Workload Generation for *ns* Simulations of Wide Area Networks and the Internet *

M. Yuksel[†], B. Sikdar[‡], K. S. Vastola[‡] and B. Szymanski[†]
[†]Department of Computer Science
[‡]Department of Electrical, Computer and System Engineering
Rensselaer Polytechnic Institute
Troy, NY 12180 USA
{yuksem,bsikdar,vastola}@networks.ecse.rpi.edu, szymansk@cs.rpi.edu

Abstract

Extreme complexity of wide area networks and the Internet make the development of analytic models very difficult and under such circumstances, simulation models are a viable alternative to understand the behavior of these networks. We use *ns* (ns 1997) as the simulation platform for this paper largely because of its wide acceptance in the networking community and its open design suitable for modification. A major drawback of *ns* is its failure to generate workloads and traffic patterns for networks which require taking into account the temporal and spatial correlation between the sources and the traffic that they generate. Also, a realistic traffic generator must maintain the proper composition of the composite traffic which results from the contributions of various protocols and applications in the network. In this paper we propose methodologies to address these issues and describe their implementation in *ns*. In addition, to address the issue of generating long-range dependent traffic in *ns*, we have implemented two self-similar traffic sources based on Fractional Renewal Processes and Markov Modulated Poisson Process. We also validate the proposed techniques and traffic generators using extensive simulations and present the simulation results.

1 Introduction

Over the last decade, considerable effort has been made to understand and characterize the behavior of wide area networks and the Internet. The extreme complexity of such large network topologies and their traffic characteristics coupled with the effects of adaptive congestion control make the development of analytic models difficult. Under such conditions, simulations are the most promising tools for understanding the behavior of these networks.

Simulating how wide area networks and the Internet behave is complicated by the heterogeneity of these networks and their fast pace of evolution. The interaction between the traffic from the diverse suite of protocols that operate over the Internet and the hierarchical nature of the topology are a few of the factors contributing to the complexity of such large networks (Paxson and Floyd 1997). Additionally, the simulators need to account for the temporal correlation between the hosts and the network and the spatial correlation amongst the traffic generated by different sessions on a link.

In this paper we focus on these issues and describe our approach for addressing them. We also introduce methodologies for implementing realistic workload generators for wide area networks which (1) maintain the proper composition of the aggregate traffic resulting from the mix of various applications supported by the network and (2) are capable of generating long range dependent or self-similar traffic. This work is part of a larger project involving online collaborative simulation for network management and control sponsored by DARPA (Vastola, Szymanski, and Kalyanaraman 1998).

For this project, we have chosen the network simulator *ns* (ns 1997) developed by UCB/LBNL and the VINT project as the simulation platform. Apart from its wide acceptance as a simulation tool in the networking community, *ns* has the additional advantage of being easy to modify. Also, *ns* comes with its library of network topology and traffic generators along with visualization tools like the network animator *nam*. However, the intricacies of the dynamics of wide area networks make the present capabilities of *ns* insufficient to accurately simulate their behavior. In this paper we address these drawbacks and implement our solutions for workload generation problem of wide area networks in *ns*.

The rest of the paper is organized as follows. In Section 2 we present the details of some of the issues that make simulating the Internet such a difficult task. Section 3 presents our approach for addressing some of these is-

*This work supported by DARPA under contract number F19628-98-C-0057.

sues while Section 4 shows some of the simulations results to validate the implementation of these approaches. Finally in Section 5 we present the concluding remarks and a discussion of the results.

2 Simulating Wide Area Networks: Issues

There are several key features of the Internet that make it extremely hard to characterize and consequently to simulate. The first feature, which is actually one of the factors for the Internet's immense success, is the large number of different protocols and policies operating within the Internet. Even a single protocol, TCP for example, may have different implementations with significantly different properties. We now give a brief description of some issues which must be addressed in order to accurately model the Internet (Paxson and Floyd 1997).

- **Traffic Composition and Protocol Differences :** One of the main factors contributing to the success of the Internet is the large number of protocols that it supports. While the support for these protocols allows the use of a number of applications, the complexity arising from the interplay of their dynamics is very difficult to model. Simulation scenarios should account for the mix of protocols and the resulting traffic streams and the complex interactions amongst them. Recent studies in (Claffy, Miller, and Thompson 1998; Thompson and Miller 1997) have given an indication of the traffic composition in the backbone of large networks. Maintaining the proper mix of traffic composition as well as the applications to simulate these protocols is essential to capturing the behavior of wide area networks. Details of these issues and our approach to deal with them are discussed in Sections 3.1 and 3.2.

- **Traffic Generation :** Traffic generation is a basic problem which must be dealt with in wide area network and Internet simulations. Though trace driven simulations might appear to be an all-encompassing solution, the authors in (Paxson and Floyd 1997) argue against such an approach as it is unable to account for adaptive congestion control and they suggest a source based approach. Another important issue with traffic generation is the overwhelming proof showing self-similarity in wide area and Ethernet traffic (Leland, Taqqu, Willinger, and Wilson 1994; Paxson and Floyd 1995). Generation of aggregate traffic which is self-similar in nature is of utmost importance in simulation scenarios as Poisson models grossly underestimate the queuing delays and overflow probabilities (Paxson and Floyd 1995). We discuss the implementation of long range dependent traffic sources in Section 3.2.2.

- **Topology Issues :** Wide area networks and the Internet may be viewed as a collection of interconnected domains controlled by diverse organizations, each with its own internal topology design. Also, such large networks have a large variation in their link bandwidths and experience dynamic routing where routes can change on time scales of seconds to days (Paxson 1996). As graph based models (Calvert, Doar, and Zegura 1997) to characterize such topologies and realistic topology generators are already available in *ns* for a variety of scenarios, we do not concentrate on this aspect of wide area network simulation.

3 Workload Generation for Wide Area Networks

In Section 2, we discussed some of the issues which make simulating large wide area networks and the Internet a difficult task. In this section, we outline our approach towards these issues and discuss their implementation in *ns*. We first deal with the aspects of session generation and achieving the desired traffic composition and protocol mix. We also look at randomizing source and destination pairs and maintaining the temporal and spatial correlation between the sources and the traffic they generate. We then describe our implementation of two self-similar traffic generators based on Fractional Renewal Processes (Ryu 1996) and Markov Modulated Poisson Processes (Andersen and Nielsen 1998) and discuss their applicability to simulate long range dependent traffic.

3.1 Traffic Composition

In Section 2, we discussed the implications of the interaction of different protocols in a wide area network. Also, to accurately simulate such large networks it is important to maintain the proper breakup in the composite traffic which arises from the contribution of each of these protocols. The composition of the traffic in the NSFNET Internet backbone in terms of packet counts in 1995 was : Other (27 %), WWW (21 %), FTP-data (14 %), NNTP (8 %), Telnet (8 %), SMTP (6 %), IP (6 %), Domain (5 %), IRC (2 %), Gopher (2 %) and FTP-control (1 %). Since then, the emergence of commercial carrier administered backbones, the competitive environment has rendered data collection and their publication more difficult. In (Apisdorf, Claffy, Thompson, and Wilder 1997), (Claffy, Miller, and Thompson 1998) and (Thompson and Miller 1997) the authors present the traffic breakup on the InternetMCI backbone and its growth trends using the OC3MON traffic monitor. We use the data from these studies as the default values for our simulations. We now give a detailed description of our implementation of statistical breakup of the various applications and protocols.

3.1.1 Session Generation

To maintain the desired composition of the traffic in network simulations using *ns*, we exploit the opportunity of creating TCP and UDP agents independently in *ns*. We introduce a parameter `PER_TCP` defined as the percentage of TCP traffic in the total traffic with the assumption that the rest of the traffic is generated by UDP agents. To statistically characterize the sessions generated by each application, we define three parameters : mean number of sessions (`MNS`), mean inter-arrival times of sessions (`MIATS`), and mean duration times of sessions (`MDTS`). `MNS` defines the average number of sessions in progress for an application at any time during the simulation while `MIATS` and `MDTS` are used for randomizing the on-off times of the sessions. Other than `MNS`, `MIATS`, and `MDTS`, that are specified for all applications, there are some additional parameters for each application depending on the characteristics of its data packet arrivals. For instance, if an individual application has Pareto data packet arrivals, then one must define mean packet size, mean burst size, mean idle time, and mean packet rate as parameters. We use a link list based approach to generate and keep track of each session, the implementational details of which, along with the default distributions for each application are given in Section 3.2.1.

3.1.2 Randomization of Source-Destination Pairs

While simulating wide area networks as well as the Internet, it is not sufficient just to generate sessions between fixed source destination pairs. Connections may originate from any source and destinations are chosen at random. Thus to model such behavior, we introduce several randomization operations in *ns* which use random number generators available in *ns*. For a given topology with the assumption with all sources being equally active, we randomize sources and destinations pairs by generating two uniformly distributed random variables varying between 1 and the number of nodes in the topology. Also, as mentioned in the previous subsection, in order to select whether to create a TCP or an UDP agent, we generate another uniformly distributed random variable between 0 and 1 and compare it with `PER_TCP`. We note that for the cases when the load is not uniform, we can easily use other distributions to characterize the workload division between the sources and use it to select the source-destination pairs.

3.2 Traffic Generation

Some aspects of traffic generation in a wide area network or the Internet were discussed in Section 2. Traffic generation for a given protocol is dependent on its dynamics as well as the prevailing conditions in the network. For wide area networks and the Internet, the dominating applications are WWW, Telnet, FTP, SMTP, and NNTP and we use the empirical distributions from (Paxson 1994; Paxson and Floyd 1995) to characterize the underlying protocols for these applications. We have also implemented two self-similar traffic generators in *ns* which may be used either as a background traffic generator or as the traffic generator for the aggregated WWW connections from a node. Such generators are useful in simulations for stochastically varying the available capacity of the links or for modeling an aggregated background load.

Application	Distribution		
	Inter-arrival	Duration	Data
Telnet	Exponential	Log-normal	Pareto
WWW	Exponential	Log-normal	Self-similar
FTP	Exponential	Log-normal	Pareto
SMTP	Exponential	Log-normal	Log-normal

Table 1: Distributions for session parameters for various applications (Paxson 1994; Paxson and Floyd 1995). The distribution for the sessions durations for WWW and FTP have been assumed to be log-normal.

3.2.1 Application Specific Traffic Generators

To generate traffic specific to each protocol, we use the empirical distributions from (Paxson 1994; Paxson and Floyd 1995) which are given in Table 1. Though for machine generated transfers like SMTP the connection arrivals are not well modeled as Poisson for large intervals, results in (Paxson and Floyd 1995) show that they can be reasonably well approximated by a Poisson process for short intervals. In the absence of any well defined models for the duration times for FTP and WWW transfers, we assume that they have a log-normal distribution to account for the long range temporal dependence. We note that the underlying distributions for any of the parameters can be changed easily according to requirements and with the availability of better models. Also, the packet arrival process of FTP and SMTP have been shown to be bursty and not Poisson (Paxson and Floyd 1995) and we use a Pareto distribution to model them. Finally, we use a self-similar traffic generator to model the packet arrival process for the aggregated WWW connections from a source (Crovella and Bestavros 1996).

The basic idea for generating traces for each application is to create sessions at a rate equal to the mean number of sessions (`MNS`) and insert them into a sorted linked list according to their ending times. Next, as each session in the list expires, we create its next occurrence according to the following rules and insert it into the linked list again if the new ending time is within the simulation time. The number of sessions which replace an expiring session are limited to 0, or 1, or 2 depending on the number of currently active sessions in the list. The new sessions are generated in a two step process. In the first step, with a probability of $(1-(\text{No. of active sessions}/2*\texttt{MNS}))$, a new session is generated. Next in the second step, with probability 0.5 a new session is generated. This two step process causes number of active sessions to vary continuously, which is a more re-

List of parameters in `SupFRP`			
Parameter	Description	Default	Range
`rate_`	Transfer rate	64 KBps	> 0
`packet_size_`	Packet size	210 B	> 0
`FRPs_`	No. of FRPs	5	> 0
`hurst_`	H parameter	0.82	(0.5,1.0)

List of parameters in `SS`			
Parameter	Description	Default	Range
`rate_`	Transfer rate	100 Pkts/s	> 0
`packet_size_`	Packet size	210 B	> 0
`correlation_`	cov. at lag 1	0.6	(0.0,1.0)
`hurst_`	H parameter	0.82	(0.5,1.0)
`time_scales_`	See text	5	> 0

Table 2: List of parameters for the self-similar traffic generators.

alistic network scenario. The first step limits the number of active sessions to an upper bound of (2*`MNS`) and a lower bound of 1. The second step causes the number of active sessions to vary continuously, whereas the first step tries to maintain the average number of sessions close to `MNS`.

3.2.2 Self-Similar Traffic Generator

As indicated in Table 1, self-similar traffic generators are the best model for aggregated WWW packet arrivals. Also, self-similar traffic generators are needed for generating background traffic. To address these requirements, we have implemented two self-similar traffic generators in *ns* (1) `SupFRP` based on an algorithm proposed in (Ryu 1996) and (2) `SS` based on Markovian models proposed in (Andersen and Nielsen 1998). There generators are embedded as the components `Application/Traffic/SupFRP` and `Application/Traffic/SS` in the class `Application` of *ns*.

The algorithm for `SupFRP` is based on a renewal method proposed in (Ryu 1996). The generator uses the superposition of a number of independent Fractal Renewal Processes (FRPs) to generate the desired traffic. The burstiness of the source can be controlled by varying the number of FRPs with a larger number leading to less bursty traffic. The traffic generator `SS` is based on the superposition of a number of Markov Modulated Poisson Processes (MMPPs) and the procedure for fitting the parameters of the self-similar process to MMPPs is outlined in (Andersen and Nielsen 1998). The parameters associated with these generators and a brief description for each is given in Table 2. The parameter `time_scales_` for the source `SS` corresponds to the number of time scales over which the source exhibits burstiness and a value of 5 is adequate for most cases. In Section 4.2 we validate the long-range dependence of each of these sources and comment on the relative merits of both.

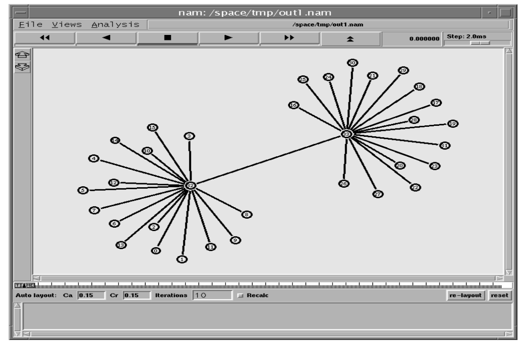

Figure 1: Topology of the network for the validation tests.

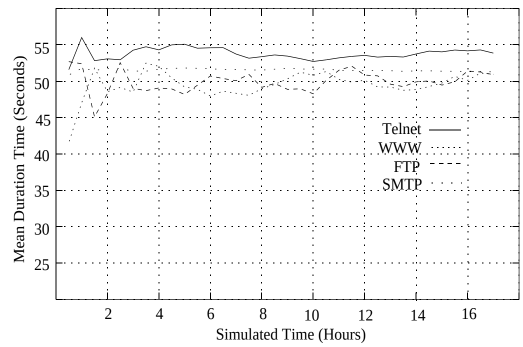

Figure 2: Mean duration times for the sessions of various protocols as a function of the simulation length.

4 Simulation Results

In this section we present simulation results to validate the additions we have made to *ns*. The new code was added to version 2 (ns-2.1b4) of *ns* and is available on request (Yuksel 1999). Figure 1 shows the simulated topology which has two sets of 16 hosts connected to two routers separated by a bottleneck link. The bandwidth of each link was assumed to be 10Mbps and had a propagation delay of 10 msec. We note that though the topology is not complex enough to account for the hierarchical nature of wide area networks, it is sufficient for testing the validity of the new tools that we have implemented.

4.1 Session Generation

Figures 2, 3 and 4 show the results for validating the randomized session generators and the application specific traffic generators. For each of these curves, we ran the simulations for desired average values of the parameters `MDT`, `MIAT` and `MNS` and tried to ascertain whether the traffic in the simulated network matched the specified values. The graphs in this section show the results for each of these three parameters which were introduced in Section 3.1.1 and elaborated on in Section 3.2.1. For each of the parameters, we specified a desired average and the figures plot the average values obtained from the simulator as the simulated time increases.

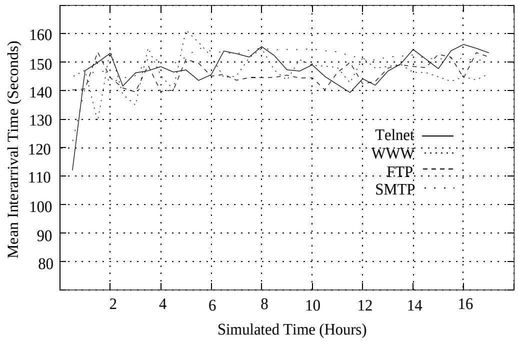

Figure 3: Mean inter-arrival times for the sessions of various protocols as a function of the simulation length.

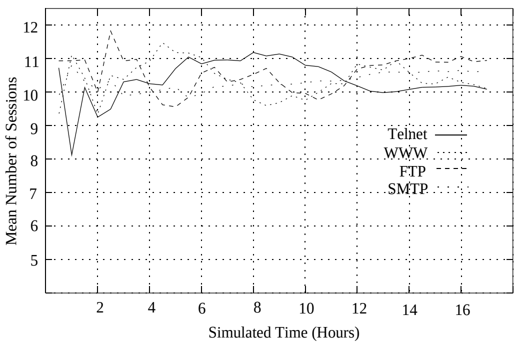

Figure 4: Mean number of sessions of various protocols as a function of the simulation length.

In Figure 2 we show the average duration time of a session as a function of the simulation time. Results are plotted for Telnet, WWW, FTP and SNMP traffic. For each of these applications, a desired average duration time of 50 seconds was specified and we can see that as the simulated time increases, the observed values are very close to the desired duration. We also note that the observed duration times reach a fairly constant value in the first few hours of simulation and there is little variation as time increases.

Figure 3 plots the average inter-arrival times between the sessions of Telnet, WWW, FTP and SNMP. For each application, the simulations were done with a desired average inter-arrival time of 150 sec. As in the previous case, the observed expected values converge to the desired value within a couple of hours of simulated time. There is a little jitter in the expected value for Telnet which we ascribe to the higher variance in the underlying distribution used to generate Telnet sessions.

The mean number of active sessions for Telnet, WWW, FTP and SNMP applications are plotted in Figure 4. As in the previous cases, the observed values converge to the desired average of 10 sessions in a short time. Also, as the simulation length increases, there is a little variance in the average number of the sessions which reflects the variance in the distributions used to generate the session durations.

4.2 Self-Similar Traffic Generators

To validate the self-similar and long range dependent nature of the traffic generated by the self-similar traffic generators `SupFRP` and `SS` we plot their covariance function in Figure 5. The graphs plot the covariance of the traffic generated by `SupFRP` and `SS` for Hurst parameters of 0.50, 0.75 and 0.90. The graphs for `SupFRP` were generated with 5 FRPs and the simulation was done to obtain a trace of 1,500,000 packets at an average of 200 packets/sec. The curves for `SS` used the superposition of 6 MMPPs with a burst time scale of 5 and correlation at lag 1 of 0.6. For both graphs we also plot the curve for a Poisson process with the same arrival rate for comparison and note that the curves for $H = 0.50$ approach that of the Poisson process.

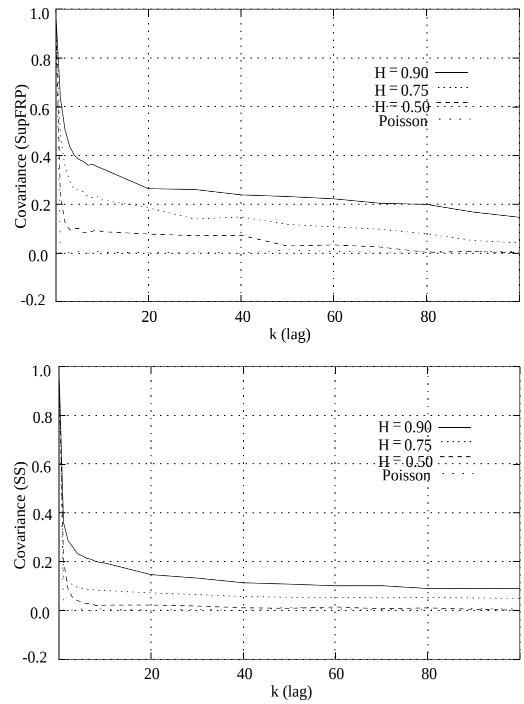

Figure 5: Variance of the traffic generated by `SupFRP`, `SS` and a Poisson process.

In Figure 5 we note that for $H = 0.5$, the variance curves tend to that of a Poisson process as expected. Also, the curve for `SS` is a better match than that for `SupFRP`. For Hurst parameters greater than 0.5, the variance function does not tend to zero at higher lags indicating the non summable nature of the autocorrelation function. Also, from the slope the autocorrelation function, it is easily verified that the generated traffic has a Hurst parameter very close to the specified value. We note that the curves for the MMPP based process results in a closer approximation of the self-similar process. However, for longer traces, the accuracy of `SupFRP` improves. We also note that with the MMPP based source, we have an additional degree of freedom in specifying the desired traffic characteristics by using the parameter `correlation_`

5 Conclusions and Discussions

In this paper we presented a methodology for generating realistic workload distributions for wide area networks and the Internet using the network simulator *ns*. This methodology captures the temporal and spatial interactions between sources and the network and the connections themselves. In addition to randomizing the source and destination pairs of the generated sessions, we have implemented tools to maintain the desired percentage of the traffic contributed by different applications and protocols. Also, we have implemented tools to generate the sessions for various protocols with statistical properties governed by user specified protocols and parameters.

A link list based approach is used to generate and keep track of the active sessions of each application. When the mean number of sessions is large, the sorted linked list results in a very good efficiency. Instead of comparing every session's ending time with the stopping time of the simulation, we can directly take the first element of the linked list for the generation of the next session. However, if the mean number of sessions is very small, then using a linked list might slow down the process. But considering the fact that the traffic generator is to be used for the workload generation of large wide area networks with big topologies, using a sorted linked list is undoubtedly the better option.

To generate realistic aggregate background traffic, we have embedded two self-similar traffic sources into *ns* based respectively on fractal renewal processes and Markov Modulated Poisson Processes. The results for the MMPP based source are more accurate for smaller traces though as the traces get longer, `SupFRP` improves its accuracy. Also, the MMPP based source has an additional parameter (`correlation_`) which allows an additional degree of freedom while specifying the characteristics of the desired traffic. The number of fractal processes used in the FRP based source can be used to control the burstiness of the traffic where the burstiness, and correspondingly the variance, is inversely proportional to the number of fractal processes. Correspondingly, depending on the smallest time scale of interest and the length of the simulation, `SS` uses a parameter `time_scales_` to specify the order of time scales over which burstiness occurs.

We are currently working on integrating the workload generator with an on-line collaborative simulator for network management and control (Vastola, Szymanski, and Kalyanaraman 1998). An interesting aspect of such a scenario would be to efficiently distribute the workload generation process to the participating nodes and observe the effect of their control mechanisms on the generated traffic. In addition to the two self-similar sources implemented here it would also be of interest to implement sources based on other techniques and compare their performance.

References

Andersen, A. T. and B. F. Nielsen (1998, June). A markovian approach for modeling packet traffic with long-range dependence. *IEEE Journal on Selected Areas in Communications 16*(5), 719–732.

Apisdorf, J., K. Claffy, K. Thompson, and R. Wilder (1997, June). Oc3mon: Flexible, affordable, high-performance statistics collection. In *Proceedings of INET'97*.

Calvert, K., M. Doar, and E. W. Zegura (1997, June). Modeling internet topology. *IEEE Communications Magazine 35*(6), 160–163.

Claffy, K., G. Miller, and K. Thompson (1998, April). The nature of the beast: recent traffic measurements from an internet backbone. In *Proceedings of INET'98*.

Crovella, M. E. and A. Bestavros (1996, December). Self-similarity in world wide web traffic: Evidence and possible causes. *IEEE/ACM Transactions on Networking 5*(6), 835–846.

Leland, W. E., M. S. Taqqu, W. Willinger, and D. V. Wilson (1994, February). On the self-similar nature of ethernet traffic (extended version). *IEEE/ACM Transactions on Networking 2*(1), 1–15.

ns (1997). `ucb/lbln/vint` network simulator - *ns* (version 2). `http://www-mash.cs.berkeley.edu/ns`.

Paxson, V. (1994, August). Empirically derived analytic models of wide-area tcp connections. *IEEE/ACM Transactions on Networking 2*(4), 316–336.

Paxson, V. (1996). End-to-end routing behavior in the internet. In *Proceedings of SIGCOMM'96*, pp. 25–38. ACM.

Paxson, V. and S. Floyd (1995, June). Wide area traffic: The failure of poisson modeling. *IEEE/ACM Transactions on Networking 3*(3), 226–244.

Paxson, V. and S. Floyd (1997, December). Why we don't know how to simulate the internet. In *Proceedings of the 1997 Winter Simulation Conference*. SCS.

Ryu, B. K. (1996). *Fractal network traffic: From understanding to implications*. Ph. D. thesis, Columbia University, New York City.

Thompson, K. and G. J. Miller (1997, November/December). Wide-area internet traffic patterns and characteristics. *IEEE Network 11*(6).

Vastola, K. S., B. Szymanski, and S. Kalyanaraman (1998). Network management and control using on-line collaborative simulation. `http://networks.ecse.rpi.edu/~olsim`.

Yuksel, M. (1999). Traffic generator for an on-line simulator. Master's thesis, Rensselaer Polytechnic Institute, Troy, Dept. of Computer Science.