6-29-1992

# NetSim: A Network Performance Simulator

Lewis Barnett III
*University of Richmond*, lbarnett@richmond.edu

## Recommended Citation

Lewis Barnett. *NetSim: A Network Performance Simulator*. Technical paper (TR-92-2). *Math and Computer Science Technical Report Series*. Richmond, Virginia: Department of Mathematics and Computer Science, University of Richmond, June, 1992.

# Netsim: A Network Performance Simulator

B. Lewis Barnett, III
Department of Mathematics and Computer Science
University of Richmond
Richmond, VA 23233
barnett@armadillo.urich.edu

June 29, 1992

**Abstract**

The performance of computer communication networks is often given only cursory treatment in undergraduate Networking or Computer Communication courses. A simulation package for the investigation of many aspects of Local Area Network (LAN) performance is introduced. Its use as a tool for allowing undergraduate Networking students to investigate the performance of a popular networking system is discussed.

## 1 Introduction

One of the difficult aspects of designing an undergraduate course in Computer Networking revolves around the role that performance analysis/ evaluation should play in such a course. Recent textbooks, for example [Wal91], indicate a trend toward greater integration of performance analysis into such courses. However, the difficulties of including a significant performance analysis component in such a course, particularly when it is not feasible to require a probability course as a prerequisite, have recently been noted in [Lan92] and [NY92]. There are essentially three approaches to increasing the emphasis on performance in an undergraduate networking course. First, standard analytic techniques can be introduced and used. Second, the students can be allowed to experiment with the behavior of real networks. Third, simulation can be used to provide experience with the behavior of networks. Each of these approaches has benefits and drawbacks. The first approach provides students with a powerful methodology for evaluating network performance which may be useful, depending on the depth of the treatment, for analyzing network systems other than those presented in the course. However, if it is not feasible to require that the students be well versed in probability before entering the course, the preparatory work necessary to bring them up to speed would cause the emphasis of the course to shift significantly away from Networking and toward performance analysis. The second approach has the great benefit of providing the students with "hands-on" experience with networks without the sometimes misleading assumptions about network attributes and traffic patterns that are necessary for much of analytic performance modeling. However, the cost of providing dedicated experimental networks on the one hand and the impact on the user community of non-dedicated networks where such experiments are carried out on the other hand make this approach unworkable. The third alternative, using network simulation packages, has several advantages. A well-constructed simulator can avoid the limiting assumptions needed by many analytic models, does not require expensive, dedicated hardware, and does not adversely affect the user community at large. It may also be possible to give students experience with a wider range of network systems through simulation than would be possible with actual hardware; most campuses have representatives of only one or two different types of networking systems.

Here we present a description of a performance simulation package for the popular Ethernet* LAN protocol which can be used to give students a good feel for the performance characteristics of a widely deployed networking system without requiring them to understand analytic models of the protocol. Though many universities and colleges have Ethernet or other LAN systems installed on their campuses, it is seldom possible to dedicate a network for actual experimentation; therefore, simulators are necessary to provide the students with experience in network performance issues. The Netsim system is highly flexible so that a wide variety of network configurations can be investigated. This flexibility makes Netsim an ideal tool for open-ended investigations of the network by students as well as for more focused exercises where a particular configuration is provided for the students. The output of the package is suitable for display using a widely available X Window System based plotting tool and could be adapted to work with other data analysis packages with some small effort.

## 2 Simulator Design

The Netsim simulator is a single process discrete event simulator. 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. Upon each attempted transmission, the queue is scanned to determine which, if any, of the pending transmission attempts will cause collisions or be deferred as a result of the current attempt, and the queue entries are adjusted accordingly. The statistics collected by the simulator are discussed in Section 3.3.

*Ethernet is a trademark of the Xerox Corporation

# 3 Important Features

One of the goals of Netsim is to allow the experimenter a great deal of flexibility in specifying the network and the traffic that will be generated. The package was also written to model the behavior of Ethernet as accurately as possible. Lastly, the package was constructed with features that make running sequences of related experiments easier. The following sections discuss many of the features of Netsim which address these aims.

## 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 the assumption that the population of stations on the network is infinite. The infinite population assumption makes analytic models more mathematically tractable and simulation models simpler, but often 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.

## 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 chosen. 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 stations 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.

## 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.

### 3.4 Automatic offered load calculation

Performance studies typically gauge the efficiency of a network protocol by comparing the achieved throughput to the offered load for a given traffic level. 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 stations which use a different output process than the default. In these preliminary experiments, a single station 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.

### 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 the effects of any particular sequence of events, then to use the averages of the collected statistics for data analysis. Netsim allows the experimenter to specify the number of times each experiment should be run and automatically calculates the average of each collected quantity.

### 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.

### 3.7 Output formatting

The data collected by the simulator is saved in two different formats. Data for individual experiments is saved in a file in "human readable" format to allow the experimenter to examine the raw data. For sequences of related experments generated using the varying parameter mechanism, a summary of the overall results of the experiments is kept in a separate file. From this file, quantities can be extracted for plotting using the `ns_graph` utility, discussed in Section 7.1.

### 3.8 Restarting interrupted experiments

A command line 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 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.

## 4 Simulator Operation

Running a Netsim experiment consists of three phases: configuration, offered load calculation, and the simulation itself. Currently configuration consists of creating an experiment description file containing the desired values for various simulator attributes, or editing an existing file which specifies a network similar to the one you wish to simulate. (A window based experiment configuration application is in the works.) Experiment description files are described in more detail in Section 5. After the contents of the experiment description file have been read in and used to initialize the simulators variables, 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.

## 5 Configuring Experiments

The data used to determine the nature of a Netsim 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, the traffic generation behavior of the stations on the network, information about repetitions of each run and about the sequence of parameters to be used, and even 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 by considering how the simulator uses this description file.

### 5.1 General format of experiment description files

The experiment description file 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. 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. Further, carefully chosen names for the entries allow the description file to act as documentation for the experiment. A more detailed description of the format of the experiment description file is included in [Bar91].

The experiment description file consists of entries that specify the physical characteristics of the network (e.g. length, data rate, maximum allowed packet size, number of attached stations, etc.) and the traffic generation behavior of the attached stations. 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. These names are used to choose a default distribution for packet length and interarrival time for the experiment as a whole. 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.

## 5.2 Configurable quantities

Netsim's main strength is its flexibility in terms of the aspects of network operation which are under user control. This section discusses some of the important quantities which can be adjusted in the experiment description file.

### General Experiment Setup

Entries are provided to allow the specification of the length of the experiment and the number of times each experiment should be run, and to set up a series of experiments using the varying parameter mechanism. A sequence of experiments is specified by choosing a single network characteristic and specifying how it should be altered from one experiment in the series to the next. The user may choose the length of the network, the data rate, the number of attached stations, the default transmission rate of messages, or the default message size as the varying parameter for a series of experiments. The chosen parameter can be increased or decreased by a fixed amount for each experiment, or a set of explicit values for the parameter can be specified.

### 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 stations on the network, the bit transmission time (and thus the data rate), the maximum allowed packet size, and the method by which stations are positioned on the network. For specifying lengths or positions, units of feet, meters, and seconds are understood. There is an upper limit of 1000 attached stations, as mandated by the Ethernet 2.0 specification [Dig82] and the IEEE 802.3 standard [Ins85]. Stations can be positioned on the network in three ways: they can be placed at regular intervals over the length of the network; they can be placed at random locations drawn from a uniform distribution; or they can be placed at individually specified locations (see **Individual station customization**, 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 for the experiment. If no other provisions are made, all stations will use these distributions, 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.

There are five types of distributions available. The first two are the standard exponential and uniform probability distributions. In the case of the former, the user specifies the mean of the distribution, and in the latter, the endpoints of the interval from which the distribution should be drawn are specified. A distribution may also be deterministic, i.e. the same value is generated every time; or (in the case of a distribution meant to be used for inter-packet times) continuously queued, in which case the value generated for the delay until the next packet is sent is always zero. Finally, a user specified discrete distribution is available where the user specifies a set of values and the probability with which each value should be generated. This final possibility is useful for specifying distributions which do not conform to a well-behaved statistical distribution, such as the well documented bimodal distribution of packets lengths observed on production Ethernet networks.

### Individual station customization

In the absence of instructions to the contrary, the placement and behavior of all stations on the network are determined by the default station placement policy and the default packet length and interpacket distributions. However, Netsim allows individual hosts to be customized. Entries are provided in the experiment description file for the location, interpacket time distribution, and the packet length distribution of individual stations. So, for example, to simulate a network of workstations served by one or more file server, the default values for interpacket time and packet length could specify typical client request patterns, while server-like distributions could be associated with one or more selected hosts in the simulated network.

## 6 The Data Files

Netsim produces several data files. In the case of an experiment with no varying parameter, an overall summary file, a file containing a histogram of packet queueing delays, and a file containing a histogram of the number of collisions suffered by packets 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. The following quantities are present in the summary and series files:

- Offered load
  The sum of the traffic load produced by each station 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 6) 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 stations, so this number will always be less than the sum of collisions from the station 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.

In addition, the summary file for individual experiments in a series contains information about the experiment setup and statistics on the performance of each station in the network.

# 7 Related Utilities

Two 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 takes these files and displays the plots on an X display.

## 7.1 The ns_graph program

Ns_graph transforms the series data file into a format which can be plotted using the xgraph program (see Section 7.2). 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.

## 7.2 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, originally 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.

# 8 Pedagogical uses

Netsim lends itself to two basic types of assignments, an open-ended "project"-type assignment or a more structured or focused lab or homework-type assignment. In the project type assignment, the students might be given a topic (or allowed to choose one of their own) and told to use the simulator to investigate the topic. A typical topic might be the delay behavior of Ethernet when all packets are under a certain size. The student would be expected to construct a network and series of experiments which would demonstrate the behavior of Ethernet in the suggested situation. For a homework-type assignment, the instructor might supply the students with an experiment description and ask the students to describe the behavior of the specified network when some network parameter was altered.

## 9 Summary

### Acknowledgements

Thanks to David Harrison, currently of BioCad Corporation (david@biocad.com) for writing xgraph. For those interested in the program, it is available by anonymous ftp from `ic.berkeley.edu` in the `pub` directory.

## References

[Bar91] B. Lewis Barnett, III. *NetSim User's Manual.* Department of Mathematics and Computer Science, University of Richmond, October 1991.

[Dig82] Digital Equipment Corporation, Maynard, MA; Intel Corporation, Santa Clara, CA; Xerox Corporation, Stamford, CT. *The Ethernet: A Local Area Network, Data Link Layer and Physical Layer Specifications, Version 2.0*, November 1982.

[Ins85] Institute of Electrical and Electronics Engineers. *Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications, American National Standard ANSI/IEEE Std. 802.3–1985*, 1985.

[Lan92] Lawrence H. Landweber. Computer Networking Courses at the University of Wisconsin-Madison. *Data Communications Review*, 22(1):52 – 61, January 1992.

[NY92] Thomas Narten and Raj Yavatkar. An Introductory Course in Computer Communication and Networks. *Data Communications Review*, 22(1):52 – 61, January 1992.

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

[Wal91] Jean Walrand. *Communication Networks: A First Course.* Richard D. Irwin and Aksen Associates, 1991.