Letters to the Editor

re "On Credibility of Simulation Studies
of Telecommunication Networks",
by K. Pawlikowski et al.,
IEEE Comms Magazine (Jan. 2002, 132-139)

Letter by Donald DuBois in response to the article: "On Credibility of Simulation Studies of Telecommunication Networks," IEEE Communications Magazine, vol. 40, no. 1, pp. 132-139, Jan. 2002.

While the admonition of Pawlikowski et al. ("On Credibility of Simulation Studies of Telecommunication Networks," IEEE Communications Magazine , vol. 40, no. 1, January 2002) to question the credibility of simulation studies of telecommunications networks is well taken, one of the two main examples they give is overblown. The authors caution that pseudo random number generators (PRNGs) in real-life applications could have potentially serious problems since, with cycle lengths on the order of 2^{32}, a CPU operating at a speed of a few hundred megahertz "could generate all numbers of a mod(2^{31} -1) PRNG in 12 minutes," with GHz processors taking correspondingly less time. In real life applications, however, this problem is more apparent than real for the following two reasons.

Firstly, in most real world simulators the percentage of CPU time used to generate random numbers is less than one percent. Typically a PRNG consists of four or five lines of code while the rest of the simulator functions such as scheduling future events and mimicking the operation of a complex system with dozens of queues taking up the vast majority of the processing time. Therefore the cycle time in real life applications is typically much longer than indicated in the article.

Secondly, the cycle length of a PRNG is potentially a problem because spurious correlations to preceding cycles may occur. But the probability of this happening in even a moderately complex real-life simulator is negligibly small for the following reason. When the PRNG cycles an identical random number stream is generated but this in and of itself is not sufficient to produce correlations with previous cycles. For correlations to occur with respect to previous cycles the same random number stream has to feed into a simulated system that is in exactly the same state that the simulator was in at the beginning of the previous cycle. As an extremely simple example, the queues of a simulated system are typically empty when the first random of the first cycle is generated. At the beginning of the next PRNG cycle, the chances that all queues are empty and in exactly the same state is infinitesimally small. And even if all the queues were empty at the beginning of the next cycle the state of the simulated system is rarely as simple as just the queue occupancy. In addition, the state of the system will include, as a minimum, the time until the next arrival (exponential interarrival times is the only case where this is not a problem), which queue will receive the next arrival ? for example, multiple queues could represent traffic input ports on a packet router ? and how that arrival will be treated in the system ? a high or low priority packet, for instance. In short, the state of the system is multidimensional and all components of the state have to recycle at exactly the same time that the PRNG is recycling for the PRNG to cause spurious correlations. This occurrence is extremely unlikely in simulators of even modest complexity.

The validity of simulators still needs to be seriously questioned for many reasons and the authors provide a timely reminder that they are not identical to the reality they are intended to model.

A reply from the authors of "On Credibility of Simulation Studies of Telecommunication Networks"

D. DuBois, in his letter to the Editor of IEEE Communications Magazine , while supporting the main theses of [1], attempts to defend pseudo-random number generators (PRNGs) with cycle lengths of order of 2^{32} as still acceptable sources of primary randomness in modern simulation studies of telecommunication networks. However, that defence is not supported by strict objective evidence.

Having finished the writing of [1] in April 1999, we stated that a PRNG with the cycle length of 2^{31} -1 "could generate all numbers... in less than 12 minutes." Early this year, PCs with clock frequency of over 2 GHz have become commercially available, and the time needed for generation of all numbers of the cycle by such a PRNG has been already reduced to about four minutes. And this has certainly been not the last manifestation of Moore's law in action.

We could agree with D. DuBois that in many "real world simulators the percentage of CPU time used to generate random numbers is less than one percent," although we wished to be able to support this claim by results of an objective statistical study of modern simulations. Unfortunately, it seems that such a survey has not been conducted in a long time. Results of a survey conducted in 1981 and published in [2] showed that then, 3 percent of CPU time spent on a typical simulation was spent on generating pseudo-random numbers. Let us note that the time spent on generation of random numbers during a simulation does not depend only on the number of instructions defining a given PRNG, but its depends also on the number of times these instructions are invoked during the simulation. As simulation models become more complex and more complicated performance measures are studied, one usually needs larger samples of simulation output data (and more pseudo-random numbers) to produce final results with an acceptably low statistical error. This trend can soon result in multiple (instead of single) repetitions of cycles by PRNGs during single simulations on (superfast) PCs, if cycles of PRNGs are not sufficiently long. Obviously, "the final results can be very misleading if correlations hidden in the random numbers and in the simulated system ¡accidently¿ interfere with each other" [3].

However, the strongest argument against applications of PRNGs generating numbers in cycles of the order of $2^{32} - 2^{48}$ in simulation practice today has a theoretical basis. As mentioned in [1], if one is concerned with two-dimensional

uniformity of pseudo-random numbers, no more than of $O(\sqrt[3]{L})$ numbers[1] from a PRNG (with cycle length = L) can be used in one simulation. Empirical analysis of some popular PRNGs reported in [4] has specified that limit as $16\sqrt[3]{L}$. This restricts the number of pseudo-random numbers available from a PRNG with the cycle of $2^{31}-1$ to just about 20,000, and to about 1,000,000 in the case of PRNGs with the cycle of $2^{48}-1$.
Fortunately, there is no reason for concern. Good PRNGs able to satisfy the most sophisticated demands of modern simulation experiments on fast computers in the electronic technology of today, as well as the all-optical technology of tomorrow, have already been proposed; see [1] for details.

References

1. K. Pawlikowski, H.-D. J. Jeong, and J.-S. R. Lee, "Credibility of Simulation Studies of Telecommunication Networks," IEEE Commun. Mag., Jan. 2002, 132-9.
2. J. C. Comfort and A. Miller, "Considerations in Design of a Multi Microprocessor Based Simulation Computer," Proc. Conf. Modeling and Simulation on Microcomputers, San Diego, CA (Jan. 28-29, 1982), L. A. Leventhal, Ed., 1982, 110-2.
3. A. Compagner, "Operational Conditions for Random-Number Generation," Phys. Rev. E,, 1995, 5634-45.
4. P. L'Ecuyer, "Software for Uniform Random Number Generation: Distinguishing the Good and the Bad," Proc. 2001 Winter Simulation Conf. (Dec. 2001), 95-105.

Krzysztof Pawlikowski, with H.-D. Joshua Jeong and J.-S. Ruth Lee
Department of Computer Science
University of Canterbury Christchurch, New Zealand

Rejoinder by Pierre L'Ecuyer to the comments of D. DuBois on the article "On the Credibility of Simulation Studies of Telecommunication Networks."

D. DuBois argues in his letter that simulating queuing systems with a small PRNG, whose period length can be exhausted in a few minutes on a PC, should not be a problem in practice because (a) generating the random numbers takes only a small fraction of the CPU time, typically less than one percent, and (b) even if cycling does occur and the same random number stream is reused, it will be reused in a different way, so the results will differ anyway (with extremely

[1] Note that in [1] (p.134, left column) this limit was incorrectly stated as $8\sqrt[3]{L}$. It should read $O(\sqrt[3]{L})$.

high probability).
I disagree with both arguments. I just tried a small simulation experiment for an M/M /1 queue using a future events list and lists of customers in queue and in service, with statistical collection, and found that generating the exponential interarrival and service times consumed approximately 50 percent of the CPU time (in both C and Java). For a large queuing network, the proportion might be a little less, but not by much, unless the network has a very complex logic whose simulation is time-consuming and requires no random numbers. As for item (b), I would not bet on it. Most stable queuing systems turn out to be regenerative systems in the wide sense (viz., Harris-recurrent systems; see, e.g., [102, 108]). The sample paths of such systems tend to have *coupling points* over time when they are generated with common random numbers from different starting states. This means that the chances of eventual synchonization, or of significant correlation between two sample paths, is not as "infinitesimally small" as some people might expect. The practical effectiveness of the *common random numbers* variance-reduction methodology (see, e.g., [103, 105]), often observed empirically even for complex systems, shows that complexity does not eliminate correlation induction.
One point I would like to insist on is that for linear congruential generators (LCGs), major statistical problems occur way before the period length is exhausted. For example, in [104, 106], very simple probability models where n balls are thrown in k urns independently and uniformly, and the number of empty urns is counted (and variants of this model) are simulated. It is found that *all* LCGs of full period length p start to give totally wrong results when the sample size n exceeds (roughly) $16\sqrt{p}$ for the model just described and $16\sqrt[3]{p}$ for a variant. This is much less than the period length. For an LCG with period length 2^{31}, for example, n = 20,000 suffices for the results to be plainly wrong for the variant.
Another important issue with RNGs is that for comparing similar systems with common random numbers (which is a very frequent activity in simulation), one needs several "independent" streams of random numbers for simulating different parts of the model and for performing multiple runs [101, 103]. Proper implementation of such tools requires generators with much longer periods than simple LCGs [107].
The most important bottleneck for the credibility of simulation results is certainly the *building of valid stochastic models* , for example, with appropriate probability distributions and taking dependencies into account. This is the difficult part. However, the importance of using a good reliable RNG should not be dismissed, and there is no excuse for neglecting this, because it is easy to find good ones.

References

101. G. S. Fishman. Monte Carlo, Concepts, Algorithms, and Applications. Springer Series in Operations Research, Springer-Verlag, New York, 1996.

102. S. G. Henderson and P. W. Glynn, "Regenerative Steady-state Simulation of Discrete- event Systems," ACM Trans. on Modeling and Comp. Simulation, vol. 11, no. 4, 2001.
103. A. M. Law and W. D. Kelton, Simulation Modeling and Analysis, New York: McGraw-Hill, 3rd ed., 2000.
104. P. L'Ecuyer, "Software for Uniform Random Number Generation: Distinguishing the Good and the Bad," Proc. 2001 Winter Simulation Conf., 2001, pp. 95-105.
105. P. L'Ecuyer and G. Perron, "On the Convergence Rates of IPA and FDC Derivative Estimators," Ops. Res., vol. 42, no. 4, 1994, pp. 643-56. **106**. P. L'Ecuyer and R. Simard, "On the Performance of Birthday Spacings Tests for Certain Families of Random Number Generators," Math. and Comp. in Simulation, vol. 55, no. 1?3, 2001, pp. 131-37.
107. P. L'Ecuyer, R. Simard, E. J. Chen, and W. D. Kelton, "An Object-Oriented Random- Number Package with Many Long Streams and Substreams," Ops. Res., 2002.
108. S. P. Meyn and R. L. Tweedie, Markov Chains and Stochastic Stability, Springer-Verlag, 1993.

Pierre L'Ecuyer
Dpartement d'informatique et de recherche oprationnelle
Universit de Montral, C.P. 6128, succ. Centre-Ville
Montreal, H3C 3J7, Canada
http://www.iro.umontreal.ca/ lecuyer