

PRACTICAL ASPECTS OF SIMULATING SYSTEMS HAVING ARRIVAL PROCESSES WITH LONG-RANGE DEPENDENCE

Robert Geist
James Westall

Department of Computer Science
Clemson University
Clemson, SC 29634, U.S.A.

ABSTRACT

Analysis of network traffic indicates that packet arrival processes have significant stochastic dependence. It has been suggested that this dependence is so strong as to be well-modeled by long-range dependent processes. The fact that no finite process can be said to be truly long-range dependent poses potentially serious obstacles to simulation modeling. In this paper, we explore some of these obstacles and propose practical methods for obtaining useful results with simulations of manageable duration. Keywords: Analysis methodology; network traffic models; stochastic dependence; long-range dependence.

1 INTRODUCTION

The foundational work of Leland, Taqqu, Willinger, and Wilson (1994) demonstrated the presence of significant stochastic dependence in LAN traffic. It is now well-established that i.i.d. processes are poor models of packet arrivals in both local and wide-area networks (Paxson and Floyd 1994) in that they can lead to serious underestimation of queuing delays (Erramilli et al. 1996).

To predict the behavior of system components when arrival processes have stochastic dependence, designers have therefore turned to simulation, and thus there is considerable interest in constructing synthetic arrival processes to drive the simulations. To ensure that the synthetic workloads are representative of those arrival processes that are likely to be encountered in the “real world”, the synthetic workloads are commonly derived from captured traces of actual arrival processes. To predict system performance accurately, the synthetic workload must reflect *both* the distributional characteristics and the stochastic dependence of the arrival process from which it was derived (Geist and Westall 2000).

Sampling from an empirical distribution is a simple and well-established procedure for ensuring that the marginal distribution of the synthetic process has the distributional characteristics of the arrival process from which it was de-

rived. An early, and still widely used, workload synthesizer, *tcplib* developed by Jamin and Danzig (1992) employs this approach.

1.1 Synthesis of Stochastic Dependence

Synthesizing stochastic dependence is more difficult. The autocorrelation function of a wide-sense stationary stochastic process $\{X_j\}_{j=0,1,2,\dots}$ having mean μ and variance σ^2 is given by:

$$r(k) = E[(X_j - \mu)(X_{j+k} - \mu)]/\sigma^2.$$

Intuitively, $r(k)$ is a normalized measure of the correlation between series elements that are a distance k apart. If the elements of the series are i.i.d., $r(k) = 0$ for $k \geq 1$. If elements at a distance k are identical, then $r(k) = 1$. Thus, the autocorrelation and its Fourier transform, the power spectrum, provide useful mechanisms for quantifying stochastic dependence.

Long-range dependence is a form of stochastic dependence that has received considerable interest in network traffic modeling. A wide-sense stationary process is said to be long-range dependent if the dependence decays very slowly, specifically, $\sum_k r(k) = \infty$. Asymptotically self-similar processes are a subclass of long-range dependent processes. They are characterized by hyperbolically decaying autocorrelation functions that satisfy $r(k) \propto k^{-(2-2H)}L(k)$ as $k \rightarrow \infty$ where $L(k)$ is a slowly varying function, and H is the so-called Hurst parameter. Values of H in the range (0.5, 1.0) are characteristic of processes having positively correlated long-range dependence. *Exactly self-similar* processes have autocorrelation functions that satisfy:

$$r(k) = (1/2)[(k+1)^{2H} - 2k^{2H} + (k-1)^{2H}]. \quad (1)$$

It is straightforward to show that exactly self-similar processes are also asymptotically self-similar.

There are a number of well-known techniques for synthesizing processes that possess stochastic dependence. Franklin (1965) provided a method for synthesizing Gaussian noise whose autocorrelation matched a prescribed $r(k)$ for a finite number of values k . Procedures for synthesizing long-range dependence were discovered much later, but they are now abundant and include: fractional ARIMA processes (Beran 1992); sampling from the population of an $M/G/\infty$ queuing system in which service times have infinite variance (Paxson 1994); synthesis of approximate fractional Gaussian noise (fGn) (Paxson 1997, Lau et al. 1995, Geist et al. 1999); and the aggregation of modulated packet-train processes in which the duration of the *ON* and *OFF* periods is heavy-tailed (Taquu et al. 1997).

1.2 Transfer of Stochastic Dependence

The problem of synthesizing a process that carries both the marginal distribution and the autocorrelation of a target process is more difficult still, but it has also received considerable attention. No existing procedure is entirely satisfactory. Present tradeoffs include: mathematical rigor, computational efficiency, accuracy of approximation, and precise and concise parameterization.

Transfer of correlation is a widely used procedure that dates at least to the work of Polge, Holliday, and Bhagavan (1973). The underlying idea is as follows. Suppose $\{x_k : k = 0, 1, \dots\}$ is a stochastic process having distribution function F_x and autocorrelation $r(k)$. Then the values, $\{u_k = F_x(x_k) : k = 0, 1, \dots\}$ are uniformly distributed in $[0.0, 1.0]$ and carry some representation of the correlation of the x_k . Given an arbitrary distribution, $F_y(y)$, if we now let $y_k = F_y^{-1}(u_k)$, then the values, $\{y_k : k = 0, 1, \dots\}$, *must* inherit the distribution $F_y(y)$ and will also inherit some form of the correlation from the u_k . The principal obstacle to the use of transfer of correlation is characterizing and compensating for the effect of the transfer on $r(k)$.

Wise, Traganitis, and Thomas (1977) showed that the effect of the transformation $g() = F_y^{-1}(F_x())$ on the autocorrelation $r(k)$ can be precisely quantified when the input process, $\{x_k\}$, is Gaussian. Building upon this work, Liu and Munson (1982) showed that under certain conditions it is possible to prescribe the autocorrelation, $r(k)$, of the initial Gaussian so that the resulting autocorrelation, $\hat{r}(k)$, of the process $F_y^{-1}(F_x(x))$ is equal to some desired target autocorrelation. They also show that for certain target distributions and autocorrelations there is *no* Gaussian from which the desired transfer of correlation is possible.

In Jagerman and Melamed's TES (Transform, Expand, Sample) processes (Jagerman and Melamed 1992, Melamed 1997) an interactive tool is used to construct a $U[0.0, 1.0]$ correlated process that is claimed to be able to transfer any desired $\hat{r}(k)$ to a target process. However, the generality of TES is both a strength and a weakness. The strength is that

any autocorrelation function can theoretically be matched. The weakness is that there is no known algorithmic procedure for deriving the innovation and stitching functions necessary to do so.

In an earlier paper (Geist and Westall 2000) we showed that transfer of correlation from fractional Gaussian noise (fGn) was a practical technique for synthesizing arrival processes having stochastic dependence. In the remainder of this paper we present some extensions and refinements to that approach.

Two arrival processes that are derived from traces of actual network traffic are used in this study. The traces capture number of packet arrivals per second. The first, *BCR*, is the BC-pAug89 archive captured at Bellcore's Morristown Research and Engineering Center. It is available at <http://ita.ee.lbl.gov/html/contrib/BC.html>. This trace represents almost one hour of LAN traffic captured on production Ethernet at Bellcore and is representative of the results reported by Leland (1994). The second, *R48*, represents two hours of inbound traffic captured in 1999 at the Clemson University Dept. of Computer Science's gateway to the remainder of the campus and the external internet. The characteristics of the data are summarized in table 1. The BCR data represents a total of about 1,000,000 packet arrivals. For the R48 data the number is 842,400.

Table 1: Workload characteristics

Parameter	BCR	R48
μ (pkts/sec)	318	117
σ	114	80
H	0.845	0.87
Length	3142	7200

The remainder of this paper is organized as follows. In section 2 we consider the problem of synthesizing Gaussian noise having sample autocorrelation that agrees statistically with some target $\hat{r}(k)$. Empirical analysis of the effects of transfer of correlation is presented in section 3. Effects on the performance of a simulated system are evaluated in section 4, and our results are summarized in section 5.

2 SYNTHESIS OF GAUSSIAN NOISE WITH STOCHASTIC DEPENDENCE

In this section we consider alternative approaches for synthesizing Gaussian noise that carries the stochastic dependence of a target arrival process. We begin with a discussion of Paxson's (1997) procedure for synthesizing fGn. A unique aspect of this approach is that it operates in Fourier transform space. We will show how to exploit this aspect of the method to synthesize Gaussian noise with general stochastic dependence.

2.1 Synthesis of Fractional Gaussian Noise

Paxson's procedure is derived from Beran's (1986) functional representation, $f(\lambda, H)$, of the power spectrum of an fGn process. He provides an approximate method for computing $f(\lambda; H)$, and the result is denoted $\tilde{f}(\lambda; H)$. The sequence of values $\{f_1, f_2, \dots, f_{N/2}\}$ where $f_j = \tilde{f}(\frac{2\pi j}{N}; H)$ thus approximates the power spectrum of an fGn process for frequencies $\{2\pi/N, \dots, \pi\}$. The discrete sequence $\{f_1, \dots, f_{N/2}\}$ is then stochastically transformed into the discrete Fourier transform (DFT) of an fGn process of N points. The first step in this transformation is to multiply the f_j 's by an exponential random variable with mean 1. This operation produces the periodogram of the self-similar time series. Next the complex series, $\{z_1, \dots, z_{N/2}\}$, where $z_i = \sqrt{f_i}e^{i\theta}$, is generated. The phase, θ , is randomly chosen from $[0, 2\pi]$. The second half of this series is completed by setting $f_{N/2+k}$ equal to the complex conjugate of $f_{N/2-k}$ so that it represents the discrete Fourier transform of a real-valued series. Approximate fGn is then obtained by applying the inverse DFT. It remains unproven that this procedure generates a Gaussian process. However, Paxson showed that for $H \leq 0.85$ the samples tested were statistically consistent with having been drawn from a Gaussian distribution. Using Whittle's estimator he confirmed that the stochastic dependence present in the synthetic processes was consistent with fGn having the target value of H for $H = \{0.50, 0.55, \dots, 0.95\}$. We have obtained similar results using Paxson's procedure.

2.2 Finite Sample Effects

The Hurst parameter provides a convenient, single-parameter mechanism for characterizing the stochastic dependence of an exactly self-similar process. However, the fact that the estimated Hurst parameter of a collection of synthetic samples matches a target H does not in itself guarantee that the same will be true of sample autocorrelations.

For a finite sample of N observations

$$\hat{R}(k) = \frac{1}{N-k} \sum_{j=1}^{N-k} (X_j - \hat{\mu})(X_{j+k} - \hat{\mu})/\hat{\sigma}^2, \quad (2)$$

where $\hat{\mu}$ and $\hat{\sigma}^2$ represent the sample mean and variance, is an asymptotically unbiased estimator of $r(k)$ (Kobayashi 1978). Although Paxson's method does an excellent job of matching a target Hurst parameter, synthetic series having length in the range useful for driving simulation studies exhibit values of $\hat{R}(k)$ that are, to a statistically significant degree, less than the values of $r(k)$ that they target. This observation, which is also true of methods based upon random midpoint displacement, is illustrated in figure 1. For this data, $H = 0.845$, and the target values of $r(k)$ are

derived from equation 1. The series labeled *fGn 64K* was obtained by use of Paxson's method to synthesize 64 series of 65536 elements each. For each lag, $k \in [1, \dots, 256]$, the mean of the 64 observed values of $\hat{R}(k)$ is plotted. For the series labeled *fGn 4K*, series of 4096 elements were synthesized.

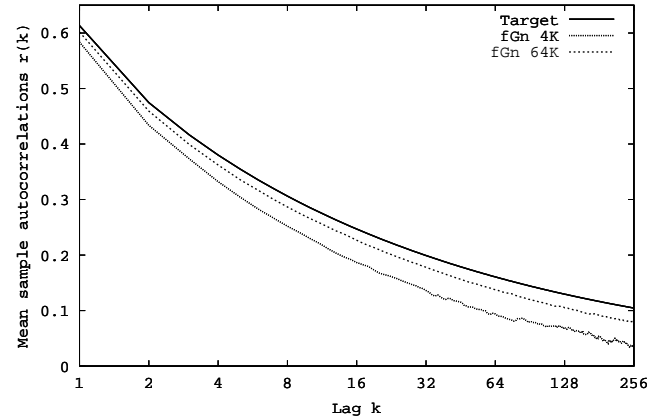


Figure 1: Target vs. Sample Autocorrelation

This data is presented in a more quantitative way in table 2. The column labeled $\Delta\mu$ contains the difference between the observed mean $\hat{R}(k)$ and the theoretical value of $r(k)$. The half-width of the two-sided confidence interval at the 90% level is contained in the columns labeled *C.I.* In no case does the confidence interval contain the target value.

Table 2: Difference in Autocorrelation

Lag	fGn 64K		fGn 4K	
	$\Delta\mu$	90% C.I.	$\Delta\mu$	90% C.I.
1	-0.0116	0.0031	-0.0282	0.0077
2	-0.0153	0.0040	-0.0393	0.0107
4	-0.0179	0.0048	-0.0463	0.0125
8	-0.0196	0.0053	-0.0527	0.0138
16	-0.0202	0.0055	-0.0588	0.0149
32	-0.0211	0.0057	-0.0615	0.0151
64	-0.0230	0.0059	-0.0652	0.0158
128	-0.0246	0.0064	-0.0583	0.0144
256	-0.0248	0.0064	-0.0636	0.0152

2.3 Synthesis of Gaussian Noise with Arbitrary Stochastic Dependence

The theory underlying Paxson's method indicates eventual convergence of $\hat{R}(k)$ to $r(k)$ when the length of the synthetic series becomes sufficiently large. However, the use of very long series to drive simulations has obvious disadvantages. In the remainder of this section we describe an extension to Paxson's method that supports synthesis of Gaussian noise having arbitrary stochastic dependence.

The Weiner-Khinchine transform (Kobayashi 1978) relates the autocorrelation function to the power spectrum, denoted $P(\lambda)$, for $-\pi \leq \lambda \leq \pi$.

$$P(\lambda) = \frac{1}{2\pi} \sum_{k=-\infty}^{k=\infty} r(k)e^{-i\pi\lambda} \quad (3)$$

$$= \frac{1}{2\pi} [r(0) + 2 \sum_{k=1}^{k=\infty} r(k)\cos(k\lambda)] \quad (4)$$

and

$$r(k) = \int_{-\pi}^{\pi} P(\lambda)\cos(k\lambda)d\lambda. \quad (5)$$

This transform provides the mechanism by which Paxson’s method may be extended so as to synthesize Gaussian noise having a target sample autocorrelation function $\hat{r}(k)$ for $\{k = 1, 2, \dots, K\}$. We first compute

$$P\left(\frac{2\pi j}{N}\right) = \frac{1}{2\pi} (\hat{r}(0) + 2 \sum_{k=1}^K \hat{r}(k)\cos\left(\frac{2\pi jk}{N}\right)) \quad (6)$$

for $j = 1, 2, \dots, N/2$. The values, $\{f_1, f_2, \dots, f_{N/2}\}$ where $f_j = P\left(\frac{2\pi j}{N}\right)$, are then stochastically transformed using Paxson’s procedure to yield approximate Gaussian noise having sample autocorrelation $\hat{R}(k)$ approximating $\hat{r}(k)$.

One application of this technique is to choose the target $\hat{r}(k)$ equal to the $r(k)$ associated with the fGn for $k = 0, \dots, K$ and $\hat{r}(k) = 0$ for $k > K$. The potential utility of this application lies in the observation that when N/K is sufficiently large, the sample autocorrelations, $\hat{R}(k)$, of the synthetic processes more closely approximate the target values, $r(k)$, for $k = 1, \dots, K$ than do those of synthetic fGn produced by any of the previously cited methods. Results of one comparison are shown in figure 2. In this example, 64 series of 4096 elements each were synthesized. The target $r(k)$ remains that of fGn with $H = 0.845$. However, for the series labeled *Gn16* values of $r(k)$ were clamped to 0 for $k > 16$ and the analogous procedure was applied to the series labeled *Gn256*.

The results are summarized quantitatively in table 3. In contrast to the results in table 2 the 90% confidence intervals cover the target value in the majority of cases.

For $K = 256$, $\hat{R}(k) < r(k)$ for all k . This is indeed evidence of a trend. After N/K becomes less than 16, the accuracy of the approximation degrades rapidly. When N/K reaches 2, accuracy of the finite approximation is worse than the original. This is to be expected since Paxson’s method uses a more accurate approximation of the $N/2$ elements of the power spectrum.

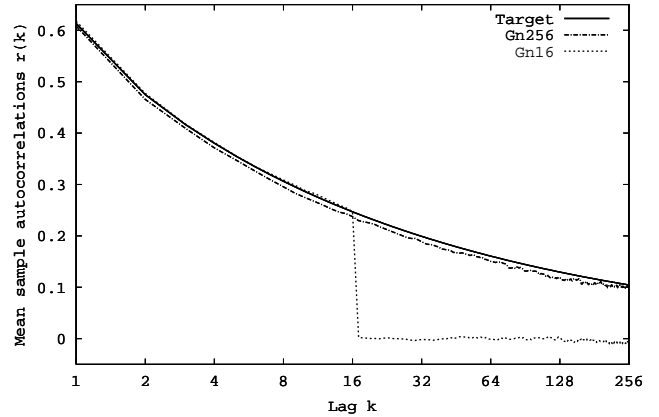


Figure 2: Sample Autocorrelation with Truncated $r(k)$

Table 3: Difference in Autocorrelation

Lag	Gn16		Gn256	
	$\Delta\mu$	90% C.I.	$\Delta\mu$	90% C.I.
1	0.0043	0.0039	-0.0051	0.0060
2	0.0026	0.0051	-0.0092	0.0083
4	0.0017	0.0054	-0.0090	0.0097
8	0.0025	0.0055	-0.0108	0.0103
16	0.0003	0.0056	-0.0094	0.0117
32	-0.2010	0.0422	-0.0106	0.0120
64	-0.1615	0.0342	-0.0103	0.0115
128	-0.1285	0.0277	-0.0124	0.0107
256	-0.1113	0.0242	-0.0021	0.0109

2.4 Matching Arbitrary Autocorrelations

Instead of targeting the autocorrelation of fGn, our procedure can also be used to generate approximate Gaussian noise whose autocorrelation targets the sample autocorrelation of a real arrival process. This application is illustrated in figure 3 and table 4. In this case the target $r(k)$ is the sample $\hat{R}(k)$ of the BCR workload. Once again the values plotted represent the means of 64 runs. The autocorrelations of the synthetic processes track their ragged target statistically well but there is an apparent trend toward low estimates. In section 4 we consider the implications of these results on the performance of simulated systems.

Table 4: Difference in Autocorrelation

Lag	Ac16		Ac256	
	$\Delta\mu$	90% C.I.	$\Delta\mu$	90% C.I.
1	-0.0022	0.0033	-0.0049	0.0052
2	-0.0031	0.0052	-0.0076	0.0072
4	-0.0046	0.0060	-0.0060	0.0086
8	-0.0029	0.0055	-0.0051	0.0094
16	-0.0001	0.0065	-0.0044	0.0102
32	-0.1997	0.0420	-0.0122	0.0105
64	-0.1906	0.0400	-0.0104	0.0099
128	-0.1206	0.0259	-0.0116	0.0099
255	-0.0403	0.0114	-0.0019	0.0088

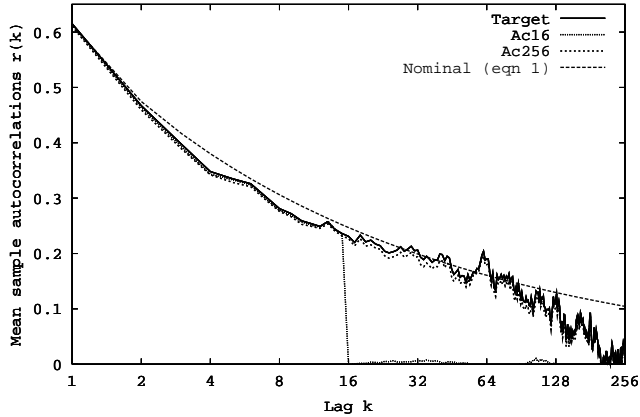
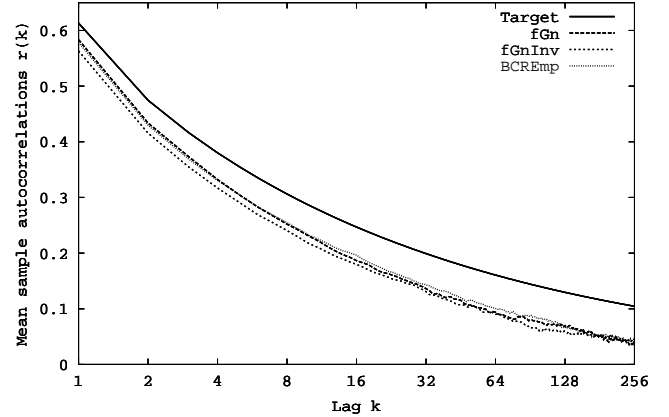
Figure 3: Sample Autocorrelation with Truncated $r(k)$ 

Figure 4: Transfer of fGn Correlation to BCR

3 TRANSFER OF CORRELATION

In this section we consider the effects of transfer of correlation from one stochastic process to another. Since the autocorrelation is invariant under linear transformation, no distortion is introduced when the function $g() = F_y^{-1}(F_x())$ is linear. It is also the case that $g()$ is linear if and only if $F_y()$ is a Gaussian. At the other extreme, it was shown by Liu and Munson (1982) that there exist target distributions, $F_y()$, and target autocorrelations, $r_y(k)$, for which there is no Gaussian process and autocorrelation, $r_x(k)$, that can yield the target $r_y(k)$. Intuitively, one might expect that the distortion introduced by the transfer of correlation is representative of some measure of the dissimilarity between F_y and a Gaussian, and this was shown to be the case by Wise, Traganitis, and Thomas (1977). They provide a precise characterization of the distortion via a measure that essentially captures the non-linearity of $g()$. This measure shows that no distortion occurs if and only if $g()$ is linear and that $r_y(k) \leq r_x(k)$ whenever $r_x(k) > 0$. In practice, we have found that the magnitude of coefficient of skewness of the distribution $F_y()$ is a useful first-order indicator of the degree of distortion.

Figure 4 illustrates the effect of transferring the correlation from approximate fGn processes of length 4096 through the marginal distribution of the BCR workload. The series labeled fGnInv is the uniformly distributed $F_z(x_i)$ where $\{x_i\}$ is fGn. The series labeled BCR_Emp is $F_{BCR}^{-1}(F_z(x_i))$. The effects of the transfer of correlation are apparent in the graph and in table 5. As before, the values shown are the means of 64 runs. Since our objective is to measure the effect of transfer of correlation, the $\Delta\mu$ values represent the difference from the mean sample autocorrelation of the fGn and *not* the nominal autocorrelation from equation (1).

The transfer from Gaussian to Uniform causes a small but statistically significant decrease in sample autocorrelation. However, the transfer from Uniform to the empirical

Table 5: Transfer of Autocorrelation

Lag	Uniform		BCREmp	
	$\Delta\mu$	90% C.I.	$\Delta\mu$	90% C.I.
1	-0.0203	0.0062	-0.0051	0.0043
2	-0.0183	0.0071	-0.0040	0.0059
8	-0.0115	0.0079	0.0029	0.0067
16	-0.0067	0.0074	0.0098	0.0069
32	-0.0055	0.0075	0.0071	0.0078
64	-0.0039	0.0073	0.0066	0.0079
128	-0.0105	0.0071	-0.0025	0.0075
256	0.0032	0.0082	0.0050	0.0078

distribution of the BCR workload compensates. It is obvious from the graph that the distortion introduced by the failure of the mean sample autocorrelation of the synthetic fGn processes to track the target $r(k)$ is significantly greater in magnitude than the distortion introduced by transfer of correlation.

An analogous transfer of correlation from Gn256 processes of length 4096 is shown in figure 5 and table 6. In this table $\Delta\mu$ *does* represent the difference between $\hat{R}(k)$ and $r(k)$ from equation (1). Although the values are low to a statistically significant degree, they are much closer than the mean sample autocorrelations of the synthetic fGn processes before the transfer of correlation.

Table 6: Transfer of Gn256 Autocorrelation

Lag	Uniform		BCREmp	
	$\Delta\mu$	90% C.I.	$\Delta\mu$	90% C.I.
1	-0.0257	0.0086	-0.0149	0.0071
2	-0.0278	0.0104	-0.0139	0.0095
4	-0.0266	0.0114	-0.0164	0.0112
8	-0.0279	0.0127	-0.0171	0.0116
16	-0.0258	0.0124	-0.0117	0.0123
32	-0.0219	0.0120	-0.0088	0.0121
64	-0.0185	0.0121	-0.0158	0.0120
128	-0.0202	0.0114	-0.0108	0.0106
256	-0.0215	0.0089	-0.0121	0.0095

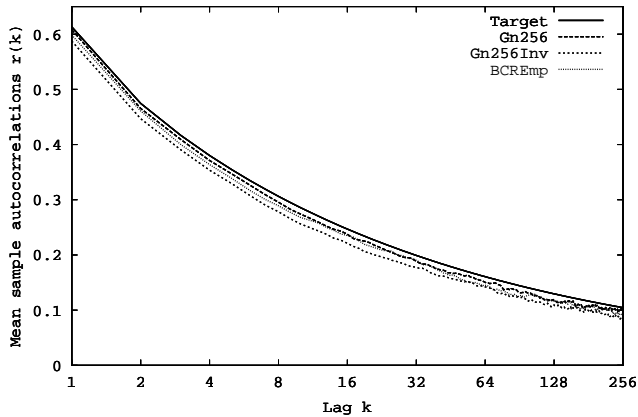


Figure 5: Transfer of Gn256 Correlation to BCR

Despite these promising results, transfer of correlation can significantly distort the autocorrelation. The R48 workload, shown in figure 6, illustrates this problem. The series labeled, *Target*, contains the equation 1 values for the Hurst parameter of 0.87. *Actual* is the computed $\hat{R}(k)$ for the 8192 values of the series. Gn256, Unif, and R48Emp were computed as with the BCR workload. Based upon a number of observations, it appears that the skewness of the target distribution is an indicator of the magnitude of the distortion of the autocorrelation. The BCR workload has a coefficient of skewness of 0.69 where R48 has 2.45.

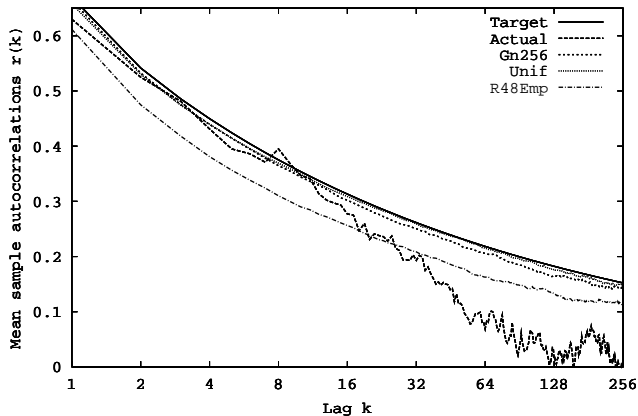


Figure 6: Transfer of Gn256 Correlation to R48

4 PERFORMANCE EFFECTS

In this section we evaluate the ability of workloads synthesized by the methods presented in the previous two sections to predict the performance of the workload from which they were derived. An FCFS server with deterministic service time and optionally constrained queue capacity is simulated. Given a target server utilization, ρ , and a known packet arrival rate, λ , mean service time, $1/\mu$, is set to ρ/λ . Mean population and packet drop rate are captured. Results reported for all workloads represent the mean value obtained

from 20 replications of the simulation. A unit time of one second is used. For the workloads referred to as *Count*, the simulation is driven by the true packet arrival counts. However, these results are not deterministic because the simulated arrival *times* of individual packets are synthetic and stochastic, (see (Geist and Westall 2000)).

Simulated run time for the *Count* workloads is identical to the length of the arrival count traces: 3142 seconds for the BCR workload; and 7200 for R48. For the synthetic workloads, elapsed simulated time is 4096 seconds and 8192 seconds for BCR and R48 respectively. In both cases the number of simulated packet arrivals per run is on the order of one million. For the synthetic workloads, both the arrival count and the arrival times are synthetic. Therefore, as expected, the confidence intervals are much wider than for the *Count* workloads.

If a previous paper (Geist and Westall 2000) we demonstrated the necessity of incorporating both distributional and correlational characteristics of the target workload in the synthetic workload. The focus here is upon the effects of different approaches to transfer of correlation.

4.1 fGn Synthesis Versus Gn256

We first compare the results obtained when driving the simulation with approximate fGn versus those obtained with Gn256. Recall that the 256 in Gn256 refers to the fact that only the first 256 lags of the target autocorrelation $r(k)$ are used in synthesizing the Gaussian noise from which the correlation is transferred. Simulations were run with bounded queue lengths ranging from 2^5 to 2^{11} . Results for both workloads with queue length constrained to a maximum of 256 are shown in tables 7 and 8. These results are representative. For most utilizations, the mean populations predicted by both the fGn and Gn256 workloads are smaller than those obtained using the target *Count* workload. However, in both of these cases the difference between fGn and Gn256 is not statistically significant.

Table 7: R48: Mean Pop. with Max Pop. = 256

Util	R48 Count	fGn	Gn256
0.25	1.36 $\bar{\pm}$ 0.002	1.36 $\bar{\pm}$ 0.019	1.39 $\bar{\pm}$ 0.022
0.30	3.16 $\bar{\pm}$ 0.004	2.98 $\bar{\pm}$ 0.061	3.02 $\bar{\pm}$ 0.071
0.35	6.03 $\bar{\pm}$ 0.007	5.51 $\bar{\pm}$ 0.083	5.71 $\bar{\pm}$ 0.091
0.40	9.53 $\bar{\pm}$ 0.008	9.00 $\bar{\pm}$ 0.085	9.32 $\bar{\pm}$ 0.112
0.45	14.27 $\bar{\pm}$ 0.009	13.06 $\bar{\pm}$ 0.096	13.39 $\bar{\pm}$ 0.154
0.50	18.96 $\bar{\pm}$ 0.011	17.95 $\bar{\pm}$ 0.135	18.25 $\bar{\pm}$ 0.196
0.55	24.13 $\bar{\pm}$ 0.008	23.52 $\bar{\pm}$ 0.224	24.15 $\bar{\pm}$ 0.272

When the population is unbounded, statistically significant differences emerge as shown in tables 9 and 10. These differences are more pronounced in the R48 workload, but there is a clear trend for lower predicted mean populations with fGn than for Gn256 as utilization increases. This

Table 8: BCR: Mean Pop. with Max Pop. = 256

Util	BCR Count	fGn	Gn256
0.40	1.02 \pm 0.004	1.00 \pm 0.0044	1.01 \pm 0.006
0.45	1.99 \pm 0.007	1.88 \pm 0.0124	1.90 \pm 0.018
0.50	3.85 \pm 0.010	3.70 \pm 0.0364	3.67 \pm 0.038
0.55	7.12 \pm 0.016	6.83 \pm 0.0646	7.01 \pm 0.091
0.60	12.65 \pm 0.018	12.29 \pm 0.1322	12.50 \pm 0.163
0.65	20.52 \pm 0.021	19.91 \pm 0.2038	20.18 \pm 0.199
0.70	30.62 \pm 0.026	30.29 \pm 0.2659	30.76 \pm 0.377

Table 9: R48: Mean Pop. with Pop. Unbounded

Util	R48 Count	fGn	Gn256
0.25	2.45 \pm 0.003	1.56 \pm 0.079	2.23 \pm 0.242
0.30	5.85 \pm 0.005	4.77 \pm 0.672	4.84 \pm 0.372
0.35	13.83 \pm 0.006	8.91 \pm 0.707	11.65 \pm 1.394
0.40	32.01 \pm 0.005	19.71 \pm 1.965	33.68 \pm 9.470
0.45	71.95 \pm 0.011	40.71 \pm 8.140	58.03 \pm 9.687

Table 10: BCR: Mean Pop. with Pop. Unbounded

Util	BCR Count	fGn	Gn256
0.40	1.02 \pm 0.005	1.00 \pm 0.0044	1.00 \pm 0.005
0.45	1.98 \pm 0.007	1.93 \pm 0.0244	1.93 \pm 0.021
0.50	4.27 \pm 0.008	3.90 \pm 0.0844	3.88 \pm 0.083
0.55	9.79 \pm 0.013	8.12 \pm 0.5366	8.76 \pm 0.653
0.60	21.52 \pm 0.018	16.87 \pm 1.0222	18.31 \pm 1.101
0.65	45.77 \pm 0.020	39.26 \pm 2.1988	49.90 \pm 11.855

trend is consistent with our observation in section 2 that $\hat{R}_{fGn}(k) < \hat{R}_{Gn256}(k)$ also in a statistically significant way.

4.2 Gn256 Synthesis versus Ac256

We now evaluate the effect of deriving the power spectrum from values of $r(k)$ taken from equation (1) versus deriving it from the sample autocorrelation, $\hat{R}(k)$, of the true arrival counts. The results for queue length bounded at 256 are shown in tables 11 and 12. For the BCR workload, the 90% confidence intervals overlap at all utilizations. For R48, a small but statistically significant trend emerges. Mean populations predicted by Gn256 are consistently larger than those of Ac256 and more accurately predict the behavior of the workload from which they were derived.

The results for unbounded population appear in tables 13 and 14. For the BCR workload there is again no statistically significant difference in the results. For R48, there is again an apparent trend for Gn256 to predict slightly higher mean populations but it is not statistically significant.

This slightly better performance of Gn256 is actually quite *good* news. It indicates that the Hurst parameter H is indeed a very useful single parameter mechanism for capturing the essence of the stochastic dependence present in the workload.

Table 11: R48: Mean Pop. with Max Pop. = 256

Util	R48 Count	Ac256	Gn256
0.25	1.36 \pm 0.002	1.33 \pm 0.019	1.39 \pm 0.022
0.30	3.16 \pm 0.004	2.92 \pm 0.061	3.02 \pm 0.071
0.35	6.03 \pm 0.007	5.45 \pm 0.083	5.71 \pm 0.091
0.40	9.53 \pm 0.008	8.95 \pm 0.085	9.32 \pm 0.112
0.45	14.27 \pm 0.009	13.20 \pm 0.096	13.39 \pm 0.154
0.50	18.96 \pm 0.011	17.79 \pm 0.135	18.25 \pm 0.196
0.55	24.13 \pm 0.008	23.51 \pm 0.224	24.15 \pm 0.272

Table 12: BCR: Mean Pop. with Max Pop. = 256

Util	BCR Count	Ac256	Gn256
0.40	1.02 \pm 0.004	1.01 \pm 0.004	1.01 \pm 0.006
0.45	1.99 \pm 0.007	1.89 \pm 0.017	1.90 \pm 0.018
0.50	3.85 \pm 0.010	3.70 \pm 0.038	3.67 \pm 0.038
0.55	7.12 \pm 0.016	7.02 \pm 0.073	7.01 \pm 0.091
0.60	12.65 \pm 0.018	12.26 \pm 0.146	12.50 \pm 0.163
0.65	20.52 \pm 0.021	19.98 \pm 0.217	20.18 \pm 0.199
0.70	30.62 \pm 0.026	30.11 \pm 0.304	30.76 \pm 0.377

Table 13: R48: Mean Pop. with Pop. Unbounded

Util	R48 Count	Ac256	Gn256
0.25	2.45 \pm 0.003	1.80 \pm 0.147	2.23 \pm 0.242
0.30	5.85 \pm 0.005	5.56 \pm 0.777	4.84 \pm 0.372
0.35	13.83 \pm 0.006	11.57 \pm 1.115	11.65 \pm 1.394
0.40	32.01 \pm 0.005	32.72 \pm 8.778	33.68 \pm 9.470
0.45	71.95 \pm 0.011	56.03 \pm 5.546	58.03 \pm 9.687

Table 14: BCR: Mean Pop. with Pop. Unbounded

Util	BCR Count	Ac256	Gn256
0.40	1.02 \pm 0.005	1.00 \pm 0.004	1.00 \pm 0.005
0.45	1.98 \pm 0.007	1.92 \pm 0.032	1.93 \pm 0.021
0.50	4.27 \pm 0.008	3.80 \pm 0.069	3.88 \pm 0.083
0.55	9.79 \pm 0.013	8.69 \pm 0.720	8.76 \pm 0.653
0.60	21.52 \pm 0.018	17.89 \pm 0.916	18.31 \pm 1.101
0.65	45.77 \pm 0.020	46.86 \pm 7.908	49.90 \pm 11.855

4.3 Bounded Queue Length Effects

In this section we present evidence indicating that bounded queue lengths are very effective in ameliorating the negative effects of long-range dependence in the arrival process. All routers and switches in IP or ATM networks drop packets or cells when queues become excessively long. This result is therefore applicable to the design of such systems since it indicates that the behavior of queue length limited systems depends primarily on the short-range dependence in the workload.

This behavior is illustrated in table 15 for the BCR workload with queue length limited to 256. For the column labeled Ac4, the power spectrum was derived using only $\hat{R}(0), \dots, \hat{R}(3)$ from the sample autocorrelation function of the BCR data with all other values clamped to 0. Nevertheless, its 90% confidence intervals overlap those of Ac256

for all utilization except 0.70. The results obtained with Ac16 are statistically indistinguishable from those obtained with Ac256.

Table 15: BCR: Mean Pop. with Max Pop. = 256.

Util	Count	Ac256	Ac2	Ac4	Ac16
0.40	1.02	1.01	1.00	1.00	1.00
0.45	1.99	1.89	1.88	1.91	1.89
0.50	3.85	3.70	3.56	3.66	3.70
0.55	7.12	7.02	6.57	6.89	6.96
0.60	12.65	12.26	11.39	12.15	12.39
0.65	20.52	19.98	18.28	19.84	20.35
0.70	30.62	30.11	27.84	29.53	30.23

The contrasting behavior for unbounded queues is shown in table 16. As utilization increases it is the case that dependence up to a lag of 256 is having a significant impact on system behavior.

Table 16: BCR: Mean Pop. with Pop. Unbounded.

Util	Count	Ac256	Ac2	Ac4	Ac16
0.40	1.02	1.00	1.00	1.00	1.01
0.45	1.98	1.92	1.88	1.92	1.95
0.50	4.27	3.80	3.61	3.84	3.91
0.55	9.79	8.69	6.86	8.11	8.84
0.60	21.52	17.89	12.41	15.45	18.36
0.65	45.77	46.86	21.89	28.94	39.33

5 CONCLUSIONS

Although it is a fact of life that *proof by simulation* is not possible, we believe that the techniques and results presented here make useful contributions to the modeling and understanding of systems having arrival processes with stochastic dependence. We have introduced a simple and efficient procedure for synthesis of Gaussian noise having arbitrary stochastic dependence. This procedure was used to demonstrate that the sample autocorrelation of a GnK process of N elements more closely approximates a target $r(k)$ for $k \leq K$ than does fGn when the N/K is sufficiently large, and that the use of GnK produces slightly better estimates of mean population than does fGn.

We provided evidence indicating that whether correlated Gaussian noise is derived from the sample $\hat{R}(k)$ or the theoretical $r(k)$ associated with the Hurst parameter of the target process does not usually affect mean population to a statistically significant degree. This observation indicates that the dependence in real network traffic arrivals *can* be usefully characterized by the single parameter H .

Finally, an example was presented indicating that when maximum queue length is bounded by some Q , then there is a value $K_{crit} \ll Q$ such that predicted mean population does not vary significantly for GnK sources of correlated

Gaussian noise for $K \geq K_{crit}$. This indicates that much of the negative impact of long-range dependence is effectively negated by simply bounding queue length.

REFERENCES

- Beran, J. 1986. *Estimation, testing, and prediction for self-similar and related processes*. Ph.D. dissertation. ETH, Zurich.
- Beran, J. 1992. Statistical methods for data with long-range dependence. *Statistical Science* 7(4): 404–427.
- Danzig, P., S. Jamin, R. Caceres, D. Mitzel, and D. Estrin. 1992. An empirical workload model for driving wide-area TCP/IP network simulations. *Journal of Internetworking* 3(1): 1–26.
- Erramilli, A., O. Narayan, and W. Willinger. 1996. experimental queueing analysis with long-range dependent packet traffic. *IEEE/ACM Trans. on Networking* 4(2): 209–223.
- Franklin, J. 1965. Numerical simulation of stationary and non-stationary gaussian random processes. *SIAM Review* 7(1): 68–82.
- Geist, R., K. Spicer, and J. Westall. 1999. Simulation modeling of self-similarity in network traffic. In *Proceedings of CMG '99*.
- Geist R., and J. Westall. 2000. Correlational and distributional effects in network traffic models. To appear in *Proceedings of IPDS 2000*.
- Jagerman, D., and B. Melamed. 1992. The transition and autocorrelation structure of TES processes part I: General theory. *Stochastic Models* 8(2): 193–219.
- Kobayashi A. 1978. *Modeling and analysis*. Reading, MA.: Addison-Wesley.
- Lau, W-C., A. Erramilli, J. Wang, and W. Willinger. 1995. Self-similar traffic generation: The random midpoint displacement algorithm and its properties. In *Proc. ICC '95*.
- Leland, W., M. Taqqu, W. Willinger, and D. Wilson. 1994. On the self-similar nature of ethernet traffic (extended version). *IEEE/ACM Trans. on Networking*, 2(1): 1–15.
- Liu, B., and D. Munson. 1982. Generation of a random sequence having a jointly specified marginal distribution and autocovariance. *IEEE Trans. on Acoustics, Speech, and Signal Processing* ASSP-30(6): 973–983.
- Melamed, B. 1997. The empirical TES methodology: Modeling empirical time series. *J. of Applied Mathematics and Stochastic Analysis* 10(4): 333–353.
- Paxson, V. 1997. Fast approximation of self-Similar network traffic. *ACM SIGCOMM Computer Communication Review* 27(5): 5–18.
- Paxson V. and S. Floyd. 1994. Wide-area traffic: The failure of poisson modeling. *Proceedings of SIGCOMM'94*,

ACM SIGCOMM Computer Communication Review
24(4): 257–268.

Polge R., E. Holliday, and B. Bhagavan. 1973. Generation of a pseudo-random set with desired correlation and probability distribution. *Simulation* 20(5): 153–158.

Taqqu, M., W. Willinger, and R. Sherman. 1997. Proof of a fundamental result in self-similar traffic modeling. *ACM SIGCOMM Computer Communication Review* 27(2): 5–23.

Wise, G., A. Traganitis, and J. Thomas. 1977. The effect of a memoryless nonlinearity on the spectrum of a random process. *IEEE Transactions on Information Theory* IT-23(1): 84–89.