Proceedings of the 2005 Winter Simulation Conference
M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, eds.

REALISTIC INTERNET TRAFFIC SIMULATION THROUGH MIXTURE MODELING AND A CASE STUDY

Song Luo

School of Computer Science
University of Central Florida
Orlando, FL 32816, U.S.A.

Gerald A. Marin

Department of Computer Science
Florida Institute of Technology
Melbourne, FL 32901, U.S.A.

ABSTRACT

Internet background traffic modeling and simulation is the main challenge when constructing a test environment for network intrusion detection experiments. However, a realistic simulation of network traffic through analytical models is difficult, because the classic distributions are usually ineffective when applied to traffic-related random variables. A modeling and simulation approach using heavy-tailed mixture distributions is introduced in this paper. In the case study, this approach is used to build analytical models for random variables of several major Internet applications (FTP, HTTP, SMTP, POP3, SSH) of a campus network. Several statistical features of an NS2 simulation are compared against those of the traffic traces being simulated. The comparison indicates that the simulation is statistically similar to the real traffic.

1 INTRODUCTION AND BACKGROUND

The modeling of the Internet traffic has long been important to networking product developers, performance analysts, network administrators, and researchers. For instance, when developing Intrusion Detection Systems (IDS), one needs a test bed composed of attack-free background traffic and intentionally inserted network attacks to evaluate the system's performance. In this case, the main challenge is how to obtain a simulation of the Internet background traffic, which preserves essential characteristics of the real traffic, or to be realistic.

However, as discussed by Floyd and Paxon (2001), simulating the Internet traffic is a difficulty task due to the heterogeneous structure, immense size, and changing property of the Internet. In the same work, Floyd and Paxon also point out that it is more appropriate to simulate the Internet traffic on the application level than the packet level. The reason is, the packet-level pattern of a network traffic is shaped by the network condition in which the traffic occurs, while the application-level pattern usually is more stable.

Danzig and Jamin (1991) introduced a library, Tcplib, to help generate realistic TCP/IP network traffic. Tcplib is an application-level empirical model that models 5 different types of Internet application traffic (FTP, SMTP, NNTP, TELNET and RLOGIN). Several limitations exist in Tcplib. First, it needs a better model of conversation arrival rates. Second, it lacks several application-specific details. For example, the interarrival time of FTP control packets and the distribution of number of request-response handshakes that occur during SMTP and NNTP conversations were not modeled. Also, because this work preceded the growth of the web, Tcplib does not include the model of HTTP traffic, which is critical for today's network traffic simulation.

Paxson and Floyd (1994, 1995) examined 3 million TCP connections from a number of wide-area traffic traces and a variety of sources. Some analytical models were derived to describe the random variables associated with TELNET, NNTP, SMTP and FTP connections. Paxson and Floyd's work might be the most referenced literature in the network traffic simulation research. However, like the Tcplib, the HTTP protocol is absent from their models. Even for protocols that had been studied, it is still necessary to revise their models, considering the long time since Paxon and Floyd first proposed their work and the applications may have changed their behavior significantly.

When modeling network traffic related random variables, it has been noticed that some variables have heavy-tailed features in their cumulative distributions (Willinger, Taqqu, Sherman, and Wilson 1995). By using traffic traces captured in February 2003 at University of Central Florida, the authors became aware that many statistical variables of modern Internet traffic differed from Paxon's distribution models and preferred more than before to have heavy-tailed features. Even using heavy-tailed distributions, some variables, such as bytes transferred by FTP, HTTP and SMTP protocols are less straight forward to be modeled. In this paper we document a successful modeling approach that leverages mixture and heavy-tailed distributions. We also present results from NS-2 simulation, which show our models are able to generate Internet background traffic

similar to the real ones in the sense that they have the same or close degree of several statistical features. The results should be of interest to those attempting to emulate network traffic environments.

2 HEAVY-TAILED DISTRIBUTION

In the usual way, we denote the cumulative distribution function (cdf) of a random variable X as

$$F_X(x) = P[X \le x]$$

and its associated probability density function (pdf) as

$$f_X(x) = F_X^{'}(x)$$

when this derivative exists. The distribution of a random variable X is said to be heavy-tailed if

$$1 - F_X(x) = P[X > x] \sim x^{-\alpha}, \text{ as } x \to \infty, 0 < \alpha < 2.$$

Heavy-tailed distributions have a number of properties that are qualitatively different from distributions more commonly used, such as Poisson, normal or exponential distributions (Crovella, A. Bestavros, 1997). As parameter α decreases, an arbitrarily large portion of the probability mass may be present in the tail of the distribution. In other words, a random variable that follows a heavy-tailed distribution can give rise to extremely large values with non-negligible probability.

To assess the presence of heavy tails in the traffic data, one can employ log-log complementary distribution (LLCD) plots. These are plots of the complementary cumulative distribution $\overline{F}(x) = 1 - F(x) = P[X > x]$ on log-log axes. Plotted this way, heavy-tailed distributions have the property that

$$\frac{d \log \overline{F}(x)}{d \log x} = -\alpha, \ x > \theta$$

for some real threshold θ and the shape parameter α>0.

Probably the most commonly used heavy-tailed distribution is the Pareto distribution with pdf given by

$$f(x) = \alpha k^{\alpha} x^{-\alpha-1}, \ \alpha > 0, \ k > 0.$$

The corresponding cdf is

$$F(x) = P[X \le x] = 1 - (k/x)^{\alpha}.$$

Note that parameter α can be measured by the slope of straight line behavior in LLCD plot; parameter k can be estimated by the minimum value of samples, starting from which the data shows heavy-tailed behavior.

3 TRAFFIC MODELING

In this section, we describe the procedure we used to model random variables of the FTP, HTTP, SMTP, POP3 and SSH traffic of a campus traffic data set. We demonstrate that how the random variables of a protocol are determined, and how the distributions of these variables are modeled. Mixture and heavy-tailed distributions are used for several random variables.

3.1 LAN Traffic Analysis

We captured IP headers from millions of Ethernet frames from the Computer Science department at UCF during a 10-hour period on February 05, 2003. Analysis shows that most packets are TCP packets, and more than half TCP connections belong to 5 Internet applications. See Figure 1 and Figure 2.

In this paper, we model random variables of 5 major Internet applications of the UCF CS department: HTTP, FTP, SMTP, POP3 and SSH. Our purpose is to propose a modeling procedure, by which a realistic background traffic simulation of a specific network can be achieved.

3.2 Modeling FTP traffic

Each FTP session includes an FTP control connection and either zero, one, or multiple FTP-DATA connections in "active" or "passive" mode. In this section we are interested in modeling distributions of the following random variables:

- A_{FTP}: FTP session arrivals;
- N_{FDC}: number of FTP-DATA connections per session;
- B_{FDC}: number of bytes transferred per FTP-DATA connection;
- I_{FDC}: Idle-time between adjacent FTP-DATA connections.

One difficulty of identifying the passive FTP-DATA traffic is that it does not use fixed port number in transferring. Not like transmissions in the active mode, which always use the port 20 on the server side, passive FTP transmissions might use any number above 1024 as the port number for the client or the server. We use the following 8 rules to distinguish valid FTP-DATA connections from the synthetic background traffic:

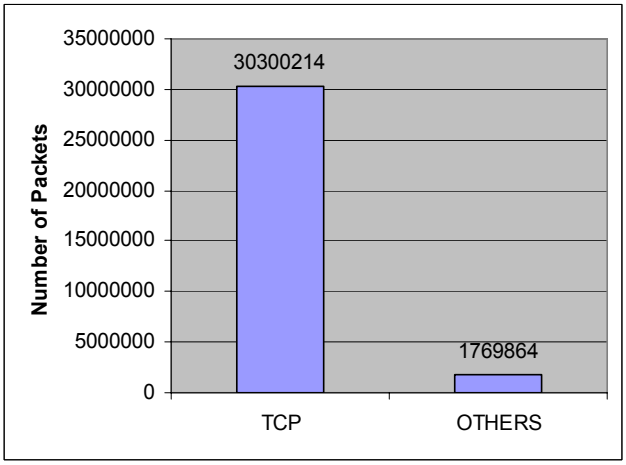


Figure 1: Categories of Captured IP Packets

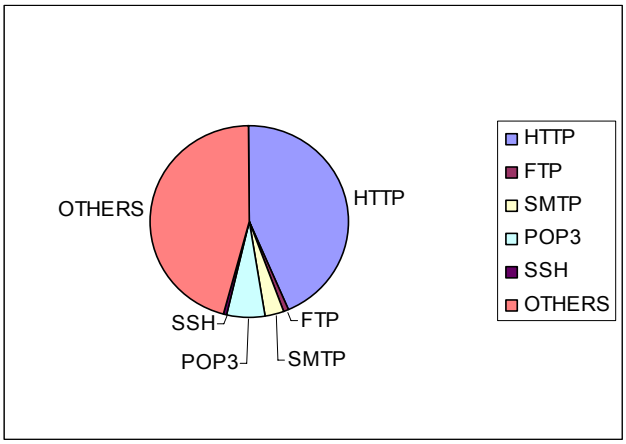


Figure 2: Categories of Captured TCP Connections

- All connections on port 20 are valid FTP-DATA connections;
- Connections with no actual data payload are not FTP-DATA connections;
- Connections that send data from both ends are not FTP-DATA connections;
- The port numbers of both ends of a passive FTP-DATA must be above 1024;
- If a connection is a passive FTP-DATA, the client of its parent control session must initiate the FTP-DATA connection;
- The time span of an FTP-DATA connection must be completely covered by the time span of its parent control session;
- An FTP control session's child data connections are not overlapped on the time;
- The port numbers of FTP-DATA connections spawned by an FTP control session should always be increasing, when the FTP-DATA connections are ordered by their creation time.

We first introduce the procedure of modeling the random variable B_{FDC}, the number of bytes transferred during a single FTP-DATA connection.

Extensive investigation indicates that the distribution of B_{FDC} does not match any known classic model. In this one the distribution does have a heavy tail, which is seen clearly in its LLCD plot, Figure 3.

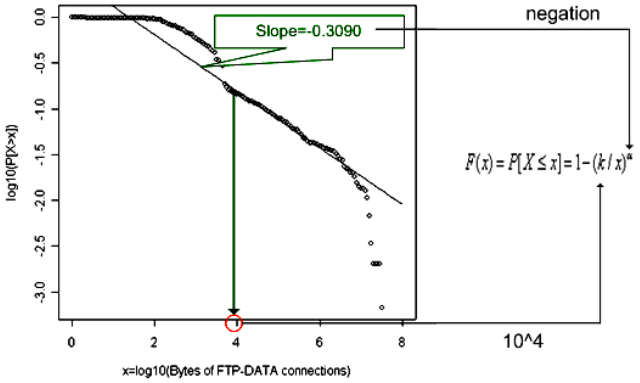


Figure 3: LLCD of B_{FDC}

The straight-line behavior in Figure 3 begins roughly at x=4. The range of heavy-tailed behavior corresponds to byte numbers greater than 10000, which accounts for about 15% of all sample values. Thus a good model for the upper 15% of B_{FDC} might be a Pareto distribution with parameter k=10000 and α=0.3090 (the negative slope of the straight line).

Now let's consider the lower 85% samples (or samples with a value less than 10000). CDF plot in Figure 4 of this part suggests an Exponential distribution with the rate parameter 0.00052.

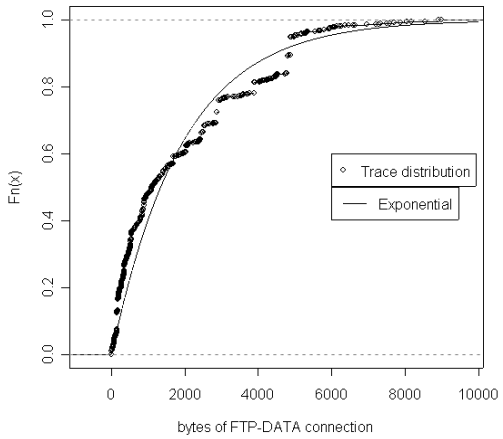


Figure 4: CDF of Lower 85% of B_{FDC}

Having determined distribution models for the upper and lower part of the distribution sample, we combine them together to build the final model for the random variable B_{FDC}, which is depicted in Figure 5. One Exponential

distribution and One Pareto distribution describes the lower and upper part of the sample distribution respectively, and the final model is a mixture distribution. The results of Chi-square test (indicated in Figure 5) support that there is no significant difference between the model and the real distribution.

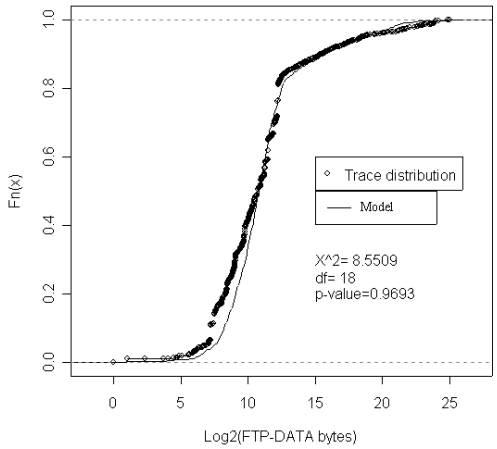


Figure 5: The Final Model of B_{FDC}

Analysis shows that the FTP session arrival is a typical Poisson process. The random variable N_{FDC} also has a heavy-tail behavior in almost all its distribution mass, thus a single Pareto model can describe it well.

When Modeling the idle time between two FTP-DATA connections (I_{FDC}), we noticed that one can best model this random variable when its values were divided into two groups, both of which can be described by a Gamma distribution. The first group includes values less than one second, which usually are results of using automated FTP client software. We call it *automated group*; The other group only contains values more than one second, and these big idle times are usually produced by human's manual manipulation of file transferring. We call the second group *manual group*. Although both groups can be well modeled by Gamma distributions, we found the *automated group* has to take the unit of microsecond when measured and need a logarithm transform first.

Table 1 summarizes models of FTP-related random variables.

3.3 Modeling HTTP traffic

We model HTTP traffic by a page-oriented structure, which was used by Reyes-Lecuona et al. (1999), except that we do not include the packet level. On the top of this structure is a HTTP session, which is defined as a user's continuous browsing activities. During one session, the user opens one or multiple web pages. One page contains one or multiple TCP connections to retrieve all objects embedded in that page. A research work by Casilari, Gonzalez, and Sandoval (2001) reveals that no matter the

Table 1: Models of FTP Traffic

Variable	Distribution	Parameters
A_{FTP}	Poisson	λ varies every 5 minute
N_{FDC}	Pareto	α=1.0595, k=3
B_{FDC}	Pareto (upper 15%)	α=1.15, k=10000
	Exponential (lower 85%)	rate=0.00052
I_{FDC}	Manual Group	shape=0.227, scale=73.962
	Automated Group	shape=202.04, scale=0.079

HTTP protocol uses persistent connection (HTTP 1.1) or separate connections (HTTP 1.0) to retrieve web page contents, the statistic characteristics of traffic traces captured in both situations have no significant difference. Based on this conclusion, we restrict the HTTP structure up to the page level, and ignore any detail within a page. In this paper, four HTTP-related random variables are modeled. They are:

- A_{HS}: HTTP session arrivals;
- N_{HPS}: number of pages per HTTP session;
- B_{HP}: bytes transferred per page;
- T_{HP}: user thinking time between pages.

Table 2: Models of HTTP Traffic

Variable	Distribution	Parameters
A_{HS}	Poisson	λ varies every 1 minute
B_{HP}	Pareto (upper 36%)	α=1.164, k=$10^{4.25}$
	Exponential (lower 64%)	rate=0.0002419939
N_{HPS}	Pareto	α=1.26, k=1
T_{HP}	Gamma	shape=0.9936, rate=0.0504

Another difference from Reyes-lecuona's work is how we determine the beginning of an HTTP session and the beginning of an HTTP page. When we come up with a new connection from a user, two time periods are measured. One is the connection spacing, which is the time between the start of the new connection and the end of the last connection from the same user. If this time exceeds 30 min., or 1800s, we consider the new connection as the beginning of a new coming HTTP session. Another measurement is the difference of starting times of two consecutive TCP con-

nections from a user, which is intended to represent the time between two mouse clicks (i.e., the openings of two web pages by a user). Reyes-Lecuona (1999) uses 30 seconds as the threshold to distinguish two pages. Thirty seconds may be appropriate for wireless networks (Reyes-Lecuona's work was done on analysis of wireless traffic), but for wired networks, it is too large. Our experiments suggest that 1 second is a good threshold for wired network.

Table 2 summarizes the models for random variables of the HTTP traffic. Both N_{HPS} and B_{HP} have heavy-tailed feature, and the modeling of B_{HP} involves mixture distributions.

3.4 Modeling SMTP and POP3 traffic

The SMTP and POP3 traffic are both modeled on the connection level.

Our experiments show that even the simplest SMTP connection (which includes one email having no message in the body, no attachment, but only sender and receiver's addresses and a very short one-word subject) has a data payload slightly more than 500 bytes in its TCP packets. We discard all SMTP connections with payloads less than 500 bytes before modeling. We assume they are generated by scans.

Experiments on POP3 connections show that, POP3 client software would always issue several commands, such as LIST and UIDL, after the connection with the server established and the authorization passed, trying to get information about the maildrop on the server. The server responds to each command from the client; however, the length of the response depends on how many email messages of the user account exist on the server.

Experiments also show that a successful POP3 conversation with a remote server, which has an empty maildrop, has a sum about 90 bytes of data payload in its connection. One POP3 connection that actually receives emails from server will have a data payload at least about 1000 bytes in its connection. Here we classify captured POP3 connections into 3 categories: *invalid* connections are POP3 connections with payload less than 90 bytes, which cannot possibly complete the simplest conversation; *unloaded* connections are those with payload between 90 and 1000 bytes; *loaded* connections as those with payload more than 1000 bytes. *Unloaded* POP3 connections log in to a server successfully, and check the information of the maildrop, but do not download any email. *Loaded* connections compose all actions of the *unloaded* connections, and retrieve at least one email from the server. It is possible that some connections have more than 1000 bytes of payload but retrieve nothing, and they should be classified as *unloaded*. In practice, we cannot distinguish them from the *loaded* connections only by TCP header information. It is appropriate and more feasible to think them as *loaded* connections. The following random variables are needed to model SMTP and POP3 traffic:

- A_{SC}: SMTP connection arrivals;
- B_{SC}: bytes transferred per SMTP connection;
- A_{PC}: POP3 connections arrivals;
- B_{PC}: bytes transferred per POP3 connection.

The SMTP and POP3 connection arrivals are close to Poisson processes, when measured in small (1-minute) time intervals. Modeling of B_{PC} uses mixture distributions. Table 3 lists all models, and Figure 6 compares the model and the actual distribution of the variable B_{PC}.

Table 3: Models of SMTP and POP3 Traffic

Variable	Distribution	Parameters
A_{SC}	Poisson	λ varies every 1 minute
B_{SC}	Pareto	α=0.8454, k=1250
A_{PC}	Poisson	λ varies every 1 minute
B_{PC}	Gamma (loaded)	shape=40.6029, rate=2.8903
	Pareto (unloaded)	shape=1.873, k=90

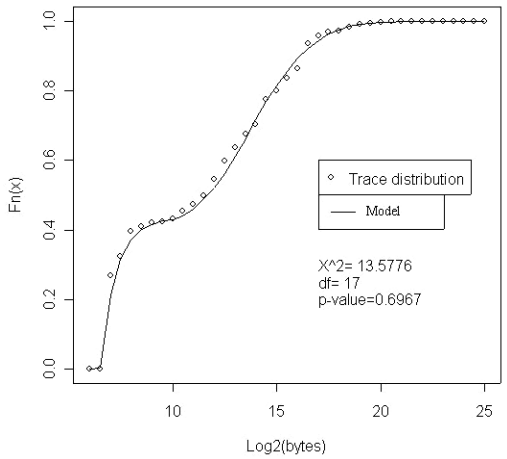


Figure 6: The Final Model of B_{PC}

3.5 Modeling SSH traffic

Generally, the usage and communication pattern between an SSH client and the server are similar with those of TELNET. We then use the same four random variables as those in Paxon's work (1994, 1995) used to model TELNET traffic for SSH traffic modeling. The random variables are:

- A_{SSH}: SSH connection arrivals;

- N_{SSH}: number of originator packets per connection;
- T_{SSH}: time between adjacent originator packets;
- B_{SSH}: bytes per SSH response.

In a normal SSH connection, the client (or the originator) sends small packets the server, and the server responds to each originator packet with one or more packets which vary in size. It is more reasonable to measure a server's response by its number of bytes, instead of the number of packets.

Table 4 summarized the models used for SSH traffic modeling.

Table 4: Models of SSH Traffic

Variable	Distribution	Parameters
A_{SSH}	Gamma	Shape=0.2784 Rate=0.2260
N_{SSH}	Log2-normal	μ=7.9613, σ=2.8304
T_{SSH}	Log10-normal	μ=3.4624, σ=1.6275
B_{SSH}	Pareto	α=1.079, k=32

4 SIMULATION DESIGN

After the models of all considered random variables are determined, we simulate the CS traffic traces in the NS-2 simulator. Our purpose is to generate network traffic which is statistically similar to the captured one. We choose NS-2 as the simulating platform because: 1) it provides authentic support for TCP, which means that in NS-2 TCP runs realistically with virtually all options supported; 2) the trace and monitoring support provided by NS-2 makes it convenient to collect all packet traces produced by the simulation.

All simulated TCP connections are scheduled as events in NS-2. The models of a specific protocol decide when a TCP connection of this protocol should be launched and how many bytes should be sent during this connection.

Figure 7 depicts a simple network structure we used in the NS-2 simulation. In this structure, all traffic of a specific protocol occurs between a client node and a server node. For example, an HTTP client sends out all simulated web requests; an HTTP server receives all the requests and generate all responding web pages. The client nodes of five protocols are connected to a router via high speed links, representing a LAN environment. The five servers are connected to another router also via high speed links. A large-delay high-throughput link connects the two routers, representing a typical Internet backbone connection. A traffic monitor is place on the backbone link, thus all TCP packets produced by the simulation can be captured and saved to files on the hard disk for further analysis.

We simulated the entire 10 hours of collected CS traffic on the NS-2 platform. The running time of the simulation in the Debian Linux 3.0 on a PC with Pentium 4 2.0 GHz CPU and 1GB memory is about 40 minutes.

5 EXPERIMENT RESULTS

The goal of the simulation is to produce network traffic that is similar to that being simulated. The similarity is examined by the following traffic features:

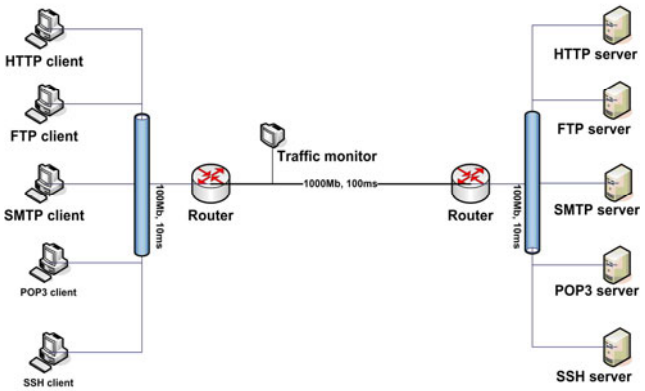


Figure 7: The Network Structure of NS-2 Simulation

- Application-level traffic pattern;
- Packet arrivals;
- Self-similarity;
- Correlation integrals and fractal dimension.

A successful simulation should have similar or close degree of the above statistical features with the real traffic.

Being built on the application level, the models of the five protocols in our experiment produced values of application-level elements (session arrivals, bytes per connection, connection idle time, etc.) in the same distributions as those of the CS traffic. Because of the limit of the space, we only plot the session arrivals of the HTTP traffic and its simulation in Figure 8.

We also compare the packet arrivals in the time period from the second 21000 to the second 25800, when the packet arrivals are most intense. Figure 9(a) and 9(b) depict the packet arrivals in this period of the CS traffic and the simulation, respectively.

Figure 9 shows that the mean and standard deviation of the packet arrivals of the simulation are close to the CS traffic. Furthermore, the burstiness of the simulation is also similar to the CS traffic: in most time, the arrival rate (packets/sec.) is below 1000; a few bursts are near 2000; the biggest burst is about 5000.

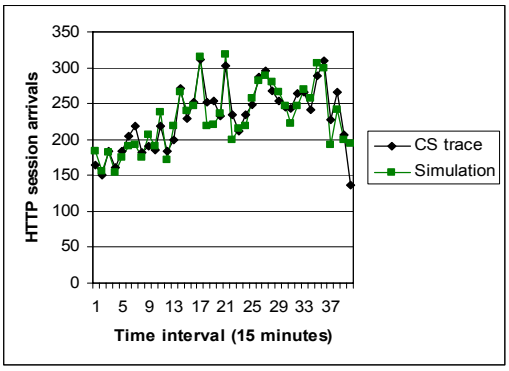
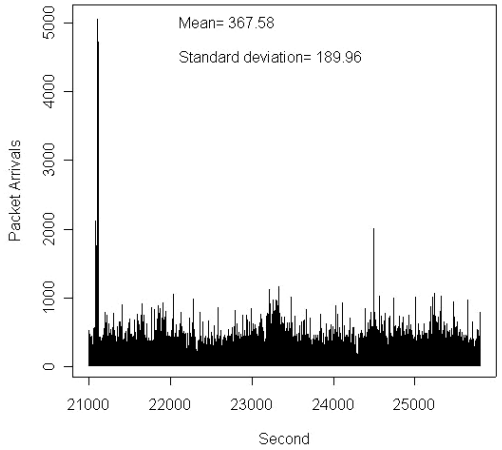
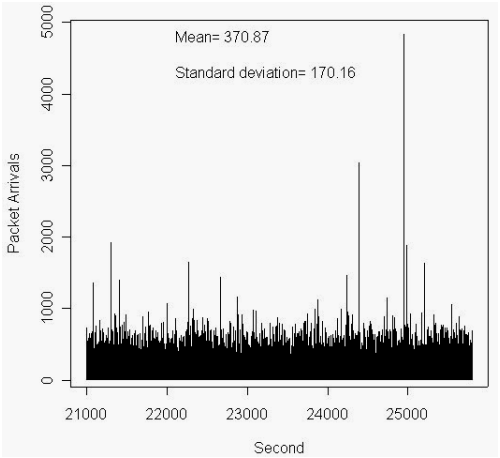


Figure 8: HTTP Session Arrivals



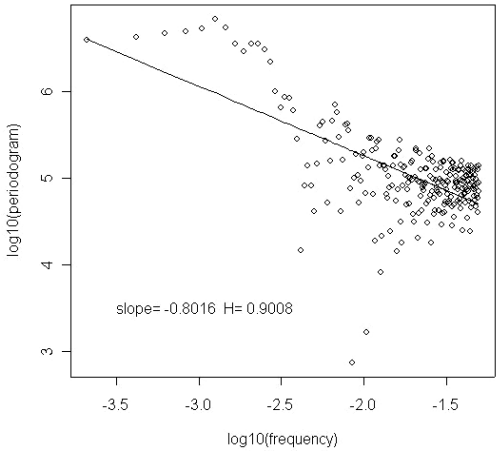
(a) Packet Arrivals of the CS Traffic



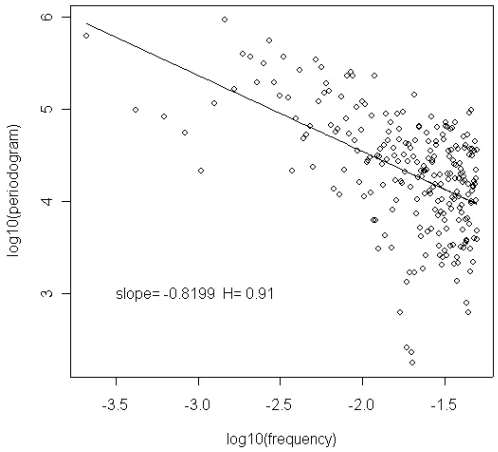
(b) Packet Arrivals of the Simulation

Figure 9: Comparison of Actual Packet to Simulated Packet Arrivals

As shown by Leland et al. (1994), and supported by later research (Beran 1995, Paxon 1995), the distribution of traffic in Local Area Networks and on the Internet commonly exhibits self-similarity, and the degree of the self-similarity can be measured by a Hurst parameter H. We test the self-similarity on the packet arrivals of both the CS traffic and the simulation. The self-similarity is computed with a log-log spectra-density plot near its origin. This method is described by Crovella (1997) and Willinger (1995). Figure 10(a) and 10(b) depict the test results for the CS traffic and its simulation. The Hurst parameter H is computed from the slope of the linear regression. The Figure shows that the simulation produced a very close degree of self-similarity, compared against the CS traffic.



(a) Self-similarity Test of the CS Traffic



(b) Self-similarity Test of the Simulation

Figure 10: Comparison of Self-Similarity of Actual Traffic to Simulated Traffic

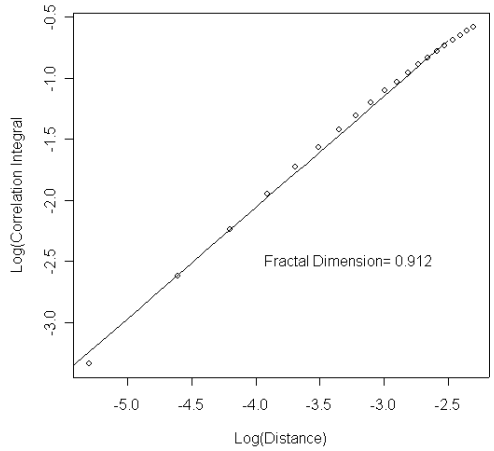
That network traffic also exhibits "fractal-like' behavior has been observed and studied by Leland (1994), Willinger and Paxson (1998). Grassberger and Procaccia (1983) give one method to calculate the fractal dimension of a time series through correlation integrals. Let x_i be the packet arrivals at the i^{th} second, we use the following formula to calculate its correlation integrals:

$$C(r) = \lim_{N\to\infty} \frac{1}{N^2} \sum_{i,j=1,i\neq j}^{N} \theta(r - |x_i - x_j|),$$

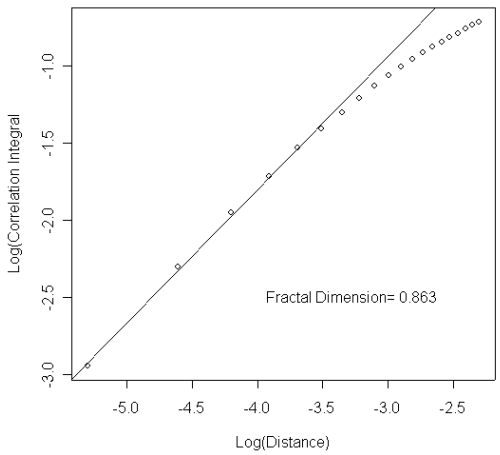
where $|x_i - x_j|$ is the Euclidean norm and θ is the Heaviside function

$$\theta(x) = \begin{cases} 0, x < 0 \\ 1, x > 0 \end{cases}.$$

The distance $|x_i - x_j|$ is defined as the difference of the number of the packet arrivals between the i^{th} and the j^{th} second.



(a) The Correlation Integrals and the Fractal Dimension of the CS Traffic



(b) The Correlation Integrals and the Fractal Dimension of the Traffic Simulation

Figure 11: Comparison of Correlation Integrals and Fractal Dimension of Actual Traffic to Simulated Traffic

Grassberger and Procaccia (1983) also show that for small r, the correlation integral C'(r) grows according to a power law:

$$C'(r) \sim r^{v},$$

where v is the estimation of the correlation fractal dimension, and it can be determined by plotting C'(r) vs. r on a log-log plot.

Figure 11(a) and 11(b) depict the correlation integrals and the fractal dimension for the CS traffic and the simulation. The correlation integrals are displayed as a sequence of dots in the figure, while the fractal dimension is the slope of the linear regression on small values of the distance r.

The fractal dimension of the CS traffic data is 0.912, while it is 0.863 of the simulation.

6 CONCLUSION

In this paper, we introduced an approach to model the Internet background traffic by using mixture heavy-tailed distributions. The approach is proposed because traditional distributions are ineffective for a number of random variables that are essential in modeling the Internet traffic. Our approach has been demonstrated for five major Internet protocols (HTTP, FTP, SMTP, POP3 and SSH) based on data collected from the CS LAN at UCF. An NS-2 simulation environment has been built and used for simulating the Internet background traffic. The experimental results are discussed and compared with the original traffic. The simulation results show that our models can produce realistic network traffic in regard to the following:

- The random variables of the simulation traffic have similar distributions with those of the real traffic;
- The pattern of the packet arrival of the simulation compares favorably with that of the actual arrivals;
- The simulation and the real traffic have close degree of self-similarity;
- The correlation integral results are comparable for both;
- The fractal dimension results are comparable for both.

7 FUTURE WORK

One limitation of this approach is that the models are protocol-specific. One must build different models for different Internet applications, and usually need to re-estimate parameters for each model when applying the method in a new network environment. One direction of future work is to classify the Internet traffic into major different categories by traffic patterns. Different applications/protocols can be classified into the same category, as long as they have the same communication pattern at the application/packet

level. Then the mixture-modeling approach can be applied on each traffic category.

We also found that, although a close degree of self-similarity to the CS traffic was achieved by the simulation, the simulated traffic does not have extremely high packet arrivals that characterize the real traffic in several specific periods of time. The simulation traffic looks less bursty than the real traffic during those periods. The reason may lie with the Pareto distribution that we used to model the heavy-tailed behavior of random variables. Other distributions with more obvious heavy-tailed feature might be suitable to produce extremely high bursts.

REFERENCES

Beran, J., R. Sherman, M. Taqqu, and W. Willinger, 1995. Long-range dependence in variable-bit-rate video traffic. *IEEE Transactions on Communications* 43(234): 1566 – 1579.

Casilari, E., F. J. Gonzalez, and F. Sandoval, 2001. Modeling of HTTP traffic. *IEEE Communication Letters* 5(6): 272 – 274.

Crovella, M. E., and A. Bestavros, 1997. Self-similarity in World Wide Web traffic: evidence and possible causes. *IEEE/ACM Transactions on Networking, 5(6):835 – 846.*

Danzig, P. B., and S. Jamin, 1991. Tcplib: A library of TCP/IP traffic characteristics. *USC Networking and Distributed Systems Laboratory TR CS-SYS-91-01.*

Floyd, S., and V. Paxon, 2001. Difficultes in simulating the Internet. In *IEEE/ACM Transaction on Network (TON)* 9(4):393 – 403.

Grassberger, P. and I. Procaccia, 1983. Characterization of strange attractors. *Physical Review Letters* 50(5): 346 – 349.

Leland, W., M. Taqqu, W. willinger, and D. Wilson, 1994. On the self-similar nature of Ethernet traffic (extended version). *IEEE-ACM Transaction on Networking* 2(1): 1 – 15.

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

Paxon, V. and S. Floyd, 1995. Wide area traffic: the failure of Poisson modeling. *IEEE/ACM Transactions on Networking,* 3(3):226 –244.

Reyes-Lecuona, A. et al, 1999. A page-oriented WWW traffic model for wireless system simulations. *Proceedings of 16th International Telegraphic Congress* 2:1271-1280.

Willinger, W. and V. Paxson, 1998. Where mathematics meets the Internet. *Notices of the American Mathematical Society* 45(8): 961 – 970.

Willinger, W., M. Taqqu, R. Sherman, and D. Wilson, 1995. Self-similarity through high-variability: statistical analysis of Ethernet LAN traffic at the source level. *IEEE/ACM Transaction on Networking* 5(1): 71 – 86.

AUTHOR BIOGRAPHIES

SONG LUO is a Ph.D. candidate in the school of Computer Science at University of Central Florida. His research interests include realistic modeling and simulation of network traffic, and network intrusion detection. His email address is <sluo@cs.ucf.edu> and his web address is <www.cs.ucf.edu/~sluo/>.

GERALD A. MARIN is a professor in the department of Computer Science at Florida Institute of Technology. His research interests include networking architecture, network performance, system design, and network security. His email address is <gmarin@cs.fit.edu> and his web address is <my.fit.edu/~gmarin/>.