Performance analysis of a queue by combining stochastic bounds, real traffic traces and histograms

Farah Ait Salaht, Hind Castel-Taleb, Jean-Michel Fourneau, Nihal Pekergin

Performance analysis of a queue by combining stochastic bounds, real traffic traces and histograms

Farah Aït-Salaht[1], Hind Castel-Taleb[2], Jean-Michel Fourneau[3] and Nihal Pekergin[4]

[1]*LIP6, Univ. Paris Ouest Nanterre, France*
[2]*SAMOVAR, CNRS, Télécom Sud Paris, Evry, France*
[3]*DAVID, Univ. Versailles St Quentin, Versailles France*
[4]*LACL, Univ. Paris Est, Créteil, France*
Email: farah.aitsalaht@u-paris10.fr, Hind.Castel@it-sudparis.eu, jmf@uvsq.fr & nihal.pekergin@u-pec.fr

We present an approach to derive performance bounds of a queue under histogram-based input traffics. The results are obtained through strong stochastic bounds on the queue length and on the output traffic. The bounds provide probability inequalities on transient behaviors and on steady-state when it exists. We consider both stationary and non stationary traffics and provide some numerical techniques in both cases. Unlike approximate methods, these bounds can be used to check if the Quality of Service constraints are satisfied or not. Our approach provides a trade-off between the accuracy of results and the computational complexity and it is much faster than the histogram-based simulation.

Keywords: Performance Evaluation, Stochastic Bounds, Measurements

Received 00 January 2009; revised 00 Month 2009

1. INTRODUCTION

Nowadays, more and more applications are based on network technologies that require high data rates, such as video on demand, health care applications, or financial transactions. Network performance has an important impact on such applications and may cause quality problems which lead to customer dissatisfactions. One important research area in the context of performance evaluation and network dimensioning is to develop accurate traffic models in order to design networks ensuring the required Quality of Service (QoS). Thus, it is essential that the underlying models reflect as much as possible the relevant characteristics of the traffic.

We propose to use some measurements on the traffic instead of well-known stochastic processes currently used in queuing theory. Usually, one derives from the measurements a complex arrival process with a fitting algorithm.

This arrival process is then integrated into a so-called structured Markov chain which models the queue. Many algorithms have been derived to solve the steady-state distribution for this type of queues (see [1] and references therein).

In this paper, we propose a different approach: we use directly the measurements to obtain a distribution of arrivals during a time slot. This direct integration of the measurements into the model without a fitting procedure is an important aspect of the approach. We claim that fitting measurements to parametrize a stochastic process gives an approximation. Such an approximation on the processes involved in the queueing model may lead to incorrect results (see [2] for service time distributions approximated by a Gaussian distribution). Instead, we construct empirical distributions which are stochastic bounds of the traffic distribution, while Hernàndez et al. [3, 4, 5] only build approximations of this distribution. Thanks to the monotonicity of the model, we use these empirical distributions in the queuing model to obtain upper and lower bounds on the performance indices. Bounding methods are sufficient to provide dimensioning solutions which guarantee the performance measure constraints and they reduce the computational times. We advocate that the traffic characteristics are much better described by discrete distributions than the two moments which are used as parameters for a queue with general distribution service or arrivals.

We do not assume the stationarity of the arrival

traffic to be as general as possible. The main results are obtained for the transient distributions of the queue length and the output traffic (departure flow) at any time. Under the stationarity traffic assumption, we obtain more powerful results (see Section 4). If the traffic is stationary or upper bounded by a stationary traffic, we obtain bounds using the steady-state distribution of an ergodic Markov chain. To avoid the curse of dimensionality, we propose to bound the traffic process by two simpler processes which can be numerically handled. These processes are used for the analysis of the transient regime and the steady-state as well. We propose new algorithms which are based on the computation of the transient distributions and which contain a steady-state detection test based on the coupling of two sample-paths.

The histogram based approach for traffic modeling and performance analysis had been introduced in the literature 20 years ago. The first work was proposed by Skelly et al. [6] in the area of network calculus to model the video sources and to predict buffer occupancy distributions. More recently, Hernàndez et al. [3, 4, 5] have proposed a performance analysis model to obtain histograms of buffer occupancy. Their approach, called HBSP (Histogram Based Stochastic Process) works directly with histograms using a set of specific operators in discrete time. It is based on a basic histogram model (called HD) for the input traffic, which enters a finite capacity buffer to receive a constant service under the First Come First Served (FCFS) server policy from a multi-server. The analysis is limited to a single node. The method consists in solving numerically the HD/D/1/K queue. As the state space and the size of the traffic description are too large, the authors approximate the histogram of traffic by a smaller one to have a numerical algorithm with a smaller complexity.

A similar approach was proposed by Tancrez et al. [7] for the performance analysis of production lines which are stochastic event graphs. Continuous distributions are discretized by dividing the support into equal subintervals and by mapping each subinterval into one single point. The mass probability of any subinterval is associated with the upper limit or the lower limit to provide strong stochastic bounds. The analysis relies on the stochastic monotonicity of the stochastic event graphs.

The technical part of the paper is organized as follows. In section 2, we introduce the queuing model and the various assumptions that we make about the input traffic. Section 3 is devoted to the description of our methodology to construct bounding histograms. The algorithm presented in [8] is based on a dynamic programming approach and it computes an optimal bounding histogram with a given size. We also prove stochastic comparisons on transient distributions when we replace the traffic by a stochastically larger or smaller arrival process. In Section 4, we analyze the stationary traffic. This is a necessary step to study more complicated traffics such as the weak stationary traffic in section 5 and to prove our numerical algorithm. We also present the existing approximative histogram reduction method (HBSP) in section 4.3 in order to compare it with our bounding approach. This part of the paper is an extension of [9]. We show that we have a trade-off between the accuracy of bounds and the computational complexity. Finally in Section 5, we show how we can compute very easily the performance indices when the traffic is only stationary during short periods of time. This approach generalizes the algorithm presented for the stationary traffic and avoids many computation steps to be much more efficient than the simulation of the underlying trace. The same arguments (i.e. stochastic monotonicity, stochastic bounds) are used to simplify the numerical computations for the steady-state in Section 4 and the step by step analysis of the transient regime with steady-state detection in Section 5. We advocate that associated with measurements and their statistical analysis, these arguments provide an efficient technique for network dimensioning.

2. QUEUEING MODEL

Following Hernández-Orallo [3, 4, 5], we use a discrete time queueing model. The number of transmission units produced by the corresponding traffic source during the k^{th} slot is denoted by discrete random variable $A(k)$. $Q(k)$ and $D(k)$ denote respectively the buffer length and the output (departure) traffic during the k^{th} slot. These output parameters are also derived as discrete random variables. The buffer size is noted by B and the service capacity during a slot by S. We now give

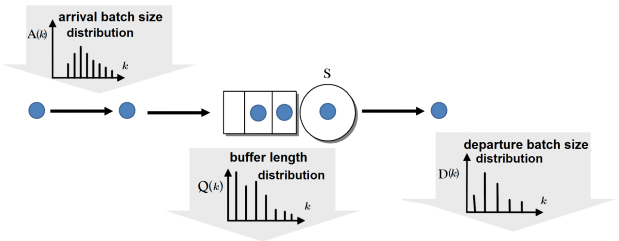


FIGURE 1: Input and output parameters of a queueing model

the evolution equations for the buffer length and the departure traffic. As we consider a discrete-time model, we have to describe the exact sequence of events during a slot. We assume that the arrivals occur first and they are followed immediately by services. We assume that the admission is performed per packet with Tail Drop policy: a packet is accepted if there is a place in the buffer, otherwise it is rejected. The buffer length (occupancy) $Q(k)$ can be expressed with the following recursive formula:

$$Q(k) = \texttt{min}(B, (Q(k-1) + A(k) - S)^+), \quad k \geq 1. \quad (1)$$

where operator $(X)^+ = \max(X, 0)$. Similarly, the output traffic $D(k)$ is:

$$D(k) = \min(S, Q(k-1) + A(k)), \quad k \geq 1. \qquad (2)$$

We assume that the input arrivals are independent of the current queue state and the past of the arrival process. Under these assumptions, the model of the queue is a time-inhomogeneous Discrete Time Markov Chain (DTMC).

In this paper we want to cope with some non stationary arrival processes. We advocate that monotonicity of the evolution equation (defined in the next sections) as well as stochastic bounds may help to solve such a queueing model when the arrival process is not stationary. However, we have to consider first the stationarity assumption to derive some results, theorems and algorithms which will be then generalized for non stationary arrival processes. More precisely, we state in the next section the monotonicity of the queueing model for the transient analysis and we derive bounds for the transient distribution of the queue length and the output traffic. Note that the stochastic ordering in distribution (defined in the appendix) will de denoted as $\leq_{st}$ while the equality will be denoted by $=_{st}$. We consider in the queueing model three cases for the arrival process:

1. First, we assume the stationarity of the arrival process to state the theoretical results and to present the simplest algorithm. For all k, we have:

   $$A(k) =_{st} \mathcal{A}$$

   As arrivals are stationary and identically independently distributed (i.i.d.), the underlying model is a time-homogeneous DTMC taking values in a totally ordered state space. Without loss of generality, we suppose in the following that the considered DTMC models are ergodic.
2. We just assume that the $A(k)$ are independent and all upper bounded by a common stationary arrival process $\mathcal{A}$. For all k, we have:

   $$A(k) =_{st} \mathcal{A}^k \quad and \quad \mathcal{A}^k \leq_{st} \mathcal{A}$$

   We will see that the analysis of the model under the stationary, bounding traffic provides some stochastic performance guarantees.
3. A common assumption in traffic modeling is a kind of weak stationarity: on short time scale the traffic is stationary, while for longer periods it is not. Such an assumption is consistent with the night and day evolution of traffic observed by long traces. In that case, we have for all time instant k in a given time interval I:

   $$A(k) =_{st} \mathcal{A}^I$$

   The process in that case is piecewise stationary. Note that during period I, the underlying model is a time-homogeneous DTMC, thus one can use the method developed for the first case to perform the analysis during transient period I.

Finally, we are interested in the performance analysis of the queue under real traffic traces. We use several strategies to extract the histogram-based traffic model from these traces according to the assumptions we made on the nature of the traffic. We illustrate in Figure 2, a real trace extracted from the MAWI traffic [10]. Precisely, it corresponds to an IP measurements during one hour for a 150 Mbps transpacific line (samplepoint-F) for the 9th of January 2007 between 12:00 and 13:00. This traffic trace has an average rate of 109 Mbps. Using a sampling interval of T = 40 ms (25 samples per second), the resulting traffic trace has 90,000 frames (periods), an average of 4.37 Mb per frame and 80511 distinct values.

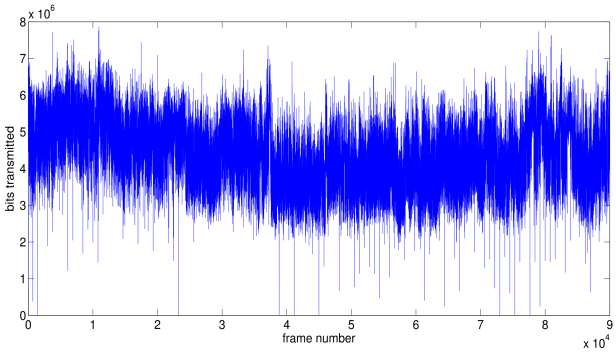


FIGURE 2: MAWI traffic trace.

3. BOUNDING APPROACH

The complexity of the numerical analysis depends on the size of the arrival distributions whatever the method we may use. We advocate that it is possible to aggregate in some sense the distribution to obtain stochastic bounds on the performance measures in an easier way. We first present how we can bound a distribution with the stochastic ordering. Then, we prove the monotonicity of the queueing model. Intuitively, this property implies that if the arrival traffic "increases" in some sense, so does some other quantities such as the queue length, the output flow and the loss probabilities.

3.1. Bounding histogram construction

In order to reduce the computation complexity, we propose to apply the bounding approach to diminish the size of the support of the input histogram (distribution) used during the computation. We use in the sequel the term of *bins* to indicate the states of a distribution. The main advantage of this approach is the computation of bounds rather than approximations. Unlike approximations, the bounds allow us to check if

QoS are satisfied or not. We consider bounds for the $\leq_{st}$ ordering (see Appendix). For a given probability mass function (discrete distribution or histogram) $\boldsymbol{d}$ defined on N bins, an upper and a lower bounding distribution, $\boldsymbol{d}1$ and $\boldsymbol{d}2$ with $n << N$ bins are built. Moreover, $\boldsymbol{d}1$ and $\boldsymbol{d}2$ are the optimal bounds with respect to a given positive, increasing reward function, $\boldsymbol{r}$. Formally, for a given distribution $\boldsymbol{d}$ defined on $\mathcal{H}$ ($|\mathcal{H}| = N$), we compute bounding distributions $\boldsymbol{d}1$ and $\boldsymbol{d}2$ defined respectively on $\mathcal{H}^u$, $\mathcal{H}^l$ ($|\mathcal{H}^u| = n$, $|\mathcal{H}^l| = n$) such that:

1. $\boldsymbol{d}2 \leq_{st} \boldsymbol{d} \leq_{st} \boldsymbol{d}1$,
2. $\sum_{i\in\mathcal{H}} \boldsymbol{r}(i)\boldsymbol{d}(i) - \sum_{i\in\mathcal{H}^l} \boldsymbol{r}(i)\boldsymbol{d}2(i)$ is minimal among the set of distributions on n bins that are stochastically lower than $\boldsymbol{d}$,
3. $\sum_{i\in\mathcal{H}^u} \boldsymbol{r}(i)\boldsymbol{d}1(i) - \sum_{i\in\mathcal{H}} \boldsymbol{r}(i)\boldsymbol{d}(i)$ is minimal among the set of distributions on n bins that are stochastically upper than $\boldsymbol{d}$.

Notice that $\forall\, i \in \mathcal{H}$ and $i \notin \mathcal{H}^u$ (resp. $\forall\, i \in \mathcal{H}$ and $i \notin \mathcal{H}^l$), $\boldsymbol{d1}(i) = 0$ (resp. $\boldsymbol{d2}(i) = 0$) to establish the stochastic comparisons (Definition A.1 of the Appendix). Thus $\boldsymbol{d}1$ and $\boldsymbol{d}2$ denote the optimal bounding distributions on n bins with respect to reward $\boldsymbol{r}$.

An algorithm to construct bounding distributions satisfying the constraints 1, 2 and 3 has been proposed in [8]. This algorithm is based on dynamic programming and has a complexity of $O(N^2\, n)$. In the following example, the upper bounding distribution obtained by this bounding algorithm is illustrated. The number of bins in the bounding histograms is fixed in the algorithm. A good number of bins satisfying the required trade-off between the accuracy of the bounds and the computation time can be determined in an incremental manner: one begins with a reduced number of bins, if the accuracy of bounds is not satisfactory, the number of bins can be incremented. The iteration can be stopped, if the the required accuracy is reached and/or the computation time of bounds exceeds a fixed threshold.

Example 1. Let $\boldsymbol{d} = [0.3, 0.1, 0.1, 0.1, 0.2, 0.2]$ be a discrete distribution defined on support $\mathcal{H} = \{1, 2, 3, 4, 5, 6\}$ ($|\mathcal{H}| = N = 6$). The reward function $\boldsymbol{r}$ is set to $\boldsymbol{r}(i) = a_i$, $\forall\, a_i \in \mathcal{H}$. The expected reward of distribution $\boldsymbol{d}$ is then $R[\boldsymbol{d}] = \sum_{a_i\in\mathcal{H}} \boldsymbol{r}(i)\ \boldsymbol{d}(i) = 3.4$.

The computation of the optimal upper bounding distribution, $\boldsymbol{d}1$ defined on 4 bins is equivalent to determine the 4-hops path having $R[\boldsymbol{d}1] - R[\boldsymbol{d}]$ minimal. We illustrate in Figure 3 the tree explored by the algorithm proposed in [8] to define the optimal upper bound distribution. In this figure, each path from the root to a leaf represents a distribution with correct reduced size. We note that the probability of a deleted bin (the state that do not appear in the path) is added to its immediate successor (to its immediate predecessor, in the lower bounding case). The optimal upper bound algorithm [8] determines the distribution

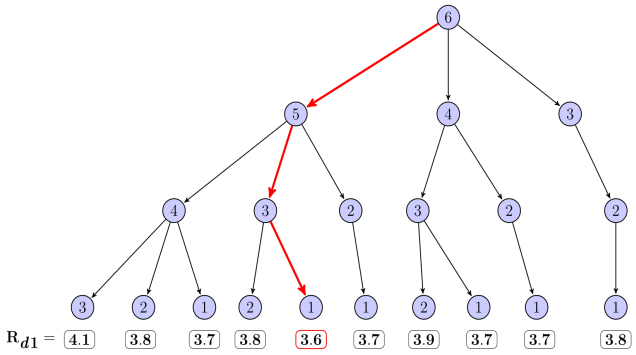


FIGURE 3: The tree explored to define the optimal 4 single hops path.

$\boldsymbol{d}1 = [0.3, 0.2, 0.3, 0.2]$ defined on $\mathcal{H}^u = \{1, 3, 5, 6\}$, with $R[\boldsymbol{d}1] = 3.6$.

From Figure 4, one can see that the cumulative distribution function of the upper bound $\boldsymbol{d}1$ is always greater or equal to that of $\boldsymbol{d}$. Therefore it follows from Definition A.1 of the Appendix that $\boldsymbol{d}1$ is a stochastic upper bound of $\boldsymbol{d}$.

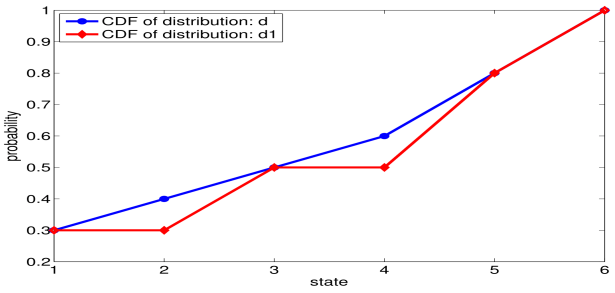


FIGURE 4: Cumulative distribution functions (cdf) of $\boldsymbol{d}$ and $\boldsymbol{d}1$.

3.2. Monotonicity of the queueing model

In this section, we prove that we can use bounding histograms for the arrival process and obtain bounds on the other histograms such as queue length, departure traffic. The evolution equations (Eq. 1, and Eq. 2) correspond to the original system under input arrival $A(k)$. For bounding models, the same evolution equations are considered under the bounding input arrival process $\tilde{A}(k)$. We note by $\tilde{Q}(k)$ (resp. $\tilde{D}(k)$) the buffer length (resp. the departure traffic) for the bounding system under bounding arrival $\tilde{A}(k)$:

$$\tilde{Q}(k) = \texttt{min}(B, (\tilde{Q}(k-1) + \tilde{A}(k) - S)^+), \quad k \geq 1,$$

$$\tilde{D}(k) = \texttt{min}(S, \tilde{Q}(k-1) + \tilde{A}(k)), \quad k \geq 1.$$

The relationship between the original and the bounding systems is established by the fact that the $\leq_{st}$ order is associated with increasing functions and the underlying

measures are defined by increasing functions. We present here only the upper bounding case, and the lower bounding case can be similarly derived. In the following theorems and the related corollaries, we assume that at the beginning, $Q(0) \leq_{st} \tilde{Q}(0)$ and $D(0) \leq_{st} \tilde{D}(0)$.

THEOREM 3.1. *[Monotonicity of the Queue length] If* $A(k) \leq_{st} \tilde{A}(k), \quad \forall k > 0$, *then* $Q(k) \leq_{st} \tilde{Q}(k), \quad \forall k > 0$.

Proof. The proof is a direct consequence of Theorem 4.3.9, page 163 in [11]. We can write:

$$Q(k) = \Psi(Q(k-1), A(k)). \tag{3}$$

It follows from Eq. 1 that function Ψ is increasing both in the first and the second parameter (with respect to $Q(k-1)$ and $A(k)$). We assume that $Q(0) \leq_{st} \tilde{Q}(0)$, and $A(k) \leq_{st} \tilde{A}(k)$. The proof goes by induction. Assume that $Q(k-1) \leq_{st} \tilde{Q}(k-1)$. Since Ψ is increasing (see Property 2 in the appendix):

$$\begin{aligned} Q(k) = \quad & \Psi(Q(k-1), A(k)) \leq_{st} \Psi(\tilde{Q}(k-1), A(k)) \\ & \leq_{st} \Psi(\tilde{Q}(k-1), \tilde{A}(k)) = \tilde{Q}(k). \end{aligned}$$

Thus the queue length under arrival $\tilde{A}(k)$ is greater or equal in the sense of the $\leq_{st}$ ordering than the queue length under arrival $A(k)$: $Q(k) \leq_{st} \tilde{Q}(k)$. □

We have a similar result for the lower bounds.

COROLLARY 3.1. *If* $\tilde{A}(k) \leq_{st} A(k), \quad \forall k > 0$, *then* $\tilde{Q}(k) \leq_{st} Q(k), \quad \forall k > 0$.

Moreover, it follows from Eq. 2 that we have bounds on the departure flows. The proof is based on the same arguments that we have used for Theorem 3.1, and it is omitted here for the sake of readability.

THEOREM 3.2. *[Monotonicity of the Output Flow] If* $A(k) \leq_{st} \tilde{A}(k), \quad \forall k > 0$, *then* $D(k) \leq_{st} \tilde{D}(k), \quad \forall k > 0$.

The above theorems assert the $\leq_{st}$ comparison for the transient behaviors of the original and bounding systems. Note that, we have only assumed that the arrivals are independent, we do not make any assumption on the stationarity of the arrival process. The various assumptions that we will make in the following sections allow us to derive corollaries of these main results. For instance, when the steady-state exists, one may obtain the stochastic comparison of the steady-state distributions.

4. ASSUMING THE STATIONARITY OF THE INPUT TRAFFIC

This section has several objectives. First, we want to state the main results for the stationary traffic. We also consider the case of a non stationary traffic stochastically bounded by a stationary process. Second, we present the numerical algorithm based on convolution and stochastic bounds. The main advantage of this algorithm is that it has a proven convergence test. It is also generalized for non stationary traffic in Section 5. Finally, we compare our results to an existing, approximative method (HBSP method) which is also based on the reduction of the size of the traffic distribution [3]. In the following δ_0 and δ_B are two distributions of probability for the queue length such that $\delta_0[0] = 1.0$ and $\delta_B[B] = 1.0$ where B denotes the buffer size.

4.1. Comparison results

We now give the comparison results as corollaries of Theorems 3.1 and 3.2. The first result asserts the $\leq_{st}$ comparison for the steady-state case when both traffics (the real one and the bounding one) are assumed to be stationary.

COROLLARY 4.1. *Let* $\mathcal{A}$ *(resp.* $\tilde{\mathcal{A}}$*) be the stationary exact (resp. upper bounding) input histogram (distribution) such that* $\mathcal{A} \leq_{st} \tilde{\mathcal{A}}$, *and* $\mathcal{Q}, \mathcal{D}$ *(resp.* $\tilde{\mathcal{Q}}, \tilde{\mathcal{D}}$*) be the stationary buffer length, departure flow under the exact* $\mathcal{A}$, *(resp. upper bounding* $\tilde{\mathcal{A}}$*) input arrival. If* $Q(0) \leq_{st} \tilde{Q}(0)$, *and* $D(0) \leq_{st} \tilde{D}(0)$, *then we have:*

$$\mathcal{Q} \leq_{st} \tilde{\mathcal{Q}} \text{ and } \mathcal{D} \leq_{st} \tilde{\mathcal{D}}.$$

Proof. Since at each time k, the arrivals are distributed by $\mathcal{A}$ and $\tilde{\mathcal{A}}$ and by construction $\mathcal{A} \leq_{st} \tilde{\mathcal{A}}$, we have $A(k) \leq_{st} \tilde{A}(k), \quad \forall k > 0$. Thus, the conditions of Theorems 3.1 and 3.2 are satisfied, and we have comparisons for all k:

$$Q(k) \leq_{st} \tilde{Q}(k) \quad \text{and } D(k) \leq_{st} \tilde{D}(k).$$

Remark that due to the ergodicity assumption, $\mathcal{Q}$ and $\mathcal{D}$ exist, thus $Q(k)$ and $D(k)$ converges in distribution to $\mathcal{Q}$ and $\tilde{\mathcal{Q}}$ when $k \to \infty$. Since the $\leq_{st}$ ordering is closed under the convergence in distribution, we have:

$$\mathcal{Q} \leq_{st} \tilde{\mathcal{Q}} \text{ and } \mathcal{D} \leq_{st} \tilde{\mathcal{D}}.$$

□

We can also obtain bounds when the traffic is non stationary but it is bounded by a stationary distribution. In this case the bounding model is constructed by considering that at each time, the input arrivals are distributed independently, identically with an upper bounding stationary histogram $\tilde{\mathcal{A}}$:

COROLLARY 4.2. *Let* $\mathcal{A}^k$ *be the input histogram at time* k *and* $\tilde{\mathcal{A}}$ *be an upper bounding histogram for all input histograms:*

$$\mathcal{A}^k \leq_{st} \tilde{\mathcal{A}}^k =_{st} \tilde{\mathcal{A}}, \quad \forall k > 0 \tag{4}$$

and $\mathcal{Q}^k, \mathcal{D}^k$ *be the histograms of the buffer length and the departure flow at time* k *under arrival histograms*

$\mathcal{A}^k$, while $\tilde{\mathcal{Q}}$ and $\tilde{\mathcal{D}}$ be the histograms of the stationary buffer length and the departure flow processes under the upper bounding, stationary arrival distribution $\tilde{\mathcal{A}}$. If at time 0, both $\mathcal{Q}^0$ and $\tilde{\mathcal{Q}}^0$ are equal to the Dirac distribution at 0: $\mathcal{Q}^0 =_{st} \tilde{\mathcal{Q}}^0 =_{st} \delta_0$, then

$$\mathcal{Q}^k \leq_{st} \tilde{\mathcal{Q}} \text{ and } \mathcal{D}^k \leq_{st} \tilde{\mathcal{D}}, \quad \forall k.$$

Proof. It is similar to the previous case, and only the buffer length case is given. Due to the condition on arrivals (Eq. 4), it follows from Theorem 3.1 that

$$\mathcal{Q}^k \leq_{st} \tilde{\mathcal{Q}}^k, \quad \forall k. \tag{5}$$

However, only the bounding model converges in distribution. Since at time 0, $\tilde{\mathcal{Q}}^0 \leq_{st} \tilde{\mathcal{Q}}$ and $\tilde{Q}(k)$ is constructed by an increasing function (Eq. 3),

$$\tilde{\mathcal{Q}}^0 \leq_{st} \tilde{\mathcal{Q}}^1 \leq_{st} \tilde{\mathcal{Q}}^2 \leq_{st} \cdots \leq_{st} \tilde{\mathcal{Q}}^\infty =_{st} \tilde{\mathcal{Q}}$$

Therefore Eq. 5 can be rewritten as

$$\mathcal{Q}^k \leq_{st} \tilde{\mathcal{Q}}, \ \forall k. \qquad \square$$

4.2. Numerical analysis

Let g_{min} and g_{max} be two distributions of probability representing indeed two realizations of the same stochastic process with two distinct initial values. When they become equal, they couple and they will stay equal for the remaining life of the process. As we assume ergodicity, this implies that the value obtained after the coupling is equal to the steady-state distribution of the Markov chain. The following theorem states that g_{min} and g_{max} provide at each time step t, two stochastic bounds for the transient distributions, $\pi^{(t)}$ of the underlying process.

1. Initialize t=0
2. Initialize $g_{min}^{(0)} = \delta_0$
3. Initialize $g_{max}^{(0)} = \delta_B$
4. Initialize $\pi^{(0)}$ (see the paragraph below)
5. Iterate
   (a) increase time instant t
   (b) compute new $g_{min}^{(t)}$ with arrival distribution $\mathcal{A}$ and distribution $g_{min}^{(t-1)}$
   (c) compute new $g_{max}^{(t)}$ with arrival distribution $\mathcal{A}$ and distribution $g_{max}^{(t-1)}$
   (d) compute new $\pi^{(t)}$ with arrival distribution $\mathcal{A}$ and distribution $\pi^{(t-1)}$
6. Until $(g_{min} = g_{max})$ or $(t = EndOfTime)$

The first stopping condition implies that we have a coupling of the two sample paths and a convergence of the numerical algorithm. The second condition means that we have not observed convergence at the end of the execution. In that case, the computation time must be increased. This typically occurs for very short traces as we have always observed that the coupling time is very small compared to the time for the measurements.

Generally, we have $\pi^{(0)} = \delta_0$ which means that the queue is empty at the beginning. But this is not necessary and we consider a general value for $\pi^{(0)}$ to help the generalization in Section 5. We represent in Fig. 5 the behavior of the distributions during the computation.

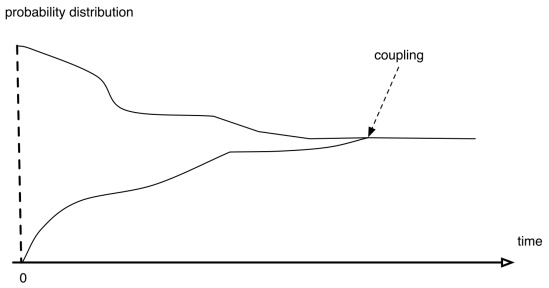


FIGURE 5: Convergence and coupling.

Theorem 4.1. *For all t, we have $g_{min}^{(t)} \leq_{st} \pi^{(t)} \leq_{st} g_{max}^{(t)}$*

Proof. Clearly, this is true for $t = 0$ as δ_0 and δ_B are the extremal values under the $\leq_{st}$ ordering for the underlying distribution. The induction proof comes from the monotonicity property of the operator. Indeed, if $g_{min}^{(t0)} \leq_{st} \pi^{(t0)}$ for $t0$, the monotonicity of the model implies that $g_{min}^{(t0+1)} \leq_{st} \pi^{(t0+1)}$ (see [9] for more details). $\square$

We advocate that our numerical method, which is based on the stochastic monotonicity of the model, has many advantages compared with well-known numerical techniques. First, we compute both the transient distributions and the steady-state distribution. Our method gives results for the transient analysis as we also compute $\pi^{(t)}$ for all t and we provide a test of the stationarity of the distribution to avoid computing the transient distribution once we have reached the steady-state. In some sense, this numerical procedure is inspired by the stationarity detection heuristic proposed by Ciardo et al. in [12] and improved by Sericola in [13] for the efficient computation of reliability. In our case, the stationarity is proved by the coupling while it was only a numerical test in Ciardo's approach. Furthermore we have a proved test of convergence in the following sense: when we stop the algorithm at step t, we have the proof that the result is within the interval $[g_{min}^{(t)}, g_{max}^{(t)}]$. We do not have such a result with iterative techniques where the convergence test consists in computing the difference between two successive distributions $\pi^{(t)}$ and $\pi^{(t-1)}$ (*i.e.* checking the norm of $(\pi^{(t)} - \pi^{(t-1)})$). As stated in [1], such a method is not an accurate test for convergence, and it may provide incorrect numerical results.

Finally, we have to address the complexity of the computation of $g_{min}^{(t)}$, $\pi^{(t)}$ and $g_{max}^{(t)}$. We have chosen

to use the convolution operator rather than a vector-matrix product for the sake of performance. Let l_X (resp. l_Y) be the size of the distribution of the queue (respectively the arrivals), then the matrix has approximatively $l_X \times l_Y$ non zero elements and the complexity of the vector matrix product in a sparse format is $O(l_X \times l_Y)$. The computation based on convolution is much simpler as stated in the following property.

PROPERTY 1. The convolution (noted $\otimes$) of the distributions of two independent random variables X and Y is a distribution with at most $l_X \times l_Y$ bins. This computation requires $O((l_X + l_Y)log(l_X + l_Y))$ with a Fast Fourier Transform (FFT) approach [14].

4.3. Approximative histogram reduction: HBSP method

Considering a real traffic trace, Hernández et al. have proposed in [3, 4, 5] to use the histogram approach and the stochastic process for characterizing network traffic and analyzing the performance of the model. They approximate the histogram of buffer occupancy for a finite capacity queue. If the original histogram denoted by A has a range of $I = [0, N_{max}]$ bins, the method proposes to define an interval size of $l_{\mathbf{A}} = N_{max}/n$ such that a binned process $\{A(k)\}$ has a reduced state space $I' = \{0, \ldots (n-1)\}$. A value a of I is mapped to i in I' such that $i = \lfloor \frac{a}{l_{\mathbf{A}}} \rfloor$, which is also denoted by $i = class_A(a)$. Inversely, a value $i \in I'$ corresponds to the midpoint of interval i: $a = l_{\mathbf{A}} \,.\, i + l_{\mathbf{A}}/2$, $a \in I$.

Assuming that the traffic is stationary, the arrival process is given by $A(k) =_{st} \mathcal{A}, \forall k$. We denote by $d\mathcal{A}$ a distribution associated with the stationary input arrival, $\mathcal{A}$. The stochastic process of the evolution of HBSP model (denoted by distribution dQ) is based on the following recurrence relation:

$$dQ(k) = \Phi_{\hat{S}}^{\hat{b}}(dQ(k-1) \otimes d\mathcal{A}). \tag{6}$$

where, $\hat{S} = class_{\mathbf{A}}(S)$ (resp. $\hat{b} = class_{\mathbf{A}}(B)$). The operator Φ limits buffer lengths so that they cannot become negative and cannot overflow the corresponding class of buffer capacity. This operator is defined as follows:

$$\Phi_a^b(x) = \begin{cases} 0, & \text{for} \quad x < a, \\ x - a, & \text{for} \quad a \le x \le b + a, \\ b, & \text{for} \quad x \ge b + a. \end{cases} \tag{7}$$

Hernández et al. have also proposed an improvement for their method and defined the notion of *oversampling*. An $m-$*oversampling* consists in splitting each class i of the HBSP histogram having probability $p(i)$ into m classes with equal probability $p(i)/m$. For example, let LH be the computed HBSP histogram with $n = 10$ bins. The use of an *oversampling* factor of 10, means to define an histogram on 100 bins. When we increase the number of bins we increase consequently the accuracy of the results. However, we should note that the results obtained by the HBSP method (without *oversampling*) on 100 bins is more accurate than the results obtained by using the HBSP histogram defined on 10 bins with an oversampling factor of 10. For this reason, in the rest of the paper, we will directly consider the HBSP method on the desired number of bins without using an oversampling. We give now an example to illustrate that our bounding histograms and the approximative histogram obtained by the HBSP method are different.

EXAMPLE 2. For the MAWI traffic histogram defined on 80511 bins, the HBSP approximation with $n = 10$ gives the green histogram in Figure 6. The red histogram corresponds to the lower bound while the black histogram is the upper bound. These bounding histograms are computed for the reward function $\boldsymbol{r}$: $\boldsymbol{r}(i) = a_i, \ \forall a_i \in \mathbf{A}$ and by using the same number of bins ($n = 10$). The expected reward of the original histogram is $R[\mathcal{A}] = 4.3756 \times 10^6$ bits, the expected reward of the HBSP histogram is $R[\mathcal{A}] = 4.3757 \times 10^6$ bits and the expected reward of our bounding histograms are $R[\mathcal{A}] = 4.1644 \times 10^6$ bits for the lower bound and $R[\mathcal{A}] = 4.5843 \times 10^6$ bits, for the upper bound. At first sight, HBSP method seems more accurate. We will see in the next section it is not true.

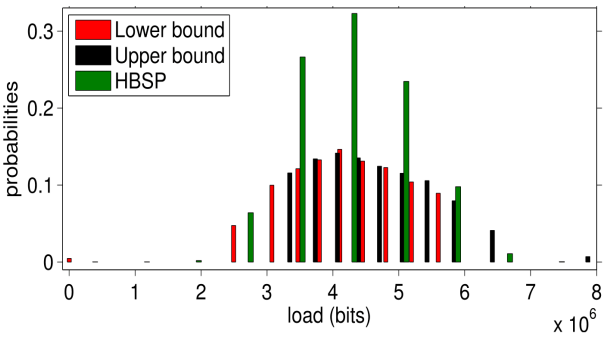


FIGURE 6: Reduced histograms with 10 bins for the MAWI traffic (some very small probability values are not readable for the HBSP method).

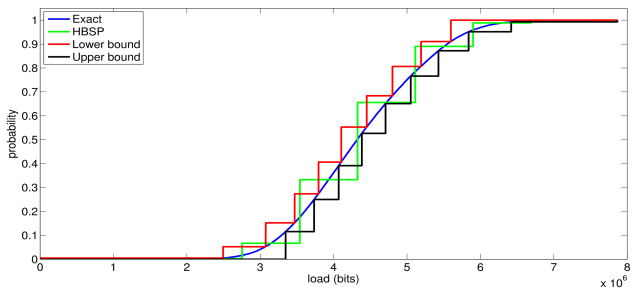


FIGURE 7: Cumulative probability distributions (cdf) for the MAWI traffic.

The cumulative probability distributions of these histograms are presented in Figure 7. We can also see

from this figure that the HBSP method does not provide bounds while the bounds supply well the coverage of the exact distribution.

4.4. Numerical results

Note that all the computations in this paper are performed with MATLAB software on a simple laptop.

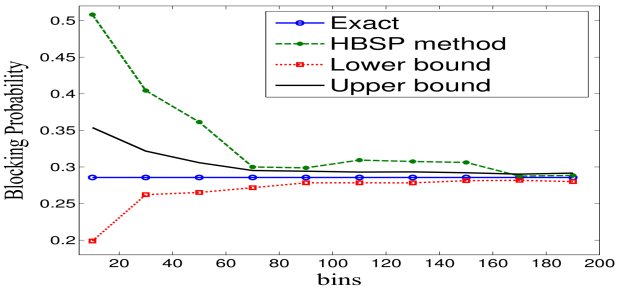
This section presents some numerical results to validate our approach and show its relevance in determining the performance measures of a single queue. The experiments illustrated here have been performed with real traffic Internet traces under the stationary arrival assumption. We compute some performance measures such as blocking probabilities, the expected length of buffers, etc. For all the experiments, the reward function is defined as $\boldsymbol{r}(i) = a_i$, $\forall\, a_i \in \boldsymbol{A}$.

Note that the histogram-based traffic model can be a powerful and compact description, if the sufficient accuracy can be reached with a small number of bins. Thus, for a simple queue with real traffic trace, the question that would be interesting to know is: how many bins are needed to obtain a good accuracy? We will try to answer this question by studying the following aspects: the influence of the size of the support (after reduction) on the accuracy of the results, the relationship between the buffer size and some performance measures for a given number of bins and then the interaction between these three factors, namely: the number of bins, the buffer size and the accuracy of the results.

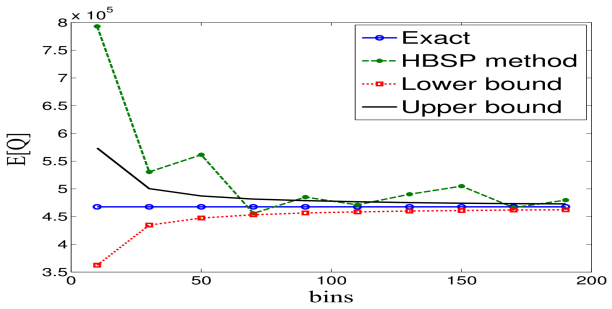
Accuracy versus the size of bounding histograms We consider a single queue under the MAWI traffic traces (Figure 2), and analyze the influence of the number of bins on the accuracy of the results. We set the mean transmission rate to 110 Mb/s and the buffer size to $B = 1$ Mb. In Figure 8, we compute the blocking probability and the mean buffer length for different number of bins (varying from 10 to 200). In each figure, we give the results computed by different methods: 1) Our method with original histogram (without size reduction). These results are noted by *Exact*. 2) HBSP method (histogram construction and reduction as given in [3]. 3) The proposed lower and upper bounds with reduced size histograms.

We observe that when the size of the support (the number of bins) increases, the results obtained by different methods become more accurate. However, for small size support, the results of the HBSP method are far worse than ours. In addition, we distinguish an oscillatory behavior for the results of HBSP which are sometimes lower and often higher than exact values. Thus the HBSP method does not provide bounding but approximate results. The upper and lower bounds become very close to the exact ones for the number of bins greater 30.

In Figure 9, we illustrate the cumulative probability distribution of the buffer length by taking the number of bins equal to 20 or 100. We see that the stochastic upper bound (resp. stochastic lower bound) is always under (resp. above) the exact curve while the HBSP distribution crosses the exact curve.

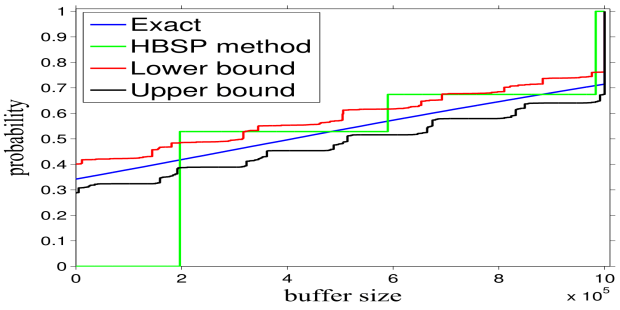


(a) Blocking probability

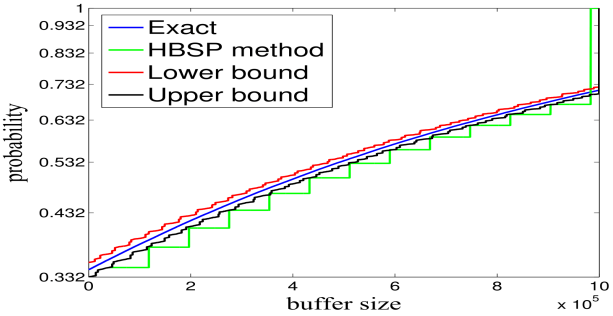


(b) Mean buffer length

FIGURE 8: Accuracy versus the number of bins: QoS parameters using the MAWI traffic



(a) bins=20

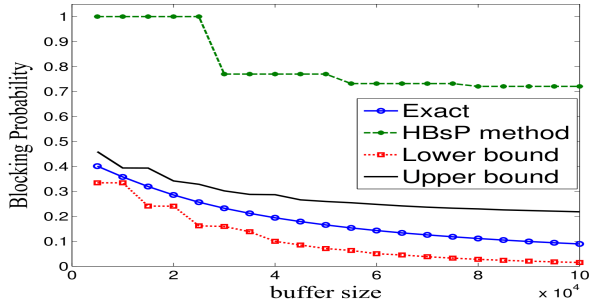


(b) bins=100

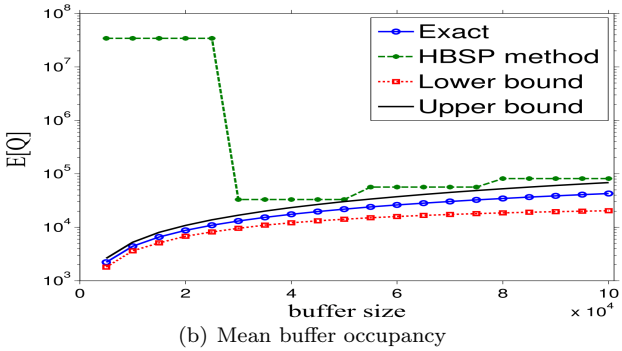
FIGURE 9: Cumulative probability distribution of the buffer occupancy under the MAWI traffic

The HBSP method does not provide a good approximation for small number of bins (bins=20). When the number of bins is equal to 100, all methods provide better results and our bounds are the most accurate ones. When the number of bins is 100, the exact computations are obtained in 1897 seconds (s), the results of the HBSP method in 0.007 s while the lower and upper bounds are respectively computed in 0.35 s and 0.33 s. So, the HBSP method and bounds are computed in less than one second against more than 30 minutes for the exact method. We remark also that the HBSP method is the fastest, but the time to derive our bounds is very short and the method is largely faster than the exact computation with very relevant precisions.

Performance measures versus the sizes of the buffer and bounding histograms. For this aspect, the performed experiment is based on the CAIDA OC-48 traffic trace [15] collected in both directions of an OC48 link at the AMES Internet Exchange (AIX) on the 24^{th} of April, 2003. The collected trace is one hour long with an average rate of 92 Mb/s. For our experiment, we take 5-minutes of packet header trace. Using a sampling period of T = 10 ms (100 samples per second), the resulting traffic trace has 30,000 frames and mean value, $E[\mathcal{A}] = 1.2885 \times 10^5\, bits$. We consider the relationship between the buffer size and the blocking probability (resp. mean buffer length) for bounding histograms, HBSP model and the exact result.



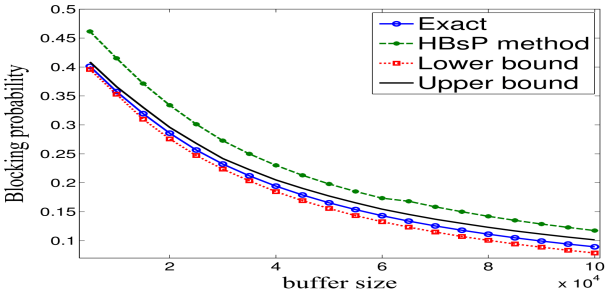
(a) Blocking Probability



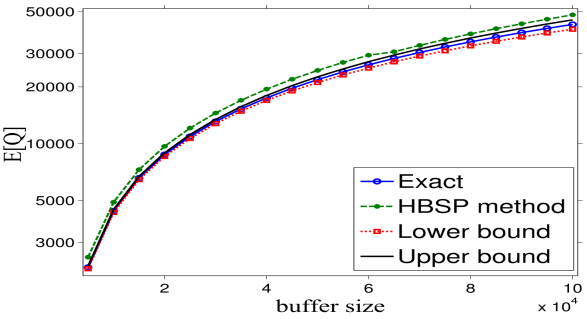
(b) Mean buffer occupancy

FIGURE 10: QoS parameters using CAIDA OC-48 traffic trace, bins=10

The performance indices are calculated by varying the buffer size from $5 \times 10^3\, bits$ to $10^5\, bits$. We note that the size of the original histogram is 24930 states. The support of the bounding histograms and the HBSP model is equal to 10 for Figure 10 while it is equal to 100 for Figure 11.



(a) Blocking probability



(b) Mean buffer occupancy

FIGURE 11: QoS parameters using CAIDA OC-48 traffic trace, bins=100

This experiment yields to the same conclusions as above. In Figure 10, the HBSP method does not converge for small buffer capacities (that is why the points were not shown). For the other points, we see that the quality of our bounds are better than the results obtained by the HBSP method. Thus, to finish our assessments, we propose to vary respectively the buffer capacity and the support size of the input traffic distribution and observe the blocking probabilities computed by the different methods.

We depict some obtained results in Table 1. We recall here that the parameters considered in these experiments are taken from [5, 3] in order to compare results.

From Table 1, we see clearly that our bounds provide a good coverage of the exact solution and it becomes more accurate with the increase of the number of bins. Regarding the HBSP method, we observe that for small number of bins (bins=10), the HBSP method does not converge when buffer size is approximately less than $\times 10^4$ and gives less accurate results elsewhere. However, our bounds provide fairly good coverages on the exact results. We notice also that when the number of bins increases, the considered methods provide closer

Buffer size	Exact	bins	Lower Bound	Upper Bound	HBSP
5×10^3	0.4011	10	0.3344	0.4591	/
		100	0.3957	0.4087	0.4614
		200	0.3958	0.4050	0.4259
10^4	0.3492	10	0.2667	0.3936	/
		100	0.3401	0.3585	0.3922
		200	0.3447	0.3529	0.3690
5×10^4	0.1631	10	0.0696	0.2571	0.7698
		100	0.1536	0.1742	0.1980
		200	0.1583	0.1683	0.1740
10^5	0.0903	10	0.0166	0.2188	0.7204
		100	0.0796	0.1023	0.1201
		200	0.0848	0.0961	0.9991

TABLE 1: Blocking probabilities versus buffer size and number of bins.

results to the exact ones.

5. PIECEWISE STATIONARY ARRIVAL PROCESS

We now assume that the traffic is stationary for short time periods and evolves when we change the time period. For the sake of simplicity we assume that these time periods have the same length. More precisely, the time interval for the trace analysis is divided into k consecutive periods of duration T. During each period, the traffic is stationary: between time instants iT and $(i+1)T$ excluded, the traffic is distributed following distribution $\mathcal{A}^{(i)}$. We still assume that the arrivals are independent. As mentioned in Section 2, the system is modeled by an inhomogeneous discrete time Markov chain.

5.1. Numerical algorithm

We want to emphasize that the numerical method based on the convolution of distributions for stochastically monotone models is still efficient to study the system as soon as T is larger than the expectation of the coupling time. Let us now explain how we modify the algorithm to cope with this new assumption on the traffic.

First note that in the time interval $[iT, (i+1)T)$, the traffic is stationary. Thus, we have to analyze the transient distribution of a homogenous Markov chain during each time interval. The whole analysis consists in computing the distribution at time iT by a numerical computation for period $[(i-1)T, iT)$ to find the initial distribution for interval $[iT, (i+1)T)$. By taking into account the monotonicity of the system and the coupling, unnecessary computations are avoided as depicted in Fig. 12. For the first time period, we proceed as the former algorithm for stationary traffic. We begin to compute the upper and lower sample-paths for the distribution. Two cases may occur: one may observe the convergence due to the coupling before T (see Fig. 12) or T is reached before the occurrence of the coupling (see Fig. 13).

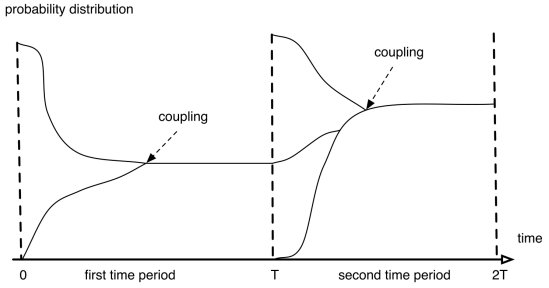


FIGURE 12: Efficient computation with coupling

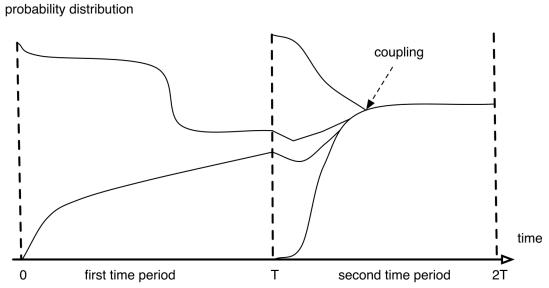


FIGURE 13: Computation without coupling in the first period.

If the first case, we jump to time instant T without any computation of the distribution as soon as we have detected the coupling. Indeed, as the two distributions have coupled, we have reached the steady-state distribution thus it is not necessary to continue the numerical process. Of course, as the traffic is not stationary after the end of the time period, the obtained distribution is not really the steady-state distribution (but we still call it steady-state to explain that the distribution does not change until the end of the time period).

1. Iterate on the period number i from 0 to k-1
   (a) Initialize t=iT
   (b) Initialize $g_{min} = \delta_0$
   (c) Initialize $g_{max} = \delta_B$
   (d) If $(i = 0)$ initialise $g_{cur} = \delta_0$
   (e) Iterate
      i. increase time instant t
      ii. compute new g_{min} with arrival distribution $\mathcal{A}^{(i)}$
      iii. compute new g_{max} with arrival distribution $\mathcal{A}^{(i)}$
      iv. compute new g_{cur} with arrival distribution $\mathcal{A}^{(i)}$

(f) Until ($g_{min} = g_{max}$) or ($t = iT + T - 1$)
(g) If coupling (i.e. $g_{min} = g_{max}$), jump to $t = iT + T - 1$ without any new computation of the distributions g_{min}, g_{max}, et g_{cur} which are all equal.

Thus, the coupling time has a strong impact on the efficiency of this numerical method. We illustrate an example with real traffic trace, we consider the MAWI trace [10] which corresponds to an IP traffic on transpacific line with link capacities of 128 Kbps, carried between the 6th of march 2007 at 18 : 00 and the 7th of march 2007 at 4 : 24 : 27. For a sampling period of 40 ms, we obtain the trace shown in Figure 14 with 922873 frames and 4579 different states.

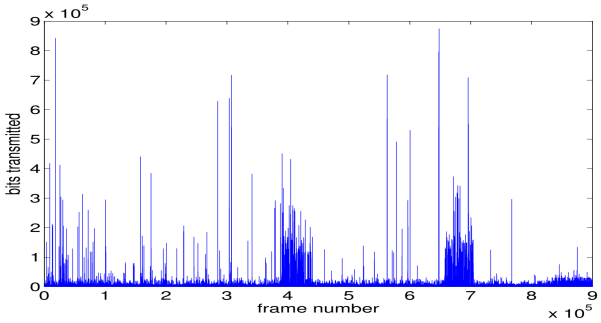


FIGURE 14: MAWI traffic trace (more than 10 hours).

The following trace is considered with a period duration of $10mn$ for a queue with size 2×10^6 bits and a service capacity of 300 kbps. Knowing that the sampling time is $40ms$, a time period, T contains 15000 time instants. During that period the traffic is supposed to be stationary.

We note that the average number of iterations before the convergence is 450, the median is 265 and the third quartile is 435. The distribution of the number of iterations before convergence has a low variability, its coefficient of variation is 0.74. In most of the cases the number of iterations is smaller than 400 and this is quite small compared to the 15000 iterations we must compute if we do not detect the steady-state. More than half of the experiments with our algorithm consists in 300 steps of computation instead of 15 000 for a naive algorithm. Clearly, our approach is much more efficient than the histogram-based simulation which requires the computation of the whole path with 15 000 time steps.

5.2. Numerical results

We consider a queue with a service capacity equal to $S = 300$ Kbps. We assume that the arrivals are extracted from the MAWI-10h trace. We assume that the traffic is stationary within period of 10 mn. The precision threshold for the numerical algorithm is $\epsilon = 10^{-9}$. We note that for our bounding results, the input traffic is reduced to 100 bins for each period. In the following tables, we report the obtained upper and lower bounds on the sample-path. As shown in Table 2 and 3, the results are quite accurate for the exact results and for the bounds as well.

Buffer size	Exact	Lower Bound	Upper Bound
2×10^5	0.0333847	0.0333695	0.0334154
5×10^5	0.0059227	0.0059153	0.0059558
10^6	0.0016837	0.0016452	0.0017109
2×10^6	0.0005061	0.0004722	0.0005254
3×10^6	0.0001911	0.0001692	0.0002031

TABLE 2: Blocking probabilities versus buffer size.

Buffer size	Exact	Lower Bound	Upper Bound
2×10^5	3559.58	3506.79	3574.81
5×10^5	7617.23	7491.65	7656.11
10^6	11231.2	11006.1	11312.2
2×10^6	15853.5	15395.2	16043.4
3×10^6	18170.1	17989.9	19027.2

TABLE 3: Expected buffer lengths versus buffer size.

Finally we report in Table 4 the computation times. The last column contains the results for histogram-based analysis where we use the original traffic with an algorithm which does not check the steady-state for the distribution of the queue length during the periods where the traffic does not change. In the second column, we give the computation times for the approach with the original traffic (exact histogram) and the steady-state detection as introduced in the first part of this section. Clearly the detection of steady-state improves considerably the performance of the algorithm. Finally it is still possible to speed up the method using bounds on the traffic as shown by the results given in the third and the fourth column.

Buffer Size	Exact	Lower Bound	Upper Bound	Hist.-Based Analysis
2×10^5	17.0	3.6	4.2	859
5×10^5	49.2	16.9	15.5	3546
10^6	172	61.4	63.0	5745
2×10^6	1315	434	469	22233
3×10^6	3239	953	994	57746

TABLE 4: Computation times in second.

6. CONCLUSION

We show how to derive stochastic bounds on the queue length and the departure traffic for a queue under input traffic histograms constructed from real traffic traces. We state that the comparison method with the $\leq_{st}$ ordering does not require that the traffic is stationary. We also present a numerical technique suitable to various assumptions about the stationary of the input process and these approaches are much faster than the trace based

simulation. We define bounding histograms of smaller sizes to manage the computationally complexity. Thus we have a trade-off between the accuracy of results and the computational complexity. The bounds are much more relevant than the approximations for network dimensioning and QoS evaluation. Finally it is worthy to remark that our approach can be used to study the performance of any system which is associated with a stochastically monotone model and such that some measurements are available.

ACKNOWLEDGEMENTS

This work was supported by grant ANR MARMOTE (ANR-12-MONU-0019).

APPENDIX A. STOCHASTIC COMPARISON

We refer to [11] for theoretical issues of the stochastic comparison method. We consider state space $\mathcal{G} = \{1, 2, \ldots, n\}$ endowed with a total order denoted as $\leq$. Let X and Y be two discrete random variables taking values on $\mathcal{G}$, with cumulative probability distributions F_X and F_Y, and probability mass functions $\boldsymbol{d}_X$ and $\boldsymbol{d}_Y$ ($\boldsymbol{d}_X(i) = \text{Prob}(X = i)$, and $\boldsymbol{d}_Y(i) = \text{Prob}(Y = i)$, for $i = 1, 2, \ldots, n$).

Definition A.1. *The three following definitions are known to be equivalent:*

- **generic definition:** $X \leq_{st} Y \iff \mathbb{E}f(X) \leq \mathbb{E}f(Y)$, *for all increasing (non decreasing) functions* $f : \mathcal{G} \to \mathbb{R}^+$ *whenever expectations exist.*
- **cumulative probability distributions:**

$$X \leq_{st} Y \Leftrightarrow F_X(a) \geq F_Y(a),\ \forall a \in \mathcal{G}.$$

- **probability mass functions:**

$$X \leq_{st} Y \Leftrightarrow \forall i,\ 1 \leq i \leq n,\ \sum_{k=i}^{n} \mathbf{d}_X(k) \leq \sum_{k=i}^{n} \mathbf{d}_Y(k) \tag{A.1}$$

Notice that we use interchangeably $X \leq_{st} Y$ *and* $\mathbf{d}_X \leq_{st} \mathbf{d}_Y$.

Property 2. If $X \leq_{st} Y$, then for any increasing function f,

$$f(X) \leq_{st} f(Y)$$

Example 3. We consider two discrete random variables with $\boldsymbol{d}_X = [0.1,\ 0.2,\ 0.1,\ 0.2,\ 0.05,\ 0.1,\ 0.25]$, and $\boldsymbol{d}_Y = [0.25,\ 0.05,\ 0.1,\ 0.15,\ 0.15,\ 0.3]$ defined respectively on support $\{1, \ldots, 7\}$ and $\{2, \ldots, 7\}$. The set $\mathcal{G}$ is the union of support of the two distributions $\boldsymbol{d}_Y$ and $\boldsymbol{d}_X$ with null probabilities if an element does not belong to one of them. We can easily verify that $\boldsymbol{d}_X \leq_{st} \boldsymbol{d}_Y$: the probability mass of $\boldsymbol{d}_Y$ is concentrated to higher states such as the probability cumulative distribution of $\boldsymbol{d}_Y$ is always below the cumulative distribution of $\boldsymbol{d}_X$ (Figure. A.1).

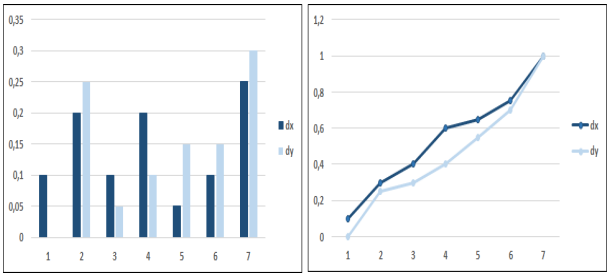


FIGURE A.1: $\boldsymbol{d}_X \leq_{st} \boldsymbol{d}_Y$: Probability mass functions (left) and cumulative distribution functions (right).

We apply the following definition to compare Markov chains.

Definition A.2. *Let* $\{X(n), n \geq 0\}$ *(resp.* $\{Y(n), n \geq 0\}$*) be a DTMC. We say* $\{X(n), n \geq 0\} \leq_{st} \{Y(n), n \geq 0\}$, *if* $X(n) \leq_{st} Y(n),\ \forall n \geq 0$.

The following definition present the stochastic monotonicity of a DTMC.

Definition A.3 (Stochastic monotonicity). *Let* $\{X(n), n \geq 0\}$ *be a DTMC, we say that* $\{X(n), n \geq 0\}$ *is stochastic monotone if*

$$X(0) \leq_{st} X(1) \ \Rightarrow\ X(n) \leq_{st} X(n+1),\ \textit{for all}\ n > 0.$$

REFERENCES

[1] Stewart, W.J. (1995) Introduction to the numerical Solution of Markov Chains. *Princeton University Press*, New Jersey.

[2] Gupta, V. and Harchol-Balter, M. and Dai, J. G. and Zwart, B. (2010) On the inapproximability of $M/G/K$: why two moments of job size distribution are not enough. *Queueing Syst.*, **64**, 5–48.

[3] Hernández-Orallo, E. and Vila-Carbó, J. (2007) Network Performance Analysis based on Histogram Workload Models. *Proceedings of MASCOTS 2007, Istanbul, Turkey, 24-26 October*, pp. 209–216. IEEE Computer Society, Los Alamitos, CA, USA.

[4] Hernández-Orallo, E. and Vila-Carbó, J. (2009) Web server performance analysis using histogram workload models. *Computer Networks*, **53**, 2727–2739.

[5] Hernández-Orallo, E. and Vila-Carbó, J. (2010) Network queue and loss analysis using histogram-based traffic models. *Computer Communications*, **33**, 190–201.

[6] Skelly, P. and Schwartz, M. and Dixit, S. S. (1993) A histogram-based model for video traffic behavior in an ATM multiplexer. *IEEE/ACM Trans. Netw.*, **1**, 446–459.

[7] Tancrez, J.-S. and Semal, P. and Chevalier, P. (2009) Histogram based bounds and approximations for production lines. *European Journal of Operational Research*, **197**, 1133–1141.

[8] Aït-Salaht, F. and Cohen, J. and Castel Taleb, H. and Fourneau, J.-M. and Pekergin, N. (2012) Accuracy vs. Complexity: the stochastic bound approach.

Proceedings of WODES 2012, Guadalajara, Mexico, 3-5 October, pp. 343–348. International Federation of Automatic Control.

[9] Aït-Salaht, F. and Castel-Taleb, H. and Fourneau, J.-M. and Pekergin, N. (2013) Stochastic Bounds and Histograms for Network Performance Analysis. *Proceedings of EPEW 2013, Venice, Italy, 16-17 September*, pp. 13–27. **8168** of LNCS, Springer Berlin Heidelberg.

[10] Sony, K. C. and Cho, K. (2000) Traffic Data Repository at the WIDE Project. *Proceedings of the Annual Conference on USENIX Annual Technical Conference, San Diego, California, 18-23 June*, pp. 51–51. USENIX Association, Berkeley, CA, USA.

[11] Muller, A. and Stoyan, D. (2002) Comparison Methods for Stochastic Models and Risks. *Wiley*, New York.

[12] Ciardo, G. and Blakemore, A. and Chimento, P. F. and Muppala, J. K. and Trivedi, K. S. (1993) Linear algebra, Markov chains, and queueing models. Springer-Verlag, New York, Berlin, Heidelberg.

[13] Sericola, B. (1999) Availability Analysis of Repairable Computer Systems and Stationarity Detection. *IEEE Trans. Computers*, **48**, 1166–1172.

[14] Robertson, J. P. (1992) The computation of aggregate loss distributions. *In Proceedings of the Casualty Actuarial Society*, **79**, 57–133.

[15] CAIDA (2003) Traces of OC48 link at AMES Internet Exchange (AIX) (April 24, 2003), accessed via DatCat - Internet Data Measurement catalog. *http://imdc.datacat.org*.