Chapter 4

LAN Environment

The local area network (LAN) modeling features of netWorks are designed to investigate connectionless networks in which data is transmitted between devices in discrete units called packets. And unlike in a connection-oriented network, a path between a source and destination is not known before transmission begins. In fact, a valid path may not exist between a source and destination in this type of network. A connectionless network relies on forwarding or routing algorithms running in various network devices to forward a packet appropriately through the network to its final destination.

In a typical LAN model network, you might have collections of Computing Device models—Workstation, Server, and Printer models—connected to a Hub or Media model representing a single LAN and a series of LANs linked by Bridge or Router models. The Workstation and Server models generate packets destined for other Computing Device models throughout the model network, and Router and Bridge models are responsible for forwarding packets between subnetworks.

The LAN environment is less mature than the telecom environment of the netWorks application, and therefore it offers comparably less flexibility and functionality than its connection-oriented sibling. You can, however, still perform very interesting and useful LAN simulations using this environment.

LAN Equipment

The LAN equipment models are developed around the premise that a LAN architecture consists of equipment connected to a transmission media through a network interface card (NIC). The default equipment models provided in the LAN environment are very different from the telecom equipment models, both in terms of structure and functionality. The four categories of LAN models are

- Network Interface Card (NIC)
- Transmission Media
- Computing Device
- Internetwork Device.

All Internetwork and Computing Device models contain at least one NIC that is used to connect the model (through arcs) to a Transmission Media model to form a model LAN. All LAN equipment models also have reliability controls for simulating equipment failure and restoration. A sample Media Reliability Controls panel is shown in Figure 4.1.

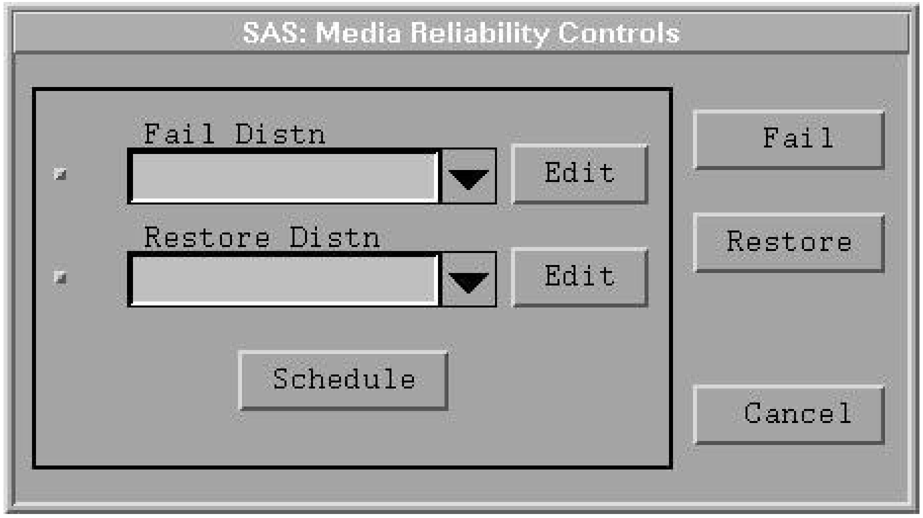

Figure 4.1. Sample Media Reliability Controls

You can immediately fail or restore an equipment model by clicking the Fail and Restore buttons on the Reliability Controls, or you can schedule these events to occur throughout the simulation. The Fail Distn and Restore Distn controls are used to select statistical distributions for scheduling reliability events. When Fail Distn and Restore Distn are operational and you click the Schedule button, the simulation samples from the respective statistical distributions and schedules the events accordingly. The check boxes to the left of the Fail Distn and Restore Distn controls can be used to disable or enable their scheduling.

NIC Model

A NIC model controls access to a Transmission Media model for its owner. That is, it provides media access control (MAC) layer functionality in the LAN simulation so that different devices know how to share a Transmission Media model. A NIC model cannot exist as a separate entity—it must always be "part of" some other model. Computing Device models always have one NIC, and Internetwork Device models typically have multiple NICs.

All NICs attached to the same Transmission Media model must be running the same MAC layer protocol for the simulation to run properly. For example, if some NICs on a LAN are configured with an Ethernet protocol while others are running a Token Ring protocol, you might expect to experience significant packet transmission problems. MAC protocols are discussed in more detail in the "**??**" of Appendix **??**, "**??**,".

A sample control panel for a NIC model is presented in Figure 4.2.

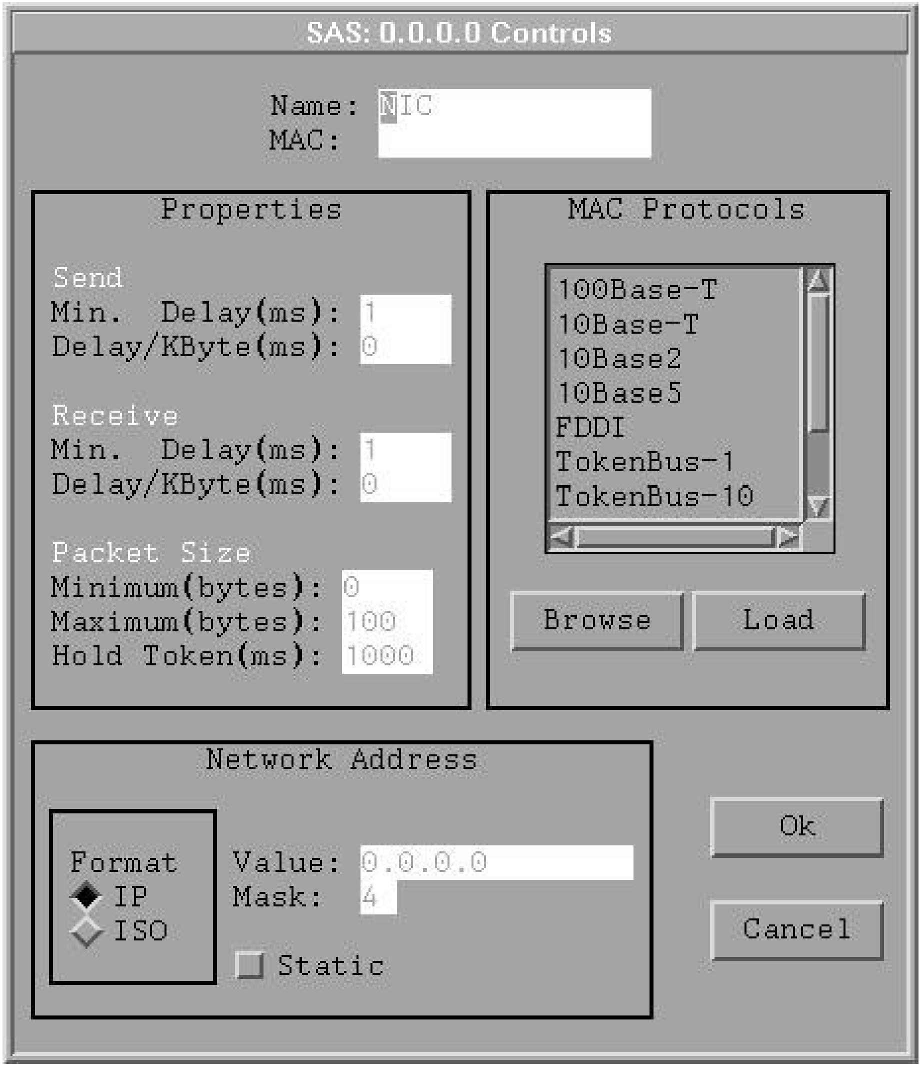

Figure 4.2. Sample NIC Model Controls

The properties of a NIC model are analogous to the attributes of a telecom model. The Send and Receive delay properties control the simulated packet transmission delay associated with a NIC, and the Packet Size properties impose size constraints on any data packets sent through this device. You can set the MAC protocol for the NIC model using the MAC Protocols list box and the Load button. Note that the loading of a MAC protocol may result in changes to the property values of the NIC model.

Each NIC model is also required to have a Network Address; the format of this address depends on the routing protocol used in your simulation. Network addresses are discussed in more detail in the "**??**" section of Appendix **??**, "**??**,". You can set the network address of the NIC through its control panel or, more commonly, you can have it assigned through a Transmission Media model. Setting the Static check box prohibits the network address of a NIC from being assigned from another source.

Each NIC keeps a count of the packets that flow through it, and you can display these counts by selecting Statistics in the NIC pop-up menu. A sample NIC Statistics window is displayed in Figure 4.3.

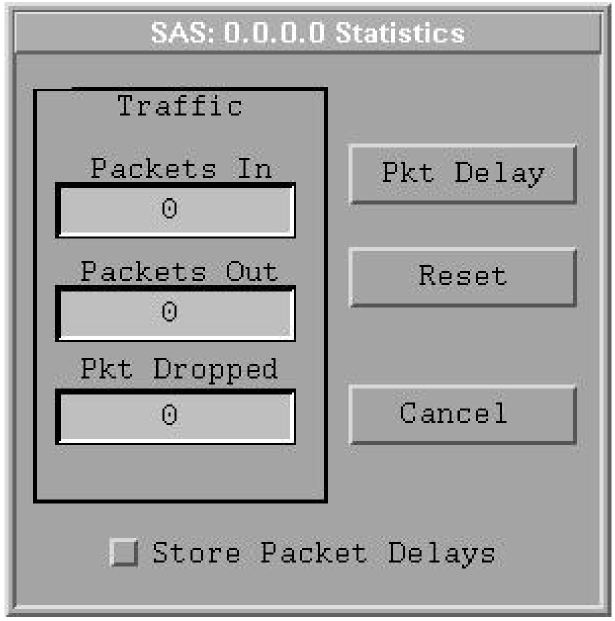

Figure 4.3. Sample NIC Statistics Controls

You can store the transmission delays associated with incoming packets by selecting the Store Packet Delays check box. As in the telecom environment, netWorks plots can be used to display the stored values.

Only NIC and Transmission Media models can have arcs between them. You can create an arc from a NIC model just as you would for a simple equipment model in the telecom environment—either using the Arc selection from its pop-up menu or using the hotspot on the drawing. See the "**??**" section of Chapter **??**, "**??**," for a complete description of creating an arc with a hotspot.

Transmission Media Models

The Transmission Media models are used to represent the physical cable over which packets travel between various devices on a LAN. They are restricted to being connected to NIC models. The two types of Transmission Media models provided in netWorks are the Hub and Media models. These models are functionally equivalent in this application with the only difference between them being their icons.

A sample control panel for a Media model is shown in Figure 4.4.

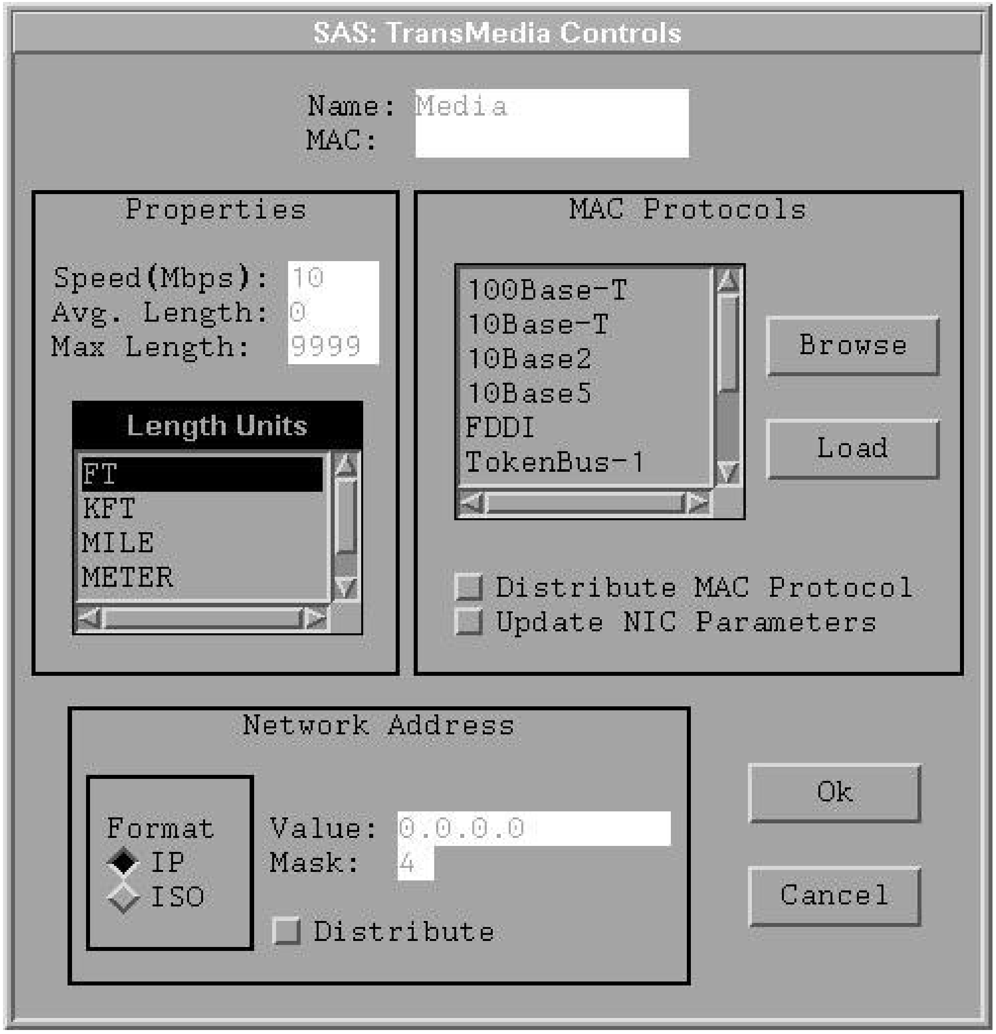

Figure 4.4. Sample Media Model Controls

Note that this control panel is very similar to the NIC control panel. The properties of the Media model (Speed, Avg. Length) are used to calculate the propagation delay of a packet over the simulated media. The Average Length value is used to represent the average distance between two devices connected to this Media model. The Network Address section of the control panel is similar to that of a NIC model except that the Static check box is replaced with a Distribute check box. You can use this feature to assign Network Addresses automatically to any NIC models attached to Transmission Media models when you exit the dialog box. (The application uses the network address of the Media model as the base address for its subnetwork and increments the host portion of the address to generate network addresses for each of the attached models.) The MAC Protocols section of the control panel has a similar check box for automatically distributing a MAC protocol to all attached NIC models as well. These features expedite the network model build process.

Computing Device Models

The Computing Device models are used to represent the sources and final destinations of data packets in the LAN environment. These models are sometimes referred to as *endnodes*. Each Computing Device model contains one NIC; in some aspects these models are similar to compound models in the telecom environment. The NIC is totally encapsulated by its Computing Device model, and you can only connect arcs to a Computing Device model through its NIC. Unlike a telecom compound model, a Computing Device model has properties associated with it that influence simulation results. The logical view of a Computing Device model is depicted in Figure 4.5.

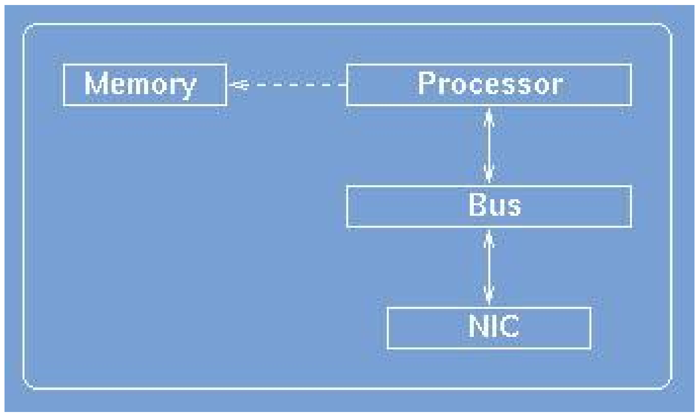

Figure 4.5. Computing Device Model Architecture

As is discussed in the individual model descriptions, you can edit the properties associated with the various logical components of a Computing Device model. (Note, however, that the Memory property is not used by the simulation in the current version of netWorks.) There are three Computing Device models—Printer, WorkStation, and Server. The details of each model are discussed in the following sections.

Printer Model

The simplest Computing Device model is the Printer equipment model. This model cannot generate independent data packets; therefore, its primary use in a LAN simulation is as a final destination for data packets from other Computing Device models. A sample control panel for a Printer model is shown in Figure 4.6.

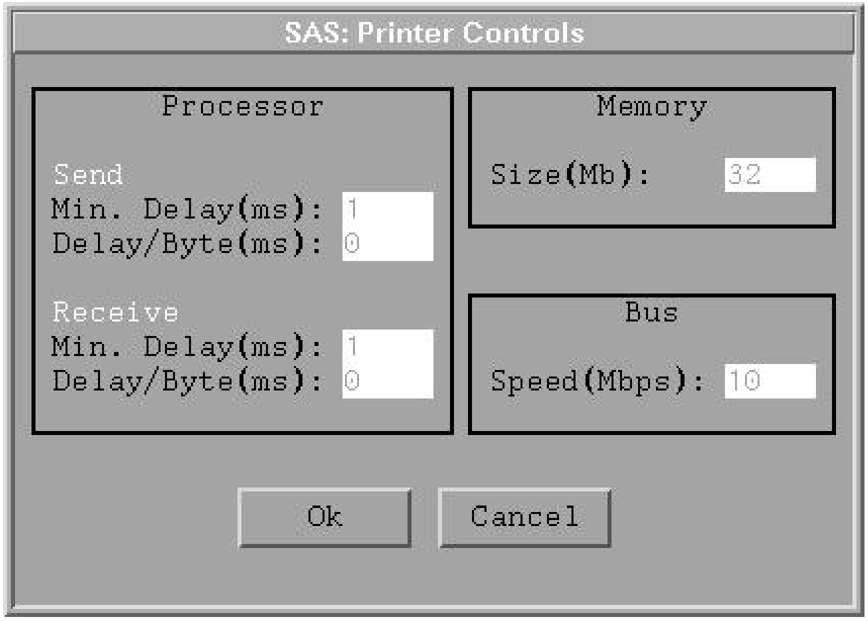

Figure 4.6. Sample Printer Model Controls

All Computing Device models have the Bus Speed, Memory Size, and Processor Delay properties. While you can edit the Memory Size property, its value is not used during a simulation and its only function is documentation. The Bus Speed property is used to determine how long it takes a packet to travel between the processor and NIC of a model. The Send and Receive Processor Delays are used to simulate packet delays associated with processor processing. The processor packet delays are calculated using the following formula:

$$delay = MinDelay * (PacketSize * \frac{Delay}{Byte})$$

Note that there is nothing in this model specific to a printer and since a Printer model can only be a destination, its properties (processor delays and bus speed) affect only the routing protocol packets this model originates. As with all Computing Device models, you can display a window with processor traffic statistics using the Options⇒Statistics selection from its pop-up menu.

Workstation Model

In addition to the common Computing Device properties, the Workstation model features network routing protocols and the capability to generate data packet traffic destined for other Computing Device models throughout your model LAN network. The Workstation equipment model is the primary traffic generator in the LAN environment. The logical architecture of a Workstation model is depicted in Figure 4.7.

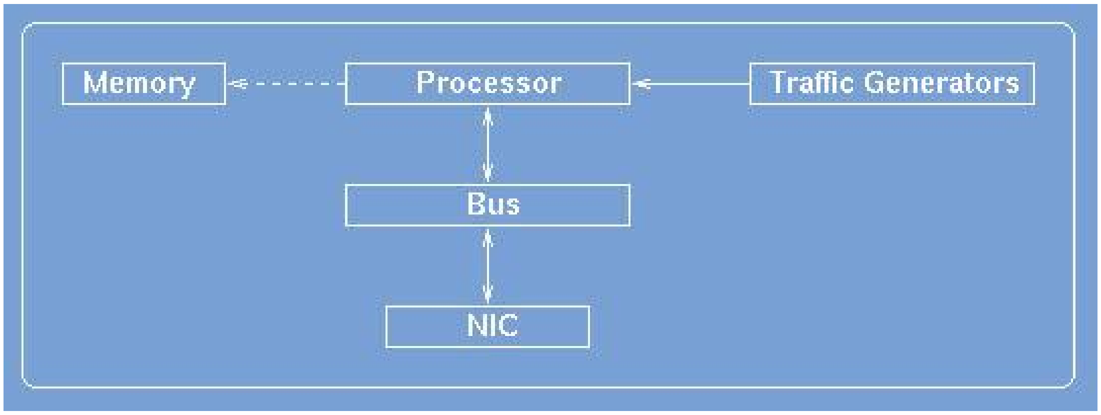

Figure 4.7. Workstation Model Architecture

This model attaches traffic generator modules to the processor unit to spawn outgoing data packets. You can have multiple traffic generators in each Workstation model, or you can have no traffic generators at all.

A sample Workstation model control panel is displayed in Figure 4.8.

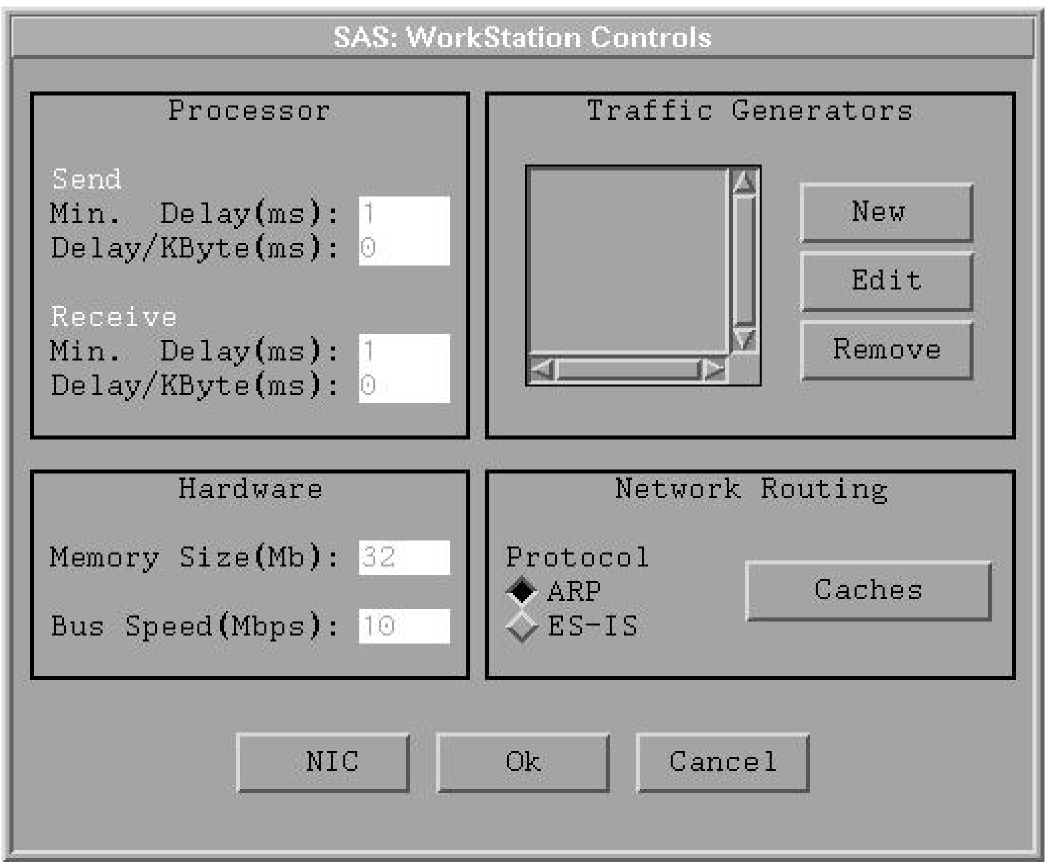

Figure 4.8. Sample Workstation Model Controls

In addition to the processor and hardware properties, this model has network routing and traffic generator controls and also a push button to access the associated NIC controls. The Network Routing protocol property is controlled through a simple radio control, and the contents of any routing caches can be displayed by selecting the Caches button. The "**??**" section of Appendix **??**, "**??**," discusses routing protocols in detail.

The list box in the Traffic Generators section of the control panel contains the names of all the traffic generators attached to this model. The Edit and Remove buttons operate on the item selected in the list box, and clicking the New button creates a new traffic generator.

The control panel for a LAN traffic generator (Figure 4.9) is displayed when you click either the New or Edit button.

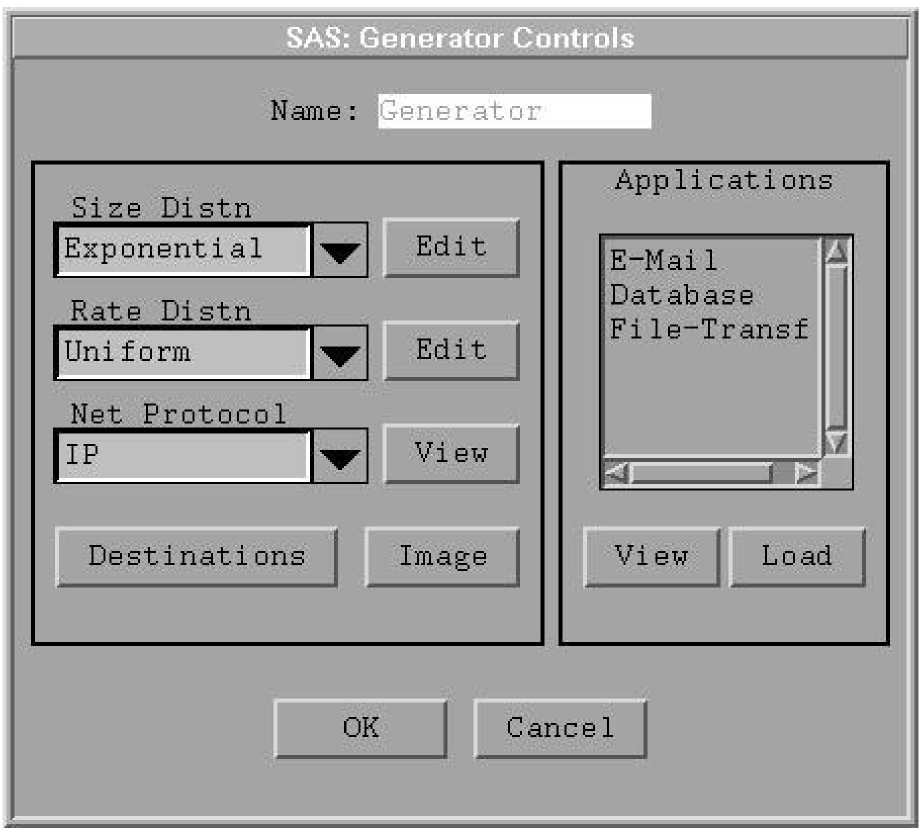

Figure 4.9. Sample LAN Traffic Generator Controls

These controls are very similar to a telecom call generator. The Size Distn determines the number of bytes of data to send as traffic to another Computing Device model, and the Rate Distn controls the time between traffic generation. The Net Protocol controls are not used by netWorks at this time, and the Destinations controls are completely analogous to the Destinations controls in the telecom environment. The intent of the Applications section is to provide predefined traffic generators modeling common network applications. The default Application values are for demonstration purposes only.

Server Model

The Server equipment model is designed to simulate a file server or database server in the LAN environment. It has all the features of a Workstation model and the additional capability of responding to incoming data packets. That is, the Server model can send data packets back to the source of an incoming data packet. The idea is to model requests for a file transfer or database query to a server along with the associated response.

Sample Server model controls are shown in Figure 4.10.

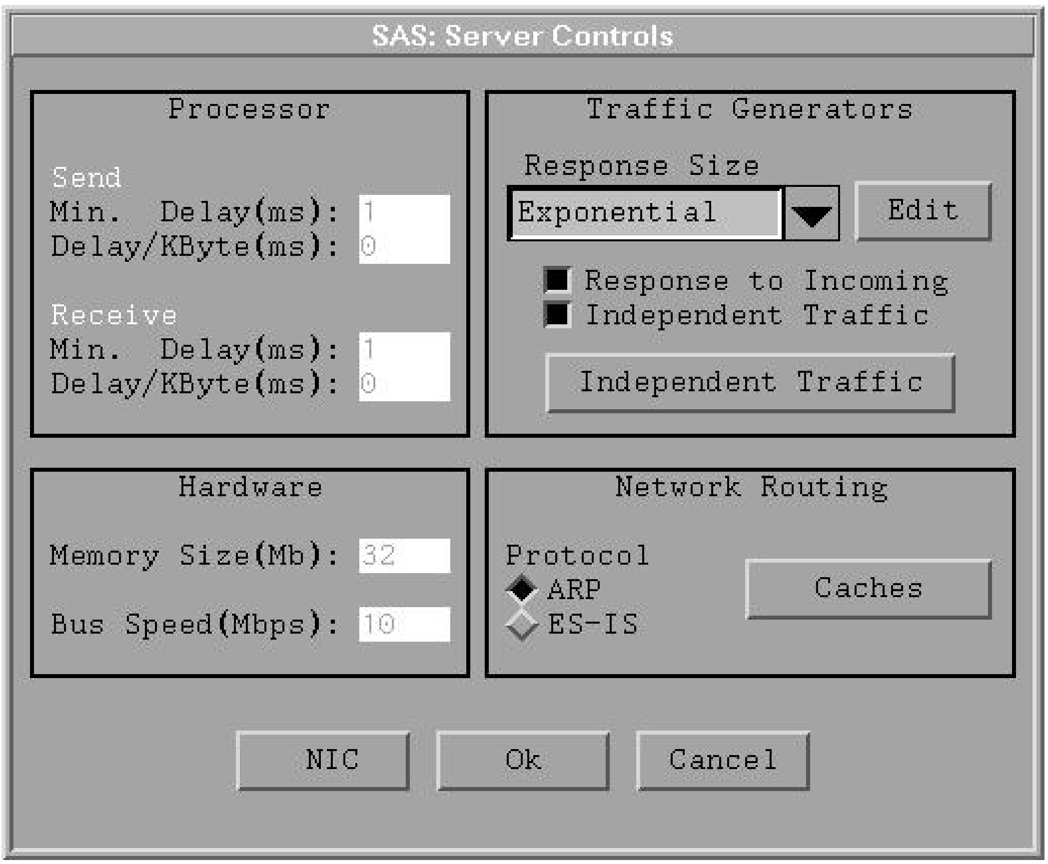

Figure 4.10. Sample Server Model Controls

The Traffic Generators section of the Server controls reflects the additional features of the Server model, and it is a bit different from the Workstation model. You can use the Response to Incoming and Independent Traffic check boxes to control the types of traffic originated by this model. The Response Size control lets you determine the size of the data packet (number of bytes) sent back to the source of an incoming data packet. Clicking the Independent Traffic button produces a new window with controls for additional traffic generators, analogous to Workstation model traffic generators.

Internetwork Device Models

The Internetwork Device models are responsible for forwarding packets between LAN segments (that is, Transmission Media models), and they are the most complex models in netWorks LAN environment. Like Computing Device models, Internetwork Device models are connected to Transmission Media models through NICs and arcs, and they also contain a logical processor and bus unit. However, these models necessarily support multiple NICs connected to the bus unit. The logical architecture of an Internetwork Device model is depicted in Figure 4.11.

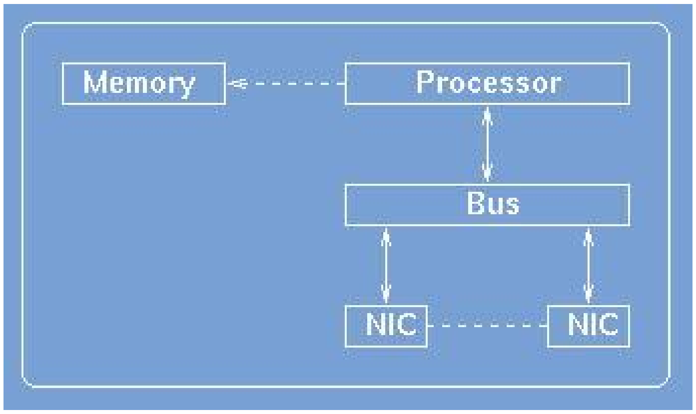

Figure 4.11. Internetwork Model Architecture

The Internetwork Device models for Bridge and Router devices are provided in netWorks.

Bridge Model

In a real LAN topology, bridges operate in the data link layer of the OSI Reference model [Tanenbaum 1996]; therefore, they ignore any network layer information in the data packets. A bridge buffers each packet in its memory and then decides if it needs to forward the packet onto another LAN. Different types of bridges use different algorithms for determining when and where to forward packets. All of these algorithms keep an internal cache containing packet forwarding information. The algorithms' details can be found in [Tanenbaum 1996] and [Perlman 1992]. The Bridge model in netWorks uses a spanning tree algorithm.

Sample Bridge model controls are shown in Figure 4.12.

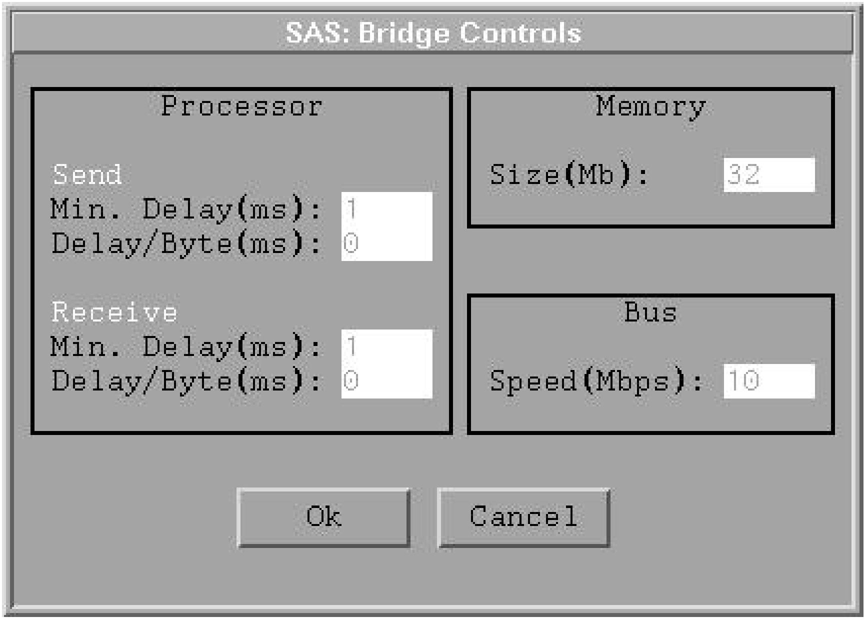

Figure 4.12. Sample Bridge Model Controls

Since a bridge buffers its packets before it forwards or drops them, the amount of memory a bridge contains can have a significant impact on its performance. If its buffers are full, it could end up dropping or missing packets it should be forwarding. The current simulation routines, however, ignore the Memory property and assume that the device has infinite memory.

Router Model

The term *router* is also referred to as an *intermediate system*, *gateway*, or *packet switch*. These devices use the network layer information to route packets from a source to a destination on a different LAN. To route packets through the network accurately and efficiently, each router must have some knowledge about the entire network topology. Usually this information can be limited to knowing the network addresses and router IDs of the routers on the network. A variety of routing protocols have been developed to coordinate the exchange of the router information and for determining the "best" paths through a network. The netWorks application currently supports two routing protocols.

The architecture of a Router model is shown in Figure 4.13, and sample Router model controls are presented in Figure 4.14.

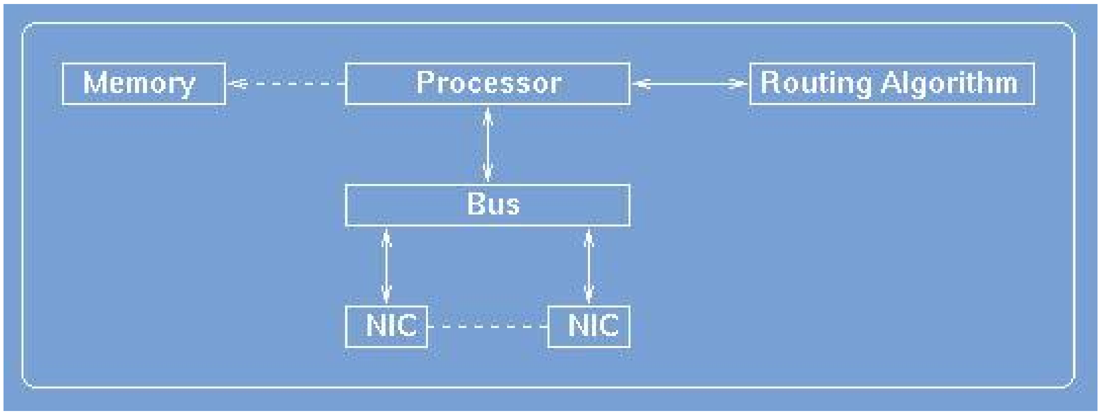

Figure 4.13. Router Model Architecture

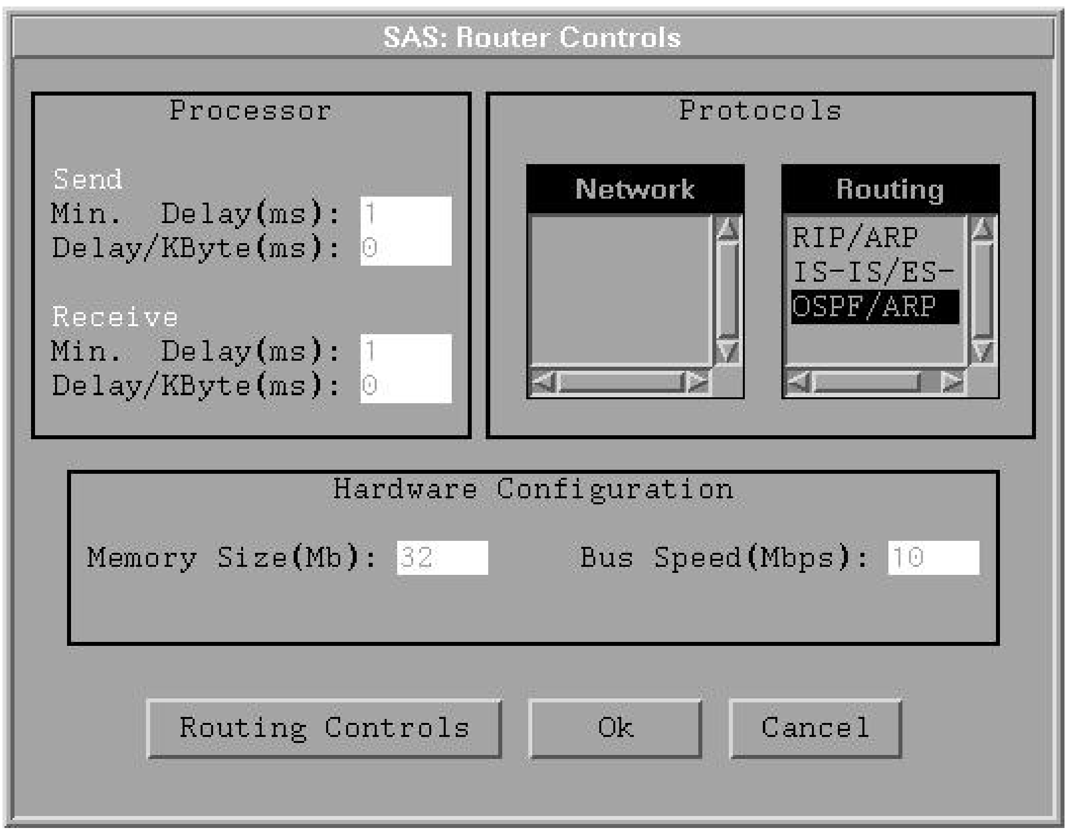

Figure 4.14. Sample Router Model Controls

As with the Bridge model, Router memory values are configurable but are not used during a simulation. You can use the Routing Protocols list box in combination with the Routing Controls button to select and tune the routing protocol specifics for a Router model. The "**??**" section of Appendix **??**, "**??**," provides details on routing protocols. Note that, while you can change the routing protocol for each individual model, every Router and Computing Device model must be running the same routing protocol for the simulation to run properly.

Building a LAN Model

You build a LAN network model as described in Chapter **??**, "**??**,"—either by dragging and dropping LAN equipment models from the LAN palette (Figure 4.15) or by selecting and positioning equipment models from the Drawing Panel pop-up menu. Remember that NIC models (within other devices) can only be connected to Transmission Media models and vice versa. And every NIC that is to participate in a simulation must have one and only one two-way arc attached to it. You must also ensure that all models on the same LAN are running the same MAC protocol and that, where applicable, all models are enabled with the same network protocol and possess valid network addresses.

Figure 4.15. LAN Palette

Running a LAN Simulation

Before you can successfully run a simulation on your LAN network model, the following criteria must be met:

- The equipment models must be connected properly.
- You must have assigned appropriate protocols and addresses to the models.
- Your network model must have traffic generators.

Traffic Generators

While the details of traffic generators are discussed in the description of the WorkStation model, netWorks also has a global parameter that affects LAN traffic generators. On the netWorks control panel there is a slider control labelled Start Delay that determines how long the individual traffic generators will wait upon receipt of the global Start command before actually invoking their data packet generation routines. The purpose of the Start Delay parameter is to let the routing protocols have time to acquire information about the network topology before data traffic commences on the

The correct bibliographic citation for this manual is as follows: SAS Institute Inc., *SAS/OR Software: The netWorks Application, Version 8*, Cary, NC: SAS Institute Inc., 1999. 89 pp.

SAS/OR Software: The netWorks Application, Version 8

ISBN 1-58025-487-X

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

1st printing, October 1999