6-3-1994

NetSim User's Manual

Lewis Barnett III

NetSim User's Manual

Lewis Barnett
University of Richmond
Department of Mathematics and Computer Science
TR-92-01

June 3, 1994

1 Introduction

NetSim is a simulation program for the study of Local Area Network (LAN) protocol performance. This document contains a description of the program, instructions for its use, details of the formats of input and data files, and information on other programs that are used cooperatively with NetSim. Less verbose directions for using this program can be found in the UNIX* man page *netsim(1)*. This manual describes version 2.4 of the program.

1.1 Purpose

NetSim is intended as a tool for investigating the effects of a wide range of parameters on the performance of the Ethernet Medium Access Control layer protocol for LANs. NetSim allows the user to investigate the effects of the packet arrival process, the size of packets transmitted, the speed of the network, and the layout of the network on performance. The number of hosts, their positions on the network, and the overall length of the network are user configurable. The simulator provides results on a number of different performance metrics, such as overall throughput, average packet delay, variance of packet delay, and number and distribution of collisions. Many of these statistics are also tracked for each simulated station. In particular, the simulator tracks network delay from the transmitting station's perspective rather than only keeping track of packet interarrival time from the network's perspective.

Many simulation studies make simplifying assumptions about network layout or behavior just as analytic models do. For example, it is not unusual for the author of a simulation of Ethernet to assume that the population of the network is infinite, and thus that transmission attempts occur from random points on the network. This makes

*UNIX is a trademark of Bell Laboratories.

it impossible to investigate the interaction of the arrival process and the network layout and ignores the effects of the Ethernet specification's rules about minimum spacing between stations. NetSim offers a very detailed simulation of Ethernet, including realistic modeling of signal propagation, the effect of the relative positions of stations on events on the network, the collision detection and handling process and the transmission deferral mechanism.

1.2 Structure

The NetSim simulator is a single process discrete event simulator. Multiple protocol modules are provided which use the same input file structure, produce output in the same formats, and are written to provide as identical a simulated operating environment for the protocols as possible. The program simulates a finite population network, which gives a more accurate picture of network behavior than simulations based on assumptions similar to those used in infinite-population modeling studies. The program does not model packet buffering at individual stations; each station is assumed to generate a new arrival only after their previous packet has been successfully transmitted. A single event queue is used for the simulation which at any given time contains one entry for each station on the network. The entry contains the simulation time of the next transmission attempt by the station and the simulation time at which the packet was originally submitted for transmission, as well as other protocol specific information.

The parameters for a given execution of NetSim are read in from an experiment description file (described in Section 3). The file consists of a list of keywords and values. The values are used to initialize the simulator's variables. When the file has been completely parsed, one or more preliminary experiments are run to determine the offered load for the experiment. This is accomplished by running an experiment with one station generating traffic according to the default station parameters for arrival rate and packet lengths. If all stations specified in the experiment description file follow the defaults, the offered load is the load generated by a single station multiplied by the number of stations specified for the experiment. If some stations have individually specified traffic processes, further offered load experiments will be run to determine the contribution to the overall load of such stations.

1.3 Features

1.3.1 Finite Population Simulation

NetSim simulates a user-specifiable finite population of stations on a network. This differs from many analytic and simulation models which rely on an infinite population model. The infinite population model makes analytic models more mathematically tractable and simulation models simpler at the expense of accuracy. For example, typical Ethernet simulations based on an infinite population model produce asymptotic delay behavior as the offered load on the network increases. A finite population simulation

produces a more accurate result, with delay leveling off as the traffic generating ability of a finite population of stations reaches its maximum.

1.3.2 Flexible Output Process Specification

The output process or sequence of interpacket times and packet lengths produced by each simulated station can be specified in several ways. These quantities can be generated randomly from traditional probability distributions such as the exponential and uniform distribution. It is also possible to specify a custom tailored discrete distribution by specifying a set of probability/packet length or probability/interpacket time pairs from which lengths or times can be randomly generated. These quantities can also be deterministic, i.e. can always be the same value. Finally, in the case of interpacket times, it is possible to simulate a continuously queued station. This type of station is always ready to transmit the next packet immediately upon completing transmission of the current one.

All stations participating in a simulation can use the same distributions for these two quantities, or individual hosts can be configured differently. For example, it would be possible to configure one station with distributions which simulate the behavior of a file server while the other stations were configured to simulate the behavior of clients. This flexibility allows a wide variety of network traffic loads to be recreated.

1.3.3 User Oriented Delay Statistics

The simulator keeps track of the queueing delay experienced by transmitted packets. This is defined as the difference between the time the packet was originally submitted for transmission and the time at which it completes successful transmission. This metric includes the time during which the packet was "queued" at the station waiting for access to the channel and thus illuminates the delay characteristics of the protocol being simulated. This quantity is more useful in determining the network response time from the point of view of the user than the network-oriented metric of overall interarrival time.

1.3.4 Automatic Offered Load Calculation

NetSim is able to automatically calculate the offered load for an experiment from the descriptions of the individual stations to be simulated. This is accomplished by automatically conducting several preliminary experiments, one for the default station distributions and one each for any hosts which use a different output process than the default. In these preliminary experiments, a single host transmits on an otherwise idle network with the default (or individual station) output process to determine the maximum offered load with the given output process.

1.3.5 Automatic Experiment Repetition

It is often desirable to repeat a simulation experiment several times starting with different random number generator seeds to diminish any effects of any particular sequence of events, then to use the averages of the collected statistics for data analysis. NetSim automatically performs this function for the experimenter.

1.3.6 Varying Parameters

Simulation studies usually involve simulating the behavior of a system a number of different times while varying a single parameter to determine the effect that the parameter has on the performance of the system. NetSim is able to run such a sequence of experiments without human intervention by the use of the *varying parameter* mechanism. The parameter to vary and the way in which to vary it are specified in the experiment description. The variation can be either by fixed sized steps up to a specified maximum value, or can be specified as an explicit sequence of values for the varying parameter.

2 Using NetSim

The command line interface to NetSim is very simple. Options on the command line have been kept to a minimum in favor of storing as much information as possible about the actual network being simulated in the experiment description file (Section 3) where it can serve as documentation on the parameters of the experiment. Therefore, the options to the NetSim command deal with input/output and with restarting experiments that are interrupted before completion. The formal specification for the NetSim command is as follows.

`netsim [-i` *input_file*`] [-o` *output_file*`] [-r` *expt_num* `] [-v] [-V]`

Square brackets indicate that the enclosed text is optional. The *-i* option allows the use to specify a filename from which the experiment description should be read. If this option is not present, NetSim takes input from the standard input. Similarly, the *-o* option specifies a file for NetSim's output. If this option is not present, output goes to the standard output. The *-v* option specifies verbose output. When this flag is present, the program produces status messages about the progress of the experiment. The *-V* option prints out the version of the program and information about the optional features which were compiled into the program.

The *-r* option allows an experiment which was interrupted to be restarted. The varying parameter mechanism allows an entire sequence of experiments to be specified by a single experiment description file. Each of these experiments writes out an output file of summary data as it completes, so experiments which are interrupted by system failures or other unforeseen circumstances need not be started again from scratch; rather, they can be restarted from the point at which they were interrupted. Each experiment

in the sequence is given an index number, which becomes part of the file name of the summary file generated by the experiment. The original value of the varying parameter has index 0, the first incremented value has index 1, and so on. It is possible to determine the index of the last completed experiment by looking at the filenames of the experiment summary files, which have the extension ".nso". *Expt_num*, specified with the *-r* option, is the index of the next varying parameter step that should be executed.

3 The Experiment Description File

The command line options to NetSim are simple; the data used to determine the nature of a simulation are stored in an *experiment description file*. This file contains all of the information necessary to carry out a simulation, including the layout of the network to be simulated and the string used as the basis for the names of the output files. The features of NetSim, including the automatically varying parameter facility, are best understood from the explanations of the various entries in the description file.

3.1 General Format

The experiment description file (a file whose name typically ends in ".edf") is loosely patterned on the structure of X Window System resource files [SG86]. Each entry in the file consists of a keyword and a value separated by a colon and at least one space or tab. (NOTE: It is **very** important that the colon be followed by at least one space or tab.) This structure has several advantages. The presence within the description file of a label for each entry means the entries are self-identifying, so a rigid arrangement or ordering is not necessary. There are a few cases where NetSim expects to see certain entries in a certain order; these are clearly explained in the next section. Further, if the names are chosen carefully, this structure allows the description file to act as documentation for the experiment. An effort has been made to choose meaningful names for the entries in the description file for this reason. This structure also makes future modifications easier; the addition or deletion of keywords can be easily handled and will not introduce incompatibilities between data files defined for different versions of the program.

The experiment description file uses the C Shell convention for indicating comments. Lines that begin with "`#`" are considered to be comments and are ignored by the program. Likewise, any characters on a line that follow a "`#`" are ignored. Blank lines are not significant, and are also ignored.

Packet length and interarrival times are generated from distributions specified in the description file. Each distribution is given a name in addition to the values that define its behavior. A list of such distributions is specified, then these names are used to choose a default distribution for packet length and interarrival time. It is also possible to configure individual stations with distributions other than the defaults so that computational activities such as file service can be modeled.

3.2 Explanation of Entries

The name of each entry is followed with an indication of whether the entry is required to appear in all experiment description files or is optional. A sample experiment description file is shown in Appendix A.

General Experiment Setup

There are several entries in the experiment description file dealing with how the specified experiment will be conducted. These entries are the character string that will be used to generate the names of various data files produced by the simulator, the number of repetitions to conduct of each experiment, and various entries dealing with the varying parameter mechanism.

Keyword: `experiment_name` (required)
Data type expected: character string
Description: This entry usually coincides with the name of the description file with the ".edf" extension stripped off. It is used to generate the names of the data files as explained in Section 4.2.

Keyword: `repetitions` (required)
Data type expected: integer
Allowed values: Must be > 0
Description: If the value is larger than one, NetSim will automatically conduct the specified number of repetitions of the experiment and present the averages of reported quantities in the output files.

Keyword: `varying_param` (optional)
Allowed values: one of `NONE`, `CABLE_LENGTH`, `BIT_TIME`, `NUM_HOSTS`, `DEF_ARR_RATE`, `DEF_MSG_SIZE`
Description: If the value is `NONE`, a single experiment run is conducted subject to the value of the `repetitions` entry, and the results are reported. However, if one of the other values is present, a series of experiments will be run, with the corresponding parameter being adjusted between runs. A summary file is produced for each value of the varying parameter. A file is also produced containing the results for the entire series of runs. If `CABLE_LENGTH`, `BIT_TIME`, or `NUM_HOSTS` is the value, the program variable associated with the respective quantity is adjusted between runs. If `DEF_ARR_RATE` is the value, then the appropriate parameter of the inter-packet arrival time distribution is adjusted between runs. Similarly, if `DEF_MSG_SIZE` is the value, the appropriate parameter of the message size distribution is adjusted. WARNING: When `CABLE_LENGTH` is the varying parameter, the locations of the stations on the cable is recomputed every time the cable length is changed. If the original experiment specified any specific positioning information for individual stations using the `host_index` and `location` keywords, this information will be ignored in all but the first experiment.
Other required entries: Either `vp_step` and `vp_max` or or one or more occurrences of `vp_pt` must be present.

Keyword: `vp_step` (optional)

Data type expected: integer or real
Allowed values: determined by the varying parameter; must be positive
Description: If the `varying_param` entry is present and is not `NONE`, `vp_step` specifies the amount by which the varying parameter is to be adjusted between runs. Ignored otherwise.
Other required entries: `vp_max`

Keyword: `vp_max` (optional unless `vp_step` is present)
Data type expected: integer or real
Allowed values: determined by the varying parameter and `vp_step`; must be greater than the initial value of the varying parameter
Description: If the `varying_param` entry is present and has a value other than `NONE`, and `vp_step` is present, `vp_max` specifies the maximum value of the varying parameter for which an experiment should be run.
Other required entries: `vp_step`

Keyword: `vp_pt` (optional)
Data type expected: integer or real
Allowed values: determined by the varying parameter
Description: If the `varying_param` entry is present and has a value other than `NONE`, the `vp_pt` entry allows the user to specify the series of values that the varying parameter should take on in successive runs. This facility is useful when not all possible values of the varying parameter are of equal interest. For example, it allows runs to be more closely spaced around the saturation point of a network to study the behavior of protocols during the transition from normal conditions to overloaded conditions.

Keyword: `num_events` (required)
Data type expected: integer
Allowed values: any integer between 1 and MAXINT
Description: Specifies the number of events to be simulated before termination. In general, successful transmissions and collisions are considered to be events for contention based protocols.

Physical characteristics of the network

Several of the entries in the file deal with the physical characteristics of the network to be simulated. The user can specify the length of the network, the number of stations or hosts on the network, the bit transmission time (and thus the data rate), the maximum allowed packet size, and the method by which hosts are positioned on the network.

Keyword: `max_cable` (optional)
Data type expected: real
Allowed values: must be positive
Description: Specifies the maximum allowed length of the cable for bus networks. In Ethernet style protocols, this quantity is used to determine the size of the backoff slot. The default units are seconds, in which case the specified value is the number of seconds

it takes for a signal to traverse the maximum cable distance in one direction (propagation delay). An optional unit specifier can be added to change the units accepted. Possible units are "ft" for specifying the cable length in feet, or "m" for specifying it in meters. All values are converted into seconds for use in the program. The unit specifier "sec" can also be used for documentation purposes, but it is not necessary for proper interpretation of the experiment description. If `max_cable` is not specified, the IEEE 802.3 figure of 25.6 microseconds is used as a default.

Keyword: `cable_length` (optional)
Data type expected: real
Allowed values: Any real greater than 0.0 and less than `max_cable`
Description: Specifies the actual length of the simulated network, and is used in determining propagation time for signals. Default units are seconds, and the same unit specifiers described for `max_cable` are available. If `cable_length` is not specified, the default value is 25.6 microseconds, the IEEE 802.3 default maximum network length.

Keyword: `num_hosts` (required)
Data type expected: integer
Allowed values: $1 \leq$ `num_hosts` ≤ 1000
Description: Number of stations on the simulated network.

Keyword: `bit_time` (optional)
Data type expected: real
Allowed values: any real number > 0.0
Description: Number of seconds it takes to transmit a bit. This quantity allows the data rate to be adjusted. Default is 100 nanoseconds, corresponding to a data rate of 10 Mbps.

Keyword: `max_pkt` (optional)
Data type expected: real
Allowed values: Any value greater than $2\times$ `max_cable`
Description: The maximum sized packet allowed. Default units are seconds, in which case the values specifies the amount of time it takes to transmit the largest allowed packet. Optional units specifiers "bytes" and "bits" allows the maximum packet to be specified in those quantities. If `max_pkt` does not appear, the IEEE 802.3 default of 1526 bytes is used.

Keyword: `host_loc_spec` (required)
Allowed values: One of `H_REGULAR`, `H_UNIFORM`, `H_USERDEF`
Description: Describes how stations are located on the network. If the value is `H_REGULAR`, stations are evenly spaced on the cable. If the value is `H_UNIFORM`, stations are placed randomly on the cable according to a uniform distribution. If the value

is H_USERDEF, the locations of stations are specified explicitly using the location keyword in individual station descriptions. (See below.)

Output process specification

The output process for a simulation consists, at the minimum, of definitions for a distribution for packet lengths to be generated and a distribution for generating the interpacket delays. At least these two distributions must be specified in the experiment description file, and they must be associated by name with the default packet length distribution and the default interpacket distribution. If no other provisions are made, all stations will use these distributions to generate traffic, resulting in a homogeneous traffic generation process. Other distributions may also be specified in the file, and these distributions can be associated with the output processes of specific stations, as described below. This allows the inclusion of stations with differing transmission behaviors to be included on the same simulated network.

Keyword: dist_name (Required)
Data type expected: character string
Description: The name of a probability distribution to be used for generating either packet lengths or interarrival times. The name will be later referenced to make the associated distribution the default for lengths or interarrival times, or to associate the distribution with the lengths or interarrival times of a specific station. All other keyword/value pairs associated with the distribution must appear before another instance of the dist_name keyword.
Other required entries: dist_type, parameter1

Keyword: dist_type (one required for each dist_name)
Allowed values: One of EXPON, UNIFORM, FIXED, DISCRETE, CONTQUEUE
Description: If the value is EXPON the distribution is exponential with mean given by the value of parameter1. If the value is UNIFORM the distribution is uniform on the interval from the value of parameter1 to the value of parameter2, inclusive. If the value is FIXED, the distribution is deterministic with the value of parameter1 as the constant value of the distribution.[†] If the value is DISCRETE the value of the distribution is determined from a set of (value, probability) pairs specified with the x_val and y_val keywords. If the value is CONTQUEUED (valid only for distributions that will be used for generation of interarrival times) the generated interarrival time will always be 0, resulting in a station that continuously has packets to send.
Other required entries: For EXPON and FIXED, parameter1 is required. For UNIFORM, parameter1 and parameter2 are required. For DISCRETE, point_cnt and at least one pair of x_val and y_val keywords are required.

Keyword: parameter1 (one required for each dist_name)

[†]To be absolutely clear: a FIXED distribution always returns the same value, and this value is the value of the keyword parameter1 associated with the distribution.

Data type expected: real
Allowed values: For distributions to be used as packet lengths, the value must be less than the maximum allowed packet length. For Interarrival time distributions, the only requirement is that the value be larger than 0.
Description: For distributions with type `EXPON`, `parameter1` is the mean of the distribution. For distributions with type `UNIFORM`, `parameter1` is the lesser endpoint of the interval from which the distribution is drawn. For distributions with type `FIXED`, `parameter1` is the fixed value of the distribution. For other types, `parameter1` is ignored. If the distribution with which `parameter1` is associated is used to generate packet lengths, the default units are seconds. Other available units specifiers are "bits" and "bytes." If the distribution is used to generate interpacket intervals, the units are packets per second.

Keyword: `parameter2` (required for any UNIFORM distribution)
Data type expected: real
Allowed values: Must be larger than the value of `parameter1`. For distributions to be used as packet lengths, the value must be less than the maximum allowed packet length.
Description: Presently, `parameter2` is used only as the greater endpoint of the interval from which the uniform distribution is drawn.

Keyword: `x_val` (optional)
Data type expected: real
Allowed values: For distributions used for packet length, the value must be less than the maximum allowed packet size. For distributions used for interpacket times, the value must be larger than zero.
Description: For discrete distributions, a list of {`x_val`, `y_val`} pairs appear in the description file. The `x_val` values are the values that will be returned by the the distribution for either packet lengths or interpacket times. Note that for discrete distributions which will be used as interpacket time distributions, the x values should be the actual length of the interval to be generated in seconds, **NOT** the rate of message generation in messages per second!
Other required entries: Each `x_val` must be followed by a `y_val`.

Keyword: `y_val` (optional)
Data type expected: real
Allowed values: Must be between 0.0 and 1.0.
Description: Netsim expects the points associated with a discrete distribution to form a cumulative distribution function. The `y_val` associated with an `x_val` should therefore be the sum of the `y_val` associated with the preceding `x_val` and the probability that the current `x_val` value will be generated. The `y_val` value for the last point in the distribution must be 1.0.
Other required entries: Should be directly preceded by the corresponding `x_val`.

Example: To specify a packet length distribution which generates 25% 76 bytes packets, 60% 100 bytes packets, and 15% 1500 byte packets, the following lines should appear in the experiment description file:

```
dist_name:      discrete_length
dist_type:      DISCRETE
point_cnt:      3
x_val:          76 bytes
y_val:          0.25
x_val:          100 bytes
y_val:          0.85
x_val:          1500 bytes
y_val:          1.0
```

Keyword: `def_interpkt_dist` (required)
Data type expected: character string
Allowed values: Must be the name of a previously defined distribution.
Description: The name of the default interpacket arrival time distribution for the experiment. Each station uses this distribution to generate interpacket times unless a station specific distribution is chosen.

Keyword: `def_length_dist` (required)
Data type expected: character string
Allowed values: Must be the name of a previously defined distribution.
Description: The name of the default packet length distribution for the experiment. Each station uses this distribution to generate packet lengths unless a station specific distribution is chosen.

Individual host specifications

In the absence of instructions to the contrary, the placement and behavior of all hosts on the network are determined by the value of `host_loc_spec` and the default packet length and interpacket distributions. If a different behavior or placement is desired for individual stations, the `host_index` keyword in combination with one or more of the other associated keywords can be used to change the characteristics for the stations.

Keyword: `host_index` (optional)
Data type expected: integer
Allowed values: $0 \leq$ `host_index` $\leq$ `num_hosts`
Description: The index of one of the simulated stations. This entry indicates the beginning of a description of a particular station on the network which will differ in some way from stations following the experiment defaults.

Keyword: `location` (optional)
Data type expected: real

Allowed values: $0 \leq$ `location` $\leq$ `cable_length`
Description: `Location` allows stations to be explicitly arranged on the network. The associated value overrides the station location which would have been generated from the `host_loc_spec` value for the experiment. Associated with the station specified by the most recent `host_index` value. Note that it is important that the location value be such that all stations maintain their relative ordering.

Keyword: `length_dist` (optional)
Data type expected: character string
Allowed values: The names of previously defined distributions.
Description: Specifies the packet length distribution that will be used by a particular station. Associated with the station specified by the most recent `host_index` value.

Keyword: `ipi_dist` (optional)
Data type expected: character string
Allowed values: The names of previously defined distributions.
Description: Specifies the interpacket interval distribution that will be used by a particular station. Associated with the station specified by the most recent `host_index` value.

4 The Data Files

NetSim produces several data files. In the case of an experiment with no varying parameter, an overall summary file (with extension ".nso"), a delay histogram file (with extension ".dhg"), and a backoff histogram file (with extension ".bhg") are produced. In the case of an experiment run using the varying parameter mechanism, these three files are produced for the experiments run with each value of the varying parameter, with a summary of the results of the entire series of experiments appearing in a series data file (with extension ".nss"). These files are discussed at some length in the following sections. A utility program, ns_graph, has been provided to combine data abstracted from multiple series files into a form which is readable by the publicly distributed graphing program for the X Window System, xgraph. (See Sections 5.1 and 5.2.)

4.1 The Series Data File (.nss)

The series data file holds a summary of an entire series of experiments run using the varying parameter facility of NetSim. Each line is a summary of the statistics for a single run or a set of repetitions run with the same parameters. The first line of the file is a descriptive header containing one entry per field of the subsequent files. The entries are separated by tabs. A description of the quantities found on each line follows.

- Offered load
  The sum of the traffic load produced by each host specified in the experiment description file transmitting on an otherwise idle network.

- Throughput
  The ratio of time spent in actual transmission to the total time elapsed.

- Actual data rate
  The actual data rate achieved on average during the experiment. This quantity is calculated by multiplying the throughput by the maximum data rate for the network.

- Average delay
  The average queueing delay (as defined in Section 1.3.3) experienced by packets transmitted during the experiment. This quantity is calculated by dividing the total delay for the experiment by the number of successful packet transmissions.

- Delay variance
  The variance of the average queueing delay, calculated from the delay histogram.

- Arrival rate
  The average number of packets transmitted per second over the duration of the experiment.

- Message size
  The average size of messages transmitted during the experiment, in bytes. Calculated by dividing the total transmission time for the experiment by the number of successful transmissions (giving) the average transmission time per packet) then converting to bytes.

- Minimum packet size
  The minimum packet size for the experiment. Calculated from the maximum cable length.

- Maximum packet size
  The maximum packet size for the experiment. User specified.

- Station count
  The number of stations (transmission sources) on the simulated network.

- Cable length
  The actual length of the cable to which the stations were attached in the experiment. Used in calculating signal propagation times.

- Maximum cable length
  The maximum allowed length of the cable for systems of the type being simulated. Used to calculate the minimum packet size and backoff slot size for Ethernet, for example.

- Maximum Data Rate
  The capacity of the channel being simulated. Calculated from the user-specified bit time.

- Total collisions
  The total number of collisions as observed from the network. Each such collision would have involved two or more hosts, so this number will always be less than the sum of collisions from the host statistics from the corresponding ".nso" file.

- Collisions per transmission
  The average number of collisions that a packet suffered before it was successfully transmitted.

- Collision rate
  The average number of collisions per second.

- Total packets
  The total number of packets which were successfully transmitted over the duration of the experiment.

The file is set up to be processed by `ns_graph`, a data analysis utility described in Section 5.1. In addition to the statistics collected for each run, the values of all of the parameters that can vary from one run to the next by the varying parameter mechanism are included. This file provides a succinct summary of the results of an entire sequence of experiments.

4.2 The Experiment Summary File (.nso)

An experiment summary file is created for each experiment of a series. A run may consist of several repetitions with the same parameters, in which case the quantities reported in the summary file are the averages over the repetitions. The name of the summary file is generated from the name of the experiment, a two letter code indicating which parameter varied in the series of experiments that the file belongs to, and a three digit index indicating which run of the series the file summarizes. If the experiment had no varying parameter, then only a single file exists whose name is composed of the experiment name plus the ".nso" suffix.

The heading of the file contains the name of the series of experiments to which it belongs (the value of the `expt_name` keyword), the version of NetSim which produced the file, the name of the account from which the experiment was run, the hostname of the computer on which the experiment was run, and the date and time at which the file was written.

The next section of the file contains a detailed listing of the parameters of the experiment. This listing includes:

- a description of the parameter that is being varied in the series of experiments, if any;
- the maximum propagation delay of the simulated network in seconds and meters;
- the actual cable length of the simulated network;
- the bit time and corresponding data rate for the network;
- the backoff slot size;
- the minimum packet length in seconds and bytes;
- the number of stations and how they are placed on the network;
- the mean packet length or interval from which lengths are chosen, if applicable;
- the type of distribution used to generate the packet lengths;
- the mean arrival rate or interval from which arrival rates are chosen, if applicable; and
- the type of distribution used to generate the arrivals.

A listing of the statistics collected during the run follows. For details on the calculation of some of the statistics, see Section 4.1 Included are:

- the duration in simulation time of the run;
- the offered load for the experiment as a percentage of the capacity of the channel;
- the throughput achieved over the duration of the experiment as a percentage of the capacity of the channel;
- the average delay experienced by packets in seconds;
- the variance of delay experienced by packets;
- the arrival rate in packets per second;
- the average packet size in seconds and bytes;
- the total number of collisions experienced during the experiment;
- the average number of collisions experienced by packets before successful transmission;
- the overall collision rate in collisions per second; and
- the total number of packets sent during the experiment.

A histogram of the number of packets experiencing a given number of collisions before transmission is given, as well as the number of packets whose delays were too great to be properly entered in the packet delay histogram and the size of the longest delay which could not be entered in the histogram (see Section 4.3).

The last item in the summary file is a listing of several statistics for each station in the network. Included for each station are:

- the number of packets sent by the station;
- the number of collisions experienced by the station;
- the average delay experienced by packets sent by the station;
- the throughput achieved by the station; and
- the load offered by the station.

Note that with the exception of the number of collisions, the sum of the figures for individual stations should be very close to the overall figures in each category. The agreement may not be exact in the case of experiments where more than one repetition was run, since the numbers are averages in this case. The sum of collisions experienced by stations will always be larger than the number of collisions observed by the network since each collision involves more than one station.

4.3 The Delay Histogram File (.dhg)

This file contains a histogram of the queueing delay suffered by packets during the experiment run. The bins for the delay histogram are 200 microseconds[‡]. Each line contains a bin number and the number of packets whose delay fell within the range of that bin. The array which holds the histogram in NetSim has 25000 elements, allowing the program to keep track of delays up to 5 seconds. Only bins from 0 up to the largest recorded delay are included in the file.

4.4 The Backoff Histogram File (.bhg)

This file contains a histogram of the number of packets that suffered a given number of collisions before they were successfully transmitted. This histogram is collected over all of the stations participating in the experiment, so the sum of collision from the bins of the histogram will be larger than the number of collisions observed from the network as reported in the series data file and the experiment summary file.

[‡]The bin size was chosen mainly for historical reasons; it was the resolution of time keeping modifications made to the Unix kernel in the Local Area Network Testbed project, in conjunction with which the precursor of this simulator was written.

5 Related Utilities

Several programs are used to process and display data generated by the simulator. Ns_graph is an interactive program which allows the user to choose quantities from one or more series data files and produces output suitable for a plotting package to display. The xgraph package or the ACE/gr package take these files and display the plots on an X display.

5.1 The ns_graph program

Ns_graph transforms the series data file (.nss extension) into a format which can be plotted using either the xgraph program (see Section 5.2) or the ACE/gr package (see Section 5.3). The series data file contains a line for the results of experiments run with each value of the varying parameter. Each of these lines consists of the average values for all of the statistics collected by the program, separated by tabs. Ns_graph allows the user to interactively choose which quantities to plot on the x and y axis of a two dimensional graph, and to specify a name for each data set. Multiple series files can be plotted on the same graph. For further information, see *ns_graph(1)*.

5.2 The xgraph program

Xgraph is an X11 based plotting program with many nice features, including the ability to zoom in on portions of a graph and facilities to produce laser printer output of graphs or data files that can be included as figures in LaTeX documents. It was written by David Harrison of the UC-Berkeley Electronics Research Lab. Data produced by NetSim can be selectively converted to the format read by xgraph using the ns_graph program. For further details, see *xgraph(1)*.

5.3 The ACE/gr package

The ACE/gr package is another X Window System plotting package available in Xview and Motif versions. It performs the same task as xgraph, but allows more interactive manipulation of the plots. It was written by Paul Turner of the Oregon Graduate Institute. For further details, see *xvgr(1)* or *xmgr(1)*.

A Appendix: Sample experiment description file

```
# Ethernet, hosts randomly placed, user defined packet length distribution,
# exponentially distributed arrivals.  Default arrival rate increases
# from one experiment to the next.

experiment_name: enhr64mlmei

repetitions: 3
varying_param: DEF_ARR_RATE
vp_step:
vp_max:
vp_pt: 5
vp_pt: 10
vp_pt: 20
vp_pt: 25
vp_pt: 30
vp_pt: 40
vp_pt: 50
vp_pt: 60
vp_pt: 75
vp_pt: 85
vp_pt: 100
vp_pt: 125
vp_pt: 150
vp_pt: 200
vp_pt: 250

# Network characteristics
num_events: 1000000
max_cable:
cable_length:
num_hosts: 64
bit_time:
max_pkt:
host_loc_spec: H_UNIFORM

# Descriptions of probability distributions to be used in the experiment.
dist_name: def-ipi
dist_type: EXPON
parameter1: 1

# A user defined distribution based on observed network traffic is used
# to generate the packet lengths.  40% of the packets are 76 bytes in
# length, 39% are 180 bytes, 3% are 975 bytes, and 18% are 1526 bytes.
dist_name: def-len
dist_type: DISCRETE
point_cnt: 4
x_val: 76 bytes
y_val: .4
```

```
x_val: 180 bytes
y_val: .79
x_val: 975 bytes
y_val: .82
x_val: 1526 bytes
y_val: 1.0

# Default distributions
def_interpkt_dist: def-ipi
def_length_dist: def-len

# Specific host descriptions:  these keywords are repeated for each
# host whose behavior should be different from the default in some way
host_index:
location:
length_dist:
ipi_dist:
```

References

[MB76] Robert M. Metcalfe and David Boggs. Ethernet: Distributed packet switching for local networks. *Communications of the ACM*, 19(7), July 1976.

[SG86] Robert Scheifler and Jim Gettys. The X Window System. *ACM Transactions on Graphics*, 5(2):79 – 109, April 1986.