

# Network Performance Analysis based on Histogram Workload Models (Extended Version)

Enrique Hernández-Orallo, Joan Vila-Carbó

Departamento de Informática de Sistemas y Computadores.

Universidad Politécnica de Valencia. Camino de Vera, S/N. Valencia, Spain.

`ehernandez@disca.upv.es`, `jvila@disca.upv.es`

January 23, 2008

## Abstract

Network performance analysis relies mainly on two models: a workload model and a performance model. This paper proposes to use histograms for characterising the arrival workloads and a performance model based on a stochastic process. This new stochastic process works directly with histograms using a set of specific operators. The result is the buffer occupancy distribution. The loss rate and network delay distribution can be obtained using this distribution. Three traffic models are proposed: the first model (the HD model) is a basic histogram model that is compact and reflects only first order statistics. The other two captures second order statistics: the  $(HD^{(N)})$  model) is based on obtaining several histograms using different time scales and the  $(HD^{(H)})$  model) is based on the Hurst parameter and it is long-range dependent.

This method is evaluated using several real traffic network workloads and the results show that the model is very accurate. The model forms an excellent basis for a decision support tool to allow system architects to predict the behavior of computer networks.

**KEYWORDS:** Network Performance Analysis, Traffic Modelling, Stochastic Analysis.

## 1 Introduction

Network performance relies mainly on routers' queues (buffers). Therefore, queueing analysis plays a crucial role in their design and performance. In the simplest queueing analysis, a traffic is fed into a single-server queue of limited capacity with a given service rate, and we wish to determine information about the queue utilization. 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.

Understanding the nature of the traffic is critical to properly model a computer network system. There has been a considerable amount of work on traffic characterisation in the literature [1]. 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 [2] [3] 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. Recently, studies has shown that this arrival process tends toward Poisson as load increases [4] [5]. 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 [6].

Several queueing analysis methods have been proposed to model and obtain performance parameters [7]: Markov Modulated Poisson Process (MMPP) [8, 9], Switched Batch Bernoulli Process (SBBP) [10] or Discrete Gaussian Models [1]. 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 systems ( $P_L(x)$ ). 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 poor bound) to the loss probability [11]. For this reason, the authors of [11] presented an estimation for the loss probability based on the tail probability. Therefore, for network performance evaluation is better to use a model with finite buffer.

Several papers have proposed the use of histograms as the basis for performance models. The *Histogram Model* [12,13], 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  $\lambda$ . For a given video sequence,  $\lambda$  is modelled as a histogram. The buffer occupancy is calculated by solving the M/D/1/N system as a function of  $\lambda$  and then weighting the solutions according to the histogram probabilities. These 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 [14]. The method is based on a modification of the MVA (Mean Value Analysis) algorithm for resolving Queueing Network Models (QNM). The drawback of these models is that they resolve 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. A more complex approach is the discrete time SBBP/G/1 queue [10]. The SBBP process is characterized by a probability generating function (pgf) and two states. The system is resolved only for the infinite capacity queue case. This model has two drawbacks when it is applied to real-traffic modeling: the complexity of defining the pgf from the traffic (the number of states can be very large) and that there is no solution for limited capacity buffer.

The criteria for a practical performance model can be resumed in three points [15]:

1. the traffic model is defined by a small number of parameters and reflects first and second order statistics,
2. the model will accurately predict the performance parameters,
3. it is easy and amenable to analysis.

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. Our model assumes that the traffic is stationary and the inter-arrival time distribution in a period is constant (deterministic). This basic histogram model will be referred to as the HD model (*Histogram Deterministic inter-arrival distribution*). Based on this basic model we introduce two new models: ii) the  $HD^{(N)}$  model is based on obtaining several histograms using different time scales and iii) the  $HD^{(H)}$  model is based on the Hurst parameter and it is long-range dependent. 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 Traffic Workload Models

In this section we present three workload models based on histograms. The first one is a simple workload definition that only captures first order statistics. The other two are extensions of the first one in order to capture second order statistics (long-range dependence).

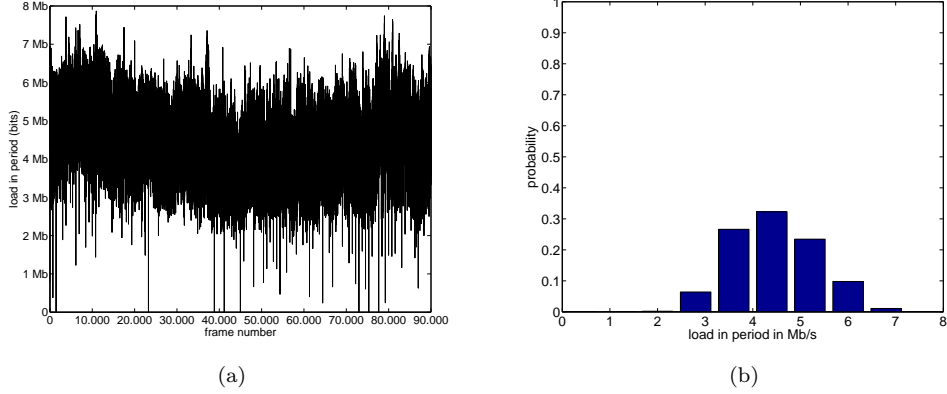


Figure 1: Traffic trace used as illustration. It corresponds to a 1-hour IP traffic trace (a) Number of bits per period (40ms) (b) Arrival load histogram using 10 classes

## 2.1 Basic Histogram Workload Model

Network workloads will be characterized by the number of transmission units produced by a traffic source during a pre-established time period called the *sampling period*<sup>1</sup>. Concretely, let  $A_t$  be a discrete random variable representing the amount of traffic entering a network link during the  $k^{th}$  sampling interval. Then  $\{A_t : t = 1, 2, \dots\}$  ( $\{A_t\}$  for short) is a discrete stochastic process with a state space  $I$  that is the set of integers between 0 and the maximum sample size  $S_{max}$  ( $I = \{0, 1, \dots, S_{max}\}$ ).

A sample realization of this process  $\{a_t : t = 1, 2, \dots\}$  ( $\{a_t\}$  for short) is shown in Figure 1a. This traffic trace was extracted from the MAWI traffic traces [16]. Specifically, we took a 1-hour trace of IP traffic corresponding to Jan 09, 2007 12:00 through 13:00 of a 150 Mb/s trans-pacific line (samplepoint-F). 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 and a state space of  $I = \{0, 1, \dots, 8Mb\}$ .

Working with this huge state space can be cumbersome. So we are going to reduce this state space using  $n$  subintervals or bins. Consequently, if we have a range of  $[0, S_{max}]$  then the interval size will be  $l_A = S_{max}/n$ . Using these intervals we define a *binned process*  $\{\mathcal{A}_t\}$ <sup>2</sup> that have a reduced state space  $I' = \{0 \dots (n-1)\}$ . The correspondence between some value  $a$  in the traffic state space of  $I$  and its subinterval or class number  $i$  (state space  $I'$ ) is given by the following equation:

$$i = class_A(a) = \left\lfloor \frac{a}{l_A} \right\rfloor \quad (1)$$

<sup>1</sup>For convenience we use the bit as the base unit. There is no variation in the precision of the results using another unit (bytes, packets)

<sup>2</sup>Note the calligraphic font style. Along the paper, the calligraphic font style ( $\mathcal{A}$ ) will denote the binned random variable with the reduced state space, and the normal font style ( $A$ ) will denote the original random variable.

The inverse function is defined as the midpoint of the subinterval:

$$a = \text{mid}_A(i) = l_A \cdot i + l_A/2 \quad (2)$$

For our traffic model we assume that the process  $\{A_t\}$  (and consequently the binned process  $\{\mathcal{A}_t\}$ ) is ergodic (and stationary). Being ergodic means that we can estimate the process statistics from the observed values of a single realization or time series  $\{a_t\}$ . Being stationary means that all the *binned* random variables  $\mathcal{A}_t$  have the same marginal distribution. In other words, the distribution does not change on time and we can work with only one random variable  $\mathcal{A}$  that has the same statistical properties of all the random variables  $\mathcal{A}_t$  of the binned process. Assuming that the process is stationary is a simplification necessary for our model. It is a matter of fact that traffic workload is not stationary: in the same day the traffic load at 5 a.m. can be very different than the traffic load at 12 p.m., or the traffic in the weekend is very different than the traffic in a working day. This implies that the precision of the model will depend on the time range of the traffic studied. It is clear that short-term traffic analysis will be more precise than long-term traffic analysis. This topic will be studied in the experiments section.

Another assumption of our model regards to the packet inter-arrival time distribution in a period (the time between the arrivals of successive packets). This distribution can be approximated using several distributions, as detailed in Table 1. In this table we can see the *probability mass function* (pmf) of the distribution and a sample of the temporal distribution. The simplest distributions are the deterministic (uniform) or burst mode. Nevertheless, this distribution is usually modeled using Poisson or Pareto distributions. For example, the MAWI traffic has the inter-arrival distribution of Figure 2a, that clearly shows a heavy-tailed distribution. Nevertheless, if we zoom in this graph and show the distribution for less than 1ms (see Figure 2b) the result is that 75% of the interarrival time is less than 0.05ms. So the deterministic distribution can be used as a simple model of the interarrival distribution.

We assume that traffic arrives at uniform rate in a period but the number of arrivals in a period have a distribution modelled by the binned random variable obtained from the traffic (that is  $\mathcal{A}$ ). In other words, if we have  $N$  packets or bits in a sampling period  $T$ , the inter-arrival distribution is deterministic with value  $N/T$ .

Using a traffic series we can obtain the set of interval probabilities  $p_{\mathcal{A}}(i)$  (that is, the pmf). This will be denoted as:

$$p(\mathcal{A}) = [p_{\mathcal{A}}(i) : i = 0 \dots n - 1] \quad (3)$$

Using the MAWI traffic trace with  $n = 10$  we have the following pmf:  $p(\mathcal{A}) = [0.0003, 0.0002, 0.0021, 0.0641, 0.2663, 0.3228, 0.2345, 0.0980, 0.0110, 0.0005]$  with an interval length ( $l_A$ ) of 0.8 Mb. That is, the probability of  $p_{\mathcal{A}}(0) = 0.0003$  corresponds to the first interval  $[0, 0.8[$  Mb with midpoint 0.4 Mb. The probability of the second interval  $[0.8, 1.6[$  Mb is  $p_{\mathcal{A}}(1) = 0.0002$  and so on. If we plot this pmf we have a kind of histogram of the original traffic trace (see Figure 1b). For convenience,

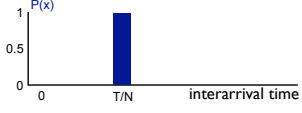
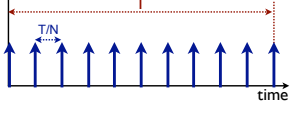
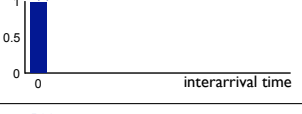
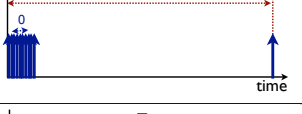
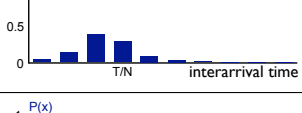
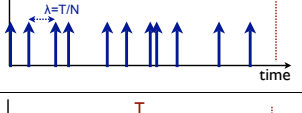
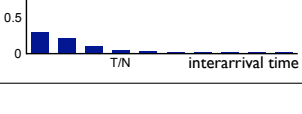
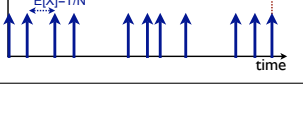
Mode	Inter-arrival distribution	pmf	temporal arrival
Deterministic	$p_X(x) = \begin{cases} 1 & x = \frac{T}{N} \\ 0 & x \neq \frac{T}{N} \end{cases}$		
Burst	$p_X(x) = \begin{cases} \frac{N-1}{N} & x = 0 \\ 0 & 0 < x < T \\ \frac{1}{N} & x = T \end{cases}$		
Poisson	$p_X(x) = Poi(\lambda) \quad \lambda = \frac{T}{N}$		
Pareto	$p_X(x) = k \frac{x_k^m}{x^{k+1}} \quad E[X] = \frac{T}{N}$		

Table 1: Approximations to the inter-arrival distributions. The last column shows an illustrative temporal arrival sequence. Our traffic model is based on a deterministic arrival distribution. Nevertheless, the inter-arrival distribution is usually modeled using Poisson or Pareto distribution.

the X-axis of the histogram will show the midpoint rather than the interval number (that is  $mid_A(i)$ ). This way, we can see the distribution for the original state space.

Summing up, the random variable  $\mathcal{A}$  will usually be managed through just two attributes: the pmf ( $p(\mathcal{A})$ ) and the interval length ( $l_A$ ). Abusing notation we will use the term histogram for referring to the pmf of the random variable  $\mathcal{A}$ .

In short, our traffic model is described using a histogram and a interval length. This model assumes that the traffic is stationary and the inter-arrival time distribution in a period is constant (deterministic). This basic histogram model will be referred to as the HD model (*Histogram Deterministic inter-arrival distribution*).

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

- The *mean value* (or expectation) of  $\mathcal{X}$  is defined as:  $E[\mathcal{X}] = \sum_0^{n-1} p_X(i) \cdot i$ . The *maximum* of  $\mathcal{X}$  is defined as  $M[\mathcal{X}] = n - 1$ . And the *variance* of  $\mathcal{X}$  is defined as:  $Var(\mathcal{X}) = \sum_0^{n-1} (i - E[\mathcal{X}])^2 \cdot p_X(i)$ . In the previous example  $M[\mathcal{A}] = 9$ ,  $E[\mathcal{A}] = 5.06$  and  $Var(\mathcal{A}) = 1.27$ . It is easy to obtain the mean and variation in the traffic state space using the interval size  $l_X$ :  $E[X] = l_X \cdot E[\mathcal{X}] + l_X/2$  and  $Var(X) = (l_X)^2 \cdot Var(\mathcal{X})$ . For example  $E[A] = 0.8 \times 10^6 \cdot 5.06 + 0.8 \times 10^6/2 = 4.45Mb$  and  $Var(A) = (0.8 \times 10^6)^2 \cdot 1.27 = 8.14 \times 10^{11}$ .
- The *scalar multiplication* of  $\mathcal{X}$  by a constant  $c$  is a new binned variable  $\mathcal{Y} = c \cdot \mathcal{X}$  where  $l_Y = c \cdot l_X$  and  $p_Y(i) = p_X(i)$  for  $i = 0 \dots n - 1$ . That is, multiplying by a scalar only affects the interval size.
- The sum of two independent binned random variables  $\mathcal{X}$  and  $\mathcal{Y}$  is defined using the *convolution* operator denoted as  $\mathcal{X} \otimes \mathcal{Y}$ . The convolution is only defined for random variables with the same interval length. Let  $n$  and  $m$  be the number

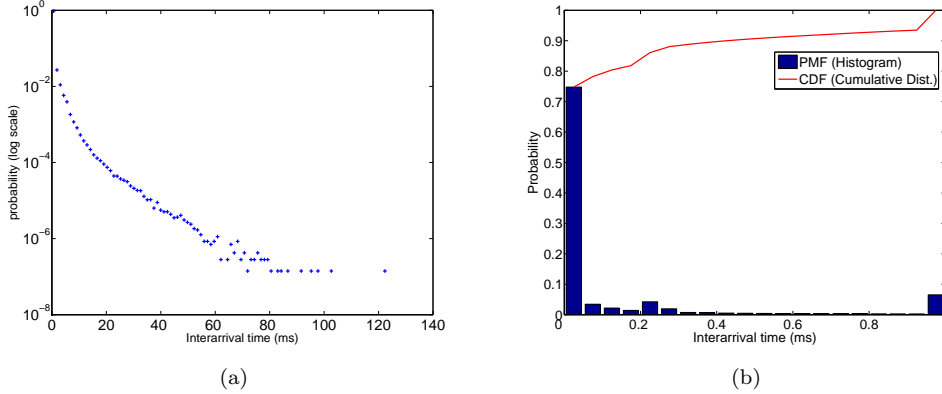


Figure 2: MAWI inter-arrival distribution. (a) Probability in log scale that clearly shows a heavy-tailed distribution (b) Zoom for inter-arrival time less than 1ms We represent the cumulative distribution and the histogram. We can see that more than 75% of the inter-arrival distribution is below 0.05ms.

of intervals of  $\mathcal{X}$  and  $\mathcal{Y}$  respectively and let  $l_X = l_Y$  be their interval length. The convolution  $\mathcal{X} \otimes \mathcal{Y}$  is a new variable  $\mathcal{Z}$  with  $n + m - 1$  intervals, the same interval length  $l_Z = l_X = l_Y$  and  $p_Z(i) = \sum_{k=0}^i p_X(i - k) \cdot p_Y(k)$ .

## 2.2 Second-order Statistics models

In this section we study how the histogram changes depending on the sampling period. Based on these studies we propose two new traffic models that reflects second-order statistics.

Let  $\{A_t\}$  be the stationary process described in the previous section with variance  $\sigma^2$  and autocorrelation function  $r(k) = \text{Cov}(A_t, A_{t+k})/\sigma^2$ . We define  $\{A_t^{(m)}\}$  as the  $m$ -aggregated process of  $\{A_t\}$ , that is obtained by aggregating and averaging the data in  $A_k$  by blocks of size  $m$

$$A_t^{(m)} = \frac{1}{m}(A_{m(t-1)+1} + \dots + A_{mt}) \quad (4)$$

and  $r^{(m)}(k)$  is defined as the autocovariance function of  $\{A_t^{(m)}\}$ . The first effect of the aggregation of the process is to smooth the traffic rate in each period, so the variation is reduced. How this variation changes depending on the factor of aggregation is related to the study of the *self-similarity* of a process.

A *self-similar process* has the property that when aggregated the new process has the same autocorrelation function as the original. That is, the process  $\{A_t\}$  is called *second-order self-similar* with Hurst parameter  $H = 1 - \beta/2$  if

$$\text{Var}(A_t^{(m)}) = \sigma^2 m^{-\beta} \quad \text{and} \quad r^{(m)}(k) = r(k) \quad \forall m = 1, 2, \dots \quad (5)$$

If  $0 < H < 0.5$  the process is short-range dependent (SRD) and if  $0.5 < H < 1$  the process is long-range dependent (LRD).

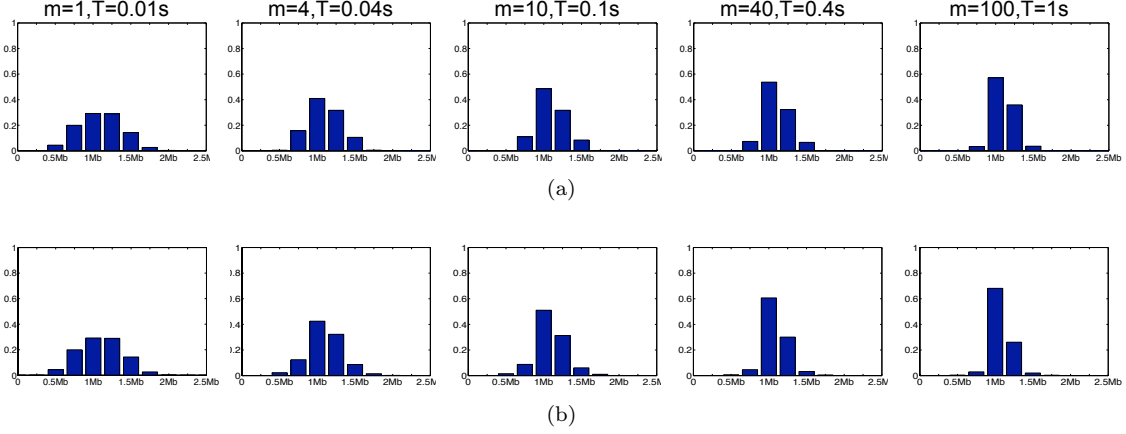


Figure 3: (a) Histograms for aggregated processes  $A_t^{(1)}$ ,  $A_t^{(4)}$ ,  $A_t^{(10)}$ ,  $A_t^{(40)}$  and  $A_t^{(100)}$ . We can see that the distribution shrinks as  $m$  increases revealing the self-similarity property of the traffic. (b) Histograms generated using the Hurst parameter in the  $HD^{(H)}$  traffic model using as base period  $T = 0.01s$ . We can see that the histograms are very similar to the upper ones

Note that this means that the process is *distributionally* self-similar: the processes  $\{A_t\}$  and  $\{A_t^{(m)}\}$  have the same distribution, up to a scaling factor. Figure 3a shows the histograms for several  $m$ -aggregated processes with  $m = 1, 4, 10, 40$  and  $100$  using a base period of  $T = 0.01s$ . We can see that the histograms shrink as  $m$  increases. This clearly denotes the self-similarity property of the MAWI traffic. The variance of the  $m$ -aggregated process can be obtained as:

$$Var(A_t^{(m)}) = m^{2H-2} \cdot Var(A_t) = m^{2H-2} \sigma^2 \quad (6)$$

The graph of the variance  $\log(Var(A_t^{(m)}))$  versus  $\log(m)$  is called the *variance-time plot*. Using this graph we can obtain the Hurst parameter fitting a least-square line with slope  $\beta = 2H - 2$  through the resulting points ignoring those for small  $m$ . In Figure 4 we can see the variance time plot of the MAWI traffic using a base period of 1ms. This traffic has a Hurst parameter of about 0.85, so it is long-range dependent.

For the binned process  $\{\mathcal{A}_t\}$  we can obtain a similar expression for Equation 7. Let  $\dot{\sigma}^2 = Var(\mathcal{A}_t)$ , then

$$Var(A_t) = (l_A)^2 \cdot Var(\mathcal{A}_t) = (l_A)^2 \cdot \dot{\sigma}^2 = \sigma^2$$

We also define the binned processes  $\{\mathcal{A}_t^{(m)}\}$  for  $\{A_t^{(m)}\}$ . We have that,

$$Var(A_t^{(m)}) = (l_{A^{(m)}})^2 \cdot Var(\mathcal{A}_t^{(m)}) = m^{2H-2} \sigma^2$$

Assuming  $l_{A^{(m)}} = l_A$  we have:

$$Var(\mathcal{A}_t^{(m)}) = \frac{m^{2H-2} \sigma^2}{(l_A)^2} = \frac{m^{2H-2} (l_A)^2 \cdot \dot{\sigma}^2}{(l_A)^2} = m^{2H-2} \dot{\sigma}^2 \quad (7)$$



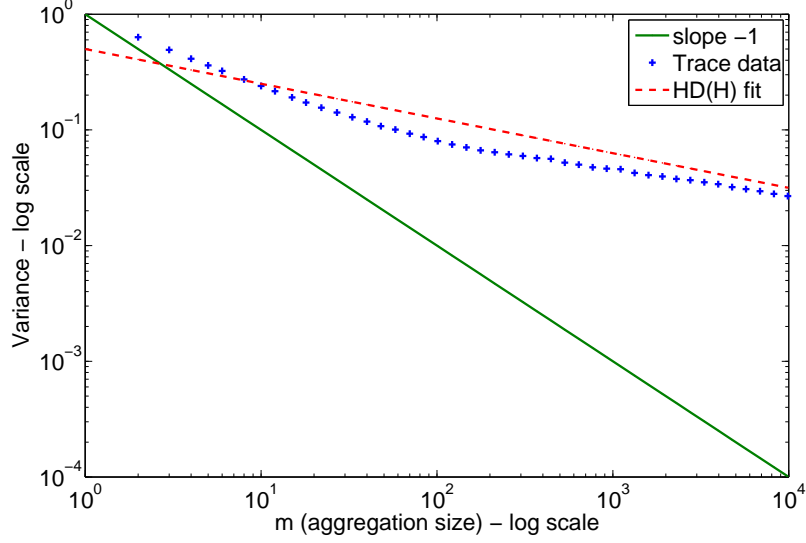


Figure 4: Variance time plot of the MAWI traffic (Base period for  $m=1$  is 1ms). The line with slope  $\beta = -1$  determines if a traffic is SRD or LRD. If the variance plot is above this line the traffic is LRD otherwise is SRD. Thus, the MAWI traffic is LRD with Hurst parameter 0.85

In order to consider the second order statistics we propose two new models that are based on the basic histogram model HD. The first one is based on obtaining the histogram for several time scales. The HD model has only one histogram for a given sample period  $T_A$ . But if we have several histograms for several periods, then the variance will be different on several time scales. For example, we can use the histograms of Figure 3a with sampling periods 0.01s, 0.1s, 1s, 10s and 100s. We have 5 histograms with different variances. Therefore this model can reflect the variance of the traffic on different time scales. Nevertheless, this model of the traffic does not capture the self-similarity characteristics of the traffic.

This model will be referred to as  $HD^{(N)}$ , where  $(N)$  reflects that there are different histograms for several sample periods (or aggregates). One of the problems of this model is the selection of the number and values of the sample periods. If we use too many sample periods the model is too complex. So, we must select only 2 or 3 sample periods in order to make the model compact. For the selection of the sampling period we can use a set of fixed periods (for example 0.01s, 0.1s and 1s) or another approach based on curve fitting.

The second model is based on the Hurst parameter. The idea is to estimate the histograms of the  $m$ -aggregated processes  $\{\mathcal{A}_t^{(m)}\}$  using the histogram of the base process  $\{\mathcal{A}_t\}$ . By Equation 7 we know that  $Var(\mathcal{A}_t^{(m)}) = m^{2H-2}\sigma^2$  and  $l_{A^{(m)}} = l_A$ .

As previously stated, self-similarity means that processes  $\{\mathcal{A}_t\}$  and  $\{\mathcal{A}_t^{(m)}\}$  have the same distribution, up to a scaling factor. In this case the scaling factor for the variance is  $m^{2H-2}$ . The proposed solution is to modify (to scale) the histogram  $\mathcal{A}$  according to this scaling factor. The resulting histogram will have a distribution similar to the original but the variance will be  $m^{2H-2}\sigma^2$ . This process is rather

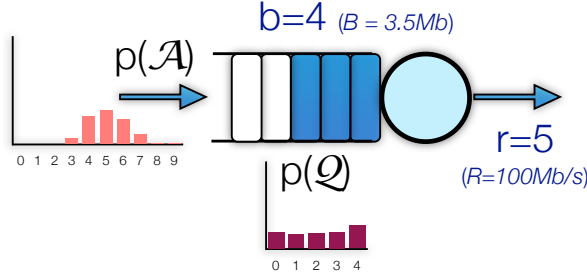


Figure 5: Sample scenario for the basic finite-buffer occupancy analysis. The output rate is  $R=100$  Mb/s (that is, a service rate of 4 Mb per sampling period) and a bounded buffer length of 3.5 Mb. These values corresponds to classes  $r = 5$  and  $b = 4$ . The process obtain the queue histogram  $\mathcal{Q}$  given the arrival histogram  $\mathcal{A}$ .

tricky and it is detailed in the Appendix A. This appendix describes the function  $\hat{\mathcal{A}} = HScale(\mathcal{A}, m, H)$  that calculates the histogram  $\hat{\mathcal{A}}$  and length size for an aggregation of  $m$  and a Hurst parameter  $H$ . For example, we can generate the histograms for the MAWI traffic using a base period of  $T = 0.01s$  and the same aggregations that in the  $HD^{(N)}$  model. The results are shown in Figure 3b and we can see that the histograms are very similar to the real ones. This model will be referred to as the  $HD^{(N)}$  model and its defined with a histogram and the Hurst parameter.

### 3 Network Peformance Model

This section introduces a stochastic process based on histograms for obtaining the buffer occupancy distribution. First, we describe the model using the basic traffic model (HD). Next, we present a simple extension of the basic model for the  $HD^{(N)}$  and  $HD^{(H)}$  traffic models. Finally, using the buffer occupancy distribution we can calculate several Quality of Service (QoS) parameters.

#### 3.1 Basic Finite-Buffer Occupancy Analysis

The analysis starts by considering a single node as shown in Figure 5. 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.

The queue or buffer length can be expressed using a recurrence equation assuming

a discrete time space  $T = 0, 1, 2, \dots$ . Let  $Q[k]$  be the queue length for period  $k \in T$ <sup>3</sup>:

$$Q[k] = \phi_0^b(Q[k-1] + A[k] - S[k]) \quad (8)$$

where expression  $A[k]$  is the cumulative number of bits that the data source puts into the buffer during the  $k$ -th period. Analogously the service rate  $S[k]$  is the number of cumulative bits that the processor removes from the buffer during the same period. Operator  $\phi$  limits buffer lengths so they cannot be negative and cannot overflow the buffer length  $b$ . This operator is defined as follows:

$$\phi_r^b(x) = \begin{cases} 0, & \text{for } x < r \\ x - r, & \text{for } r \leq x < b + r \\ b, & \text{for } x \geq b + r \end{cases} \quad (9)$$

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  $A[k] \leq r$  will be served constantly and buffer occupancy is not increased<sup>4</sup>. If  $A[k] > r$  the buffer occupancy will increase (up to the queue limit  $b$ ).

$$Q[k] = \phi_0^b(Q[k-1] + A[k] - r) = \phi_r^b(Q[k-1] + A[k]) \quad (10)$$

This recurrence equation is the basis for defining a new stochastic process. We eliminate the time dependence of  $A[k]$  using a binned 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:

$$\mathcal{Q}_k = \Phi_r^b(\mathcal{Q}_{k-1} \otimes \mathcal{A}) \quad (11)$$

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

$$p(\Phi_r^b(\mathcal{X})) = \left[ \sum_{i=0}^r p_X(i), p_X(r+1), p_X(r+2), \dots, p_X(r+b-1), \sum_{i=r+b}^{n-1} p_X(i) \right] \quad (12)$$

As an example of how this operator performs, given a random variable  $\mathcal{X}$  with histogram  $p(\mathcal{X}) = [0.0003, 0.0002, 0.0021, 0.0641, 0.2663, 0.3228, 0.2345, 0.0980,$

<sup>3</sup>This equation is also detailed in [17]. As stated in the paper there is an easy solution when  $b = \infty$ . In this case this recurrence equation is known as the Lindey's equation. Nevertheless, when  $b < \infty$ , they said that the solution was 'complicated' and only presented the values for the first 2 iterations

<sup>4</sup>This is the second assumption of our traffic model. The traffic arrives at uniform rate so it can be sent at uniform rate. The Burst mode is easy to model too, all the traffic  $A[k]$  of the period is accumulate first in the buffer and send uniformly. So equation 8 would be  $Q[k] = \phi_0^b(Q[k-1] + A[k]) - S[k]$ . The other models are far more complex to evaluate in this equation

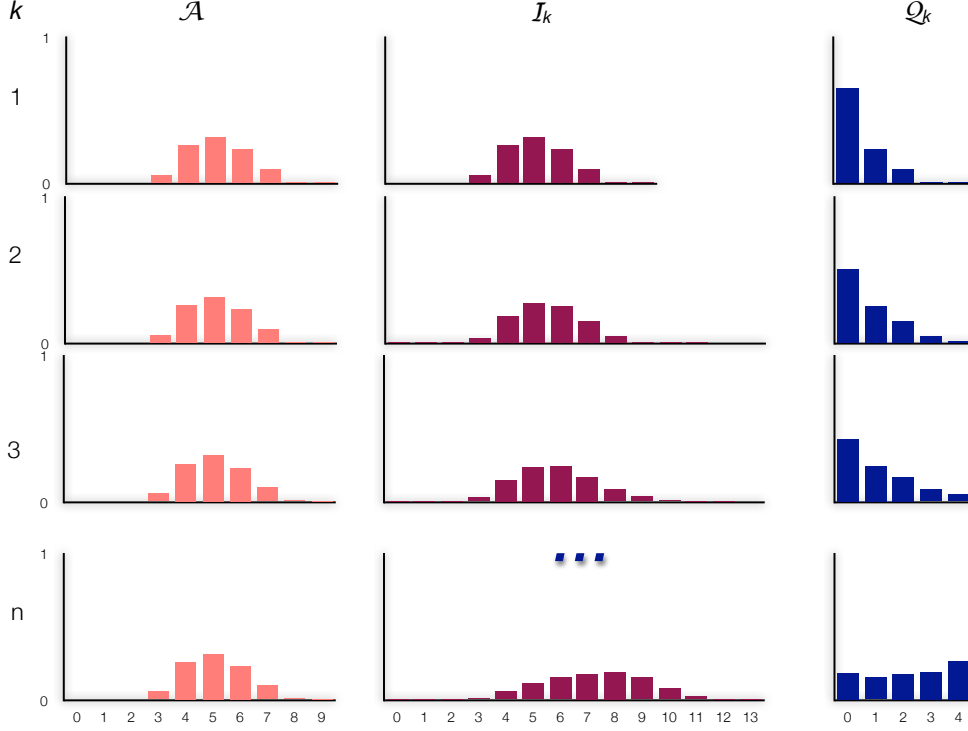


Figure 6: Evolution of the  $\{Q_k\}$  process using the MAWI histogram. The column  $I_k$  shows the result of the convolution between the previous queue and the arrival load  $I_k = Q_{k-1} \otimes \mathcal{A}$ . The row  $Q_k$  shows the evolution of the process. The steady state of the process is shown in the last row.

0.0110, 0.0005] (the MAWI histogram of Figure 1b), then  $\mathcal{Y} = \Phi_5^3(\mathcal{X})$  has  $p(\mathcal{Y}) = [0.0003+0.0002+0.0021+0.0641+0.2663+0.3228, 0.2345, 0.0980, 0.0110+0.0005] = [0.6558, 0.2345, 0.0980, 0.0115]$ .

Equation 11 is the definition of a new discrete time stochastic process  $\{Q_k\}$ . 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 B.

The explanation of this process is provided using the MAWI histogram of Figure 1b. The process is described using an output rate  $R=100$  Mb/s (that is, a service rate of 4 Mb per sampling period) and a bounded buffer length of 3.5 Mb. In terms of the pmf, those values correspond to  $r=class_A(4Mb)=5$  and  $b=class_A(3.5Mb)=4$ .

In the first iteration, the pending execution histogram  $Q$  is obtained by summing classes 0..5 of  $\mathcal{A}$  (this workload is processed without queueing, assuming a deterministic arrival) and shifting it to the left (first row of Figure 6):  $Q_1 = \Phi_5(\mathcal{A})$ ,  $p(Q_1) = [0.6558, 0.2345, 0.098, 0.011, 0.0005]$ . Since the buffer computation time is  $b = 4$ , 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:  $Q_1 = \Phi_5^4(\mathcal{A})$ ,  $p(Q_1) = [0.6558, 0.2345, 0.098, 0.011, 0.0005]$ .

In the second iteration, the buffer already stores a pending workload of  $Q_1$  and,

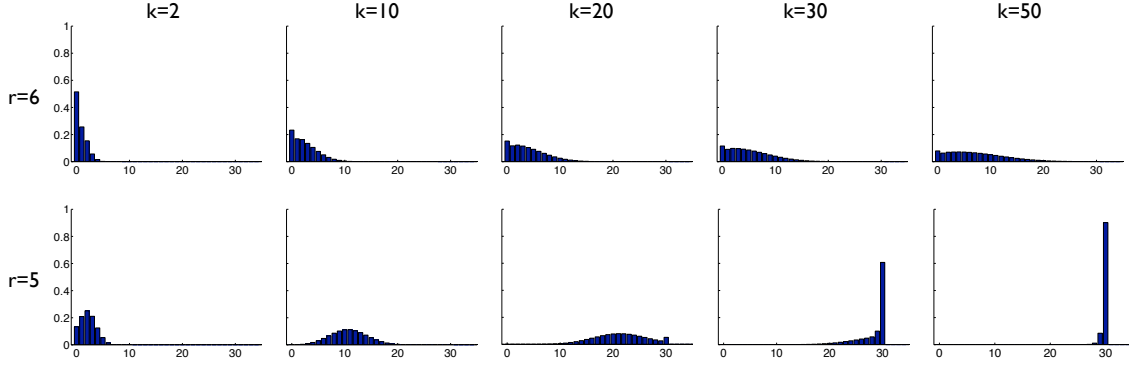


Figure 7: Evolution of the  $\{Q_k\}$  process using the MAWI histogram and a finite buffer  $b = 30$ . The first row shows the evolution when  $E[\mathcal{A}] \leq r$  ( $5.06 < 6$ ): in this case the process converges and the buffer is uniformly used. The second row shows the case when  $E[\mathcal{A}] > r$  ( $5.06 > 5$ ): in this case there is no convergence.

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 (second row of Figure 6):  $\mathcal{I}_2 = \mathcal{Q}_1 \otimes \mathcal{A}$ ,  $p(\mathcal{I}_2) = [0.0002, 0.0002, 0.0015, 0.0426, 0.1899, 0.2804, 0.2563, 0.1539, 0.0569, 0.0153, 0.0024, 0.0002, 0.0000, 0.0000]$ . Now the effect of the finite buffer (4 classes) will produce a loss in the cases where there is a probability that the buffer length is greater than 4. That is, classes 0.5 are accumulated in class 0 (they are sent), classes from 6 to 9 (that is  $b+r$ ) are stored in the buffer and classes greater than 9 are discarded, so this probability has to be added to the probability of class 9. According to this, the result of the second iteration is:  $\mathcal{Q}_2 = \Phi_5^4(\mathcal{I}_2)$ ,  $p(\mathcal{Q}_2) = [0.5147, 0.2563, 0.1539, 0.0569, 0.0179]$ .

Using an iterative method, the steady state is  $p(\mathcal{Q}) = [0.1912, 0.1585, 0.1852, 0.1960, 0.2691]$  (last row of Figure 6). 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  $r$  value and the mean value of  $\mathcal{A}$ . When  $M[\mathcal{A}] \leq r$  ( $r = \text{class}_A(R \times T_X)$ ), the buffer occupancy is zero because it is easy to prove, from its definition, that  $\Phi_r(\mathcal{A})$  is zero in this case. The case when  $M[\mathcal{A}] > r$  is the most interesting one, because statistical analyse allow arrival rates to exceed occasionally the output rate capacity during transitory overloads. With a finite buffer, the process always converges because it is always bounded by operator  $\Phi_r^b(\cdot)$ . Two cases can be considered when  $M[\mathcal{A}] > r$ . If  $E[\mathcal{A}] \leq r$  the buffer is uniformly used and when  $E[\mathcal{A}] > r$  the buffer distribution tends to the right side. System evolution for the above considered cases is shown in Figure 7. We use the MAWI workload that has  $E[\mathcal{A}] = 5.06$  and  $M[\mathcal{A}] = 9$ . The first row shows that the stochastic process converges to a steady state solution for a finite buffer of 30 and a constant service rate  $r = 6$ , since  $E[\mathcal{A}] \leq r$ . The second row shows the situation with  $r = 5$  and a finite buffer  $b = 30$ . 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

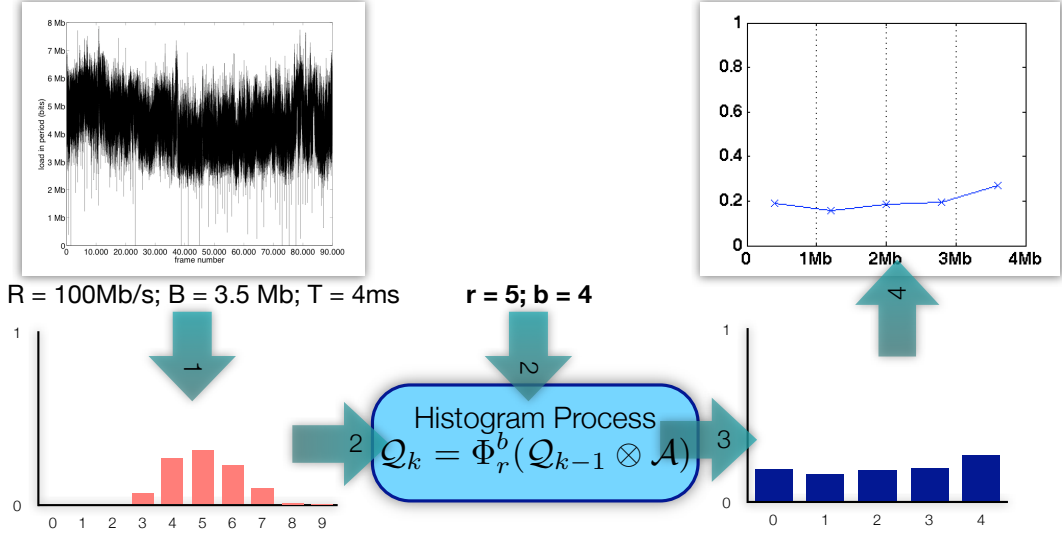


Figure 8: The HBSP Method for obtaining the buffer distribution: (1) The histogram is obtained for a traffic workload using a sampling period  $T_A$ . The interval length ( $l_A$ ) is calculated using the number of classes and the sample size  $S_{max}$ . (2) For a given output rate  $R$  and buffer size  $B$  we obtain the corresponding classes  $b$  and  $r$ . Using the arrival histogram  $\mathcal{A}$  with the  $b$  and  $r$  parameters we can apply the stochastic process. (3) The steady state of this stochastic process is the buffer distribution  $Q$ . Note that the stochastic process only works with classes. (4) From the buffer histogram we obtain the midpoints and represent it using a line graph. Using this kind of graph is better for comparing several data sets.

on this stochastic process. This method will be named the HBSP (Histogram Based Stochastic Process) method and is outlined in Figure 8:

### 3.2 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. If there are too many intervals, the representation will be cumbersome and histogram processing will be expensive, as the complexity of algorithms mostly depends on the number of classes. On the other hand, too few intervals may result in losing information about the distribution and masking trends in data. The experiments shows that 10 classes are enough to obtain accurate results.

Another important problem is that histogram processing using a reduced number of classes produces results that has poor precision. It is paradoxical that these errors occur even if a small number of classes is sufficient to properly describe a given workload without losing a great deal of information. The reason for these inaccuracies seems to be the effect of the low number of classes when using the iterative algorithms. The number of classes for the histogram of the buffer is  $b + 1$ . If  $b$  is very low (for example 2 or 3) we have only 3 or 4 classes for the resulting buffer, so the precision is very low. The solution proposed in this paper consists in *overclassing* the histogram.

Overclassing  $\mathcal{X}$  by a factor  $k$  means applying a transformation  $\Delta_k^+ : \mathcal{X} \rightarrow \mathcal{Y}$  to increase uniformly the number of classes of  $\mathcal{X}$  by splitting each interval of  $\mathcal{X}$  into  $k$  intervals of  $\mathcal{Y}$ . The interval length of  $\mathcal{Y}$  is  $l_Y = l_X/k$  and its *pmf* has a larger number of classes  $m = k \times n$ . Probabilities  $p_Y(j)$  are obtained by linear interpolation of values  $p_X(i)$  in the range  $i : 0 \dots n - 1$  to the range  $j : 0 \dots m - 1$ . This way, if we overclass the arrival histogram by a factor of 10, the classes for  $b$  will also be increased by a factor of 10, and the resulting buffer histogram will have more classes and the results will be more accurate. In practical terms the experiments show that the best results are obtained using over 10 or 20 classes with an overclassing factor of 10.

### 3.3 Buffer analysis for HD<sup>(N)</sup> and HD<sup>(H)</sup> models

In subsection 3.1 we have described a method to obtain the finite buffer distribution histogram using the traffic histogram (that is, the HD/D/1/K queue solution). The solution for the buffer analysis of the HD<sup>(N)</sup> and HD<sup>(H)</sup> models is simple and effective: first, we obtain the buffer histogram for each aggregated histogram and then, using these buffer histograms we calculate the mean histogram.

For the HD<sup>(N)</sup> model we have a set of  $h$  histograms  $\vec{\mathcal{A}} = \{\mathcal{A}^{(m_1)}, \mathcal{A}^{(m_2)}, \dots, \mathcal{A}^{(m_h)}\}$ . Using the HD/D/1/K queue solution we can obtain a set of buffer histograms  $\vec{\mathcal{Q}} = \{\mathcal{Q}^{(m_1)}, \mathcal{Q}^{(m_2)}, \dots, \mathcal{Q}^{(m_h)}\}$ . The resulting histogram  $\mathcal{Q}$  will be a function of this set of histograms ( $\vec{\mathcal{Q}}$ ). We experimented with several functions and the best approach was the mean histogram (for the interval length  $l_Q$  we can use  $l_{Q^{(m_1)}}$ ):

$$p(\mathcal{Q}) = [p_{\mathcal{Q}}(i) = \sum_{j=1}^h p_{\mathcal{Q}^{(m_j)}}(i)/h : i = 0 \dots n - 1] \quad (13)$$

In the HD<sup>(H)</sup> model we do not have a set of histograms, but we can generate them using the *HScale* function described in subsection 2.2. First, we must select the set of values for  $\vec{m} = \{m_1, m_2, \dots, m_h\}$ . Using this set of values we obtain a set of histograms  $\vec{\mathcal{A}} = \{HScale(\mathcal{A}, m_i, H) \mid i = 1 \dots h\}$ . Then, we can follow the method as detailed for the HD<sup>(N)</sup> model. The problem is the selection of the values for  $m$ . This will be studied in the evaluation section.

### 3.4 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 resulting buffer histogram ( $\mathcal{Q}$ ).

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:

$$\mathcal{D} = \mathcal{U} \otimes \mathcal{T} \quad (14)$$

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 an output rate of  $R$ , and a buffer length characterized by random variable  $\mathcal{Q}$  the queueing delay is proportional to  $\mathcal{Q}$ , so it has the same histogram.

$$\mathcal{U} = \frac{1}{R} \cdot \mathcal{Q} \quad (15)$$

In statistical terms, multiplying  $\mathcal{Q}$  by a scalar  $1/R$  (*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  $T_d$  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  $d = \text{class}_U(T_d)$ . In statistical terms,  $\mathcal{T}$  has the following distribution:  $p(\mathcal{T})=[t_0, \dots, t_d]$  with  $t_i = 0$  for  $i \leq d$  and  $t_i = 1$  for  $i = d$ . Then,  $\mathcal{D} = \mathcal{U} \otimes \mathcal{T}$  can be calculated convolutioning  $\mathcal{Q}$  and  $\mathcal{T}$ :

$$\mathcal{D} = \mathcal{Q} \otimes [0, \dots, 0, 1_d] \quad (16)$$

As an example, consider the buffer length  $\mathcal{Q}$  obtained for the histogram used in subsection 3.1 using a service rate  $R = 100$  Mb/s ( $r = 5$ ), and assume that the transmission delay is  $T_d = 20$  ms. First, we obtain the interval length of  $\mathcal{U}$  as  $l_U = l_A/R = 0.8/100 = 8$  ms. The class of  $T_d$  is  $d = \text{class}_U(20 \text{ ms}) = 2$ . The router delay is calculated as:  $\mathcal{D} = \mathcal{Q} \otimes \mathcal{T}$ , that is  $p(\mathcal{D}) = [0.1912, 0.1585, 0.1852, 0.1960, 0.2691] \otimes [0, 0, 1] = [0, 0, 0.1912, 0.1585, 0.1852, 0.1960, 0.2691]$ . Obtaining the midpoints of  $\mathcal{D}$  we have [ 4ms, 12ms, 20ms, 28ms, 36ms, 44ms, 52ms]. That means, for example, that probability for 20ms delay (third interval) is about 0.1912.

The calculus of the loss ratio can be clearly understood using the same example. Consider the stationary cumulative workload pmf ( $\mathcal{I} = \mathcal{Q} \otimes \mathcal{A}$ ,  $p(\mathcal{I}) = [0.0001, 0.0001, 0.0005, 0.0127, 0.0616, 0.1163, 0.1584, 0.1851, 0.1960, 0.1528, 0.0844, 0.0286, 0.0031, 0.0001]$ ). With a service rate class  $r = 5$  and a buffer length class  $b = 4$ , from a workload of 13 units, 5 units are sent, 4 units are stored in the buffer and 4 units are lost. Therefore, the *histogram pmf* ( $\mathcal{C}$ ) can be obtained by shifting (with accumulation)  $r + b = 5 + 4 = 9$  positions to the right. Using the bound operator:

$$\mathcal{C} = \Phi_9(\mathcal{I}), \quad p(\mathcal{C}) = [0.8836, 0.0844, 0.0286, 0.0031, 0.0001] \quad (17)$$

Histogram  $\mathcal{C}$  reflects that 0.8836 is the probability of no loss, 0.0844 is the probability that 1 unit is lost and 0.0286 is the probability that 2 units are lost and so on. Accordingly, the probability of at least 1 unit being lost is the following weighted sum:  $E[\mathcal{C}] = 0.0844 * 1 + 0.0286 * 2 + 0.0031 * 3 + 0.001 * 4 = 0.1513$ . Then, the loss ratio is the proportion between all the units that are lost and the mean of the arrival workload:

$$P_L(b) = \frac{E[\mathcal{C}]}{E[\mathcal{A}]} = \frac{0.1513}{5.06} = 2.99\% \quad (18)$$



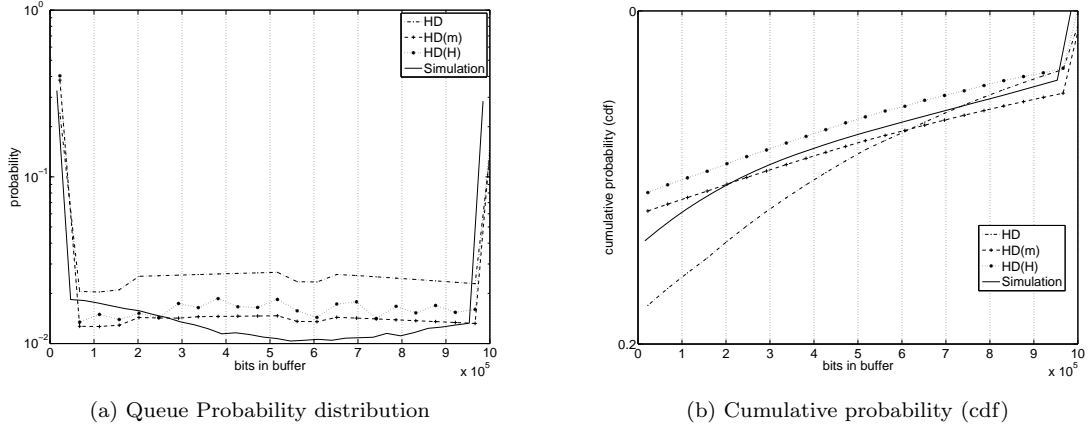


Figure 9: Results of the evaluation using MAWI Traffic trace for  $R = 110$  Mb/s and  $B = 1$  Mb. All the models has 10 classes with an overclassing factor of 10. The HD model has period 40ms, the  $\text{HD}^{(N)}$  model 3 periods (10ms, 50ms, 200ms) and the  $\text{HD}^{(H)}$  model a base period of 10ms, and  $m$  ranging from 1 to 100 (10ms to 1s). The results are very near to the simulated ones. The best results are obtained using the  $\text{HD}^{(H)}$  model

## 4 Evaluation

This section presents an evaluation aimed to validate the models presented in this paper. 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. We compare the results obtained analytically using the 3 models (HD,  $\text{HD}^{(N)}$  and  $\text{HD}^{(H)}$ ) with results obtained through simulation. The traffic used in these evaluations are the MAWI trafic [16] trace described in Figure 1 and the Caida OC-48 traffic [18]. The CAIDA OC48 is a packet header trace of an OC48 link at AMES Internet Exchange (AIX) on Apr 24, 2003 (1 hour) with an average rate of 92 Mb/s.

### 4.1 Evaluating the traffic models

In order to evaluate that the models are accurate and realistic, and 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 we used the MAWI traffic trace (see Figure 1a), 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  $r$  and  $b$  are obtained as  $r = \text{class}_A(110 \text{ Mb/s} \times 0.04 \text{ s}) = \text{class}_A(4.4 \text{ Mb})$  and  $b = \text{class}_A(B) = \text{class}_A(1 \text{ Mb})$ . For example, for 10 classes ( $l_A = 0.8 \text{ Mb}$ )  $r = 5$  and  $b = 1$ , and for 100 classes ( $l_A = 0.08 \text{ Mb}$ )  $r = 50$  and  $b = 12$ . We obtained analytically the buffer histograms using the following traffic models: a) an HD model with 10 classes and period 40ms (as shown in Figure 1b) , b) the  $\text{HD}^{(N)}$  model with three sample periods (10ms, 50ms, 200ms) and, c) the  $\text{HD}^{(H)}$  model with a base period of 10ms, and  $m$  ranging from 1 to 100

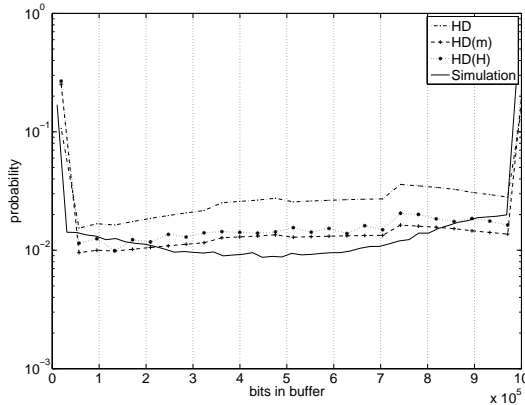


Figure 10: Queue Probability Distribution using the CAIDA OC-48 traffic trace for  $R = 90$  Mb/s and  $B = 1$  Mb. The  $HD^{(N)}$  model is the most accurate.

(10ms to 1s). For the histograms we used 10 classes with an overclassing factor of 10. For comparison, we obtained the buffer histogram using simulation. The queue probability distribution is shown in Figure 9a (note that the y-axis is in log scale). We can appreciate better the differences using the cumulative probability (see Figure 9b). The results for the three models are very accurate, although best results are obtained using the  $HD^{(N)}$  model. Regarding the loss ratio, the simulation provided a value of 0.0640109, while the HD model estimated a value of 0.041934, the  $HD^{(N)}$  model 0.0576173 and 0.0583313 for the  $HD^{(H)}$  model.

For the following experiment we used the CAIDA OC-48 traffic. The output rate was set to  $R = 90$  Mb/s and the buffer length was set to  $B = 1$  Mb. We obtained analytically the buffer histograms using the same sample periods for the traffic models of the MAWI experiments. The results are in Figure 10. We can see that the best results are obtained using the  $HD(m)$  model. Regarding the loss ratio, the simulation provided a value of 0.0494095, while the HD model estimated a value of 0.0344655, the  $HD^{(N)}$  model 0.0436831 and 0.0426831 for the  $HD^{(H)}$  model.

Previous experiments used a 1-hour trace for obtaining the histogram, producing very good results. The following experiment uses the MAWI 12-hour trace (from 8:00 to 20:00 of the Jan 09, 2007 traces). The rate was set to  $R = 120$  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 11. Regarding the loss ratio, the HD model predicted 0.0146371,  $HD^{(N)}$  0.0214733 and  $HD^{(H)}$  0.0202713 while the simulation yielded 0.0275981. 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 a different day and hour, and traces from the NLANR repository), output rates and buffer lengths. Results were very similar to the ones presented here (See [19] for more experiments).

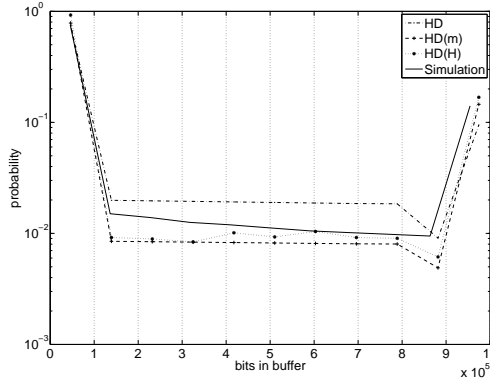


Figure 11: MAWI long-term traffic queue distribution for  $R = 120$  Mb/s and  $B = 1$  Mb. The results of using a 12-hour trace are still very accurate

## 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.

One of the keys aspect of these traffic models is the selection of the sample periods. We analyse the relation between the sampling period and accuracy using the basic HD model. In Figure 12a is the result of obtaining the buffer histogram for several sample periods (base period = 10ms,  $m = \{1, 2, 5, 10, 100\}$ ). We can see that the precision is clearly affected by the sample period. The best results are obtained using  $m = 5$ , that is a sampling period of 50ms. Figure 12b shows the *normalized difference*<sup>5</sup> between the buffer histograms obtained using the HD traffic and that obtained through simulations varying the sample rate from 10ms to 20s. This figure also shows the loss ratio error: that is, the relative error between the loss ratio predicted by our model and the ones obtained in the simulation. The best precision is obtained using periods between 20ms and 100ms.

The following experiment analyses the relation between buffer length and loss ratio for the HD<sup>(N)</sup> traffic model. Loss ratios are calculated for different output rates varying the buffer length between 10 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 13). The prediction of loss rate using the HD<sup>(N)</sup> traffic model is very accurate, since it is very close to simulations.

Regarding on the number of classes several experiments were done varying the number of histogram classes. The key question is: how many classes are necessary to get a good accuracy?. In the following 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

---

<sup>5</sup>the normalized difference of 2 vectors  $A=[a_1, \dots, a_n]$  and  $B=[b_1, \dots, b_n]$  is defined as  $\sqrt{(a_1 - b_1)^2 + \dots + (a_n - b_n)^2}$

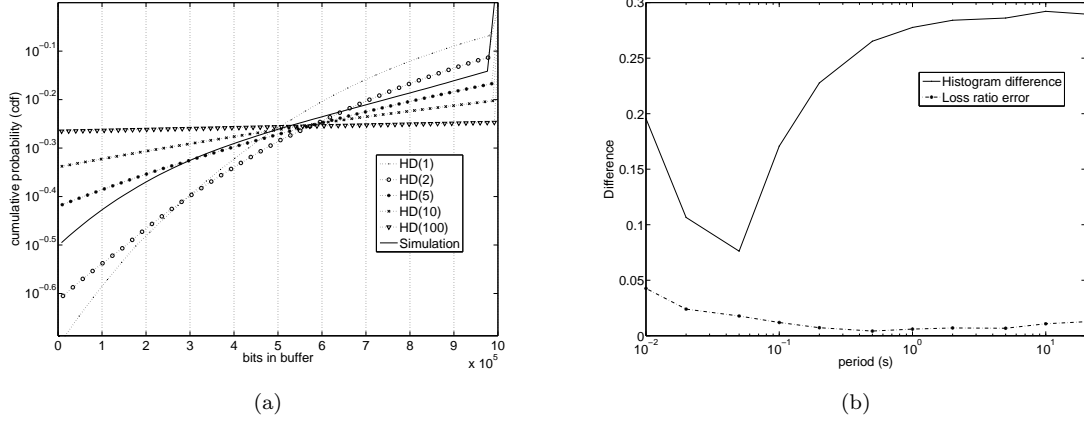


Figure 12: Sample Period and precision. a) Queue Probability distribution for several sample periods (base period = 10ms). We can see the precision of the result depends clearly on the base period. b) This graph shows the Normalized difference between the histograms and the loss ratio error obtained using the HD model and the simulations for sample rate between 10ms and 20s. The best results are obtained for periods between 20ms and 100ms.

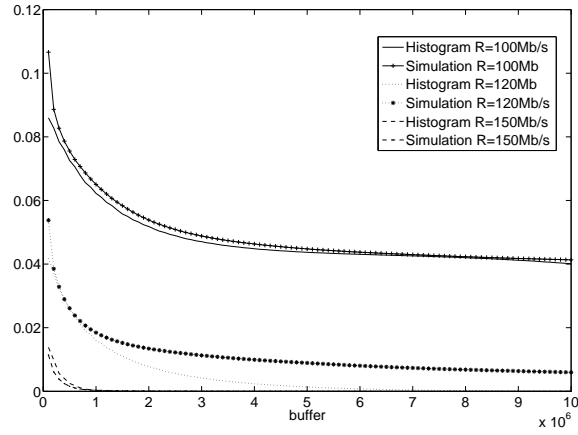


Figure 13: Loss Ratio Curve: the loss ratio is obtained for the the  $HD^{(N)}$  model and is compared to the results obtained in the simulation using different output rates ( $R=100, 110$  and  $120$  Mb/s) and varying the buffer length. The results are very accurate for  $R=100$  Mb/s. For  $R=120$  Mb/s there is a loss of precision in some ranges

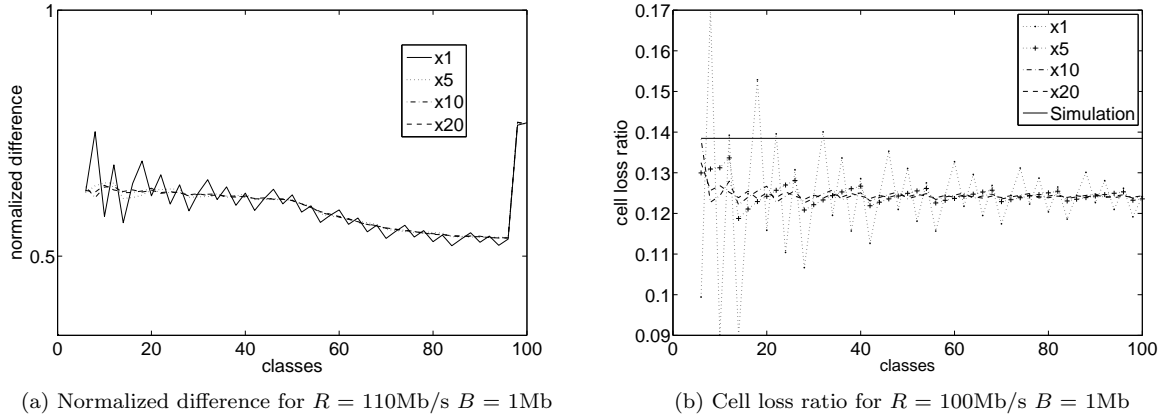


Figure 14: Relation between Number of Classes, overclassing and precision. the number of classes was varied from 6 to 100 and 4 histograms were calculated: 'x1': original histogram, 'x5', 'x10' and 'x20' with overclassing factors of 5, 10 and 20. We can see that accuracy is not greatly improved using more than 15 classes and better results are obtaining using an overclassing factor of 10.

through simulation is shown in Figure 14a. The loss ratio is compared with the loss ratio of simulations in Figure 14b. 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 14a) 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.

### 4.3 Long-range dependence implications on the models

In [17] 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. Similar conclusions were obtained in [20]: 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 [20], the authors considered the buffer behavior at the time-scale beyond the CTS is no significantly affected. The experiments of this paper showed that the best model was  $HD^{(N)}$  that is not strictly LRD. Therefore, the results confirms the results of [20] and [17]. Our experiments shows that when selecting correctly the sampling period the results are very accurate.

Another question is the selection of the model. Each model has its pros and cons:

- HD model: this model is simple and compact and the results are still very accurate. It is a good approximation.

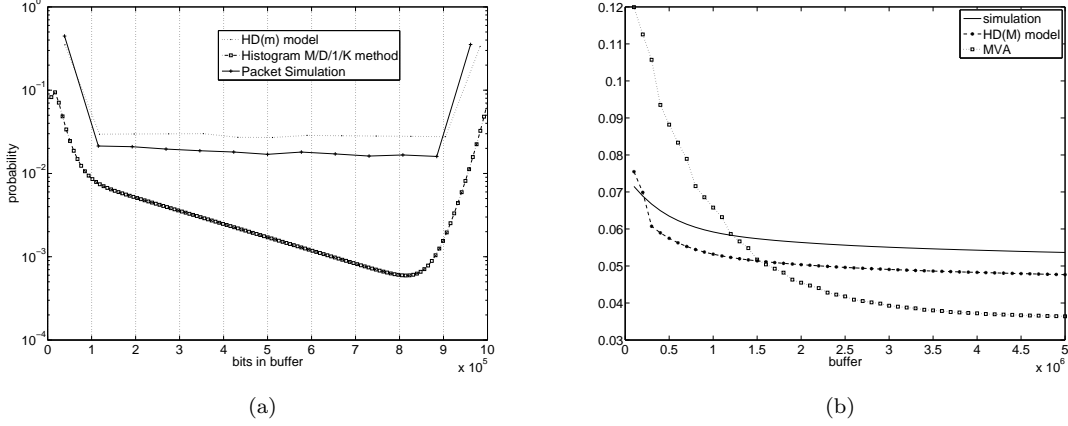


Figure 15: Comparison with other methods. a) Comparison of the buffer histograms obtained using our  $HD^{(N)}$  model and the MD1K approach by Skelly and Shroff ([12] for  $R = 110\text{Mb/s}$   $B = 1\text{Mb}$ . We can see that the MD1K approach collapses when the buffer is large b) Comparison of the cell loss ratio obtained using the Maximum Variance Asymptotic (MVA) method versus the  $HD^{(N)}$  model results  $R = 110\text{M/s}$ . The graph shows a better precision of the  $HD^{(N)}$  model

- $HD^{(N)}$ : it is the more accurate model but it is not so compact. We must obtain several histograms for several periods, so it can be cumbersome.
- $HD^{(H)}$ : this model is still compact (we only need one histogram and the Hurst parameter) and the results are very good.

#### 4.4 Comparison with other methods

This section compares the model with previously published methods for analyzing buffer length and loss ratio. To simplify we only compare the results of the  $HD^{(N)}$  model. Regarding the calculation of the buffer length, the best known approach is the method introduced by Skelly and Shroff (known as the *Histogram Model* [12] or the *Generalized Histogram Model* [13] and used with few modifications in [21] and [22]). 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  $HD^{(N)}$  model. Figure 15a shows that the differences between the HBSP model and the M/D/1/N method are large. The results using the M/D/1/N are imprecise. The problems with the M/D/1/N is that the buffer curve collapses when the buffer size is high. The results presented in [12] 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 [11]. In Figure 15b we can see the loss probability

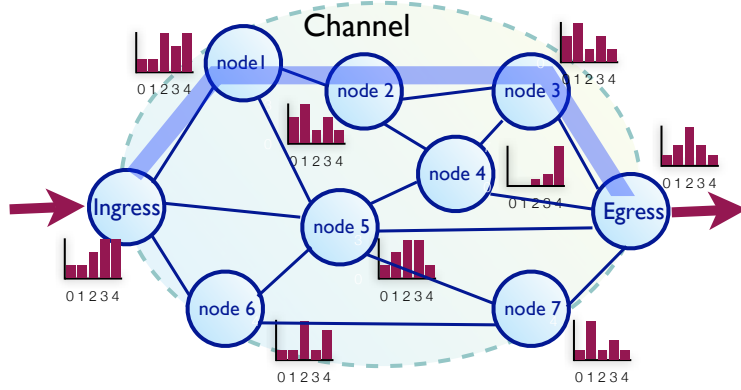


Figure 16: Sample network with the load histograms of the nodes. This can be seen as a visual monitorization of the network load. With this information we can obtain the loss ratio, the end-to-end delay distribution for a given path.

depending on the buffer size. The graph shows that the  $HD^{(N)}$  cell loss curve is more precise than the MVA curve.

## 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 a packet traverses:

$$\mathcal{D}_N = \bigotimes_{i \in path} \mathcal{D}_i \quad (19)$$

For example, using Figure 16 we can obtain the end-to-end delay distribution for the path marked with blue. 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 [23]. 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. For example, if we have a network like the one

presented in Figure 16 we can see that node 4 is highly loaded and we can decide to increase its resource or to change some route in order to reduce the load in node 4.

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 [24]).

A great advantage of the HBSP model is the easy implementation of the histograms. Is very easy to capture and store a load histogram with few classes (about 10) in a network node.

## 6 Conclusions

This paper introduces three traffic models and a performance model to obtain the finite buffer occupancy distribution. The performance model is based on a stochastic process working with histograms for resolving the HD/D/1/K queue. 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 [17]. 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. From the buffer histogram and the arrival histogram it is easy to obtain the cell loss ratio.

Three models are described in this paper. The first model (HD), is a basic histogram model that is compact and short-range dependent. The second one (the  $\text{HD}^{(N)}$  model) is based on obtaining several histograms using different time scales. The third one (the  $\text{HD}^{(H)}$  model) is based on the Hurst parameter and it is long-range dependent. Experiments were performed using several real-traffic traces. The best results are obtaining using the  $\text{HD}^{(N)}$  model although it can be cumbersome. So, the best approach is the  $\text{HD}^{(H)}$  model, that is compact and it is very precise.

There is a wide spectrum of applications of this model. We can obtain the traffic QoS parameters, as loss ratio or node delay. The delay histogram 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.



## 7 Acknowledgments

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

## References

- [1] Ronald G. Addie, Moshe Zukerman, and Timothy D. Neame. Broadband traffic modeling: Simple solutions to hard problems. *IEEE Communications Magazine*, pages 88–95, August 1998.
- [2] Will E. Leland, Murad S. Taqqu, Walter Willinger, and Daniel V. Wilson. On the self-similar nature of ethernet traffic (extended version). *IEEE/ACM Trans. Netw.*, 2(1):1–15, 1994.
- [3] Vern Paxson and Sally Floyd. Wide area traffic: the failure of poisson modeling. *IEEE/ACM Trans. Netw.*, 3(3):226–244, 1995.
- [4] Jin Cao, William S. Cleveland, Dong Lin, and Don X. Sun. *Nonlinear Estimation and Classification*, chapter Internet Traffic Tends Toward Poisson and Independent as the Load Increases. Springer, 2002.
- [5] E. Casilari, J.M. Cano-Garcia, F.J. 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.
- [6] 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.
- [7] David L. Jagerman, Benjamin Melamed, and Walter Willinger. Stochastic modeling of traffic processes. pages 271–320, 1997.
- [8] David 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.
- [9] Alexander Klemm, Christoph Lindemann, and Marco Lohmann. Modeling ip traffic using the batch markovian arrival process. *Performance Evaluation*, 54:149–173, 2003.
- [10] On Hassida, Yoshitaka Takahashi, and Shinsuke Shimogawa. Multiscale nature of network traffic. *Switched Batch Bernoulli Process (SBBP) and the Discrete-time SBBP/G/1 queue with Application to Statistical Multiplexer Performance*, 9(3):394–401, 1991.
- [11] Han S. Kim and Ness B. Shroff. On the asymptotic relationship between the overflow probability and the loss ratio. *IEEE/ACM Trans. Netw.*, 9(6):755–768, 2001.
- [12] 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, August 1993.

- [13] Ness B. Shroff and Mischa Schwartz. Video modeling withing networks using deterministic smoothing at the source. In *IEEE Infocom*, pages 342–349, 1994.
- [14] 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.
- [15] Moshe Zukerman, Timothy D. Neame, and Ronald G. Addie. Internet traffic modeling and future technology implications. In *IEEE Infocom*, 2003.
- [16] K Cho and et al. Traffic data repository at the wide project. In *USENIX 2000 FREENIX Track*, 6 2000.
- [17] Daniel 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.
- [18] DatCat. Caida oc48 traces 2003-04-24: download in <http://imdc.datcat.org>.
- [19] E. Hernandez and J.Vila. A stochastic analysis of network traffic based on histogram workload modelling. Technical Report UPV-DISCA-06-09, September 2006. Download in [http://www.disca.upv.es/enheror/pdf/TR\\_DISCA\\_06\\_09.pdf](http://www.disca.upv.es/enheror/pdf/TR_DISCA_06_09.pdf).
- [20] Bong K. Ryu and Anwar 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.
- [21] Ness B. Shroff and M. Schwartz. Improved loss calculations at an ATM multiplexer. *IEEE/ACM Transactions on Networking*, 6(4):411–21, August 1998.
- [22] Seok-Kyu Kweon and Kang 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, April 2001.
- [23] D. Awduche and et al. Overview and principles of internet traffic engineering. *RFC*, 3272, May 2002.
- [24] Enrique Hernández-Orallo, Joan Vila-Carbó, Sergio Saez-Barona, and Silvia Terrasa-Barrena. Provisioning expedited forwarding diffserv channels using multimedia aggregates. In *Euromicro 2004*, 9 2004.

## A Method for scaling and histogram

This appendix describes how to obtain the m-aggregated histogram  $\hat{\mathcal{A}}$  with a variance according to the scaling function  $m^{2H-2}$ . The function that obtains this new histogram will be named *HScale*. That is,  $\hat{\mathcal{A}} = HScale(\mathcal{A}, m, H)$  generates a new histogram  $\hat{\mathcal{A}}$  with variance  $m^{2H-2} \cdot Var(\mathcal{A})$ , pmf  $p(\hat{\mathcal{A}}) = [p_{\hat{\mathcal{A}}}(i) : i = 0 \dots n - 1]$  and interval length  $l_{\mathcal{A}} \cdot m$ .

As the variance of  $\mathcal{A}$  must be reduced we are going to shrink the histogram. For example, if we have the histogram  $p(A) = [0.2, 0.6, 0.2]$  that has mean 1 and variance 0.4, and we want to decrease its variance to 0.3 the new histogram will be  $[0.15, 0.7, 0.15]$ .

The goal is to shrink the histogram reducing the values at the edges and accumulating this reduction for the classes in the mean. Assume that the histogram has  $n$  classes and the classes in the mean are  $h = \lfloor E(\mathcal{A}) \rfloor$  and  $h + 1$ . We start in the left edge. The probability of class 0 ( $p_0 = p_{\mathcal{A}}(0)$ ) is reduced by a factor  $c$  ( $0.5 < c < 1$ ). That is,  $\hat{p}_0 = p_{\mathcal{A}}(0) = c \cdot p_0$ . This reduction  $((1 - c) \cdot p_0)$  is accumulated to class 1 that is also reduced by  $c$ :  $\hat{p}_1 = c \cdot p_1 + (1 - c) \cdot p_0$ . For class  $h - 1$  we have  $\hat{p}_{h-1} = c \cdot p_{h-1} + (1 - c) \cdot p_{h-2}$ . For the classes in the mean, we must calculate another reducing factor  $c'$ . The reason is to reduce these classes according to the mean. For example, if the mean is very near the class  $h$  we must reduce the class  $h + 1$  accordingly. On the other hand if the mean is near to class  $h + 1$  we must reduce the class  $h$ . This factor is obtained as  $c' = 0.5 - (h - E(\mathcal{A}))$ . Assume  $c' > 0$  (mean near class  $h$ ), then we have  $\hat{p}_h = p_h + (1 - c) \cdot p_{h-1} + c' \cdot p_{h+1}$  and  $\hat{p}_{h+1} = (1 - c') \cdot p_{h+1} + (1 - c) \cdot p_{h+2}$ . For  $c' < 0$  is similar. The right side is equivalent to the left. So, we have:

$$\begin{array}{llll}
\hat{p}_0 & = & c \cdot p_0 & = & p_0 \cdot c \\
\hat{p}_1 & = & c \cdot p_1 + (1 - c) \cdot p_0 & = & p_0 + (p_1 - p_0) \cdot c \\
\vdots & & \vdots & & \vdots \\
\hat{p}_{h-1} & = & c \cdot p_{h-1} + (1 - c) \cdot p_{h-2} & = & p_{h-2} + (p_{h-1} - p_{h-2}) \cdot c \\
\hat{p}_h & = & p_h + (1 - c) \cdot p_{h-1} + c' \cdot p_{h+1} & = & p_h + p_{h-1} + c' \cdot p_{h+1} + -p_{h-1} \cdot c \\
\hat{p}_{h+1} & = & (1 - c') \cdot p_{h+1} + (1 - c) \cdot p_{h+2} & = & (1 - c') \cdot p_{h+1} + p_{h+2} + -p_{h+2} \cdot c \\
\hat{p}_{h+2} & = & c \cdot p_{h+2} + (1 - c) \cdot p_{h+3} & = & p_{h+3} + (p_{h+2} - p_{h+3}) \cdot c \\
\vdots & & \vdots & & \vdots \\
\hat{p}_{n-1} & = & c \cdot p_{n-1} & = & p_{n-1} \cdot c
\end{array} \tag{20}$$

It is easy to see that  $\sum_0^{n-1} \hat{p}_i = 1$ .

The goal of the following method is to transform a histogram  $\mathcal{A}$  to a scaled histogram  $\hat{\mathcal{A}}$  with variance  $v = m^{2H-2} \cdot \text{Var}(\mathcal{A})$ . We have:

$$\text{Var}(\hat{\mathcal{A}}) = \sum_0^{n-1} \underbrace{(i - E[\hat{\mathcal{A}}])^2}_{d_i} \cdot \hat{p}_i = \sum_0^{n-1} d_i \cdot \hat{p}_i = v \tag{21}$$

Then, we multiply all the terms in equation 20 by  $d_i$  and summing all the probabilities and grouping terms we have:  $\sum_0^{n-1} d_i \cdot \hat{p}_i = S_c \cdot c + S = v$  (the  $S_A$  coefficient is the sum of all the coefficients of the  $c$  variable and  $S_B$  is the sum of the coefficients without variable). Then the reducing factor  $c$  is obtained as:

$$c = \frac{v - S}{S_c} \tag{22}$$

For obtaining the coefficients we must obtain the values of  $d_i$ . We need to estimate the value of  $E[\hat{\mathcal{A}}]$ . The mean of the original histogram ( $E[\mathcal{A}]$ ) is a good estimator. Using this mean we obtain the sum of the coefficients and resolve equation 22. With this factor  $c$  we apply equations 20 for obtaining the distribution of  $\hat{\mathcal{A}}$ . Normally, the variance of  $\hat{\mathcal{A}}$  will not be exactly  $v$  because we are estimating the mean as  $E[\mathcal{A}]$ . So we can repeat the calculations using the mean of the calculated histogram  $E[\hat{\mathcal{A}}]$ . Practically, in 2 or 3 iterations we obtain the desired histogram with the required variance. If the result of  $c$  is not in the range  $0.5 < c < 1$  this mean that the desired variance can not be obtained in one step. So, we apply a factor of 0.5 to obtain a new histogram that is shrunk by a factor of 0.5 and using this new histogram we can obtain a new factor  $c$ , until the factor is greater than 0.5.

The algorithm of the *HScale* function is outlined in Figure 17. In the algorithm we have 2 functions: The function *SumCoef* obtains the sum of the coefficients  $S_A$  and  $S_B$  for a given mean, and the function *Getpmf* calculate the pmf according to equation 20.

As an example of how the function works, we will use the MAWI histogram of Figure 1b,  $p(\mathcal{A}) = [0.0003, 0.0002, 0.0021, 0.0641, 0.2663, 0.3228, 0.2345, 0.0980, 0.0110, 0.0005]$  that has a base period of 0.04ms, variance 1.282 and an interval length of  $l_A = 0.08Mb$ . For  $m = 10$  and  $H = 0.85$  the result is:  $p(\hat{\mathcal{A}}) = [0.0001, 0.0002, 0.0010, 0.0289, 0.1478, 0.6580, 0.1105, 0.0481, 0.0050, 0.0002]$  with an interval length of 8Mb. The variance is 0.6334 according to  $v = var(\mathcal{A}) \cdot m^{2H-2}$ .

## B Buffer analysis as a DTMC

In this appendix we show that the stochastic process  $\{\mathcal{Q}_k\}$  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  $k \rightarrow \infty$ ).

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  $\{\mathcal{Q}_k\}$  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:

$$P[\mathcal{Q}_k = j] = \sum_i P[\mathcal{Q}_{k-1} = i] \cdot P[\mathcal{Q}_k = j | \mathcal{Q}_{k-1} = i] \quad (23)$$

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 Equation 24 is reduced to:

$$P[\mathcal{Q}_k = j] = \sum_i P[\mathcal{Q}_{k-1} = i] \cdot p_{ij} \quad (24)$$

---

**Algorithm**  $\hat{\mathcal{A}} = \text{HScale}(\mathcal{A}, m, H)$

**input**

$\mathcal{A}$                     *'Original histogram*

$m$                     *'Aggregation*

$H$                     *'Hurst parameter*

**output**

$\hat{\mathcal{A}}$                     *'Histogram for the m-aggregation*

---

```

1  begin
3     $v = \text{Var}(\mathcal{A}) \cdot m^{(2-2H)}$ ;  $\mathcal{A}_c = \mathcal{A}$ 
4    while true
5       $\text{mean} = E[\mathcal{A}_c]$ 
6      do
7         $[S, S_c] = \text{SumCoef}(\mathcal{A}_c, \text{mean})$ 
8         $c = (v - S)/S_c$ 
9         $\mathcal{A}_f = \text{Getpmf}(\mathcal{A}_c, c, \text{mean})$             'Calculate the pmf for c
10        $\text{mean} = E[\mathcal{A}_f]$                     'Repeat with a new mean
11       while  $(0.5 < c < 1)$  and  $(|\text{Var}(\mathcal{A}_f) - v| < \epsilon)$ 
12       if  $0.5 < c < 1$ 
13          $\hat{\mathcal{A}} = \mathcal{A}_f$                     'The histogram has Var=v!!!
14         return
15       else
16          $\text{mean} = E[\mathcal{A}_c]$ 
17          $\mathcal{A}_c = \text{Getpmf}(\mathcal{A}_c, 0.5, \text{mean})$             'Scale the histogram by 0.5
18       end if
19     end while
20  end

```

---

Figure 17: The fv algorithm.

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

$$P = [p_{ij}] = \begin{bmatrix} p_{00} & p_{01} & p_{02} & \cdots \\ p_{10} & p_{11} & p_{12} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \quad (25)$$

The components of this matrix are easy to obtain using the definition of the stochastic process  $Q_k$ . 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  $\Phi_r^b([1, 0, 0, 0, \dots] \otimes \mathcal{A})$ . Using the matrix  $P$  we can obtain the pmf of  $Q_k$  as:

$$\mathcal{Q}_k = \mathcal{Q}_1 P^k \quad (26)$$

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

$$\mathbf{v} = \mathbf{v}P \quad v_j \geq 0, \quad \sum_j v_j = 1 \quad (27)$$

This matrix has a finite size of  $(b+1) \times (b+1)$ . Nevertheless, numerically resolving Equation 27 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,  $p(\mathcal{A}) = [0.0003, 0.0002, 0.0021, 0.0641, 0.2663, 0.3228, 0.2345, 0.0980, 0.0110, 0.0005]$  with  $r=5$  and  $b=4$  we obtain the following matrix:

$$P = \begin{bmatrix} 0.6558 & 0.2345 & 0.0980 & 0.0110 & 0.0005 \\ 0.3330 & 0.3228 & 0.2345 & 0.0980 & 0.0115 \\ 0.0667 & 0.2663 & 0.3228 & 0.2345 & 0.1095 \\ 0.0026 & 0.0641 & 0.2663 & 0.3228 & 0.3440 \\ 0.0005 & 0.0021 & 0.0641 & 0.2663 & 0.6668 \end{bmatrix} \quad (28)$$

We can obtain the second iteration state as  $\mathcal{Q}_2 = \mathcal{Q}_1 P = [0.5147, 0.2563, 0.1539, 0.0569, 0.0179]$ . The steady state probability vector is  $\mathbf{v} = [0.1912, 0.1585, 0.1852, 0.1960, 0.2691]$ . This is the pmf of  $\mathcal{Q}$  ( $p(\mathcal{Q})$ ).