International Journal of Applied Mathematics

Volume 29 No. 4 2016, 465-483

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

doi: http://dx.doi.org/10.12732/ijam.v29i4.5

IMPROVED GENERATOR OF LONG-MEMORY PROCESSES

M.E. Sousa-Vieira

E.E. Telecomunicación, University of Vigo
Vigo – 36310, SPAIN

Abstract: Simulations with long-memory input processes are hindered both by the slowness of convergence displayed by the output data and by the high computational complexity of the on-line methods for generating the input process. We present an optimized algorithm for simulating efficiently the occupancy process of the M/G/∞ system, which can be used as a sequential pseudorandom number generator of a broad class of long-memory sample paths. Our previous approach is the decomposition of the service time distribution as a linear combination of memoryless random variables, plus a residual term. Then, the original M/G/∞ system is replaced by a number of parallel, independent, virtual and easier to simulate M/G/∞ subsystems, the dynamics of which can be replicated sequentially or in parallel too. In this work we improve our previous algorithm, taking into account the generation time of the random variables.

AMS Subject Classification: 60G18, 62P30, 65C10

Key Words: persistent correlations, M/G/∞ process, efficient on-line generation

1. Introduction

Detailed observations of communications networks have revealed singular statistical properties of the measurements, such as self-similarity and long-range dependence, that cannot be overlooked in modeling Internet traffic. Self-similarity refers to the preservation of a common stochastic law describing the burstiness of traffic across different time scales, from seconds to days. Long-range dependence (LRD) means the existence of substantial correlations between the

Received: June 5, 2016

volume of traffic generated in widely separated time intervals, that is, a correlation structure decaying slowly in time. Both properties were discovered in Beran et al. [3] for variable-bit-rate (VBR) video traffic, in Paxson et al. [23] for traffic in wide area networks, in the landmark papers Leland et al. [14] and Willinger et al. [32] for Ethernet networks, and in Crovella et al. [6] for Web traffic, all these classical studies today. The finding of self-similarity and LRD, in turn, spurred the research on queueing models with "fractal" input traffic Norros [21] and Likhanov et al. [16], and the assessment of their impact on network performance Conti et al. [4] and Erramilli et al [9] as well. As these works demonstrate, the effect of LRD on packet loss and delay may be drastic, leading to subexponential decay of the buffer overflow probability and, consequently, buffer sizes much larger than those predicted by memoryless or short-memory processes.

Aside from the specific application in communications networks, self-similar processes are ubiquitous across many other fields in science and engineering Novak [22]. Prevalent examples appear in statistical physics (the diffusion of particles in ground, water or air), chemistry and biology (as in propagation of ions and molecules through cell membranes). They are also of interest as underlying models in time series analysis, like meteorological and hydrological data, stock markets data, rare events in finance and insurance, or biometric signals.

Several stochastic processes consistent with self-similarity and LRD have been proposed to model the behavior of traffic —or any other physical magnitude exhibiting fractal variability— in a tractable way, such as fractional Gaussian noise Norros [21], fractional ARIMA López et al. [18], the superposition of on-off sources with heavy-tailed distributions Willinger et al. [32] and the $M/G/\infty$ occupancy process Cox et al. [5] and Eliazar [8]. More recently, the multi-scale nature of traffic has also been successfully modeled with wavelet functions Abry et al. [1], Riedi et al. [26] and Ma et al. [19].

In the realm of simulation, the introduction of LRD models poses two fundamental problems related to the efficiency. The first concerns the statistical inference over the sample path data produced by discrete-event simulators, regardless of the type of process used as input. Aggregated LRD processes converge very slowly in probability or distribution, and the same holds for self-similar processes Li et al. [15], Livny et al. [17], Norros [20], Beran [2] and Fishman et al. [10]. Thus, extremely long simulation runs are generally needed to estimate a statistical parameter of the process with acceptable confidence level. The second problem is generating the input data with methods computationally efficient. Since simulations must produce a long series of samples, preferably with indefinite length, a long pseudo-random LRD sequence must be synthesized on demand. Unfortunately, because of the persistence of correlation, for some types of stochastic processes a sample X_n cannot be produced without knowing all the previous values $X_1, \ldots, X_{n-1}$, and typical algorithms for generating self-similar sequences exhibit no less than $O(n)$ complexity Paxson [24] and López et al. [18]. More importantly, drawing a new sample X_{n+1} fitted simultaneously to a given autocorrelation function and to a prescribed marginal distribution function, might require to repeat the n previous steps anew. So, the length should be determined in advance, perhaps in a very conservative way.

In view of the intrinsical invariance of the statistical properties of a self-similar process, that results in a high complexity for computing its trajectories, it turns out that efficient algorithms for generating traces of LRD processes are a key issue in simulation practice. In this paper, we focus on the M/G/∞ process as the base stochastic object for fast simulation of self-similar and LRD processes. This is the occupancy process of a M/G/∞ queueing system, and has several outstanding advantages both over the family of Gaussian processes (fBn, fGn and f-ARIMA) and over the wavelet-based approach. First, the process is theoretically simple and amenable to analysis. As such, there exists a substantial body of research results about the system's behavior Duffield [7], Tsoukatos et al. [30] and Resnick et al. [25]. Second, by varying the service time distribution, many different forms of the autocorrelation function can be obtained, either short-range or long-range dependent. Finally, and more importantly, the process can be trivially simulated in discrete time so as to obtain a sample path of length n with linear complexity, in an incremental and on-line method Krunz et al. [13], and Suárez et al. [29].

In our previous approach Sousa et al. [27] we decompose both the Poisson arrival process and the service times into computationally simpler subprocesses. To aid in the efficient generation of samples, the service time is approximated by a linear combination of memoryless random variables. In this way, we get an algorithm for fast generation of self-similar and LRD processes based on the efficient simulation of a M/G/∞ queue. The algorithm is generic, i.e., applicable to any distribution of the service times, including the subexponential and heavy-tailed classes; efficient, achieving a substantial reduction in the computing effort; on-line, in that the samples are generated individually and incrementally.

In this work we have improved our previous algorithm taking into account the generation times of the different random variables.

The remainder of the paper is organized as follows. In Section 2 we describe the M/G/∞ process. In Section 3 we remember briefly our previous algorithm. Next, in Section 4 the improved version is proposed and evaluated to quantify the savings in running time, compared to the previous implementation. Concluding remarks are given in Section 5.

2. The M/G/∞ Queue

The M/G/∞ process Cox et al. [5] is a stationary version of the occupancy process of an M/G/∞ queueing system. In this queueing system, customers arrive according to a Poisson process, occupy a server for a random time with a generic distribution X, the service time distribution, with finite mean, and leave the system.

Though the system operates in continuous time, it is easier to simulate it in discrete-time, so this will be the convention henceforth Suárez et al. [29]. The number of busy servers at time $t \in \mathbb{Z}^+$ is

$$Y_t = \sum_{i=1}^{\infty} A_{t,i}, \tag{1}$$

where $A_{t,i}$ is the number of arrivals at time $t-i$ which remain active at time t, i.e., the number of active customers with age i. For any fixed t, $\{A_{t,i}, i = 1, \dots\}$ are a sequence of independent Poisson variables with parameter $\lambda \mathbb{P}(X \geq i)$, where λ is the rate of the arrival process. The expectation and variance of the number of servers occupied at time t is

$$\mathbb{E}(Y_t) = \mathrm{Var}(Y_t) = \lambda \sum_{i=0}^{\infty} \mathbb{P}(X \geq i) = \lambda \mathbb{E}(X).$$

The discrete-time process $Y_t, t = 0, 1, \dots$ is time-reversible and wide-sense stationary, with autocovariance function

$$\gamma(h) = \mathrm{Cov}(Y_{t+h}, Y_t) = \lambda \sum_{i=h+1}^{\infty} \mathbb{P}(X \geq i), \quad h = 0, 1, \dots$$

Note that the function $\gamma(h)$ determines completely the expected service time

$$\mathbb{E}(X) = \frac{\gamma(0)}{\lambda}$$

and the distribution of X, the service time, because

$$\mathbb{P}(X = i) = \frac{\gamma(i-1) - 2\gamma(i) + \gamma(i+1)}{\lambda}, \quad i = 1, 2, \dots. \tag{2}$$

By (2), the autocovariance is a non-negative convex function. Alternatively, any real-valued sequence $\gamma(h)$ can be the autocovariance function of a discrete-time M/G/∞ occupancy process if and only if it is decreasing, non-negative and integer-convex Krunz et al. [13]. In such a case, $\lim_{h\to\infty}\gamma(h)=0$ and the probability mass function of X is given by (2).

If $A_{0,0}$, i.e., the initial number of customers in the system, follows a Poisson distribution with mean $\lambda\mathbb{E}(X)$, and their service times have the same distribution as the residual life $\widehat{X}$ of the random variable X,

$$\mathbb{P}\big(\widehat{X}=i\big)=\frac{\mathbb{P}(X\geq i)}{\mathbb{E}(X)},$$

then $\{Y_t, t=0,1,\dots\}$ is strict-sense stationary, ergodic, and enjoys the following properties

- the marginal distribution of Y_t is Poisson for all t, with mean value $\mu=\mathbb{E}(Y_t)=\lambda\mathbb{E}(X)$,
- the autocovariance function is $\gamma(h)=\gamma(0)\mathbb{P}\big(\widehat{X}>h\big)\forall h\geq 0$.

If the autocovariance function is non summable, $\sum_{i=0}^{\infty}\gamma(i)=\infty$, then the process exhibits LRD. In particular, this happens when X has infinite variance, as in the case of some heavy-tailed distributions. The latter are the discrete probability distribution functions satisfying $\mathbb{P}(X>k)\sim k^{-\alpha}$ asymptotically as $k\to\infty$. For exponents $1<\alpha<2$, $\mathbb{E}(X)$ is finite, but $\mathbb{E}(X^2)=\infty$ and it is easy to check that $\gamma(h)\sim k^{1-\alpha}$ asymptotically. Then, as shown in Tsybakov et al. [31], the occupancy process $\{Y_t, t=1,2,\dots\}$ is asymptotically second-order self-similar with Hurst parameter Hurst [12] $\mathsf{H}=(3-\alpha)/2$.

3. Simulating the M/G/∞ System Efficiently

In order to simulate the dynamics of a M/G/∞ system, the stationary representation (1) is not very useful. Instead, the evolution of the occupancy process can be better written as

$$Y_t=Y_{t-1}+A_t-D_t,\qquad t=1,2,\dots \tag{3}$$

where A_t, $t=1,2,\dots$ is a sequence of iid Poisson random variables with rate λ and D_t denotes the number of departures at the time t. The system state is simply the value of Y_t and the list $(\widehat{X}_1,\dots,\widehat{X}_{Y_t})$ of residual times for each customer. Therefore, the direct simulation of (3) requires, for every new value of t, the generation of

	$r(1) = 0.1$	$r(1) = 0.5$	$r(1) = 0.9$
$\mu = 4$	4.6	3	1.4
$\mu = 64$	58.6	33	7.4
$\mu = 1024$	922.6	513	103.4
$\mu = 16384$	14746.6	8193	1639.4

Table 1: Direct generation of the M/G/∞ process; $\mathbb{E}(N) = \lambda + 1$.

- one sample of A_t,
- A_t samples of the random variable X.

If N denotes the random number of values that have to be generated for each single new value of Y_t, then

$$\mathbb{E}(N) = 1 + E(A_t) = 1 + \lambda$$

so, for large values of λ the procedure may be grossly inefficient.

In most situations of interest, not only the autocovariance function $\gamma(h)$ must fit the empirical data, but also the marginal distribution of Y has to match a given pattern, predefined or inferred from collected traces. Since Y is Poisson, the change of distribution is rather to be applied when the mean number of busy servers in the M/G/∞ system is large (> 1000). In that case, Y converges to a Gaussian distribution which is easier to transform. A large value for μ, in turn, demands for a large value of the arrival rate λ, especially if the first-lag autocorrelation factor $r(1) = \gamma(1)/\gamma(0)$ is low, because $\mathbb{E}(X)$ is then correspondingly low. Therefore, for practical use, the direct simulation of the M/G/∞ system leads to poor efficiency due to the need of drawing $1 + \lambda$ samples, in average, for every random value of Y. To get a rough idea, Table 1 lists the cost, in number of random values, of generating each point in a trajectory of Y, when the service time distributions are those presented in Appendix A.

In the following sections we remember our method for reducing N by several orders of magnitude.

3.1. Geometric Composition Method (GC)

Consider a M/G/∞ system with service times X geometrically distributed with parameter p. In a discrete-time simulation, every sample of the occupancy process Y_t is obtained after adding the instantaneous arrivals A_t, and subtracting the departures D_t happening at instant t

$$Y_t = Y_{t-1} + A_t - D_t, \qquad t = 1, 2, \ldots$$

By the memoryless property of the geometric distribution, the conditional distribution of D_t given Y_t is binomial, i.e., the number of customers leaving the system at t is obtained by sampling randomly and independently with probability p among the Y_{t-1} customers in service. Therefore, the only system's state variable is Y_{t-1} and each simulation step consist of the generation of only two random values: a Poisson random variable (A_t) and a (Y_{t-1}, p)-binomial random variable (D_t).

The idea of the geometric composition (GC) method is that of approximating the arbitrary distribution of the service times, X, by a linear combination of shifted geometric variables, plus a remainder term for the distribution's tail. This time, the generic X is represented as

$$X = \sum_{i=1}^{m} \alpha_i X_i + \beta R_m, \tag{4}$$

where $X_i = x_i + G_i$, $\{G_1, \ldots, G_m\}$ are a set of independent variables geometrically distributed, $\alpha_i > 0$ is a suitable scaling factor and x_i is a properly chosen time shift, for $i = 1, \ldots, m$. Under the condition $\sum_{i=1}^{m} \alpha_i + \beta = 1$, R_m is a random variable capturing the difference between X and the mixture of X_i's.

The decomposition in (4) is equivalent to the splitting of the arrival process into $m + 1$ independent with rates $\lambda\alpha_1, \ldots, \lambda\alpha_m$ and $\lambda\beta$. Customers in group $i = 1, \ldots, m$ demand a service time distributed as X_i, whereas customers in the last group demand R_m units of time.

In order to use the GC method, it suffices to implement the corresponding M/G/∞ subsystems in (4) in parallel. For groups $i = 1, \ldots, m$, let $Y_t^{(i)}$ the occupancy process yielded by customers with service time X_i. Then,

$$Y_t^{(i)} = Y_{t-1}^{(i)} + A_t^{(i)} - D_t^{(i)}, \qquad t = 1, 2, \ldots$$

is the stochastic dynamic equation to simulate, with $A_t^{(i)}$ denoting the arrivals at t and $D_t^{(i)}$ is the number of departures. An array of size x_i is needed for delaying the departures x_i units of time and, at each instant t, $D_t^{(i)}$ is binomially distributed with parameters $(Y_{t-1} - W_{t-1}, p_i)$, where W_{t-1} is the total number of customers in the array of delay. Figure 1 illustrates the procedure. The number of servers busy with customers from class $m + 1$ can be simulated in

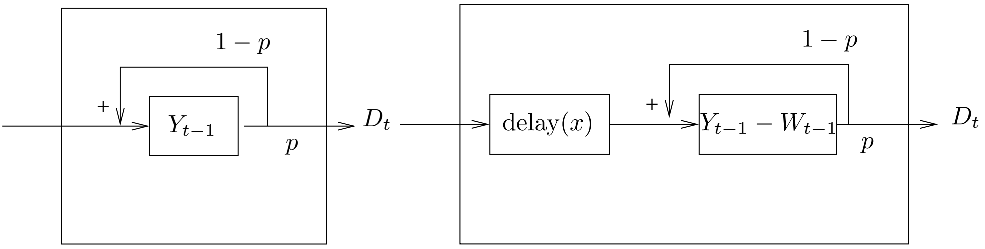

Figure 1: Generation of departures in an M/G/∞ system with geometric distribution (left) and shifted geometric distribution (right) for the service time.

the usual way, and the sample path of the system's occupancy is formed as

$$Y_t = \sum_{i=1}^{m+1} Y_t^{(i)},$$

i.e., as the sum of the individual outputs of each subsystem.

The computational cost of the GC method is the result of

- generating one sample for each input group, Poisson distributed with rates $\lambda\alpha_1, \ldots, \lambda\alpha_m$ and $\lambda\beta$,
- generating one sample for each binomial $D_t^{(i)}$, for $i = 1, 2, \ldots, m$,
- generating the service times of customers in group $m+1$, as iid samples from the residual random variable R_m.

So, the total amount of random numbers per sample value of Y_t is

$$N = 2m + 1 + A$$

with A a Poisson point process of rate $\lambda\beta$. On average

$$\mathbb{E}(N) = 2m + 1 + \lambda\beta = 2m + 1 + \lambda\big(1 - \sum_{i=1}^{m} \alpha_i\big).$$

The GC method leaves freedom for choosing $3m$ parameters, the triplet (α_i, x_i, p_i) of scaling factor, time-shift and exponential decay rate of each component X_i.

Finding the set of optimal parameters which minimizes N turns out to be a difficult nonlinear optimization problem. However, the fundamental observation is that, for very large λ, any procedure that yields $\sum_{i=1}^{m} \alpha_i \approx 1$, even a suboptimal one, actually captures to a large extent the efficiency gain due to the splitting of the arrival process. Moreover, it might be preferable to perform a simpler and faster algorithm to identify the unknown parameters than to spend a larger time searching for the optimal decomposition, if the improvement is only fractional with the later.

Based on these considerations, in Sousa et al. [27] we proposed a simple iterative algorithmic procedure to compute the number of groups m, the parameters of the random variables $\{X_i\}$ and the scaling factors $\{\alpha_i\}$ in the geometric approximation such that N is substantially lower in comparison to the direct generation of the discrete M/G/∞ process when λ increases. At every step, the goal of the algorithm is to find a new X_i such that α_i is large. Since minimizing N is equivalent to minimizing $2m - \sum_{i=1}^{m} \lambda\alpha_i$, the iterations should continue until $\lambda\alpha_i < 2$. Therefore, the stopping criterion guarantees a local minimum for the average number of samples. Moreover, the number m of Poisson subprocesses comes out as a result of the sequential execution, without having to be specified blindly in advance.

4. Improved Method

In the GC method we deal with three types of random variables, Poisson, binomial and the residual random variable.

In order to generate samples of the Poisson and residual random variables we have developed the classes `IntPoisson` and `IntR`, both extensions of the class `IntRandom`, that implements a generic tabular method to invert the distribution function of a non-negative discrete random variable Suárez et al. [29] and Sousa et al. [28].

And in order to generate samples of the Binomial random variable we have developed the class `Binomial`, that implements the method described in Appendix B.

We have measured the ratio between the generation time of a sample path of the Poisson random variable and that of a sample path of the residual random variable, f_P, obtaining a result that is practically independent of the parameters of the distributions. On the other hand, in order to get the ratio between the generation time of a sample path of the binomial random variable and that of a sample path of the residual random variable, f_B, we have tabulated the values measured for different combinations of the parameters of the binomial distribution, p and n (see Table 2). Then, for other values (remember that in the GC method the parameters of the binomial distributions are random numbers)

	$np = 0.5$	$np = 1$	$np = 10$	$np = 10^2$	$np = 10^3$
$p = 0.5$	1.7	2.4	4.8	4.3	3.9
$p = 0.05$	1.7	2.3	4.4	4.3	3.8
$p = 0.005$	1.7	2.4	4.4	4.3	3.8

Table 2: f_B

	$\mathsf{H} = 0.6$	$\mathsf{H} = 0.75$	$\mathsf{H} = 0.9$
$\mu = 256$	$5.654 \to 6.854$	$5.732 \to 6.242$	$4.725 \to 4.725$
$\mu = 512$	$6.309 \to 10.708$	$6.645 \to 9.484$	$6.027 \to 6.449$
$\mu = 1024$	$7.617 \to 7.617$	$7.738 \to 7.929$	$7.053 \to 9.899$
$\mu = 2048$	$8.289 \to 10.235$	$8.477 \to 10.859$	$8.121 \to 9.107$
$\mu = 4096$	$9.575 \to 15.471$	$9.954 \to 9.954$	$9.242 \to 13.215$

Table 3: Mean number of random values ($\mathbb{E}(N)$ previous algorithm $\to \mathbb{E}(N)$ improved algorithm).

f_B is obtained interpolating between the tabulated values. So, taking these ratios into account, the condition $\lambda\alpha_i < 2$ changes to $\lambda\alpha_i < f_P + f_B$.

In order to verify the improvement, we have generated samples of the M/G/∞ process, taking the S distribution (see Appendix A) for the service time and considering different combinations of the parameters H and $r(1)$, and we have compared the performance of the GC method considering both stopping conditions: $\lambda\alpha_i < 2$ (previous algorithm) and $\lambda\alpha_i < f_P + f_B$ (improved algorithm).

	$\mathsf{H} = 0.6$	$\mathsf{H} = 0.75$	$\mathsf{H} = 0.9$
$\mu = 256$	0.971	0.948	1
$\mu = 512$	0.975	0.975	0.925
$\mu = 1024$	1	0.955	0.975
$\mu = 2048$	0.956	0.936	0.934
$\mu = 4096$	0.959	1	0.938

Table 4: Ratio between the running time with the improved algorithm and the previous one.

In this case, we get the best improvements for low degree of short-term correlation, $r(1)$, and moderate mean value, μ.

In Tables 3 and 4 we show some examples for $r(1) = 0.1$. Table 3 compares the mean number of random values ($\mathbb{E}(N)$) to generate a sample of the M/G/∞ process and Table 4 compares the running times to generate a sample path. We can observe that although in some cases the mean number of random values to generate a sample increases, the running time always diminishes (ratio < 1) or remains the same (ratio 1).

5. Conclusions

In this paper, we have proposed an improved version of a generator of correlated traces based on the M/G/∞ process, able to deal with a wide range of input parameters, and flexible enough to model traffic with different statistical properties. Improving our previous algorithm taking into account the generation times of the different random variables, we have obtained running time reductions that in some cases are closed to 10%.

A. S Distribution

In a previous work Suárez et al. [29], we proposed to use a new discrete-time random variable built so that the derived M/G/∞ occupancy process displays long-range dependence. The distribution of this random variable is heavy-tailed, and has only two parameters that allow to tune, independently, the short-term and the long-term correlations.

Denoting the random variable by S, and the ensuing queueing model by M/S/∞, the probability mass function is given by

$$\mathbb{P}(S=k)=\begin{cases}1+\dfrac{x^{\alpha}}{\alpha x-x^{\alpha}}\left[(k+1)^{1-\alpha}-k^{1-\alpha}\right] & k=1\\ \dfrac{x^{\alpha}}{\alpha x-x^{\alpha}}\left[(k+1)^{1-\alpha}-2k^{1-\alpha}+(k-1)^{1-\alpha}\right] & \forall k>1,\end{cases}$$

for $x \leq 1$, and by

$$\mathbb{P}(S=k)=\begin{cases}1+k-x+\dfrac{x^{\alpha}}{\alpha-1}\left[(k+1)^{1-\alpha}-x^{1-\alpha}\right] & k=\lfloor x\rfloor\\ 1+x-k+\dfrac{x^{\alpha}}{\alpha-1}\left[(k+1)^{1-\alpha}-2k^{1-\alpha}+x^{1-\alpha}\right] & k=\lceil x\rceil\\ \dfrac{x^{\alpha}}{\alpha-1}\left[(k+1)^{1-\alpha}-2k^{1-\alpha}+(k-1)^{1-\alpha}\right] & \forall k>\lceil x\rceil\end{cases}.$$

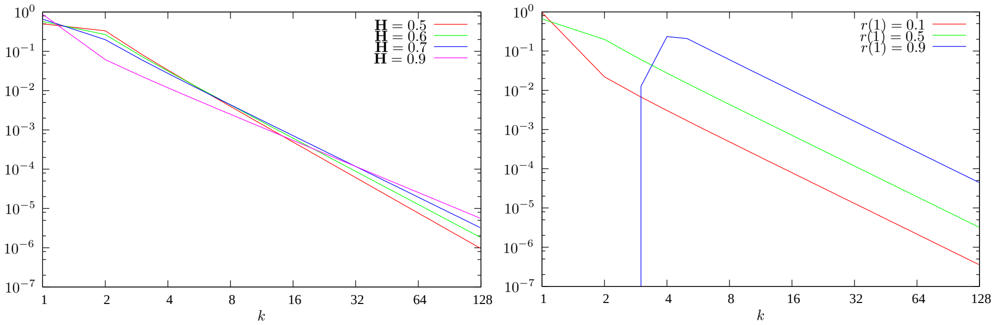

Figure 2: Probability mass function of S for $r(1) = 0.5$ (left) and $\mathsf{H} = 0.7$ (right).

for $x > 1$. The expected value is

$$\mathbb{E}(S) = \begin{cases} \dfrac{\alpha x}{\alpha x - x^{\alpha}} & \forall x \in (0, 1] \\ \dfrac{\alpha x}{\alpha - 1} & \forall x \geq 1. \end{cases}$$

and the resulting autocorrelation function can be written as

$$r(k) = \begin{cases} 1 - \dfrac{\alpha - 1}{x\alpha} k & \forall k \in [0, x] \\ \dfrac{1}{\alpha} \left(\dfrac{x}{k}\right)^{\alpha - 1} & \forall k \geq x. \end{cases}$$

Thus, for $k \to \infty$, the autocorrelation function decays as $k^{1-\alpha}$. If $\alpha \in (1, 2]$ then the Hurst parameter satisfies $\mathsf{H} \in [0.5, 1)$ and $\sum_{k=0}^{\infty} r(k) = \infty$. Hence, in this case, the process is LRD.

The parameters x (the "minimum" of the range of S) and α (the long-term decay rate of the correlation) are directly determined by the basic characterization of Y, the stationary occupancy process in the associated M/S/∞ system. So, if Y has to possess mean value μ, Hurst parameter H and a one-lag autocorrelation coefficient $r(1)$, then S must be parameterized with $\alpha = 3 - 2\mathsf{H}$ and

$$x = \begin{cases} \left(\alpha r(1)\right)^{\frac{1}{\alpha - 1}}, & r(1) \in \left(0, \alpha^{-1}\right] \\ \dfrac{1 - \alpha^{-1}}{1 - r(1)}, & r(1) \in \left[\alpha^{-1}, 1\right) \end{cases}$$

and the rate of the Poisson arrivals must be

$$\lambda = \begin{cases} \mu \dfrac{\alpha x - x^{\alpha}}{x\alpha} & \forall x \in (0,1] \\ \mu \dfrac{\alpha - 1}{x\alpha} & \forall x \geq 1. \end{cases}$$

Figure 2 depicts the form of the distribution in several representative cases of H and $r(1)$.

B. Binomial Distribution

In order to generate samples of the binomial random variable in a efficient way, we use the BTRD algorithm Hormann [11] combined with the BIN algorithm, provided that the first one is optimized for high mean values and the second one is fast enough in other cases.

The probability density function of the binomial random variable, B, with parameters p and n, is

$$\mathbb{P}(B = k) = \binom{n}{k} p^k (1-p)^{n-k} \quad \forall k > 0, \quad 0 \leq p \leq 1, \quad n \geq 1.$$

The BIN algorithm is based on the method of the inverse transformation, that for the binomial random variable can be implemented by using the recursive formula

$$\mathbb{P}(B = 0) = (1-p)^n,$$

$$\mathbb{P}(B = k) = \mathbb{P}(B = k-1) \frac{p(n-k+1)}{(1-p)k} \quad \forall k > 1.$$

The resulting pseudo-code is

```
1. q=1-p, sp/q, a=(n+1)*s, r=pow(q,n).
2. u=U(0,1), x=0.
3. If (u<=r) return x.
4. u=u-r, x=x+1, r=((a/x)-s)*r, go to 3.
```

The BTRD algorithm is based on the transformed rejection method, a combination of inversion and rejection that can be applied to various continuous distributions, included the binomial. Between all the algorithms based on this method to generate the binomial random variable that we have found available in the literature, this is the faster when the parameters of the binomial are random numbers and the mean value is high. Following, we explain it briefly.

When we use the rejection method to generate samples of a random variable with density function f, we need a dominating density or hat function h and a real number α, such that

$$f(x) \leq \alpha h(x) \quad \forall x.$$

In order to generate a sample of the desired random variable, we need to generate a sample of the dominating density, x, and a random number uniformly distributed between 0 and 1, v. If $v \leq \frac{f(x)}{\alpha h(x)}$, then x is accepted as a sample of the random variable with density function f and in other case it is rejected and the procedure starts again.

The transformed rejection method that uses the BTRD algorithm starts with $G(u)$, the inverse of the distribution function of the dominating density.

Due to $h(x) = (G^{-1})'(x) = \frac{1}{G'(u)}$, for $x = G(u)$ the condition of acceptance can be transformed into $v \leq \frac{f(G(u))G'(u)}{\alpha}$.

Let $m = \lfloor (n+1)p \rfloor$ be the mode of the binomial distribution and

$$f(x) = \frac{f_B(\lfloor x \rfloor)}{f_B(m)} = \frac{m!(n-m)!}{\lfloor x \rfloor!(n - \lfloor x \rfloor)!} \left(\frac{p}{1-p} \right)^{\lfloor x \rfloor - m}$$

the rescaled histogram, with $f(m) = 1$. To obtain high probability of acceptance, $G(u)$ should be close to the inverse distribution function of the random variable to be generated, in our case the binomial. The function

$$G(u) = \left(\frac{2a}{0.5 - |u|} + b \right) u + c, \quad -0.5 \leq u \leq 0.5$$

is very flexible and turns out to yield high probability of acceptance for a variety of distributions, including the binomial.

With these ideas, and trying to minimize the number of random numbers uniformly distributed and the number of evaluations of the condition of acceptance, the following pseudo-code is obtained for the TRD algorithm

```
1. v=U(0,1). If (v<=2*u_r*v_r) return floor(G(v/v_r-u_r)).
2. If (v>=v_r) u=U(-0.5,0.5),
   else u=v/v_r-(u_r+0.5), u=sign(u)*0.5-u, v=U(0,v_r).
3. If (v<=f(G(u))*G'(u)/alpha) return floor(G(u)),
   else go to 1.
```

Applied to the binomial distribution, optimal parameters for $np \geq 10$ and $p < 0.5$ are Hormann [11]

$$c = np + 0.5,$$

$$b = 1.15 + 2.53\sqrt{np(1-p)},$$
$$a = -0.0873 + 0.0248b + 0.01p,$$
$$\alpha = (2.83 + \frac{5.1}{b})\sqrt{np(1-p)},$$
$$u_r = 0.43,$$
$$v_r = 0.92 - \frac{4.2}{b}.$$

Finally, optimizing the evaluation of the condition of acceptance Hormann [11], the pseudo-code of the BTRD algorithm results

```
1.  [Set-up]
    f_c[0]=0.08106146679532726, f_c[1]=0.04134069595540929,
    f_c[2]=0.02767792568499834, f_c[3]=0.02079067210376509,
    f_c[4]=0.01664469118982119, f_c[5]=0.01387612882307075,
    f_c[6]=0.01189670994589177, f_c[7]=0.01041126526197209,
    f_c[8]=0.00925546218271273, f_c[9]=0.00833056343336287,
    f_c[k]=(1/12-(1/360-1/1260/pow(k+1,2))/pow(k+1,2))/(k+1)
    for (k>=10);
    m=floor((n+1)*p), r=p/(1-p), n_r=(n+1)*r,
    n_p_q=n*p*(1-p), b=1.15+2.53*sqrt(n_p_q),
    a=-0.0873+0.0248*b+0.01*p, c=n*p+0.5,
    alpha=(2.83+5.1/b)*sqrt(n_p_q), v_r=0.92-4-2/b,
    u_r_v_r=0.86*v_r.
2.  v=U(0,1).
    If (v<=u_r_v_r)
    u=v/v_r-0.43, return floor((2*a/(0.5-|u|)+b)*u+c).
3.  If (v>=v_r) u=U(-0.5,0.5),
    else u=v/v_r-0.93, u=sign(u)*0.5-u, v=U(0,v_r).
4.1.u_s=0.5-|u|, k=floor((2*a/u_s+b)*u+c).
    If ((k<0) or (k>n)) go to 2.
    v=v*alpha/(a/(u_s*u_s)+b), k_m=|k-m|.
    If (k_m>15) go to 4.3.
4.2.[Recursive evaluation of f[k]]
    f=1.
    If (m<k) i=m, repeat i=i+1, f=f*(n_r/i-r)
                  until (i==k),
    else if (m>k) i=k, repeat i=i+1, v=v*(n_r/i-r)
                         until (i==m).
    If (v<=f) return k,
    else go to 2.
4.3.[Squeeze acceptance or rejection]
    v=log(v),
    rho=(k_m/n_p_q)*(((k_m/3+0.625)*k_m+1/6)/n_p_q+0.5),
    t=-k_m*k_m/(2*n_p_q).
    If (v<t-rho) return k, if (v>t+rho) go to 1.
4.4.[Set-up for 4.5.]
    n_m=n-m+1,
    h=(m+0.5)*log((m+1)/(r*n_m))+f_c(m)+f_c(n-m).
4.5.[Final acceptance rejection test]
    n_k=n-k+1.
    If (v<=h+(n+1)*log(n_m/(n_k)+(k+0.5)*log(n_k*r/(k+1))-
    f_c[k]-f_c[n-k]) return k,
    else go to 2.
```

References

[1] P. Abry and D. Veitch, Wavelet analysis of long-range dependent traffic, *IEEE Transactions on Information Theory*, **44**, No 1 (1998), 2–15.

[2] J. Beran, *Statistics for Long-Memory Processes*, Chapman and Hall, New York (1994).

[3] J. Beran, R. Sherman, M. S. Taqqu and W. Willinger, Long-range dependence in variable-bit-rate video traffic, *IEEE Transactions on Communications*, **43**, Nos 2-4 (1995), 1566–1579.

[4] M. Conti, E. Gregori and A. Larsson, Study of the impact of MPEG-1 correlations on video sources statistical multiplexing, *IEEE Journal on Selected Areas in Communications*, **14**, No 7 (1996), 1455–1471.

[5] D. Cox and V. Isham, *Point Processes*, Chapman and Hall, New York (1980).

[6] M.E. Crovella and A. Bestavros, Self-similarity in world wide wed traffic: Evidence and possible causes, *IEEE/ACM Transactions on Networking*, **5**, No 6 (1997), 835–846.

[7] N. Duffield, Queueing at large resources driven by long-tailed M/G/∞ processes, *Queueing Systems*, **28**, Nos 1-3 (1987), 245–266.

[8] I. Eliazar, The M/G/∞ system revisited: Finiteness, summability, long-range dependence and reverse engineering, *Queueing Systems*, **55**, No 1 (2007), 71–82.

[9] A. Erramilli, O. Narayan and W. Willinger, Experimental queueing analysis with long-range dependent packet traffic, *IEEE/ACM Transactions on Networking*, **4**, No 2 (1996), 209–223.

[10] G. Fishman and I. J. Adan, How heavy-tailed distributions affect simulation generated time averages, *ACM Transactions on Modeling and Computer Simulation*, **16**, No 2 (2006), 152–173.

[11] W. Hormann, The generation of binomial random variables, *Journal of Statistical Computation and Simulation*, **46**, Nos 1-2 (1993), 101–110.

[12] H.E. Hurst, Long-term storage capacity of reservoirs, *Transactions of the American Society of Civil Engineers*, **116** (1951), 770–799.

[13] M. Krunz and A. Makowski, Modeling video traffic using M/G/∞ input processes: A compromise between Markovian and LRD models, *IEEE Journal on Selected Areas in Communications*, **16**, No 5 (1998), 733–748.

[14] W.E. Leland, M.S. Taqqu, W. Willinger and D.V. Wilson, On the self-similar nature of Ethernet traffic (extended version), *IEEE/ACM Transactions on Networking*, **2**, No 1 (1994), 1–15.

[15] S.Q. Li and C.L. Hwang, Queue response to input correlation functions: Discrete spectral analysis, *IEEE/ACM Transactions on Networking*, **1**, No 5 (1993), 522–533.

[16] N. Likhanov, B. Tsybakov and N.D. Georganas, Analysis of an ATM buffer with self-similar (fractal) input traffic, In: *Proc. INFOCOM'95* (Boston, MA, 1995), 985–992.

[17] M. Livny, B. Melamed and A.K. Tsiolis, The impact of autocorrelation on queueing systems, *Management Science*, **39**, No 3 (1993), 322–339.

[18] J.C. López, C. López, A. Suárez, M. Fernández and R.F. Rodríguez, On the use of self-similar processes in network simulation, *ACM Transactions on Modeling and Computer Simulation*, **10**, No 2 (2000), 125–151.

[19] S. Ma and C. Ji, Modeling heterogeneous network traffic in wavelet domain, *IEEE/ACM Transactions on Networking*, **9**, No 5 (2001), 634–649.

[20] I. Norros, A storage model with self-similar input, *Queueing Systems*, **16** (1994), 387–396.

[21] I. Norros, On the use of fractional Brownian motion in the theory of connectionless networks, *IEEE Journal on Selected Areas in Communications*, **13**, No 6 (1995) 953–962.

[22] M. Novak, *Thinking in Patterns: Fractals and Related Phenomena in Nature*, World Scientific (2004).

[23] V. Paxson and S. Floyd, Wide-area traffic: The failure of Poisson modeling, *IEEE/ACM Transactions on Networking*, **3**, No 3 (1995), 226–244.

[24] V. Paxson, Fast, approximate synthesis of fractional Gaussian noise for generating self-similar traffic, *Computer Communications Review*, **27**, No 5 (1997), 5–18.

[25] S. Resnick and H. Rootzen, Self-similar communication models and very heavy tails, *Annals of Applied Probability*, **10**, No 3 (2000), 753–778.

[26] R.H. Riedi, M.S. Crouse, V.J. Ribeiro and R.G. Baraniuk, A multifractal wavelet model with applications to network traffic, *IEEE Transactions on Information Theory*, **45**, No 3 (1999), 992–1018.

[27] M.E. Sousa, A. Suárez, J.C. López, C. López and M. Fernández, On improving the efficiency of an M/G/∞ generator of correlated traces, *Operations Research Letters*, **36**, No 2 (2008), 184–188.

[28] M.E. Sousa, Efficient on-line generation of the correlation structure of the fGn process, *Journal of Simulation*, **7**, No 2 (2013), 83–89.

[29] A. Suárez, J.C. López, C. López, M. Fernández, R. Rodríguez and M.E. Sousa, A new heavy-tailed discrete distribution for LRD M/G/∞ sample generation, *Performance Evaluation*, **47**, Nos 2/3 (2002), 197—219.

[30] K.P. Tsoukatos and A.M. Makowski, Heavy traffic analysis for a multiplexer driven by M/G/∞ input processes, In: *Proc. 15th International Teletraffic Congress* (Washington, DC, 1997), 497–506.

[31] B. Tsybakov and N.D. Georganas, On self-similar traffic in ATM queues: Definitions, overflow probability bound and cell delay distribution, *IEEE/ACM Transactions on Networking*, **5**, No 3 (1997), 397–409.

[32] W. Willinger, M.S. Taqqu, R. Sherman and D. V. Wilson, Self-similarity through high variability: Statistical analysis of Ethernet LAN traffic at the source level, *IEEE/ACM Transactions on Networking*, **5**, No 1 (1997), 71–86.