Stochastic Bounds and Histograms for Network Performance Analysis

F. Aït-Salaht[1], H. Castel-Taleb[2], J.M. Fourneau[1], and N. Pekergin[3]

[1] PRiSM, Univ. Versailles St Quentin, UMR CNRS 8144, Versailles France
safa,jmf@prism.uvsq.fr
[2] SAMOVAR, UMR 5157, Télécom Sud Paris, Evry, France
hind.Castel@it-sudparis.eu
[3] LACL, Univ. Paris Est-Créteil, France
nihal.pekergin@u-pec.fr

Abstract. Exact analysis of queueing networks under real traffic histograms becomes quickly intractable due to the state explosion. In this paper, we propose to apply the stochastic comparison method to derive performance measure bounds under histogram-based traffics. We apply an algorithm based on dynamic programming to derive bounding traffic histograms on reduced state spaces. We indeed obtain easier bounding stochastic processes providing stochastic upper and lower bounds on buffer occupancy histograms (queue length distributions) for finite queue models. We evaluate the proposed method under real traffic traces, and we compare the results with those obtained by an approximative method. Numerical results illustrate that the proposed method provides more accurate results with a tradeoff between computation time and accuracy. Moreover, the derived performance bounds are very relevant in network dimensioning.

Keywords: Network QoS, Histogram-based traffic models, Stochastic Comparison.

1 Introduction

Queueing-based models are very efficient modeling and evaluation tools for a variety of practical situations in telecommunication and computer network systems. The stochastic behavior prediction of queues gives a theoretical insight into the dynamics of these shared resources and how they can be designed to provide better utilization. The probability theory in queueing analysis plays a central role as it provides mathematical equations for performance measure computations such as queue length, response time, and server utilization. Most of the queueing theory is based on exponential assumption. However, this assumption can be applied only for certain applications as in telephone networks. In the Internet, several traffic traces are available, and are used to be approximated by a theoretical probability distribution (for example, the phase distribution). Unfortunately, some problems arise: the accuracy of the model compared to the traffic, and the difficulty to exploit the model when the number of parameters is high. So most of the time, we must limit the number of parameters in the detriment of precision. Moreover, the exact analysis of the queueing network with the real traffic traces is in general impossible, as their sizes are too large to be used directly.

There has been a several amount of works on the Histogram-based approach for performance models. In the area of network calculus, the histogram model was introduced by Skelly et al [11] to model the video sources, to predict buffer occupancy distributions and cell loss rates for multiplexed streams. It was also applied by Kweon and Shin [8] to propose an implementation of statistical real-time communication in ATM networks using a modified version of the Traffic Controlled Rate-Monotonic Priority Scheduling (TCRM). These works used an analysis method based on a M/D/1/N queueing system. More recently, Hernández and al.[5–7] have proposed a new performance analysis to obtain buffer occupancy histograms. This new stochastic process called HBSP (Histogram Based Stochastic Process) works directly with histograms using a set of specific operators. The model is based on a basic histogram model (HD) as an input traffic which is supplied through finite capacity buffers with deterministic (D) service time distribution under First Come First Served (FCFS) policy. Considering a single node, the analysis method solves the HD/D/1/K queueing system, by reducing the state space of traffic trace into n subintervals (classes or bins) in order to avoid working with huge state spaces.

Another approach based on reducing the initial histogram in n subintervals has been presented by Tancrez and al.[14] in a slightly different context. The problem consists in building an upper (lower) bounding discrete distribution of a continuous distribution which models the service duration in a production line. They divide the support into n equal subintervals. Each of these subintervals of the continuous distribution is associated with one single point of the discrete one. This point is the upper limit (lower limit) of the interval and the probability mass of the sub-interval is associated to that point. As the production lines considered can be modeled by a decision free stochastic Petri-net, it is known since the seminal work of Baccelli et al. [3] that bounding the distribution of service times in the queues provides a bound on the end to end delay.

In this paper, we apply the stochastic bounding method for network performance analysis under histogram-based traffic. The goal is to generate bounding histograms with smaller sizes in order to make possible the analysis of the queueing network. We use the strong stochastic ordering (denoted $\leq_{st}$) [9]. We propose to use algorithmic techniques developed in [2] to obtain optimal lower and upper stochastic bounds of the buffer occupancy histogram. These algorithms allow to control the size of the model and compute the most accurate bound with respect to a given reward function. The bounding histograms are then used in the state evolution equations to derive bounds for performance measures both for a single queue and a tandem queueing network. To show the relevance of our work, we analyze systems with real traffic traces. We compare our bounds with the results under exact traffic traces and those obtained from the HBSP approximation method. The proposed method provides the most accurate results for blocking probability and mean buffer occupancy. Another important point is that HBSP only provides approximative results which are neither conservative nor optimistic. Our bounding approach gives, at the same time, upper and lower performance measures which could be used to check QoS constraints for network dimensioning.

This paper is organized as follows: in Section 2, we first describe the histogram traffic models, and the state evolution equations of the queuing model under study. Then, we explain the Histogram Buffer Stochastic Process (HBSP) method proposed by Hernandez and al. In Section 3, we introduce our approach based on the stochastic bounds to derive performance measure bounds. Finally in Section 4, we give numerical results based on real traffic measurements in order to study the accuracy of the bounds, compared to the exact results and those obtained by HBSP algorithm. These results are obtained for a single node analysis and also for a tandem queueing network.

2 Queueing model description

2.1 Histogram traffic model

Here a histogram describe a discrete distribution and its graphical representation. Figure 1 shows a plot of MAWI traffic trace [12] corresponding to a 1-hour trace of IP traffic of a 150 Mb/s 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 Mb/s. Using a sampling period of T = 40 ms (25 samples per second), the resulting traffic trace has 90,000 frames (periods) and an average rate of 4.37 Mb per frame, the corresponding histogram is given in Figure 2.

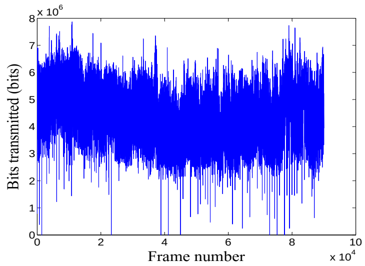


Fig. 1. MAWI traffic trace

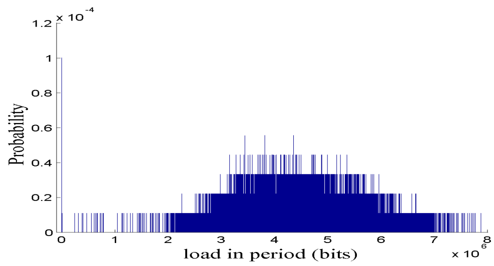


Fig. 2. MAWI arrival load histogram

The arrival workload is characterized with the number of transmission units produced by the corresponding traffic source during a pre-established time period $T = 40\,ms$. Let $A(k)$ be a discrete random variable representing the amount of traffic entering to the system during the k^{th} sampling interval (slot). We assume that the traffic is stationary and independently and identically distributed (iid). So, all random variables $A(k)$ follow the same distribution $\mathcal{A}$ characterized by a couple $(\mathbf{A}, p(\mathbf{A}))$, where $\mathbf{A}$ is the support and $p(\mathbf{A})$ is the corresponding probabilities.

2.2 State evolution equations

We denote by $Q(k)$ and $D(k)$ respectively random variables corresponding to buffer length and the output traffic (departure) during the k^{th} slot (Figure 3).

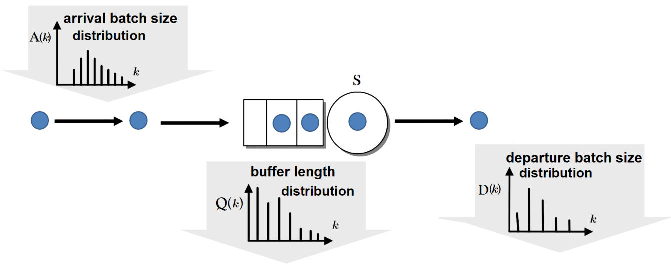


Fig. 3. Input and output parameters of a queueing model

Let B be the buffer size and S be the transmission (service) rate. We assume that the following sequence of events during a slot: acceptation of arrivals and then service. The queue or buffer length $Q(k)$ can be expressed with the following recursive formula:

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

where operator $(X)^+ = \texttt{max}(X, 0)$. As we assume a Tail Drop policy, the departure distribution is defined as follows:

$$D(k) = \texttt{min}(S, Q(k-1)) + A(k)), \quad k \in \mathbb{N}. \tag{2}$$

The transmission channel utilization is defined as $\rho = \frac{E[A]}{S}$, where $E[A]$ is the average traffic. Equation 1 defines a Markov chain in discrete time (DTMC) if the arrivals $(A(k))$ are stationary and iid. As this chain is finite, it suffices to verify that the arrival process makes the chain irreducible and aperiodic thus ergodic. We give below some sufficient conditions to ensure both properties.

Proposition 1. *If the following conditions are satisfied, then the DTMC is ergodic:*

1. *there exists $i < S$ in the support of A such that $p(i) > 0$,*
2. *there exists $j > S$ in the support of A such that $p(j) > 0$,*
3. *$j = S + 1$ or $i = S - 1$.*

Proof. The first property implies that starting from state 0 we go back with a probability greater than or equal to $p(i)$. Indeed, if it arrives i customers, the buffer length before the services is i, as $i < S$, we return to state 0 after the end of service. So state 0 is aperiodic. In addition, the first property implies that the buffer length can be reduced to 0 by a sequence of transitions from one arrival of a batch of size $i < S$. So 0 is reachable from any states of the chain. Condition 2 implies that we can reach state B from state 0. Finally, the last condition implies that we can reach all states from 0 or B by jumps with amplitude 1 that are possible under assumptions 1 and 2.

We suppose in the following that the Markov chain is ergodic.

Let X (resp. Y) be a discrete random variable taking values in a set $\mathcal{G}_X$ (resp. $\mathcal{G}_Y$) of size $l_X > 1$ (resp. $l_Y > 1$).

Proposition 2. *The computation of the convolution of the distributions of two independent random variables generates a distribution with at most* $l_X \times l_Y$ *states. This computation requires* $O(l_X \times l_Y)$ *operation (+) using a naive approach and* $O((l_X + l_Y)log(l_X + l_Y))$ *for a Fast Fourier Transform (FFT) [10].*

The computation complexity depends on the size of the distributions and thus on the number of classes considered.

2.3 Histogram reduction: HBSP method

The Histogram Buffer Stochastic Process (HBSP) model is proposed by Hernández and al. [5–7]. Since working with a huge distribution can be cumbersome, the method suggests to reduce this size using n classes or bins. Consequently, if we have a range of $I = [0,\, N_{max}]$, then the interval size will be $l_{\mathbf{A}} = N_{max}/n$. Using these intervals we define a binned process $\{A(t)\}$ that 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: $a = l_{\mathbf{A}} \,.\, i + l_{\mathbf{A}}/2$, $a \in I$.

The traffic is assumed to be stationary, $\mathcal{A} = A(t)$, $\forall t$, thus the time dependence of $A(t)$ is suppressed and replaced by a discrete random variable $\mathcal{A}$ which is defined by a couple of attributes $(\mathbf{A},\, p(\mathbf{A}))$. Each attribute is a vector of size n, first vector is interval midpoints while the second gives the corresponding probabilities.

$$\mathcal{A} = (\mathbf{A},\, p(\mathbf{A})) \quad \begin{cases} \mathbf{A} = \{a_i : i = 0 \ldots n-1\}, \\ p(\mathbf{A}) = [p_{\mathbf{A}}(i) : i = 0 \ldots n-1]. \end{cases}$$

The stochastic process of the evolution of HBSP model is based on the following recurrence relation:

$$Q(k) = \Phi_{\hat{S}}^{\hat{b}}(Q(k-1) \otimes \mathcal{A}). \tag{3}$$

where, $\hat{S} = class_{\mathbf{A}}(S)$ (resp. $\hat{b} = class_A(B)$), $\otimes$ is the convolution operator of distributions. $Q(k)$ denotes here the corresponding distribution and operator Φ limits buffer lengths so that they can not become negative and cannot overflow the corresponding class of buffer length. This operator is defined as follows:

$$\Phi_a^b(x) = \begin{cases} 0\ , & \text{for } x < a, \\ x\ , & \text{for } a \le x \le b+a, \\ b\ , & \text{for } x \ge b+a. \end{cases} \quad (4)$$

Example 1. For the MAWI traffic trace with $n = 10$, the HBSP traffic is defined by $\mathcal{A} = (\mathbf{A}, p(\mathbf{A}))$ with $\mathbf{A} = \{0.3933, 1.1799, 1.9666, 2.7532, 3.5398, 4.3265, 5.1131, 5.8997, 6.6864, 7.4730\}\ Mb$ and $p(\mathbf{A}) = [0.0003, 0.0002, 0.0021, 0.0641, 0.2663, 0.3228, 0.2345, 0.0980, 0.0110, 0.0005]$ (Figure 4).

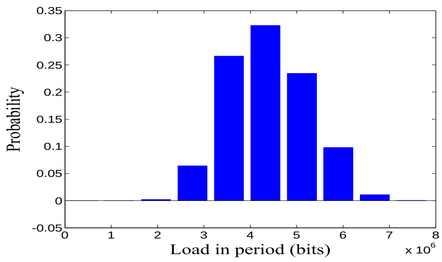


Fig. 4. Arrival workload histogram of MAWI traffic using 10 classes.

3 Bounding approach

We first present briefly the stochastic comparison method and we then present the proposed bounding algorithm for the reduction of the number of classes for a histogram. The application of this bounding approach for the network performance analysis will be given in the next section.

3.1 Stochastic comparison

We refer to Stoyan's book ([9]) 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 $\le$. 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 $\boldsymbol{d}2$ and $\boldsymbol{d}1$ ($\boldsymbol{d}2(i) = Prob(X = i)$, and $\boldsymbol{d}1(i) = Prob(Y = i)$, for $i = 1, 2, \ldots, n$). We give different manners to define the strong stochastic ordering $\le_{st}$ for this case:

Definition 1. – **generic definition:** $X \le_{st} Y \Longleftrightarrow \mathbb{E}f(X) \le \mathbb{E}f(Y)$, *for all non decreasing functions* $f : \mathcal{G} \to \mathbb{R}^+$ *whenever expectations exist.*

– **cumulative probability distributions:**

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

– **probability mass functions**

$$X \leq_{st} Y \Leftrightarrow \forall i,\ 1 \leq i \leq n,\ \sum_{k=i}^{n} \mathbf{d}2(k) \leq \sum_{k=i}^{n} \mathbf{d}1(k) \quad (5)$$

Notice that we use interchangeably $X \leq_{st} Y$ *and* $\mathbf{d}2 \leq_{st} \mathbf{d}1$.

Example 2. We consider $\mathcal{G} = \{1, 2, \ldots, 7\}$, and two discrete random variables with $\boldsymbol{d}2 = [0.1, 0.2, 0.1, 0.2, 0.05, 0.1, 0.25]$, and $\boldsymbol{d}1 = [0, 0.25, 0.05, 0.1, 0.15, 0.15, 0.3]$. We can easily verify that $\boldsymbol{d}2 \leq_{st} \boldsymbol{d}1$: the probability mass of $\boldsymbol{d}1$ is concentrated to higher states such as the probability cumulative distribution of $\boldsymbol{d}1$ is always below the cumulative distribution of $\boldsymbol{d}2$ (Figure. 5).

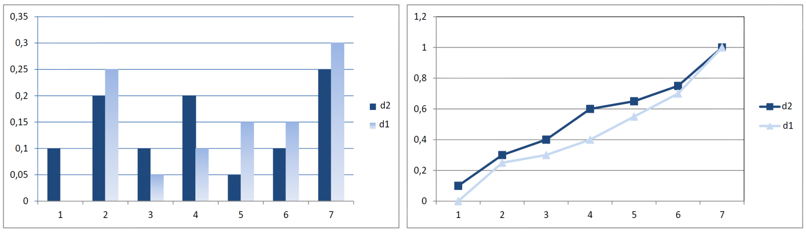


Fig. 5. $\boldsymbol{d}2 \leq_{st} \boldsymbol{d}1$: Their pmf (left) and their cumulative distribution functions (right).

3.2 Bounding histogram reduction

In order to reduce the computation complexity of evolution equations, we propose to apply the bounding approach to diminish the number of classes. The main advantage of this approach is the ability of computing bounds rather than approximations. Unlike approximation, the bounds allow us to check if QoS are satisfied or not. For a given distribution $\boldsymbol{d}$, defined as a histogram with N classes, we build two bounding distribution $\boldsymbol{d}1$ and $\boldsymbol{d}2$ which are defined as histograms with $n < N$ classes. Moreover, $\boldsymbol{d}1$ and $\boldsymbol{d}2$ are constructed to be the closest with respect to a given reward function. Two algorithms are given in [2] to construct such bounds. More 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 states 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 states that are stochastically upper than $\boldsymbol{d}$.

$\boldsymbol{d}1$ and $\boldsymbol{d}2$ will be denoted as the optimal bounding distributions on n states according to reward $\boldsymbol{r}$. We now present the bounding algorithm that will be used in this paper.

Optimal Algorithm based on Dynamic Programming We will transform our problem dealing with a discrete distribution into a graph theory problem. First, we consider the weighted graph $G = (V, E)$ such that:

- V is the set of vertices such that $V = \mathcal{H} \cup \{EndState\}$ where *EndState* is a new state larger than all the states in $\mathcal{H}$.
- E is the set of arcs such that $(u, v) \in E$ if and only if $u < v$ or if $v = EndState$ and $u \in \mathcal{H}$. The weight of arc $e = (u, v)$, denoted by $\boldsymbol{w}(e)$, and it is defined as follows: $\boldsymbol{w}(e) = \begin{cases} \sum_{j \in \mathcal{H}: u<j<v} \boldsymbol{d}(j)(\boldsymbol{r}(j) - \boldsymbol{r}(u)) & \text{if } v \in \mathcal{H}, \\ \sum_{j \in \mathcal{H}: u<j} \boldsymbol{d}(j)(\boldsymbol{r}(j) - \boldsymbol{r}(u)) & \text{otherwise.} \end{cases}$

where *MinState* denotes the minimal state of $\mathcal{H}$. A distribution is associated with a path. For the remaining, we focus on certain paths P provided with distribution $\boldsymbol{d}_P$ from state *MinState* to state *EndState* in graph G.

In fact, computing $\boldsymbol{d}_P$ is equivalent to compute a shortest path in G from state *MinState* to state *EndState* with n arcs. Such an algorithm based on dynamic programming with complexity $O(N^2\, n)$ is given in [4].

Example 3. Let $\mathcal{A} = (\mathbf{A}, p(\mathbf{A}))$ be a discrete distribution with $\mathbf{A} = \{0, 2, 3, 5, 7\}$ and $p(\mathbf{A}) = [0.05, 0.3, 0.15, 0.2, 0.3]$. We aim to reduce the state space to $n = 3$ states and the reward function $\boldsymbol{r}$ is defined as follows: $\forall\, a_i \in \mathbf{A}$, $\boldsymbol{r}(a_i) = a_i$. The reward of the initial distribution, $R[\mathcal{A}] = \sum_{a_i \in \mathbf{A}} \boldsymbol{r}(a_i)\, p_{\mathbf{A}}(i) = 4.15$. The computation of the optimal upper bound $(\overline{\mathcal{A}})$ corresponds to explore all 3-hops paths from $EndState = 7$ such that $R[\overline{\mathcal{A}}] - R[\mathcal{A}]$ is minimal (see Figure 6). This can be done by applying the algorithm presented in [4]. The optimal upper bound is $\overline{\mathcal{A}} = (\overline{\mathbf{A}}, p(\overline{\mathbf{A}}))$ with $\overline{\mathbf{A}} = \{2, 5, 7\}$, $p(\overline{\mathbf{A}}) = [0.35, 0.35, 0.3]$ and $R[\overline{\mathcal{A}}] = 4.55$.

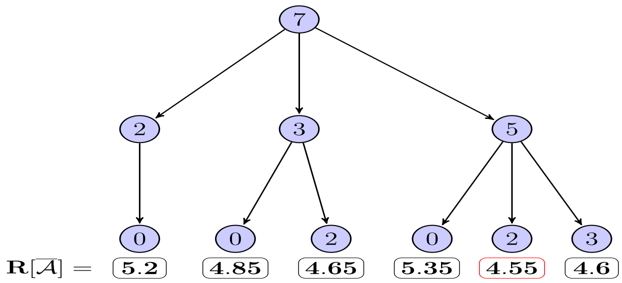


Fig. 6. Optimal upper bound histogram for $n = 3$ classes.

3.3 Performance measure bounds

In this section, we prove that performance measures of the single node with bounding histograms provide bounds for exact performance measures. We compare the buffer length under the exact traffic histogram with that obtained under the bounding traffic histogram. The buffer length at slot k ($Q(k)$) under an input traffic $A(k)$ is given by Equation 1. Similarly, the buffer length of the same system under input arrival $\tilde{A}(k)$, denoted by $\tilde{Q}(k)$ is given as

$$\tilde{Q}(k) = \mathtt{min}(B, (\tilde{Q}(k-1) + \tilde{A}(k) - S)^+), \quad k \in \mathbb{N}.$$

We have the following bounds, if the input arrivals are comparable in the sense of the $\leq_{st}$ order.

Proposition 3. *If $A(k) \leq_{st} \tilde{A}(k), \forall k \geq 0$, then $Q(k) \leq_{st} \tilde{Q}(k), \forall k \geq 0$.*

Proof. The proof is by induction: we suppose that $Q(k) \leq_{st} \tilde{Q}(k)$. We apply theorem 4.3.9 of Stoyan. As the function `min` is an increasing function, and $A(k) \leq_{st} \tilde{A}(k)$, then we can deduce that $Q(k+1) \leq_{st} \tilde{Q}(k+1)$.

Similarly, it follows from Equation 2 that we have bounds on the departure processes.

Proposition 4. *If $A(k) \leq_{st} \tilde{A}(k), \forall k \geq 0$, then $D(k) \leq_{st} \tilde{D}(k), \forall k \geq 0$.*

We can now give the main theorem, by assuming that input arrivals $\tilde{A}(k)$ are bounds built as explained in subsection 3.2. We give here only the upper bounding case and the lower bounds can be similarly obtained.

Theorem 1. *Let $\mathcal{A}$ (resp. $\tilde{\mathcal{A}}$ be the stationary exact (resp. upper bounding) input histogram, and $\mathcal{Q}$, $\mathcal{D}$ (resp. $\tilde{\mathcal{Q}}$, $\tilde{\mathcal{D}}$) be the stationary buffer length and the departure processes under the exact $\mathcal{A}$, (resp. upper bounding $\tilde{\mathcal{A}}$) input arrival, then we have:*

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

Proof. By construction $\mathcal{A} \leq_{st} \tilde{\mathcal{A}}$, and it follows from the above propositions that we have comparisons for all k, thus also for stationary processes when $k \to \infty$. Remark that by construction $\mathcal{Q}$ and $\mathcal{D}$ exist (due to the ergodicity assumption).

In the case when we consider a node in a tandem network, we have the same evolution equations as in the single node case, but the arrivals to a node are either external arrivals or the arrivals from the other nodes. By construction of histograms, we have bounding histograms for external arrivals. The internal arrivals are indeed the departure histograms of other nodes which are bounds. Therefore, we can also derive bounds for a node in a tandem network.

In order to compute the steady state distribution, we need an algorithm with a proved convergence test. Note that computing the difference between two successive distribution as [7] is not a correct test for convergence (see Stewart's book [13]). We propose the following algorithm based on the computation on a stochastic envelope Q^L and Q^U to prove the convergence.

Algorithm 1 State evolution algorithm

$Q^U(0) = \delta_B$, (Dirac at state B).
$Q^L(0) = \delta_0$, (Dirac at state 0).
$k = 0$.
repeat
  $Q^U(k+1) = f(Q^U(k)) = \min(B, (Q^U(k) + \mathcal{A} - S)^+)$.
  $Q^L(k+1) = f(Q^L(k)) = \min(B, (Q^L(k) + \mathcal{A} - S)^+)$.
until $||Q^U(k+1) - Q^L(k+1)||_\infty < \epsilon$.

Theorem 2. *Assume that the chain is ergodic and the steady state is* π. *We have*

$$Q^L(k) \leq_{st} Q^L(k+1) \leq_{st} \pi \leq_{st} Q^U(k+1) \leq_{st} Q^U(k).$$

Furthermore, the limit of $Q^L(k)$ *and* $Q^U(k)$ *is* π.

Proof. Remember that, for any non decreasing function f if $X \leq_{st} Y$ then, $f(X) \leq_{st} f(Y)$ [9]. Note that $\delta_0 \leq_{st} X$ is true for any distribution X defined on $\{0..B\}$. Therefore, $Q^L(0) \leq_{st} Q^L(1)$ and $f(Q^L(0)) \leq_{st} f(Q^L(1))$ because f is not deceasing. Thus $Q^L(1) \leq_{st} Q^L(2)$. By induction, we have $Q^L(k) \leq_{st} Q^L(k+1)$. The proof for $Q^U(k)$ is similar, noting that $X \leq_{st} \delta_B$ is true for any X.

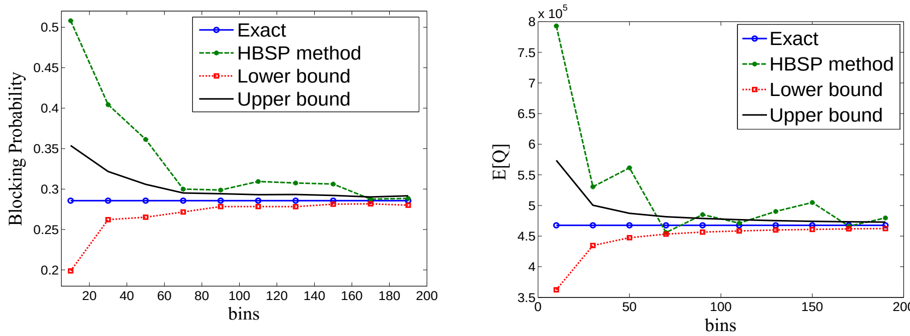
As $Q^L(k) \leq_{st} \delta_B$ the sequence is bounded and increasing. Therefore, the limit exists. Similarly, the sequence of $Q^U(k)$ has a limit. Finally, by the ergodicity of the chain, both limits are equal and the iteration of Q^L, Q^U converges. Checking the difference between Q^L and Q^U provides a proved test of convergence.

4 Real traffic experiments

We compute the performance measures of interest under real traffic traces by applying three methods: exact computation, HBSP method and our method. We are interested in blocking probability, buffer occupancy histogram and mean buffer occupancy. We first, consider a single finite buffer case and then study a network of nodes. For all the experiments, we suppose that the stationarity is reached according to Algorithm 1 for $\epsilon = 10^{-6}$, reward function is defined by $\boldsymbol{r}(a_i) = a_i$, $\forall\, a_i \in \boldsymbol{A}$. Real traffics are generally defined with large number of classes N. In order to accelerate the computation time of our bounds, we propose to reduce the initial size to n classes in two steps. First, we apply Tancrez's approach [14] to obtain bounds on N' ($n < N' < N$) states. In the following experiments we take N'=4000. In the second step, we apply our method to have bounding histograms on n states. The parameters considered in these experiments are taken from [7] to compare results.

4.1 Single node

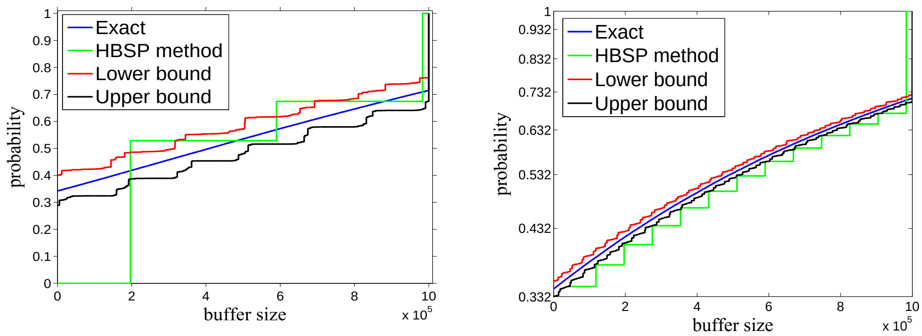
We first consider the single node under the MAWI traffic traces (Figure 1). We want to analyze the influence of the number of classes on the accuracy of the results. We set the mean transmission rate to $S = 110$ Mb/s and the buffer size to $B = 1$ Mb. We compute performance measures (Figure 7) for different number of bins (varying from 10 to 200). In each figure, we give the results computed by different methods: 1) exact result, 2) HBSP method, 3) Lower bound and 4) Upper bound. Obviously, when the number of bins increases the results become more accurate. But we must notice that the results provided by our bounds are very close to the exact ones.



(a) Blocking Probability (b) Mean buffer occupancy

Fig. 7. Number of classes vs Accuracy: QoS parameters using MAWI traffic trace

We can remark that for small values of bins, HBSP method gives worse results. Moreover, we see clearly from Figure 7.b that HBSP method does not provide bounds.

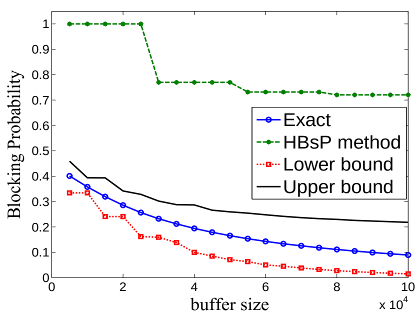


(a) bins=20 (b) bins=100

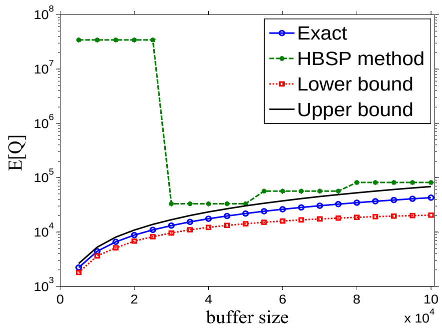
Fig. 8. Cumulative probability (cdf) of buffer occupancy under MAWI traffic trace

In Figure 8 we illustrate the cumulative probability distribution of buffer occupancy by taking number of bins equal to 20 or 100. Again, we see the HBSP method is not a bound and it does not provide a good approximation for small values of bins (bins=20). For bins equal to 100 all methods provide better results and our bounds are the most accurate ones. To get an idea of the execution times of the considered methods, we give the times for number of bins equal to 100. We find that the exact computations are obtained in 1897 seconds (s), the HBSP method in 0.007 s, the lower and upper bounds are respectively obtained in 0.35 s and 0.33 s. So, we remark that the HBSP method is the fastest one, but our bounds remain faster than the exact computation.

The second experiment is based on the CAIDA OC-48 traffic trace [1] 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 $E[\mathcal{A}] = 1.2885\,10^5\,bits$. We consider the relationship between buffer size and blocking probability (resp. mean buffer length) for bounding histograms, HBSP model and the exact result. The performance indices are calculated by varying the buffer size from $5.\,10^3\,bits$ to $10^5\,bits$.

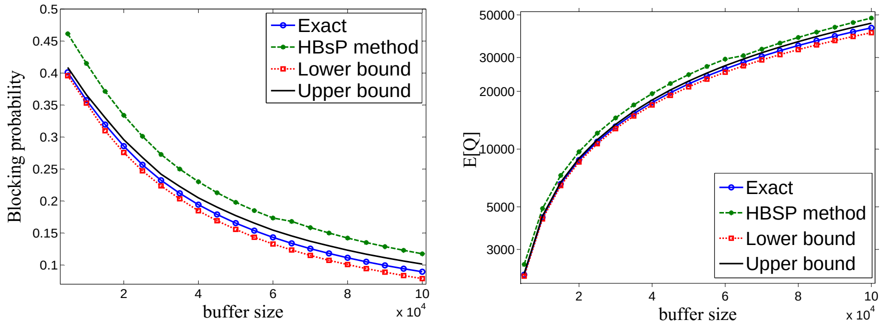


(a) Blocking Probability



(b) Mean buffer occupancy

Fig. 9. QoS parameters using CAIDA OC-48 traffic trace, bins=10

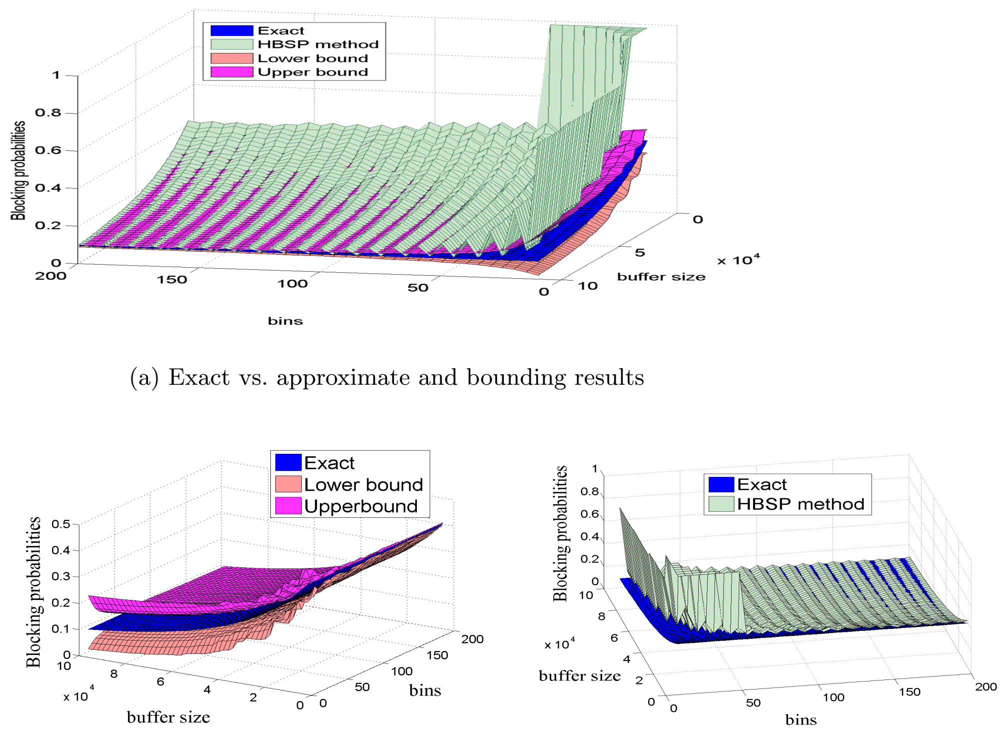


(a) Blocking Probability

(b) Mean buffer occupancy

Fig. 10. QoS parameters using CAIDA OC-48 traffic trace, bins=100

In Figure 9, we present the results with bins equal to 10 while Figure 10 presents the results with a number of bins equal to 100. When a number of bins equal to 10, the results obtained by HBSP method for blocking probability and mean buffer occupancy for small buffer sizes are not accurate. However, we see that the bounds are closer to the exact values. When the number of bins increases the accuracy of the HBSP method is improved and the bounds becomes very tight.



(a) Exact vs. approximate and bounding results

(b) Exact vs. bounding results

(c) Exact vs. HBSP results

Fig. 11. Blocking probability using CAIDA OC-48 traffic trace

In Figure 11, we give 3D representations to study the impact of the buffer size and the number of bins on blocking probabilities. We see from Figure 11.a that for small number of classes the HBSP method does not converge when buffer size is approximately less than $3\,.10^3$ and gives less accurate results elsewhere. However, our bounds let us to provide fairly good coverage on the exact results. We notice also that when the number of classes increases, the used methods gives a closer results to the exact ones. Moreover, Figure 11.c illustrates well that the HBSP method does not provide bounds.

4.2 Queueing network

In this section, we study a tandem queueing network with the MAWI traffic trace (Figure 2) as input arrival histogram. The network is defined by 3 service nodes having the following deterministic service rate: S_1=110 Mb/s, S_2=107.5 Mb/s and S_3=106.5 Mb/s. The buffers sizes are set to B_1=2 Mb, B_2=1 Mb and B_3=1 Mb. The analysis of the network is performed for two reductions 100 and 500 respectively on the input histogram of each queue (see Table 1).

We study as performance measures: blocking probability ($Prob(B)$), mean buffer length ($E[Q_i]_{i=\{1,2,3\}}$) and throughput (expected value of the departure histogram) ($E[D_i]_{i=\{1,2,3\}}$). We compute also mean transmission delays ($E[T_i]_{i=\{1...4\}}$) in each queue by using Little's theorem. In the last row, we present the execution time for the whole network analysis. These examples show that our method is fast even if it is slightly higher than HBSP. Moreover, we derive bounds which are more accurate than the results given by HBSP.

		Bins	100			500		
		Exact	**Lower b.**	**Upper b.**	**HBSP**	**Lower b.**	**Upper b.**	**HBSP**
Queue 1	$Prob(B)$	0.1818	0.1714	0.1937	0.2147	0.1798	0.1846	0.1854
	$E[Q_1]$	938529	908137	969289	1019260	931778	945367	950271
	$E[D_1]\ .10^6$	4.26185	4.25076	4.27272	4.24416	4.25954	4.26416	4.26055
	$E[T_1]$	0.2202	0.2136	0.2269	0.2402	0.2188	0.2217	0.2230
Queue 2	$Prob(B)$	0.1735	0.1200	0.2052	0.1551	0.1481	0.1809	0.1559
	$E[Q_2]$	488719	425739	524331	468094	464797	497174	474322
	$E[D_2]\ .10^6$	4.24692	4.23377	4.25604	4.23441	4.24325	4.24903	4.24776
	$E[T_2]$	0.1151	0.1006	0.1232	0.1105	0.1095	0.1170	0.1117
Queue 3	$Prob(B)$	0.1635	0.0782	0.2379	$1.18\ .10^{-6}$	0.1286	0.1799	0.2223
	$E[Q_3]$	505240	388396	585229	39418.4	463092	524231	564428
	$E[D_3]\ .10^6$	4.2408	4.22816	4.24768	4.23441	4.23732	4.2425	4.23809
	$E[T_3]$	0.1191	0.0919	0.1378	0.0093	0.1093	0.1236	0.1332
Ex. Time (s)		21868	2.20	2.16	0.13	14.57	18.21	0.28

Table 1. Numerical results for the network using MAWI Traffic trace

5 Conclusions

Performance analysis of communication networks under general traffic is very difficult and sometimes impossible by simulation and queueing theory. We propose in this paper to develop a formalism based on stochastic bounds in order to reduce the size of the traffic histograms. We apply an algorithm based on dynamic programming to define bounding histograms. We analyze the performance of tandem queueing networks. We consider real traffic traces and derive bounds on different performance measures as blocking probabilities and buffer occupancy. We compare our results with the system under the exact traffic trace, and those obtained from the HBSP approximation. We show clearly that our results are more accurate and can be obtained within very interesting execution times. The more important point is the fact that we derive stochastic bounds which provide guarantee for non decreasing rewards. We will extend the theory in the near future to deal with non stationary flows.

References

1. Caida, traces of oc48 link at ames internet exchange (aix) (april 24, 2003), accessed via datcat - internet data measurement catalog, http://imdc.datacat.org.
2. F. Aït-Salaht, J. Cohen, H. Castel-Taleb, J. M. Fourneau, and N. Pekergin. Accuracy vs. complexity: the stochastic bound approach. In *11th International Workshop on Disrete Event Systems*, pages 343–348, 2012.
3. F. Baccelli, G. Cohen, G. J. Olsder, and J.-P. Quadrat. *Synchronization and Linearity: An Algebra for Discrete Event Systems*. Willey, New York, 1992.
4. R. Guérin and A. Orda. Computing shortest paths for any number of hops. *IEEE/ACM Trans. Netw.*, 10(5):613–620, 2002.
5. E. Hernández-Orallo and J. Vila-Carbó. Network performance analysis based on histogram workload models. In *MASCOTS*, pages 209–216, 2007.
6. E. Hernández-Orallo and J. Vila-Carbó. Web server performance analysis using histogram workload models. *Computer Networks*, 53(15):2727–2739, 2009.
7. E. Hernández-Orallo and J. Vila-Carbó. Network queue and loss analysis using histogram-based traffic models. *Computer Communications*, 33(2):190–201, 2010.
8. S.-K. Kweon and K. G. Shin. Real-time transport of mpeg video with a statistically guaranteed loss ratio in atm networks. *IEEE Transactions on Parallel and Distributed Systems*, pages 12–4, 2001.
9. A. Muller and D. Stoyan. *Comparison Methods for Stochastic Models and Risks*. Wiley, New York, NY, 2002.
10. J. P. Robertson. The computation of aggregate loss distributions. In *In Proceedings of the Casualty Actuarial Society*, pages 57–133, 1992.
11. P. Skelly, M. Schwartz, and S. S. Dixit. A histogram-based model for video traffic behavior in an atm multiplexer. *IEEE/ACM Trans. Netw.*, 1(4):446–459, 1993.
12. K. C. Sony and K. Cho. Traffic data repository at the wide project. In *In Proceedings of USENIX 2000 Annual Technical Conference: FREENIX Track*, pages 263–270, 2000.
13. W. Stewart. *Introduction to the numerical Solution of Markov Chains*. Princeton University Press, New Jersey, 1995.
14. J.-S. Tancrez, P. Semal, and P. Chevalier. Histogram based bounds and approximations for production lines. *European Journal of Operational Research*, 197(3):1133–1141, 2009.