Some considerations in simulating an M/M/1 queue *

Julio Rojas-Mora
LIA, University of Avignon
339, chemin des Meinajaries,
Agroparc BP 1228,
84911 Avignon Cedex 9,
France.
julio.rojas@univ-avignon.fr

Tania Jiménez
LIA, University of Avignon
339, chemin des Meinajaries,
Agroparc BP 1228,
84911 Avignon Cedex 9,
France.
tania.jimenez@univ-avignon.fr

Eitan Altman
Maestro group
INRIA, 2004 Route des Lucioles
06902 Sophia Antipolis
Cedex, France.
eitan.altman@sophia.inria.fr

ABSTRACT

In spite (or perhaps due) to its simplicity, the question of simulating the M/M/1 queue has attracted much interest. It has served as a benchmark as various properties related to the simulation, or to the simulated performance measures, are known for this queue. In this paper we report on some experience that we obtained from this experiment that we believe will be of importance in simulating many other queueing types as well as networks of queues. In particular, we present some results on improving precisions or on reducing simulation time by processing the simulation results with the bootstrap method, and by using the quantile method to obtain confidence intervals. As a secondary objective we have made a comparison of ns-2 and ns-3 for the particular case of the M/M/1 queue.

Keywords

M/M/1 queue. Simulations. Confidence interval. Bootstrap.

1. INTRODUCTION

The complexity of the systems that need to be simulated grows at a rate that is not slower than that of the speed or memory size of the computers we use for the simulations. In order to become more efficient in simulating a whole system it becomes necessary to simulate efficiently its building block, which motivated us to focus in this paper on the M/M/1 queue. Having chosen the M/M/1 queue, we are able to make use as a starting point the rich existing theoretical knowhow in its simulation, as well as simple known formulas for the performance measures of the queue. We shall present both generic simulation aspects that are relevant for any type of queue, as well as some aspects that are specific to the simulations in ns-2[10][1].

In spite of the simplicity of the M/M/1 queue, its simulation turns out to be quite harder than one might expect (especially in high load) and to exhibit unexpected behavior (such as systematic deviations of the empirical distribution of the average queue length from the Gaussian one), see [9, Chapter 11]. This motivates us to study methods for accelerating its simulation and improving the statistical precision.

Whereas many of the ideas presented in this paper might be familiar to the community of statisticians and to specialists in simulations, we believe that they may be new to many young researchers in networking and in queueing models that use simulations as a validation or as an experimental and exploration tool.

Accelerating simulations and improving precision

The simulation time of an M/M/1 queue may be extremely long if heavy load is considered. Indeed, the queue size process of an M/M/1 queue is known to converge (when scaled appropriately) to the reflected Brownian motion which is null recurrent (and thus does not have a steady state probability distribution). Other phenomena that make the simulations of the M/M/1 queue hard are discussed in [9, Chapter 11] (e.g. the empirical distribution of the average queue size turns out to have a heavier tail than the one of a Gaussian distribution even for the case of a very large number of replicas - 20000).

In this paper we are interested in obtaining the first two moments of the queue size by using simulations. Here again we observe that the duration of the simulation depends a lot on what we wish to simulate; when we want to obtain the second moment of the M/M/1 queue, the simulation time required for a given precision becomes much larger than that of the first moment.

This paper is organized as follows: in section 2 we present some theoretical basics that we will use throughout the paper. The simulation and data trace processing methods are described in section 3. Section 4 presents the results analysis, with section 5 containing some conclusions, as well as some future work we would like to address on this subject.

*This work has been partially supported by the French national project ANR WiNEM, ANR-06-TCOM-05.

[1] http://www.isi.edu/nsnam/ns/

GAMECOMM 2009, October 23, Pisa, Italy

DOI 10.4108/ICST.VALUETOOLS2009.7567

2. RELATED WORK

M/M/1 queue

We briefly summarize known formulas on the M/M/1 queue as well as some previous knowhow concerning its simulation.

Let L be the queue size in steady state (including the customer in service). The service times are i.i.d. with mean $1/\mu$; the customers arrive according to a Poisson process with rate λ. We define $\rho = \lambda/\mu$ (and assume throughout a stable queue, i.e. $\rho < 1$. Then we have the following [13, 74–75]:

$$E[L] = \frac{\rho}{1-\rho}, \ \mathrm{var}[L] = \frac{\rho}{(1-\rho)^2}, \tag{1}$$

and for the queue size (not including the customer in service):

$$E[L_q] = \frac{\rho^2}{1-\rho}, \ \mathrm{var}[L_q] = \sigma^2_{L_q} = \frac{\rho^2(1+\rho-\rho^2)}{(1-\rho)^2}, \tag{2}$$

which will be the ones used for comparisons with our simulation results.

Basic facts concerning simulations.

Assume that we wish to have a $(1-\alpha)100\%$ confidence interval for the average queue size, and let $z_{1-\alpha}$ be such that $\Psi(z_{1-\alpha}) = 1-\alpha$ where ψ is the standard Normal commulative distribution function.

- Assume we wish the variations of the average simulated L to be within an interval of size Δ with probability $1-\alpha$. Then the minimum simulation run length is given by

$$T_a(\Delta, \alpha) = \frac{8\rho(1+\rho)z^2_{1-\alpha/2}}{(1-\rho)^4\Delta^2}. \tag{3}$$

- Assume we wish the variations of the average simulated L to be within an interval of relative size $\Delta E[L]$ with probability $1-\alpha$. Then the minimum simulation run length is given by

$$T_a(\Delta, \alpha) = \frac{8\rho(1+\rho)z^2_{1-\alpha/2}}{\rho(1-\rho)^2\Delta^2}. \tag{4}$$

The sample variance of the queue size is of the order of $(1-\rho)^{-4}$, see [9, Theorem 11.0.1].

Bootstrap

Bootstrap is a method created by Efron [4] for non-parametrical estimation, which can be used to statistically process simulation results. From a given set of n simulated elements (traces), we create a larger amount k of elements by randomly picking elements from the original sample of n elements.

Let θ be a parameter of a completely unspecified distribution F, for which we have a sample of i.i.d. observations $X_1, \ldots, X_n$, and $\hat{\theta}$ be the estimation made of the parameter. From the sample, we will make k resamples with replacement, $X^*_{1,j}, \ldots, X^*_{n,j}$ $\forall j = 1, \ldots, k$, with each element having probability $1/n$ of being selected, and for each resample an estimator $\hat{\theta}^*_j$ will be calculated. By Monte Carlo approximation the distribution of $\hat{\theta}$ is then estimated by the distribution of $\hat{\theta}^*$.

Algorithm 1 Bootstrap

1. Make n simulations of an M/M/1 queue and let X_i, $\forall i = 1, \ldots, n$, be the estimation of the parameter of interest obtained from each simulation.
2. For $j = 1, \ldots, k$ do:
   (a) Let $X^*_{1,j}, \ldots, X^*_{n,j}$ be a resample, with replacement, taken from $X_1, \ldots, X_n$.
   (b) Let $\hat{\theta}^*_j = n^{-1}\sum_{i=1}^n X^*_{i,j}$.
3. Calculate $\hat{\theta}^* = k^{-1}\sum_{j=1}^k \hat{\theta}^*_j$, the Monte Carlo aproximation of the bootstrap estimation of θ.
4. Calculate confidence intervals for $\hat{\theta}^*$ using the quantile method (see definition below).

By central limit theorem, when $k \to \infty$, the quantiles of $\hat{\theta}^*$ will be better than those obtained directly by the sample. Computing power has made the problem of finding a suitable k a secondary one, as a big number of resamples can be made in a very short ammount of time. Singh[14] and Bickel and Freedman[2] are good references to understand the asymptotic characteristics of the bootstrap.

The bootstrap is a well-known method to improve the estimation of the parameters when there is high variability. In spite of this, there is, in our knowledge, no use of it on networking papers which use simulation studies. We only would like to encourage the use of such techniques.

In our work we used the bootstrap algorithm 1 for confidence interval reduction of the parameters of interest.

Remark. An alternative way to accelerate the simulations is the importance sampling or more generally, variance reduction techniques see e.g. [13, Chap 4] for a general introduction and [15] for its application to the M/M/1 queue. An alternative performant way is introduced in [7]. Both methods are different than the bootstrap approach in that both are based on simulating another model (e.g. use a larger load in order to obtain better estimate of a rare event of reaching a large queue size). Then some knowledge of the system is needed in order to transform the simulated results of the new model to that of the original one. The bootstrap method that we study is a post-simulation approach: it concerns statistical processing of simulated traces. It can be used on top of variance reduction techniques, such as importance sampling, when they are available.

Quantile-based confidence interval

Assume we wish to obtain the confidence interval of the estimation of some function of a simulated process X_t. The quantile approach to derive confidence intervals is based on running a number N of i.i.d. simulations (each simulation corresponds in our case to the queue length process or to functions of this process). We then use these to compute the empirical distribution of the function of the random variable.

Listing 1: ns-2 simulation script.

```
set ns [new Simulator]
set rep [lindex $argv 0]
set lambda [lindex $argv 1]
set mu [lindex $argv 2]
set tmax [lindex $argv 3]
set rng1 [new RNG]
set rng2 [new RNG]
for {set i 1} {$i<$rep} {incr i} {
        $rng1 next-substream;
        $rng2 next-substream;
}
set qsize [open qsize$rep.out w]
set n1 [$ns node]
set n2 [$ns node]
set link [$ns simplex-link $n1 $n2 100kb 0ms DropTail]
$ns queue-limit $n1 $n2 100000
# generate random interarrival times and packet sizes
set InterArrivalTime [new RandomVariable/Exponential]
$InterArrivalTime use-rng $rng1
$InterArrivalTime set avg_ [expr 1/$lambda]
set pktSize [new RandomVariable/Exponential]
$pktSize use-rng $rng2
$pktSize set avg_ [expr 100000.0/(8*$mu)]
set src [new Agent/UDP]
$src set packetSize_ 100000
$ns attach-agent $n1 $src
# queue monitoring
set qmon [$ns monitor-queue $n1 $n2 stdout 0.05] ;
set time 0.5
proc finish {} {
    exit 0
}
proc sendpacket {} {
    global qmon qsize ns src InterArrivalTime pktSize
    set length [$qmon set pkts_]
    set now [$ns now]
    puts $qsize "$now [$qmon set pkts_] "
    $ns at [expr $now + [$InterArrivalTime value]] "sendpacket"
    set bytes [expr round ([$pktSize value])]
    $src send $bytes
}
set sink [new Agent/Null]
$ns attach-agent $n2 $sink
$ns connect $src $sink
$ns at 0.0001 "sendpacket"
$ns at $tmax "finish"
$ns run
```

Thus, we define the $(1-\alpha)\cdot 100\%$ confidence interval for a $(1-\alpha)\cdot 100\%$ confidence level, as the interval between the $(\alpha/2)\cdot 100$-th and the $(1-\alpha/2)\cdot 100$-th quantiles[2] of the sample. This means, that for every 100 replications of the experiment, the estimation of the statistic will fall, at least, $(1-\alpha)\cdot 100$ times in this interval.

3. SIMULATION

ns-2 and ns-3

The Network Simulator ns-2 is a well known discrete event simulator for networking. The Network Simulator ns-3 is a new simulator, completely written from scratch, viewed as the natural replacement of ns-2 as the main tool for networking research and education. We used the version ns-2.33 and ns-3.4 with their default PRNG, i.e., MRG32k3a, by Pierre L'Ecuyer [8].

Listing 1 shows the code of a very simple model we used for the M/M/1 queue simulation. We made the same example on ns-3 [11] (3.4 version).

We ran simulations for 240 scenarios composed by combinations of the following parameters:

1. Simulation length: from 50000 to 90000 seconds, every 10000 seconds. The data was warmed-up using traditional batch means test [3].

Table 1: Number of events for low and high rate simulations.

ρ	T	λ_{LR}	μ_{LR}	Events	λ_{HR}	μ_{HR}	Events
0.9	50000	9	10	$4.5\cdot 10^5$	90	100	$4.5\cdot 10^6$
0.9	60000	9	10	$5.4\cdot 10^5$	90	100	$5.4\cdot 10^6$
0.9	70000	9	10	$6.3\cdot 10^5$	90	100	$6.3\cdot 10^6$
0.9	80000	9	10	$7.2\cdot 10^5$	90	100	$7.2\cdot 10^6$
0.9	90000	9	10	$8.1\cdot 10$	90	100	$8.1\cdot 10^6$

2. ρ: 0.5, 0.6, 0.7, 0.8, 0.9, 0.925, 0.95, 0.975. For each value of ρ we performed high and low event rates simulations:

   (a) Low event rate: $\rho_{LR}=\frac{\lambda}{10}$.

   (b) High event rate $\rho_{HR}=\frac{10\cdot\lambda}{100}$.

   Although mathematically equivalent, in simulations of the same length (in seconds), the quantity of generated arrivals is directly proportional to the quantity by which λ is multiplied, as table 1 shows. For the same load, simulations with high event rates are slower, but give best results, in terms of estimated relative half-width (ERHW, defined in section 4), than the ones using low event rates. It is for this reason that both λ and μ should be specified in simulation description.

3. Number of replications: 20, 50 and 100.

4. The queue length L_q was sampled at each arrival.

Experience on simulating in ns-2 queues with other service time distributions can be found in [1]. One can also find there a discussion on finite buffers.

The output data was processed with R.

The R statistical programming language

We constructed a wrapper[3] for the ns-2 script using GNU R version 2.6.2[4] [12], an open source statistical programming language. The data for each scenario was passed by this wrapper to ns-2 and the resulting trace file was read and processed in R. We developed a library for this wrapper, both mono and multicore[5] bootstrap and results plotting. This library can be sent by request and, after the documentation process is finished, it will be available for download.

4. ANALYSIS

In figures 1 and 2, for the mean queue size we can see how going from 20 to 50 replicates gives us a precision[6] gain of close to 30%, while with bootstrap we gain twice the precision. This same effect, combined with a faster convergence to a normal distribution, can be seen on figures 3 and 4.

We made the same scenario on ns-2 (in blue) and ns-3 (in red), with the same number of events, and we can't see any significative difference between them. We can also see that both statistics are very close to the theoretical value (black solid line) for ns-2 (blue dashed line) and ns-3 (red dashed

[2] Quantiles are equidistant points taken from a cummulative distribution function of a random variable or from the empirical distribution of a sample. More information on the different calculation methods used by statistical packages can be obtained in [6].

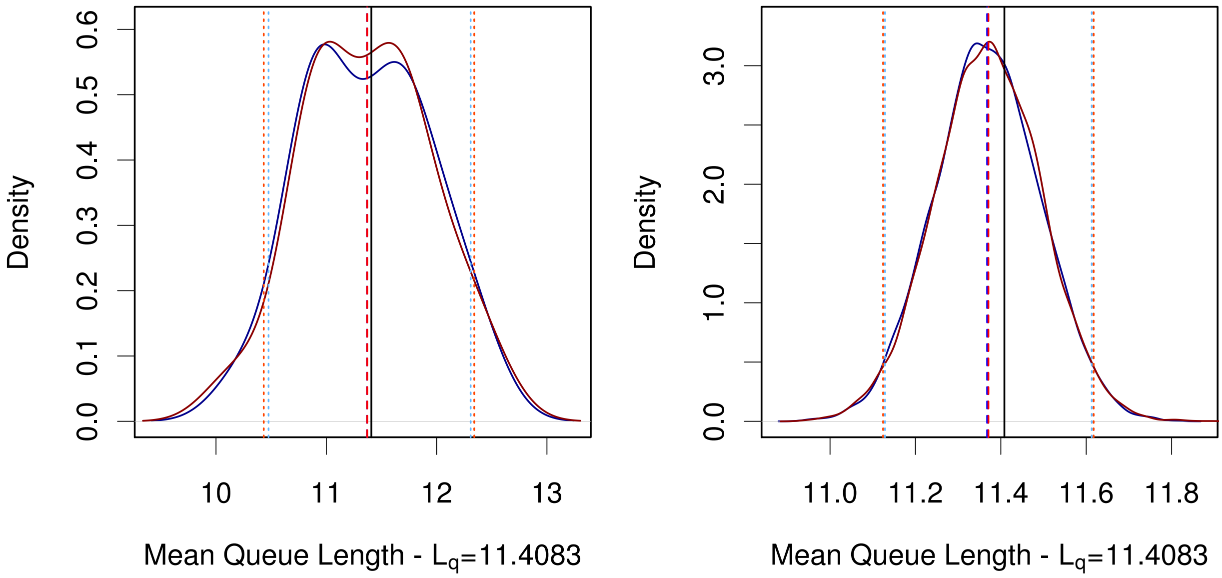
[3] A wrapper is a program that masks the execution of another program. In our work, R routines masked the ns-2 script used to run the simulations.

[4] http://www.r-project.org

[5] To parallelize the bootstrap algorithm in multicore processors.

[6] Peak of density value.

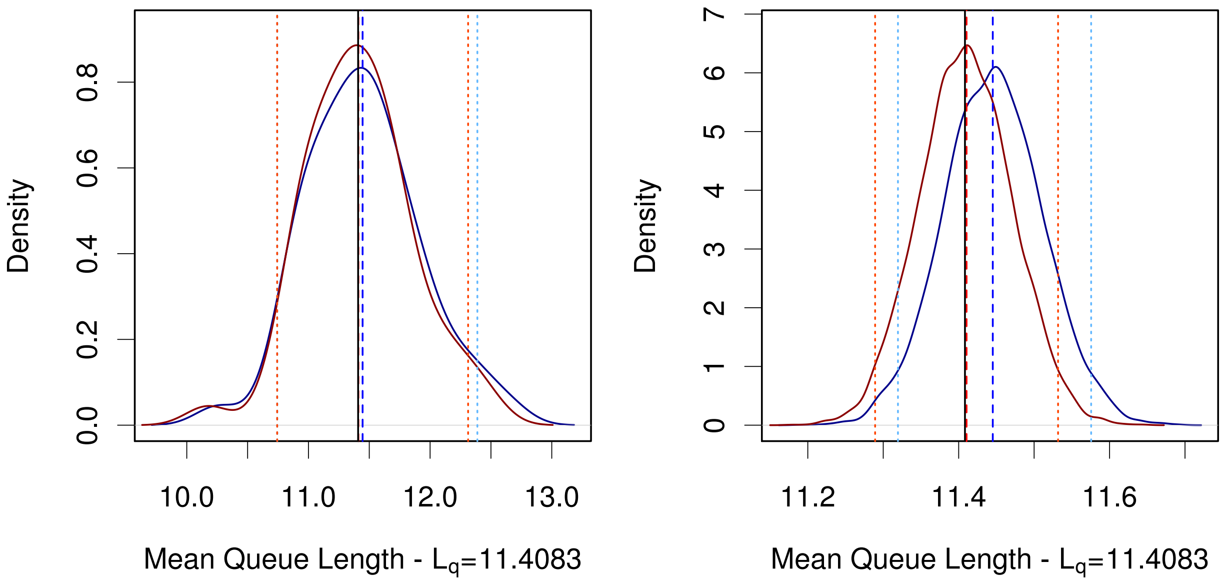
Figure 1: Density of the M/M/1 queue length average with $\rho = 0.925 = 9.25/10$ and 450000 arrivals. 20 independent replicates.



(a) Without bootstrap.

(b) With bootstrap (10000 resamples)

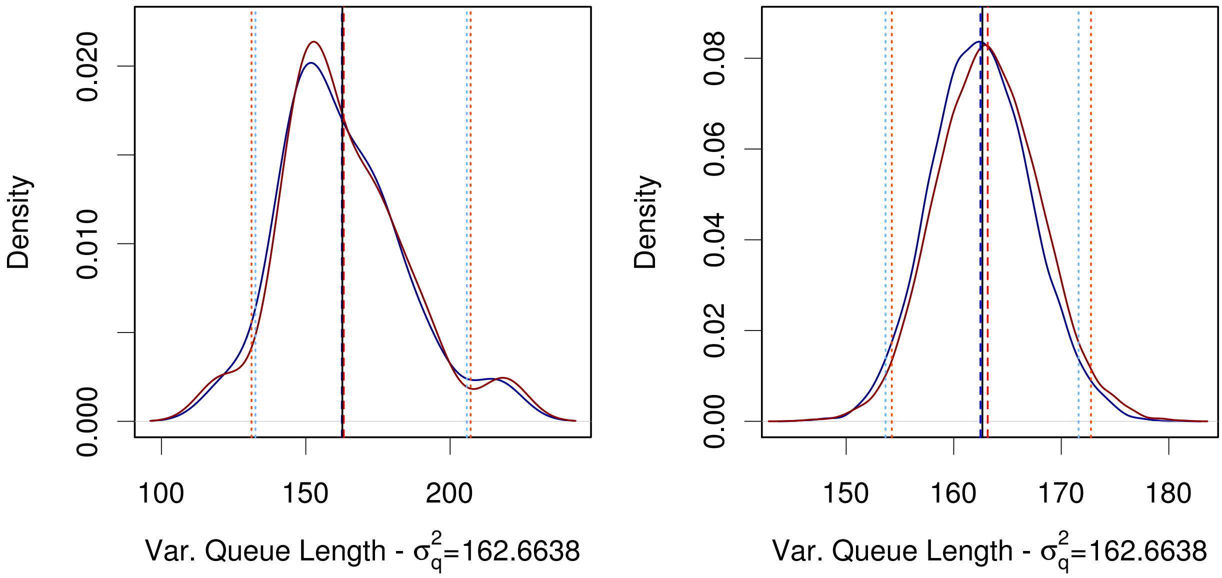
Figure 2: Density of the M/M/1 queue length average with $\rho = 0.925 = 9.25/10$ and 450000 arrivals. 50 independent replicates.



(a) Without bootstrap.

(b) With bootstrap (10000 resamples)

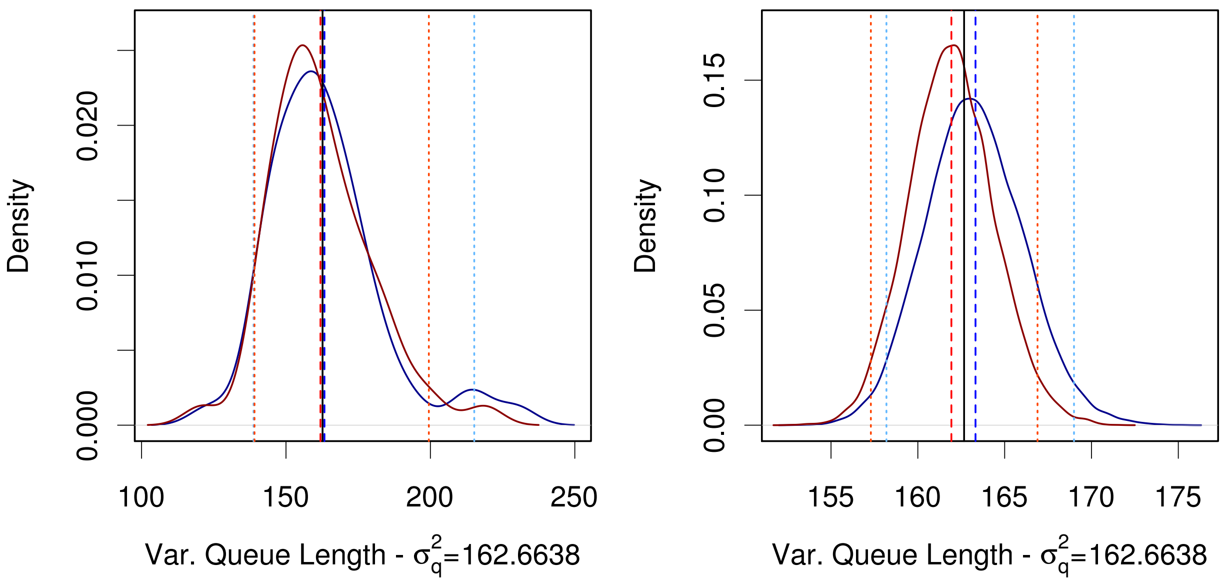
Figure 3: Density of the M/M/1 queue length variance with $\rho = 0.925 = 9.25/10$ and 450000 arrivals. 20 independent replicates.



(a) Without bootstrap.

(b) With bootstrap (10000 resamples)

Figure 4: Density of the M/M/1 queue length variance with $\rho = 0.925 = 9.25/10$ and 450000 arrivals. 50 independent replicates.



(a) Without bootstrap.

(b) With bootstrap (10000 resamples)

line). Distributions (solid blue line for ns2 and solid red line for ns3) follow very close shapes and confidence intervals are also very close between them (dotted blue line for ns2 and dotted red line for ns3). This is, in some way, a validation of ns-3, because ns-2 have been used as the standard network simulator for many years, while ns-3 is in the first years of developement.

The main objective of using bootstrap in this simulation problem is the reduction of confidence intervals calculated by the quantiles method, without needing additional simulation length nor increasing the event rate. This objective has been achieved, as shown in figure 5 and 6, especially in the case of variance estimation, while simulating and processing times have been reduced from days to seconds. We can also see, in figure 4a, how extreme points can affect the variance when using small replication numbers. Instead of a reduction of the confidence interval at high simulation lengths, there is a lump that shows the effect of variability on the second moment estimation, that bootstrap easily smooths out. On figure 4b we can see how extreme the reduction of the confidence intervals is for a big replication number and a high event rate.

To make comparisons between simulations with high and low event rates throughout the whole range of scenarios, we used the estimated relative half-width (ERHW), defined as:

$$\mathrm{ERHW}_\theta = \frac{|\mathrm{CI\ width}|}{2\theta}$$

where θ is the estimated value of the parameter (mean or variance in our case).

On figure 7 we can see the trade-off points between running simulations with low event rate using bootstrap and high event rate without bootstrap. There is some point where using bootstrap for low event rates won't improve ERHW, no matter how long the simulation runs, while longer simulations of high event rates will offer a better (smaller) ERHW. Of course, simulations with high event rates take much more computing time than the ones with low event rates. If enough computing power is at hand, we should use long simulations, with high event rates and bootstrap. If this is not the case, we can run shorter simulations, with low event rates and bootstrap without sacrificing too much precision. The figures shown the worst case scenario for the number of replications in our study. With more replications the results are better.

5. CONCLUSIONS AND FUTURE WORK

As it is known, the simplicity of the M/M/1 queue does not help much when it comes to estimating its second moments when heavy load is considered. Indeed, simulating times needed are long for a decent estimation. We report the improvement in the precision of simulations that is achieved by using the bootstrap approach; equivalently, this allowed us to decrease considerably the simulation times needed in order to achieve a given precision level.

Also, reliability is proved with the availability of theoretical formulas. This is not the case when using a real networking scenario. After proving this reliability we can extend the methodology to some more complex problems.

We think we have made a textbook case that can help young researchers get much more reliable results and conclusions in their simulation experiments. The studied case can be used as a validation method for new and old simulators results. The use of open source simulators and statistical analysis tools also help improving credibility [5] due to repeatability, and all the efforts invested by the community that has been developing ns-3 [11] and R may be helpful in this direction.

We are also extending this study to the case of heavy tail distributions of the file size, using processor sharing models.

6. REFERENCES

[1] Eitan Altman and Tania Jiménez. ns simulator for beginners. `http://www-sop.inria.fr/maestro/personnel/Eitan.Altman/COURS-NS/n3.pdf`.

[2] Peter J. Bieckel and David A. Freedman. Some asymptotic theory for the bootstrap. *The Annals of Statistics*, 9(6):1196–1217, 1981.

[3] Charles R. Cash, Barry L. Nelson, David G. Dippold, J. Mark Long, and William P. Pollard. Evaluation of tests for initial-condition bias. In *WSC '92: Proceedings of the 24th conference on Winter simulation*, pages 577–585, New York, NY, USA, 1992. ACM.

[4] B. Efron. Bootstrap methods: Another look at the jackknife. *The Annals of Statistics*, 7(1):1–26, 1979.

[5] Tom Henderson. Improving simulation credibility through open source simulations. `http://www.tomh.org/talks/simutools08-keynote-final.ppt`, 2008. Keynote Talk of Simutools 2008.

[6] Rob J. Hyndman and Yanan Fan. Sample quantiles in statistical packages. *The American Statistician*, 50(4):361–365, 1996.

[7] S. Kyriazopoulou-Panagiolopoulou, I. Koutoyiannis, and S. P. Meyn. Control variates as screening functions. In *Proceedings of Valuetools*, 2008.

[8] Pierre L'Ecuyer. Good parameters and implementations for combined multiple recursive random number generators. *operation research*, 47(1):159–164, 1999.

[9] Sean Meyn. *Control Techniques for Complex Networks*. Cambridge University Press, 2007.

[10] ns-2 developing core team. The network simulator ns-2. `http://www.isi.edu/nsnam/ns/`, 2008. Official webpage.

[11] ns-3 developing core team. The network simulator ns-3. `http://www.nsnam.org/`, 2008. Official webpage.

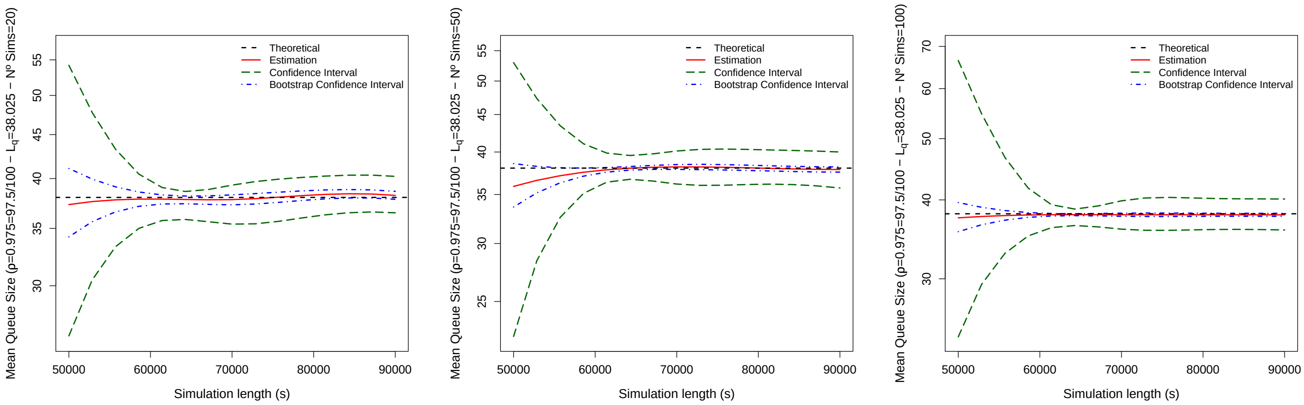
[12] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2008. ISBN 3-900051-07-0.

[13] Reuven Y. Rubinstein and Benjamin Melamed. *Modern simulation and modeling*. Wiley Series in Probability and Statistics. Wiley, New York, 1998.

[14] Kesar Singh. On the asymptotic accuracy of Efron's bootstrap. *The Annals of Statistics*, 9(6):1187–1195, 1981.

[15] Semih Yön and Dave Goldsman. Variance reduction via importance sampling. *Istanbul Ticaret Üniversitesi Fen Bilimleri Dergisi*, 5(10):35–41, 2006.

Figure 5: Confidence interval improvement for the mean queue size with bootstrap.

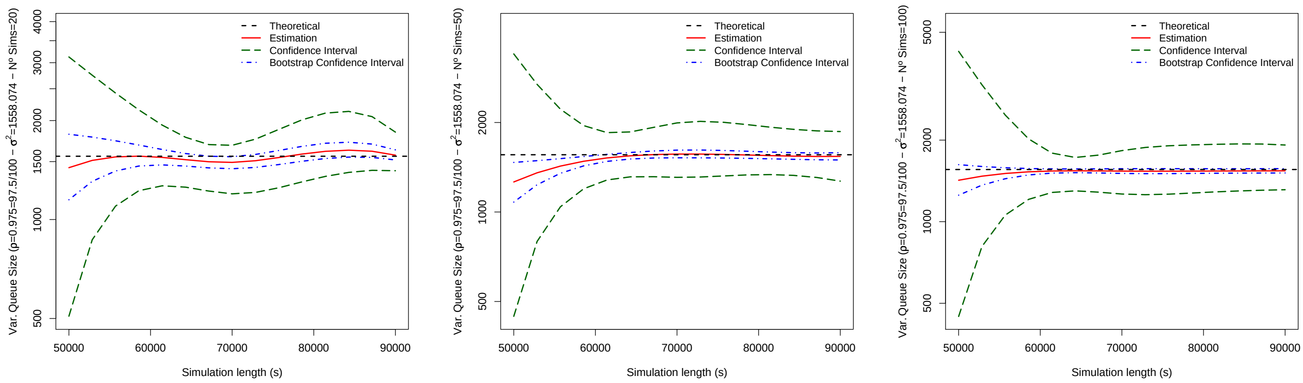


(a) Mean queue size for 20 simulations with $\rho = 97.5/100 = 0.975$

(b) Mean queue size for 50 simulations with $\rho = 97.5/100 = 0.975$

(c) Mean queue size for 100 simulations with $\rho = 97.5/100 = 0.975$

Figure 6: Confidence interval improvement for the queue size variance with bootstrap.

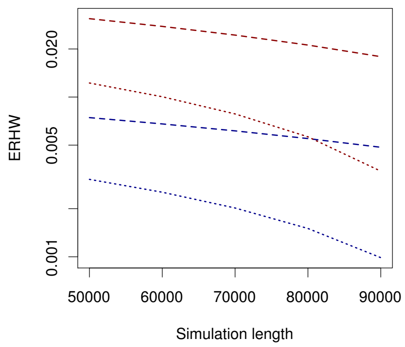


(a) Variance of the queue size for 20 simulations with $\rho = 97.5/100 = 0.975$

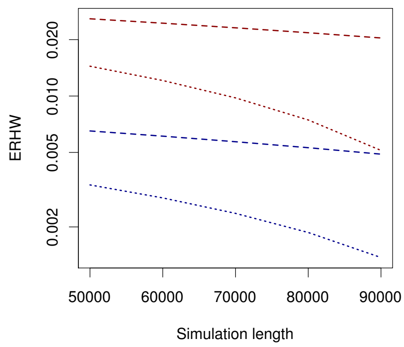
(b) Variance of the queue size for 50 simulations with $\rho = 97.5/100 = 0.975$

(c) Variance of the queue size for 100 simulations with $\rho = 97.5/100 = 0.975$

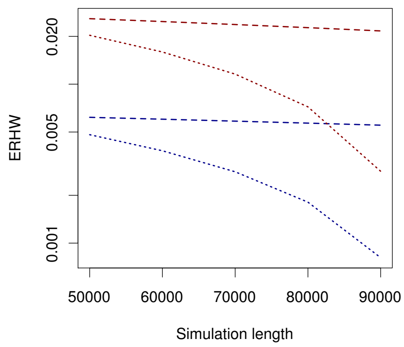
Figure 7: ERHW for low (red line) and high (blue line) rates with (dotted line) and without (dashed line) bootstrap



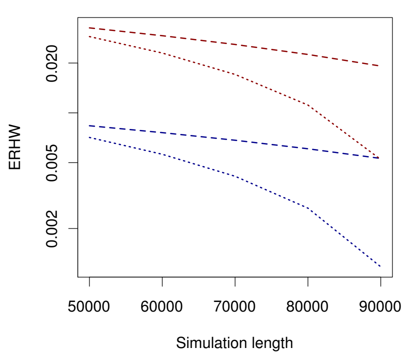
(a) $\rho = 0.5$



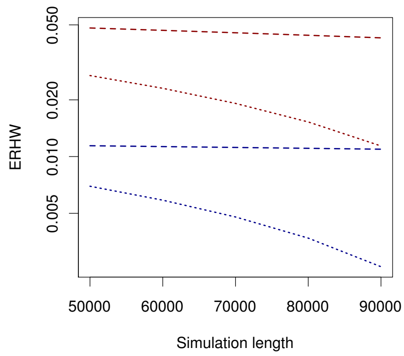
(b) $\rho = 0.6$



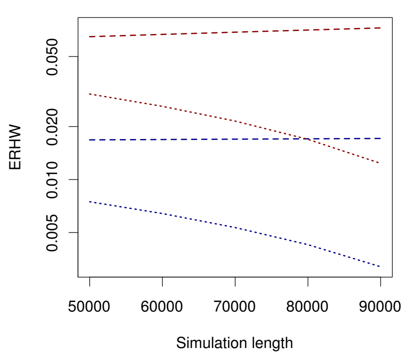
(c) $\rho = 0.7$



(d) $\rho = 0.8$



(e) $\rho = 0.9$



(f) $\rho = 0.925$