A bounding histogram approach for network performance analysis

F. Aït-Salaht*, H. Castel-Taleb†, J.M. Fourneau*and N. Pekergin‡
*PRiSM, Univ. Versailles St Quentin, UMR CNRS 8144, Versailles France
Email: safa, jmf@prism.uvsq.fr
† SAMOVAR, UMR 5157, Télécom Sud Paris, Evry, France
Email: Castel@it-sudparis.eu
‡LACL, Univ. Paris Est, Créteil, France
Email: nihal.pekergin@u-pec.fr

***Abstract*—We analyze the performance of a network under general traffics derived from traces. We apply stochastic comparisons in order to derive bounding histograms with a reduced size and complexity. We prove some stochastic monotonicity properties for the network elements, in order to derive bounds on the performance measures such as delays and losses. We show clearly that this approach provides an attractive solution as a trade-off between accuracy of the results and computation times. Moreover, we compare our results with an approximative method previously published, in order to show the accuracy of the bounds, and to highlight the benefits of our approach for network dimensioning.**

I. Introduction

In this paper, we propose a new performance evaluation method based on stochastic ordering and real traces. We only assume stationarity of the traffic and using real traces, we get rid the Poisson assumption. Unlike many approximative methods such as diffusion or fluid, our method provides a proof for a guarantee on QoS requirements. We consider real traffic traces obtained from network measurements. As the histogram is too large, we have developed a stochastic bound approach to deal with distributions simpler to handle and which provides upper and lower bounds on the exact results.

There has been several amount of work on Histogram-based approach for performance models. It was first introduced by Skelly and al. [13] in the area of network calculus, to model the video sources, predict buffer occupancy distributions and cell loss rates for multiplexed streams. Hernández and al. [5]–[8] have proposed a new performance analysis for obtaining histogram of buffer occupancy distribution by reducing the state space of the traffic trace. This new stochastic process called HBSP (Histogram Based Stochastic Process) is based on a histogram model HD as an input traffic which is supplied through buffers of finite capacity and the First Come First Served service discipline with deterministic (Constant) distribution. The analysis method consists in solve the HD/D/1/K queueing system, by dividing the state space into equal subintervals, in order to derive a probability distribution on a smaller state space. In Tancrez and al. [16], the authors consider a continuous time histogram, and they also divide the support into equal subintervals. Each of these subintervals is mapped into a single point of the discrete distribution. This point is the upper limit (or the lower limit), and the mass probability of the sub-interval is associated to that point. Such a mapping of the points and the mass probability provides bounds for the $\leq_{st}$ ordering [10].

Our approach is based on the following key ideas: we consider the traffic histogram derived from a real trace with a discrete sampling period. We suppose that the traffic is stationary. Then, we apply a dynamic programming algorithm introduced in [2] in order to derive an optimal bounding traffic histogram. The bounds are based on the strong stochastic ordering $\leq_{st}$, and the algorithms allow to control the size of the distribution and compute the most accurate bound according to a reward function. Finally, we show that networking elements are monotone in the following sense: when we use a stochastic upper (resp. lower) bound of the input process, we get a stochastic upper (resp. lower) bound of the distribution of the buffer length, the departure process, the response times and the loss probabilities.

Unlike many approaches based on histograms, we can analyze networks (even if we only consider tree networks in that paper for the sake of conciseness). This is one of our main contribution and it is due to the stochastic monotony proved for the networking element. The proof is based on Lindley's equation [9]. The guarantee on QoS requirements is the other main contribution. We show how to reduce the size of the histogram to handle it efficiently and obtain some bounds on output performance measures such as average response times, loss probabilities, and buffer length. Our method provides an attractive solution for the dimensioning problem as a tradeoff between accuracy and computational complexity.

The paper is organized as follows: first, we describe the model and the traffic histograms. Then, in section III, we present the analysis of the queue and the queueing network in order to compute the performance measures. We prove the monotonicity properties to derive the bounds. We give in section IV the numerical results for different queueing networks, using real traffic traces.

II. Model and Notations

We consider networks which are composed of i) traffic sources (input flows); ii) Finite capacity queues and iii) Splitter (routing elements). The topology of the network is supposed to be a tree, thus the networking elements can be analyzed in a sequential and greedy manner following the topological ordering associated with the tree. The elements are numbered according to this ordering: traffic source receives the lowest number and if there exists a directed link from networking element i to networking element j, then we must have $i < j$.

Flows and delays are specified by discrete distributions. In the sequel, we use interchangeably the terms histogram and discrete probability distribution because many published papers have used this terminology [5]–[8]. The distributions for the delays are expressed on integer multiples of the time interval, which is called a slot. The measurements of the network traffic are made in bits per second. But we do not need such a detailed description for the amount of data. We introduce a data unit which gathers a constant number of bits (say $\mathbf{D}$) in order to reduce the sizes, hence the complexity of underlying models. As an example for some numerical computations, we have taken a data unit equal to $1\,Kb$ ($1\,Kb = 10^3\,bits$). We assume that the arrivals and services occur by batch of data units and they are stationary. Under histogram based input flows, the analysis of network elements consists into the analysis of a Discrete Markov Chains (DTMC).

For a distribution H, $H(i)$ will denote the probability of i and E^H will be the set of elements such that $H(i)$ is positive (i.e. $H(i) > 0$). The size of the distribution H is, by definition, the cardinal of set E^H. It is really important to note that all the theoretical results are obtained on distributions on sets $\mathbb{N}$ or $\mathbb{Z}$ (we need negative numbers when we combine arrivals and departures) but the algorithms take only into account the elements with a positive probability. Furthermore, the compression algorithms we advocate here, reduces the size of the distributions and the complexity of the computation while computing stochastic bounds on the results. This is the key idea for the trade-off between the complexity of the resolution (clearly related to the size of the distributions) and the accuracy. We give in Fig. 1, the histogram of MAWI traffic trace [14] that will be used in numerical results.

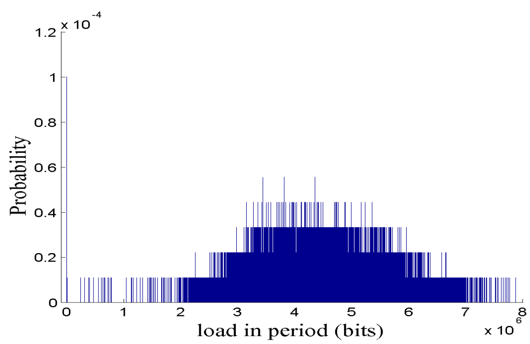

Fig. 1. MAWI input arrival histogram

We give in Fig. 2, the model of a single queue with input and output parameters. The number of servers is fixed to one, and buffer capacity is supposed to be finite and equal to $\mathbf{B}$. **The Input Arrivals** (the number of data units arrived during a slot) are considered to be a sequence of i.i.d., stationary flow defined by histogram H_1 taking values on state space $E^{H_1} \subset \{0 \cdots \mathbf{N}\}$ ($\mathbf{N} < \mathbf{B}$). Without loss of generality, we assume that state 0 is in E^{H_1}. **The Service Policy** is First In First Out (FIFO), and work-conserving, i.e. the server cannot be idle if there are some data units in the buffer. **The service capacity** (the number of data units that can be served in each slot) is independent and identically distributed and specified by histogram H_2 defined on $E^{H_2} \subset \{1 \cdots \mathbf{C}\}$. Note that 0 is not in E^{H_2} due to the work conserving property. The histograms of the output result are computed at steady-state. **The Buffer Length** (the number of data units in the buffer) is specified by histogram H_3 defined on set $E^{H_3} \subset \{0 \cdots \mathbf{B}\}$. **The Departure Process** (the number of data units leaving the queue after service per slot) is specified by histogram H_5 defined on set $E^{H_5} \subset \{0 \cdots \mathbf{C}\}$. **The Response Time** (Delay) (the number of slots between the arrival and departure time for a data unit) is specified by histogram H_4 defined on state space $E^{H_4} \subset \mathbb{N}$.

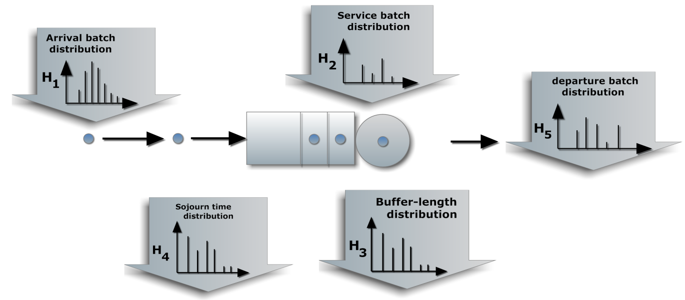

Fig. 2. Input and output parameters of a queueing model

III. QUEUE AND NETWORK ANALYSIS

A. Basic Properties of the Model

The buffer length evolution in the queue is given by a time-homogeneous Discrete-Time Markov Chain (DTMC) $\{X_n,\, n \geq 0\}$ taking values in a totally ordered state space, $\{0 \cdots \mathbf{B}\}$. Let A (resp. S) be an i.i.d. distributed random variable representing the number of data units received (resp. the service capacity) during a time slot with pmf H_1 (resp. H_2). We assume that arrivals take place at the beginning of a slot while departures occur at the end. Therefore, the equation of the queue length evolution is [9]:

$$X_{n+1} = \mathtt{min}(\mathbf{B},\, (X_n + A - S)^+), \tag{1}$$

where $(X)^+ = \mathtt{max}(0, X)$. Clearly, due to the assumptions, $\{X_n, n \geq 0\}$ is a Discrete-Time Markov chain. In order to construct transition probability matrix P associated to $\{X_n, n \geq 0\}$, we consider a set of matrices $\mathcal{U} = \{U_k\}_{k \in \mathcal{K}}$, representing all possible modifications from a state depending on buffer capacity, arrival and service distributions:

$$\mathcal{K} = \{x - y, \forall x \in E^{H_1}, \forall y \in E^{H_2}\}.$$

$$k \in \{\mathcal{K} | k \geq 0\},\;\; U_k[i,j] = \begin{cases} 1 & if \;\; j = \mathtt{min}(\mathbf{B},\, \mathrm{k}+\mathrm{i}), \\ 0 & \quad otherwise. \end{cases}$$

$$k \in \{\mathcal{K} | k < 0\},\;\; U_k[i,j] = \begin{cases} 1 & if \;\; j = \mathtt{max}(0,\, k+i), \\ 0 & \quad otherwise. \end{cases}$$

Example 1: For $\mathbf{B} = 3$, U_1 and U_{-2} are:

$$U_1 = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \text{ and } U_{-2} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

Proposition 1 (Construction of P): Transition probability matrix P is described by:

$$P = \sum_{k\in\mathcal{K}} U_k . \left(\sum_{i\in E^{H_1}} \sum_{j\in E^{H_2}} H_1(i)\ H_2(j)\ \mathbb{1}_{\{\min(\mathbf{B},\, i-j)=k\}} \right).$$

Proof: For each row (state) i, the impact of a possible transition $k \in \mathcal{K}$ is defined by matrix U_k. Its probability is described by the sum of probabilities inside the brackets which enumerates all the combination of arrivals and departures which lead to k. After summing over all transitions (events) $k \in \mathcal{K}$, each row i is a probability vector, and P is a transition probability matrix. ■

Example 2: Let $H_1 = [0.2, 0.5, 0.3]$ with $E^{H_1} = \{0, 1, 2\}$, $H_2 = [0.5, 0.5]$ with $E^{H_2} = \{1, 2\}$, and $\mathbf{B} = 3$.

$P = \sum_{k\in\mathcal{K}} L_k$, where $\mathcal{K} = \{-2, -1, 0, 1\}$ and

$$L_k = U_k . \left(\sum_{i\in E^{H_1}} \sum_{j\in E^{H_2}} H_1(i)\ H_2(j)\ \mathbb{1}_{\{\min(3,\, i-j)=k\}} \right), \text{with}$$

$$L_{-2}=\begin{pmatrix} 0.1 & 0 & 0 & 0 \\ 0.1 & 0 & 0 & 0 \\ 0.1 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 \end{pmatrix}, \qquad L_{-1}=\begin{pmatrix} 0.35 & 0 & 0 & 0 \\ 0.35 & 0 & 0 & 0 \\ 0 & 0.35 & 0 & 0 \\ 0 & 0 & 0.35 & 0 \end{pmatrix},$$

$$L_0=\begin{pmatrix} 0.4 & 0 & 0 & 0 \\ 0 & 0.4 & 0 & 0 \\ 0 & 0 & 0.4 & 0 \\ 0 & 0 & 0 & 0.4 \end{pmatrix}, L_1=\begin{pmatrix} 0 & 0.15 & 0 & 0 \\ 0 & 0 & 0.15 & 0 \\ 0 & 0 & 0 & 0.15 \\ 0 & 0 & 0 & 0.15 \end{pmatrix},$$

$$P = \begin{pmatrix} 0.85 & 0.15 & 0 & 0 \\ 0.45 & 0.4 & 0.15 & 0 \\ 0.1 & 0.35 & 0.4 & 0.15 \\ 0 & 0.1 & 0.35 & 0.55 \end{pmatrix}.$$

We now prove one of the most important properties of the networking elements: the strong stochastic monotony. We proceed in few steps for the sake of clarity.

Property 1: For all $k \in \mathcal{K}$, positive matrix $U_k . \left(\sum_{i\in E^{H_1}} \sum_{j\in E^{H_2}} H_1(i)\ H_2(j)\ \mathbb{1}_{\{\min(\mathbf{B},\,\mathrm{i-j})=\mathrm{k}\}} \right)$ is st-monotone.

Proof: By construction, matrices U_k are st-monotone. Since the probability sum in the brackets is a positive constant depending only on k, the weighted versions are also st-monotone (see Property 3 of the appendix). ■

Proposition 2 (St-Monotonicity of the Markov chain): $\{X_n, n \geq 0\}$ is st-monotone.

Proof: Since the sum of st-monotone positive matrices is st-monotone (see Property 3 of the appendix), it follows from Property 1 and the construction of P in Proposition 1 that P is st-monotone. Thus, the associated DTMC is st-monotone (see Definition 4 of the appendix). ■

Monotonicity property is used for the convergence proof of our method [1]

Proposition 3 (Stationarity of the Markov Chain): Let $\{X_i, i \geq 0\}$ be the DTMC with probability transition matrix P constructed as in Proposition 1 such that $\max(E^{H_1}) > \min(E^{H_2})$, we assume that the chain is irreducible, then it admits a unique steady state vector π computed as follows:

$$\pi \,.\, P = \pi, \ \ \pi(i) \geq 0, \qquad \sum_{i=0}^{\mathbf{B}} \pi(i) = 1. \tag{2}$$

Proof: First, one can check that the chain is aperiodic. Clearly, from state 0 we have a positive probability to stay in 0 after arrival of an empty batch. No service take place and $P(0, 0) > 0$. Thus, the chain is aperiodic.
Thus, the finite-state Markov chain is ergodic, which implies the existence of a steady-state distribution π. ■

The computation of the steady-state probabilities, taking into account the matrix formulation, can be performed using several numerical techniques (see for instance [15]). We advocate that, when we compute guarantees for the Quality of Service, one must provide stable and accurate algorithms with a proved test of convergence. We have found that the numerical technique used in [5], [6], [8] could be numerically unstable. Moreover, there is no proof of convergence for this algorithm, which is based on a fixed point iteration on the distribution. Many references (see again for instance [15] for this problem with Markov chain) has pointed out that this type of method may lead to important numerical errors due to false convergence test. So, in the numerical experiments we have used a new algorithm [1], which takes into account the monotony of matrix P to prove a convergence test. It also has the same complexity as the approach presented in [5]. Note that we have two kinds of numerical problems: steady-state analysis and the computation of the distribution of the response time. This last computation is based on a vector matrix multiplication in sparse format as we will see in the next section. We have developed a solver in Matlab® for a simple testbed. It is important to remark that our approach can be associated with any numerical solvers for DTMC (for instance XBorne [4] or OLYMP [11]).

B. Computation of output parameters

We now show how the output histograms can be computed when the corresponding DTMC is ergodic.

Proposition 4 (Buffer Length, H_3): As the queue is operated with the Arrival First mode, the buffer length distribution H_3 before the instant of arrivals corresponds to the steady-state distribution π of the DTMC.

We now build the distribution of the number of data units departing from the queue at the steady-state. H_3 is the steady-state distribution just before the arrival instants. It is, therefore, the distribution of the state seen by a batch of arrivals. The arrivals modify this distribution, adding a new group of data units represented by distribution (H_1). Therefore, after arrivals, we observe a buffer length distributed with H_q:

$$H_q = H_3 \otimes H_1 \tag{3}$$

where $\otimes$ is the convolution on the distributions. Now remember that the network elements are work-conserving. Thus, the number of data units emitted is the minimum between the channel capacity and the queue size before the service.

Proposition 5 (Batch Departure, H_5): The departure histogram H_5 is defined on $\mathcal{S}$ such that $\mathcal{S} = \{k \,|\, \forall i \in$

E^{H_q} and $\forall j \in E^{H_2},\ k = \min(i, j)\}$ and computed from H_q as follows

$$H_5(w) = \sum_{i \in E^{H_q}} \sum_{j \in E^{H_2}} H_q(i)\, H_2(j)\, \mathbb{1}_{\{\min(i,j)=w\}}, \quad \forall w \in \mathcal{S} \tag{4}$$

Let H_L be the distribution of the number of data units lost at the entrance of a finite queue. With a Tail Drop policy and an Arrival First assumptions modeled by Eq. 1, the number of losses at time n is $(X_n + A - S - \mathbf{B})^+$ Let $(-H_2)$ be the distribution defined by:

$$E^{-H_2} = \{x | -x \in E^{H_2}\} \quad and \quad -H_2(x) = H_2(-x)$$

In a more abstracted description, we first compute $H_n = H_3 \otimes H_1 \otimes (-H_2)$ and then we translate and truncate the distribution. Note that set E^{H_n} can contain some negative elements.

Proposition 6 (Losses, H_L): The distribution of losses under the Tail Drop policy is:

$$\begin{cases} H_L(k - \mathbf{B}) &= H_n(k) \qquad k > \mathbf{B} \\ H_L(0) &= \sum_{k \le \mathbf{B}} H_n(k) \end{cases}$$

Then, the loss probability P_L can be defined as follow:

$$P_L = \frac{\mathbb{E}[H_L]}{\mathbb{E}[H_1]}.$$

Response Time Bounds: Let H_4 denote its distribution. The computation of histograms for the response time depends on the service policy. Here, we consider FIFO policy and we derive upper and lower bounds instead of the exact distribution. Both distributions are obtained in an algorithmic way. The stochastic bounds are sufficient for checking the QoS requirements and they are much easier to compute. Before proceeding with the proofs, we explain the approach. We consider a data unit arriving at the network element and we model its response time in the queue as an absorbing Markov chain. We compute the distribution of the time before being absorbed. We condition on the state seen by an arriving batch. Then, we have to take into account where is the data unit in the batch. For upper bounds, we consider a batch with a maximum size and we assume that the data unit is the last one of the batch. For the lower bound, we suppose that the data unit is at the beginning of the batch. We first build stochastic matrix R as follows: the transitions of R are obtained letting the distribution of H_1 be a Dirac at 0 (i.e. $H_1(0) = 1$) and making state 0 absorbing. Intuitively speaking, matrix R represents the chain when the arrivals are stopped after the arrival of the data unit.

$$R = \sum_{j \in E^{H_2}} U_{-j}\ H_2(j).$$

Let us denote by H_4^l the lower bound of the response time. The head of the distribution is exact for the response time of the first customer of the batch and the tail gives a lower bound. We have a new trade-off between accuracy and computation of the tail of the distribution introduced by constant $\mathbf{T}$. First, we get the distribution of the queue length including the new data unit. Let α_1 be this distribution. Then, we compute by iteration on the time instants the probability that the data unit leaves the element at this particular instant. As state 0 is absorbing, at line 7, $\beta(0)$ is the probability to be in 0 at time t and $\delta(0)$ to be in that state at time $t-1$. e_j is a row vector whose entries are all equal to 0 except component j which is 1.

Algorithm 1 Lower bound of response time under FIFO policy

Require: $\pi : 1 \times \mathbf{B}+1$ vector; $R : \mathbf{B}+1 \times \mathrm{B}+1$ matrix.
Ensure: Lower bound distribution of response time H_4^l.

1: $\alpha_1(i) = \pi(i-1), \quad 0 < i < \mathbf{B}$.
2: $\alpha_1(0) = 0$; $\alpha_1(\mathbf{B}) = \pi(\mathbf{B}-1) + \pi(\mathbf{B})$.
3: **for** $(j = 0$ to $\mathbf{B} \mid \alpha_1(\mathrm{j}) > 0)$ **do**
4: $\quad \delta = e_j$.
5: $\quad$ **for** $t = 1$ to $\mathbf{T}$ **do**
6: $\quad\quad \beta = \delta\ R$.
7: $\quad\quad \eta_j(t) = \beta(0) - \delta(0)$; $\delta = \beta$.
8: $\quad$ **end for**
9: $\quad \eta_j(\mathbf{T}+1) = 1 - \sum_{t=1}^{\mathbf{T}} \eta_j(t)$.
10: $\quad \eta_j(t) = 0, \quad \forall t > \mathbf{T}+1$.
11: **end for**
12: $H_4^l(t) = \sum_j \alpha_1(j)\ \eta_j(t), \quad 0 < t \le \mathbf{B}$.

Clearly, for t between 1 and $\mathbf{T}$, $H_4^l(t)$ is the exact distribution of the response time of the first data unit when it stays less than $\mathbf{T}+1$ slots in the networking element. The lower bound is obtained by considering the smallest delay larger than $\mathbf{T}$ slots for the remaining part of the distribution. Clearly, using a larger value of $\mathbf{T}$ provides a more accurate lower bound of the response time.

We use a similar approach for the distribution of the upper bound H_4^u. First, we assume that the data unit is the last one in the batch and that the batch size is $\mathbf{N}$. We compute the distribution (say $\alpha_{\mathbf{N}}$) of the size of the queue including the batch of $\mathbf{N}$ data units which have entered the elements at the beginning of the time slot. Then, we compute, in an iteration on the time instants up to time $\mathbf{T}$, the probability that a data unit completes its service.

Algorithm 2 Upper bound of response time under FIFO policy

Require: $\pi : 1 \times \mathbf{B}+1$ vector; $R : \mathbf{B}+1 \times \mathbf{B}+1$ matrix.
Ensure: Upper bound distribution of response time H_4^u.

$\alpha_{\mathbf{N}}(i) = 0, \quad \forall\ 0 \le i < \mathbf{N}$.
$\alpha_{\mathbf{N}}(i) = \pi(i - \mathbf{N}), \quad \forall\ \mathbf{N} \le i < \mathbf{B}$.
$\alpha_{\mathbf{N}}(\mathbf{B}) = 1 - \sum_{\mathrm{i}=0}^{\mathbf{B}-1} \alpha_{\mathbf{N}}(\mathrm{i})$;
for $(j = 0$ to $\mathbf{B} \mid \alpha_{\mathbf{N}}(\mathrm{j}) > 0)$ **do**
$\quad \delta = e_j$.
$\quad$ **for** $t = 1$ to T **do**
$\quad\quad \beta = \delta\ R$.
$\quad\quad \eta_j(t) = \beta(0) - \delta(0)$; $\delta = \beta$.
$\quad$ **end for**
$\quad \eta_j(t) = 0, \quad \forall t = \mathbf{T}+1 \cdots \mathbf{B}-1$.
$\quad \eta_j(\mathbf{B}) = 1 - \sum_{\mathrm{t}=1}^{\mathbf{T}} \eta_{\mathrm{j}}(\mathrm{t})$.
end for
$H_4^u(t) = \sum_j \alpha_{\mathbf{N}}(j)\ \eta_j(t), \quad 0 < t \le \mathbf{B}$.

The remaining part of the distribution is set to $H_4^u(\mathbf{B})$, because the element is work conserving and FIFO and the response time is, therefore, smaller than the buffer capacity.

C. Monotonicity Results

We now prove some monotonicity properties for a queue. We aim to show that if the input parameters (i.e., arrival and service histograms) are comparable with the $\leq_{st}$ ordering, then the output parameters (i.e., buffer-occupancy, departure flow and response time histograms) are also comparable with the same ordering. This property will be called H-monotonicity of an output parameter with respect to the arrivals H_1 and services H_2 histograms. We first prove that the Markov chains are comparable.

Proposition 7 (H-Monotonicity of DTMC): Let $\{X_i^a, i \geq 0\}$ be a DTMC associated with arrival (resp. service) histograms H_1^a (resp. H_2^a), and let $\{X_i^b, i \geq 0\}$ be a DTMC associated with arrival (resp. service) histograms H_1^b (resp. H_2^b). We assume that stationarity conditions (Proposition 3) are satisfied for both chains. If $H_1^a \leq_{st} H_1^b$ and $H_2^a \geq_{st} H_2^b$, then

$$\{X_i^a, i \geq 0\} \leq_{st} \{X_i^b, i \geq 0\}.$$

Proof: For the sake of simplicity, we give here the proof based on Equation 1. The random arrivals A and S are defined by histograms H_1 and H_2. Let A^a (resp. A^b) be the random variable corresponding to H_1^a (resp. H_1^b) and similarly S^a (resp. S^b) be the random variable corresponding to H_2^a (resp. H_2^b). Since $H_1^a \leq_{st} H_1^b$, then $A^a \leq_{st} A^b$. We have also $-S^a \leq_{st} -S^b$, from Property 2 and the hypothesis that $H_2^a \geq_{st} H_2^b$. We now consider two DTMCs $\{X^a\}$ and $\{X^b\}$ respectively governed by processes A^a, $-S^a$ and A^b, $-S^b$. We show by induction that $X_n^a \leq_{st} X_n^b$, $\forall n$.
Let $X_0^a = X_0^b = 0$. Since $\mathtt{max}(0,.)$ and $\mathtt{min}(\mathrm{B},.)$ are increasing functions, it follows from the closure properties of the $\leq_{st}$ ordering (Property 2) that $X_1^a \leq_{st} X_1^b$. Similarly, if $X_n^a \leq_{st} X_n^b$, then from Property 2 we have: $X_{n+1}^a \leq_{st} X_{n+1}^b$. ■

Theorem 1: [H-monotonicity for H_3 and H_5]
If $H_1^a \leq_{st} H_1^b$ and $H_2^a \geq_{st} H_2^b \Longrightarrow \begin{cases} H_3^a \leq_{st} H_3^b. \\ H_5^a \leq_{st} H_5^b. \end{cases}$

Proof: From Proposition 7, the corresponding DTMCs are also comparable and they are ergodic by assumptions. Thus, the steady-state distributions are comparable due to Proposition 8 (i.e. $\pi^a \leq_{st} \pi^b$) and $H_3^a \leq_{st} H_3^b$ holds from Proposition 4. The comparison of departure processes (H_5) follows from Proposition 5. Indeed, the convolution and the truncation of distributions are consistent with the stochastic ordering. ■

Theorem 2 (H-Monotonicity for H_L):
If $H_1^a \leq_{st} H_1^b$ and $H_2^b \leq_{st} H_2^a$ and the element is work conserving and operated under the Tail Drop policy, then the losses are monotone, i.e. $H_L^a \leq_{st} H_L^b$.

Proof: According to Theorem 1, the assumptions of the proposition imply that the steady-state distributions are also comparable: i.e. $H_3^a \leq_{st} H_3^b$. Clearly, $H_2^b \leq_{st} H_2^a$ implies that $-(H_2^a) \leq_{st} -(H_2^b)$. It is well known [10] that the convolution of discrete distributions is consistent with the strong stochastic ordering. Therefore, $H_3^a \otimes H_1^a \otimes (-H_2^a) \leq_{st} H_3^b \otimes H_1^b \otimes (-H_2^b)$. Finally, the translation does not change the ordering relation between the distributions. ■

Theorem 3 (H-monotonicity for H_4):
Under FIFO policy,

if $H_1^a \leq_{st} H_1^b$ and $H_2^a \geq_{st} H_2^b \Longrightarrow \begin{cases} H_4^{l,a} \leq_{st} H_4^{l,b}. \\ H_4^{u,a} \leq_{st} H_4^{u,b}. \end{cases}$

Proof: We give the proof for the lower bound. The proof for the upper bound is similar and is omitted. We denote α_1^a (resp. α_1^b) the buffer length seen by the new customer (including this one) under arrival and service distributions H_1^a, H_2^a (resp. H_1^b, H_2^b). Since α_1 is computed through π (H_3), it follows from Theorem 1 and Algorithm 1, that $\alpha_1^a \leq_{st} \alpha_1^b$. Similarly, matrix R is constructed by stopping the arrivals, the corresponding absorbing DTMC are comparable in the sense of the $\leq_{st}$ ordering, thus it follows from Proposition 9 that $\forall j$ the absorption time when there are j data units, η_j are comparable: $\eta_j^a \leq_{st} \eta_j^b$. And we have the monotonicity: $\eta_j^a \leq_{st} \eta_{j+1}^a$. Since $\eta_i^a \leq_{st} \eta_i^b$, we can write $\forall 0 < t \leq \mathbf{B}$: $\sum_t^{\mathbf{B}} H_4^{l,a}(t) = \sum_i \alpha_1^a(i) \sum_t^{\mathbf{B}} \eta_i^a(t) \leq \sum_i \alpha_1^a(i) \sum_t^{\mathbf{B}} \eta_i^b(t)$.

We can write the last expression as

$\sum_i \alpha_1^b(i) \sum_t^{\mathbf{B}} \eta_i^b(t) + \sum_i (\alpha_1^a(i) - \alpha_1^b(i)) \sum_t^{\mathbf{B}} \eta_i^b(t)$.

From Lemma 1 of the appendix, the last expression is less than

$\sum_i \alpha_1^b(i) \sum_t^{\mathbf{B}} \eta_i^b(t) = \sum_t^{\mathbf{B}} H_4^{l,b}(t)$. Thus $H_4^{l,a} \leq_{st} H_4^{l,b}$. ■

D. Routing and Network analysis

In this section, we study split network operation (or splitter) which involves multiple streams [12]. When the input flow denoted H_S crosses a splitter, it is divided into m flows (or classes): $H_{S,1}, \ldots, H_{S,m}$. We define for the split operator the H-monotonicity as follows:

Definition 1: A splitter is said to be H-monotone, iff

$$H_S^a \leq_{st} H_S^b \Rightarrow \forall i,\ H_{S,i}^a \leq_{st} H_{S,i}^b.$$

Now, we study the case where the whole batch is splitted according to a routing probability. All the data units in a batch follow the same route (i.e., we route the batch rather than the data units). Let $p_i, 1 \leq i \leq m$ (such that $\sum_{i=1}^m p_i = 1$), be the routing probability of the batch to output i of the splitter. We also assume that successive batches are routed independently according to these routing probabilities. The probability distribution of any output flow i is for all $i \leq m$:

$$H_{S,i}(k) = p_i\ H_S(k),\ k \in E^{H_S},\ k > 0$$
$$H_{S,i}(0) = 1 - \sum_{k \neq 0} H_{S,i}(k).$$

Theorem 4: With a batch routing, the splitter is H-monotone.

Proof: We have the following equation $1 \leq \forall i \leq m$:

$$\sum_{k=l}^{n} H_{S,i}^a(k) = \sum_{k=l}^{n} p_i\ H_S^a(k), \quad \sum_{k=l}^{n} H_{S,i}^b(k) = \sum_{k=l}^{n} p_i\ H_S^b(k).$$

As if $H_S^a \leq_{st} H_S^b$, we get: $\sum_{k=l}^n H_S^a(k) \leq \sum_{k=l}^n H_S^b(k)$. Thus, for all $i \leq m$, we obtain:

$$\sum_{k=l}^{n} p_i H_{S,i}^a(k) \leq \sum_{k=l}^{n} p_i\ H_S^b(k).$$

Finally, $\sum_{k=l}^n H_{S,i}^a(k) \leq \sum_{k=l}^n H_{S,i}^b(k)$, which means that for all $i \leq m$, $H_{S,i}^a \leq_{st} H_{S,i}^b$. ■

IV. NUMERICAL RESULTS

We compute the performance measures of interest under real traffic traces by applying three methods: exact computation, HBSP algorithm [5]–[8] and our method [2] (lower bound: $L.b$ and upper bound: $U.b$). We are interested in blocking probability (the probability that the buffer is full, denoted by $Prob(\mathbf{B})$), buffer length histogram, loss probability, response time histograms and average buffer length. We first consider a single finite buffer case and then study a network of nodes. The parameters considered in these experiments are taken from [5], [8] to compare results. The experimental study are performed under MAWI traffic trace [14] 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 defined on 80511 elements (bins), see Figure 1. The size of this histogram makes the exact analysis time consuming.

A. Single node

Buffer capacity (10^5 bits)	**Exact**	$L.b$	$U.b$	HBSP	Tancrez U.b
0.5	0.456	0.411	0.469	/	0.491
2	0.425	0.374	0.437	/	0.464
3	0.405	0.371	0.437	/	0.460
4	0.386	0.338	0.407	0.407	0.429
5	0.367	0.333	0.380	0.407	0.429
8	0.317	0.271	0.354	0.326	0.372
9	0.301	0.241	0.330	0.326	0.349
10	0.285	0.238	0.325	0.326	0.347
20	0.182	0.124	0.235	0.155	0.258
30	0.131	0.076	0.173	0.102	0.215
& Ex. Time (s)	3378	4.5	5.3	0.01	0.06

Buffer capacity (10^5 bits)	**Exact**	$L.b$	$U.b$	HBSP	Tancrez U.b
0.5	0.456	0.451	0.460	/	0.471
2	0.425	0.418	0.431	0.464	0.442
3	0.405	0.398	0.412	0.448	0.415
4	0.386	0.379	0.393	0.418	0.398
5	0.367	0.361	0.374	0.403	0.379
8	0.317	0.307	0.326	0.349	0.331
9	0.301	0.294	0.309	0.338	0.317
10	0.285	0.277	0.297	0.326	0.298
20	0.182	0.171	0.194	0.215	0.197
30	0.131	0.120	0.144	0.161	0.148
& Ex. Time (s)	3378	19.5	23	0.02	0.07

TABLE I. BLOCKING PROBABILITIES ($Prob(\mathbf{B})$) FOR BINS = 20 (UP) AND BINS = 100 (DOWN).

We take data unit $\mathbf{D} = 1\,Kb$, and vary the buffer size ($\mathbf{B}$) from $5\,10^4$ $bits$ to $3\,10^6$ $bits$. In Tables I, we present results on blocking probabilities for the number of bins 20 and 100 while Tables II are devoted to the average buffer length. The Table I presents also the execution times of greatest buffer capacity ($\mathbf{B} = 3\,Mb$) for the considered methods. We can see first that the execution times of our bounding method are significantly lower than those of the exact method. Furthermore, they are higher than for the HSBP and the Tancrez methods, but the results are more accurate. In the last column of these tables, we also give the upper bounds obtained by Tancrez's approach [16]. We remark that when the number of bins is 20, the HBSP method does not converge for buffer sizes less than $3\,10^5$, and the results are worse than the Tancrez's and our bounds. We can notice that our bounds are very close to the exact results. Moreover, when the number of bins increases the accuracy of the HBSP and Tancrez method is improved and our bounds become very tight. We can see from Table II that the HBSP method does not provide bounds.

Buffer capacity (10^5 bits)	**Exact**	$L.b$	$U.b$	HBSP	Tancrez U.b
0.5	23301	20557	24960	/	25310
2	92992	85096	100169	/	100830
3	139407	128363	150760	/	150451
4	185845	169664	201530	356563	202097
5	232309	211768	252709	356563	253244
8	372917	335126	409923	510300	410125
9	420127	376429	464449	510300	464813
10	467475	415772	518405	510300	518724
20	938529	785393	1094650	923901	1137680
30	1399670	1098820	1709590	1156000	1778910

Buffer capacity (10^5 bits)	**Exact**	$L.b$	$U.b$	HBSP	Tancrez U.b
0.5	23301	23017	23648	/	23827
2	92992	91711	94531	114894	95229
3	139407	137410	141771	152699	142842
4	185845	183102	189121	228551	190567
5	232309	228704	236561	266694	238432
8	372917	366337	380648	421122	384001
9	420127	412384	429214	460333	433114
10	467475	458388	478066	499711	482500
20	938529	910570	970608	1019260	980361
30	1399670	1343350	1463610	1544850	1479740

TABLE II. AVERAGE BUFFER LENGTH ($\mathbb{E}[H_3]$) FOR BINS = 20 (UP) AND BINS = 100 (DOWN).

B. Tandem Network

We study a tandem queueing network under the MAWI traffic trace. The network is composed of three service nodes with finite buffer capacities of sizes: $\mathbf{B_1} = 2\,Mb$, $\mathbf{B_2} = 1\,Mb$ and $\mathbf{B_3} = 1\,Mb$. The service is deterministic in each queue and it's taken to $110\,Mb/s$, $107.5\,Mb/s$ and $106.5\,Mb/s$ respectively.

In order to accelerate the computation time, we set data unit to $\mathbf{D} = 5\,Kb$. The analysis of the network is performed with two reduced histogram sizes: $n = 100$ and 200 on the input histogram of each queue (see Table III and Table IV).

We study as performance measures for each queue: the blocking probability ($Prob(\mathbf{B})$), the average buffer length ($\mathbb{E}[H_3]$ and throughput (expected value of the departure histogram, $\mathbb{E}[H_5]$). We compute also average transmission delays ($\mathbb{E}[T]$) by using Little's theorem. In the last row, we present the execution time for the whole network analysis. We note that our method is fast even if it is slightly less than the HBSP method. We derive bounds which are more accurate than the results obtained by the HBSP method.

		Exact	L. b	U. b	HBSP
Queue 1	$Prob(B)$	0.1818	0.1737	0.1967	0.2147
	$\mathbb{E}[H_3]\ bits$	938529	916776	977657	1019260
	$\mathbb{E}[H_5]\ bits$	4261850	4253810	4275090	4244160
	$\mathbb{E}[T]$	0.2202	0.2155	0.2287	0.2402
Queue 2	$Prob(B)$	0.1735	0.1551	0.2127	0.1551
	$\mathbb{E}[H_3]\ bits$	488719	465232	530817	468094
	$\mathbb{E}[H_5]\ bits$	4246920	4239950	4257230	4234410
	$\mathbb{E}[T]$	0.1151	0.1097	0.1247	0.1105
Queue 3	$Prob(B)$	0.1635	0.1243	0.2485	$1.18\ .10^{-6}$
	$\mathbb{E}[H_3]\ bits$	505240	465232	594579	39418.4
	$\mathbb{E}[H_5]\ bits$	4240800	4234810	4248200	4234410
	$\mathbb{E}[T]$	0.1191	0.1098	0.1399	0.0093
Execution Time (s)		21868	2.94	2.89	0.13

TABLE III. NUMERICAL RESULTS FOR THE NETWORK USING MAWI TRAFFIC TRACE AND BINS=100.

		Exact	L. b	U. b	HBSP
Queue 1	$Prob(B)$	0.1818	0.1795	0.1915	0.1974
	$\mathbb{E}[H_3]\ bits$	938529	931775	962347	966804
	$\mathbb{E}[H_5]\ bits$	4261850	4259190	4269830	4253710
	$\mathbb{E}[T]$	0.2202	0.2187	0.2254	0.2273
Queue 2	$Prob(B)$	0.1735	0.1683	0.1963	0.1109
	$\mathbb{E}[H_3]\ bits$	488719	481468	513482	424992
	$\mathbb{E}[H_5]\ bits$	4246920	4244590	4253090	4246330
	$\mathbb{E}[T]$	0.1151	0.1134	0.1207	0.1001
Queue 3	$Prob(B)$	0.1635	0.1512	0.2106	0.1229
	$\mathbb{E}[H_3]\ bits$	505240	488868	556369	456314
	$\mathbb{E}[H_5]\ bits$	4240800	4238830	4245290	4242080
	$\mathbb{E}[T]$	0.1191	0.1153	0.1310	0.1076
Execution Time (s)		21868	5.34	6.57	0.15

TABLE IV. NUMERICAL RESULTS FOR THE NETWORK USING MAWI TRAFFIC TRACE AND BINS=200.

C. Tree Network

We consider a tree network model depicted in Fig. 3 with five nodes. We take MAWI traffic trace as input arrival histogram (H_1). Each node is a split element or a finite capacity queue ($\mathrm{B_i} = 10\,\mathrm{Mb}$, $i = 1 \cdots 4$). The service is assumed deterministic and given by $110\,Mb/s$, $75\,Mb/s$, $82.5\,Mb/s$ and $70\,Mb/s$ for each queue respectively.

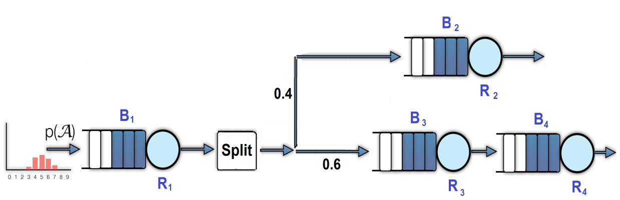

Fig. 3. A tree network with five nodes.

We compute the performance measures of interest under real traffic traces by applying two methods: exact computation (with exact input and service histograms) and our stochastic bounding method developed in [2] (bounding input and service histograms) with number of bins equal to 100. We are interested in the following distributions: queue length (H_3), departure batch (H_5), lower bound of response times H_4^l and upper bound of the response times H_4^u under the FIFO policy, and loss probabilities (P_L).

		$\mathbb{E}[H_3]$	$\mathbb{E}[H_5]$	$\mathbb{E}[H_4^l]$	$\mathbb{E}[H_4^u]$	P_L
Queue 1	$L.\,b$	4232760	4351000	1.5525	/	0.00504
	$U.\,b$	4353820	4353930	/	1.5766	0.00545
Queue 2	$L.\,b$	1081130	1739270	1.1098	/	0.00065
	$U.\,b$	1084180	1740420	/	1.1102	0.00066
Queue 3	$L.\,b$	2099190	2600510	1.3656	/	0.00386
	$U.\,b$	2107100	2602110	/	1.3688	0.00392
Queue 4	$L.\,b$	2480090	2595280	1.5044	/	0.00191
	$U.\,b$	2494890	2596770	/	1.5091	0.00195

TABLE V. BOUNDING RESULTS.

From Table V, we remark that the bounds are provided for each intermediate stage (due to the H-monotonicity of the network elements). We can also see that the results provided by our bounds are very accurate (the bounds are close). The execution times of the bounds takes respectively $6.8\,s$ for the lower bound and $7.2\,s$ for the upper, where the exact computation cannot be obtained after three days of computation. Therefore, the exact measurements can be bounded using the proposed method with relatively small computation complexity.

V. CONCLUSION

We have proposed a novel approach based on stochastic bounds in order to evaluate the performance of queueing networks with general traffic trace measurements. The relevance of our work is first to obtain bounding histograms whose size is adjustable depending on the accuracy and the complexity of the results. Secondly, we prove the stochastic monotonicity of the bounding histograms in order to derive bounds for performance measures of the queueing network. We will now consider the merge network element to study more complex networks. Another objective is to analyze a specific network as a cloud in order to study the impact of the scalability and variation of resource provisioning on the performance. We will also extend the method to del with non stationary flows.

ACKNOWLEDGMENT

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

APPENDIX A
STOCHASTIC COMPARISON

We refer to Stoyan's book ([10]) for theoretical issues and several applications of the stochastic comparison method. We consider state space $\mathcal{G} = \{1, 2, \dots, n\}$ endowed with a total order denoted as $\leq$. Let X and Y be two discrete random variables taking values on $\mathcal{G}$ with probability mass functions given by distribution or histograms H_X and H_Y.

Definition 2: $X \leq_{st} Y \Leftrightarrow \forall i, \sum_{k=i}^{n} H_X(k) \leq \sum_{k=i}^{n} H_Y(k)$

Notice that we use interchangeably $X \leq_{st} Y$ and $H_X \leq_{st} H_Y$. We state here the closure properties of the $\leq_{st}$ order that will be used in this paper [10].

Property 2: If $X \leq_{st} Y$, then i)$-Y \leq_{st} -X$; ii)$f(X) \leq_{st} f(Y)$, for all increasing functions f; iii)$\leq_{st}$ is closed under addition: if $W \leq_{st} Z$, then $X + W \leq_{st} Y + Z$.

Example 3: We use the notation given in Section 2 for the comparison of two discrete random variables X and Y: $E^{H_X} = \{3,\ 4,\ 6,\ 7\}$, $H_X = [0.2,\ 0.35,\ 0.15,\ 0.3]$ and $E^{H_Y} = \{1,\ 2,\ 5,\ 7\}$, $H_Y = [0.3,\ 0.3,\ 0.15,\ 0.25]$. $\mathcal{G} = E^{H_X} \cup E^{H_Y} = \{1, 2, \dots, 7\}$. The distribution H_X (resp. H_Y) is defined on set $\mathcal{G}$ with null probabilities if an element does not belong to E^{H_X} (resp. E^{H_Y}). We can easily verify that $H_Y \leq_{st} H_X$: the probability mass of H_X is concentrated to higher states than that of H_Y.

Definition 3 (st-monotonicity of positive matrices): Let A and B be $n \times n$ matrices (or vectors) with positive coefficients.

- A is st-monotone iff
  $\forall 1 \leq i \leq j \leq n, \ \ \forall k, \ \ \sum_{l=k}^{n} A[i,\, l] \leq \sum_{l=k}^{n} A[j,\, l]$
- A is the st-upper of B (denoted by $B \leq_{st} A$) iff

$$\forall 1 \leq i \leq n, \ \ \forall 1 \leq k \leq n, \ \ \sum_{l=k}^{n} B[i,\, l] \leq \sum_{l=k}^{n} A[i,\, l]$$

Property 3: The st-monotonicity of positive matrices is closed under addition and multiplication by a positive constant.

Let $\{X_i, i \geq 0\}$ (resp. $\{Y_i, i \geq 0\}$) be a time-homogeneous DTMC with probability transition matrices $P1$ (resp. $P2$).

Definition 4 (Comparison and monotonicity of DTMC):

• $\{X_i, i \geq 0\} \leq_{st} \{Y_i, i \geq 0\}$ iff $X_i \leq_{st} Y_i, \forall i$

• $\{X_i, i \geq 0\}$ is st-monotone, if the corresponding transition probability matrix $P1$ is st-monotone.

Proposition 8 (steady-state comparison): Let π_X (resp. π_Y) be the steady-state distribution of the ergodic DTMC $\{X_i, i \geq 0\}$ (resp. $\{Y_i, i \geq 0\}$).

If $\{X_i, i \geq 0\} \leq_{st} \{Y_i, i \geq 0\}$, then $\pi_X \leq_{st} \pi_Y$

In the case of absorbing Markov chains, the time to absorptions can be compared [3].

Proposition 9 (absorption time): Let $\{X_i, i \geq 0\}$ (resp. $\{Y_i, i \geq 0\}$) be a DTMC with one absorbing state which is the smallest one, and T_X (resp. T_Y) is the time to absorption.

If $\{X_i, i \geq 0\} \leq_{st} \{Y_i, i \geq 0\}$, then $T_X \leq_{st} T_Y$

We give here the technical lemma without proof which is used in the proof of Theorem 3.

Lemma 1: Let p and q two non negative vectors with the same size n and the same norm $N1$. Without ambiguity, we use the strong stochastic ordering to compare vectors like p and q. Let γ_i be a discrete distribution of probability for all i between 1 and n. Assume that:

1) $p \leq_{st} q$,
2) for all $1 \leq i \leq n-1$, $\gamma_i \leq_{st} \gamma_{i+1}$,

then there exists a decomposition of $\sum_i \gamma_i(p(i) - q(i))$ into $\sum_{j,k} \lambda_{j,k}(\gamma_j - \gamma_k)$ with $j < k$ and $\lambda_{j,k} > 0$. This decomposition is not unique.

REFERENCES

[1] F. Aït-Salaht, H. Castel Taleb, J. M. Fourneau, and N. Pekergin. Stochastic bounds and histograms for network performance analysis. In *European Performance Engineering Workshop, Italy, Sept. 2013*, LNCS. Springer, 2013.

[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] M. Ben Mamoun, A. Busic, J.-M. Fourneau, and N. Pekergin. Increasing convex monotone Markov chains: theory, algorithms and applications. In *Markov Anniversary Meeting*, pages 189–210. Boson Books, Raleigh, North Carolina, 2006.

[4] J.-M. Fourneau, M. Le Coz, N. Pekergin, and F. Quessette. An open tool to compute stochastic bounds on steady-state distributions and rewards. In *11th International Workshop on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS 2003), Orlando, FL*. IEEE Computer Society, 2003.

[5] E. Hernández-Orallo and J. V. i Carbó. A histogram-based stochastic process for finite buffer occupancy analysis. In *VALUETOOLS*, page 44, 2007.

[6] E. Hernández-Orallo and J. Vila-Carbó. Network performance analysis based on histogram workload models. In *MASCOTS*, pages 209–216, 2007.

[7] E. Hernández-Orallo and J. Vila-Carbó. Web server performance analysis using histogram workload models. *Computer Networks*, 53(15):2727–2739, 2009.

[8] E. Hernández-Orallo and J. Vila-Carbó. Network queue and loss analysis using histogram-based traffic models. *Computer Communications*, 33(2):190–201, 2010.

[9] L. Kleinrock. *Queueing Systems*, volume I: Theory. Wiley Interscience, 1975.

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

[11] P. Pecka, S. Deorowicz, and M. Nowak. Efficient representation of transition matrix in the markov process modeling of computer networks. In T. Czachrski, S. Kozielski, and U. Stanczyk, editors, *Man-Machine Interactions 2*, volume 103 of *Advances in Intelligent and Soft Computing*, pages 457–464, 2011.

[12] M. Schleyer. *Discrete Time Analysis of Batch Processes in Material Flow Systems*. Wissenschaftliche Berichte des Institutes für Fördertechnik und Logistiksysteme des Karlsruher Instituts für Technologie. Univ.-Verlag Karlsruhe, 2007.

[13] 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.

[14] 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.

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

[16] 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.