Simulating Bandwidth Sharing with Pareto distributed File Sizes

Eitan Altman, Julio Rojas-Mora, Tania Jimenez

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Simulating Bandwidth Sharing with Pareto distributed File Sizes

Julio Rojas-Mora — Tania Jiménez — Eitan Altman

N° 6926

May 2009

Thème COM

Rapport de recherche

ISRN INRIA/RR--6926--FR+ENG

ISSN 0249-6399

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

INRIA

centre de recherche
SOPHIA ANTIPOLIS - MÉDITERRANÉE

Simulating Bandwidth Sharing with Pareto distributed File Sizes

Julio Rojas-Mora* , Tania Jiménez , Eitan Altman†

Thème COM — Systèmes communicants
Équipe-Projet Maestro

Rapport de recherche n° 6926 — May 2009 — 20 pages

Abstract: The traffic on the internet has known to be heavy tailed: the size of file transfers through FTP or HTTP applications, as well as those transferred by P2P applications has been observed to have a very heavy tail. Typically modeled as Pareto distributed with parameter between 1.05 to 1.5, the file size has infinite variance. This is the source of many difficulties in simulating data traffic: convergence is very slow, simulations have to be very long, and the standard methods for deriving confidence intervals, based on the CLT, are not applicable here. We illustrate these well known problems through the simulation study of a processor sharing queue, which is often used to model session level resource sharing in the internet. We test bootstrap methods to accelerate convergence and improve the precision of simulations, and test a direct approach to obtain confidence interval based on the histogram of the empirical distributions. The conclusion drawn are then compared to those obtained when simulating in ns2 data transfer using TCP.

Key-words: M/G/1 queue. Processor Sharing. Simulations. Confidence interval. Bootstrap.

* LIA, University of Avignon, 339, chemin des Meinajaries, Agroparc BP 1228, 84911 Avignon Cedex 9, France. {julio.rojas,tania,jimenez}@univ-avignon.fr

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

Centre de recherche INRIA Sophia Antipolis – Méditerranée
2004, route des Lucioles, BP 93, 06902 Sophia Antipolis Cedex
Téléphone : +33 4 92 38 77 77 — Télécopie : +33 4 92 38 77 65

Simulation de Partage de Ressources entre fichiers dont la taille a une distribution de Paréto

Résumé : La taille des fichiers transférés sur l'Internet par FTP ou HTTP, ainsi que ceux transférés par les applications P2P, a souvent été observée et mesurée. La distribution observée a été à queue lourde et elle a souvent été modélisée par une une distribution de Pareto avec un paramètre entre 1,05 et 1,5; cela signifie que la taille des fichiers a une variance infinie. Ceci est la source de nombreuses difficultés dans la simulation du trafic de données: la convergence est très lente, les simulations doivent être très longues, et les méthodes habituelles de calcul d'intervalles de confiance, basées sur le Théorème Limite Centrale, ne sont pas applicables. Nous illustrons ces problèmes bien connus à travers la simulation d'une file "Processor Sharing", qui est souvent utilisée pour modéliser le partage des ressources au niveau session dans l'Internet. Les contribution principale de notre travail sont (i) l'application de la méthode de Bootstrap qui nous permet d'accélérer la convergence et d'améliorer la précision des simulations, et (ii) l'étude de l'approche pour obtenir des intervales de confiance basés sur la distribution empirique. Puis nous comparons les conclusions à celles obtenues lors de la simulation en NS2 des transfers de fichiers sur l'Internet.

Mots-clés : File M/G/1, Processor Sharing, Simulatiojs, Interval de Confiance, Bootstrap

1 Introduction

The processor sharing queue has been perhaps the most common model for bandwidth sharing of sessions (having the same RTT) of data transfer over the Internet [4], along with the Discriminatory Processor sharing Sharing queue [7] adapted to the case of unequal round trip times. Several research groups [17, 14, 11, 13] have examined its validity through simulations. The expected sojourn time of a customer in a processor sharing queue is known to be insensitive to the file size distribution (it depends on the distribution only through the expectation). Its variance, however, is not insensitive anymore, and in particular, it is infinite when the service time of a client (or equivalently in our setting - the size of a file that is transferred) has an infinite second moment [3].

The size of a data transfer over the Internet is known to be heavy tailed [12]. (This is the case both for FTP transfers as well as for the "on" periods of HTTP transfers). Among several candidates for modeling the distribution of these transfers, the Pareto distribution with parameter between 1.05 and 1.5 has been the one that gave the best fit with experiments for the last twenty years, see [8, 2, 6, 15].

This very heavy tail has been causing various serious problems for simulations of data transfers:

- The authors of [11] write: "since a substantial part of the distribution is in the tail, if the simulation is not run for very long the average of the sampled file sizes would be less than the nominal average, thus leading to a lower offered load and hence overestimation of the throughput."
- Central limit based confidence intervals are not available since the second moment of the number of sessions or of the workload in the system are infinite.
- Due to the last points one may have problems in interpretation of simulation results . Indeed, various papers in which sharing bandwidth is modeled by processor sharing report differences between the expected theoretical value and the simulation value [13, 14, 11, 17] that vary from around 10% in [11] and go up to a factor of 10 in some situations in [13]. When such deviations occur, it is important to know whether they can be due to the imprecisions in the simulation or to real phenomena.
- The warm-up time is extremely long, see [9].

Other approaches to simulating networks with Pareto distributed file size

One way to avoid heavy tails is to truncate them. This is in spirit of the suggestion in the paper "Difficulties in simulating queues with Pareto service" [10] that says: *"Since for any finite simulation run length, there is always a maximum value of the random variables generated, we, in actuality, are simulating a truncated Pareto service distribution. It has also been argued that there is always a maximum file size or claim amount so, in reality, we are always dealing with truncated distributions."* But where should we truncate the distribution? Truncation at some size M would be valid if the difference between performance with truncation at any other value L that is greater than M has a negligible impact on performance. Simulating with a truncated Pareto distribution may not be sufficient and several other tests of truncation at larger threshold values may be needed.

Our contributions

Our main contribution is in introducing to the networking context statistical methods that, to the best of our knowledge have not been used before in the networking context. In particular,

- We obtain an impressive improvement in the precision of simulations using the bootstrap approach; equivalently, this allows us to decrease considerably the simulation run times or the number of simulations needed in order to achieve a given precision level. We shall finally present detailed experimentations with some of them.
- **Confidence Intervals** We use the quantile based approach as an alternative to the central limit based approach for obtaining the confidence intervals. The central limit approach is not directly applicable whenever the variance is infinite, which is the case with the stationary sojourn time of a data transfer session (having a Pareto distribution with a shape parameter K lower than 2).
- We then apply these ideas directly to the simulation of competing non-persistent TCP transfers of files that have heavy tail distribution. We compare the simulation results to those obtained for the processor sharing queue. We identify various reasons for the deviation of the behavior of TCP from the ideal processor sharing model and manage to quantify the impact of some of these.

Structure of the paper

We begin by presenting in the next section a background on the bootstrap method and on the quantile method for obtaining confidence interval. In Section 3 we present our approach concerning the use of simulation to estimate the average number of packets in the processor sharing queue. In Section 4 we study some issues concerning the simulation of the workload of the processor sharing queue. We then present in Section 5 simulations of TCP connections that share a common bottleneck link and compare the precision that can be obtained (with and without the bootstrap approach) to the precision obtained in simulating the processor sharing queue. We provide explanations for the differences between the TCP scenario and its corresponding processor sharing model. A concluding section summaries the paper. An appendix clarifies some basic questions concerning confidence intervals.

Algorithm 1 Bootstrap algorithm for estimating the mean queue size

1. Make n simulations of the queue queue size 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 approximation of the bootstrap estimation of θ.

4. Calculate confidence intervals for $\hat{\theta}^*$ using the quantile method.

2 Background

Bootstrap

Bootstrap is a method created by Efron[20] for non-parametrical estimation. 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 let $\hat{\theta}$ 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 $\hat{\theta}^*$. When $k \to \infty$, the estimation of $\hat{\theta}$ will be better and, in turn, the real distribution of θ will also be better estimated. We note that the time and memory overhead for resampling and performing the bootstrap algorithm are often much smaller than the ones needed to create more samples, and can be performed within a very short amount of time. Singh [21], and Bickel and Freedman [19] are good references to understand the asymptotic characteristics of the bootstrap.

Remark. An alternative way to accelerate the simulations is the important sampling or more generally, variance reduction techniques see e.g. [22, Chap 4] for a general introduction. They are different than the bootstrap approach in that they 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 when they are available.

Quantile-based confidence interval

Assume we wish to obtain the confidence interval of the estimation of some parameter 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.

For $(1-\alpha)\cdot 100\%$ confidence level, the $(1-\alpha)\cdot 100\%$ confidence interval by the quantile method is the interval between the $(\alpha/2)\cdot 100$- th and the $(1-\alpha/2)\cdot 100$-th points of the sorted sample.

3 Simulating the queue size of a processor sharing queue

Through a series of simulations performed in JAVA (available from the authors by request) we exhibit the power of the bootstrap approach: its ability to increase the precision of the simulation of Internet traffic that is throttled by some bottleneck link and at the same time decrease the required duration of the simulation. In this section we restrict to study of the processor sharing queue which had often been proposed as a model for TCP transfers sharing a bottleneck link in the Internet. A TCP session is then represented by one customer in the PS queue.

Below we used the PS queue with a Poisson arrival process with a rate of $\lambda = 1$ customers per second (A customer represents a file when the processor sharing queue is used to model file transfers). We consider three service time distributions: Pareto with shape parameter 1.5, Pareto with shape 2.5, and exponentially distributed. We vary the average service time σ so as to obtain an average load $\rho = \lambda\sigma$ that takes the values 0.6 and 0.7.

Each one of the figures below correspond to the queue size of the processor sharing averaging over 100 independent samples. When considering the PS queue as a model for bandwidth sharing in the Internet, the queue size should be interpreted as the number of ongoing sessions.

The duration of each simulation is $6{\cdot}10^6$ sec and there is a warm up time of 300000 sec. In each one of the scenarios described in Figure 1, we present:

1. the theoretical steady state expected queue size
2. the value obtained by the simulations
3. the confidence interval corresponding to a 95% percentile of obtained using the quantile method
4. the confidence interval corresponding to a 95% percentile obtained after applying the bootstrap method.

We took 100000 resamples out of our 100 original samples for the bootstrap.

Figure 1 displays the precision of the simulations with and without the bootstrap approach for $\rho = 0.6$ (up) and $\rho = 0.7$ (down). The left subfigures are for an exponential distributed service time, the middle and right ones are for a

Pareto distributed service time with parameter $K = 2.5$ and $K = 1.5$, respectively. The average service time is the same in the all three sub-figures figures corresponding to the same ρ. (It was chosen so that indeed $\rho = \lambda \cdot \sigma$ will have the values 0.6 and 0.7, respectively.)

We observe the following points from the simulations:

- The simulations show well the insensitivity of the PS regime to the service time distribution. Indeed, in each set of simulations having the same load ρ, we see convergence to the same average rate for the exponential case, the Pareto distribution with parameter $K = 2.5$ and for the Pareto distribution with $K = 1.5$.
- The 95% confidence interval without bootstrap is around 10 times larger than that of the bootstrap (which establishes clearly the advantage of using it). This is seen to hold for any duration of the simulation.
- We see that the confidence interval are around 10 times smaller in the case of exponential and Pareto with shape parameter of 2.5 than in the case of Pareto shape parameter 1.5. This can be expected since the tail of the latter distribution is much heavier.
- Duration of the simulation: For all three distributions, and for the different loads, we see that the precision obtained with bootstrap after already 400000 sec is more than three times better than that without bootstrap after we see that even after 6 million seconds. It thus seems that to get the same precision without bootstrap, one would need to use simulations much longer than 15 times as much as with bootstrap.

4 On the simulation of the workload in a queue with heavy tailed service time

The state space needed to represent the evolution of the number of packets in a processor sharing is quite complex: we need the residual time of each packet that is in the system. In contrast, the evolution workload in the system is easy to describe as it can be written as a one dimensional recursive equation. We thus chose to illustrate some of the difficulties in simulating the processor sharing queue by a short discussion concerning the simulation of the workload process.

We consider in this Section the workload of an M/G/1 processor sharing queue with a Poisson arrival process with rate λ and with i.i.d. service times. The workload is the time it takes to complete transmission of all the connections that are present in the system. Assume throughout that $\rho < 1$ so the workload process is ergodic.

We note that the workload process is the same as the one of a FIFO queue with the same arrivals and service times.

In case service times do not have a finite second moment, the expected workload at steady state is infinite. (Indeed, the workload is invariant under the service discipline, and is thus equal to the one of FIFO discipline. Due to the PASTA property, the expected stationary workload equals to that at arrival epochs which equals the expected waiting time. The latter is infinite in the M/G/1 FIFO queue since since the arrival has to wait more than the residual service time of the customer in service, and the latter is proportional to the second moment of the service time.)

Consider now the case where the service time has a finite second moment but infinite third moment. In that case the expected stationary workload is finite but its variance is infinite.

Assume that we wish to estimate the expected stationary workload $E[V]$ in the system. The workload satisfies the recursion

$$V_0(v) = v, \quad V_{n+1}(v) = (V_n(v) + \Theta_n - \tau_n)^+, \; n \geq 0$$

where V_n is the workload as seen by the nth arrival, Θ_n is the workload brought by the nth arrival, and τ_n is the time between the nth arrival epoch T_n and the $(n+1)th$ arrival instant T_{n+1}. $E[V]$ can then be estimated by

$$\widehat{V}_n = \frac{\sum_{i=1}^{n} V_i}{n}$$

for n large (due to the PASTA property, estimation at arrival instants indeed converges to the stationary random variable). Define the bias function:

$$R_n(v) = E[S_n(v)] - nE[V] \; where \; S_n(v) = \sum_{i=1}^{n} V_i(v).$$

We note that $R_n(0)$ is monotone decreasing in n. It has a limit as $n \to \infty$ which we denote by R. $R_n(0)$ is known as the bias; if it were zero then $\widehat{V}_n = E[V]$. Otherwize the total expected estimation error at time T_n is given by $R_n(v)/n$.

To obtain an estimation error smaller than ϵ, we need to choose n such that $R_n/n < \epsilon$. It is advocated in [18] to choose n such that $R/n < \epsilon$. It ensures that $|R_n|/n < \epsilon$. Following this approach, we discover, unfortunately, that the simulation length should be taken to be infinite since $|R|$ turns to be infinite. We next establish this conclusion.

Let V_n^* be the stationary workload process We assume that V_n^* and V_n are defined on a joint probability space: the arrival process and service times are the same in both. The only difference is in the initial state.

Define $\eta(v)$ to be the first time n that V_n couples with V_n^*. Let $\theta(v)$ be the first time that it hits 0. Then

$$R_n(v) = \sum_{i=0}^{n} E\Big(1\{\eta(v) > i\}[V_i(v) - V_i^*]\Big)$$

We make the following key observation: When $v > V_0^*$ then $\eta(v) = \theta(v)$. Indeed, as long as V_n^* and V_n are both positive then the difference $|V_n - V_n^*|$ is constant. It start decreasing when the smaller of the two hits zero and it vanishes when they are both zero.

Let $Q_0(v) = v$ and define for $n > 0$:

$$Q_n(v) = \sum_{i=0}^{n} 1\{\theta(v) > i\} V_i(v)$$

Note that $R_n(v) \geq P(V^* < v) Q_n(v)$.

Since for all $n > 0$ we have $Q_n(v) \geq v$, this gives (using Jensen's inequality)

$$E[Q_{n+1}(v)] \geq v + E[Q_n(v - \tau_1)^+] \geq v + E[Q_n(v - E[\tau])^+]$$
$$\geq v + (v - E[\tau])^+ + E[Q_{n-1}(v - 2E[\tau])^+]$$

Continuing iteratively, we get for $n \geq v/E[\tau]$

$$E[Q_{n+1}(v)] \geq \frac{v^2}{2E[\tau]}.$$

Since Q^n increases with n, we get by the monotone convergence Theorem

$$\lim_{n\to\infty} E[Q_n(v)] \geq \lim_{n\to\infty} E[Q_n((v + \Theta - \tau)^+)]$$

$$\geq \lim_{n\to\infty} E[Q_n((\Theta - \tau)^+)] \geq \frac{1}{2E[\tau]} E[((\Theta - \tau)^+)^2] = \infty$$

since $E[\Theta^2] = \infty$. It then follows that $R(v) = \infty$ for any $v > 0$.

The fact that the bias is infinite suggests a very low rate of convergence and a long warm up time T where we estimate $E[V]$ by

$$\widetilde{V}_n = \frac{\sum_{i=t+1}^{t+1} V_i}{n}.$$

5 Simulating TCP connections

We compare in this section simulations that we performed with ns2.33 of TCP sessions with the simulation of the processor sharing queue. The network we simulate is given in Figure 2.

We took 200 input links each of speed 100 Mbps. The packet size was taken to be 1 KByte. The average session size was taken to be 200 KBytes. The total round trip delay is 0.4 msec. We simulate the New Reno version without the delayed Ack option. The maximum window size is of 20 packets, which is the default size of ns2. We later change this value.

To make comparisons between precision of simulations that have different event rates and different averages, we find it convenient to normalize the confidence interval. we used the estimated relative half-width (ERHW) of the confidence interval defined as half the difference between the upper and the lower values of the interval divided by the average value.

5.1 Bootstrap and the confidence interval

We are interested in comparing the precision (confidence interval) obtained with bootstrap when simulating a processor queue, on one hand, and when simulating TCP, on the other hand.

Figure 3-5 depict the comparison for $\rho = 0.6$ and 0.7, the first Figure reports simulations for the exponential distribution and the other two are for Pareto distribution with parameter $K = 2.5$ and $K = 1.5$, respectively. Each subfigure contains four curves:

1. The precision (ERHW) obtained by using bootstrap with TCP,

2. The precision (ERHW) obtained in simulating TCP without the bootstrap,
3. The same for simulating the processor sharing queue.

5.2 Some insight on the bias

As already mentioned, many papers already studied the question of how good a processor sharing queue is able to model bandwidth sharing by TCP connections, see e.g. [13, 14, 11, 17]. They all reported some differences between the expected theoretical value predicted by the processor sharing queue and the simulated value obtained by TCP sessions. All reported that for a given load ρ, the actual TCP throughput corresponded to a processor sharing queue with a higher value of ρ. The reported differences vary from around 10% in [11] and go up to a factor of 10 in some situations in [13].

In this section we try to contribute to understanding where the differences come from. We list some of our findings and some recommendations in order to reduce these differences.

5.3 Packet sizes

In ns2, packet sizes are fixed. If we use a probability distribution that has a continuous support (such as the exponential or the Pareto distribution) for the file size, then the actual file size will be a little longer since the last packet will be rounded up. In some applications (see [1]), the average TCP transfer size is around 8-12 KBytes, so if packets of length 1.5 Kbytes are used then the rounding error which is of 0.75 KBytes in the average, will contribute to an increase in ρ of around 7-10 %.

A second source of underestimation of ρ is that in many variants of the ns simulator, when we declare the size of the packet we wish to use then the actual packet size will get 40 bytes added (representing the extra IP and TCP headers). With a packet size of 1 KBytes this will contribute to yet another 4 % underestimation of ρ.

We have incorporated the above considerations in the simulations reported in this paper.

5.4 Burstiness and vacations

Server vacations: It may occur quite frequently that the bottleneck queue is empty but there is at least one ongoing session. The likelihood of this event increases as the bandwidth delay product increases. Note that for a given ρ, the probability that the system is empty (no sessions) is expected to be 1-ρ, so due to the PASTA property, the probability that an arriving session would find the system empty is also 1-ρ. As long as the arriving session is the only one in the system, and as long as it is in the slow start phase, the queue at the bottleneck is often empty and the server is then not busy. This can be modeled as a vacation which results in an increase in the workload in the system (as compared to the case in which the server is not on vacation).

Burstiness: We have evidence from simulations that TCP traffic can be quite bursty: many successive packets can belong to the same connection [16]. We suspect that the larger this burstiness is, the less we can use the processor sharing discipline as a model for the session level evolution; the latter becomes closer to the FIFO discipline. Note that with the Pareto file size distribution with $K = 1.5$, the expected number of customers in a FIFO M/G/1 queue is infinite for every $\rho > 0$. Thus this could explain a larger number of sessions.

5.5 Maximum window size

TCP is a window based protocol for reliable communication and congestion control. Each time it has a packet to send, it stamps the packet with a sequence number. To ensure reliability it uses Acknowledgements from the destination to learn about possible losses of packets. The window size indicates the maximum number of packets it can send before receiving an acknowledgement. The larger the window is, the larger the transmission rate is. In absence of congestion (i.e. as long as Acknowledgements arrive regularly and losses are not detected) the window size keeps growing, until it reaches a maximum size. The default value for this size is 20 packets in ns2. The larger the maximum value is, the more we can expect the connection to be bursty, so we can expect to a larger average number of sessions as argued in Section 5.4.

We have tested through simulations the impact of the maximum TCP window size on the expected number of ongoing sessions and discovered that the latter is indeed sensitive to the maximum window size. The larger the maximum window size is, the larger is the average number of ongoing sessions and the average transfer time of a connection. This could perhaps be explained by the burstiness.

We present below our experiments on the impact of the maximum window size on the average number of active sessions as well as on other parameters.

Figure 6 reports on the empirical distribution of the average number of ongoing connections as a function of the maximum TCP window size.

For each value of maximum window size, we did 20 simulations. Each simulation lasted till 2000000 arrivals of sessions occurred. The 600000 first sessions were ignored (this was the warm up time). The other parameters of the simulations are as in Section 5.1.

For each value of maximum window size, we give the empirical probability density of the average number of sessions. This is described by the contour of the beanplots (that represents the histogram). Each white bar inside a beanplot represents the average size in one of the twenty simulations. The black bar that traverses each one of the beanplot gives the average obtained from the set of 20 simulations. The dotted horizontal line gives the theoretical average number of customers in the corresponding processor sharing queue.

As we can see, the expected number of ongoing TCP sessions that fully agrees with the processor sharing model is the one obtained with a maximum window size of 8. All other values of maximum window size below 20 gave deviations not greater than 10% with respect to the theoretical value. However, we see that for a maximum size of 100, the expected number of sessions is almost double the theoretical value.

Figure 7 reports on the empirical distribution of the maximum number of ongoing connections that were present simultaneously at some time during the simulation, as a function of the maximum TCP window size. Note that unlike the case of average sizes, in which each sample average takes another value, the number of different values of the maximum number of sessions that we had within our simulations takes finitely many values, and some values appear several times during the simulations. The number of times that a value appears in the simulations is represented by the length of a white line (and when this value is so large that the line exceeds the boundary of the beanplot, then the bar continuous in black).

Figure 8 reports on the empirical distribution of the fraction of arrivals of sessions that found the system non-empty upon arrival, and the fraction of arrivals that found the bottleneck queue non-empty. The difference between these indicate that the from time to time there are no transmissions and yet there are ongoing sessions. We shall return to that point towards the end of the section.

5.6 Buffer size

The buffer size at the bottleneck queue turned out to be yet another factor that has an influence on the average number of ongoing sessions. With a maximum window size of 8 and with $\rho = 0.6$, the size of the buffer for which we obtained full agreement of the average number of sessions with the theoretical value (of 1.5) given by the processor sharing was 64. However, we observe that the simulations give good approximations for any larger value of the buffer size, see Figures 9-10. In both figures the largest value of buffer size that we tested was of 1 million packets. We write "INF" in the curves for "Infinite buffer" since with the size of 1 million we had no packet losses, so any buffer of larger size than 1 million would give the same results in this simulation.

The second of these figures uses bootstrap which is seen to result in a considerably better precision.

To understand the deviations from the theoretical value we measure the fraction of time that the queue is empty but there are ongoing sessions. We took a maximum window size of 8, $\rho = 0.6$, $K = 1.3$. We obtained around 8% for the case of buffer size of 12 packets, and 0.36% for a buffer of size 64.

We thus attribute the large deviations from the theoretical value to many losses that occur and that result in large periods during which the queue is empty although there are ongoing sessions. During these times, the processor sharing queue has "service vacations" and the theoretical results for a queue without vacation are not valid anymore.

This phenomena is countered when using a smaller value of the window sizes and therefore in spite of small buffers one gets better agreement with theoretical results if the maximum window size is smaller.

Note that the "rule of thumb" for selecting buffer size as the bandwidth delay product would give a value of 2 packets in our case which gives values of number of sessions much larger than the theoretical value predicted by the processor sharing queue.

For $\rho = 0.7$ we obtained very similar results. The theoretical average number of sessions in a processor sharing queue is 2.33. For a maximum window size of 8 we obtained the following values for the average number of sessions: 7.41, 3.723 and 2.423 for a buffer of size 12, 24 and 64, respectively.

6 Conclusions

We have studied in this paper the benefits that the bootstrap method can bring to the simulations of internet traffic sharing a common bottleneck link, and more generally, of the processor sharing queue with heavy tailed service times, which has served as a model for TCP sharing common resources. We found out that both the precision of the simulations can be improved and, at the same time, the simulation durations can be substantially shortened. We have analyzed the discrepancy between the results predicted by using the processor sharing queue and those obtained by simulating directly the short lived TCP connections that share a common bottleneck queue. We identified various possible reasons for the discrepancy and provided some recommendations that can help understand and minimize them.

References

[1] K. Thomspson, G. J. Miller, R. Wilder, "Wide area Internet traffic patters and characteristics, *IEEE Network* **6**(11), 1997, pp 10-23.

[2] B. Sikdar, S. Kalyanaraman and K. S. Vastola, "An Integrated Model for the Latency and Steady-State Throughput of TCP Connections", *Performance Evaluation*, v.46, no.2-3, pp.139-154, September 2001.

[3] Teunis J. Ott, "The Sojourn-Time Distribution in the M/G/1 Queue with Processor Sharing", Journal of Applied Probability, Vol. 21, No. 2 (Jun., 1984), pp. 360-378

[4] J.W. Roberts, "A survey on statistical bandwidth sharing". Computer Networks 45, 319-332 (2004).

[5] K. Tutschku and Ph. Tran-Gia, "Traffic characteristics and performance evaluation of Peer-to-Peer systems". In: Steinmetz, R., Wehrle, K. (eds.) Peer-to-Peer Systems and Applications. LNCS, pp. 383-397. Springer, Heidelberg (2005)

[6] Philippe Olivier, "Internet Data Flow characterization and bandwidth sharing modelling", L. Mason, T. Drwiega, and J. Yan (Eds.), ITC 2007, LNCS 4516, pp. 986-997, 2007.

[7] E. Altman, K. Avratchenkov and U. Ayesta, "A survey on discriminatory processor sharing", Queueing Systems, Vol. 53, No. 1-2, pp. 53-63, June 2006.

[8] Edward Chlebus and Rahul Ohri, "Estimating Parameters of the Pareto Distribution by Means of Zipf's Law: Application to Internet Research", IEEE GLOBECOM 2005

[9] Argibay Losada P., A. Suarez Gonzalez, C. Lopez Garcia, R. Rodriguez Rubio, J. Lopez Ardao, D. Teijeiro Ruiz. "On the simulation of queues with Pareto Service". Proc. 17th European Simulation Multiconference, pp. 442-447, 2003.

[10] Gross D., M. Fischer, D. Masi, J. Shorte. D. Gross. "Difficulties in Simulating Queues with Pareto Service." Proceedings of the Winter Simulation Conference, pp. 407-415, 2003.

[11] Arzad A. Kherani and Anurag Kumar, Performance Analysis of TCP with Nonpersistent Sessions, NCC 2000, New Delhi, January 2000.

[12] Allen B. Downey, Lognormal and Pareto distributions in the Internet Computer Communications 28 (2005) 790-801.

[13] Nandita Dukkipati, Masayoshi Kobayashi, Rui Zhang-Shen, and Nick McKeown, "Processor Sharing Flows in the Internet", Thirteenth International Workshop on Quality of Service (IWQoS), Passau, Germany, June 2005

[14] P. Brown, D. Collange, "Simulation Study of TCP in Overload", Proceedings of the Advanced International Conference on Telecommunications and International Conference on Internet and Web Applications and Services (AICT/ICIW 2006)

[15] Sally Floyd and Vern Paxson, "Difficulties in Simulating the Internet", Proceedings of the 1997 Winter Simulation Conference, Atlanta, GA, 1997.

[16] E. Altman and T. Jiménez, "Simulation analysis of RED with short lived TCP connection", *Computer Networks*, Vol 44 Issue 5, pp. 631-641, April 2004.

[17] E. Altman, T. Jimenez , D. Kofman, "DPS queues with stationary ergodic service times and the performance of TCP in overload", Proceedings of IEEE Infocom, Hong-Kong, March 2004.

[18] Aad P. A. van Moorsel, Latha A. Kant and William H. Sanders, "Computation of the Asymptotic Bias and Variance for Simulation of Markov Reward Models", *IEEE Proceedings of Simulation*, 1996.

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

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

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

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

Appendix: Reminder on confidence interval

We end this contribution with some elementary points on confidence interval. Consider the sample average $Z_n = \sum_{i=1}^n X_i/n$ as a function of n, where X_i are i.i.d. random variables. If there were a CLT then this sample average would behave asymptotically like $N/\sqrt{n}$ where N is a Normally distributed RV, so that $\log Z_n \sim -0.5\log(n) + \log(N)$. $\log(Z_n)$ should thus be asymptotically linear decreasing in $\log(n)$.

In addition, assuming that the CLT exists, the confidence interval (corresponding to a confidence level of 95%) is given by

$$CI_n = Z_n \pm 1.96SD = Z_n \pm 1.96\frac{1}{\sqrt{N}}\sqrt{\frac{1}{n}\sum_{i=1}^{n}(X_i - Z_n)^2} \tag{1}$$

decreases also like $1/\sqrt{n}$, as $\frac{1}{n}\sum_{i=1}^{n}(X_i - Z_n)^2$ converges weakly to a Gaussian random variable. Thus the logarithm of CI_n would be linear in $\log(n)$.

We plot in Figure 11 on a log-log scale the confidence interval as computed in (1). In the left part of Figure 11, X_i are exponentially distributed where as in the right part, they have Pareto distribution with parameter $K = 1.5$. We indeed observe this linear behavior for the exponential random variable (for which a CLT exists) and we do not have convergence for the Pareto one.

What happens when the second moment does not exist? One can still use equation (1) to compute a confidence interval, it will be finite for each n and by the Strong Law of Large Numbers. However, the probability that the average sample belongs to that interval is no more 0.95 asymptotically (for large n); instead we have to use the quantile approach to estimate it.

We now go back to discuss the simulation of the average number of packets in a processor sharing queue or the number of TCP connections sharing a bottleneck link. Unless otherwize stated, we ran 100 independent simulations for each experiment that is described in this paper. We did the experiments for various values of simulation duration t. Each one of the 100 simulations provides another simulated value for the sample average of the number of sessions till time t. For fixed t, the average number of ongoing sessions till time t *has finite second moment* even for a Pareto distributed session size with parameter smaller than 2. Therefore a CLT exists for each fixed t. The bootstrap approach then gives a better confidence interval even in that case, when the number of simulations is small. We illustrate this in the left part of Figure 12 for which the number of simulations was 5. As the number of simulations become large, the CLT approach gives confidence intervals very close to those obtained by applying the quantile method to the bootstrap approach, and considerably more precise than those obtained by the quantile approach (without the bootstrap). We illustrate this in the right part of Figure 12. The large difference between the two shows that the CLT approach (which is asymptotic in nature) does not give a good estimation of the empirical confidence interval.

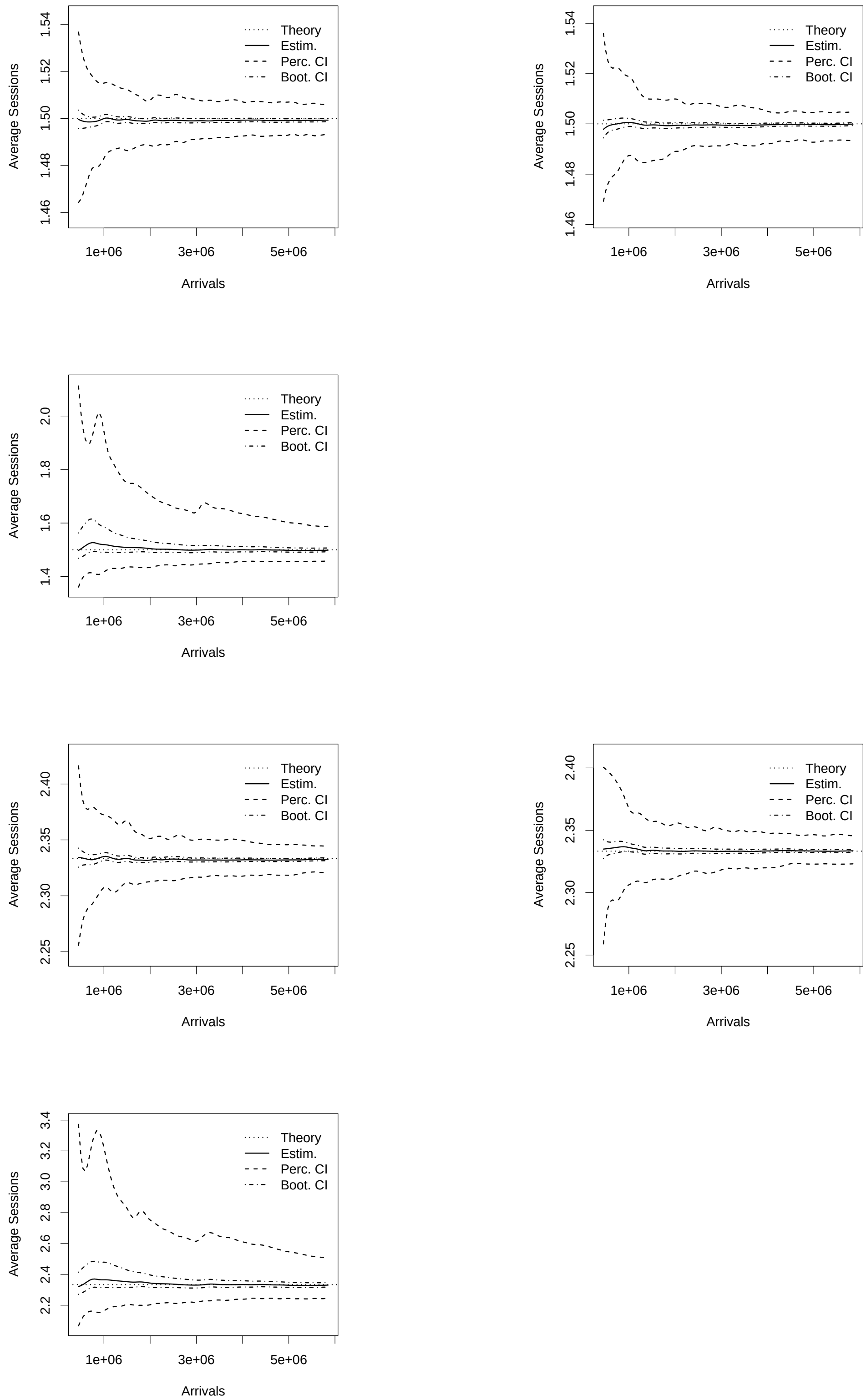

Figure 1: Average number of sessions with $\rho = 0.6$ (up) and $\rho = 0.7$ (down). Exponential distribution (left) and Pareto distribution with parameter $K = 2.5$ (middle) and $K = 1.5$ (right)

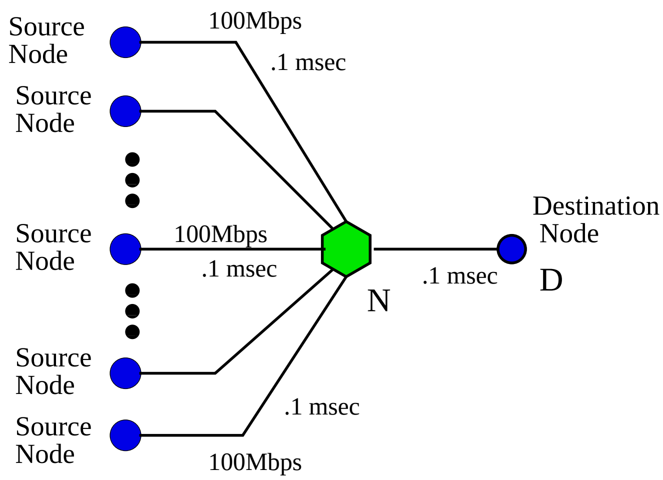

Figure 2: Network Topology

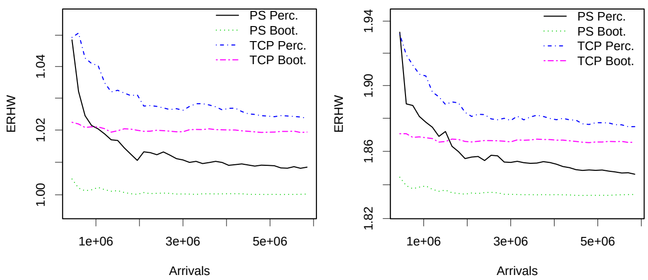

Figure 3: ERHW for exponentially distributed session size. $\rho = 0.6$ (left) and $\rho = 0.7$ (right).

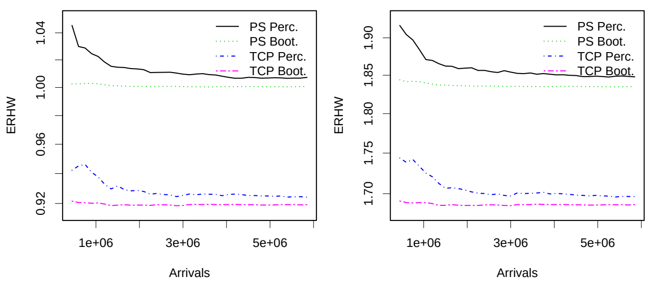

Figure 4: ERHW for Pareto distributed session size with K=2.5; $\rho = 0.6$ (left) $\rho = 0.7$ (right)

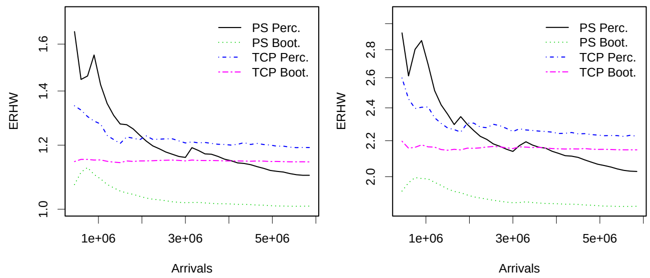

Figure 5: ERHW for Pareto distributed session size with K=1.5; $\rho = 0.6$ (left) and $\rho = 0.7$ (right)

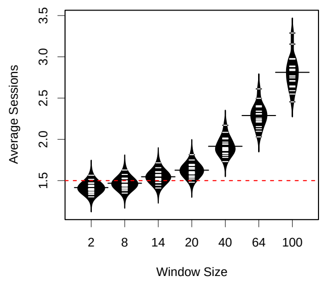

Figure 6: Average number of ongoing connections as a function of the maximum TCP window size.

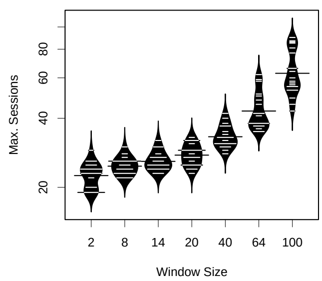

Figure 7: Maximum number of ongoing connections as a function of the maximum TCP window size.

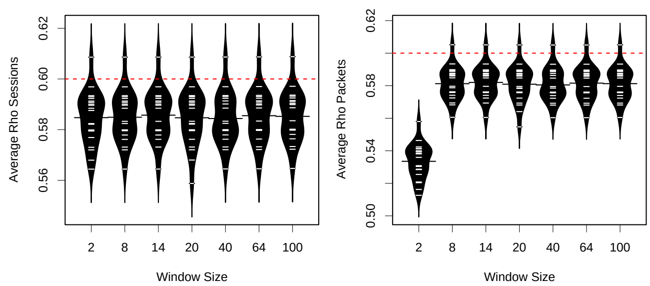

Figure 8: Fraction of arrivals of sessions that found the system non-empty upon arrival, and the fraction of arrivals that found the bottleneck queue non-empty, as a function of the maximum window size.

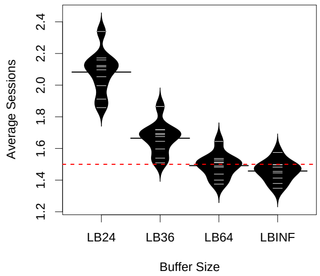

Figure 9: Average number of ongoing connections as a function of the queue size. Confidence intervals are obtained by the quantile approach.

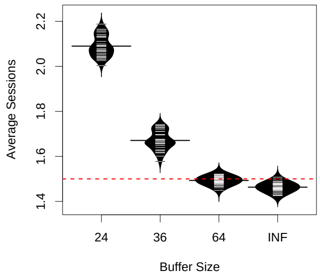

Figure 10: Average number of ongoing connections as a function of the queue size. Confidence intervals are obtained from the bootstrap.

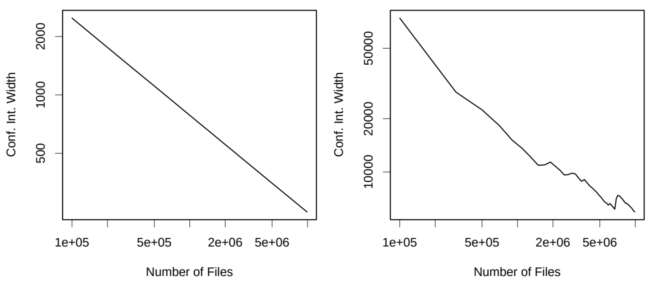

Figure 11: Confidence Interval for the sample average of X_i as a function of number of i.i.d. elements. X_i is exponentially distributed (left) and has a Pareto distribution (right) with parameter 1.5.

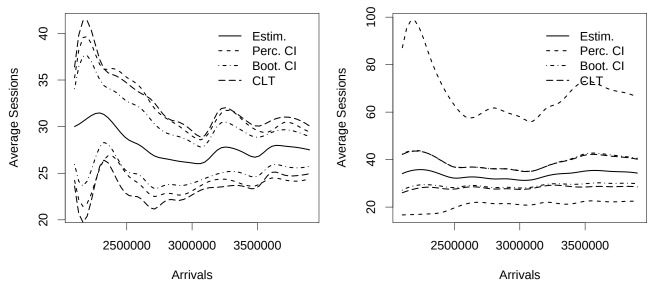

Figure 12: Comparison of various ways to compute confidence intervals. Left: 5 simulations. Right: 30 simulations. In both cases $\rho = 0.95$ and session size are Pareto distributed with $K = 1.5$. We compare the CI obtained with the percentile approach (Perc CI), the bootstrap approach (Boot CI) and the CLT approach. The sample averaged over time till time t, and then averaged over the number of simulations is depicted too (Estim).

Centre de recherche INRIA Sophia Antipolis – Méditerranée
2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Centre de recherche INRIA Bordeaux – Sud Ouest : Domaine Universitaire - 351, cours de la Libération - 33405 Talence Cedex
Centre de recherche INRIA Grenoble – Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier
Centre de recherche INRIA Lille – Nord Europe : Parc Scientifique de la Haute Borne - 40, avenue Halley - 59650 Villeneuve d'Ascq
Centre de recherche INRIA Nancy – Grand Est : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex
Centre de recherche INRIA Paris – Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex
Centre de recherche INRIA Rennes – Bretagne Atlantique : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex
Centre de recherche INRIA Saclay – Île-de-France : Parc Orsay Université - ZAC des Vignes : 4, rue Jacques Monod - 91893 Orsay Cedex

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399