IMPROVING CONFIDENCE IN NETWORK SIMULATIONS

Michele C. Weigle

Department of Computer Science
Old Dominion University
Norfolk, VA 23529, U.S.A.

ABSTRACT

Simulation has become the tool of choice for an increasing number of networking researchers. Unfortunately, standard statistical techniques often cannot be applied when Internet-like heavy-tailed workloads are used as input. We present issues involved in using heavy-tailed distributions in network simulations, including three different methods for dealing with such distributions in simulation. We also discuss the proper use of the random number generator implemented in the *ns*-2 simulator and the impacts of improper usage.

1   INTRODUCTION

Simulation has become the tool of choice for an increasing number of networking researchers. Unfortunately it appears that as the number of users of simulations increases, the number of sound simulation studies has not. Recently, several researchers (Kurkowski, Camp, and Colagrosso 2005; Pawlikowski 2003; Pawlikowski, Jeong, and Lee 2002; Perrone, Yuan, and Nicol 2003) have investigated the "credibility crisis" in network simulation analysis. Pawlikowski *et al.* found that a large majority of papers accepted to several conferences and journals did not consider aspects of randomness found in most simulation studies. Kurkoswksi *et al.* found similar issues present in the mobile ad-hoc networking community. These factors contribute not only to studies where the results may not be credible, but also to studies where the results may not be repeatable, especially if significant detail about the simulations is lacking. Compounding this issue is the use of heavy-tailed distributions with infinite variance to describe certain characteristics of Internet traffic, including file sizes. These heavy-tailed workloads often prevent researchers from using standard statistical methods for analyzing output data.

There have been several papers recently describing the process of data output analysis from simulation (Alexopoulos and Seila 2001, Nakayama 2002, Law 2004). Simulations are typically driven by samples from random variables. When sampling from a random variable distribution, the statistics derived from those samples are only approximations of the actual distribution. One of the main goals of output analysis is to assess the accuracy with which the samples reflect the true nature of the distribution. Often, the initial state of the simulation is not representative of the steady-state of the system (*e.g.*, simulations are often started with network queues in an empty and idle state). When running steady-state simulations, this transient period should be removed before computing statistics on the simulation data. Procedures for determining the length of the transient period in steady-state simulations and for determining how well the sample mean and variance approximate the true mean and variance of the system are well-described by Alexopoulos and Seila (2001), Nakayama (2002), and Law (2004).

The goal of this paper is to raise important issues in the analysis of data from network simulations. We highlight problems encountered when running simulations that require workloads drawn from heavy-tailed distributions. Many networking studies today focus on Internet-like traffic where file sizes and service times are based on heavy-tailed distributions. Unfortunately, in these situations, well-known statistical techniques for determining confidence are not applicable. We present the problem and several approaches that have been taken to address the problem. Another important, yet often ignored, issue in simulation studies is the proper use of random number generators (RNGs). Since *ns*-2 (Breslau et al. 2000) is one of the most popular network simulators, we present a guide for the proper use of the sophisticated RNG implemented in *ns*-2.

2   EFFECTS OF USING HEAVY-TAILED DISTRIBUTIONS

In this section, we describe heavy-tailed distributions, and in particular, the Pareto family of heavy-tailed distributions.

We show how these distributions have been used in network simulations and what properties make them difficult to use in simulation.

2.1 Heavy-Tailed Distributions

Distributions where the tail follows a power law are called heavy-tailed. Previous studies (Mah 1997; Park, Kim, and Crovella 1996) have shown that the distribution of the size of files transferred over HTTP is heavy-tailed. This is caused by the large number of small files and the non-negligible number of extremely large files associated with web content.

The most-often used heavy-tailed distribution to describe Internet file sizes is the Pareto distribution. Pareto actually describes a family of distributions. We discuss the Pareto of the first kind and the Pareto of the second kind, also known as the Lomax distribution. The probability distribution function (pdf) of the Pareto of the first kind is

$$f(x) = \frac{\alpha k^\alpha}{x^{\alpha+1}},$$

where $\alpha > 0$ and $x \geq k > 0$. The pdf of the Lomax is

$$f(x) = \frac{\alpha k^\alpha}{(x+k)^{\alpha+1}},$$

where $\alpha > 0$, $k > 0$, and $x \geq 0$. The α parameter is often called the shape parameter, and k is often called the scale parameter. When $\alpha \leq 2$, a Pareto distribution has infinite variance, and when $\alpha \leq 1$, the distribution has infinite mean.

2.2 Usage in Network Simulations

One criteria to tell how long to run a simulation is to run the simulation long enough for the sample mean to reach the theoretical mean. Crovella and Lipsky (1997) show that when sampling from heavy-tailed distributions, statistics such as the mean can take a very long time to reach steady-state. The PackMime HTTP model (Cao et al. 2004), for example, has a HTTP response size distribution where the body is described empirically and the tail is a Pareto of the first kind with $\alpha = 1.23$ and $k = 1400$. To achieve two-digit accuracy for the mean of such a Pareto distribution, over 10^{10} samples are required. Even then the mean is still unstable because of "swamping" observations that can double the mean with just one sample. The probability that a swamping observation could occur in a simulation that uses a Pareto ($\alpha = 1.23$) distribution is greater than 1 in 100. For these reasons, Crovella and Lipsky state that when $\alpha < 1.5$, simulation convergence becomes impractical. If the simulation statistic can not reach steady-state in a reasonable amount of time, then the simulation is always in a transient state.

To demonstrate the variability of heavy-tailed distributions, we ran a set of simulations with the PackMime-HTTP traffic generator in *ns*-2. In general, the HTTP protocol is simple: a web client sends a request to the web server, and after some delay, the web server sends back a response. In our experiments, we used PackMime-HTTP to model the request sizes as Pareto ($\mu = 50$ bytes, $\alpha = 1.2$, $k = 8.3$) and the response sizes as Pareto ($\mu = 7000$ bytes, $\alpha = 1.2$, $k = 1400$). New connections were initiated according to an Exponential distribution with a mean of 50 new connections per second. All network links were 100 Mbps, the round-trip time was 50 ms, and the maximum TCP window size was 64 KB, resulting in an uncongested network.

Figure 1 shows the offered load generated by the web servers, and Figure 2 shows the cumulative mean of the response sizes, with a theoretical mean of 7000 bytes. Notice that in Figure 2, the sample mean is not converging to the theoretical mean, and in fact appears to be diverging from the theoretical mean. The large spike in offered load around time 1160 seconds corresponds to the large jump in the average response size around sample 57,000. This spike is caused by a single response of size 64,403,669 bytes. Though the large response size observed is from the tail of the distribution, such large responses are not uncommon. We ran 30 replications of the simulation, generating 100,000 response size samples in each. In almost all of the replications, there is at least one large spike similar to the one shown in Figures 1 and 2.
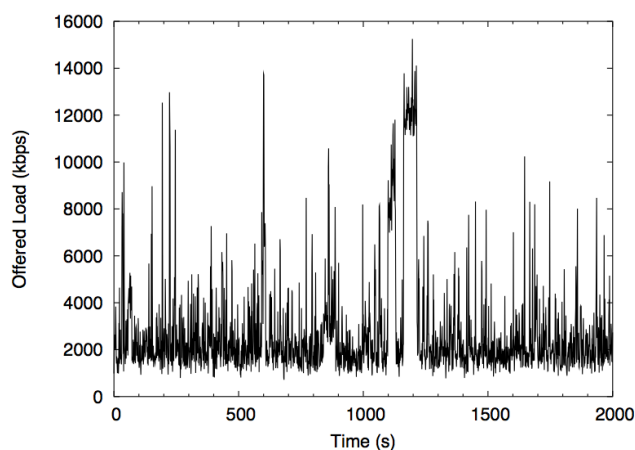

Figure 1: Offered Load with Pareto ($\alpha = 1.2$) Response Sizes

3 APPROACHES TO DEALING WITH HEAVY-TAILED DISTRIBUTIONS

When using heavy-tailed workloads as input, networking researchers have few options for statistically valid simulation analysis. The extremely long sample path lengths needed to
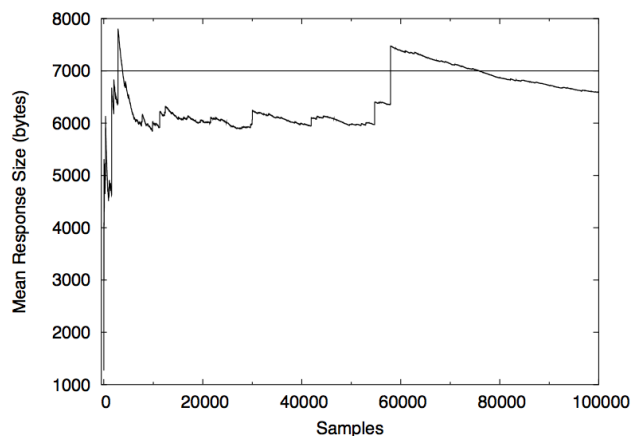
Figure 2: Mean Response Sizes from Pareto ($\alpha = 1.2$)

obtain a sample mean with high confidence often preclude standard statistical analysis. We discuss a few approaches that have been taken to handle heavy-tailed workloads in simulation. These include using a Bounded Pareto distribution that has finite mean and variance, approximating the Lomax distribution with a Lognormal distribution, or treating the simulation that uses Pareto distributions as transient.

3.1 Use Bounded Pareto

Crovella and Lipsky (1997) suggest using time scale as a simulation parameter and bounding the distribution. The time scale should be measured in the number of samples drawn from the distribution. For Pareto with shape α and scale k, they suggest defining the maximum observation to be Y, and

$$E[Y] = knB(n, 1 - 1/\alpha) \approx E[X]n^{1/\alpha},$$

where n is the number of samples and B is the Beta function. If a maximum possible observation p is known (*e.g.* maximum file size on a web server), then the simulation should be run long enough for $E[Y]$ to approach p. In Table 1, we show the number of samples n needed for $E[Y]$ to approach certain values of p for a Pareto distribution with $k = 1400$ and $\alpha = 1.2$.

Table 1: Number of Samples from Pareto ($\alpha = 1.2$) Required for $E[Y]$ to Approach Various Maximum File Sizes

Maximum File Size (p)	Samples (n)
MP3 - 5 MB	2500
TV episode - 350 MB	400,000
CD image - 650 MB	850,000
DVD image - 4.7 GB	9,000,000

This idea is expanded by Harchol-Balter et al. (1999) who have proposed using a Bounded Pareto distribution in order to allow for computation of the mean and variance for Pareto distributions where the mean is highly variable and the variance is infinite. The pdf of a Bounded Pareto is

$$f(x) = \frac{\alpha k^\alpha}{1 - (k/p)^\alpha} x^{-\alpha-1},$$

where α is the shape parameter (similar to α in Pareto), k is the smallest possible sample, p is the largest possible sample, and $k \le x \le p$. The idea is that for network studies, and heavy-tailed distributions of file sizes in particular, there is generally a maximum file size. This file size can be used to produce the Bounded Pareto distribution that has the same power-law tail characteristics as the unbounded original Pareto distribution. The Bounded Pareto distribution has been used in several studies that involve heavy-tailed workloads (Urvoy-Keller and Biersack 2002; Rai, Urvoy-Keller, and Biersack 2003; Alparslan, Akar, and Karasan 2004).

3.2 Approximate Pareto with Lognormal

Fishman and Adan (2006) give guidelines for creating a Lognormal distribution that approximates a Lomax distribution. They observed that distributions such as Lognormal converge much faster than Lomax. Given a Lomax shape parameter α, where $1 < \alpha < 2$, and a Lomax scale parameter k, they provide a set of parameters for the Lognormal distribution given the desired fit of the tail of the Lomax. To approximate the Lomax distribution, Fishman and Adan match the means of the Lomax and Lognormal distributions along with the $1 - \zeta$ quantiles of each distribution. The parameter ζ sets the point at which the tails of the distribution diverge. The smaller ζ, the better the fit. Fishman and Adan also provide the required sample path length (after the warmup samples have been removed) given the desired accuracy of the mean, $\tau(v) = \min\{\tau \ge 0 : \operatorname{var}\bar{x}(\infty, \tau) \le \mu v\}$, where μ is the actual mean and $\bar{x}$ is the sample mean. *High* accuracy is represented by $v = 10^{-4}$, *moderate* accuracy by $v = 10^{-3}$, and *gross* accuracy by $v = 10^{-2}$.

In Figure 3 we show the complementary CDFs (CCDFs) of file sizes drawn from a Lomax distribution ($\alpha = 1.2$, $k = 1400$) along with the corresponding Lognormal distributions with two different fits for the tail with enough samples for *moderate* accuracy of the mean. For $\zeta = 10^{-3}$, 330,000 samples were collected, and for $\zeta = 10^{-5}$, 100,000,000 samples were collected. Due to run time and memory constraints, only 643,783,352 samples were collected for the Lomax CCDF even though about 10^{12} are required for accuracy. Generating the 100 million Lognormal samples took almost 50 minutes on a 2.8 GHz Xeon processor with 4 GB of memory. This time included only generating the
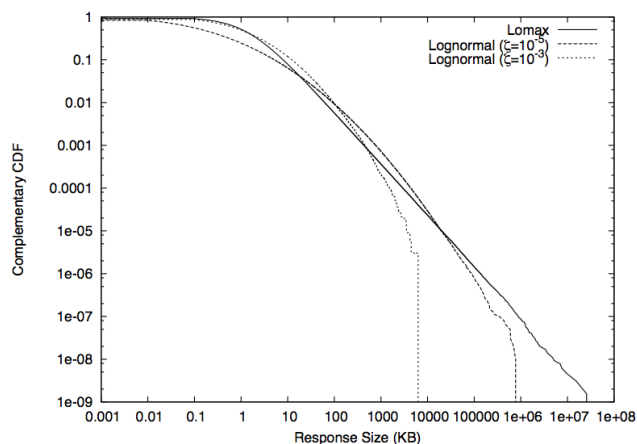
Figure 3: Complementary CDFs of File Sizes Generated by Lomax and Lognormal Approximations, with Moderate Accuracy for the Mean ($v = 10^{-3}$)

samples and writing them to disk. It would take much longer to run a complex packet-level simulation that required at least 100 million samples. Even though Lognormal converges with fewer samples than Lomax, for high levels of accuracy, simulation still may not be feasible until new methods for fast packet-level simulation are developed.

3.3 Treat as Transient

Other researchers have chosen to treat data resulting from heavy-tailed workloads as essentially transient. In previous work (Weigle, Jeffay, and Smith 2005), we used the PackMime-HTTP traffic generator in *ns*-2 and ran simulations long enough to observe a specific number of completed HTTP request-response pairs, usually 250,000. This large number allows the simulation to encounter several response sizes from the tail of the Pareto distribution. We also required that in each experiment, there was at least one HTTP response larger than 10 MB. We compared the performance of web traffic with different types of TCP congestion control and queuing mechanisms. Although the simulations were always in a transient state, we ran each protocol with the same input traffic, using the same RNG seed. This way we were able to compare the performance of the protocols in equivalent environments.

Some of the first work in evaluating web performance through simulation (Ott, Lakshman, and Wong 1999; May, Bonald, and Bolot 2000) ran simulations for a certain period of time, at most 100 seconds. Other work has run web-based simulations for 160 seconds (Claypool, Kinicki, and Hartling 2004). One of the problems running simulations with web-like workloads is that the actual running time of the packet-level simulation is very long. For example,

the PackMime-HTTP simulations described above often took over 6 hours to run a simulation that ran about 40 minutes in simulation time. Other work studying web cache performance (Williamson and Markatchev 2003, Houtzager and Williamson 2003) ended simulations after sending a specific number of requests into the system. In those cases, it was about 5000 requests. Actual running time and memory resources for these types of simulations appears to be a limiting factor in the length of the simulations performed.

If treating the simulation as transient, the running time, or number of samples gathered, may not be the most important factor in how well the CCDF of the sample fits the theoretical CCDF of the Pareto distribution. Large samples can occur at any time, so users of these heavy-tail distributions should be aware of the number of large samples present in their simulations and run their simulations long enough to observe some reasonable number of these large samples.

4 RANDOM NUMBER GENERATION

When using samples from a probability distribution as input to simulations, the random number generator used by the simulator becomes very important. In order to sample from a distribution, a simulator needs a random stream of samples from U(0,1) (Fishman 2001). To produce these samples, a random number generator is used to produce a stream of non-negative integers, $Z_0, Z_1, Z_2, ..., Z_n$. This stream of integers can be thought of as on a circle, resulting in there being a finite number of integers that can be generated before the cycle repeats (*i.e.*, $Z_i = Z_0$). This number is known as the *period* of the random number generator. Different types of random number generators have different periods.

4.1 RNG in *ns*-2

Some simulation and data analysis packages still use a form of the linear congruential generator (LCG) with a period of 2^{31}, which has been shown to be inappropriate given the computational power of today's computers (L'Ecuyer 2001). We added an implementation (L'Ecuyer et al. 2002) of the MRG32k3a combined multiple recursive generator (L'Ecuyer 1999) to the popular *ns*-2 network simulator. This RNG has been the default in *ns*-2 since version ns-2.1b9, released in April 2002.

The implementation of the MRG32k3a generator in *ns*-2 provides several useful features for those performing simulations. The implementation divides the period into 2^{64} *streams*, each of which consists of 2^{51} *substreams*. Each of these substreams has a period of 2^{76}, resulting in a total period of approximately 2^{191}. The relationship between streams and substreams is shown in Figure 4. Each of the streams is guaranteed to be independent of every other stream, and each substream is guaranteed to be
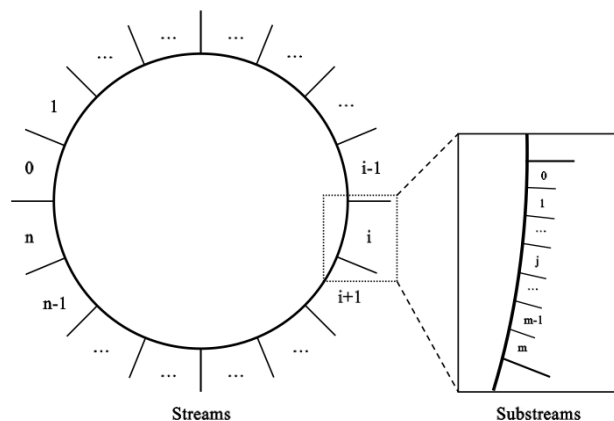
Figure 4: RNG Streams and Substreams in *ns*-2

independent of every other substream. The implication of this is that users of *ns*-2 no longer need to explicitly set a seed for each RNG used in a simulation. Once the seed of the `defaultRNG` is set, each subsequent call to create a new RNG object (`[new RNG]`) moves the generator to the next stream, thus guaranteeing independence. To perform multiple replications of simulations with multiple random variables (such as generating packet sizes and packet arrival times), each random variable should use a different stream, and each replication should use a different substream (L'Ecuyer et al. 2002). So, the ith replication should use the ith substream of each stream. Figure 5 shows an example *ns*-2 script that generates the third replication of output from two random variables, and Figure 6 shows the output when the script is run. We contributed a version of this example to the *ns-2* Manual (Fall and Varadhan 2006).

4.2 Effects of Improper RNG Usage

One of the noticeable effects of improper RNG usage is decreased reproducibility. We use the PackMime-HTTP traffic generator in *ns*-2 as an example. PackMime-HTTP uses random variable distributions to choose the HTTP request size, HTTP response size, time between the start of new connections, and the server delay time (time between the server receiving a request and sending the response). We also have random variables that determine the round-trip time (RTT) of each connection. The request size, response size, and next connection start time are all drawn when a new connection is started. The server delay time is not drawn until the HTTP request reaches the server. If we were using separate RNGs for each distribution, as described in the previous section, there would be no problem with repeatability, because samples from each distribution would be drawn from their own independent substream. If all of the random variables used the same RNG, a change in

```
# set the replication number (3rd replication)
set run 3

# seed the default RNG
global defaultRNG
$defaultRNG seed 9999

# create the RNGs
set arrivalRNG [new RNG]
set sizeRNG [new RNG]

# set the RNGs to the appropriate substream
for {set j 1} {$j < $run} {incr j} {
    $arrivalRNG next-substream
    $sizeRNG next-substream
}

# arrival_ is an exponential random variable
# describing the time between consecutive
# packet arrivals
set arrival_ [new RandomVariable/Exponential]
$arrival_ set avg_ 5
$arrival_ use-rng $arrivalRNG

# size_ is a uniform random variable describing
# packet sizes
set size_ [new RandomVariable/Uniform]
$size_ set min_ 100
$size_ set max_ 5000
$size_ use-rng $sizeRNG

# print the first 5 inter-arrival times and sizes
for {set j 0} {$j < 5} {incr j} {
    puts "[$arrival_ value] [$size_ value]"
}
```

Figure 5: Example Use of *ns*-2 RNG Functions

```
% ns rng-example.tcl

2.515083e+00  1.119200e+03
3.153944e+00  3.118043e+03
9.672774e+00  1.200644e+03
1.334614e+01  2.515018e+03
7.051881e+00  2.114656e+03
```

Figure 6: Output of *ns*-2 RNG Example

network settings (*e.g*, router queue size, link speed, RTT) could affect the sizes of the requests and responses drawn and used in the simulation.

To demonstrate this, we ran four simulations using the PackMime-HTTP traffic generator with the same basic setup as described in Section 2.2:

- An uncongested network (100 Mbps) with proper use of the RNG (separate streams for each random variable and separate substreams for each replication);
- A congested network (7.5 Mbps) with proper use of the RNG;
- An uncongested network with improper use of the RNG (only a single substream used for all random variables);
- A congested network with improper use of the RNG.

The only difference between the uncongested and congested cases is the link speed. Ideally, network settings should not affect the sizes of requests or responses or the times at which the connections are initiated. We show the HTTP response sizes and the times the connections were initiated for the uncongested and congested cases with the proper use of the RNG in Figure 7 and with improper use in Figure 8. Since there were a very large number of responses less than 100 KB, we show only those greater than 100 KB in order for the graphs to be readable. As expected, there is a perfect match of response sizes when the RNG is used properly. When samples from all distributions are drawn from the same RNG substream, the same response sizes are not drawn once network conditions change. Just by random chance, the congested case has no responses larger than 10,000 KB, while the uncongested case has three such responses. As shown in Section 2.2, the response sizes can have a large impact on the offered load in the network, so the offered load would likely be very different between these two cases.

5   CONCLUSIONS

Network simulations that employ heavy-tailed workloads have become more prevalent as more researchers study Internet traffic. We have discussed some of the high variability present in heavy-tailed distributions. We have also demonstrated how a single large sample from a heavy-tailed file size distribution can greatly affect offered loads on a network. Unfortunately, due to this high variability, standard statistical output analysis is often unsuitable. We presented three different approaches to dealing with heavy-tailed workloads. In addition, we discussed the impact of improper use of the RNG available in the *ns*-2 simulator and examples for its proper usage. All *ns*-2 simulation scripts used in this
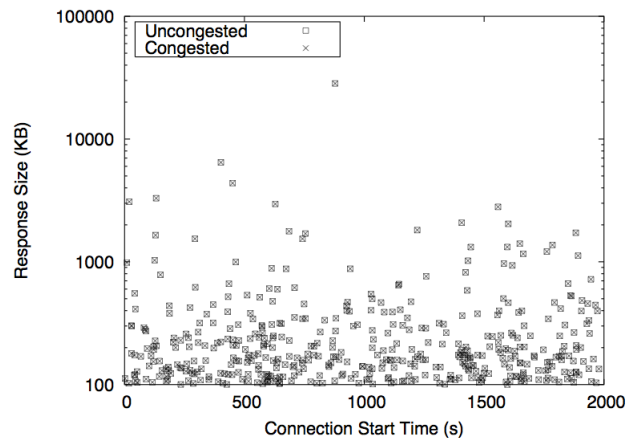

Figure 7: HTTP Response Sizes with Proper Use of the RNG in Uncongested and Congested Environments
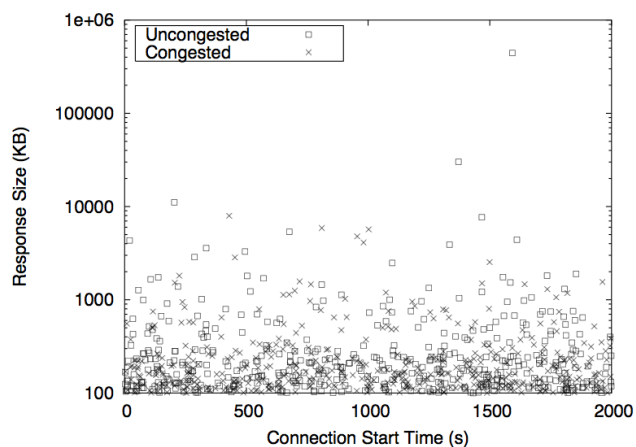

Figure 8: HTTP Response Sizes with Improper Use of the RNG in Uncongested and Congested Environments

paper are available at `<www.cs.odu.edu/~mweigle/ research/netsim/>`.

REFERENCES

Alexopoulos, C., and A. F. Seila. 2001. Output data analysis for simulations. In *Proceedings of the Winter Simulation Conference*, 115–122.

Alparslan, O., N. Akar, and E. Karasan. 2004. AIMD-based online MPLS traffic engineering for TCP flows via dis-

tributed multi-path routing. *Annales Des Telecommunications*.

Breslau, L., D. Estrin, K. Fall, S. Floyd, J. Heidemann, A. Helmy, P. Huang, S. McCanne, K. Varadhan, Y. Xu, and H. Yu. 2000, May. Advances in network simulation. *IEEE Computer* 33 (5): 59–67.

Cao, J., W. S. Cleveland, Y. Gao, K. Jeffay, F. D. Smith, and M. C. Weigle. 2004, March. Stochastic models for generating synthetic HTTP source traffic. In *Proceedings of IEEE INFOCOM*. Hong Kong.

Claypool, M., R. Kinicki, and M. Hartling. 2004, April. Active queue management for web traffic. In *Proceedings of IEEE International Performance, Computing, and Communications Conference (IPCCC)*.

Crovella, M., and L. Lipsky. 1997. Long-lasting transient conditions in simulations with heavy-tailed workloads. In *Proceedings of the Winter Simulation Conference*, 1005–1012. Atlanta, GA.

Fall, K., and K. Varadhan. (Eds.) 2006. *ns Manual*. <www.isi.edu/nsnam/ns/doc/ns_doc.pdf>.

Fishman, G. 2001. *Discrete-event simulation: Modeling, programming, and analysis*. Springer-Verlag.

Fishman, G. S., and I. J. B. F. Adan. 2006. How heavy-tailed distributions affect simulation-generated time averages. *ACM Transactions on Modeling and Computer Simulation*. To appear.

Harchol-Balter, M., M. Crovella, and C. Murta. 1999, November. On choosing a task assignment policy for a distributed server system. *Journal of Parallel and Distributed Computing, Special issue on software support for distributed computing* 59 (2): 204–228.

Houtzager, G., and C. Williamson. 2003. A packet-level simulation study of optimal web proxy cache placement. In *Proceedings of IEEE MASCOTS*.

Kurkowski, S., T. Camp, and M. Colagrosso. 2005, October. MANET simulation studies: The incredibles. *ACM SIGMOBILE Mobile Computing and Communications Review* 9 (4): 50–61.

Law, A. M. 2004. Statistical analysis of simulation output-data: The practical state of the art. In *Proceedings of the Winter Simulation Conference*, 67–72.

L'Ecuyer, P. 1999. Good parameters and implementations for combined multiple recursive random number generators. *Operations Research* 47 (1): 159–164.

L'Ecuyer, P. 2001. Software for uniform random number generation: Distinguishing the good and the bad. In *Proceedings of the Winter Simulation Conference*, 95–105.

L'Ecuyer, P., R. Simard, E. J. Chen, and W. D. Kelton. 2002, November. An object-oriented random-number package with many long streams and substreams. *Operations Research* 50 (6): 1073–1075. Software available at <www.iro.umontreal.ca/~lecuyer/myftp/streams00/>.

Mah, B. 1997. An empirical model of HTTP network traffic. In *Proceedings of IEEE INFOCOM*, 592–600.

May, M., T. Bonald, and J. Bolot. 2000, March. Analytic evaluation of RED performance. In *Proceedings of IEEE INFOCOM*, 1415–1424.

Nakayama, M. K. 2002. Simualtion output analysis. In *Proceedings of the Winter Simulation Conference*, 23–34.

Ott, T., T. Lakshman, and L. Wong. 1999, March. SRED: Stabilized RED. In *Proceedings of IEEE INFOCOM*, 1346–1355.

Park, K., G. Kim, and M. Crovella. 1996. On the relationship between file sizes, transport protocols, and self-similar network traffic. In *Proceedings of the IEEE International Conference on Network Protocols (ICNP)*.

Pawlikowski, K. 2003, February. Do not trust all simulation studies of telecommunications networks. In *Proceedings of the International Conference on Information Networking (ICOIN)*.

Pawlikowski, K., H.-D. J. Jeong, and J.-S. R. Lee. 2002, January. On credibility of simulation studies of telecommunication networks. *IEEE Communications Magazine* 40 (1): 132–139.

Perrone, L., Y. Yuan, and D. Nicol. 2003. Modeling and simulation best practices for wireless ad hoc networks. In *Proceedings of the Winter Simulation Conference*, 685–693.

Rai, I. A., G. Urvoy-Keller, and E. W. Biersack. 2003. Analysis of LAS scheduling for job size distributions with high variance. In *Proceedings of ACM SIGMETRICS*, 218–228.

Urvoy-Keller, G., and E. Biersack. 2002, October. A multicast congestion control model for overlay networks and its performance. In *Proceedings of the Workshop on Networked Group Communication*.

Weigle, M. C., K. Jeffay, and F. D. Smith. 2005, May. Delay-based early congestion detection and adaptation in TCP: Impact on web performance. *Computer Communications* 28 (8): 837–850.

Williamson, C., and N. Markatchev. 2003, July. Network-level impacts on user-level web performance. In *Proceedings of SPECTS*.

AUTHOR BIOGRAPHY

MICHELE C. WEIGLE is an Assistant Professor of Computer Science at Old Dominion University. She received her Ph.D. from the University of North Carolina at Chapel Hill in 2003. Her research interests include network protocol evaluation, network simulation and modeling, Internet congestion control, and mobile ad-hoc networks. She is a member of ACM and ACM SIGCOMM. Her e-mail address is <mweigle@cs.odu.edu>.