A Histogram-Based Stochastic Process for Finite Buffer Occupancy Analysis

Enrique Hernandez-Orallo
Departamento de Informatica de Sistemas y Computadores
Universidad Politecnica de Valencia
Valencia, Spain
ehernandez@disca.upv.es

Joan Vila-Carbo
Departamento de Informatica de Sistemas y Computadores
Universidad Politecnica de Valencia
Valencia, Spain
jvila@disca.upv.es

ABSTRACT

This paper proposes to use histograms for characterising network traffic and a simple stochastic process for network performance analysis. The result of this process is the buffer occupancy histogram (queue length distribution) using a finite queue model. From this buffer occupancy histogram we detail how to obtain another interesting performance parameters like cell loss ratio and network delay distribution. The proposed method has been extensively evaluated using real traffic traces. These evaluations show that the model is accurate. Applications of this model are very wide: analysis and prediction of QoS parameters, network dimensioning and provisioning, traffic admission control, etc.

Categories and Subject Descriptors

C.2 [**Computer-Communication Networks**]: Miscellaneous; C.4 [**Performance of Systems**]: Modeling techniques

Keywords

Traffic modeling, Network QoS, Stochastic Analysis

1. INTRODUCTION

Providing Quality-of-Service (QoS) requirements is a key issue in today's network communication. Guaranteeing performance on this communication usually requires some network resource allocation, like bandwidth and buffers. Accurately evaluating these resources is one of the main challenges. This problem has been analysed in the literature using two main approaches [3]: *deterministic* and *statistical* techniques.

Deterministic approaches are based on simplistic workload characterisations and worst-case analyses using network calculus. *Statistical* techniques have been studied in two different contexts. Real-time transmission approaches are extensions or modification of deterministic real-time bounds [9, 25]. The classic *traffic engineering* approaches use statistical techniques such as queuing theory to predict and engineer the behaviour of telecommunications networks such as telephone networks or the Internet.

Systems performance analysis relies mainly on two models: a workload model and a performance model. The *workload model* capture the resource demands and workload intensity characteristics. This model must capture the static and dynamic behavior of the real load and it must be compact and accurate. The *performance model* is used to predict the performance of a system as a function of the system description and the workload model.

In order to properly model a system it is critical to understand the nature of the traffic. There has been a considerable amount of work on traffic characterisation in the literature [2]. In classic networks, the Poisson process has since long been used for call arrivals, because calls are generated independently from each other. However, Internet traffic does not fit into this description. Two pioneering articles [17, 20] showed two properties: i) *self-similarity*: counts of packet arrivals in equally-spaced intervals of time are long-range time dependent and have a large coefficient of deviation, and ii) *heavy tailed*: packet inter-arrival have a marginal distribution that has a longer tail than the exponential. Nevertheless, recent studies has shown that this arrival process tends toward Poisson as load increases [5, 6]. Several distributions were proposed to fit these traffic characteristics. The Pareto and Weibull distributions are often used in order to reflect the heavy-tailed distribution. The self-similarity property can be modeled by an aggregate of multiple heavy-tailed ON/OFF sources. More complex models are based on fractional Gaussian noise (fGN), fractional autoregressive integrate moving average (fARIMA) and wavelets [1].

Several queueing analysis methods has been proposed to model and obtain performance parameters [13]: Markov Modulated Poisson Process (MMPP) [15,18], Switched Batch Bernouilli Process (SBBP) [10] or Discrete Gaussian Models [2]. There are several practical problems with these models. First, we must fit the traffic with the model. Nevertheless, the problem is that when the number of parameters are high the model usually become intractable, so we must use few parameters and this implies losing precision. Second, most of the papers deal with the tail probability (or overflow probability) $P(Q > t)$ rather than the loss probability. Nevertheless, real networks have finite buffer so it is necessary to study the loss probability in finite buffer sys-

Valuetools '07, October 23-25, 2007, Nantes, France

tems ($P_L(x)$). In infinite queue models the loss probability is often approximated as $P_L(x) \approx P(Q > x)$. However, this approximation usually provides an upper bound (sometimes a very pour bound) to the loss probability [14]. Therefore, for network performance evaluation is better to use a model with finite buffer. In [14] the authors presented an estimation for the loss probability based on the tail probability.

An interesting approximation for traffic characterisation are *histogram* based models which describe traffic as a discrete statistical distribution. They extend *deterministic* models, that usually describe traffic with one or two classes (average and peak rate), with a discrete number of classes, quantifying their probability. The model, known as the *Histogram Model* [24] [22], was introduced by Skelly to predict buffer occupancy and loss rate for multiplexed streams. These works use an analysis method based on a M/D/1/N queueing system. The number of ATM cells generated during a frame period is approximated to a Poisson distribution with a given rate λ. For a given video sequence, λ is modelled as a histogram. The buffer occupancy is calculated by solving the M/D/1/N system as a function of λ and then weighting the solutions according to the histogram probabilities. This methods yields good results with a reduced number of cells in the buffer, but the inaccuracy increases with the number of cells. Another histogram-based performance analysis was presented in [19]. The method is based on a modification of the MVA (Mean Value Analysis) algorithm for solving Queueing Network Models (QNM). The drawback of these models is that they solve for each value of the histograms classes independently (using M/D/1/N or MVA), not taking into account the dependencies between the histogram classes. Another more complex approach is the distrete time SBBP/G/1 queue [10]. The SBBP process is characterized by a probability generating function (pgf) and two states. The system is solved only for the infinite capacity queue case. This model has two drawbacks when it is applied to real-traffic modeling: the complexity of definining the pgf from the traffic (the number of states can be very large) and there is no solution for limited capacity buffer.

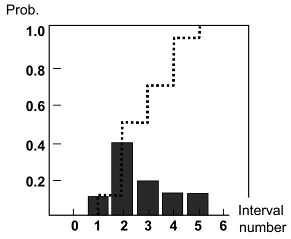
In this paper we propose using histograms as the network traffic model and introduce a stochastic process that works directly with histograms. This stochastic process obtains the queue length distribution as a histogram using a finite queue model. The proposed method does not require approximating traffic to a Poisson distribution nor solving queueing models. The best way to verify the correctness of our model is to compare the predicted results to the ones obtained using a real model with real traffic. These experiments are detailed in section IV and the results are very accurate. These evaluations also analyse the influence of the number of histogram classes on precision, showing that 10 classes are enough to obtain good results.

2. HISTOGRAM TRAFFIC MODEL

Network workloads will be characterised by the number of transmission units produced by a traffic source during a pre-established time period called the *sampling period*. Concretely, let A_k be a discrete random variable representing the amount of work entering the system during the k^{th} sampling interval. Then $\{A_k \mid k \in T\}$ is a discrete stochastic process and it is assumed to be stationary and ergodic. The proposed model characterises variable workloads not as a function of time, but as a discrete statistical distribution. The bit rate during a sampling period is, in general, variable. We assume that traffic arrives at uniform rate in a period but the number of arrivals in a period are independent and have a distribution modeled by a histogram. In other words, if we have N bits in a sampling period, the inter-arrival distribution is deterministic with value $1/N$. The method proposed in this paper is based on defining histogram operators for a stochastic process based on a recurrence relation. The steady state of this stochastic process is the buffer occupancy distribution.

Class number	Interval	Midpoint A	Probability $p(A)$	Cumulative probability
i	$[a_i^-, a_i^+[$	a_i	$p_A(i)$	$p_A^+(i)$
0	$[0, 20[$	10	0	0
1	$[20, 40[$	30	0.1	0.1
2	$[40, 60[$	50	0.4	0.5
3	$[60, 80[$	70	0.2	0.7
4	$[80, 100[$	90	0.15	0.85
5	$[100, 120[$	110	0.15	1.0

(a) Grouped probability distribution



(b) Histogram

Figure 1: A variable workload sample.

A *histogram* is a form of a bar graph representation of a *grouped probability distribution* (gpd) which is a table representing the values of a random variable $\mathcal{A}$ against their corresponding probabilities (or frequencies). The range of values of the variable is, in general, continuous and divided into intervals, also referred to as *classes*. The probabilities of values in an interval are grouped together. Figure 1a shows the grouped probability distribution of a sample workload and Figure 1b shows the corresponding histogram. For convenience, the X-axis of the histogram will show the interval number or its midpoint rather than the interval limits.

All intervals have the same width and are characterised by the following attributes: class number i, interval lower limit a_i^-, interval upper limit a_i^+, interval midpoint a_i, and interval probability $p_A(i)$. Formally: $p_A(i) \equiv P(a_i^- \leq \mathcal{A} < a_i^+)$. Sometimes, it is also useful to include in these attributes the cumulative probability $p_A^+(i) = \sum_0^i p_A(i)$.

The example of Figure 1a corresponds to a workload (A_k) that has been analysed using a sampling period of $T_A = 0.1$ s. The range of the transmission units measured during this sampling period is in $[0, 120[$ kb. This range is divided into $n = 6$ intervals or classes so the *interval length* is $l_A = 20$ kb (from now on we omit the units). Class 0 corresponds to interval $[0, 20[$ whose midpoint is $a_0 = 10$. The probability that the traffic source produces a number of transmission units in this interval is $p_A(0) = 0$. Similarly, class 1 corresponds to interval $[20, 40[$ with probability $p_A(1) = 0.1$, and so on.

A given gpd $\mathcal{A}$ will be usually managed only using two attributes: the set of midpoints a_i and the set of interval probabilities $p_A(i)$, also known as the *probability mass function* or pmf. This will be denoted as:

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

In some contexts, where the midpoints are not relevant, $\mathcal{A}$ will refer only to $\mathcal{A} = [p_A(i) : i = 0 \ldots n-1]$. In the example of Figure 1a: $\mathcal{A} = (A, p(A))$ ($A = [10, 30, 50, 70, 90, 110]$; p(A) = [0, 0.1, 0.4, 0.2, 0.15, 0.15]).

It is important to note that a gpd $\mathcal{A}$ is defined over the traffic domain A while its corresponding pmf $p(A)$ is usually defined over a domain of integers $i = 0 \ldots n-1$ representing the class numbers. The correspondence between some value a in the domain of A and its class number $\hat{a}$ is given by the following equation:

$$\hat{a} = class_A(a) = \left\lfloor \frac{a}{l_A} \right\rfloor \tag{2}$$

For example, given $a = 55$, its corresponding class can be obtained as: $\hat{a} = class_A(55) = \lfloor 55/20 \rfloor = 2$.

It is easy to see that this traffic model has short-range dependence (SRD). In [12] is discussed the impact of the long-range dependence (LRD) on the buffer occupancy and indicated that LRD does not affect the buffer occupancy when the busy periods of the system are not large. The same conclusions were obtained in [21]: short-term correlation have dominant effect on cell loss ratio. More important is to choose the critical time scale (CTS), that is related with the sample period. In [21], the authors considered the buffer behavior at the time-scale beyond the CTS is no significantly affected. Our experiments shows that selecting correctly the sampling period the results are very accurate.

2.1 Histogram Operators

Some important operators on random variables that will be used throughout the paper are introduced next:

- The *mean value* (or expectation) of $\mathcal{X}$ is defined as: $E[\mathcal{X}] = \sum_0^{n-1} p_X(i) \cdot x_i$. The *normalised mean value* of $\mathcal{X}$ is defined as $\hat{E}[\mathcal{X}] = \sum_0^{n-1} p_X(i) \cdot i$. The *maximum* of $\mathcal{X}$ is defined as $M[\mathcal{X}] = \max(x_i : p_X(i) > 0)$ and $\hat{M}[\mathcal{X}] = n - 1$.

- The *scalar multiplication* of $\mathcal{X}$ by a constant c is a new random variable $\mathcal{Y} = c \cdot \mathcal{X}$ where $y_i = c \cdot x_i$ and $p_Y(i) = p_X(i)$ for $i = 0 \ldots n$. Note that variable $\mathcal{Y}$ has the same pmf than $\mathcal{X}$, that is, $p(X) = p(Y)$. Multiplying by a scalar only affects the interval length: $l_Y = c \cdot l_X$.

- The *convolution* of two random variables $\mathcal{X}$ and $\mathcal{Y}$, denoted as $\mathcal{X} \otimes \mathcal{Y}$, is only defined for variables with the same interval length. Let n and m be the number of intervals of $\mathcal{X}$ and $\mathcal{Y}$ respectively. The convolution $\mathcal{X} \otimes \mathcal{Y}$ is a new variable $\mathcal{Z} = (Z, p(Z))$ with $n + m - 1$ intervals, with the same interval length and $p_Z(i) = \sum_{k=0}^{i} p_X(i-k) \cdot p_Y(k)$.

2.2 Histogram classes and precision

One key issue is to determine the number of classes of a histogram. This is in general a trade off between representation economy and precision: with too many intervals, the representation will be cumbersome and histogram processing expensive, since the complexity of algorithms mostly depends on the number of classes but, on the other hand, too few intervals may cause losing information about the distribution and masking trends in data. The experiments in the evaluation section shows that 10 classes are enough to obtain accurate results.

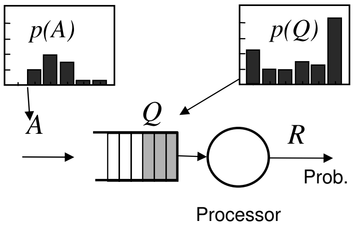


Figure 2: Single node scenario

Another important problem is that histogram processing with a low number of classes is that the result has poor precision. It is paradoxical that these errors occur even if that low number of classes is enough to properly describe a given workload without losing much information. The reason for those inaccuracies seems to be the effect of the low number of classes when using the iterative algorithms. The solution proposed in this paper consists in *overclassing* the histogram which is a transformation for "artificially" increasing the number of classes by splitting each class i with probability $p(i)$ into m classes with the same probability $p(i)/m$. That implies increasing the *zoom factor* of the distribution by m.

3. THE HISTOGRAM BASED PROCESS

This section introduces a stochastic process based on histograms for obtaining the buffer occupancy distribution. Using the buffer occupancy distribution we can calculate several Quality of Service (QoS) parameters.

3.1 Method foundation

The analysis starts by considering a single node as shown in Figure 2. Input traffic is supplied through buffers of finite capacity. These buffers accumulate pending traffic that cannot be transmitted over a sampling period. The system will be said to be *stable* if the pending traffic converges to a finite value. The server discipline is First Come First Served (FCFS) with deterministic (constant) distribution. Using Kendall's notation we are trying to resolve a HD/D/1/K queue where HD stands for Histogram Deterministic Interarrival Distribution.

The queue or buffer length can be expressed using a recurrence equation assuming a discrete time space $T = 0, 1, 2, \ldots$. Let $Q[k]$ be the queue length for period $k \in T$[1]:

$$Q[k] = \phi_0^l(Q[k-1] + A[k] - S[k]) \tag{3}$$

where expression $A[k]$ is the cumulative number of bits that the data source puts into the buffer during the k-th period.

[1] This equation is also detailed in [12]. As stated in the paper there is an easy solution when $l = \infty$. In this case this recurrence equation is known as the Lindey's equation. Nevertheless, when $l < \infty$, they said that the solution was 'complicated' and only present the values for the first 2 iterations

Analogously the service rate $S[k]$ is the number of cumulative bits that the processor removes from the buffer during the same period. Operator ϕ limits buffer lengths so they cannot be negative and cannot overflow the buffer length l. This operator is defined as follows:

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

The service rate can be expressed as a constant r, that is the output rate R multiplied by the period T_A ($r = R \times T_A$). Then, arrivals are spread uniformly over the period and the traffic is processed at constant rate, an arrival rate of $A[k] \le r$ will be served constantly and buffer occupancy is not increased. If $A[k] > r$ the buffer occupancy will increase (up to the queue limit l).

$$Q[k] = \phi_0^l(Q[k-1] + A[k] - r) = \phi_r^l(Q[k-1] + A[k])$$

This recurrence equation is the basis for defining a new stochastic process. We eliminate the time dependence of $A[k]$ using a discrete random variable $\mathcal{A}$ that describes the arrival process. As stated in the previous section, our traffic model assume that traffic is stationary so $\mathcal{A} = \mathcal{A}_k \quad \forall k \in T$. The queue length is converted to a new random variable that depends on the period. This way, the stochastic process is defined as follows:

$$\mathcal{Q}_k = \Phi_r^l(\mathcal{Q}_{k-1} \otimes \mathcal{A}) \qquad (5)$$

where the *bound operator* $\Phi_a^b()$ is defined as the statistical generalisation of the previously defined $\phi_a^b()$ operator. If $\mathcal{X}$ is a random variable with n intervals, then $\mathcal{Y} = \Phi_a^b(\mathcal{X})$ is a random variable with $b+1$ intervals where:

$$p(\Phi_a^b(\mathcal{X})) = [\sum_{i=0}^{a} p_X(i), p_X(a+1), p_X(a+2), \dots,$$

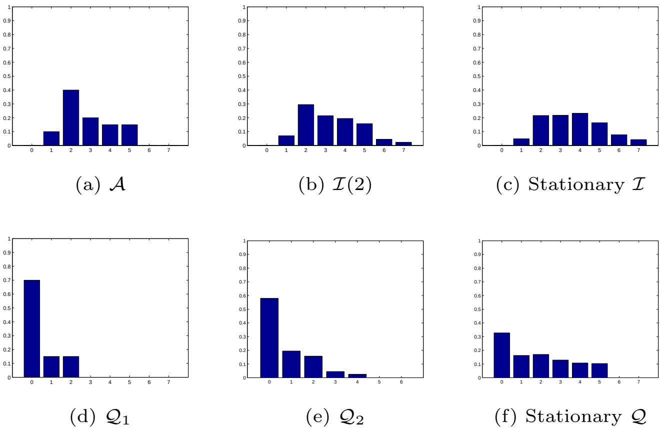
$$p_X(a+b-1), \sum_{i=a+b}^{n-1} p_X(i)] \qquad (6)$$

As an example of how this operator performs, given a random variable $\mathcal{X}$ with histogram $p(X)$ = [0, 0.1, 0.4, 0.2, 0.15, 0.15] (see Figure 1b), then $\mathcal{Y} = \Phi_2(\mathcal{X})$ has $p(Y)$ = [0+0.1+0.4, 0.2, 0.15, 0.15] = [0.5, 0.2, 0.15, 0.15] and $\mathcal{Z} = \Phi_2^2(\mathcal{X})$ has $p(Z)$ = [0+0.1+0.4, 0.2, 0.15+0.15] = [0.5, 0.2, 0.3].

Equation 5 is the definition of a new discrete time stochastic process $\{\mathcal{Q}_k \,|\, k \in T\}$. It will be referenced as the Histogram Buffer Stochastic Process (HBSP). Although the arrival process is deterministic the states of this process are defined using the arrival process, that is, the number of arrivals in a period, and it is assumed to be independent. Consequently, this stochastic process is shown to be a Discrete Time Markov Chain (DTMC) as detailed in the Appendix.

The explanation of this process is provided using the example of Figure 3a. The mean and maximum values of $\mathcal{A}$ are $\hat{E}[\mathcal{A}] = 2.85$, $\hat{M}[\mathcal{A}] = 5$. The process is described using an output rate R=600 kb/s (that is, a service rate of r = 60 kb per sampling period) and a bounded buffer length of 100 kb. In terms of the pmf, those values correspond to $\hat{r} = class_X(r) = 3$ and $\hat{b} = class_X(b) = 5$.

In the first iteration, the pending execution histogram $\mathcal{Q}$ is obtained by summing classes 0..3 of $\mathcal{A}$ (this workload is processed without queueing, assuming a deterministic arrival) and shifting it to the left (Figure 3d): $\mathcal{Q}_1 = \Phi_3(\mathcal{A})$, $p(Q_1) = [0.7, 0.15, 0.15]$.



(a) $\mathcal{A}$ (b) $\mathcal{I}(2)$ (c) Stationary $\mathcal{I}$

(d) $\mathcal{Q}_1$ (e) $\mathcal{Q}_2$ (f) Stationary $\mathcal{Q}$

Figure 3: Buffer Histogram Evolution in HBSP.

The probability that the pending execution is 1 and 2 is 0.15 in each case. Since the buffer computation time is $\hat{b} = 5$ there is no probability of exceeding buffer capacity after the first iteration but, in general, the bound operator establishes an upper limit on the queued workload due to finite buffer length: $\mathcal{Q}_1 = \Phi_3^5(\mathcal{A})$, $p(Q_1) = [0.7, 0.15, 0.15]$.

In the second iteration, the buffer already stores a pending workload of $\mathcal{Q}_1$ and, in addition, a new workload $\mathcal{A}$ arrives. The *cumulative workload* histogram in the buffer after this iteration, is the convolution of the previous histograms (Figure 3b): $\mathcal{I}_2 = \mathcal{Q}_1 \otimes \mathcal{A}$, $p(I_2)$ = [0, 0.07, 0.295, 0.215, 0.1950, 0.1575, 0.0450, 0.0225].

Now the effect of the finite buffer (5 classes) will produce a loss in the cases where there is a probability that the buffer length is greater than 5. For example, for a 6 units length, 5 units are stored into the buffer and the other one is discarded, so this probability has to be added to the probability class 5. Analogously, in the case of 7 units, 5 are accumulated and 2 are discarded. According to this the result of the second iteration is (Figure 3e): $\mathcal{Q}_2 = \Phi_3^5(\mathcal{I}_2)$, $p(\mathcal{Q}_2) = [0.5800, 0.1950, 0.1575, 0.0450, 0.0255]$.

Using an iterative method, the steady state of $\mathcal{Q}$ =(Q, $p(Q)$) is ([10, 30, 50, 70, 90, 110], [0.3275, 0.1625, 0.1699, 0.1291, 0.1077, 0.1033]) (see Figure 3f). Note that the transformation to the histogram class domain produces discretization errors. The effect of this transformation will be studied in detail in the evaluation experiments.

The evolution in time of this stochastic process can be analysed in terms of the mean value of $\mathcal{A}$. When $\hat{M}[\mathcal{A}] \le \hat{r}$ ($r = R \times T_A$), then the pending computation time (or buffer length) is zero because it is easy to prove, from its definition, that $\Phi_{\hat{r}}(\mathcal{A})$ is zero in this case. The case when $\hat{M}[\mathcal{A}] > \hat{r}$ is the most interesting one because, statistical analyses allow arrival rates to exceed occasionally the output rate capacity during transitory overloads and still have a stable system depending on $\hat{E}[\mathcal{A}]$. Two subcases must be considered: using an infinite or a finite buffer. In the infinite buffer case, the system converges to a steady-state pmf iff $\hat{E}[\mathcal{A}] \le \hat{r}$. With a finite buffer, the process always converges because it is always bounded by operator $\Phi_a^b()$.

System evolution for the above considered cases is shown in Figures 4, 5 and 6. The example workload $\mathcal{A} = [0, 0.1, 0.4, 0.2, 0.15, 0.15]$ of Figure 1b has $\hat{E}[\mathcal{A}] = 2.85$ and $\hat{M}[\mathcal{A}] = 5$. Figure 4 shows that the stochastic process converges to a steady state solution for an infinite buffer and a constant service rate $\hat{r} = 3$, since $\hat{E}[\mathcal{A}] \leq \hat{r}$. Figure 5 shows the evolution for the case of an infinite buffer and $\hat{r} = 2$. With $\hat{E}[\mathcal{A}] > \hat{r}$ the system is unstable and the pmf of the buffer length is shifted to the right in each iteration. Figure 6 shows the situation with $\hat{r} = 2$ and a finite buffer $\hat{b} = 30$. It can be seen that the probability of full buffer tends to 1.

Summing up, the method for obtaining the queue occupancy distribution is based on this stochastic process. First, we obtain the histogram for the traffic workload and using this stochastic process we obtain the buffer histogram.

3.2 QoS parameters

Some of the most important performance parameters of a router are delay and loss ratio. This section shows how to obtain these parameters using the histogram method.

The *router delay* D is the time between message arrival at that station and message departure from the station. It is the sum of the *queuing delay* U and the *transmission delay* T. This can be expressed in statistical terms as:

$$\mathcal{D} = \mathcal{U} \otimes \mathcal{T} \tag{7}$$

The *queueing delay* is the time spent by the message waiting for previous buffered messages to be transmitted. In the case of a router with a output rate of R, and a buffer length characterised by a gpd $\mathcal{Q}$ the queueing delay is proportional to $\mathcal{Q}$, so it has the same pmf.

$$\mathcal{U} = \frac{1}{R} \cdot \mathcal{Q} \tag{8}$$

In statistical terms, multiplying $\mathcal{Q}$ by a scalar $\frac{1}{R}$ (*scalar multiplication*) only affects its interval length. Then the interval length of $\mathcal{U}$ is $l_U = l_Q/R$, expressed in seconds.

The *transmission delay* is the time spent by the network interface in processing the message and it is closely related to the transmission speed. Assuming d is the delay for any transmission unit of size lesser than the MTU (Maximum Transmission Unit) and using the same interval length of $\mathcal{U}$ we obtain the class interval as $\hat{d} = class_T(d)$. In statistical terms, $\mathcal{T}$ is a deterministic distribution of the form $\mathcal{T} = (T, p(T)) = ([l_U, 2 \cdot l_U, \ldots \hat{d} \cdot l_U], [t_0, \ldots, t_{\hat{d}}])$ with $t_i = 0$ for $i \leq \hat{d}$ and $t_i = 1$ for $i = \hat{d}$. Then, $\mathcal{D} = \mathcal{U} \otimes \mathcal{T}$ can be calculated convolutioning $\mathcal{Q}$ and $\mathcal{T}$:

$$\mathcal{D} = \mathcal{Q} \otimes [0, \ldots, 0, 1_k] \tag{9}$$

As an example, consider the buffer length $\mathcal{Q}$ obtained for the histogram used in subsection 3.1 using a service rate $R = 600$ kb/s ($\hat{r} = 3$), and assume that the transmission delay is $T_d = 0.1$ s. First, we obtain the interval length of $\mathcal{U}$ as $l_U = l_A/R = 20/600 = 0.0333$ s. The class of d is $\hat{d} = class_U(0.1) = 3$. The router delay is calculated as: $\mathcal{D} = \mathcal{Q} \otimes \mathcal{T} = [0.3275, 0.1625, 0.1699, 0.1291, 0.1077, 0.1033\ ] \otimes [0,0,0,1] = [0, 0, 0, 0.3275, 0.1625, 0.1699, 0.1291, 0.1077, 0.1033]$. Obtaining the midpoints of $\mathcal{D}$ we have $(D, p(D) = ([\ 0.033, 0.066, 0.1, 0.133, 0.166, 0.2, 0.233\ ,0.266, 0.3], [0, 0, 0, 0.3275, 0.1625, 0.1699, 0.1291, 0.1077, 0.1033]\ )$.

The calculus of the loss ratio can be clearly understood using the same example. Consider the stationary cumulative workload pmf ($\mathcal{I} = \mathcal{Q} \otimes \mathcal{A} = [0, 0.0327, 0.1472, 0.1475, 0.1625, 0.1699, 0.1291, 0.1077, 0.0562, 0.0317, 0.0155]$). With a service rate class $\hat{r} = 3$ and a buffer length class $\hat{b} = 5$, from a workload of 10 units, 3 units are sent, 5 units are stored in the buffer and 2 units are lost. Therefore, the *loss pmf* ($\mathcal{C}$) can be obtained by shifting (with accumulation) $\hat{r} + \hat{b} = 3 + 5 = 8$ positions to the right. Using the bound operator: $\mathcal{C} = \Phi_8(\mathcal{I}), \quad p(C) = [0.9528, 0.0317, 0.0155]$.Histogram $\mathcal{C}$ reflects that 0.9528 is the probability of no loss, 0.0317 is the probability that 1 unit is lost and 0.0155 is the probability that 2 units are lost. According to this, the probability of at least 1 unit is lost is the following weighted sum: $\hat{E}[\mathcal{C}] = 0.0317 * 1 + 0.0155 * 2 = 0.0627$. Then, the loss ratio is the proportion between all the units that are lost and the mean of the arrival workload:

$$P_L(b) = \frac{\hat{E}[\mathcal{C}]}{\hat{E}[\mathcal{A}]} = \frac{0.0627}{2.85} = 2.2\% \tag{10}$$

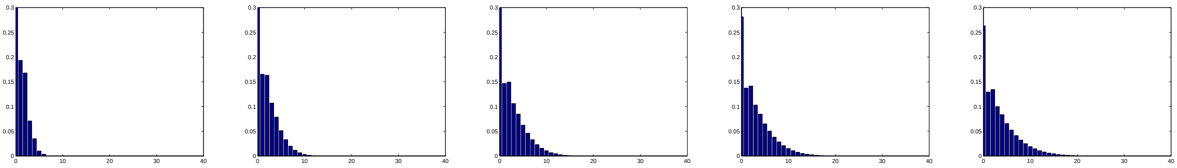
4. EVALUATION

This section presents an evaluation aimed to validate the HBSP model. The best way to validate a performance model is to compare the predicted results (buffer occupancy distribution and loss ratio) with the ones obtained using real traffic traces and real network models. Another key aspect of the experiments is to evaluate the accuracy of the HBSP model, since using discrete variables may introduce important deviations. Accuracy evaluation is estimated by comparing the results of the HBSP obtained analytically, with results obtained through simulation. Finally, the results of the HBSP model are compared to the results of some other methods.

4.1 Real Traffic Experiments

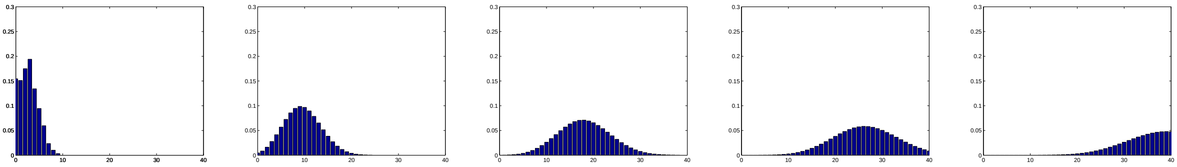
Real traffic experiments are based on the MAWI traffic traces [7] due to their high resolution. Specifically, we took a 1-hour trace of IP traffic corresponding to Jan 09, 2007 12:00 through 13:00 of a 100 Mbps trans-pacific line (samplepoint-F). This traffic trace has 71,545,586 packets with a total size of about 49 Gbytes and an average rate of 109 Mb/s. These MAWI traces are in tcpdump raw format, so we distilled them to obtain a simple file that contains the arrival time (in microseconds) and size (in bytes) of all the packets transmitted during this hour. Using a sampling period of T=40 ms (25 samples per second), the resulting traffic trace has 90,000 frames (see Figure 7a). The arrival load histogram of this traffic using 10 classes is shown in Figure 7b that has $E[\mathcal{A}]$= 4.37 Mb and an interval length $l_A = 0.8$ Mb. In order to evaluate the model an accurate and realistic event driven simulation using these real traffic trace was performed. In each period, the simulation calculates the buffer length and the number of lost packets.

In the first experiment, an infinite buffer was considered and the output rate was set to $R = 180$ Mb/s. The parameter $\hat{r}$ is obtained as $class_A(R \times T_A) = class_A(180\ Mb/s \times 0.04\ s) = class_A(7.2\ Mb)$. For example, for 10 classes $\hat{r} = 9$ and for 100 classes ($l_A = 0.08$ Mb) $\hat{r} = 90$. The HBSP model model was used considering the following arrival workloads: a) a pmf of 10 classes (as shown in Figure 7b) , b) a pmf of 100 classes and c) 10 classes with an overclassing factor of 10. For each simulation a histogram ($\mathcal{Q}^S$) was obtained. The results are in Figure 8a and show that there is loss of accuracy for buffer lengths greater than 0.6Mb. This is mainly due mainly to accuracy in performing convolutions: values



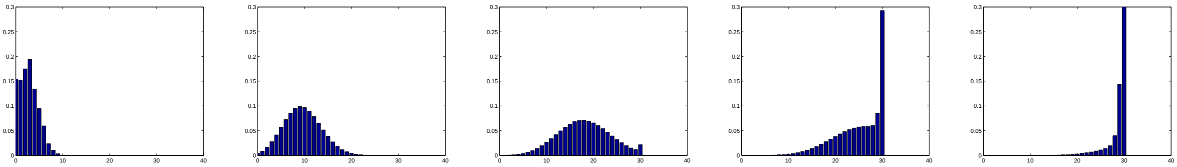
(a) Iteration 2 (b) Iteration 10 (c) Iteration 20 (d) Iteration 30 (e) Iteration 45

Figure 4: Stochastic process with $\hat{E}[\mathcal{A}] = 2.85$, $\hat{r} = 3$ and infinite buffer



(a) Iteration 2 (b) Iteration 10 (c) Iteration 20 (d) Iteration 30 (e) Iteration 45

Figure 5: Stochastic process with $\hat{E}[\mathcal{A}] = 2.85$, $\hat{r} = 2$ and infinite buffer



(a) Iteration 2 (b) Iteration 10 (c) Iteration 20 (d) Iteration 30 (e) Iteration 45

Figure 6: Stochastic process with $\hat{E}[\mathcal{A}] = 2.85$, $\hat{r} = 2$ and a finite buffer of 30 bits

more to the right are the result of accumulating a lot of products. Nevertheless, the *normalized difference* [2] between the simulated histograms and the histogram obtained obtained using the HBSP model is about 1.69×10^{-4}, so they are really close. In the second experiment the output rate was set to aproximately the mean rate $R = 110$ Mb/s and the buffer length was set to $B = 1$ Mb. The parameters $\hat{r}$ and $\hat{b}$ are obtained as $\hat{r} = class_A(110\ Mb/s \times 0.04\ s) = class_A(4.4\ Mb)$ and $\hat{b} = class_A(B) = class_A(1\ Mb)$. For example, for 10 classes $\hat{r} = 5$ and $\hat{b} = 1$, and for 100 classes $\hat{r} = 50$ and $\hat{b} = 12$. The same workloads than in previous experiments were considered. Figure 8b shows the results. Histograms with 100 classes and 10 classes with overclassing exhibit very accurate results. Regarding the loss ratio, the simulation provided a value of 0.0640109 while the HBSP model estimated a value of 0.148567 with 10 classes, 0.0460523 with 100 classes and 0.0501956 with 10 classes and overclassing, that are very close to the simulated one.

Previous experiments used a 1-hour trace for obtaining the histogram, producing very good results. The following experiment uses a 12-hour trace (from 8:00 to 20:00 of the Jan 09, 2007 traces). The rate was set $R = 120$ Mb/s and the buffer length to $B = 1$ Mb. This means using long-term traces instead of short-term traces. Results are still very accurate, as shown in Figure 9. Regarding the loss ratio, the HBSP model predicted a value of 0.01463, while the simulation yielded 0.02259. In summary there is a little loss of accuracy when using long-term traces, as it could be expected, due to information loss in the histogram representation.

Previous experiments were also repeated with different traffic traces (using MAWI traces from another day and hour, the CAIDA OC-48 traces and traces from the NLANR repository), output rates and buffer lengths. Results were very similar to the ones presented here.

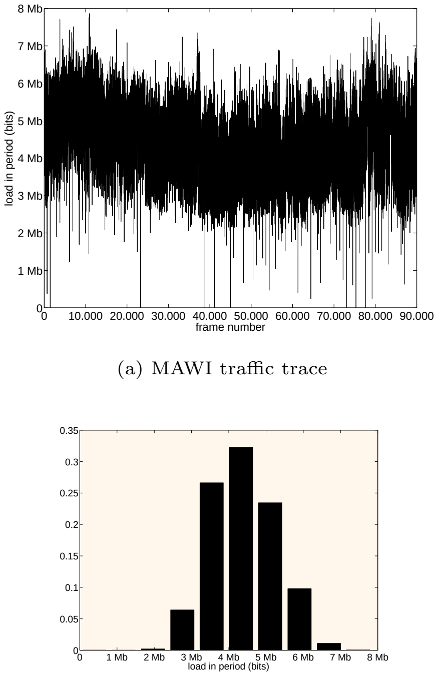
4.2 Accuracy

This subsection is devoted to identify and evaluate factors that may affect the accuracy of the results. The selection of the number of classes and the sample period was based on the results of the following experiments.

The first experiment analyses the relation between buffer length and loss ratio. Loss ratios are calculated for different output rates varying the buffer length between 100 kb and 10 Mb (this corresponds to a maximal queue delay of less than 0.1 s). Results are presented in the form of a loss ratio curve (see Figure 10). The prediction of loss rate using the HBSP model is very accurate, since it is very close to simulations. Best results are obtained when the loss ratio is high. There is a loss of accuracy when the loss ratio is very low.

The second experiment evaluate the relation between the number of classes of a histogram and the accuracy. Previous experiments already showed that accuracy depends on the number of classes and the overclassing (increasing the number of classes by splitting each class into a set of classes with equal probability) factor. Histograms can be a powerful and compact description of the traffic as long as they allow to obtain good accuracy on a low number of classes. The key questions are: how many classes are necessary to get a good accuracy? and, when is necessary to use overclassing in order to obtain good results?. The second experiment reveals the relation between the sample period and accuracy. The experiments uses the same scenarios than previous subsection (the 1-hour MAWI traffic). The output rate was set

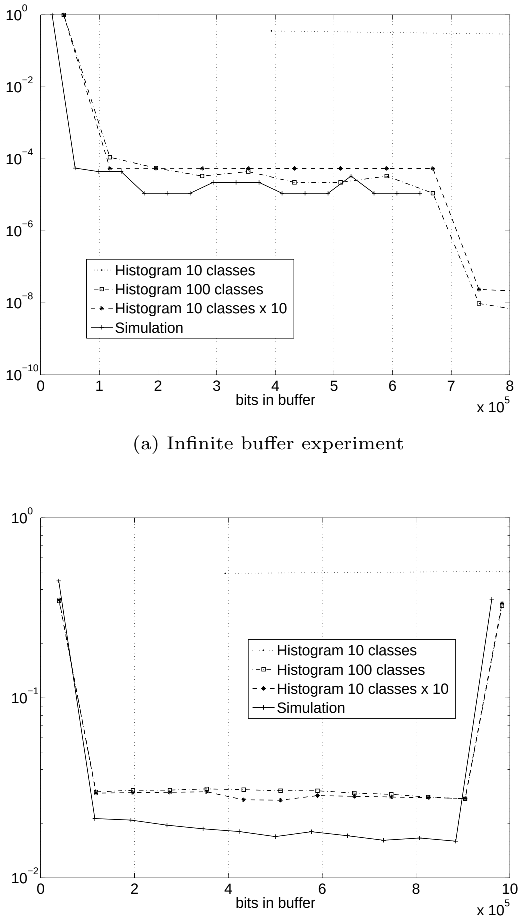
[2] the normalized difference of 2 vectors A=$[a_1, \cdots a_n]$ and B=$[b_1, \cdots b_n]$ is defined as $\sqrt{(a_1 - b_1) + \cdots + (a_n - b_n)}$



(a) MAWI traffic trace

(b) MAWI arrival load histogram

Figure 7: MAWI Traffic traces.



(a) Infinite buffer experiment

(b) Finite buffer experiment

Figure 8: Experiment results using MAWI Traffic traces.

initially to $R = 150$ Mb/s and $B = 1$ Mb was considered.

In the first experiment the number of classes was varied from 6 to 100 and 4 histograms were calculated: the first one using the original histogram with no overclassing and the other 3 using overclassing factors of 5, 10 and 20. The normalized difference between these histograms and the one obtained through simulation is shown in Figure 11a. The loss ratio is compared with the loss ratio of simulations in Figure 11b. Results show that the main effect of overclassing is to smooth the results reducing the original peaks. It can be also seen that there is no significant variation using a overclassing factor greater than 10. Regarding on the number of original classes, it can be seen that accuracy is not greatly improved using more than 15 of 20 classes. For the original histogram, accuracy is better in some cases using more than 60 classes (see Figure 11a) but in some other cases it is worst. Therefore, in the average case, it is better to use overclassing. The final conclusion is that the best results are obtained using 10 to 20 classes with an overclassing factor of 10.

The third experiment analyses the relation between the sampling period and accuracy. This experiment shows the differences between histograms obtained using the HBSP model and simulations varying the sample rate from 0.01 s to 20 s. Results are shown in Figure 12. The best precission is obtained using periods between 20 ms and 200 ms. So, this is the appropiate time scale for this traffic trace.

4.3 Comparison with other methods

This section compares the HBSP model with previous published methods for analysing buffer length and loss ratio. Regarding the calculation of the buffer length, the best known approach is the method introduced by Skelly and Shroff (known as the *Histogram Model* [24] or the *Generalized Histogram Model* [22] and used with few modifications in [23] and [16]). This approximation is based on resolving an M/D/1/N queue for each arrival rate of the histogram. We implemented this method and compared the obtained buffer length histogram with the one obtained using the HBSP model. Figure 13a shows that the differences between the HBSP model and the M/D/1/N method are really high. The results using the M/D/1/N are very bad. The problems with the M/D/1/N is that the buffer curve collapses when the buffer size if high. The results presented in [24] used very low buffer lengths (about 50 cells) so the results were more accurate. Nevertheless when larger buffer (about 500) the buffer curve begins to collapse.

Regarding the loss ratio most of the papers deal with the tail probability (or overflow probability) $P(Q > t)$ rather than the loss probability. In this paper we compare the HBSP method versus the Maximum Variance Asymptotic (MVA) approximation for loss detailed in [14]. The loss probability depending on the buffer size for an output rate $R = 110$ Mb/s is shown in in Figure 13b. The graph shows that the HBSP cell loss curve is more precise than the MVA curve.

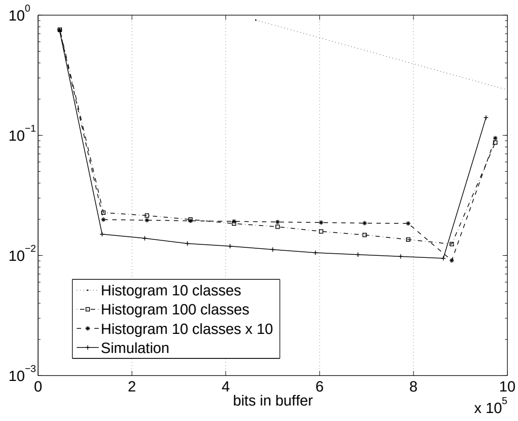


Figure 9: 12-hour finite buffer experiment

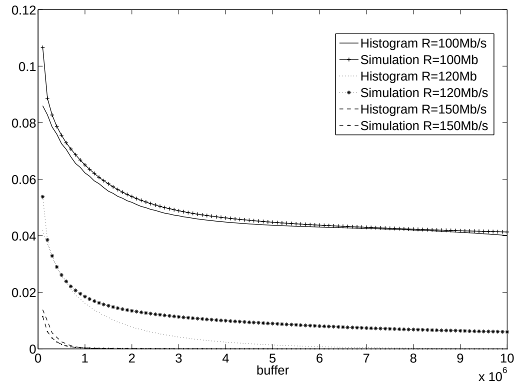
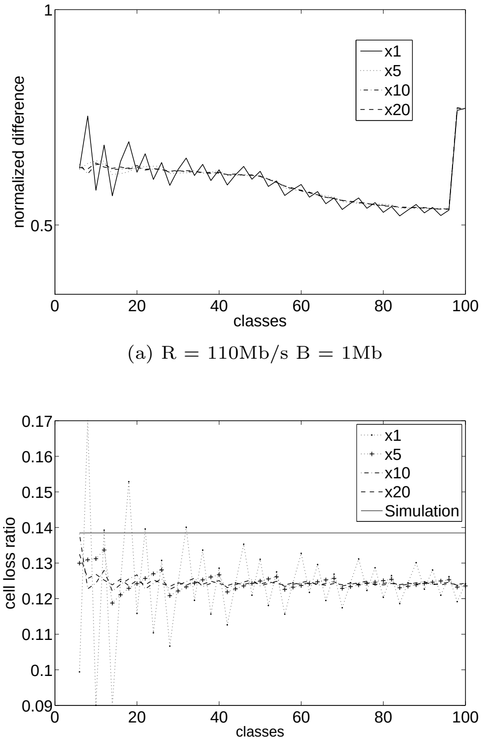


Figure 10: Loss Ratio Curve.



(a) R = 110Mb/s B = 1Mb

(b) CLR R = 100Mb/s B = 1Mb

Figure 11: Classes and precision.

5. APPLICATIONS OF THE MODEL

There is a wide spectrum of applications of the HBSP model. We can obtain the traffic QoS parameters, as loss ratio or node delay using the HBSP model. Using the router delay of the nodes we can obtain the network delay pmf $\mathcal{D}_N$. This pmf is obtained as the sum (convolution) of the node pmfs that traverses a packet:

$$\mathcal{D}_N = \otimes_{i \in path} \mathcal{D}_i \tag{11}$$

This pmf is very useful because we can obtain the mean delay, or for example, the probability that a packet is delayed more than a certain value. For example, if we transmit video or audio, the delay histogram can be useful in the end nodes to adapt their transmissions rates or to configure the buffer in the reception nodes. This information can be used for *admission control* as well.

Another important application is for *traffic provisioning and network configuration.* Optimal provisioning of network resources is crucial for reducing the service cost of network transmission. This is the goal of *Traffic Engineering*: the design, provisioning, performance evaluation and tuning of operational networks. The fundamental problem with provisioning is to have methods and tools to decide the network resource reservation for a given Quality of Service requirements [4]. Therefore, the HBSP method can be very useful for Traffic Engineering.

The HBSP method allows to obtain the load histogram of the nodes of a network. These histograms can be used to configure the network. It also allows to evaluate parameters like the loss ratio (for a given buffer and output ratio), the node delay, the buffer/output ratio needed for a required loss, etc. One important decision that must be taken is the time-scale of the provisioning. The measured traffic can be a long-term trace (daily or weekly traces) or a short-term trace (hourly traces). This depends on the network capability to support dynamical variation in the reservation of the channel resources (for example, an hour) (see [11]).

A great advantage of the HBSP model is the easy implementation of the histograms. Is very easy to capture and

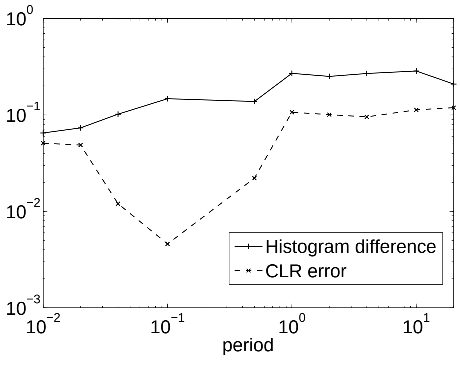
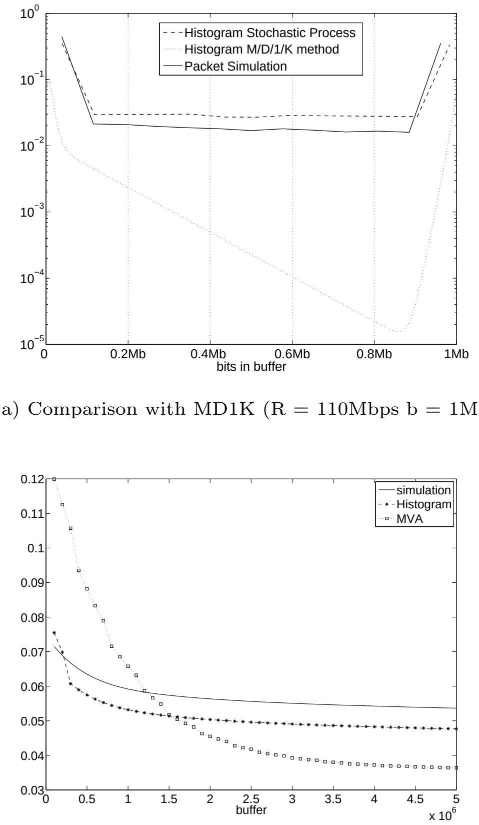


Figure 12: Sample Period and precision.



(a) Comparison with MD1K (R = 110Mbps b = 1Mb)

(b) CLR R = 110Mbps

Figure 13: Comparison with other methods.

store a load histogram with few classes (about 10) in a network node.

6. CONCLUSIONS

This paper deals with a well known problem that can be resumed using the following question by Addie et al. [2]: *'Is there an accurate and useful traffic model in the form of a simple stochastic process which can be described by a small number of parameter and, when fed into a single server queue gives the same performance as a real traffic stream?'.*

This paper presents a new model to answer the question. The model is based in a stochastic process working with histograms (the HBSP model). The result of this stochastic process is a histogram of the buffer distribution. This buffer distribution has an easy solution for the infinite buffer case but it seems to have a complicated solution for the finite buffer case [12]. For this reason, most of the papers obtains the tail probability $P(Q > t)$ using an infinite buffer model and approximate the cell loss using this tail probability. The model presented in this paper is a solution of the finite buffer case. Consequently, from the buffer histogram and the arrival histogram it is easy to obtain the cell loss ratio.

This model is shown to be very accurate. Experiments were performed using synthetic and real-traffic traces. The results show that using a histogram of about 10 classes is enough to obtain good results, so the HBSP model is very compact.

Finally. we can affirm that the HBSP model: (a) *is compact*: about 10 classes are needed to obtain accurate results (b) *is easy to implement*: is simply to sample and store the traffic load of network routers in classes (c) *is accurate*: the results obtained are very accurate (d) *is practical*: from the buffer load histogram we can obtain another useful QoS parameters as loss ratio and delay .

7. ACKNOWLEDGMENTS

This work was developed under grants of the *Generalitat Valenciana* (GV/2007/192) and the Spanish Government (CICYT TIN2005-08665-C03-03).

8. REFERENCES

[1] P. Abry, R. Baraniuk, P. Flandrin, R. Riedi, and D. Veitch. Multiscale nature of network traffic. *IEEE Signal Processing Magazine*, 19(3):28–46, 2002.

[2] R. G. Addie, M. Zukerman, and T. D. Neame. Broadband traffic modeling: Simple solutions to hard problems. *IEEE Communications Magazine*, pages 88–95, Aug. 1998.

[3] C. M. Aras, J. F. Kurose, D. S. Reeves, and H. Schulzrinne. Real-time communication in packet-switched networks. *Proceedings of the IEEE*, 82(41):122–139, Jan. 1994.

[4] D. Awduche and et al. Overview and principles of internet traffic engineering. *RFC*, 3272, May 2002.

[5] J. Cao, W. S. Cleveland, D. Lin, and D. X. Sun. *Nonlinear Estimation and Classification*, chapter Internet Traffic Tends Toward Poisson and Independent as the Load Increases. Springer, 2002.

[6] E. Casilari, J. Cano-Garcia, F. Gonzalez-Canete, and F. Sandoval. Modelling of individual and aggregate web traffic. In *IEEE International Conference on High Speed Networks and Multimedia Communications HSNMC*, pages 84–95, 2004.

[7] K. Cho and et al. Traffic data repository at the wide project. In *USENIX 2000 FREENIX Track*, 6 2000.

[8] J. Dìaz. *Tecnicas Estocasticas para el Calculo del Tiempo de Respuesta en Sistemas de Tiempo Real.* Phd thesis, Universidad de Oviedo, Spain, 2003.

[9] D. Ferrari and D. Verma. A scheme for real-time channel establishment in wide-area networks. *IEEE Journal of Selected Areas Communication*, 8(2):368–379, Apr. 1990.

[10] O. Hassida, Y. Takahashi, and S. Shimogawa. Switched batch bernoulli process (SBBP) and the discrete-time SBBP/G/1 queue with application to statistical multiplexer performance. *IEEE Journal of Selected Areas Communication*, 9(3):394–401, 1991.

[11] E. Hernández-Orallo, J. Vila-Carbó, S. Saez-Barona, and S. Terrasa-Barrena. Provisioning expedited forwarding diffserv channels using multimedia aggregates. In *Euromicro 2004*, 9 2004.

[12] D. P. Heyman and T. V. Lakshman. What are the implications of long-range dependence for VBR-video traffic engineering? *IEEE/ACM Trans. Netw.*, 4(3):301–317, 1996.

[13] D. L. Jagerman, B. Melamed, and W. Willinger. *Stochastic modeling of traffic processes.* CRC Press, Inc., Boca Raton, FL, USA, 1997.

[14] H. S. Kim and N. B. Shroff. On the asymptotic relationship between the overflow probability and the loss ratio. *IEEE/ACM Trans. Netw.*, 9(6):755–768, 2001.

[15] A. Klemm, C. Lindemann, and M. Lohmann. Modeling ip traffic using the batch markovian arrival process. *Performance Evaluation*, 54:149–173, 2003.

[16] S.-K. Kweon and K. G. Shin. Real-time transport of MPEG video with a statistically guaranteed loss ratio in ATM networks. *IEEE Transactions In Parallel and Distributed Computing*, 12(4):387–403, Apr. 2001.

[17] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson. On the self-similar nature of ethernet traffic (extended version). *IEEE/ACM Trans. Netw.*, 2(1):1–15, 1994.

[18] D. M. Lucantoni. The BMAP/G/1 queue: A tutorial. In *Performance Evaluation of Computer and Communication Systems, Joint Tutorial Papers of Performance '93 and Sigmetrics '93*, pages 330–358, London, UK, 1993. Springer-Verlag.

[19] J. Luthi, S. Majumdar, and G. Haring. Mean value analysis for computer systems with variabilities in workload. In *IPDS '96: Proceedings of the 2nd International Computer Performance and Dependability Symposium*, page 32, Washington, DC, USA, 1996. IEEE Computer Society.

[20] V. Paxson and S. Floyd. Wide area traffic: the failure of poisson modeling. *IEEE/ACM Trans. Netw.*, 3(3):226–244, 1995.

[21] B. K. Ryu and A. Elwalid. The importance of long-range dependence of VBR video traffic in atm traffic engineering: myths and realities. In *SIGCOMM '96*, pages 3–14, New York, NY, USA, 1996. ACM Press.

[22] N. B. Shroff and M. Schwartz. Video modeling withing networks using deterministic smoothing at the source. In *IEEE Infocom*, pages 342–349, 1994.

[23] N. B. Shroff and M. Schwartz. Improved loss calculations at an ATM multiplexer. *IEEE/ACM Transactions on Networking*, 6(4):411–21, Aug. 1998.

[24] P. Skelly, M. Schwartz, and S. Dixit. A histogram-based model for video traffic behavior in an ATM multiplexer. *IEEE/ACM Transactions on Networking*, 1(4):446–459, Aug. 1993.

[25] H. Zhang and E. W. Knightly. RCSP and stop-and-go: A comparison of two non-work-conserving disciplines for supporting multimedia communication. *ACM/Springer-Verlag Multimedia Systems Journal*, 4(6), Dec. 1996.

APPENDIX

A. BUFFER ANALYSIS AS A DTMC

In this appendix we show that the $\{A_k \,|\, k \in T\}$ stochastic process is a Discrete-Time Markov Chain (DTMC). Additionally, we can easily obtain the transition probability matrix P. Using this probability matrix we can obtain the values for $\mathcal{Q}_k$. The problem of using DTMC it that is not easy to obtain an analytical solution for the steady state (that is, when $n \to \infty$).

A Discrete-Time Markov Chain is a stochastic process whose probabilities distributions in state j only depends on the previous state i, and not on how the process arrived to state i. It is easy to proof that $\{A_k \,|\, k \in T\}$ is a DTMC. The probability that the buffer in period k takes the value j can be expressed using the buffer probabilities of period $k-1$ as follows:

$$P[\mathcal{Q}_k = j] = \sum_i P[\mathcal{Q}_k = i] \cdot P[\mathcal{Q}_k = j|\mathcal{Q}_{k-1} = i] \qquad (12)$$

The term $p_{ij}(k-1,k) = P[\mathcal{Q}_k = j|\mathcal{Q}_{k-1} = i]$ denotes the probability that the process makes a transition from state i at period $k-1$ to state j at period k. This probability is obtained from the arrival load $\mathcal{A}$ and given that $\mathcal{A}$ is the same in all the periods, then the $p_{ij}(k-1,k)$ does not depend on the period k. Therefore, we can represent $p_{ij}(k-1,k)$ as p_{ij} and Eq.13 is reduced to:

$$P[\mathcal{Q}_k = j] = \sum_i P[\mathcal{Q}_k = i] \cdot p_{ij} \qquad (13)$$

p_{ij} is known as the one-step transition probability. From this we can obtain the transition probability matrix:

$$P = [p_{ij}] = \begin{bmatrix} p_{00} & p_{01} & p_{02} & \cdots \\ p_{10} & p_{11} & p_{12} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \qquad (14)$$

The components of this matrix are easy to obtain using the definition of the stochastic process $\{\mathcal{Q}_n\}$. That is, for obtaining the i-row of P we apply one iteration of the stochastic process using an initial load of one unit in j. For example, the first row is obtained as $\Phi_{\hat{r}}^{\hat{b}}([1,0,0,0,\ldots] \otimes \mathcal{A})$. Using the matrix P we can obtain the pmf of $\mathcal{Q}_k$ as:

$$\mathcal{Q}_k = \mathcal{Q}_1 P^k \qquad (15)$$

Nevertheless, determining the asymptotic behavior (that is, the *steady state*) poses problems. This implies obtaining the steady-state probability vector $\mathbf{v}$ as:

$$\mathbf{v} = \mathbf{v}P \qquad v_j \geq 0, \quad \sum_j v_j = 1 \qquad (16)$$

As the matrix dimensions depends on the $\hat{r}$ and $\hat{b}$ values two cases are studied. When b is infinite (the no-buffer case), this matrix is infinite. This matrix is well studied in [8]. It is shown that the matrix presents a certain regularity in its rows and when the utilization is less than 1 it converges (it is a positive recurrent chain). In the other hand, when $\hat{b}$ is finite the matrix has a finite size of $\hat{b}+1 \times \hat{b}+1$ and it more amenable to work with it. Nevertheless, numerically resolving Eq.16 is not easy even for a little matrix. Therefore, we must use iterative methods as the *power method* or something similar.

Using the example of subsection 3.1, $\mathcal{A}$ =[0, 0.1, 0.4, 0.2, 0.15, 0.15] with $\hat{r}$=3 and $\hat{b}$=5 we obtain the following matrix:

$$P = \begin{bmatrix} 0.70 & 0.15 & 0.15 & 0.00 & 0.00 & 0.00 \\ 0.50 & 0.20 & 0.15 & 0.15 & 0.00 & 0.00 \\ 0.10 & 0.40 & 0.20 & 0.15 & 0.15 & 0.00 \\ 0.00 & 0.10 & 0.40 & 0.20 & 0.15 & 0.15 \\ 0.00 & 0.00 & 0.10 & 0.40 & 0.20 & 0.30 \\ 0.00 & 0.00 & 0.00 & 0.10 & 0.40 & 0.50 \end{bmatrix} \qquad (17)$$

We can obtain the second iteration state as $\mathcal{Q}_2 = \mathcal{Q}_1 P =$ [0.580, 0.195, 0.1575, 0.045, 0.0225]. The steady state probability vector is $\mathbf{v}$ = [0.3275, 0.1625, 0.1699, 0.1291, 0.1077, 0.1033]. This is the pmf of $\mathcal{Q}$ ($p(\mathcal{Q})$).