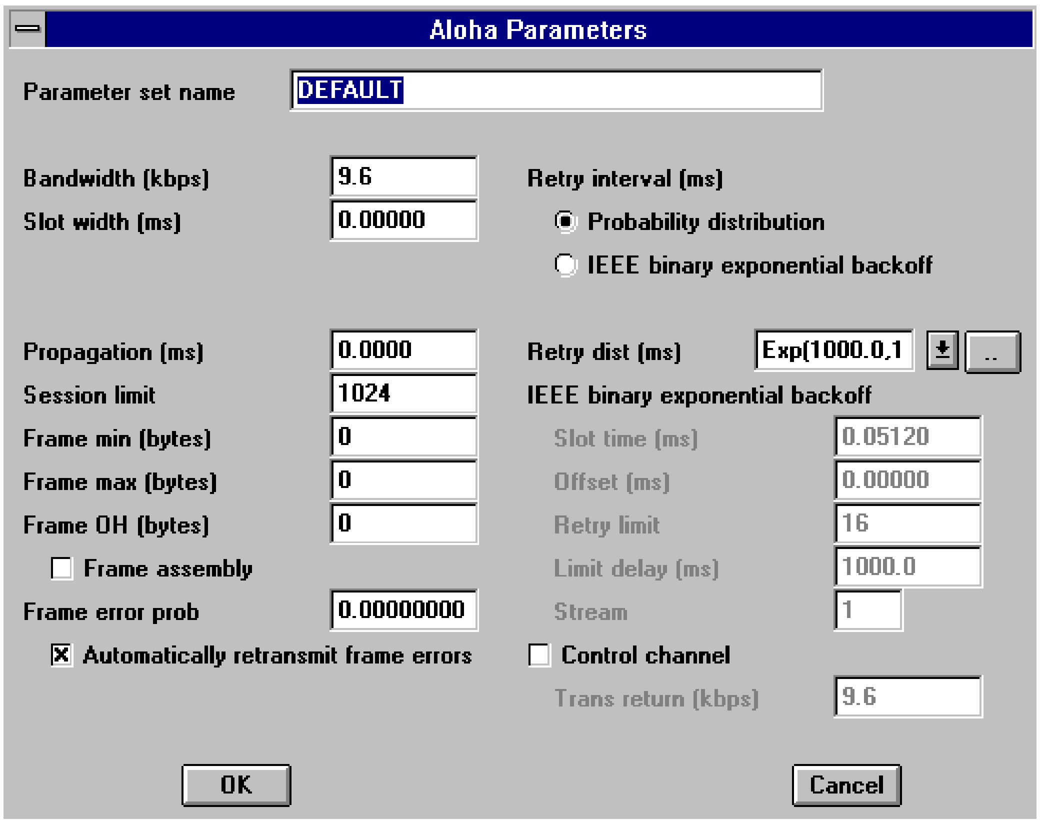


Figure 6.2 Aloha Parameters Window

3. Carrier Sense Multiple Access (CSMA) and Carrier Sense Multiple Access With Collision Detection (CSMA/CD) Link Characteristics and Parameters

CSMA and CSMA/CD links are multi-access links which may be used to model random access links. Nodes on both types of links first listen to the link to see if it is busy prior to transmitting packets. The difference in the modeling of the two link types is that CSMA is modeled to detect a collision at the end of a transmission whereas CSMA/CD is modeled to detect the collision upon its occurrence. The modeling of the CSMA is not true to the actual implementation of this data link protocol as retransmissions are rescheduled based on time outs and not true collision detection. However, the method chosen to model a CSMA link is a good approximation.

The logic applied in the modeling of these links in addition to transmission and propagation delays as described by the COMNET III User's Manual [Ref. 2 p. 88-89] is done in the following manner. A node with packets ready to be transmitted first checks to see if the link is busy. If busy, transmission of the packet is deferred until the link becomes idle. When the link becomes idle, the packet is transmitted and the link is placed in an unsettled state for the time period of the collision window. The link is then placed in a busy state for the time period of the transmission delay after which the link becomes idle again. If multiple nodes detect the link idle and transmit at the same time, or if a node transmits during the collision window, a collision will occur on the link.

Parameter sets for the CSMA/CD type links are included in the COMNET III library for the IEEE 802.3 CSMA/CD 10Base2, 10Base5, 10BaseT, and 10Base5 Star standards; and for the IEEE 802.3 Ethernet 10Base5 standard. No parameter sets are defined in the library for CSMA type links. The editing of the parameters of either type of link will bring up the window shown in Figure 6.3. The additional fields which may be specified for these links are as follows:

- Collision window: The time period a link is vulnerable to collision after a packet is transmitted from a node defined in units of milliseconds.
- Jam interval: On CSMA/CD type links, the jam interval models the sending of a jamming signal by the link upon the detection of a collision. For CSMA

type links, the jam interval models the interval from the end of a transmission until the sending node learns that retransmission is required.

- Interframe Gap: The amount of time in milliseconds a node delays the transmission of a packet once the link is detected as being idle.
- Control channel: Checking this box allows selecting a control node in the link detail window.
- Transmission return rate: Specifies the transmission rate of the control channel in kilobits per second.

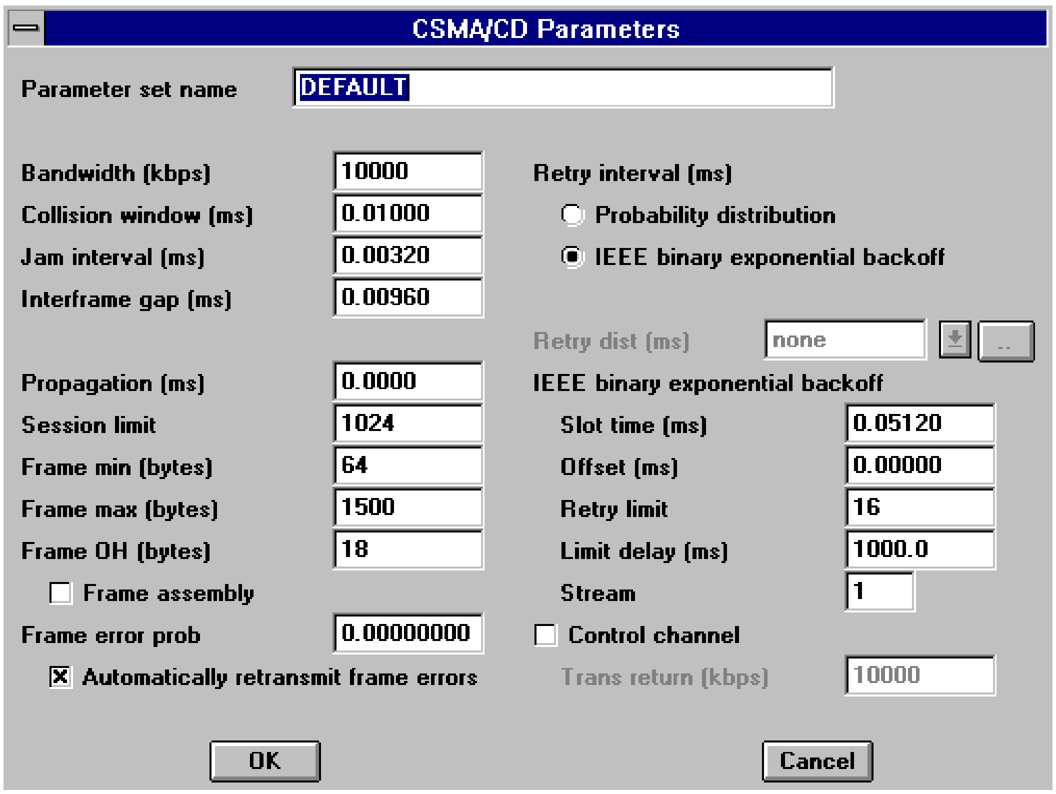


Figure 6.3 CSMA or CSMA/CD Parameters Window

4. Point-To-Point Link Characteristics and Parameters

Point-to-point links may be used to model the links of a wide area network, satellite links, phone lines for modem transmissions from a terminal device, or any situation where a dedicated link occurs between two terminals. The point-to-point link is modeled to have full duplex capabilities. Frames are multiplexed onto the link in the order in which they appear in the link buffer which is defined on the port. The logic which is applied to this link type when running a simulation is the general case which applies to all links.

Editing the parameters of this link will cause the point-to-point parameter window shown in Figure 6.4 to appear. The additional parameters which may be set in modeling the operation of this link for data transmission include setting the number of circuits the link is to model, and setting the bandwidth per circuit from nodes X to Y and from Y to X. Thus, one point-to-point link may be used to model several actual links to simplify the modeling.

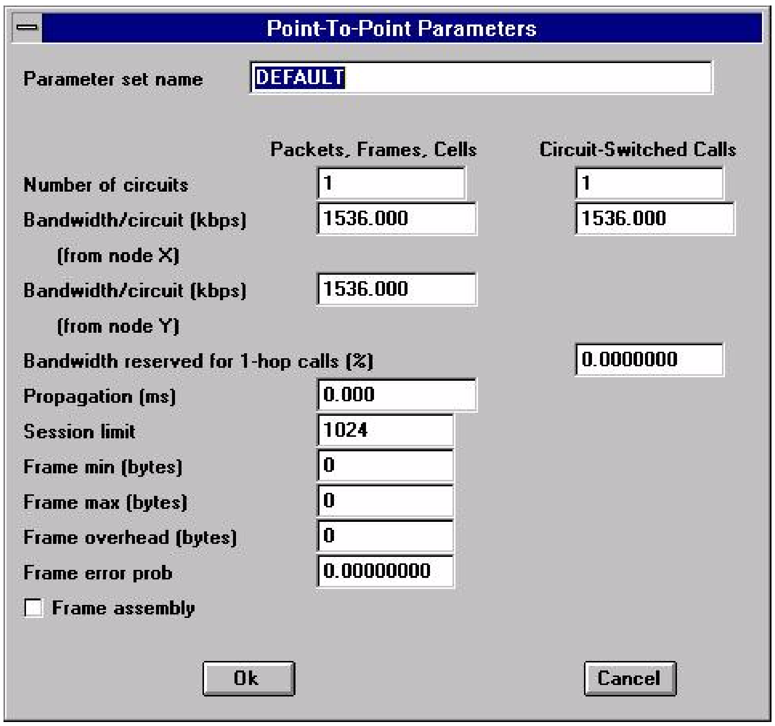


Figure 6.4 Point-To-Point Parameters Window

5. Polling Link Characteristics and Parameters

Polling links were created in COMNET III specifically to model multi-drop telephone lines which have a centralized control node which polls other devices on the line [Ref. 2 p. 99]. The link operates in a round-robin order such that the controlling node queries or polls each node on the link using a control channel. When polled, a node will transmit all packets which are currently ready to be transmitted on the link's common channel. If a node has no packets, the node sends a negative acknowledgment to the control node which will then query another node. The transmission of the frames across the link is done using the general logic applied to all links when running a simulation.

Editing the parameters of this link will cause the polling link parameters window shown in Figure 6.5 to appear. In addition to the common parameters of all link types, the polling link's operation is specified by the following:

- Control channel: This box must be checked for a polling link and allows selecting the node connected to the link which will act as the controlling node in the link detail window.
- Transmission return rate: Specifies the transmission rate in kilobits per second of the control channel.
- Poll needed to send: Checking this box specifies that the controlling node must also be included in the polling scheme to transmit packets.
- Poll Delay: Used to model the time in milliseconds required to send a poll from the control node to any other node on the link.
- NAK delay: Used to model the time in milliseconds required for a node to send a negative acknowledgment back to the control node.

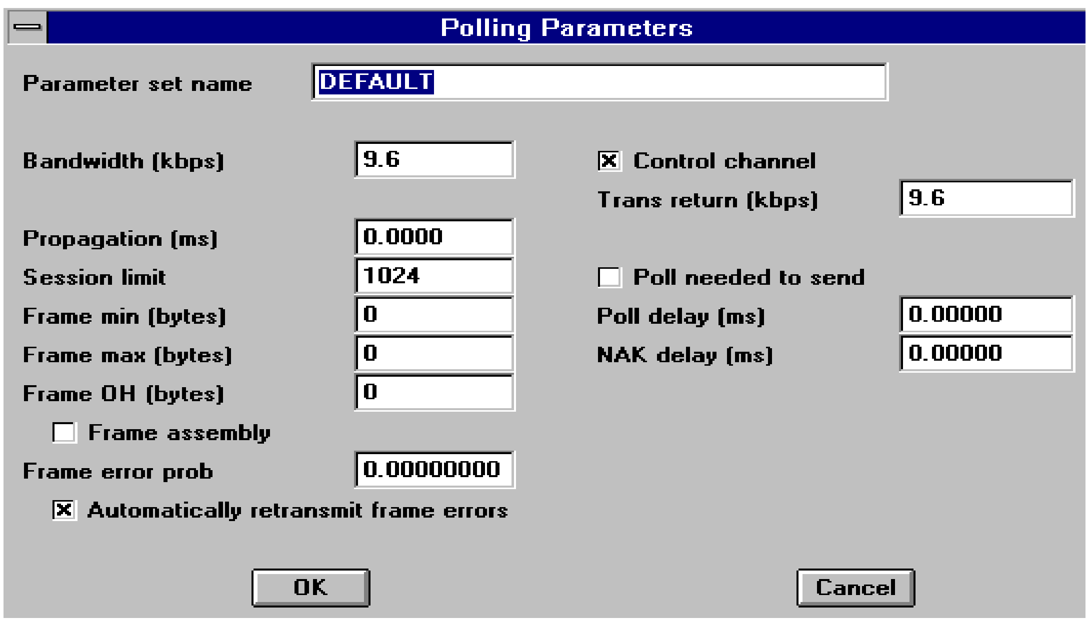


Figure 6.5 Polling Link Parameters Window

6. Token Passing Link Characteristics and Parameters

Token passing links are used to model the characteristics of the IEEE 802.4 and 802.5 specifications and the ANSI Fiber Distributed Data Interface (FDDI) standard. As such, parameters sets are included in the COMNET III library to model links with characteristics of the 1Mbps, 5 Mbps, and MAP 10 Mbps IEEE 802.4 standard; 4 Mbps and 16 Mbps IEEE 802.5 standard; and a parameter set to model FDDI operation.

The modeling of the operation of a token passing link is performed in the following manner as defined in the COMNET III User's Manual [Ref. 2 p. 106, 107]. The token passing link sends in round-robin order a token to each node on the link. When the token is received by a node, the node may transmit packets for a time specified by the token holding limit. A frame continues to be sent if it is more than halfway transmitted when the holding limit time is up. If upon receipt of the token a node has no packets to send, the token is passed on to the next node immediately. The logic applied upon transmission of a packet is the same as the general case for all links. The token itself is not modeled within COMNET III. It is instead modeled by a time calculation. When a node has a packet to transmit, it determines which node currently has the token and the amount of time which will pass until it receives the token again. The node then delays for the amount of time calculated before sending the packet.

Additional parameter sets for token passing links may be defined. Editing of these parameters will cause the token passing parameters window shown in Figure 6.6 to

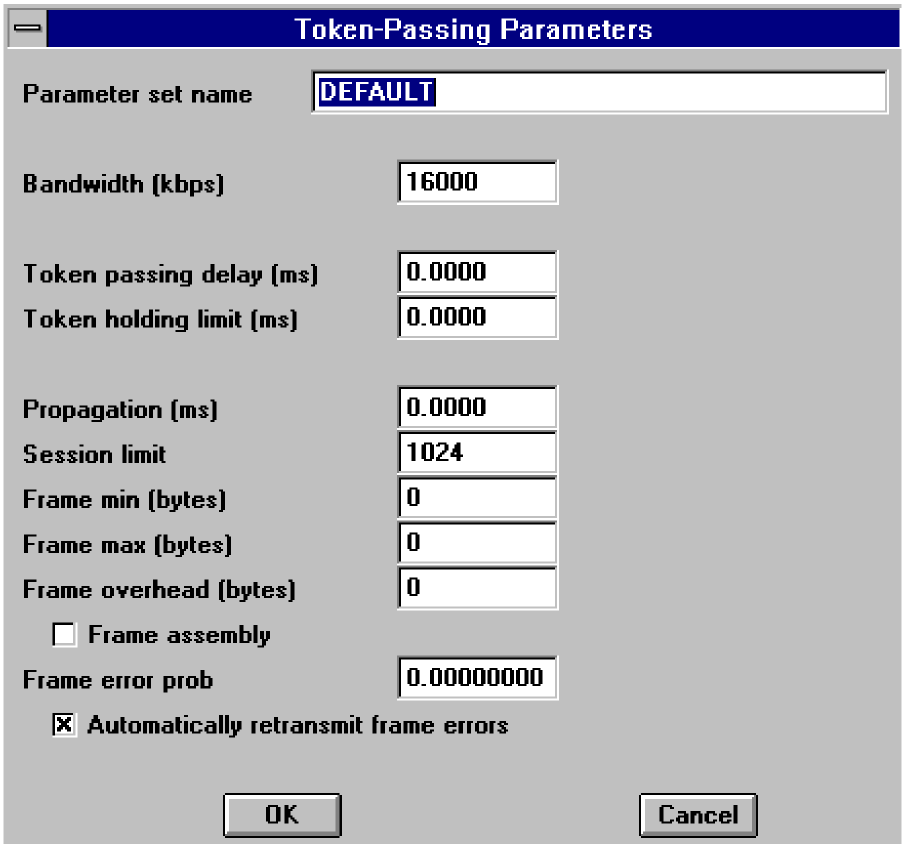


Figure 6.6 Token Passing Link Parameters Window

appear. The following parameters in addition to the common link parameters may be set to define the operation of the link.

- Token passing delay: The amount of time in milliseconds required to pass the token from one node to the next.
- Token holding limit: The amount of time in milliseconds a node may hold the token and transmit packets.

7. FDDI & Priority FDDI with Target Token Rotation Time

COMNET III includes two models of FDDI--a Basic FDDI model and a Priority FDDI model. As with the token-passing link, the COMNET III FDDI models do not actually schedule each token-passing event; instead, the model keeps track of the last node to have relinquished the token and the time when the node relinquished the token. Given the last token location and the time that the frame left that location, COMNET III computes the appropriate amount of time (token passing delay) required for the token to advance the next time some node has a frame to transmit. COMNET III monitors the actual token rotation time and the token-ring report includes a count of the number of times that the actual token rotation time exceeds twice the target token rotation time to help identify heavily loaded conditions that may trigger FDDI re-initialization.

The Basic FDDI model has a Token-Passing Delay that determines the time required to pass the token from one node on the FDDI's connection list to the next node on the connection list. The time that a node can hold the token is controlled through the Target Token Rotation Time (TTRT). When a node receives the token, before transmitting a frame it determines the time that has elapsed since the node last received the token. If the elapsed time exceeds the TTRT, then the node passes the token to the next node without transmitting the waiting frame. The node can continue to transmit frames as long as the elapsed time since last receiving the token does not exceed the TTRT.

The Priority FDDI model allows the TTRT to vary by frame priority which is the highest priority of the packets it carries. The TTRTs can also vary by port, so that some nodes on the Priority FDDI link can hold the token longer than other nodes. The priority FDDI parameter set maintains a list of tables for different token rotation time targets (T_Pr) that vary by priority and the different tables are available for assignment to different ports (node interfaces). By default, there is one table called "DEFAULT" and it will be assigned to each new port added to the link.

The Priority FDDI parameter set dialog box has a button "T_Pr tables…" that manages the sets of tables that may be assigned to the individual ports. Use the "T_Pr tables…" button to edit the list of port properties available to be assigned to each port connecting to the priority FDDI link. Each port properties dialog box consists of a table of threshold values for different levels of priority that may be edited as shown in the following dialog. The table shows the lower limit of the frame priority that will use the threshold values in the T_Pr column. Priorities in COMNET III are such that the larger the number the higher the priority. Generally the threshold token-rotation times should increase for increasing priorities to allow those priorities to use the link longer than the lower priorities.

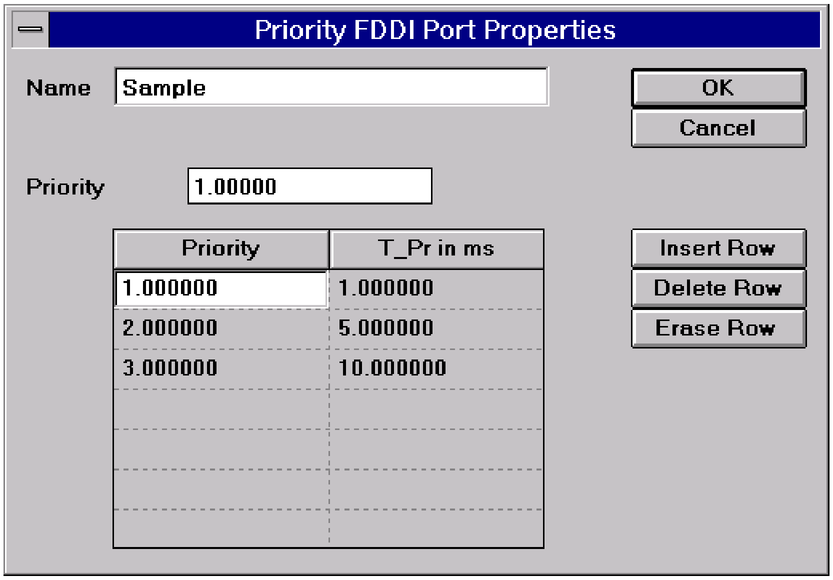


Figure 5. Priority FDDI port properties

The priority FDDI parameter sets maintain this list of port properties so that the port properties may be assigned to a set of ports requiring the same parameters. Thus, the port properties dialog reuses data for multiple ports, while the parameter set itself allows the data to be reused for different links.

The port properties are assigned to the various ports through the port properties dialog attached to the arc between the node and the link icons. The bottom field of the port properties dialog allows the assignment of a specific priority-FDDI port parameter set to be assigned to this port. The ".." button next the field will view the selected parameter set, but will not allow editing. Editing the parameter set can only be done through the link's parameter dialog because these port parameters may be used in multiple places.

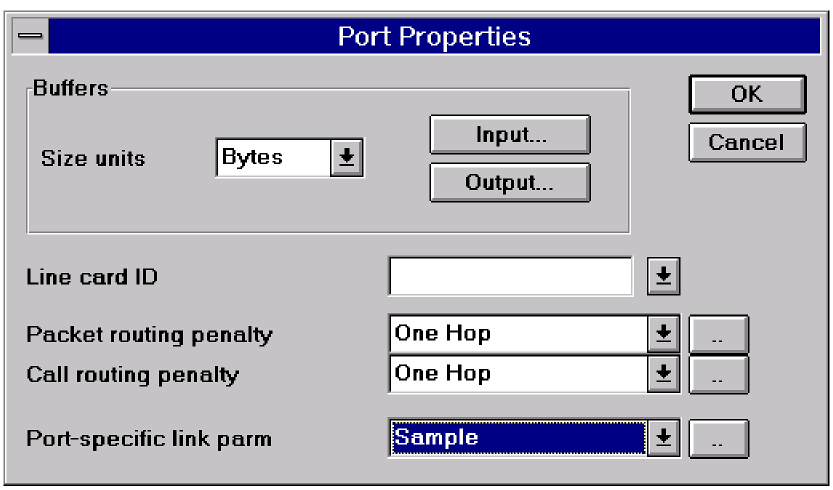


Figure 6. Port properties (on arc between node and priority-FDDI link)

The priority FDDI is closely related to the priority scheme used in the IEEE 802.4 standard except that in 802.4 the highest priority frame has permission to transmit for a holding limit instead of a token-rotation threshold time. The Priority FDDI parameter set has an option to use this 802.4 option of priority operation and provides a field for specifying the holding limit value.

8. Priority Token Ring and Token Ring Enhancements

Release 1.3 improves the token passing model for better performance, more flexibility, and better monitoring. In addition new link types derived from the token ring model provide better models for priority token-rings (including 100VG-AnyLAN), and FDDI.

The token-passing model has a new option for a holding limit based on the number of frames instead of a holding time, and this may be particularly useful when the node may hold a token for exactly one frame.

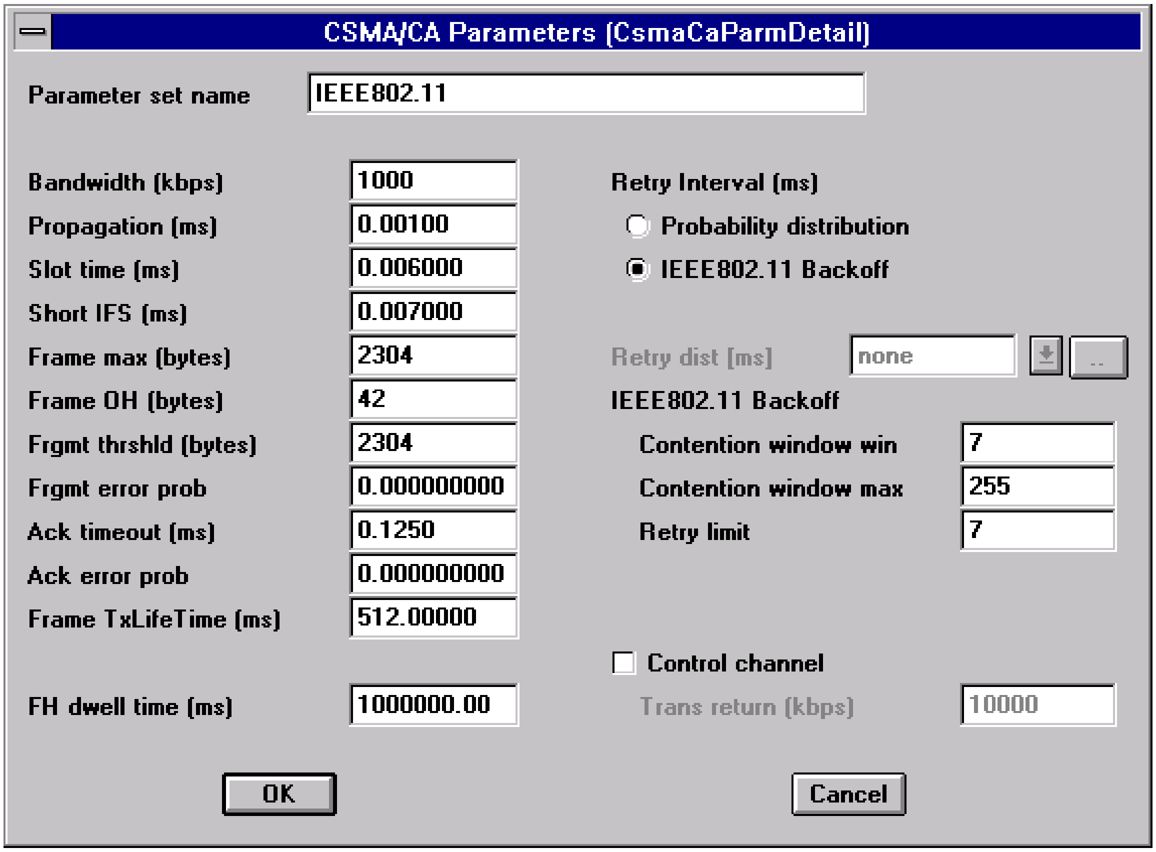
New statistics monitors and a token-ring report are available to measure the actual token rotation times, and the number of active nodes waiting for the token and how long these nodes remain actively accessing the link. These monitors and reports are available for all the token-passing link types (token passing, priority token ring, FDDI, priority FDDI).

The priority token-ring model provides parameter sets for IEEE 802.4 and 802.5 token rings, and 2 new parameter sets for 100VG-AnyLAN: 1 set for ethernet ports and 1 set for token ring ports.

9. IEEE 802.11 CSMA/CA Wireless LAN

The CSMA/CA model is a shared medium access mechanism based on the Distributed Convergence Function (DCF) of the IEEE 802.11 Wireless LAN specification. The basic procedure for accessing the shared medium is as follows.

Each station wishing to transmit first senses the medium to determine if another station is transmitting. If the medium is not busy, the transmission may proceed. The IEEE802.11 distributed algorithm mandates that a gap of a minimum specified duration (called the Distributed Inter-Frame Space, or DIFS) exists between contiguous frame transmissions. A transmitting station ensures that the medium is idle for this DIFS duration before attempting to transmit. If the medium is sensed busy, the station defers until the end of the current transmission. After deferral, or prior to attempting to transmit again immediately after a successful transmission, the station selects a random backoff interval and decrements the backoff interval counter while the medium is free. When the backoff interval counter reaches zero, the station proceeds to transmit the data. The CSMA/CA protocol is designed to reduce rather than eliminate the collision probability between multiple stations accessing a medium at the point where collisions would most likely occur. Just after the medium becomes free following a busy period is when the highest probability of a collision occurs. Therefore a random backoff arrangement is immediately applied in this situation. In addition, all directed (non-multicast) traffic uses immediate positive acknowledgments (ACK frames). Retransmission is scheduled by the sender if no ACK is received.



- Bandwidth: This parameter defines the bit rate supported by the 802.11 wireless LAN physical medium. The supported data rates in the IEEE standard are either 1 Mbps or 2 Mbps (but COMNET III does not limit you to these values).

- Propagation: This parameter specifies the air propagation time it takes a transmitted signal to go from the transmitting station to the receiving station. A nominal value of 1 microsecond has been allocated for this parameter in the IEEE 802.11.

- Slot time: The parameter *Slot time* is used to define *the Short* and *Distributed Interframe Spaces* and to update the *backoff interval* in the shared medium access algorithm. A nominal value of 6 microseconds is specified for the infrared physical system. *Users shall specify this value according to their manufacturer's product specification.*

- Short IFS: The *Short Interframe Space* is the shortest of the interframe spaces defined in IEEE 802.11. It is used to provide an efficient MAC Service Data Unit (MSDU) delivery mechanism. Once the medium is seized, the station will keep it for the duration of the frame exchange it has to perform. In the implemented model, the *Short IFS* is used between fragments of one MSDU and their ACK frames. Using the smallest gap between transmissions within the frame exchange prevents other stations, which are required to wait for the medium to be free for a longer gap (the *Distributed IFS*, see below) from attempting to use the medium, thus giving priority to completion of the frame exchange in progress.

  A nominal value of 7 microseconds is specified for infrared physical system. *User shall specify this parameter based on manufacturer's product specification.*

  The *Short IFS* (*SIFS*) is also used to compute the *Distributed IFS* (*DIFS*) based on the following equation:

  *Distributed IFS = Short IFS + 2 * Slot Time*

  The *DIFS* is used by the CSMA/CA model to contend for the channel for initial frame transmissions. A station is allowed to transmit if it detects the medium to be free for the *DIFS* duration and its backoff time has expired.

- Frame max: This parameter specifies the maximum MSDU length that will be accepted for transmission. The default value is 2304 octets.

- Frame OH: The frame overhead parameter includes the 802.11 MAC header, an IEEE 32-bit CRC, and 8 octets of shared key for authentication. In the case of fragmentation (see below), this is also the overhead of transmitting a fragment. The default value of the frame overhead is 42 octets.

- Frgmt thrshld: The fragmentation threshold specifies the current maximum size, in octets, of the Mac Protocol Data Unit (MPDU) that will be delivered to the physical layer. The default value is 2304, which is equal to the maximum frame size. Fragmentation creates MPDUs smaller than the MSDU (frame) size to increase reliability by increasing the probability of successful transmission of the MSDU (frame) in cases where channel characteristics limit transmission reliability for longer frames. When a frame is received with MSDU size greater than the specified "Fragmentation Threshold," the frame must be fragmented. The MPDUs (or fragments) resulting from the fragmentation of an MSDU are sent as independent transmissions, each of which is separately acknowledged. This permits transmission retries to occur per fragment, rather than per MSDU (frame).

Unless interrupted due to medium occupancy limitations for a given physical system, such as a dwell time boundary in a Frequency Hopping Spread Spectrum (FHSS) system, the fragments of a single MSDU are sent as a burst, using a single invocation of the CSMA/CA medium access procedure. In other words, once the station has contended for the channel, it will continue to send fragments (by using the *Short Interframe Space* instead of the *Distributed Interframe Space*) until either all fragments of a MSDU have been sent, an acknowledgment is not received for directed data due to collision or transmission error, or the station can not send any additional fragments due to a dwell time boundary. Should the sending of the fragments be interrupted due to one of these reasons, the frame or the fragment will enter a retransmission backoff period; when the next opportunity for transmission occurs the station resumes sending the rest of the fragments of the frame.

- Frgmt error prob: This parameter specifies the error rate of transmitting one fragment (or frame, in case of no fragmentation) over the physical medium.

- ACK timeout: This parameter specifies the length of time by which an ACK frame should be received in response to transmission of a frame which requires acknowledgment. It is timed from the instant that the transmitting station finishes the transmission of the frame or fragment.

  The reception of an unicast frame (or fragment in the case where fragmentation is needed) requires the receiving station to respond with an ACK frame, if the received frame (fragment) is correct. Lack of reception of an expected ACK frame indicates to the source station that an error has occurred and causes the station to retransmit the frame (or fragment). An erroneous ACK frame causes the retransmission of the transmitted frame (or fragment) as well. The transmission of the ACK frame commences after the *Short IFS* without regard to the busy/free state of the medium.

  The *ACK Timeout* may not be less than (*2 * Propagation + Short IFS + Ack Transmission Time*). Otherwise an error message will occur in the dialog box.

- ACK error prob: This parameter specifies the error rate of transmitting an ACK frame over the physical medium. An ACK frame is 14 octets long.

- Frame TxLifeTime: The frame transmission life time parameter is defined as the elapsed time, after initial transmission of an MSDU, after which further attempts to transmit the MSDU will be terminated. Therefore, this parameter is per frame and not per fragment. The default value is 512 milliseconds.

- FH dwell time: The FH dwell time parameter is specified for an FHSS physical system. The attribute is defined as the time the physical medium dependent system can dwell on an FH channel and meet the requirements of the current regulatory domain.

  In the implemented CSMA/CA model, the FH dwell time is used for determining if the station can retain the control of the channel during the process of transmitting fragments of an MSDU. Once the dwell boundary is up, the frame has to enter the backoff period and contend again for the channel.

- IEEE 802.11 Backoff: This part of the dialog box provides a group of parameters for the backoff algorithm. A station that desires to initiate transfer of data and finds the medium busy follows the backoff procedure. To begin the backoff procedure, the station selects a backoff time based on the following equations.

  *Backoff Time = INT(CW * Random()) * Slot Time*

  where:

  CW = An integer between the values of CW min and CW max
  Random() = Pseudo-random number between 0 and 1
  Slot Time = the value specified above

  The *Contention Window (CW)* parameter takes an initial value of the *CW min* for every MPDU queued for transmission. The *CW* will take the next value in the series at every retry to send a particular MPDU until it reaches the value of the *CW max*. The set of *CW* values are 7 (*CW min*), 15, 31, 63, 127, 255 (*CW max*).

  The parameter "*Retry limit*" indicates the maximum number of retransmission attempts of a fragment. It is set as the default value 7. Once this limit is reached, a failure condition is indicated to the next higher layer. Note that the statistics report on "Number of tries to resolve" may have a bigger value than this parameter. This is because the statistics field is on frame basis while the retry limit in the implemented CSMA/CA is on fragment basis.

  All backoff periods occur following a DIFS period during which the medium is free. A station in backoff monitors the medium for carrier activity during backoff periods. If no carrier activity is seen for the duration of a particular slot, then the random backoff process decrements the backoff timer by *Slot time* specified above. If there is carrier activity sensed at any time during a backoff period, then the backoff procedure is suspended; that is, the backoff timer will not be decrement for that slot. The medium must be sensed as idle again for the duration of a DIFS before the backoff procedure is allowed to resume. Transmission commences whenever the backoff timer reaches zero.

A station that had just transmitted an MSDU and has another MSDU ready to transmit will perform the backoff procedure in order to produce a level of fairness of access to the medium amongst stations.

Some of the performance measures on the output reports have special interpretations for CSMA/CA links. On the Channel Utilization report, the column for FRAMES RST/ERR shows a count of the number of frames that reach the Retry Limit or the XmitLifeTime. On the same report, utilization does not include the time required for transmissions in error. On the Collision Stats report, NBR OF TRIES TO RESOLVE includes retransmissions due to both collisions and errors.

10. Demand-Assigned Multiple Access Link (TDMA, FDMA, CDMA)

The Demand-Assigned Multiple Access (DAMA) link allows multiple nodes to share a pool of circuits. The link has the same set of parameters as a point-to-point link and can carry both packets and circuit-switched calls. As with the point-to-point link, the DAMA link allows calls and packets to share the same bandwidth using the call-based link loading option. Each node connected to the DAMA link can access only 1 circuit of the DAMA link's pool of circuits at a time. A circuit is allocated to a node for the amount of time required to transmit a packet. When several nodes are waiting to transmit packets, the node with the earliest arriving packet uses the first circuit to become available.

C. COMNET Baseliner

COMNET Baseliner is an interface utility provided within the COMNET III application which is used to import message packets captured from an actual network into a model as well as network topology files gathered by network management consoles. The utility produces an external message file which maps the packets captured in the data file to the names of the nodes in the model. When a simulation is run, the packets captured are routed across the modeled network from their source to destination. A complete description of the method used to import traffic data files is given in The COMNET Baseliner User's Manual.

1. Capturing Packets

The capturing of packets on a network may be performed using either a hardware or software packet analyzer. Both methods monitor the transmission of packets across a network in a promiscuous mode which implies that all packets on the network are captured and stored for later use. COMNET III directly supports importing traffic files from the HP NetMetrix network analyzer, the Network General Sniffer LAN hardware analyzer, and Expert Sniffer analyzer, the Wandel & Goltermann's Domino Analyzer, Frontier Technology's RMON probe, and CastleRock's SMNP probe. Other packet analyzers may be used to collect data files, but the file must be formatted into a text space delimited file and format information entered into the COMNET Baseliner utility in order to create the external traffic file to run in a simulation. Important considerations when selecting a traffic analyzer include:

- The protocols supported by the analyzer.
- Time stamp precision for capturing packets.
- The ability of the analyzer to provide the necessary information to import the captured data file into COMNET III.
- The total number of packets the analyzer is capable of capturing and saving in one run without dropping packets.

In order for a captured data file to be imported into COMNET III the file must contain the following information:

- The delta or elapsed time between captured packets.
- The packet size in bytes.
- The source or node which generated the packet.
- The destination of the packet.
- The packet identity.

2. Scaling External Traffic Files

Once an external message file has been created, a scale factor may be applied to this external file to represent an increase or decrease in network loading when running a simulation. For example, to represent a 40 percent increase in traffic on a network a scale factor of 1.4 is used. The traffic is scaled when running a simulation by either shortening the delta time between packets to model an increase in network loading or lengthening the delta time between packets to model a decrease in network loading. The total number of packets in the external file does not change in either case. The implication of using a scaling factor with an external file is that the simulation run time will also need to be adjusted by the same scaling factor for simulation results to have the greatest accuracy.

D. PROBLEM STATEMENT

On February 7, 1996, the Systems Technology Lab's 10Base2 Ethernet network was experiencing excessive delays. In order to determine the cause of these delays the protocol analyzer Snoop, which is part of the Sun Solaris release 2.4 operating system, was used to capture packets from the network and saved to a file. The desire of the lab personnel was to build a model of the network topology and use the data collected to determine the network loading and delays at the time the data was captured.

E. ANALYSIS

The capturing of packets on the Systems Technology Lab's ethernet network was performed running the Snoop application on a Sun Sparc 20 workstation. The Snoop protocol analyzer was chosen to capture packets as it was the only protocol analyzer application available in the Systems Technology Lab, and it provided the necessary information for importing a data file into COMNET III. The command used to capture the packets was as follows:

snoop -d le1 -s 34 -c 2000 -o /tmp/1net63 -t d -S

This string entered in the UNIX command window is interpreted as follows [Ref. 6]:

- snoop: Start the snoop application
- -d le1: Capture packets from the ethernet network
- -s 34: Truncate each packet so that only the first 34 header bytes are saved
- -c 2000: Capture 2000 packets
- -o /tmp/1net63: Save packets captured to the file 1net63

- -t d: Use the delta time between packets as the time step
- -S : Record the total length in bytes of each packet

Only 2000 packets were captured as previous attempts to capture 10000 and 5000 packets resulted in core dumps of the workstation. This was later determined to have been caused by inadequate disk space on the workstation to write the packets as they were captured. The decision to truncate the packets was made to reduce the amount of data which would need to be written to a file and to minimize the chance of dropping packets while writing to the disk which results in gaps of lost packets. The second problem had been shown to happen in previous experimentation with the application when the entire packet was saved.

In order to convert the data file saved to a form which was palatable for COMNET III, the file was first converted from a binary format to a text format. This text file was then edited using a spreadsheet application to remove extraneous information and to provide only the data needed within COMNET III of source, destination, length, time stamp, and a unique identifier for which the protocol identifier was used. The file was saved in the text space-delimited (*.prn) format required for COMNET III. A word processing program was then used to determine column start points in the file in order to be able to describe the format of the file to the COMNET Baseliner utility. The packets captured were also looked at to determine the total amount of time over which packets were captured to set the simulation and the nodes these packets were routed which is needed to build the model topology.

F. BUILDING THE MODEL

This section gives step-by-step instructions for each object which will be created in making the model used to analyze the problem statement. Figure 6.7 displays a view of the model topology upon completion.

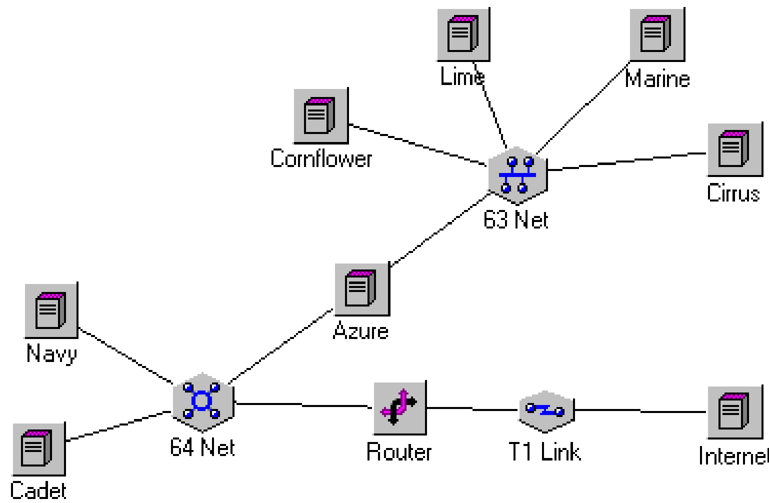


Figure 6.7 Completed Model Topology

1. Creating the Network Links

The two networks which make up the Systems Technology Lab will first be created to model the 10Base2 ethernet network and FDDI networks located in the lab

a) Create a collision and token passing link in the workspace.

b) Edit the collision link to the following parameters in the link detail window:

- Name: 63 Net
- Type: CSMA/CD
- Parameters: 802.3 CSMA/CD 10BASE2

c) At the bottom of the link detail window press the *Statistics* button and edit the list so that the observations are saved and statistics are gathered for both the *Contention Channel Utilization* and *Contention Channel Transmission Delay* options in the list. When these options are set, clear the link detail window by pressing the *OK* button.

d) Edit the token passing link to the following parameters in the link detail window:

- Name: 64 Net
- Type: Token passing
- Parameters: FDDI

e) At the bottom of the link detail window press the *Statistics* button and edit the list so that the observations are saved and statistics are gathered for both the *Utilization* and

Transmission Delay options in the list. When these options are set, clear the link detail window by pressing the *OK* button.

2. Creating the Computer Parameter Set

The computer and communications node parameter set that is defined will be used to represent the nodes on both networks. Since no applications will be defined in this model, only the port processing delays will be modeled.

a) Define a computer and communications node parameter set using the *Node/Computer and Communication* option on the *Define* menu.

b) Press the *Add* button in the computer and communication parameters window which appears, and edit the following fields in the node parameter window:

- Name: STL Workstation
- Port processing default time: 0.05 milliseconds for both the input and output buffers

c) Press the *OK* button in the node parameters window and the *Done* button in the computer and communications node parameters window to clear these windows.

3. Creating a T1 Point-To-Point Link

The T1 point-to-point link will be used to model the connection of the FDDI network to outside sources.

a) Create a point-to-point link and place it in the vicinity of the 64 Net link.

b) Edit the point-to-point link fields in the link detail window to the following:

- Name: T1 Link
- Type: Point-to-point
- Parameters: T1

c) When all fields are entered, press the *OK* button at the bottom of the link detail window.

4. Creating the Network Nodes

The review of the data captured revealed that packets were captured from seven nodes on the network. Several sources outside the network were also captured. These sources will be modeled as all coming from a single node connected to the T1 link.

a) Create and edit eight computer and communications nodes so that the fields in the node parameters window for each is set to the values shown in Table 6.1. After creating and editing each node, attach the node the link(s) also defined in Table 6.1. It is important that the node names are typed as explicitly shown for later matching names when creating the external traffic file for the model.

Node Name	Type	Parameters	Link
Cadet	C&C Node	STL Workstation	64 Net
Navy	C&C Node	STL Workstation	64 Net
Cornflower	C&C Node	STL Workstation	63 Net
Lime	C&C Node	STL Workstation	63 Net
Marine	C&C Node	STL Workstation	63 Net
Cirrus	C&C Node	STL Workstation	63 Net
Azure	C&C Node	STL Workstation	63 Net & 64 Net
Internet	C&C Node	STL Workstation	T1 Link

Table 6.1 Computer and Communication Nodes Parameters

b) Create a router node and edit the fields in the node detail window to the following:

- Name: Router
- Type: Router
- Parameters: Proteon DNX 300, V12.0

c) Connect the router node to both the T1 Link and the 64 Net link

d) The network topology is now defined. Save the model at this time.

5. Creating the Model's External Traffic Source

The following steps describe the manner in which the COMNET Baseliner utility creates the external message file from the captured network traffic:

a) From the *Define* menu select the *External Traffic/Messages* option. The window shown in Figure 6.8 will then appear.

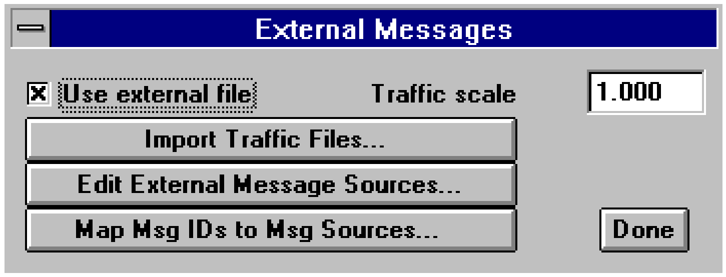


Figure 6.8 External Messages Window

b) Press the *Import Traffic Files* button and the window shown in Figure 6.9 will appear.

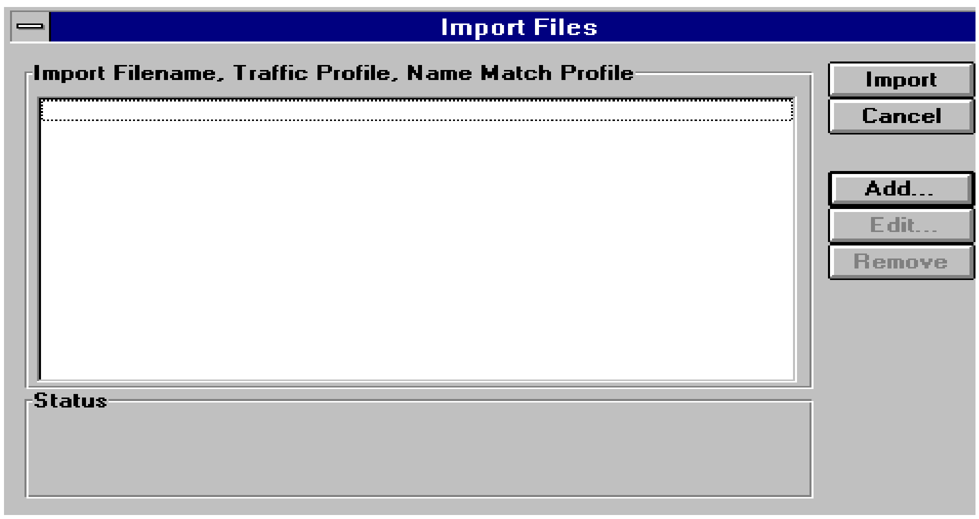


Figure 6.9 COMNET Baseliner Import Files Dialog Window

c) Click the *Add* button and the following dialog window in Figure 6.9a will appear:

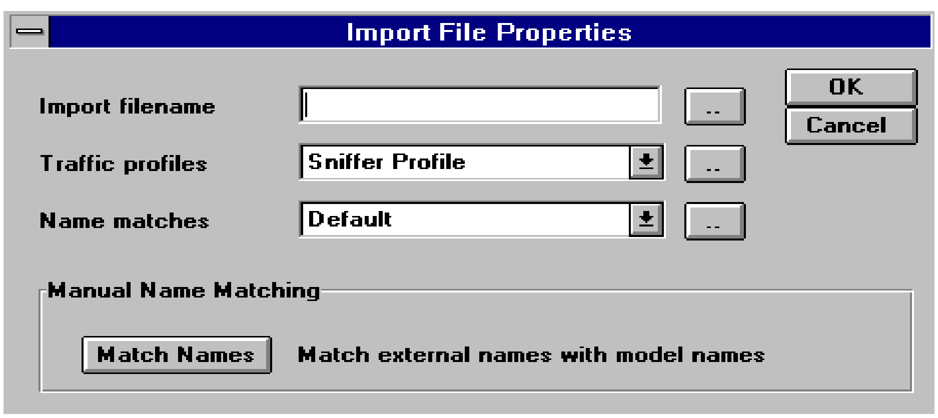


Figure 6.9a Import File Properties Window

d) Using the button with double dots on it at the end of the Import filename field, select the file **stl3.prn** .

e) Click the double dot button next to the Traffic profiles field. This will open up the External Traffic Profiles dialog window shown in Figure 9.6b.

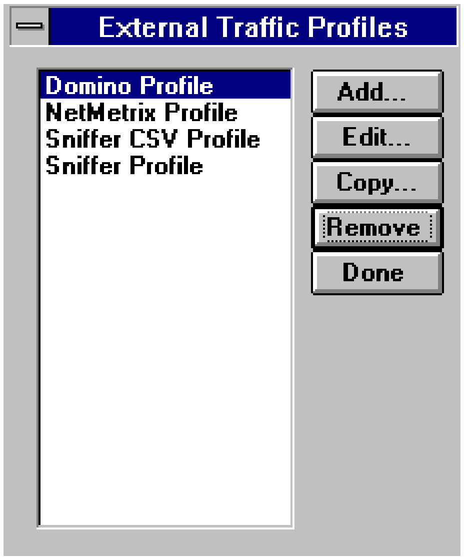


Figure 6.9b External Traffic Profiles

f) Click on the *Add* button in order to add a Traffic Profile for the **stl3.prn** external traffic file. This will open up the Library Selections dialog window shown in Figure 6.9c.

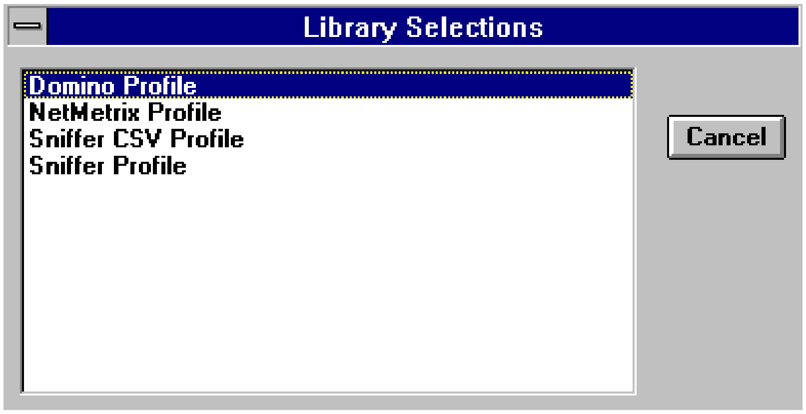


Figure 6.9c Library Selections

g) Single click on the *Sniffer Profile* listing. This will open up the Traffic Profile Properties dialog window shown in Figure 6.9d.

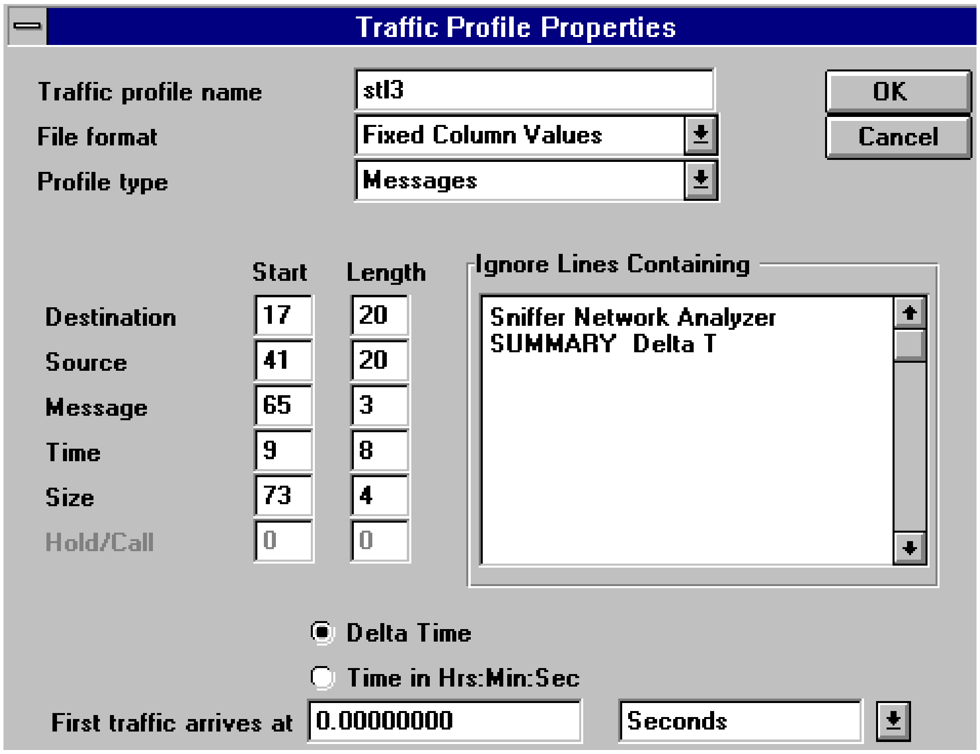


Figure 6.9d Traffic Profile Properties

h) Edit the fields on fields shown in Figure 6.9d to the following:

- Source Start: 17
- Source Length: 20
- Destination Start: 41
- Destination Length: 20
- Message/ID Start: 65
- Message/ID Length: 3
- Time Start: 9
- Time Length: 8
- Size: 73
- Size Length: 4
- Overhead per Message: 0
- Ignore Lines Containing fields: All fields cleared
- Time in Hours Minutes Seconds box: Not checked

g) Type ***stl3*** into the Traffic Profile name field and press the *OK* button to save the profile entered for later use. Click the *Done* button in the External Traffic Profiles Window. This creates a .pro file which may be reloaded if needed at a later time.

h) Next click the Match Names button in the lower portion of the Import File Properties dialog window. This will open up the Name Match Profile dialog window which will allow you to match up the external traffic file events contained in **stl3.prn** with the nodes contained in the COMNET III model. Type in the name ***stl3*** in the Name Match Profile Field, and then click on the AutoMatch button located on the right side of the dialog window. The AutoMatch feature will match up the traffic sources from the external **stl3.prn** traffic file to the nodes defined in the model. The result will be what appears in Figure 9.6e shown below:

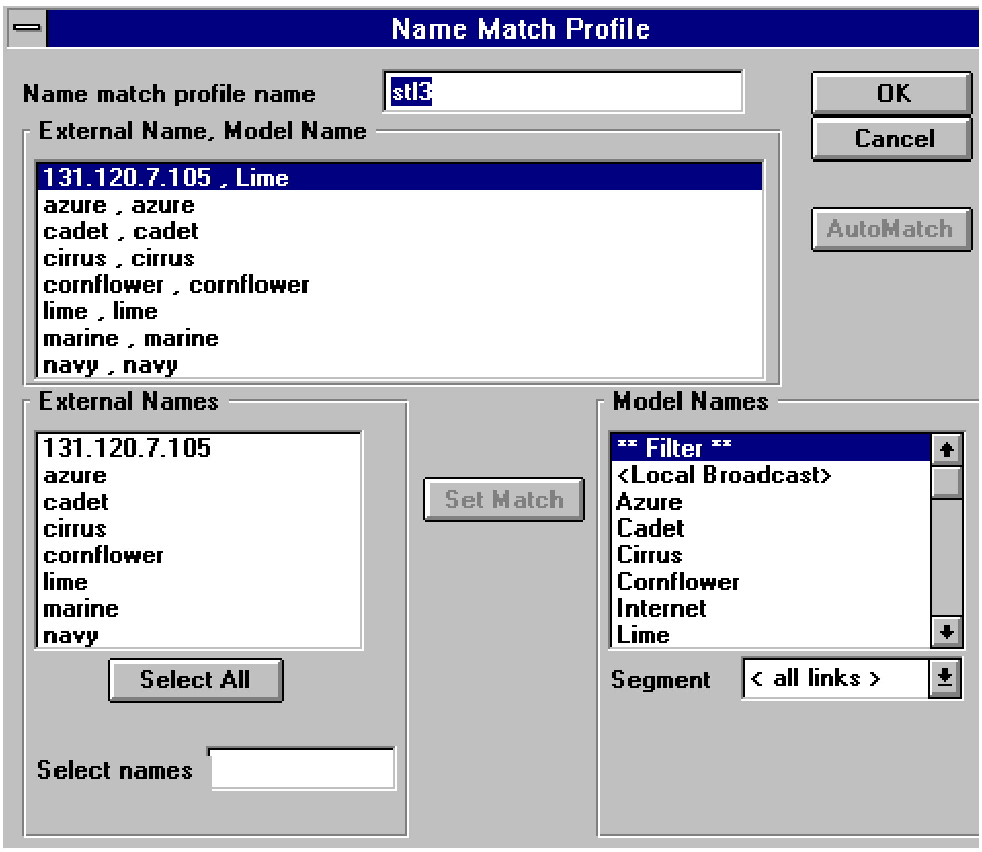


Figure 6.9d **stl3** Name Match Profile

h) Press the *OK* button this will return to the Import File Properties dialog window shown in Figure 6.9a.

i) Press the *OK* button in the Import File Properties dialog window. This will return to the Import Files dialog window shown below in Figure 6.9e.

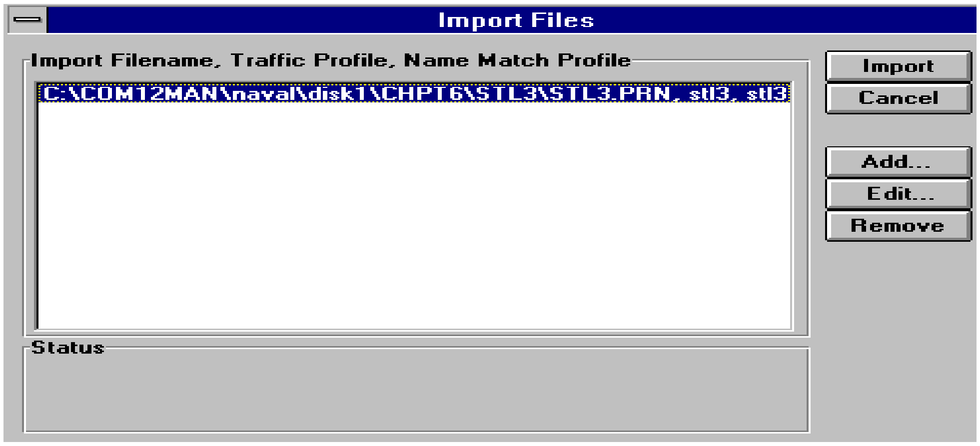


Figure 6.9e Import Files

j) Press the *Import* button to start the import of the external traffic data. When the import is done press the *OK* button.

k) Verify and save the model.

G. SIMULATION AND RESULTS

The amount of time packets were captured amounted to 3.3 seconds of data. As such, the simulation run time should be set to 3.3 seconds. Prior to running the simulation select the following reports:

- Node Reports: Received Message Counts for all nodes
- Link Reports: Channel Utilization for all links in the model
- Message and Response Source Reports: Message Delay for all nodes

The reports generated after running the simulation calculated the utilization of the ethernet network to be at 9.2 percent as an average for the 3.3 seconds of simulation time, and an average transmission delay of 0.191 milliseconds. Plots of the link utilization and delay which are shown in Figures 6.15 and 6.16 respectively over the simulation time interval were produced by plotting the observations saved for the ethernet link. These plots indicated the transmission delay to be falling off from the beginning to the end of the data captured and the utilization demonstrates the burstiness of traffic on the network.

The utilization and delays calculated by the model seemed to be less than those experienced on the network the day the data file was captured. The message delays for message and response sources does indicate which nodes were generating the majority of the traffic on this day. By investigating who was running which application on what workstation the problems with the network were discovered. One problem was caused by a student running the MATLAB application on an SGI Indy workstation named Cornflower which was generating large numbers of file requests to the file server named Navy. In addition, a professor using the workstation named Cirrus had established an MBONE session where a large number of multicast UDP packets were sent across the network. The student was asked to run the MATLAB application during the evening to minimize the effects on the network.

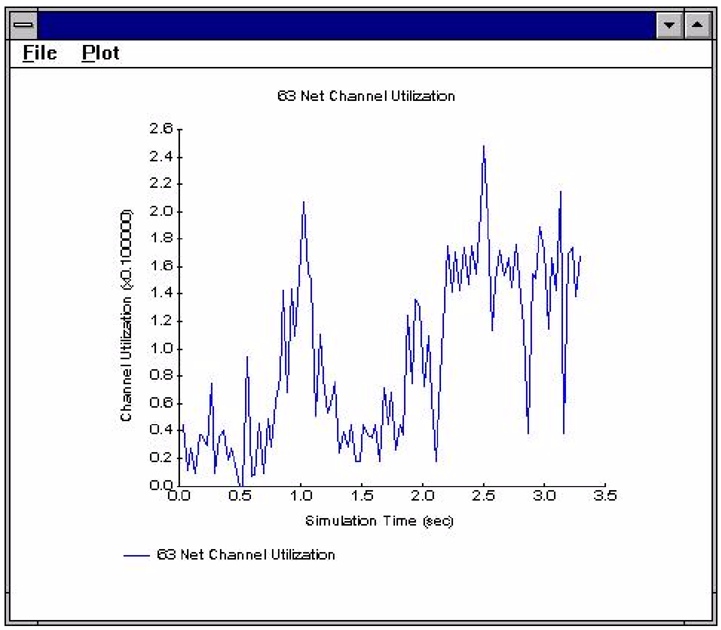


Figure 6.15 Ethernet Utilization

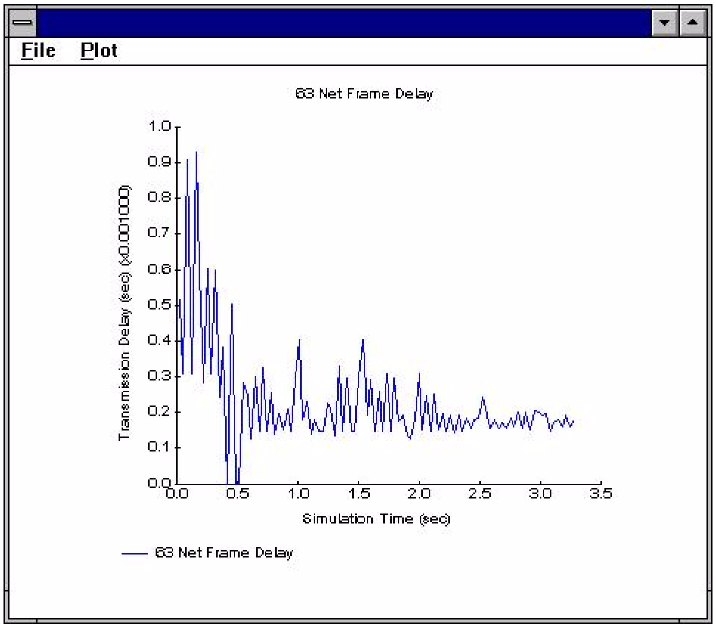


Figure 6.16 Ethernet Transmission Delay

VII. ROUTERS

A. INTRODUCTION

The goal of this chapter is to introduce the characteristics and parameters associated with the router node in the COMNET III application. A network model describing the use of a router in the segmentation of a local area network will be used to demonstrate the use of the router node.

B. ROUTER NODES

The router node was developed in the COMNET III application to allow the modeling of router-based networks in a packet switched environment. The node was designed to model network hardware such as routers, hubs, and bridges. When modeling a router node, the premise used is that the performance of the node may be described by the input and output buffering characteristics of the node and its internal switching rate [Ref. 2 p. 135]. In actuality, the router node is a computer and communications node which has the additional characteristic of modeling the transmission rate of the node's internal bus for switching packets.

Router nodes may be connected to any type of link available in COMNET III. Each link connected to the node is modeled as having its own port corresponding to a line card in the node. Editing of each port attached to the node allows entering a line card identification. Ports having the same non-blank line card identification are modeled as being connected to the same line card in the node. Packets which are routed through the same line card in the node will not incur a bus transfer delay as they will not have to be switched across the nodes internal bus. Ports having different or blank line card identification require switching across the internal bus when routing decisions are made within the node.

The logic applied when routing packets through a router node in a simulation is performed in the following manner. When all frames which make up the packet arrive at the node, the packet is attempted to be placed in the node's input buffer. Tests are performed in the same manner as for a computer and communications node to determine if there is space in the port buffer and that the total buffer capacity of the node is not exceeded. If the packet is accepted, it is placed in the input buffer based on its priority and ordered in first-in-first-out order. The packet will then undergo the port processing delay after which it is eligible to be switched across the node's internal bus if it needs to be routed to a different line card. The switching is modeled as a time delay which is based on the size of the packet to be switched and the transmission rate of the internal bus. The choice of which packet is to be switched across the bus is made by determining which packets have waited the longest in the various input buffers. Once the packet has been switched, it is attempted to be placed in an output buffer. Tests are performed again to determine if the output buffer capacity and the total node capacity will be exceeded as with the input buffer. The packet then undergoes an output port processing delay prior to being transmitted.

Router nodes may be created by using either the toolbar or the *Create* menu. Editing of the parameters of a router node will cause the window shown in Figure 7.1 to appear. The fields used to define the modeling characteristics of the router node are the same as the computer and communications node with the following exceptions:

- Bus rate: Specifies the node's internal bus transmission rate.
- Bus count: Specifies the number of packets which the node may switch simultaneously.

The COMNET III library contains parameter sets for various commercial routers built into the system library. The values used to define the operating characteristics of these parameter sets are based on tests performed at the Harvard Network Device Test Laboratory. The tests run at this laboratory generally consist of routing a packet stream through the device and measuring the buffer and switching delays of the device. Several tests are performed where the packet protocol and packet size are varied between tests. The delays measured in each run are then averaged to provide the delay characteristics of the router.

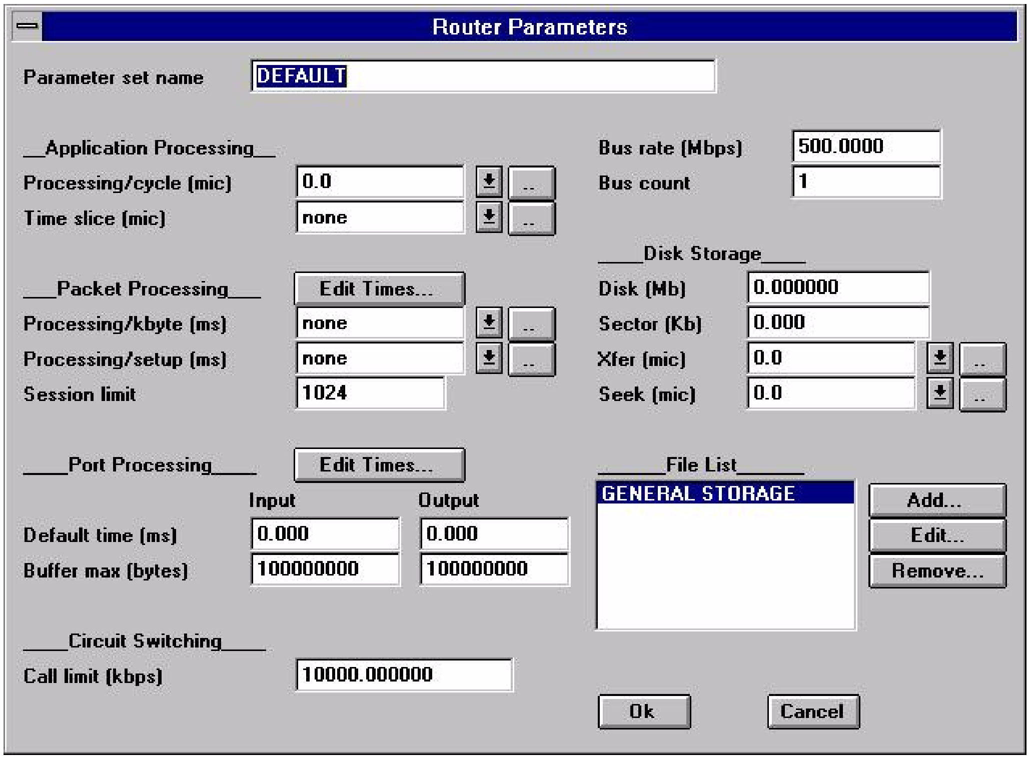


Figure 7.1 Router Parameters Window

C. PROBLEM STATEMENT

The utilization of the Systems Technology Lab's 10Base2 ethernet network is estimated to increase by a factor of 3.5 within the next two years based on the probability of additional computers being placed on the network and the use of computers currently on the network for thesis research or processing. The lab manager also believes the use of multicast backbone (MBONE) utilities available in the laboratory will increase in use as desktop video conferencing becomes a more prevalent method of performing thesis research. There are currently four workstations available to students throughout the lab which have desktop video conferencing capabilities. It is estimated that only two would be used for this purpose at a single time.

The lab personnel desires a model to be built to estimate the network utilization and delays based on the current network configuration. Also, the lab would like a proposal of a method to segment the ethernet network into two separate networks by using a router.

D. ANALYSIS

Three pieces of information are needed to construct the traffic generators which will be used in a model to analyze the problem. The first is an estimate of the network loading on the ethernet network. The second piece of information is an understanding of how the multicast backbone operates in a local area network. The third piece of information needed is estimates of the size and interarrival time of packets generated when using desktop video conferencing tools over the network.

To obtain the estimate of current network loading, the protocol analyzer Snoop was again used to capture packets from the ethernet network. 10000 packets were captured from the ethernet network on April 3,1996. Of these packets, 9136 provided the information necessary for importing into COMNET III using the COMNET Baseliner utility. The total length of time over which the packets were captured was 110 seconds. The data file captured was converted to a text space delimited format using a spreadsheet application. Also, a list was made of nodes from which packets were captured for reference in building the model.

A Silicon Graphics Indy workstation (Cadet) is used as the multicast router on the System Technology Lab's ethernet network. All multicast packets generated using a MBONE utility are directed to the multicast router. The multicast router then broadcasts the packet to all nodes on the network. For transmission to another network, the multicast router will encapsulate the multicast packets to appear to be unicast packets using the UDP/IP protocol to routers along the path to the other network. In order to model this process, a message source using received message scheduling may be used. Upon receipt of a multicast packet, the message source will broadcast the identical message received to all nodes on the network using a multicast list.

To gather the necessary information of packet size and interarrival times for desktop video conferencing, the MBONE application NetVideo was used. A desktop video conference was established on the ethernet network using the NetVideo application. The application was set to deliver packets at a transmission rate of 128 kbps which is the expected transmission rate of video on the multicast backbone [Ref. 7 p. 3]. The Snoop application was again used and was set to capture packets only generated by the workstation used to establish the video conference session. A total of 4000 packets were captured from the workstation of which 3148 were determined to be multicast packets generated from the NetVideo application. This determination was made by reviewing the identity of the protocol of each packet and discarding packets that did not use the UDP/IP multicast protocol.

From the data captured, the goal was to be able to describe the packet size and interarrival time as a statistical distribution which could then be used in a message source to model video conferencing across a network. The histogram shown in Figure 7.2 displays the frequency of packet size for the packets sorted into bins of 10 byte widths. The histogram shown in Figure 7.3 displays the frequency of the interarrival times for packets sorted into bins of 0.01 seconds in width. Neither histogram implied that the data collected could be modeled using one of the statistical distributions available in COMNET III, so the decision was made to model the interarrival times as discrete

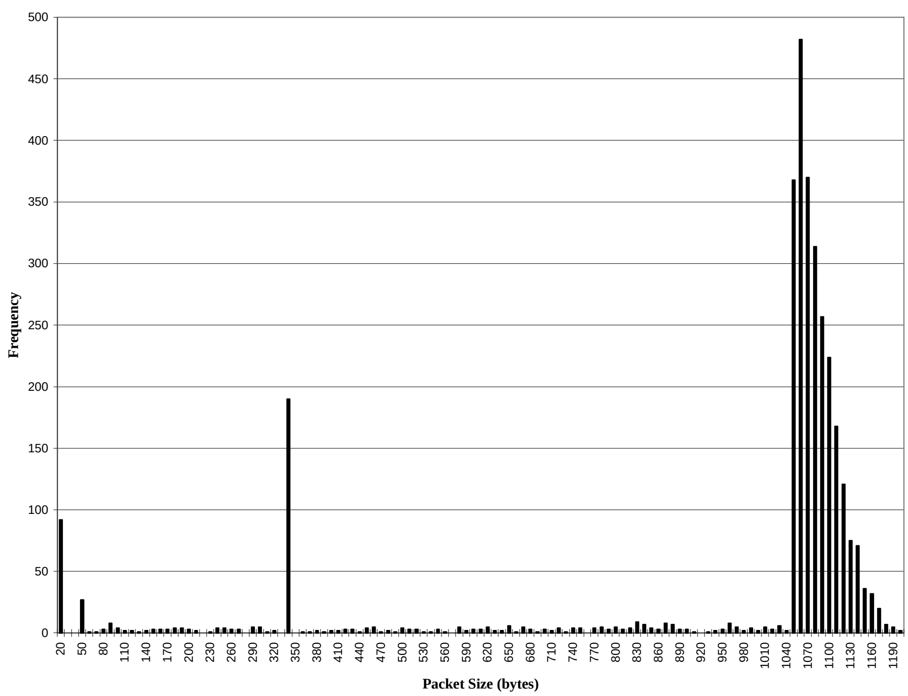


Figure 7.2 Packet Size Histogram

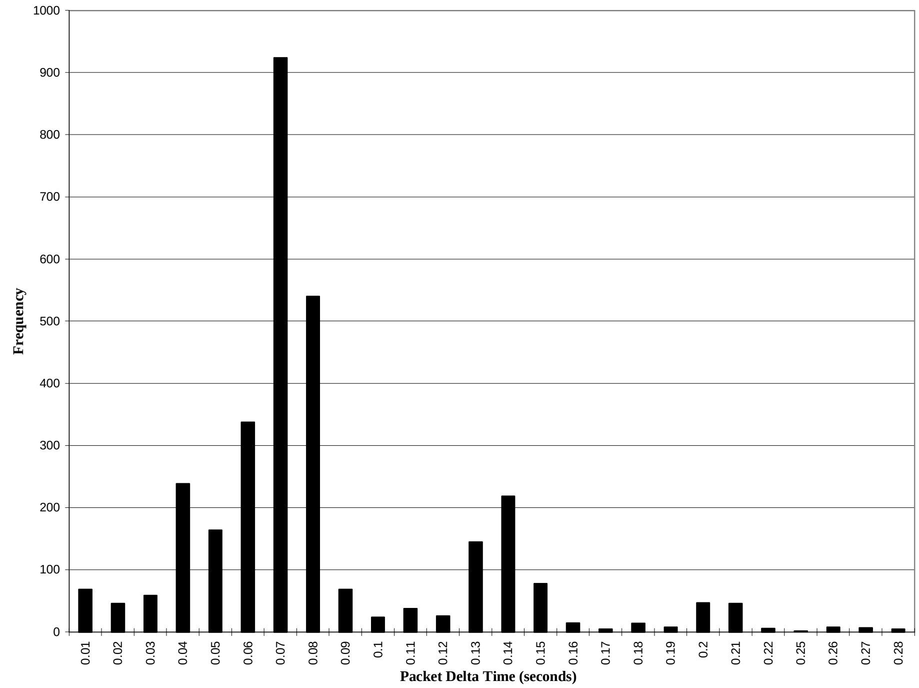


Figure 7.3 Packet Interarrival Time Histogram

tabular distributions. Table 7.1 displays the packet size, probability, and cumulative probability which will be used to create a tabular distribution for modeling the packet

Packet Size (bytes)	Probability	Cumulative Probability
20	0.03	0.03
50	0.01	0.04
340	0.07	0.11
1050	0.13	0.24
1060	0.17	0.41
1070	0.13	0.54
1080	0.11	0.65
1090	0.09	0.74
1100	0.08	0.82
1110	0.06	0.88
1120	0.04	0.92
1130	0.03	0.95
1140	0.02	0.97
1150	0.01	0.98
1160	0.01	0.99
1170	0.01	1.00

Table 7.1 Packet Size and Probabilities

size. Data which sorted into a bin having a probability less than 1 percent was not included in the modeling of the packet size. Table 7.2 displays the packet interarrival time, probability and cumulative probability which will be used to create the tabular distribution for modeling the packet interarrival time. Data which sorted into a bin having a probability less than 0.1 percent was not included for modeling.

Packet Interarrival Time (seconds)	Probability	Cumulative Probability
0.01	0.022	0.022
0.02	0.015	0.037
0.03	0.018	0.055
0.04	0.076	0.131
0.05	0.052	0.183
0.06	0.107	0.290
0.07	0.294	0.584
0.08	0.172	0.756
0.09	0.022	0.778
0.10	0.008	0.786
0.11	0.012	0.798
0.12	0.008	0.806
0.13	0.046	0.852
0.14	0.069	0.921
0.15	0.025	0.946
0.16	0.005	0.951
0.17	0.002	0.953
0.18	0.004	0.957
0.19	0.003	0.960
0.20	0.015	0.975
0.21	0.015	0.990
0.22	0.002	0.992
0.25	0.001	0.993
0.26	0.003	0.996
0.27	0.002	0.998
0.28	0.002	1.000

Table 7.2 Packet Interarrival Times and Probabilities

The final problem to consider prior to building the model is determining a method of segmenting the ethernet network. The method chosen was to segment the network according to the workstations which are capable of desktop video conferencing and those which are not. The reason this method was chosen is due to the manner in which the multicast router broadcasts packets to all nodes on a network and the relatively high bandwidth required for video conferencing over a network. Segmentation of the network in this manner would isolate the other network from all MBONE applications.

E. BUILDING THE MODELS

This section outlines the steps required to build the two models which will be used to analyze the problem in an object-by-object manner. Upon completion of building the two models, they should appear similar to the model topologies shown in Figure 7.4 for the current network topology and Figure 7.5 for the proposed changes.

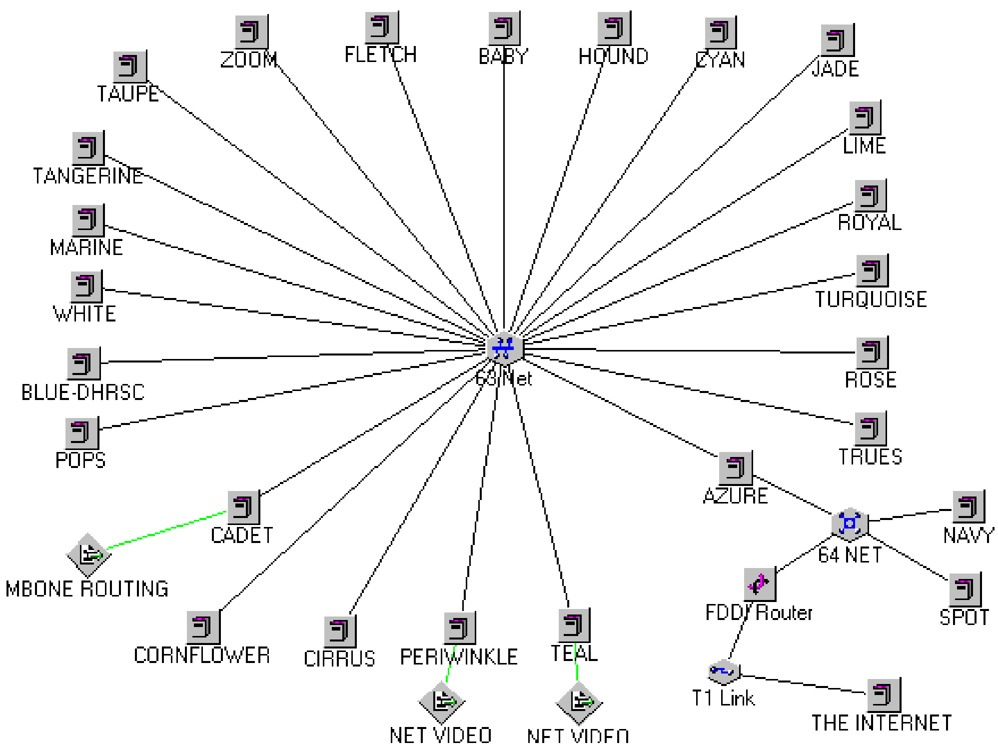


Figure 7.4 Current Network Topology

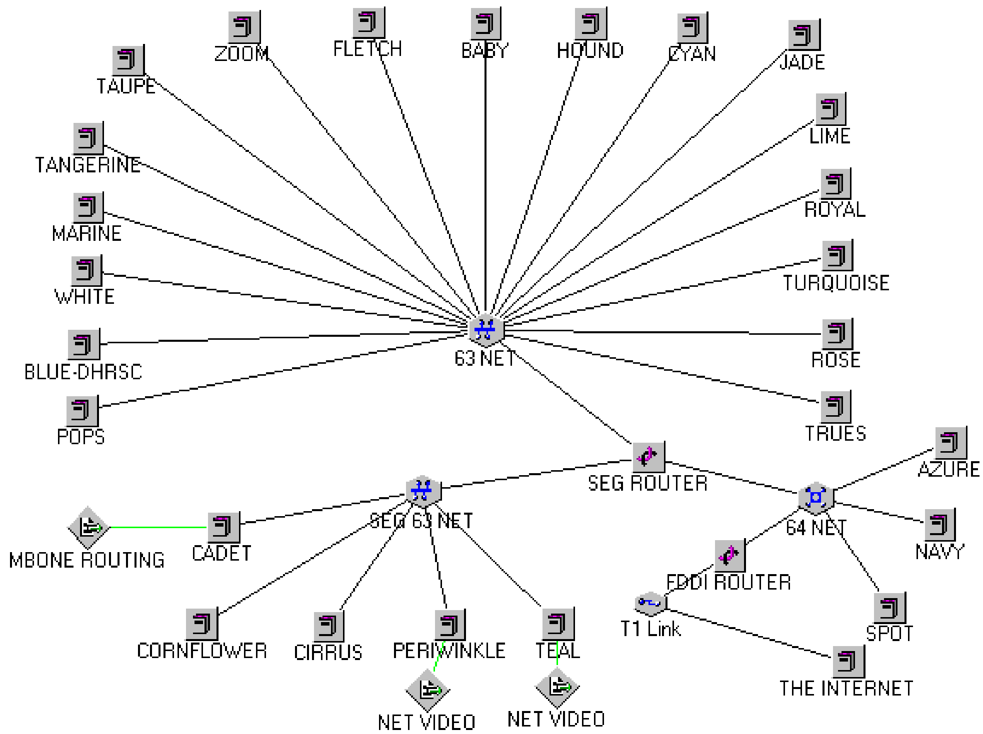


Figure 7.5 Proposed Network Topology

1. Creating the Network Links

The following steps outline the procedure for creating the links which will be used in modeling the network.:

a) Create a collision link, token passing link, and a point-to-point link.

b) Edit the fields in the link detail window of the collision link to the following:

- Name: 63 NET
- Type: CSMA/CD
- Parameters: 802.3 CSMA/CD 10BASE2

c) Press the *Statistics* button at the bottom of the link detail window, and set the contention channel utilization option so that observations are saved and statistics are gathered when running a simulation.

d) Edit the fields in the link detail window for the token passing link to the following :

- Name: 64 NET
- Type: Token passing
- Parameters: FDDI

e) Edit the link detail window for the point-to-point link to the following values:

- Name: T1 Link
- Type: Point-to-Point
- Parameters: T1

2. Creating the Network Nodes

The nodes which are defined in the network are based on the names captured in the data file which will be used to define the external traffic. It is important to name the nodes as explicitly written to allow the name matching to be performed correctly when running the COMNET Baseliner utility. The parameter set which will be used for the computer and communications node is modeled using only the default port processing times. This choice was made as only this attribute of the node affects the traffic generated by the external file and modeled message sources.

a) Define a computer and communications node parameter set and edit the following fields in the node parameters window:

- Name: STL Workstation
- Port Processing Default Time: 0.05 milliseconds for both the input and output buffers.

b) Create 26 computer and communications nodes (C&C nodes). Edit the fields of the node detail window to those shown in Table 7.3. Connect the nodes to the link(s) specified in Table 7.3. When creating the C&C nodes, group the first five nodes listed in Table 7.3 in the same area in the work space.

c) Create a router node and connect it to the T1 and 64 NET links. Edit the fields in the node detail window for the router to the following:

- Name: FDDI ROUTER
- Type: router
- Parameters: Proteon DNX 300, V12.0

Node Name	Type	Parameters	Link
CADET	C&C Node	STL Workstation	63 NET
CORNFLOWER	C&C Node	STL Workstation	63 NET
CIRRUS	C&C Node	STL Workstation	63 NET
PERIWINKLE	C&C Node	STL Workstation	63 NET
TEAL	C&C Node	STL Workstation	63 NET
POPS	C&C Node	STL Workstation	63 NET
BLUE-DHRSC	C&C Node	STL Workstation	63 NET
WHITE	C&C Node	STL Workstation	63 NET
MARINE	C&C Node	STL Workstation	63 NET
TANGERINE	C&C Node	STL Workstation	63 NET
TAUPE	C&C Node	STL Workstation	63 NET
ZOOM	C&C Node	STL Workstation	63 NET
FLETCH	C&C Node	STL Workstation	63 NET
BABY	C&C Node	STL Workstation	63 NET
HOUND	C&C Node	STL Workstation	63 NET
CYAN	C&C Node	STL Workstation	63 NET
JADE	C&C Node	STL Workstation	63 NET
LIME	C&C Node	STL Workstation	63 NET
ROYAL	C&C Node	STL Workstation	63 NET
TURQUOISE	C&C Node	STL Workstation	63 NET
ROSE	C&C Node	STL Workstation	63 NET
TRUES	C&C Node	STL Workstation	63 NET
AZURE	C&C Node	STL Workstation	63 NET & 64 NET
NAVY	C&C Node	STL Workstation	64 NET
SPOT	C&C Node	STL Workstation	64 NET
THE INTERNET	C&C Node	Default	T1 LINK

Table 7.3 Computer and Communication Node Parameters and Links

3. Creating the Tabular Distributions

The tabular distributions created will be used to describe the message size and interarrival times of the NetVideo message source.

a) From the *Define* menu select the *Tables* option.

b) Press the *Add* button in the tabular distributions window, and edit the fields of the table detail window to the following:

- Name: NVSIZE-TAB
- Table type: discrete
- Probabilities and Values: Enter the cumulative probabilities and packet size from Table 7.1 into the probability and values fields respectively.

c) When all fields are entered, press the *OK* button at the bottom table detail window. Then, press the *Add* button in the tabular distributions window to create the second distribution.

d) Edit the fields of the table detail window to the following:

- Name: NVINTER-TAB
- Table type: discrete
- Probabilities and Values: Enter the cumulative probabilities and packet size from Table 7.2 into the probability and values fields respectively.

e) When all fields are entered, press the *OK* button at the bottom table detail window. Then, press the *Done* button in the tabular distributions window to create the second distribution.

4. Creating the NetVideo Message Source

The NetVideo message source will be used to model the broadcast of multicast packets for desktop video conferencing.

a) Create a message source and edit the fields of the message source window to those shown in the following list. Figure 7.6 depicts the message source window upon completion.

- Name: NET VIDEO
- Schedule by: Iteration time
- Interarrival: NVINTER-TAB
- Message size calculation: Probability distribution
- Probability distribution: NVSIZE-TAB
- Priority: 1
- Routing class: Standard
- Transport protocol: UDP/IP-Sun
- Packetize: 0.5 milliseconds
- Message text option: Copy message name
- Destination type: Set to weighted list and create a list containing only the node CADET.

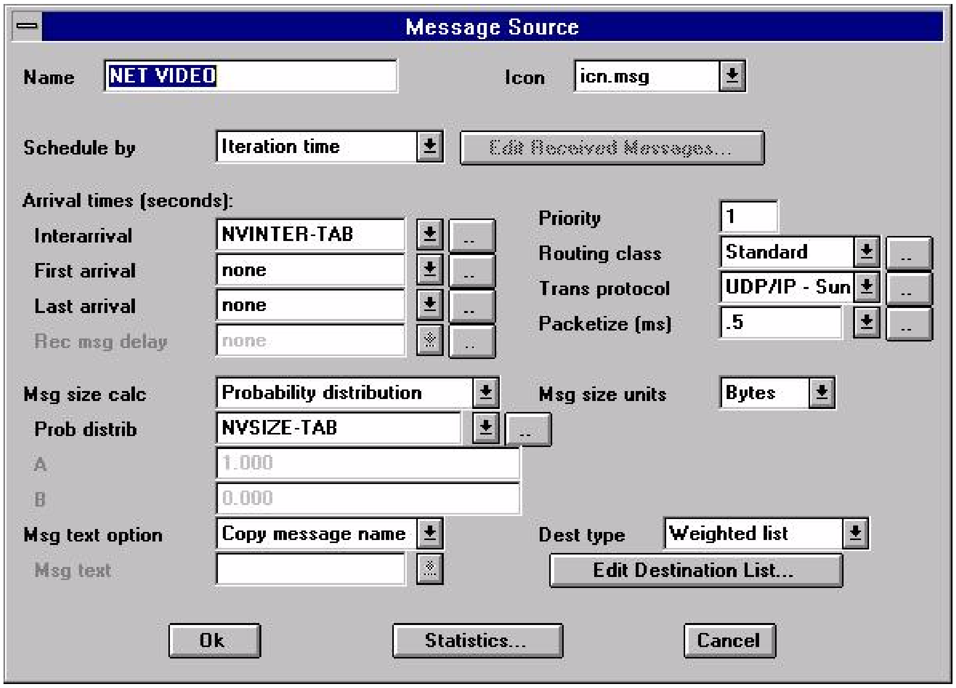


Figure 7.6 NetVideo Message Source

b) When all fields have been entered press the *OK* button at the bottom of the message source window.

c) Create a copy of the NET VIDEO message source.

d) Attach one copy of the message source to the node PERIWINKLE and the other to the node TEAL.

5. Creating the Multicast Router Message Source

The multicast router message source is used to model the broadcast of packets to all nodes on the network upon receipt of a message from the NetVideo message source.

a) Create a message source and connect it to the node CADET.

b) Edit the fields of the message source window to those shown in the following list. Figure 7.7 depicts the message source window upon completion.

- Name: MBONE ROUTING
- Schedule by: Set to received message and use the *Edit Received Messages* button so that the receipt of one message with the text NET VIDEO will trigger the message source.
- Message size calculation: (A x Msg bytes) + B
- A: 1.000

- B: 0.000
- Priority: 1
- Routing class: Standard
- Transport protocol: UDP/IP-Sun
- Packetize: 0.5 milliseconds
- Message text option: Copy message name
- Destination type: Set to multicast list and create a list containing all nodes except CADET, NAVY, FDDI ROUTER, and SPOT.

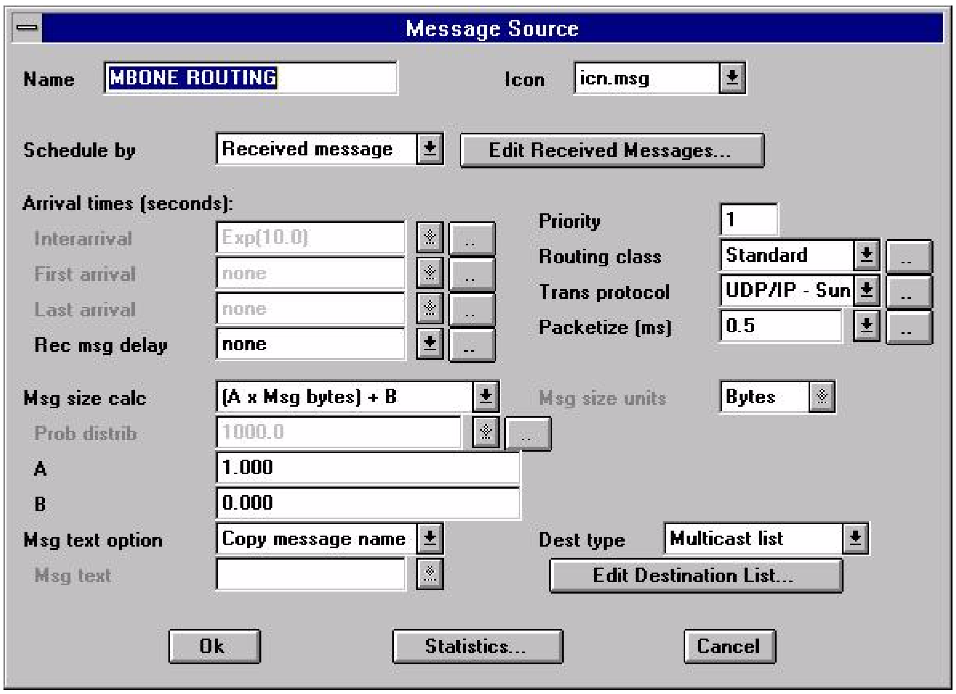


Figure 7.7 MBONE ROUTING Message Source

c) Save the model at this time using the name **1net63.c3**.

6. Defining the External Message Traffic

The external message traffic will be created using the COMNET Baseliner utility to model the network loading.

a) Select the *External Traffic/Messages* option from the *Define* menu. In the E*xternal Messages* dialog box which appears click on the *Use external file* check box, and then press *Import Traffic Files* button.

b) In the COMNET Baseliner *Import Files* dialog window click on the *Add* button. This opens up the *Import File Properties* dialog window. Next you will click on the ".." (double dot) button next to the *Import filename* field and define the path to the file **1net63.prn** file. Next you will need to define a custom traffic profile in the *Traffic profiles* field. To do this click on the ".." button to the right of the *Traffic profiles* field. This will open up the *External Traffic Profiles* dialog window where you will click the *Add* button which will open up the *Library Selections* dialog window where you will select the **Sniffer Profile** option by clicking on **Sniffer Profile**. This will open up the *Traffic Profiles* dialog window shown if Figure 7.8.

c) Edit the fields in the COMNET Baseliner historical file layout window to the following values. Figure 7.8 depicts the window upon completion.

- Source Start: 17
- Source Length: 20
- Destination Start: 41
- Destination Length: 20
- Message Start: 65
- Message Length: 4
- Time Start: 9
- Time Length: 7
- Size Start: 77
- Size Length: 4
- Hold/Call: 0
- Ignore lines containing: All fields empty
- Time units: Seconds
- Delta Time box: Checked
- Time in hours minutes seconds box: Not checked

d) When all values have been entered use the *OK* button to save the file using the name **1net63**. This file may be later reloaded if needed to describe the layout of the file **1net63.prn**. After completion of saving the layout file, press the *Done* button in the COMNET BASELINER III *External Traffic Profiles* dialog window.

e) In the *Import File Properties* dialog window click on the *Match Names* button down at the bottom of the window. This will open up a dialog window called *Name Match Profiles*. You will click the *Auto Match* button located on the right hand side of the dialog box. This will match up the external traffic events with model device names so that traffic can be generated appropriately within the model when a simulation is run. Traffic events that do not correlate to model names will be randomly assigned by the Auto Match feature. This is not a bad thing as each event will produce traffic within the mode, but for the purposes of this example you will need to explicitly match the 15 traffic events that were randomly assigned to model names. To do this you will select an external name or names from the *External Names* window, and then select a model name from the *Model Names* window and then click the *Set Match* button. This will update the matched name pair in the *External* Name, *Model Name* window. You will now explicitly match the external name 131.120.63.255 to the model name WHITE. Match all other mismatched external names to THE INTERNET. **[TIP: To do this more easily, hold down the CTRL key while selecting external names from the *External Name* window - see Figure 7.8a]**

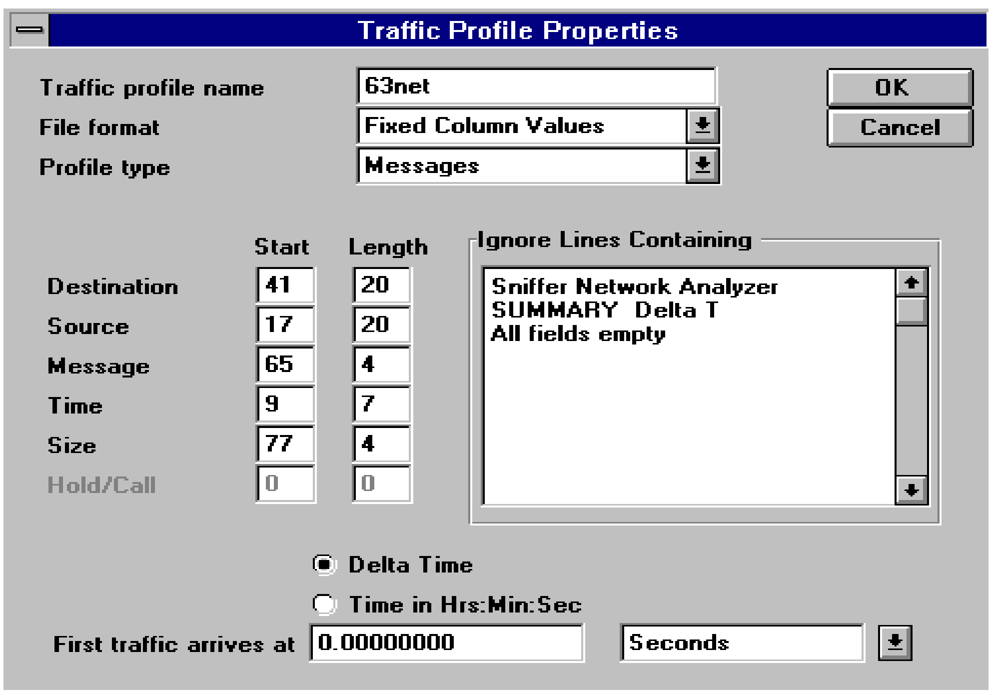


Figure 7.8 External Traffic Profile Properties for the File 1net63.prn

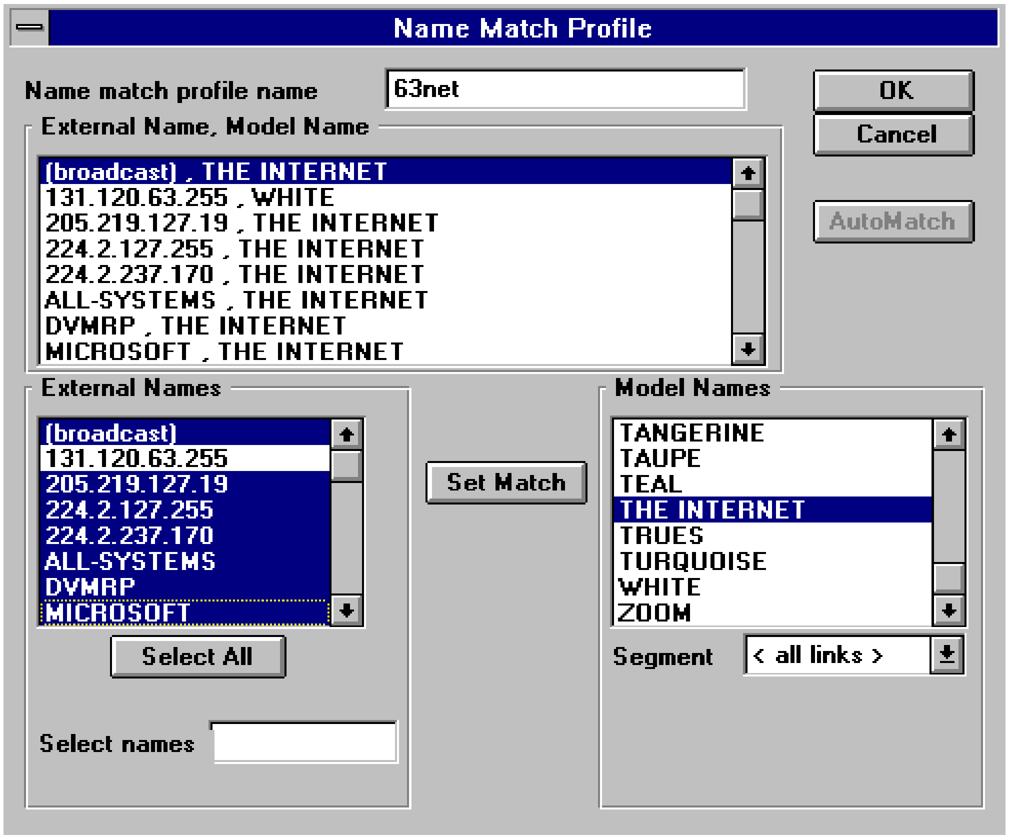


Figure 7 8a Explicitly matching external traffic events to model names

g) When all external file names are matched press the *OK* button and save the file using the name **1net63**. Then, press the *OK* button in the *import File Properties* dialog window. Finally click the *Import* button in the *Import Files* dialog window. The COMNET Baseliner III utility will then match all external events from the trace file to the model. Once this has been completed you can run the model and external traffic will be generated.

h) The first model is now complete. Verify and save the model **1net63.c3** again.

7. Creating the Second Model

The following steps describe the method of changing the first model produced to create the proposed changes to the network.

a) Use the *Save As* option from the *File* menu to save the file **1net63.c3** using the name **1net63a.c3**.

b) Check that the file **1net63a.c3** is the file currently being modified, clear the ports connecting the nodes CADET, CORNFLOWER, CIRRUS, PERIWINKLE, TEAL, and AZURE to the 63 NET link.
c) Make a copy of the 63 NET link, and rename the link SEG 63 NET. Attach the nodes CADET, CORNFLOWER, CIRRUS, PERIWINKLE, and TEAL to this new link.
d) Make a copy of the FDDI ROUTER node, and rename the node SEG ROUTER. Connect this node to the 63 NET, SEG 63 NET, and 64 NET links.
e) Edit the MBONE ROUTING message source destination list such that the multicast list contains only the following nodes in the list: TEAL, CORNFLOWER, CIRRUS, PERIWINKLE, and THE INTERNET.
f) When the file **1net63a.c3** was created, all parts of the file **1net63.c3** were included except the external message file **message.ext**. The external message file may either be recreated using the COMNET Baseliner III utility, as before, or copied from the directory 1net63 into the directory 1net63a.
g) The second model is now completed. Verify and save the model.

F. SIMULATION AND RESULTS

One simulation for each model will be run to gather results. For both models the simulation run time should be set to 30 seconds. This value is used as the external file contained 110 seconds of captured data, and the scale factor of 3.5 used in both models will reduce the time to approximately 30 seconds. Prior to running each simulation select the following reports to be generated at the completion of the simulation for each model.

- Link Reports: Channel Utilization for the 63 NET and SEG 63 NET links.
- Nodes: Received Message Counts for all nodes.
- Message and Response Sources: Message Delays for all nodes.

The simulation run using the current network configuration showed an overall network utilization of 17 percent for the ethernet network, with an average transmission delay of 1.6 milliseconds. Figure 7.9 depicts the network utilization over the simulation time period.

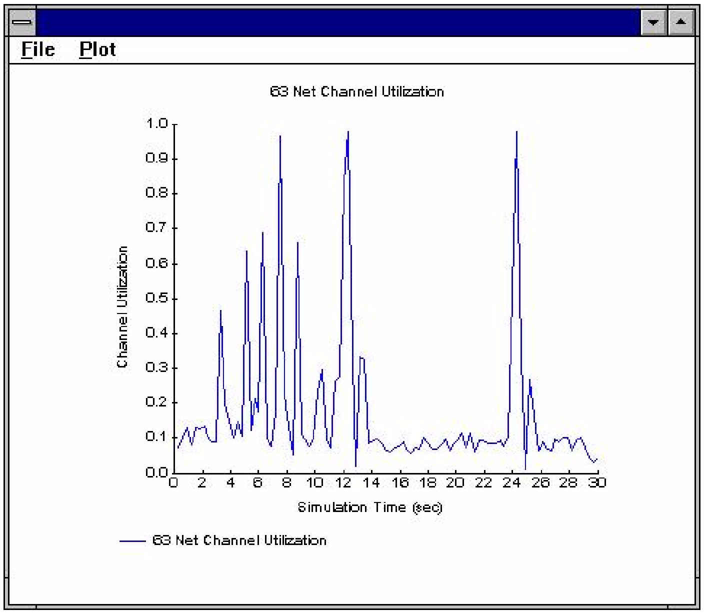


Figure 7.9 Current Network Configuration Utilization

The simulation ran for the model of the proposed changes showed a network utilization of 2 percent for the original ethernet network with a delay of 0.9 milliseconds. The proposed new network showed a utilization of 14 percent with the same delay. Figure 7.10 and Figure 7.11 depict the network utilization for the original network and the new network under the proposed changes respectively.

The method proposed to alter the System Technology Lab's ethernet network does provide one way of isolating the existing network from the volume of traffic generated by video teleconference packets created by the workstations and the multicast router. This is just one proposed plan of action. Other methods, such as moving the desktop video conferencing capable computers to the higher bandwidth FDDI network or a different method of segmenting the ethernet network, would need to be analyzed prior to making a change in the current network configuration.

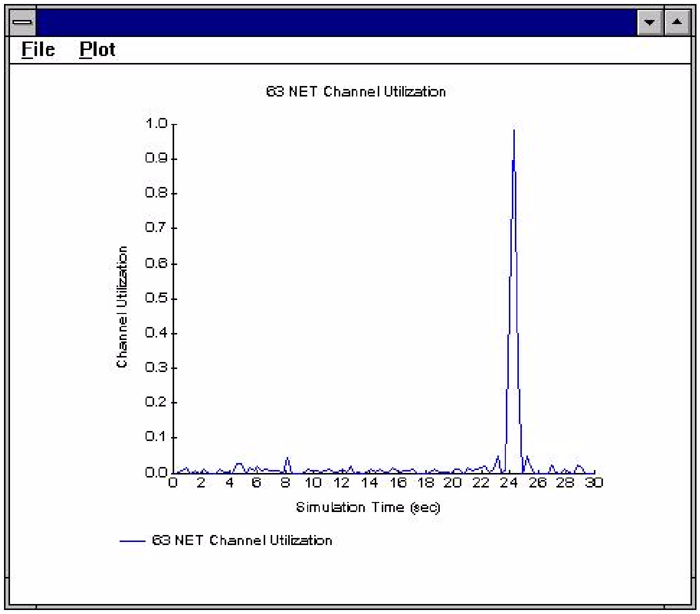


Figure 7.10 Original Ethernet Network Utilization With Proposed Network Changes

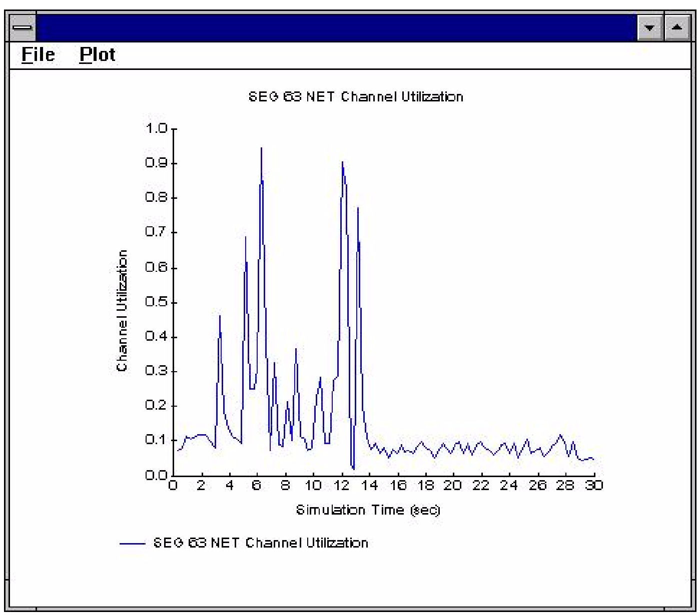


Figure 7.11 Segmented Ethernet Network Utilization Under Proposed Network Changes

VIII. MODELING WIDE AREA NETWORKS

A. INTRODUCTION

The goal of this chapter is to introduce the objects available within COMNET III that may be used to model wide area networks (WANs). Specifically, the ATM switch node, subnetworks, the wide area network (WAN) cloud, and the routing protocols available will be covered. A model of a frame relay WAN used as a portion of the multicast backbone will be modeled using the WAN cloud object and by building the WAN piece-by-piece using switches and point-to-point links.

B. ATM SWITCH NODES

The ATM switch node is aimed at modeling switches, hubs, or routers which have a high bandwidth internal switching structure resulting in no contention in switching packets across the switch's bus [Ref. 2 p. 118]. The ATM node is described in COMNET III only by its port buffering characteristics. When a packet enters the buffer of this node, it incurs a port processing delay in the same manner as for a computer and communications node or a router node. The frame is then switched to an output buffer instantaneously with parallel switches between different ports allowed. The difference between an ATM node and other nodes available in COMNET III is its non-blocking characteristic. When a packet is switched from an input to an output buffer, the checks that the output buffer and total buffering capacity of the ATM switch node are made in the same manner as with a router or computer and communications node. However, if this test fails, the ATM node will not block the packet. It instead stores the packet until room is available in the output buffer the packet needs to be routed.

The ATM node may be created from either the toolbar or from the *Create* menu. As many links as desired and all types of links may be connected to the node. The ATM node may only act as a switching point for traffic and may not be used as the destination or origin of traffic. Editing of the node will cause the window shown in Figure 8.1 to appear. The fields used to describe the operating characteristics of the node have the same meaning as for a computer and communications or router node.

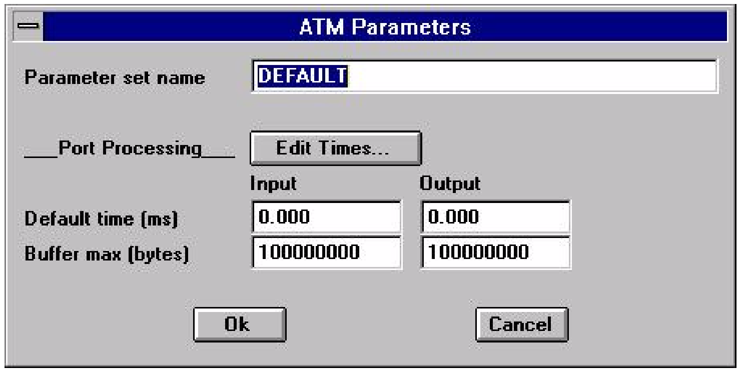


Figure 8.1 ATM Parameters Window

C. SUBNETWORKS

The subnetwork provides two functions within COMNET III. The first is a method of managing the display of a large network. The second is providing a method to allow different routing protocols to be used in different portions of the network model.

A subnetwork is created by choosing the subnetwork icon from the toolbar. Maneuvering about a subnetwork is done in a different manner than other objects in COMNET III. To edit the name or routing protocol used in the subnetwork, the subnetwork icon must first be selected with a single click of the mouse button. Then, choosing the *Properties* option from the *Edit* menu will cause the window shown in Figure 8.2 to appear. This window allows specifying the name of the subnetwork and the routing protocol which will be used in the subnetwork. The traffic scale field provides a method of scaling the traffic generated by all nodes built within the subnetwork without having to individually edit each application command or traffic source. The amount of time between routing table updates for the nodes in the subnetwork may also be specified. Double clicking on the subnetwork icon or selecting the icon and then choosing the *Enter* option from the *View* menu will cause a clear work area to appear. The portion of the network which is to be modeled at the subnetwork level is in this area. To return to the higher level, which is called the backbone view, either double click in the work area or select the *Leave* option from the *View* menu.

A subnetwork is connected to the backbone view by means of an access point which is created using the toolbar. A minimum of one access point is required within the subnetwork. Each access point created may be connected to only one node in the subnetwork. The access point acts as an extension of the node it is connected to and allows a link in the backbone view to be connected to the subnetwork. After an access point is created and after leaving the subnetwork view, the subnetwork icon in the backbone view will have a dot on the edge of the icon for each access point created. This is the point where links in the backbone view are connected to the subnetwork.

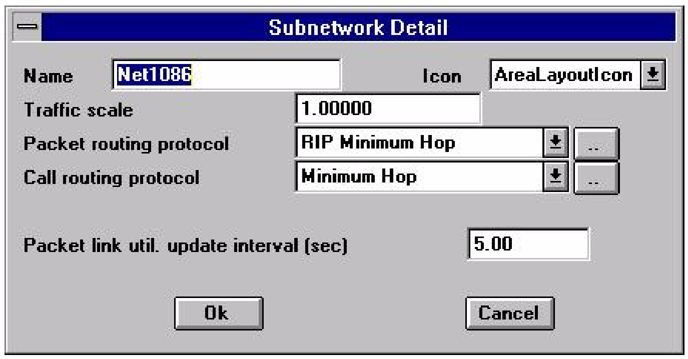


Figure 8.2 Subnetwork Detail Window

D. WIDE AREA NETWORK (WAN) CLOUDS

The WAN cloud provides an abstract method of modeling wide area networks which provides an alternative to modeling a WAN explicitly using router nodes, ATM switch nodes, and point-to-point links. The WAN cloud behaves similarly to the link objects in that it delivers frames and models a delay of these frames across the network. The cloud's internal structure is defined using access links and virtual circuits. The access links provide the point-of-presence to the WAN cloud. The virtual circuits are optional within the cloud, but may be used to specify the grade of service that frames sent across the cloud will receive.

A WAN cloud may be created by either using the toolbar or by choosing the *Cloud* option from the *Create* menu. Manipulation of the WAN cloud is similar to manipulating a subnetwork. To edit the detail of a cloud, the cloud icon must first be selected using the mouse, and the *Detail* option then chosen from the *Edit* menu. The window shown in Figure 8.3 will appear when this is done. Within this window, the parameters field allows selecting the parameter set which governs the framing characteristics and discard eligibility of frames traversing the network. The interval for mean statistics field specifies in seconds the interval over which statistics on the burst size is collected when running a simulation. This interval may be changed while running a simulation to adjust the sampling rate. The lower half of Figure 8.3 contains fields which become activated when the enhanced model option is chosen for the parameter set selected for the cloud. These fields may be used to define when changes in the congestion level of the cloud will occur when running a simulation. The three congestion states which may be modeled are normal, moderate, and extreme. The congestion state sets the delay of frames across the cloud and the probability that a frame will be dropped within the cloud. The simulation time that a state change is to occur may be specified along with the time the change will take to occur and the time to recover from this change. State changes from either moderate or extreme congestion will always be back to a state of normal congestion. The probability of extreme congestion may also be specified. A random number is pulled from a uniform (0,1) distribution and compared to the extreme congestion probability to determine what the next state will be. In essence, these parameters allow modeling delays on a network due to additional traffic without having to model the traffic itself.

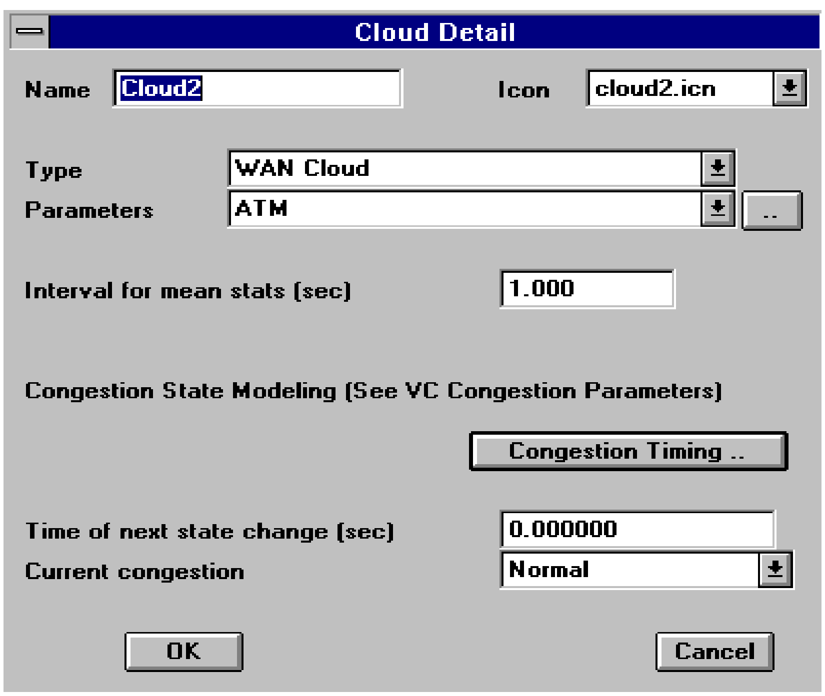


Figure 8.3 Cloud Detail Window

To define the internal structure of the WAN cloud either double click on the WAN cloud icon or select the icon and then choose the *Enter* option from the *Edit* menu. A clear work area will appear which allows creating the access links and virtual circuits within the cloud. To return back to the backbone view, either double click on the background in the work area or select the *Leave* option from the *View* menu.

1. WAN Cloud Parameter Sets

A WAN cloud parameter set defines the framing characteristics which are used within the cloud, the probability of frames being dropped within the cloud, and the delays associated with traversing simulated switches in the cloud. The parameter set may be defined by pressing the ".." button next to the parameter field in the cloud detail window. This will cause the window shown in Figure 8.4 to appear. The fields of this window govern the operating characteristics of the cloud and are used for the following purposes:

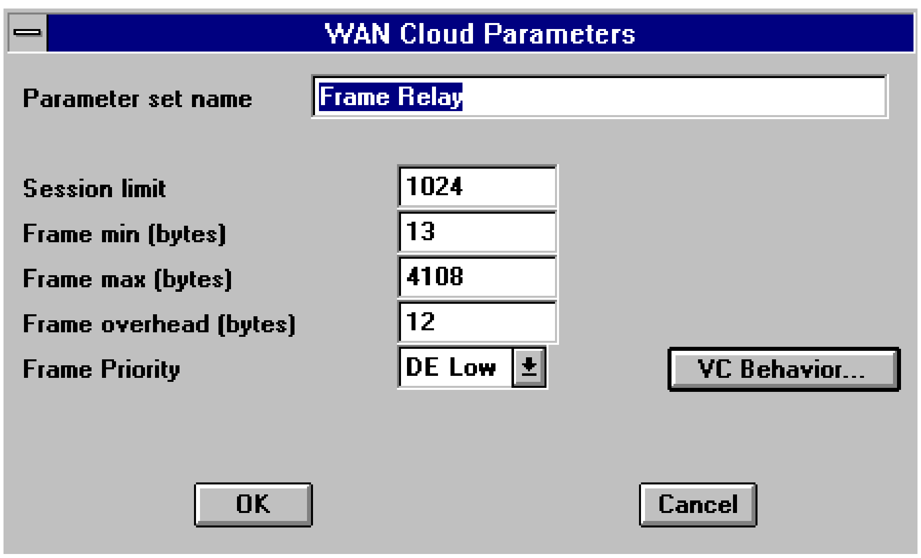


Figure 8.4 WAN Cloud Parameters Window

- Session limit: Specifies the number of sessions which may be concurrently routed through the cloud at a single time.

- Frame Min, Max, and Overhead: These fields specify the maximum and minimum payload capability and the associated overhead caused by the encapsulation of packets which are to be sent across the cloud. The maximum and minimum frame sizes are inclusive of the frame overhead.
- Frame priority: Specifies the manner that frames will be treated when arriving at the access link buffer. The frame priority may be set to none which treats all frames as equals, equal to the original frame priority, or based on the discard eligibility of the frame.

The behavior of the Virtual Circuits within the cloud may be further defined by clicking on the *VC Behavior* button. This will cause the window shown in Figure 8.4a to appear. The fields of this window govern the operating characteristics of the cloud and are used for the following purposes

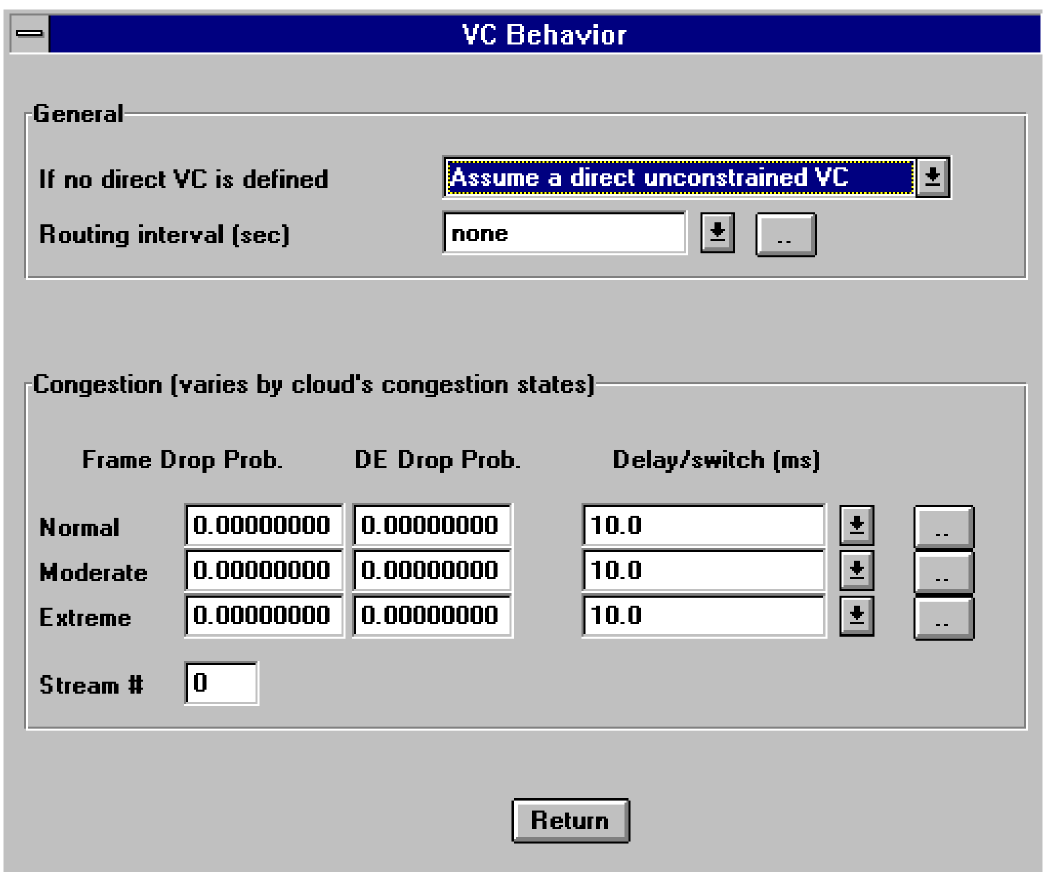


Figure 8.4a Behavior of WAN Cloud Virtual Circuits

- If no direct VC is defined: Assume a direct unconstrained VC is the default mode for the behavior of traffic inside a cloud. The other option is to Multiply transit cloud along VCs.
- Routing Interval: Used with the enhanced modeling option to specify a random interval between the updating of the number of switches present in a virtual circuit. This feature is used to model variation as a result of varying lengths across different paths within a network.
- Delay/switch: Used to specify the delay a frame will incur as it traverses switches in a virtual circuit. The delay may be modeled as a constant or by using a statistical distribution. Different values may be specified for the three congestion levels when enhanced modeling is used.
- Frame drop probability: Specifies the probability of a frame being dropped by the network based on the congestion state of the cloud and the discard eligibility (DE) of the frame.

2. Cloud Access Links

The cloud access link is used to model the access line from a local area network to the wide area network's point of presence. Access links are created within the WAN cloud by either using the point-to-point link icon on the toolbar or by selecting the *Cloud/Access Link* option from the *Create* menu. Each access link created must be attached to an access point. The access point is an extension of the access link which allows connecting the access link to a node in the backbone view.

Editing an access link will cause the window shown in Figure 8.5 to appear. The fields of this window set the high level operating characteristics of the access link and are used for the following purposes:

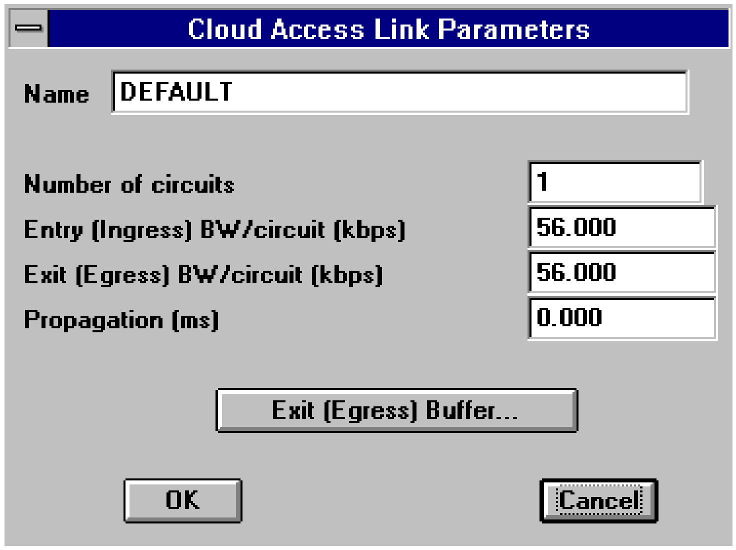


Figure 8.5 Cloud Access Link Parameters Window

- Number of circuits: Specifies the total number of circuits modeled by the access link.
- Entry and Exit bandwidth / circuit: Specifies the transmission rate to and from a node and the WAN cloud.
- Propagation: Used to model the delay from the local area network to the wide area network's point of presence.

The following fields as defined by clicking on the *Exit (Egress) Buffer*.. of the Cloud Access Link shown in Figure 8.5a. The *Exit (Egress) Buffer* controls the buffering that is available at the input side of the exit part of the access link. A particular access link may be connected to several virtual circuits that deliver frames at the same time, in which case the frames may have to queue to wait for the earlier frames to get transmitted. This buffer is sorted by priority of the frames (the method for determining frame priority is set in the cloud parameter set.

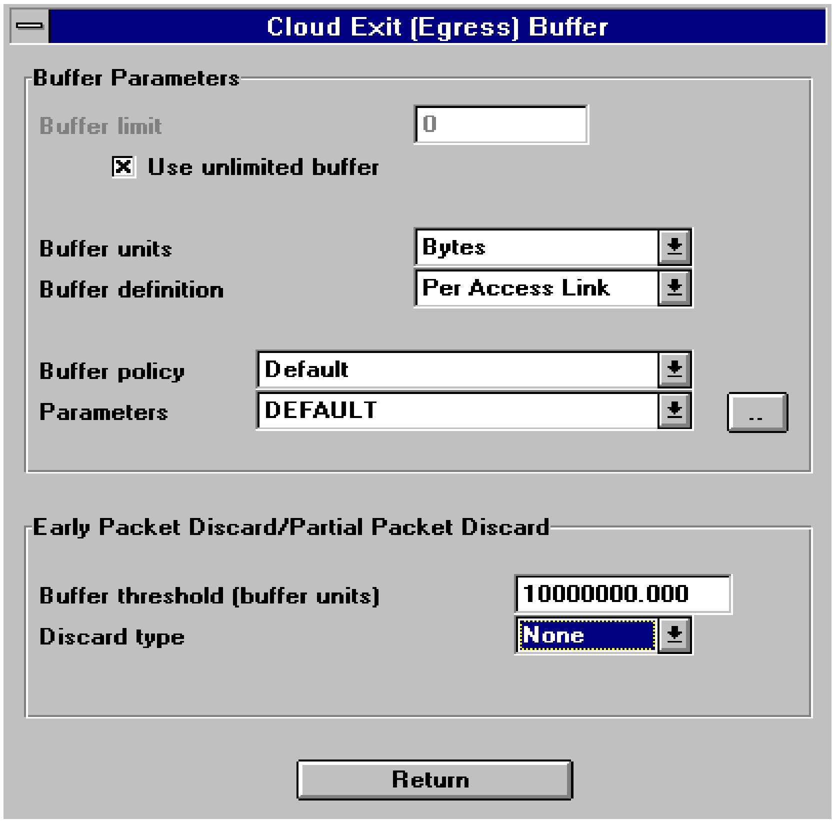


Figure 8.5a Cloud Exit (Egress) Buffer Parameters

- Buffer limit: If this value is exceeded then the buffer will block incoming frames.
- Use unlimited buffer: If this box is checked there is no buffer limit and the buffer will always accept incoming frames.
- Buffer units: The buffer size can be specified in either packets or bytes.
- Buffer definition: The buffer may be separate for each VC (Per VC) or just a single buffer for the whole access link (Per Access Link).
- Buffer policy: There are three choices for controlling how the *Exit (Egress) Buffer* admits packets into its buffer - Default, Threshold, & Preemption. For a more detailed description please review the section on Buffer Processing in the COMNET III User’s Manual.
- Buffer threshold (buffer units): Specifies at what level of buffer congestion the logic for Early Packet Discard and Partial Packet Discard will become active.

- Discard type: Early packet discard and partial packet discard are options for rejecting frames once a previous frame from the same packet is lost.

3. Virtual Circuits

Virtual circuits are optional within the WAN cloud as long as the "Transmit Non-VCs" box is checked for the parameter set chosen for the cloud. The main purpose of the virtual circuit is to specify the traffic burst constraints which allows evaluating grade of service for the wide area network. Burst constraints are set using the following parameters:

- Committed Information Rate (CIR): The maximum sustained subscriber throughput rate the network commits to supporting per virtual circuit. [Ref. 8]. CIR is expressed in units of kilobits per second.
- Continuous Burst Size (Bc): The maximum number of bits which are committed or guaranteed to be transmitted over a burst interval if received contiguously by the switch [Ref. 8].
- Excess Burst Size (Be): An additional number of bits above the Bc which will be allowed to be transmitted if the bandwidth is available on the network [Ref. 9].
- Burst Interval (Tc): The time interval over which the virtual circuit is monitored for grade of service. The time interval is normally calculated as Bc/CIR.

The burst constraints specified for a virtual circuit within COMNET III will cause one of three things to occur to a frame entering the network when running a simulation. If the frame results in exceeding the committed plus excess burst size, the frame is automatically dropped. If the frame results in a burst size only greater than the committed burst size, the frame is marked as discard eligible (DE). If less than the committed burst size, the frame is marked as normal.

Virtual circuits may be created only within the WAN cloud and are used to connect two access links within the cloud. They provide data flow in only one direction.

Thus, for two way data flow, two virtual circuits connected in opposite directions would need to be modeled. Virtual circuits are created by either using the toolbar, or by choosing the *Cloud VC* option from the *Create* menu. Editing of the virtual circuit will cause the window shown in Figure 8.6 to appear.

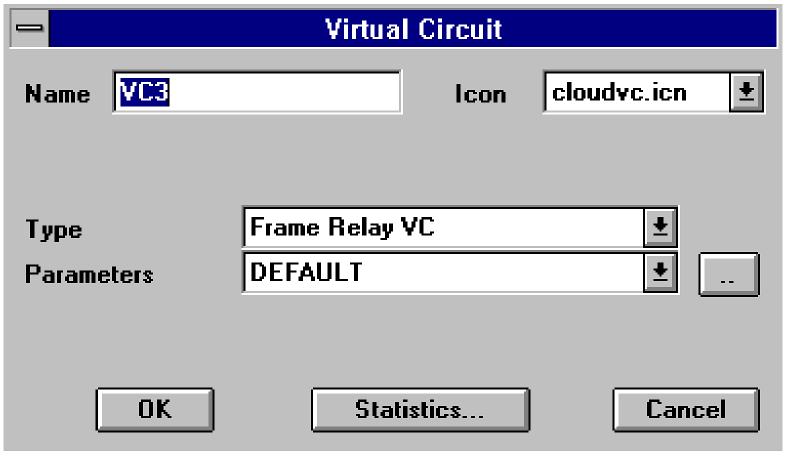
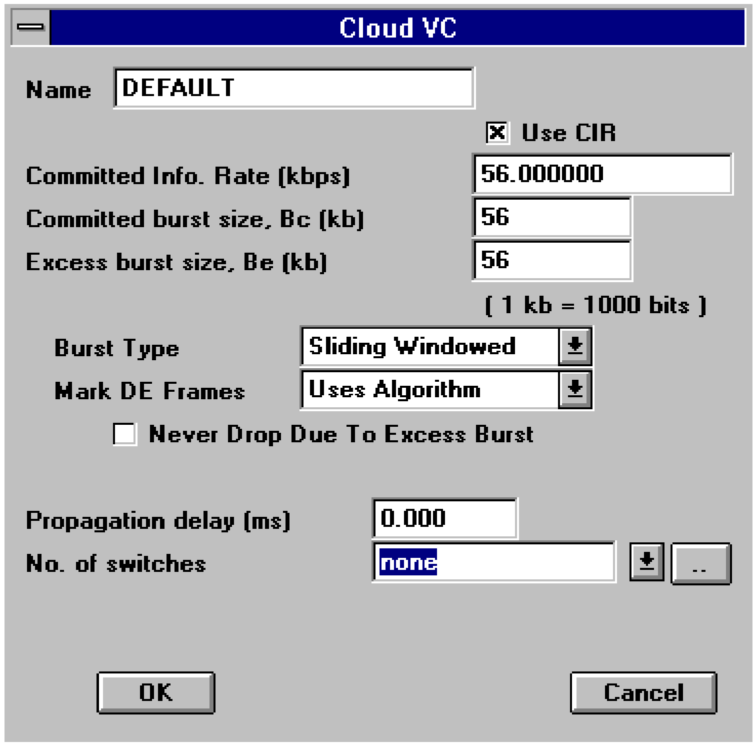


Figure 8.6 Cloud Virtual Circuit Window

To edit the Virtual Circuit parameter set shown in Figure 8.6a click on the ".." (double dot) button to the right of the Parameters field.



The fields associated with the Cloud VC window are used for the following purposes:

- Committed Information Rate (CIR): Specifies the maximum sustained throughput of the virtual circuit in kilobits per second. This value, along with the committed burst size value, provides the default method for calculating the burst interval. If enhanced modeling is used and the "Use CIR in VCs" box is not checked in the WAN cloud parameters window, this field changes to allow explicit setting of a burst interval.
- Committed burst size: Specifies the threshold on burst size above which frames will be marked as discard eligible.
- Excess burst size: Specifies the threshold above the committed burst size which will cause frames to be dropped immediately if exceeded.
- Burst type: Sets the type of burst measurement used: leaky bucket, jumping window, or sliding window.
- Mark DE Frames: This option allows you to bypass the traffic policing when it is know that a certain type of traffic will never (or always) be set to discard

eligible. The options for bypassing the algorithm for setting discard eligibility of packets are to always set the DE flag or to never set the flag.

- Never Drop due to Excess Burst: This is a checkbox for preventing the algorithm from rejecting any excess traffic that occurs after the burst exceeds the committed plus excess burst size.
- Propagation delay: Used to set a delay which models the time required for a frame to traverse the distance between access links.
- Number of switches: Used to set the number of switches the path of the virtual circuit may contain when enhanced modeling is used. Otherwise, the number of switches is assumed to be one.
- Show burst size: Checking this box will cause the burst size of the virtual circuit to be displayed in the work area when running a simulation. This value is updated based on the update interval specified in the cloud detail window.

E. PACKET ROUTING PROTOCOLS

As packets traverse the networks defined in the backbone and subnetwork views created, routing decisions must be made for a packet to reach its final destination. The routing protocols which are built into COMNET III include:

- IGRP
- Link-State Shortest-Path First
- RIP Minimum Hop
- Minimum Penalty
- Shortest Delay
- User-Defined Routing Tables

Different routing protocols may be defined for the backbone network and for each subnetwork. Routing protocols are set for the backbone network by choosing the *Backbone Properties* option from the *Define* menu. This will bring up the window shown in Figure 8.7 which allows setting the packet routing protocol to be used at the backbone level and the interval for updating routing tables when running a simulation. Subnetwork routing protocols and update intervals are set in the subnetwork detail window as show in Figure 8.2.

When a simulation is first started, each node in the model calculates all possible routes to other nodes within its subnet. A routing table is then created which lists the hops

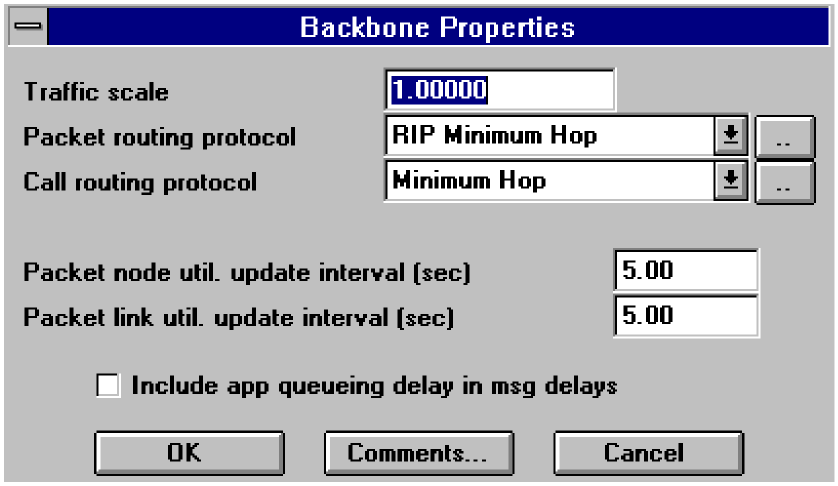


Figure 8.7 Backbone Detail Window

required along a given path to reach a destination. The hops are weighted based on the routing protocol chosen. When a packet requires routing, the hops in the table are evaluated in the order in which they appear in the table using the following logic:

- The hop limit for the routing class must not be exceeded.
- Both the node and the link associated with a hop must be up.
- The hop must not return to a previously visited node.
- If a session setup packet is being routed, the link and next node must be under there session limit.

If the tests for a hop are passed, the packet is attempted to be routed along that path. Similar decisions occur at each node along the path until the packet reaches its destination. If the first hop did not pass the test, the other hops are evaluated in sequential order until a hop is found. If all hops fail, the packet is blocked.

Several of the routing protocols allow specifying a deviation percent which may be used to balance traffic load among several links. The deviation percentage sets similar hops whose weighting factor falls within the deviation percent to the same weighting factor. These hops are then alternated in a round robin order to balance the loads between the links.

1. IGRP

The IGRP method calculates weighting factors in the routing tables based on a link's bandwidth, utilization, and delay. The weighting factors are calculated using the following equation:

K1 * bandwidth factor + K2 * bandwidth factor/(256 - load) + K3 * delay factor

- K1, K2, and K3 are specified by the routing class of a packet. The default values are 1, 0, and 1, respectively.
- Bandwidth factor = 10^{10}/bandwidth
- Load = utilization percent * 255
- Delay factor = link delay in units of 10 microseconds

2. Link-State Shortest-Path First

The Link-State Shortest-Path First protocol calculation of hop weighting factors is based on penalty tables applied to the links in the model. Only the penalty values of the first line of the penalty table are used. The weighting factor of each hop remains constant throughout the entire simulation. The deviation percent is used with this routing protocol based on the penalty calculated for a hop.

a. Penalty Tables

The penalty table allows a method of applying penalties to links. The penalty values used may represent the distance covered by the link, cost of using the link, or any other metric needed to be modeled. The penalties applied to a link may be based on either the routing class of the packet, the link delay, or the link utilization. The penalty tables created are applied to a link by specifying the penalty table to be used in the port which is attached to the link.

Penalty tables are created by choosing the *Routing Penalties* option from the *Define* menu. This will cause the window shown in Figure 8.8 to appear. By pressing the *Add* button in this window, the window shown in Figure 8.9 will appear which allows creating the penalty table. The threshold field allows entering the values based on link utilization or delays. Penalties based on the routing class may be set for each threshold value defined. A default column is always present which applies a penalty to each routing class listed unless a different penalty value is assigned.

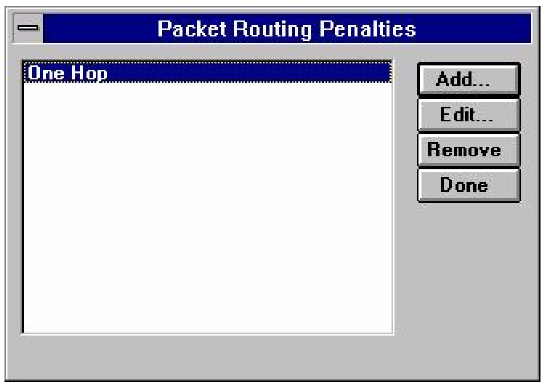


Figure 8.8 Packet Routing Penalties Window

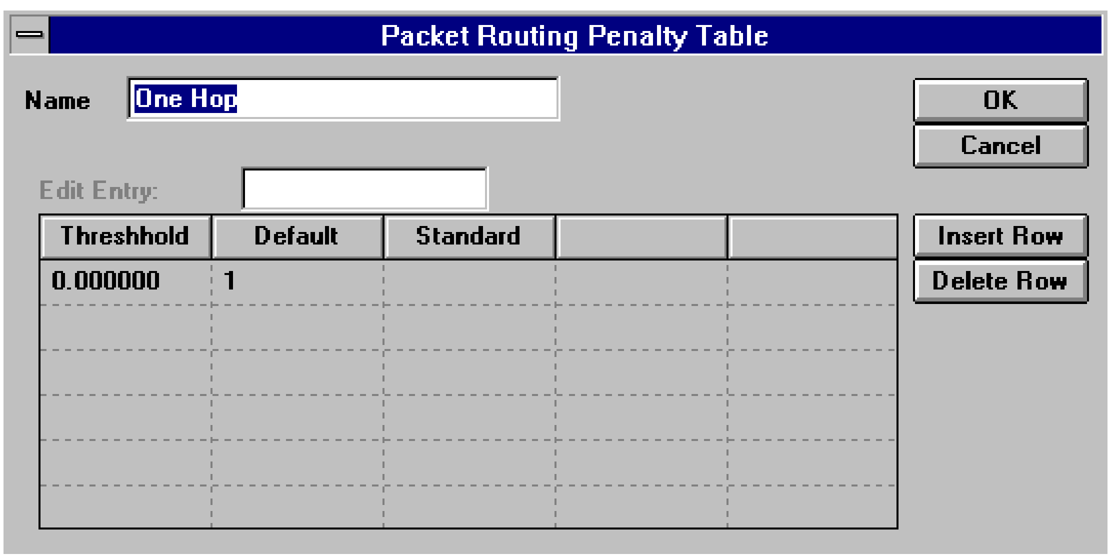


Figure 8.9 Packet Routing Penalty Table Window

3. RIP Minimum Hop

RIP Minimum Hop calculates weighting factors based on the number of hops required along the path of a packet from its origin to its destination. The first route available with the lowest hop count is used. The deviation percent is based on the total hop count of a route.

4. Minimum Penalty

Minimum penalty routing calculates weighting factors based on penalty tables applied to links. It is similar to Open-State Shortest-Path First routing except all rows defined in a penalty table are used in determining weighting factors for a hop. The hop weighting factors are calculated at the beginning of a simulation and are updated based on the interval set in the subnetwork or backbone detail windows. Deviation percent is based on the penalty metric used for the route.

5. Shortest Delay

Shortest Delay routing calculates weighting factors based on the packet delays experienced on a link. Routing tables are updated to account for congestion changes on a link based on the interval specified in the subnetwork or backbone detail windows. Deviation percent is based on the total link delay of all hops along a route.

6. User-Defined Routing Tables

When user-defined routing is used, routes are selected based on routing tables created at nodes in the model. To create the routing tables, the user-defined routing option must first be selected in the backbone or subnetwork detail window. This activates the *Packet Routing Table* button in the node detail window for ATM switch, router, and computer and communication nodes. When this button is pressed, the window shown in Figure 8.10 will appear. By highlighting a destination in this window, routes to that destination may be specified after pressing the *Edit Selected* button which will cause the window shown in Figure 8.11 to appear. The left half of this window is used to define various routes, specify the routes as either primary or secondary routes, and to specify the order in which these routes are placed in the table which affects the order in which routes are looked at when running a simulation. The right half of the window allows building the routes which are desired to be specified. When more than one route is set to a destination,

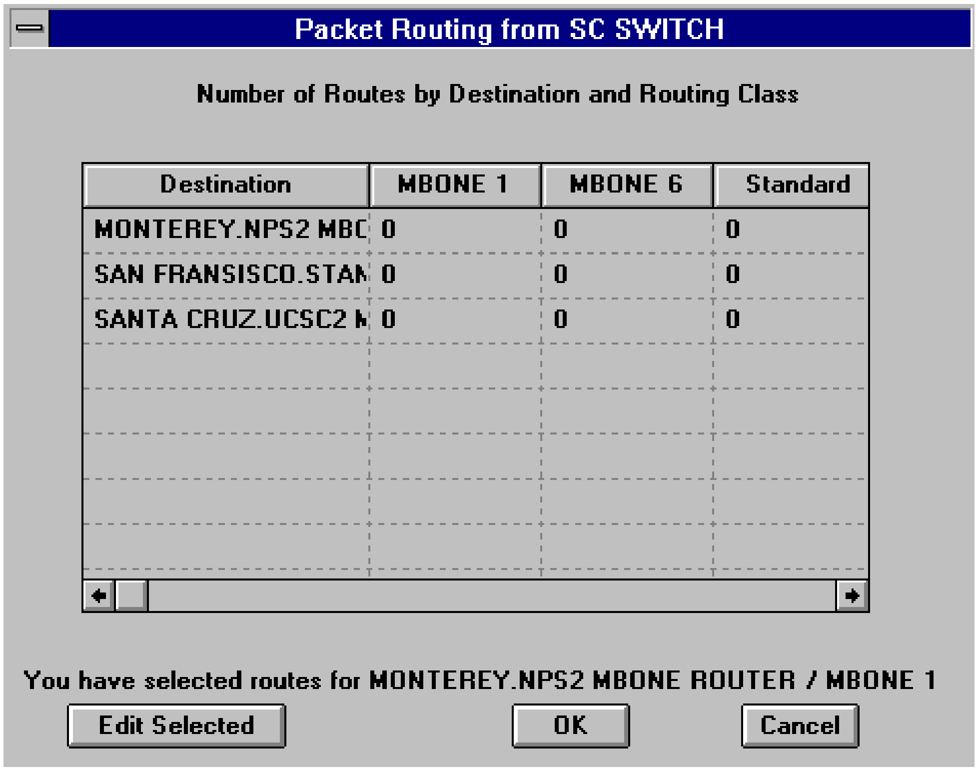


Figure 8.10 Packet Routing Table Window

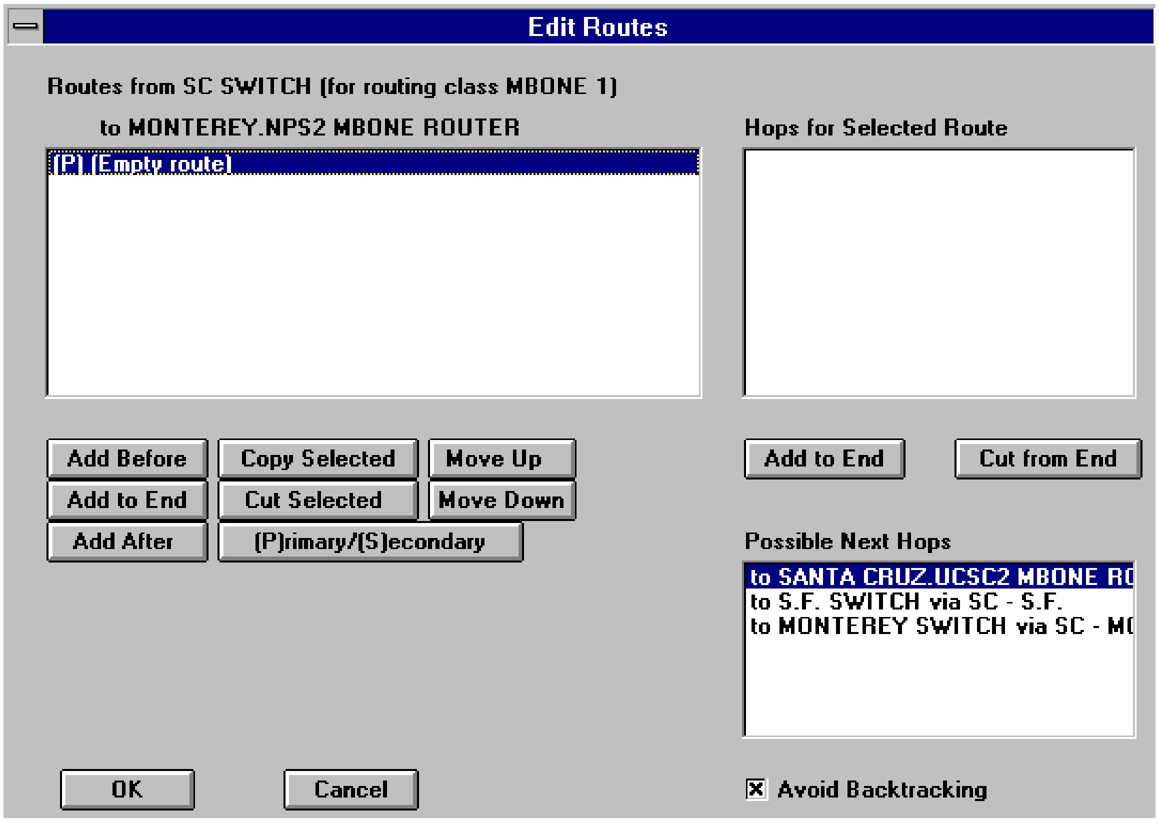


Figure 8.11 Edit Routes Window

selection criteria is required to determine which route will be used. The selection criteria may be based on the following options:

- First Available Route
- Maximum Unused Bandwidth
- Minimum Delay
- Minimum Queue
- Minimum Sessions
- Random List
- Round Robin

F. PROBLEM STATEMENT AND ANALYSIS

Desktop video conferencing using the multicast backbone (MBONE) is becoming highly popular especially in university environments as the availability of workstations with adequate processing power and built-in audio capability are becoming more common. Frame relay is also becoming a common method of packet switching for public and private wide area networks, but has not been tested for use as part of the multicast backbone. In order to determine if frame relay would be a suitable method of transport between local area networks, a what-if scenario is used to determine the probable delays in transmission of multicast video packets across a frame relay network. Discussion with Professor Donald P. Brutzman of the Naval Post Graduate School who has done extensive research in the multicast backbone indicated that delays in area of 1 second are usually tolerable when receiving multicast video.

The multicast packets generated from desktop video conference sessions on a local area network are first sent to network's multicast router. The multicast router then retransmits the packets to all nodes in the network and to other multicast routers on separate local area networks. The connections between different multicast routers are defined by specifying virtual tunnels within the wide area network. A major concern with the transmission of multicast packets is the possible flooding of a network with multicast packets. To control this problem, the multicast backbone utilities available require setting a time-to-live for packets generated by each session. A time-to-live of 16 would limit packets to a campus-sized area, while a time-to-live of 127 to 255 could send the packets worldwide [Ref. 9].

To model the possible use of a frame relay wide area network, two separate models will be built. One model will use a WAN cloud to model the wide area network, and the other will use ATM switch nodes and point-to-point links to build the network. To model the generation of video packets, the data used and discussed in Chapter VII will be used. Three separate local area networks connected by the wide area network will be modeled. Each local area network will consist of a two nodes generating video packets and a multicast router attached to a FDDI network. Two message sources will be used to model the routing required by the multicast router. One message source will model the routing of packets received from the wide area network. The other message source will model the routing of video packets generated within the local area network.

A problem with modeling multicast video packets in COMNET III is that the application does not have a method of specifying a time-to-live of a packet. A method to work-around this problem is to use routing classes to specify a hop limit for a routing class which will minimize the total distance a packet may travel.

The message generator used to create video packets is based on the data collected from a workstation set to transmit video packets at a rate of 128 kbps. As two nodes each will be generating packets within a local area network, the tunnel from the network to the wide area network and the links in the wide area network will be modeled as having a transmission rate of 256 kbps. The tunnel from the wide area network to the local area network will be modeled as having a transmission rate of 512 kbps.

G. BUILDING THE NETWORK MODELS

Three separate models will be built to model the transmission of video packets across a frame relay wide area network. The first model will be used to create the local area networks and message sources. This model will then be used in the two other models where the wide area network will be added. One of the models will use the WAN cloud to model the frame relay network. In the other model, the wide area network will be built using ATM switch nodes and point-to-point links.

1. Defining Computer and Communication Node Parameter Sets

The computer and communication parameter set defined will be used for all computer and communications nodes created in the model.

a) Select the *Node Parameters/Computer and Communications* option from the *Define* menu. In the computer and communications node parameters window which appears press the *Add* button. Edit the following fields in the node parameter window which appears:
   - Name: WORKSTATION
   - Time slice: Set to a uniform distribution with a minimum of 10000 and a maximum of 1000000. The stream value may be set to any integer from 0 to 99.
   - Port processing default times: Set to 0.005 milliseconds for both input and output.

2. Creating the Local Area Network Topology

The local area network topology will be created within a subnetwork. Each of the three local area networks which will be modeled in exactly the same manner.

a) Create a subnetwork using the toolbar. Change the name of the subnetwork SANTA CRUZ.

b) Enter the subnetwork and create a token passing link. Edit the parameters of the link to the following:

- Name: UCSC FDDI
- Type: Token Passing
- Parameters: FDDI

c) Create three computer and communication nodes in the work area and edit the fields in the node detail window for each to those values shown in Table 8.1. Attach each node to the UCSC FDDI link.

d) Create an access point using the toolbar and attach the access point to the node UCSC2 MBONE ROUTER.

Name	Type	Parameter
UCSC1	C&C Node	WORKSTATION
UCSC2 MBONE ROUTER	C&C Node	WORKSTATION
UCSC3	C&C Node	WORKSTATION

Table 8.1 Local Area Network Computer and Communication Node Parameters

3. Defining the NetVideo Message Source Tabular Distributions

The tabular distributions created will be used to describe the message size and interarrival times of the NetVideo message source.

a) From the *Define* menu select the *Tables* option.

b) Press the *Add* button in the tabular distributions window, and edit the fields of the table detail window to the following:

- Name: NVSIZE-TAB
- Table type: discrete
- Probabilities and Values: Enter the cumulative probabilities and packet size from Table 8.2 into the probability and values fields respectively.

Packet Size (bytes)	Probability	Cumulative Probability
20	0.03	0.03
50	0.01	0.04

340	0.07	0.11
1050	0.13	0.24
1060	0.17	0.41
1070	0.13	0.54
1080	0.11	0.65
1090	0.09	0.74
1100	0.08	0.82
1110	0.06	0.88
1120	0.04	0.92
1130	0.03	0.95
1140	0.02	0.97
1150	0.01	0.98
1160	0.01	0.99
1170	0.01	1.00

Table 8.2 Packet Size and Probabilities

c) When all fields are entered, press the *OK* button at the bottom table detail window. Then, press the *Add* button in the tabular distributions window to create the second distribution.

d) Edit the fields of the table detail window to the following:

- Name: NVINTER-TAB
- Table type: discrete
- Probabilities and Values: Enter the cumulative probabilities and packet size from Table 8.3 into the probability and values fields respectively.

e) When all fields are entered, press the *OK* button at the bottom table detail window. Then, press the *Done* button in the tabular distributions window to create the second distribution.

Packet Interarrival Time (seconds)	Probability	Cumulative Probability
0.01	0.022	0.022
0.02	0.015	0.037
0.03	0.018	0.055
0.04	0.076	0.131
0.05	0.052	0.183
0.06	0.107	0.290
0.07	0.294	0.584
0.08	0.172	0.756
0.09	0.022	0.778
0.10	0.008	0.786
0.11	0.012	0.798
0.12	0.008	0.806

0.13	0.046	0.852
0.14	0.069	0.921
0.15	0.025	0.946
0.16	0.005	0.951
0.17	0.002	0.953
0.18	0.004	0.957
0.19	0.003	0.960
0.20	0.015	0.975
0.21	0.015	0.990
0.22	0.002	0.992
0.25	0.001	0.993
0.26	0.003	0.996
0.27	0.002	0.998
0.28	0.002	1.000

Table 8.3 Packet Interarrival Times and Probabilities

4. Defining Routing Classes

The routing classes defined using the following steps will be used to specify hop limits for the multicast packets generated by the routing and generation of multicast video packets.

a) From the *Define* menu select the *Routing Class/ Packets* option. In the packet routing classes window which appears, press the *Add* button.

b) Edit the following fields in the packet routing class window.

- Name: MBONE 1
- Hop limit: 1

c) When all fields are entered, press the *OK* button in the packet routing class window. Then, press the *Add* button in the packet routing classes window again.

d) Edit the following fields in the packet routing class window.

- Name: MBONE 6
- Hop limit: 6

e) When all fields are entered press the *OK* button in the packet routing class window. Then, press the *Done* button in the packet routing classes window.

5. Creating the NetVideo Message Source

The NetVideo message source will be used to model the broadcast of multicast packets for desktop video conferencing.

a) Create a message source, and edit the fields of the message source window to those shown in the following list. Figure 8.12 depicts the message source window upon completion.

- Name: UCSC NET VIDEO
- Schedule by: Iteration time
- Interarrival: NVINTER-TAB
- Message size calculation: Probability distribution
- Probability distribution: NVSIZE-TAB
- Priority: 1
- Routing class: MBONE 1
- Transport protocol: UDP/IP-Sun
- Packetize: 0.05 milliseconds
- Message text option: Copy message name
- Destination type: Set to multicast list, and create a list containing only the node UCSC2 MBONE ROUTER.

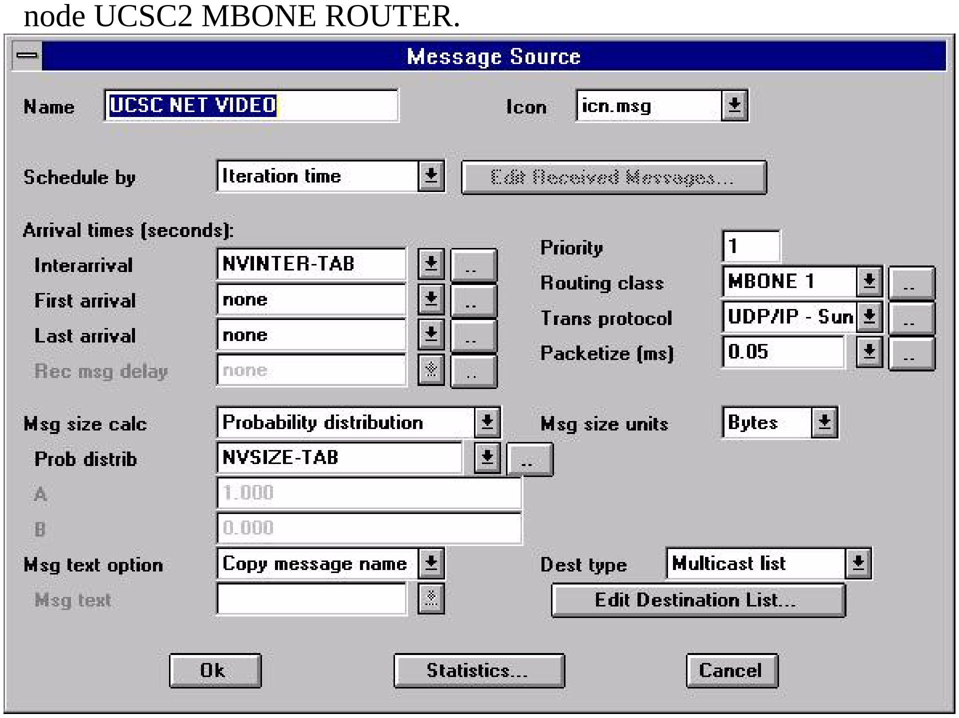


Figure 8.12 NetVideo Message Source

b) When all fields have been entered, press the *OK* button at the bottom of the message source window.
c) Create a clone of the NET VIDEO message source.
d) Attach one copy of the message source to the node UCSC1 and the other to the node UCSC3.

6. Creating the Multicast Routing Message Sources

The multicast router will route packets both within the local area network and to other multicast networks across the wide area network. Two message sources are necessary to model routing both in and out of the local area network. The following steps outline the steps for creating these sources:

a) Create a message source and attach it to the node UCSC2MBONE ROUTE.
b) Edit the fields of the message source window to those shown in the following list. Figure 8.13 depicts the message source upon completion.
    - Name: UCSC MBONE INT ROUTING
    - Schedule by: Received message. The received message list will be created later.
    - Priority: 1
    - Routing class: MBONE 1
    - Transport protocol: UDP/IP-Sun
    - Packetize: 0.05 milliseconds
    - Message size calculation: (A * Msg bytes) + B
    - A: 1.000
    - B: 0.000
    - Message text option: Use original message
    - Destination type: Set to multicast list, and create a list containing only the nodes UCSC1 and UCSC3.

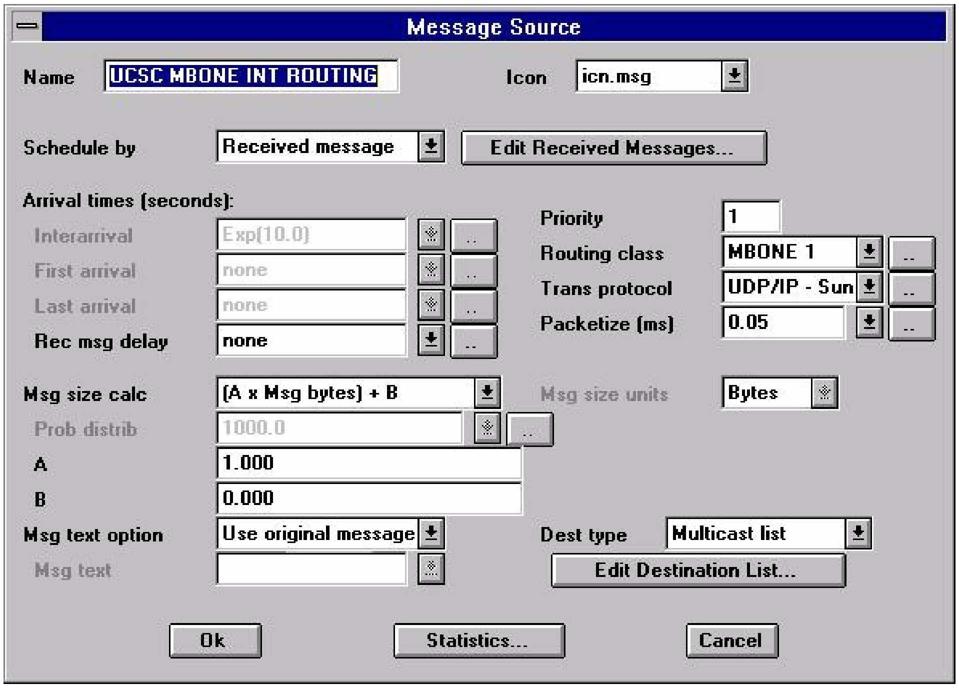


Figure 8.13 MBONE INT ROUTING Message Source

c) When all fields are entered, press the *OK* button at the bottom of the message source window.

d) Create a copy of the UCSC MBONE ROUTING message source and attach the copy to the node UCSC2 MBONE ROUTER. Change the following fields of this message source:

- Name: UCSC MBONE EXT ROUTING
- Routing Class: MBONE 6

e) The internal structure of the first local area network is now completed with the exception of defining the received message lists and destinations for the message sources which will be used for routing. The view inside the SANTA CRUZ subnetwork should look similar to that of Figure 8.14.

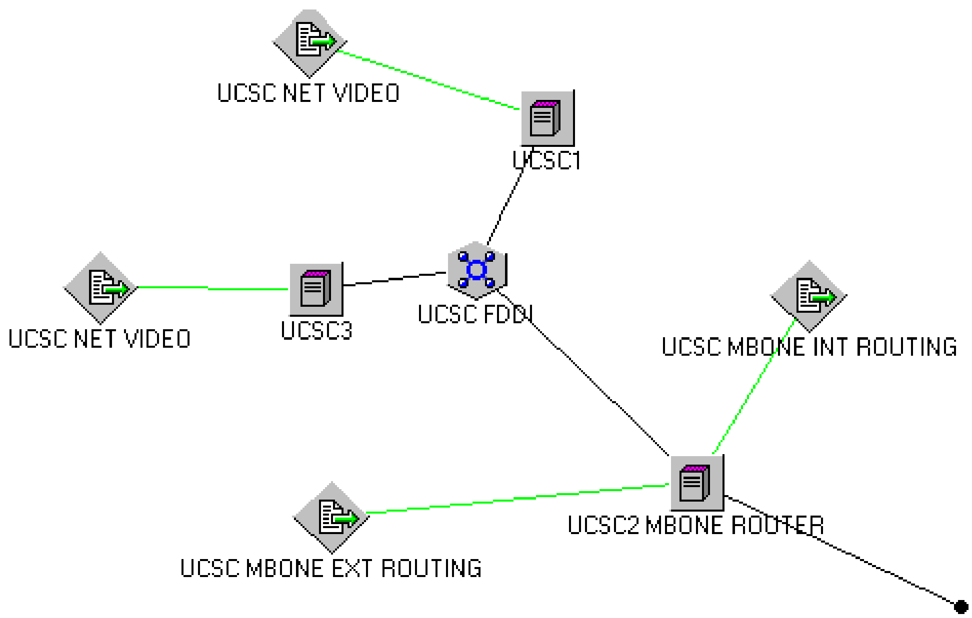


Figure 8.14 Subnetwork Model View

7. Creating the Other Local Area Networks

The two other local area networks are identical to the local area network created in the SANTA CRUZ subnetwork. These additional networks will be created by cloning the SANTA CRUZ subnetwork and then renaming the nodes inside the subnetwork.

a) Leave the SANTA CRUZ subnetwork view to return to the backbone view.

b) Select the SANTA CRUZ subnetwork and make two clones of this subnetwork. Rename the cloned subnetworks MONTEREY and SAN FRANCISCO.

c) Enter the MONTEREY subnetwork and rename the nodes and message sources within the subnetwork to those shown in Table 8.4. After all the nodes and message sources have been renamed in the MONTEREY subnetwork, leave this subnetwork and enter the SAN FRANCISCO subnetwork. Rename all of the nodes and message sources in this subnetwork to those shown in Table 8.4.

d) The received message lists and destination list for each message source created are listed in Table 8.5. Enter and leave the subnetworks as necessary to create these received message lists and destination lists.

SANTA CRUZ Subnetwork Name	MONTEREY Subnetwork Name	SAN FRANCISCO Subnetwork Name
UCSC1	NPS1	STANFORD1
UCSC2 MBONE ROUTER	NPS2 MBONE ROUTER	STANFORD2 MBONE ROUTER
UCSC3	NPS3	STANFORD3
UCSC NET VIDEO	NPS NET VIDEO	STANFORD NET VIDEO
UCSC MBONE INT ROUTING	NPS MBONE INT ROUTING	STANFORD MBONE INT ROUTING
UCSC MBONE EXT ROUTING	NPS MBONE EXT ROUTING	STANFORD MBONE EXT ROUTING

Table 8.4 Subnetwork Node and Message Source Names

Message Source	Received Message List	Destination List
UCSC NET VIDEO	N/A	UCSC2 MBONE ROUTER
UCSC MBONE INT ROUTING	NPS NET VIDEO, STANFORD NET VIDEO	UCSC1, UCSC3
UCSC MBONE EXT ROUTING	UCSC NET VIDEO	NPS2 MONE ROUTER, STANFORD2 MBONE ROUTER, UCSC1, UCSC3
NPS NET VIDEO	N/A	NPS2 MBONE ROUTER
NPS MBONE INT ROUTING	UCSC NET VIDEO, STANFORD NET VIDEO	NPS1, NPS3
NPS MBONE EXT ROUTING	NPS NET VIDEO	UCSC2 MONE ROUTER, STANFORD2 MBONE ROUTER, NPS1, NPS3
STANFORD NET VIDEO	N/A	STANFORD2 MBONE ROUTER
STANFORD MBONE INT ROUTING	UCSC NET VIDEO, NPS NET VIDEO	STANFORD1, STANFORD3
STANFORD MBONE EXT ROUTING	STANFORD NET VIDEO	UCSC2 MONE ROUTER, NPS2 MBONE ROUTER, STANFORD1, STANFORD3

Table 8.5 Received Message Lists and Destination Lists for Message Sources

e) The local area network topologies and the message sources are all now completed. Verify the model. A list of errors will appear that should only contain errors stating that the subnetworks are not connected to any links within the model. Save the model using the name **mbone6.c3** .

8. Modeling the Wide Area Network Using ATM Switch Nodes and Point-To-Point Links

The following section defines the steps for modeling the wide area network using switches and point-to-point links. This model should look similar to Figure 8.15 upon completion.

a) Save the model **mbone6.c3** using the name **mbone6a.c3** to create a new model.

b) Select the *Node Parameters/ATM* option from the *Define* menu. In the ATM parameters window which appears press the *Add* button. Edit the fields in the second ATM parameters window which appears to the following:

- Name: POP SWITCH
- Port processing default times: Set to 0.005 milliseconds for both input and output.

c) Create and ATM switch node in the backbone view and edit the fields of the node parameters window to the following:

- Name: SANTA CRUZ SWITCH
- Type: ATM Node
- Parameters: POP SWITCH

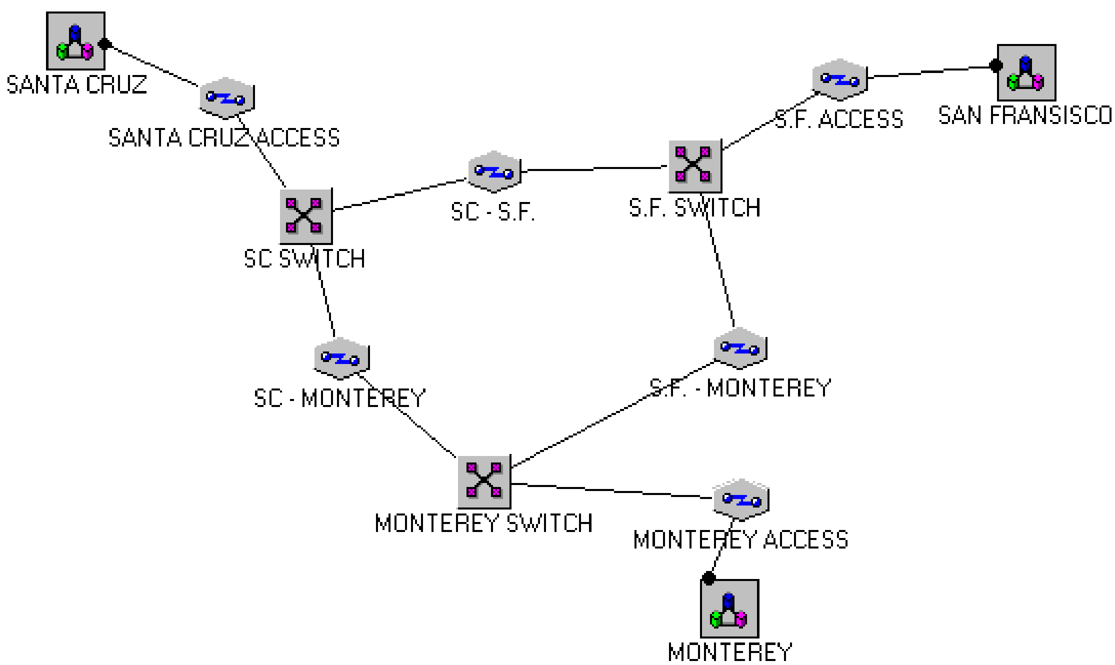


Figure 8.15 Model Backbone View

d) Create two copies of the SANTA CRUZ SWITCH ATM node. Rename these copies MONTEREY SWITCH and S.F. SWITCH.

e) Select the *Link Parameters/Point-to-Point* option from the *Define* menu. In the point-to-point parameters window which appears press the *Add* button. In the library selection window which appears choose default in the list.

f) Edit the fields of the point-to-point parameters window to the values shown in the following list. Figure 8.16 depicts the parameter set upon completion.

- Parameter set name: 256 KBPS FRAME RELAY
- Number of circuits: 1
- Bandwidth/circuit (from node X): 256
- Bandwidth/circuit (from node Y): 256
- Frame min: 13
- Frame max: 4108
- Frame overhead: 12

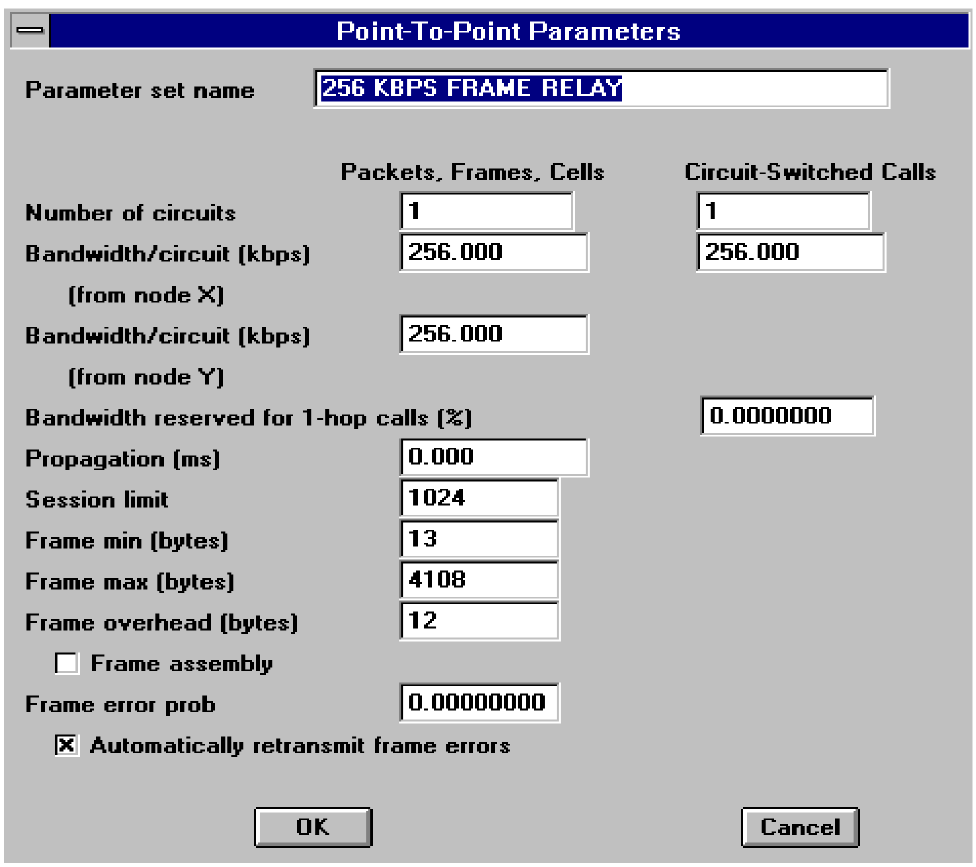


Figure 8.16 256 Kbps Frame Relay Point-To-Point Link Parameter Set

g) Create a second point-to-point link parameter set in the same fashion as the first using the following values:

- Parameter set name: 256/512 KBPS FRAME RELAY
- Number of circuits: 1
- Bandwidth/circuit (from node X): 256
- Bandwidth/circuit (from node Y): 512
- Frame min: 13
- Frame max: 4108
- Frame overhead: 12

h) Create a point-to-point link and attach the link to the SANTA CRUZ subnetwork and the SANTA CRUZ SWITCH node. Edit the fields in the link detail window of this point-to-point link to the following:

    - Name: SANTA CRUZ ACCESS
    - Type: point-to-point
    - Parameters: 256/512 KBPS FRAME RELAY

i) Create two copies of the SANTA CRUZ ACCESS link. Rename one copy MONTEREY ACCESS and attach it to the MONTEREY SWITCH node and the MONTEREY subnetwork. Rename the other copy S.F. ACCESS and attach it to the S.F. SWITCH node and the SAN FRANCISCO subnetwork.

j) Create another point-to-point link and attach it to the SANTA CRUZ SWITCH and MONTEREY SWITCH nodes. Edit the fields of the link detail window to the following:

    - Name: SC - MONTEREY
    - Type: point-to-point
    - Parameters: 256KBPS FRAME RELAY

k) Create two copies of the SC - MONTEREY link. Rename one copy S.F. - MONTEREY and attach it to the MONTEREY SWITCH node and the S.F. SWITCH . Rename the other copy SC - S.F. and attach it to the S.F. SWITCH node and the SANTA CRUZ SWITCH.

l) The second model is now completed. Verify and save the model.

9. Modeling the Wide Area Network Using a WAN Cloud

The following steps outline the method for modeling the frame relay wide area network using a WAN cloud. The backbone view of the model should look similar to Figure 8.17 upon completion.

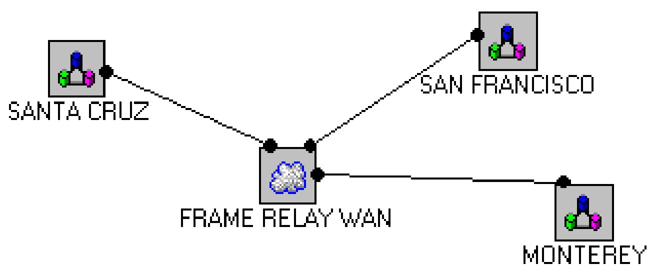


Figure 8.17 Model Backbone View

a) Open the model **mbone6.c3** and save it using the name **mbone6c.c3** .

b) Create a WAN cloud and place it in the work area. Select the cloud and choose the *Detail* option from the *Edit* menu. Edit the following fields in the cloud detail window:

- Name: FRAME RELAY WAN
- Type: WAN Cloud
- Parameters: Frame Relay

c) Enter the WAN cloud.

d) From the *Define* menu choose the *WAN Cloud parameters/Access Link* option. Edit the fields in the cloud access link parameters window to the values shown in the following list. Figure 8.18 depicts the access link parameter window upon completion.

- Parameter set name: 256/512 KBPS ACCESS
- Number of circuits: 1
- Entry bandwidth/circuit: 256
- Exit bandwidth/circuit: 512
- Exit buffer: 16834

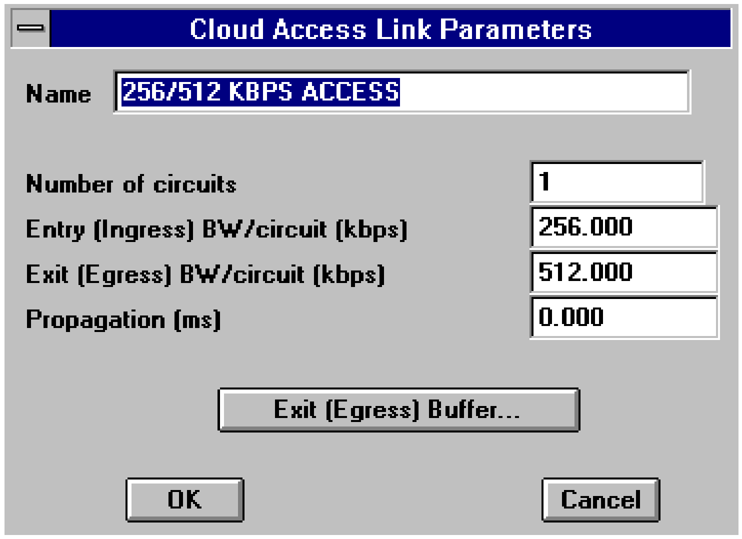


Figure 8.18 Cloud Access Link Parameter Window

e) Create an access link within the WAN cloud. Edit the following fields in the access link window:

- Name: SANTA CRUZ
- Parameters: 256/512 KBPS ACCESS

f) Create two copies of the SANTA CRUZ access link. Rename the links MONTEREY and SAN FRANCISCO.

g) Create three access points in the WAN cloud. Attach one to each of the access links.

h) Create one virtual circuit in the WAN cloud. Edit the fields for this circuit to the values shown in the following list. Figure 8.19 depicts the circuit parameters upon completion.

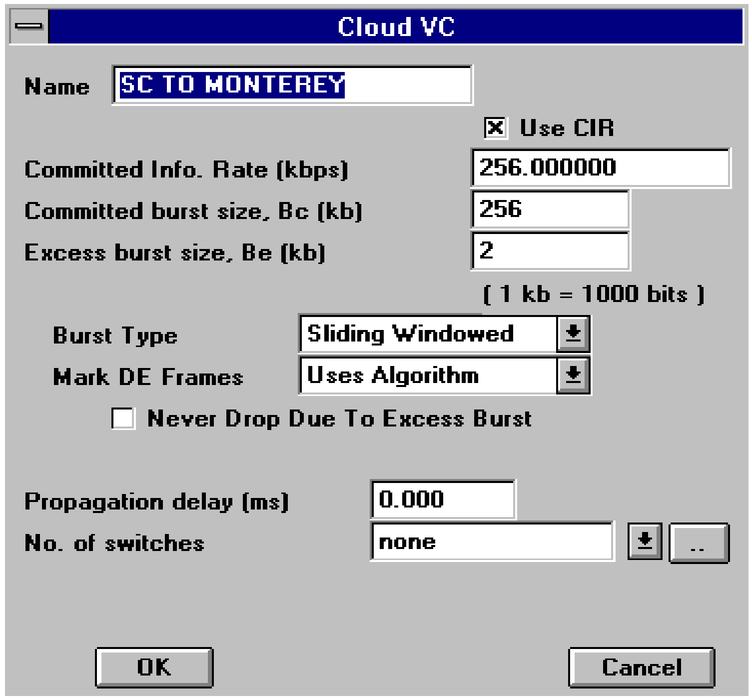


Figure 8.19 Virtual Circuit Parameters

- Name: SC TO MONTEREY
- Committed info rate: 256
- Committed burst size: 256
- Excess burst size: 2
- Show burst size box: checked

i) Create five clones on the SC TO MONTEREY virtual circuit. Rename the circuits according to the names shown in Table 8.6 and connect the access point as listed in the same table. It is important that the virtual circuits be connected as per the table as they are one way circuits. Figure 8.20 depicts a view of the interior of the WAN cloud with all access links and virtual circuits defined.

Name	From Access Link	To Access Link
SC TO MONTEREY	SANTA CRUZ	MONTEREY
MONTEREY TO SC	MONTEREY	SANTA CRUZ
MONTEREY TO S.F.	MONTEREY	SAN FRANCISCO
S.F. TO MONTEREY	SAN FRANCISCO	MONTEREY
SC TO S.F.	SANTA CRUZ	SAN FRANCISCO
S.F.TO SC	SAN FRANCISCO	SANTA CRUZ

Table 8.6 Virtual Names and Access Link Connections

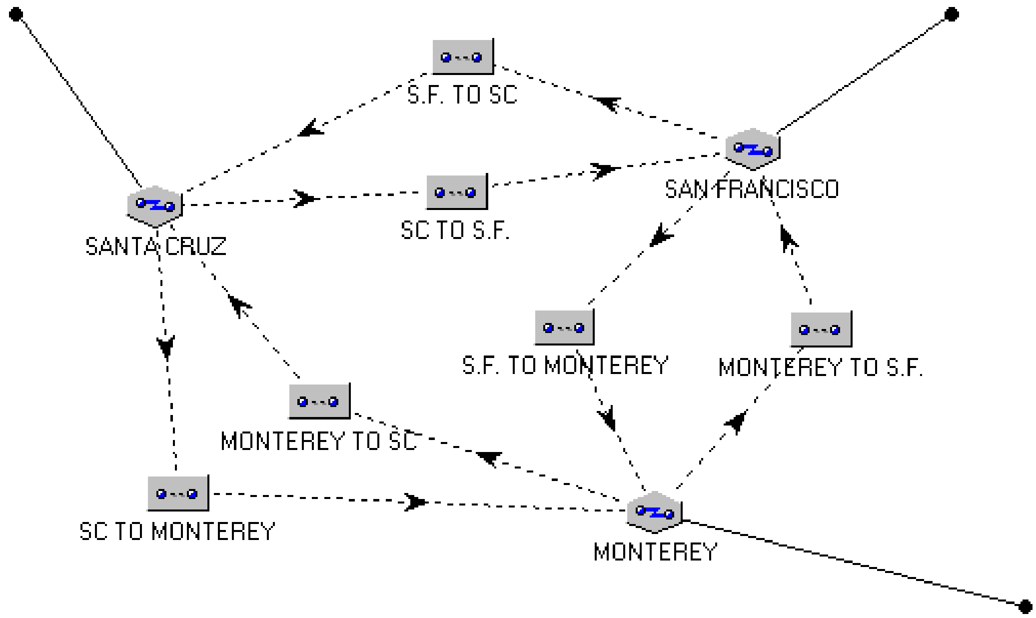


Figure 8.20 Model WAN Cloud View

j) Leave the WAN cloud and attach each of the access points of the WAN cloud an access point on one subnetwork.

k) This model is now completed. Verify and save the model.

H. SIMULATION AND RESULTS

For each model, one simulation replication was run for each model using a simulation time of 60 seconds and a warmup length of 5 seconds. Prior to running the simulation the following reports were selected:

- Link Reports: Channel Utilization for all links in both models.
- WAN Cloud Reports: Frame Delay by VC, Frame Count by VC, and Access Link Statistics for the model containing the WAN cloud.
- Message and Response Source reports: Message Delay for all message sources in both models.

The results of the two simulations in determining the delay across the wide area network differed by 15 milliseconds. The WAN cloud model had an average delay across the network of 62 milliseconds while the total average delay across all links in the wide area network was 77 milliseconds. Both simulations showed that the ability of the multicast router to process and resend packets very rapidly will greatly affect the delays of the video packets. The simulations do show that the transmission of video packets across a frame relay network should be possible. Which method of modeling the wide area network is better? The simulation results for both were fairly close. The main advantage of using the WAN cloud to model the network is the ease of making a simpler model and the ability of the cloud to specify the grade of service of the virtual circuits which point-to-point links cannot provide.

IX. CONCLUSION

A. SUMMARY

The purpose of this thesis was to provide a hands-on tutorial to allow users of the COMNET III application to quickly ramp up on the capabilities which this application provides to model and simulate local and wide area networks. The COMNET III Users manual contains complete information on the theory and capabilities of each object, but it presents no logical structure for understanding the manner in which these objects tie together when modeling a network.

Each chapter presents the theory of several objects which are available for modeling networks, presents a network problem which relates to the objects discussed, and provides instruction to build a model to demonstrate the ability of COMNET III to analyze network problems. The goal of each model, when possible, was to use information which was gleaned from actual networks present on the Naval Postgraduate School campus. The primary networks from which data was gathered was the Systems Technology Lab's ethernet and fiber distributed data interface (FDDI) networks. The gathering of data was performed using the Snoop protocol analyzer program or by conducting interviews with network managers to obtain estimates of the types of traffic on the network, current network problems, and commonly used applications which are used on computers attached to the network.

B. COMNET III APPLICABILITY WITHIN THE MILITARY

If the current trend toward the use of commercial technology for the development of military networks continues, COMNET III could have wide applicability in assisting in the design and performance analysis of packet-switched and local area networks used by the military. Operation Desert Shield/Desert Storm provides good evidence that this trend will continue as military personnel brought not only the weapons necessary to fight the war but also their personal computers. To meet the information needs in this conflict, Hewlett Packard workstations were used to act as gateways for ad hoc wide area networks and local area networks which provided connectivity between personal computers used by the CENTCOM staff.

The other major factor which makes simulation of networks a needed tool available within the military is the shrinking Department of Defense budget combined with an increased dependence within the military on the ability to transfer information to any level that it may be needed in a timely manner. COMNET III's graphical user interface allows users to quickly develop network models which may be used to assess high level performance measures of a network. Proposed network designs could be modeled to assess their suitability and reliability in delivering data and necessary adjustments made prior to a network being physically built. This would provide great cost saving in minimizing network modifications.

BILBLIOGRAPHY

1. Held, G., *Local Area Network Performance: Issues and Answers*, John Wiley and Sons Ltd., England, 1994

2. Law, A. and Kelton, W., *Simulation Modeling and Analysis*, McGraw-Hill Book Company, 1982

3. Mercer, J., Farrell, C. A., and Schulze, M., *Accurate Network Load Monitoring: Loadman*, Department of Computer Science, Curtin University of Science, Australia, 1994

4. Minoli, D., *Enterprise Networking: Fractional T1 to SONET, Frame Relay to BISDN*, Artech House, Boston, 1993

5. Van Norman, H. J., *LAN/WAN Optimization Techniques*, Artech House Inc., Boston,1992