Efficient Simulation of Delay Threshold Probabilities in ATM Switches with Heterogeneous Traffic

Ahmet A. Akyamaç
J. Keith Townsend
Center for Advanced Computing and Communication
North Carolina State University
Raleigh, N.C.

Abstract

Future high speed networks using the Asynchronous Transfer Mode will be able to carry a broad range of traffic classes and will also have to provide a quality of service (QoS) measure to many of the traffic classes. Typical QoS measures are cell loss probability (CLP) and cell transfer delay (CTD), which includes queueing delays and service times.

For ATM networks with heterogeneous input traffic, namely constant bit rate (CBR) and variable bit rate (VBR) traffic, we consider the aggregate probability that the cell transfer delay exceeds a given threshold (delay threshold probability). We use the connection traffic descriptors standardized by the ATM Forum to characterize the input traffic. To estimate the rare delay threshold probability, we first present a multinomial Monte Carlo simulation model. The multinomial formulation effectively removes the correlations associated with the bursty events. We then present a three part Importance Sampling procedure to increase the efficiency of the simulation. For the experimental systems considered, the improvement in simulation efficiency is inversely proportional to the probability being estimated [1].

[1] This work was supported by the Center for Advanced Computing and Communication, North Carolina State University. The authors would like to thank Brad Makrucki of IBM (formerly of BellSouth) for suggesting this problem and his helpful feedback.

1 Introduction

Future high speed networks using the Asynchronous Transfer Mode will be able to carry a broad range of traffic classes and will also have to provide a quality of service (QoS) measure to many of the traffic classes. Typical QoS measures are cell loss probability (CLP) and cell transfer delay (CTD), which includes queueing delays and service times.

The events associated with the QoS measures in ATM networks are typically rare. For example, the CLP and the probability that the cell delay exceeds a given high threshold (delay threshold probability) are expected to be in the range of 10^{-9} to 10^{-12}. Second, due to the definition of the different classes of bursty and non-bursty traffic, it is difficult to find an accurate statistical model to account for all classes of traffic. These facts, along with the large buffer sizes make ATM switches difficult to analyze. A solution is to estimate the QoS parameters using Monte Carlo simulation. Unfortunately, Monte Carlo simulation becomes infeasible in the range of probabilities of interest.

In [1] and [2], we developed and demonstrated efficient methods for simulating for the cell loss probability and delay threshold probability, respectively, for single-stage ATM switches with homogeneous input traffic. These methods used Importance Sampling as a means of generating efficient simulations. We extended our work to include cell loss probability in the case of heterogeneous traffic [3].

In this paper, we present, for the first time, a simulation model to estimate the aggregate delay threshold probability resulting from a single stage ATM switch with heterogeneous input traffic, namely constant bit rate (CBR) and variable bit rate (VBR) traffic. Since the probabilities involved are rare, we use Importance Sampling (IS) as a means of generating efficient simulations.

Rather than using statistical models to characterize the input traffic, we use the connection traffic descriptors standardized by the ATM Forum [4]. These descriptors are the peak cell rate $\hat{\lambda}$ in Mbps, the mean cell rate $\bar{\lambda}$ in Mbps and the maximum burst duration $\hat{B}$ cells at the peak rate. This approach has been called the operational approach in [5]. Our model does not place any restrictions on the input traffic other than the fact that it has to be UPC compliant.

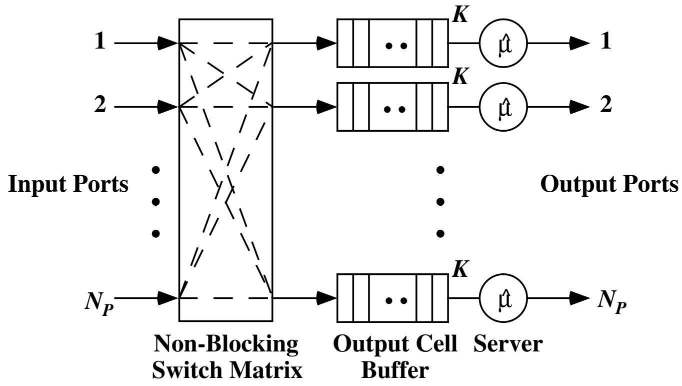

Figure 1: ATM switch model.

This paper is organized as follows: In Section 2, we give a description of the system and the input traffic. We also list our assumptions and give a description of the slotted-time system model we use in our simulations. In Section 3 we demonstrate that an exhaustive solution to finding the delay threshold probability quickly becomes intractable for realistic systems. We then proceed to describe a Monte Carlo method to estimate the rare delay threshold probability. The IS method that we use to speed up our simulations is described in Section 4. We present some experimental results in Section 5 and observe that the improvement we obtain using IS instead of MC is inversely proportional to the probability being estimated. Finally, we draw conclusions in Section 6.

2 System Description

The model we use for the ATM switch is shown in Fig. 1. The switch has N_P input ports and N_P output ports and we assume the number of connections is $N_C = N_P$. We assume there are two classes of traffic: CBR traffic has a triplet $(\hat{\lambda}_{\mathrm{CBR}}, \bar{\lambda}_{\mathrm{CBR}}, \hat{B}_{\mathrm{CBR}})$, VBR traffic has a triplet $(\hat{\lambda}_{\mathrm{VBR}}, \bar{\lambda}_{\mathrm{VBR}}, \hat{B}_{\mathrm{VBR}})$. For the CBR traffic, $\hat{\lambda}_{\mathrm{CBR}} = \bar{\lambda}_{\mathrm{CBR}}$ and $\hat{B}_{\mathrm{CBR}} = 1$. We also assume that $\hat{\lambda}_{\mathrm{VBR}} = k\hat{\lambda}_{\mathrm{CBR}}$, where $k \in \mathcal{Z}^+$ and $k \geq 2$. For this reason, the VBR connection is referred to as the dominant class and the CBR connection is referred to as the nondominant class. Each connection is routed uniformly to one of the output ports through a nonblocking shared bus that operates at a speed of $N_P\hat{\lambda}_{\mathrm{VBR}}$. We analyze a single "tagged"

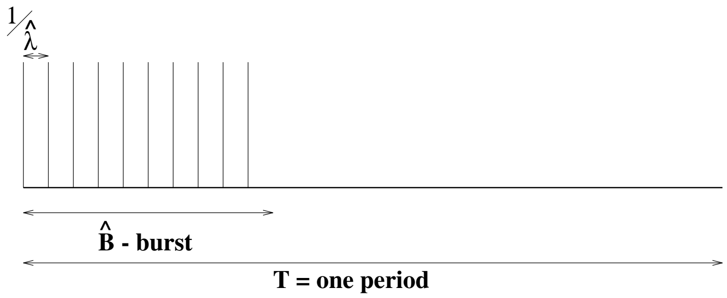

Figure 2: Greedy ON/OFF traffic.

output buffer to represent the performance of all output buffers. The output buffers have a finite size of K cells, a service rate of $\hat{\mu}$ Mbps and use a FIFO queueing discipline.

We assume the routing is instantaneous in that it has negligible effect on the cell's end-to-end delay. We also assume worst case VBR traffic. We consider worst case UPC-compliant VBR traffic to be the periodic greedy ON/OFF traffic pattern. In the ON period, the source generates traffic at a peak rate of $\hat{\lambda}_{\mathrm{VBR}}$ for a burst duration of $\hat{B}_{\mathrm{VBR}}$, and the OFF period is required to average out to the mean rate $\bar{\lambda}_{\mathrm{VBR}}$. The greedy ON/OFF traffic is illustrated in Fig. 2. Although there is a debate as to whether or not the greedy ON/OFF pattern defines the worst case UPC compliant traffic, it has been considered the Worst Case Traffic (WCT) in [6], [7], [8], [9] and [10]. This traffic has been shown to cause more congestion in the case of finite buffers [9] and has been demonstrated to yield results very similar to the alternate worst cases [6], [11] under infinite buffers.

For our simulations, we use a slotted time model resulting from normalization with respect to the peak rate of the VBR source, $\hat{\lambda}_{\mathrm{VBR}}$. Hence, one slot corresponds to the transmission of a 53-byte ATM cell. With this normalization, the equivalent arrival rate of the VBR source is 1 cell/slot and that of the CBR source is $\hat{\lambda}_{\mathrm{CBR}}/\hat{\lambda}_{\mathrm{VBR}} = 1/k$ cells/slot, where k was defined above. Both the VBR and the CBR traffic are periodic. The corresponding periods are given as follows:

$$T_{\mathrm{VBR}} = \frac{\hat{\lambda}_{\mathrm{VBR}}\hat{B}_{\mathrm{VBR}}}{\bar{\lambda}_{\mathrm{VBR}}} \tag{1}$$

$$T_{\mathrm{CBR}} = \frac{\hat{\lambda}_{\mathrm{VBR}}\hat{B}_{\mathrm{CBR}}}{\bar{\lambda}_{\mathrm{CBR}}} = \frac{k\hat{\lambda}_{\mathrm{CBR}}}{\bar{\lambda}_{\mathrm{CBR}}} = k \tag{2}$$

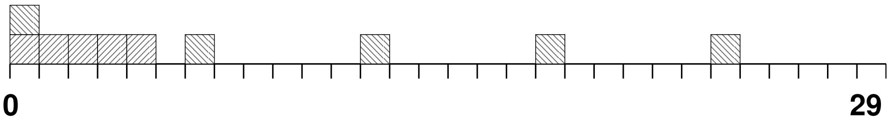

Figure 3: Slotted-time example pattern for CBR/VBR traffic.

Since both the VBR and CBR traffic are periodic, the entire input traffic has a super-period of $T_{\mathrm{SP}} = \mathrm{lcm}(T_{\mathrm{VBR}}, T_{\mathrm{CBR}})$. We assume T_{VBR} to be an integer multiple of k so that $T_{\mathrm{VBR}} = T_{\mathrm{SP}}$. After normalization, the equivalent service rate is given by $\mu = \hat{\mu}/\hat{\lambda}_{\mathrm{VBR}}$ cells/slot. An example of the resulting slotted-time pattern is plotted in Fig. 3 for $\hat{\lambda}_{\mathrm{VBR}} = 6$ cells/slot, $\bar{\lambda}_{\mathrm{VBR}} = 1$ cell/slot, $\hat{B}_{\mathrm{VBR}} = 5$ cells and $\bar{\lambda}_{\mathrm{CBR}} = \hat{\lambda}_{\mathrm{CBR}} = 1$ cell/slot and $\hat{B}_{\mathrm{CBR}} = 1$ cell where both connections start at slot 0.

The randomness in the simulation model is found in the connection starting-slot positions. Each of the N_{VBR} connections can be characterized by a discrete-time random variable, uniform on $[0, T_{\mathrm{SP}} - 1]$ representing the starting-slot of the VBR connection. Similarly, each of the N_{CBR} CBR connections can be characterized by a discrete-time random variable, uniform on $[0, T_{\mathrm{CBR}} - 1]$. Here, $N_C = N_{\mathrm{CBR}} + N_{\mathrm{VBR}}$. We represent the starting-slots as an N_C-dimensional vector $\underline{v}$, which can be partitioned into two vectors $\underline{v}_{\mathrm{CBR}}$ and $\underline{v}_{\mathrm{VBR}}$ representing the starting-slots of the VBR and CBR connections, respectively, as follows:

$$\underline{v} = (\underline{v}_{\mathrm{VBR}} | \underline{v}_{\mathrm{CBR}}) \tag{3}$$

For simplicity, we fix one VBR connection to start at slot 0. Hence, the first component of $\underline{v}$ is always 0.

Each arrival slot is composed of N_C service slots as in Fig. 4. If at an arrival slot the number of cell arrivals is $N < N_C$, we assume these cells occupy the first N service slots. In each service slot, one cell can be loaded into the output buffer and μ/N_C cells can be serviced. For a cell that is not lost due to buffer overflow, the delay threshold τ represents a fraction of the buffer length, $K_m = \alpha K$. The queue length is compared to K_m right after the arrival (but before the service) of a cell in a service slot. The delay threshold will be represented by τ and K_m interchangeably in this paper. Since we estimate the aggregate

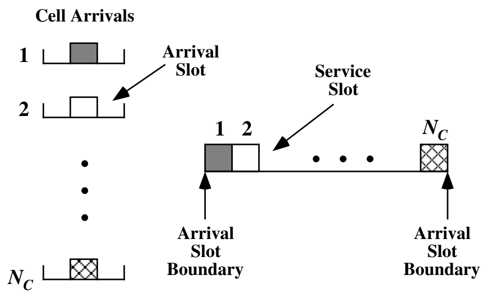

Figure 4: Arrival slot and service slot relationship.

delay threshold probability, there is no distinction between cells in an arrival slot. Hence, all cells have equal priority upon arrival in an arrival slot.

3 Simulation Framework and Monte Carlo Simulation

3.1 Upper Bound for Finite Buffers

A problem in collecting end-to-end delay statistics and cell loss statistics simultaneously comes from the fact that often the two statistics are correlated. From an empirical study, we observed the following in [12]:

$$\Pr(D_K > \tau) \leq \Pr(D_\infty > \tau) - \Pr(\text{cell loss}) \tag{4}$$

where the subscripts K and ∞ refer to finite and infinite buffers, respectively and D refers to the cell delay making no distinction between CBR cells and VBR cells. The expression in (4) upper bounds $\Pr(D_K > \tau)$ by $\Pr(D_\infty > \tau) - \Pr(\text{cell loss})$. In the remainder of this paper, we will focus on estimating $\Pr(D_\infty > \tau)$ for the infinite buffer, hereby denoted by P_{TH} for conciseness. The cell loss results of [3] can be combined with the delay results here to generate the upper bound estimate.

3.2 Exhaustive Solution to P_{TH}

Let D_{max} be the maximum number of cells that can exceed the given delay threshold for the infinite buffer. In [12, 13], we show that for homogeneous traffic sources, the worst case congestion occurs when all connections start at the same slot. For the combined CBR/VBR traffic, worst case congestion occurs when all CBR and VBR connections start at the same slot. Since one VBR connection is fixed to start at slot 0, the worst case congestion occurs for the aligned-at-zero (AAZ) case. Hence, D_{max} can be found by running the AAZ vector given by $\underline{v} = (0 \ldots 0|0 \ldots 0)$. This run is equivalent to a single simulation run and adds negligible overhead to the overall simulation.

Let V be the set of all connection starting-slot vectors and n_{TH_j} be the number of connection starting-slot vectors that map to exactly j cells that exceed the threshold τ in steady state. Here, $|V| = (T_{\mathrm{VBR}})^{N_{\mathrm{VBR}}} \cdot (T_{\mathrm{CBR}})^{N_{\mathrm{CBR}}}$, where $|\cdot|$ denotes the cardinality of the set. As in [2], the probability that j cells exceed the threshold in steady state is given by $p_{TH_j} = n_{TH_j}/|V|$ for $j = 0, \cdots, D_{max}$. Note that $\sum_{j=0}^{D_{max}} p_{TH_j} = 1$. The average number of cells $\bar{n}_{TH}$ that exceed the threshold in steady state is then given by $\bar{n}_{TH} = \sum_{j=0}^{D_{max}} j p_{TH_j}$. Hence, as in [2], the delay threshold probability for the infinite buffer is given by:

$$P_{TH} = \sum_{j=0}^{D_{max}} \left(\frac{j}{N_{\mathrm{cells}}} \right) p_{TH_j} \tag{5}$$

where N_{cells} is the total number of cells that arrive in a superperiod,

$$N_{\mathrm{cells}} = N_{\mathrm{VBR}} \hat{B}_{\mathrm{VBR}} + N_{\mathrm{CBR}} \cdot \frac{T_{\mathrm{SP}}}{\hat{\lambda}_{\mathrm{VBR}}/\hat{\lambda}_{\mathrm{CBR}}} \tag{6}$$

In this formulation, P_{TH} is expressed as a weighted sum of multinomial probabilities. Thus, we have a "multiple bin" probability structure composed of bins numbered 0 to D_{max}, where bin j corresponds to j cells that exceed the threshold. To obtain P_{TH}, we have to run all possible vectors in $|V|$ and collect the probabilities p_{TH_j}. This exhaustive solution has a complexity of $O((T_{\mathrm{VBR}})^{N_{\mathrm{VBR}}} \cdot (T_{\mathrm{CBR}})^{N_{\mathrm{CBR}}})$ and quickly becomes intractable for realistic systems.

3.3 Monte Carlo Simulation

Using Monte Carlo simulation, we form an unbiased estimate of P_{TH} as follows:

$$\hat{P}_{TH} = \frac{1}{N_{\mathrm{cells}}} \sum_{j=1}^{D_{max}} j \hat{p}_{TH_j} \tag{7}$$

where $\hat{p}_{TH_j}$ are estimates of the individual probabilities p_{TH_j}, $j = 1, \ldots, D_{max}$. The estimate $\hat{p}_{TH_j}$ is formed by running N_{MC} simulation runs, with one connection starting-slot vector being drawn uniformly from the sample space V for each run as follows [2]:

$$\hat{p}_{TH_j} = \frac{1}{N_{MC}} \sum_{s=1}^{N_{MC}} I_j(s) \tag{8}$$

where $I_j(s)$ is the indicator function for j cells exceeding the threshold in steady state for vector s.

Since the delay threshold probability is rare, if j cells do not exceed the threshold, then most of the time 0 cells will. Thus, we consider the bins to be independent since bin 0 is not included in our estimates [12]. This multiple bin structure also helps us generate proper confidence intervals for our estimates.

The variance of the estimator P_{TH} is given by:

$$\sigma^2(\hat{P}_{TH}) = \frac{1}{(N_{\text{cells}})^2} \sum_{j=1}^{D_{max}} j^2 \frac{p_{TH_j}(1 - p_{TH_j})}{N_{MC}} \tag{9}$$

Since we do not know the probabilities p_{TH_j} *a priori*, we estimate this variance as follows:

$$\hat{\sigma}^2(\hat{P}_{TH}) = \frac{1}{(N_{\text{cells}})^2} \sum_{j=1}^{D_{max}} j^2 \hat{\sigma}^2(\hat{p}_{TH_j}) \tag{10}$$

where $\hat{\sigma}^2(\hat{p}_{TH_j})$ are the estimates of the estimator variances for the individual bins:

$$\hat{\sigma}^2(\hat{p}_{TH_j}) = \frac{1}{N_{MC}(N_{MC} - 1)} \sum_{s=1}^{N_{MC}} (I_j(s) - \hat{p}_{TH_j})^2 \tag{11}$$

Following [3] and [2], we generate confidence intervals using the result that the confidence interval of a weighted sum of individual multinomial probabilities follows a χ^2 distribution [14]. The χ^2 distribution results in higher confidence intervals than using the normal distribution, but the multinomial framework effectively removes the correlation between important events.

3.4 Range for N_C

For homogeneous traffic with a given threshold τ, there is a minimum number of connections necessary, $N_C = N_{C_0}(\tau)$, such that if there are less than $N_{C_0}(\tau)$ connections, no cells will exceed the threshold [13, 2]. The same holds true in the case of heterogeneous traffic. However, for heterogeneous traffic, $N_{C_0}(\tau)$ can have different values depending on different combinations of the number of CBR and VBR sources. We arbitrarily fix the number of VBR (dominant) connections to the minimum number required with N_C^n CBR (nondominant) connections, $N_{C_0}^p(\tau)$, such that if there are less than $N_{C_0}^p(\tau)$ VBR connections, no cells will exceed the threshold. We then vary the number of CBR sources and estimate delay threshold probabilities. Varying the number of VBR sources will result in delay threshold probability curves very similar to the ones in [12, 2].

Let the number of CBR connections be fixed at N_C^n at a threshold of τ. If only VBR sources were present, the minimum number of connections required, $N_{C_0}(\tau)$, is given by [12, 2]:

$$N_{C_0}(\tau) = \left\lfloor \frac{\mu + \frac{K_m}{\hat{B}_{\mathrm{VBR}}} + \sqrt{(\mu + \frac{K_m}{\hat{B}_{\mathrm{VBR}}})^2 - \frac{4\mu}{\hat{B}_{\mathrm{VBR}}}}}{2} \right\rfloor + 1 \tag{12}$$

Hence, $N_{C_0}(\tau)$ is an absolute upper bound for $N_{C_0}^p(\tau)$. An absolute lower bound for $N_{C_0}^p(\tau)$ can be found by assuming that the N_C^n CBR connections are in fact of the same class as the VBR connections. Then, the absolute lower bound for $N_{C_0}^p(\tau)$ is $N_{C_0}(\tau) - N_C^n$. Thus, we have:

$$N_{C_0}(\tau) - N_C^n \leq N_{C_0}^p(\tau) \leq N_{C_0}(\tau) \tag{13}$$

To find $N_{C_0}^p(\tau)$, we run the AAZ case from $N_{C_0}(\tau) - N_C^n$ VBR connections to $N_{C_0}(\tau)$ VBR connections. Then, $N_{C_0}^p(\tau)$ is the minimum number of VBR connections for which at least one cell exceeds the threshold. Note that this stage is equivalent in run-time to at most $N_C^n + 1$ simulation runs, which represents a negligible simulation overhead.

Given $N_{C_0}^p(\tau)$ VBR connections, there is a certain number of CBR connections $N_{C_g}^n$ such that if there are more than $N_{C_g}^n$ CBR connections, then all cells will eventually exceed the threshold in steady-state. So, we must have:

$$N_{C_0}^p(\tau)\bar{\lambda}_{\mathrm{VBR}} + N_{\mathrm{CBR}}\bar{\lambda}_{\mathrm{CBR}} \leq \hat{\mu} \tag{14}$$

Thus, we find:

$$N_{C_g}^n = \left\lfloor \frac{\hat{\mu} - N_{C_0}^p(\tau)\bar{\lambda}_{\mathrm{VBR}}}{\bar{\lambda}_{\mathrm{CBR}}} \right\rfloor \tag{15}$$

where $\lfloor \cdot \rfloor$ denotes the floor operator.

Simulations are performed for $N^p_{C_0}(\tau)$ VBR connections and CBR connections in the range $N^n_C \leq N_{\mathrm{CBR}} \leq N^n_{C_g}$.

4 Importance Sampling Method

For ATM networks, the delay threshold probability range of interest is 10^{-9} to 10^{-12}. In order to obtain sufficiently accurate estimates in this range, at least 10^{11} to 10^{14} Monte Carlo simulation runs are necessary. Thus, MC simulation quickly becomes intractable at these low probabilities.

We use Importance Sampling to modify or "bias" the initial probability density function (pdf) $f_{\underline{V_1},\underline{V_2}}(\underline{v}_1, \underline{v}_2)$ to $f^*_{\underline{V_1^*},\underline{V_2^*}}(\underline{v}_1^*, \underline{v}_2^*)$ such that the estimate is formed with this new pdf. Here, V_1 refers to the set of all possible connection starting-slot vectors $\underline{v}_1$ for the VBR source and V_2 refers to the set of all possible connection starting-slot vectors $\underline{v}_2$ for the CBR source. Note that $V = V_1 \times V_2$. Let us call the new estimate $\hat{P}^*_{TH}$. We require that the variance of the new estimate be reduced for a given number of simulation runs, or equivalently, that the number of simulation runs required to achieve a given variance be reduced.

Note that due to the independence of the VBR and CBR connections, the pdf can be written as: $f_{\underline{V_1},\underline{V_2}}(\underline{v}_1, \underline{v}_2) = f_{\underline{V_1}}(\underline{v}_1) \cdot f_{\underline{V_2}}(\underline{v}_2)$. Since there is little to be gained by biasing the CBR source, we only bias the VBR source.

To keep $\hat{P}_{TH}$ unbiased, each important event (identified by cells that exceed the threshold) must be appropriately weighted or "unbiased". Since we only bias the VBR source, $f^*_{\underline{V_1^*},\underline{V_2^*}}(\underline{v}_1^*, \underline{v}_2^*) = f^*_{\underline{V_1^*}}(\underline{v}_1^*) \cdot f_{\underline{V_2}}(\underline{v}_1^*)$. Hence, the weight used to unbias the IS estimate is given by $w(\underline{v}_1^*, \underline{v}_2^*) = w(\underline{v}_1^*) \cdot 1$. Additionally, we require that the biased pdf $f^*_{\underline{V}^*}(\underline{v}^*) > 0$ whenever the original pdf $f_{\underline{V}}(\underline{v}) > 0$. Here, $V^* = V_1^* \times V_2$.

We solve the problem of delay threshold probability calculation with three algorithms. The first algorithm (distance calculations) generates distances that identify the set V^* in the implementation of IS. The second algorithm (interval reduction) , adapted from [1] samples a connection starting-slot vector and generates an IS weight given a set of distances. The third algorithm (distance-shrinking) extends the distance calculation algorithm to subsequent bins.

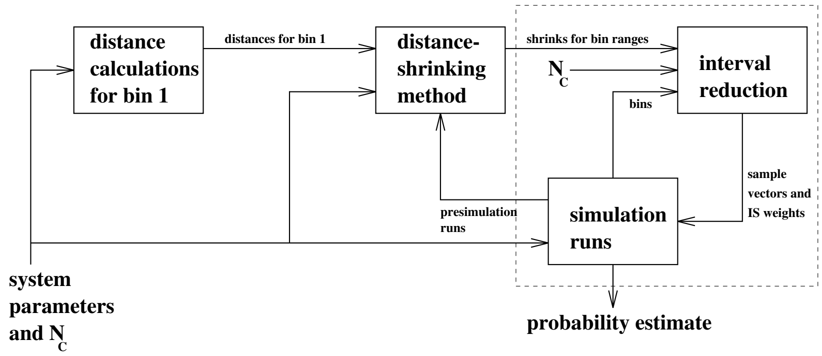

Figure 5: Relation between the three IS algorithms and the simulation runs.

The functions and relations of these three IS algorithms are shown in Fig. 5. The distance calculation algorithm and the distance-shrinking algorithm involve presimulation calculations and presimulation runs, respectively. The simulation runs are represented by the dashed box in Fig. 5. The interval reduction algorithm, using the results of the presimulation algorithms, samples a vector and generates an IS weight. The simulation runs accumulate the necessary statistics for the sampled vector.

4.1 Theory

The important region can be identified by a subset V_{TH} of V such that every vector $\underline{v} = (\underline{v}_1|\underline{v}_2)$ in V_{TH} results in at least one cell that exceeds the threshold. Due to the multiple bin structure, $V = V_0 \cup V_1 \cup \cdots \cup V_{D_{max}}$ and $V_{TH} = V_1 \cup \cdots \cup V_{D_{max}}$, where the set V_j contains only those vectors that result in j cells that exceed the threshold in steady state. For realistic ATM systems, $|V_0| \gg |V_j|$ for $j = 1 \cdots D_{max}$. Monte Carlo simulation becomes intractable since it samples from the entire set V, including the set V_0 which is composed of the vectors that cause no important events. Ideally with IS, we would like to sample exclusively from the set V_{TH}. However, the set partitions V_j are not known *a priori*. Instead, we sample from a set $V^* = V_1^* \cup \cdots \cup V_{D_{max}}^*$, where $|V_j^*| \ll |V|$ and $|V^*| \ll |V|$. However, the sets V_j^* may no longer be disjoint and may even have intersections with V_0.

We form the unbiased estimate $\hat{P}_{TH}^*$ as a linear combination of individual estimates $\hat{p}_{TH_j}^*$

as before:

$$\hat{P}^*_{TH} = \sum_{j=1}^{D_{max}} \left(\frac{j}{N_{\text{cells}}}\right) \hat{p}^*_{TH_j} \tag{16}$$

But for the IS case, the individual estimates $\hat{p}^*_{TH_j}$ are formed as follows:

$$\hat{p}^*_{TH_j} = \frac{1}{n_j} \sum_{s=1}^{n_j} I^*_j(s) w^*_j(s) \tag{17}$$

where $I^*_j(s)$ is the indicator function of a cell exceeding the threshold under the new pdf $f^*_{\underline{V_1^*},\underline{V_2^*}}(\underline{v}^*_1, \underline{v}^*_2)$ and $w^*_j(s)$ is the vector-dependent weight function described in Section 4. For run s, the weight for vector $\underline{v}^*(s)$ is given by $w^*_j(s) = f_V(\underline{v}^*(s))/f^*_{V^*}(\underline{v}^*(s))$ Note that there may be a different number of runs n_j for each bin j, as opposed to the constant number of runs N_{MC} in the case of MC simulation. The unbiased estimate of the variance of $\hat{P}^*_{TH}$ is given by:

$$\hat{\sigma}^2(\hat{P}^*_{TH}) = \frac{1}{(N_{\text{cells}})^2} \sum_{j=1}^{D_{max}} j^2 \hat{\sigma}^2(\hat{p}^*_{TH_j}) \tag{18}$$

where

$$\hat{\sigma}^2(\hat{p}^*_{TH_j}) = \frac{1}{n_j(n_j-1)} \sum_{s=1}^{n_j} (I^*_j(s) w^*_j(s) - \hat{p}^*_{TH_j})^2 \tag{19}$$

4.2 Distance Calculations

As mentioned in Section 4, we bias only the VBR source since there is little to be gained from biasing the CBR source. Here, the function of the distance calculation algorithm is to identify the set V_1^* from which the VBR starting-slot vector $\underline{v}_1$ is to be sampled. Given a number of VBR and CBR connections, we define a d-distance as the *maximum* distance in arrival slots between the connection starting-slots of the first and last bursts of the VBR source such that at least one cell can exceed the threshold for at least one of the CBR vectors $\underline{v}_2 \in V_2$. Thus, a d-distance is such that, unless all VBR connections start their bursts within d slots of each other, no cells will exceed the threshold. The d-distances depend on the relative values of the service rate μ and the number of connections N_{VBR}.

4.3 Case 1: $\mu < 1$

For this case, one cell in an arrival slot is sufficient to increase the queue length for that arrival slot. Let d_r denote the d-distance for $N_{\mathrm{VBR}} = N^p_{C_0}(\tau) + r$ connections. We first consider the case where the number of connections is $N^p_{C_0}(\tau)$. For a cell to exceed the threshold, all $N_{C_0}(\tau)$ connections must arrive within d_0 slots of each other.

To find d_0, we consider the configuration shown in Fig. 6. In this starting-slot configuration, one VBR burst is fixed at slot 0 and another VBR burst is fixed at slot d_0. The remaining $N^p_{C_0}(\tau) - 2$ VBR bursts are distributed over the interval $[0, \ldots, d_0]$. By definition, all VBR bursts in this configuration start within d_0 slots of each other.

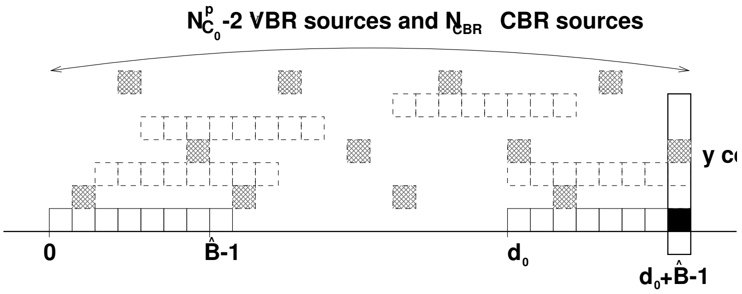

Figure 6: Calculation of d_0 for Case 1.

In the configuration in Fig. 6, we force one cell to exceed the threshold. By definition, $N^p_{C_0}(\tau)$ VBR bursts are required for at least one cell to exceed the threshold. Hence, the queue length cannot drop to 0 in the interval $[0, \ldots, d_0]$. Also, since $\mu < 1$, if one cell in a VBR burst exceeds the threshold, the remaining cells in that burst will also exceed the threshold. Hence, to find d_0, we force the last arriving cell of the VBR burst fixed at slot d_0 to exceed the threshold. Then, d_0 is found by maximizing the distance obtained by varing the starting-slot positions of the $N^p_{C_0}(\tau) - 2$ bursts distributed over the interval $[0, \ldots, d_0]$.

Note also that the queue length will not drop to 0 throughout the entire support of the bursts. Thus, the effect of the $N^p_{C_0}(\tau) - 2$ VBR connections and the N_{CBR} CBR connections can be quantified by the number of cells of these connections that arrive at slot $d_0 + \hat{B}_{\mathrm{VBR}} - 1$. Let y denote the number of cells that arrive in slot $d_0 + \hat{B}_{\mathrm{VBR}} - 1$. Then, $1 \le y \le N_{\mathrm{CBR}} + N^p_{C_0}(\tau) - 1$.

Let x be the total number of CBR cells in the range from slot 0 to slot $d_0 + \hat{B}_{\mathrm{VBR}} - 1$. Then, we have:

$$x = N_{\mathrm{CBR}} \cdot \left\lceil \frac{d_0 + \hat{B}_{\mathrm{VBR}}}{\hat{\lambda}_{\mathrm{VBR}}/\hat{\lambda}_{\mathrm{CBR}}} \right\rceil \tag{20}$$

Forcing the black cell in Fig. 6 to exceed the threshold, we have:

$$N^p_{C_0}(\tau)\hat{B}_{\mathrm{VBR}} + x - \mu(d_0 + \hat{B}_{\mathrm{VBR}}) + (N_C - y + 1)\frac{\mu}{N_C} > K_m \tag{21}$$

where $N_C = N_{\mathrm{CBR}} + N^p_{C_0}(\tau)$ is the total number of connections. Hence,

$$\mu d_0 < \hat{B}_{\mathrm{VBR}}(N^p_{C_0}(\tau) - \mu) + (N_C - y + 1)\frac{\mu}{N_C} - K_m + x \tag{22}$$

To simplify the above equation, we use the following relation:

$$\lceil a + b \rceil \leq \lceil a \rceil + \lceil b \rceil \leq a + 1 + \lceil b \rceil \tag{23}$$

Thus, we upper bound x as follows:

$$x \leq \overline{x} = d_0 \cdot \frac{N_{\mathrm{CBR}}}{\hat{\lambda}_{\mathrm{VBR}}/\hat{\lambda}_{\mathrm{CBR}}} + N_{\mathrm{CBR}} + N_{\mathrm{CBR}} \cdot \left\lceil \frac{\hat{B}_{\mathrm{VBR}}}{\hat{\lambda}_{\mathrm{VBR}}/\hat{\lambda}_{\mathrm{CBR}}} \right\rceil \tag{24}$$

Let $\overline{d_0}$ be the upper bound for d_0 generated by replacing x with $\overline{x}$. Thus, using Eq. 24 in Eq. 22, we have:

$$\overline{d_0} < \frac{\hat{B}_{\mathrm{VBR}}(N^p_{C_0}(\tau) - \mu) - K_m + (N_C - y + 1)\frac{\mu}{N_C} + N_{\mathrm{CBR}} + N_{\mathrm{CBR}} \cdot \left\lceil \frac{\hat{B}_{\mathrm{VBR}}}{\hat{\lambda}_{\mathrm{VBR}}/\hat{\lambda}_{\mathrm{CBR}}} \right\rceil}{\mu - \frac{N_{\mathrm{CBR}}}{\hat{\lambda}_{\mathrm{VBR}}/\hat{\lambda}_{\mathrm{CBR}}}} \tag{25}$$

Th expression in Eq. 25 is maximized when $y = 1$, indicating only one cell in slot $d_0 + \hat{B}_{\mathrm{VBR}} - 1$. Incidentally, this cell is the last cell of the VBR burst fixed to start at slot d_0. Incorporating $y = 1$ and choosing the maximum possible integer value for $\overline{d_0}$, we get:

$$\overline{d_0} = \left\lceil \frac{\hat{B}_{\mathrm{VBR}}(N^p_{C_0}(\tau) - \mu) - K_m + \mu + N_{\mathrm{CBR}} + N_{\mathrm{CBR}} \cdot \left\lceil \frac{\hat{B}_{\mathrm{VBR}}}{\hat{\lambda}_{\mathrm{VBR}}/\hat{\lambda}_{\mathrm{CBR}}} \right\rceil}{\mu - \frac{N_{\mathrm{CBR}}}{\hat{\lambda}_{\mathrm{VBR}}/\hat{\lambda}_{\mathrm{CBR}}}} \right\rceil - 1 \tag{26}$$

For the case of $N_{\mathrm{VBR}} = N^p_{C_0}(\tau) + r$ VBR connections, we use a similar configuration as in Fig. 6, but for which $N_{\mathrm{VBR}} - 2$ VBR bursts are distributed over the interval $[0, \ldots, d_r]$. Proceeding identically to the case of d_0, we obtain:

$$\overline{d_r} = \left\lceil \frac{\hat{B}_{\mathrm{VBR}}(N^p_{C_0}(\tau) + r - \mu) - K_m + \mu + N_{\mathrm{CBR}} + N_{\mathrm{CBR}} \cdot \left\lceil \frac{\hat{B}_{\mathrm{VBR}}}{\hat{\lambda}_{\mathrm{VBR}}/\hat{\lambda}_{\mathrm{CBR}}} \right\rceil}{\mu - \frac{N_{\mathrm{CBR}}}{\hat{\lambda}_{\mathrm{VBR}}/\hat{\lambda}_{\mathrm{CBR}}}} \right\rceil - 1 \tag{27}$$

The expression in Eq. 27 can be used by itself to generate the d-distances. However, it is known to represent an upper bound due to the upper bound of x. For increased efficiency, the following iteration is used to generate the exact d-distances:

- Find $\overline{d_r}$, $0 \leq r \leq N_{\text{VBR}} - N^p_{C_0}(\tau)$.
- For $0 \leq r \leq N_{\text{VBR}} - N^p_{C_0}(\tau)$, do the following:
  - while $(N^p_{C_0}(\tau) + r)\hat{B}_{\text{VBR}} + x - \mu(\overline{d_r} + \hat{B}_{\text{VBR}}) + \mu \leq K_m$,
    - $\overline{d_r} \leftarrow \overline{d_r} - 1$
- $d_r \leftarrow \overline{d_r}$, $0 \leq r \leq N_{\text{VBR}} - N^p_{C_0}(\tau)$.

4.4 Case 2: $\mu \geq 1$

For this case, at least $\lfloor \mu \rfloor + 1$ cells per arrival slot are required for the queue length to increase for that arrival slot. We also assume that $N^p_{C_0}(\tau) - 1 \geq \mu$. Again, we first consider $N^p_{C_0}(\tau)$ VBR connections.

Since $\mu \geq 1$, a physical constraint for this case is that $d_0 \leq \hat{B}_{\text{VBR}}$. To find d_0, we start with all VBR connections at slot 0 and cancel the effects of cells pertaining to one of the bursts such that they no longer contribute to the final queue length (in essence, we remove the effects of the cells on the queue length, or simply "remove" the cells). We remove cells from the last burst starting from the rightmost support and proceed towards slot 0 or starting from the leftmost support and proceed towards slot $\hat{B}_{\text{VBR}} - 1$, as seen in Fig. 7. The rationale behind removing cells from the last VBR burst and not any other burst is that, by definition, $N^p_{C_0}(\tau) - 1$ bursts alone are not sufficient for a cell to exceed the threshold.

As before, let the number of CBR cells in the support from slot 0 to slot $\hat{B}_{\text{VBR}}$ be denoted by x. For this case, x is given by the following:

$$x = N_{\text{CBR}} \cdot \left\lceil \frac{\hat{B}_{\text{VBR}}}{\hat{\lambda}_{\text{VBR}}/\hat{\lambda}_{\text{CBR}}} \right\rceil \tag{28}$$

Let f be the number of cells removed. We find the maximum possible value of f such that one cell is guaranteed to exceed the threshold. The cell that exceeds the threshold varies depending on the direction of the removal process. For both cases, the cell that is guaranteed to exceed the threshold is shown in black in Fig. 7.

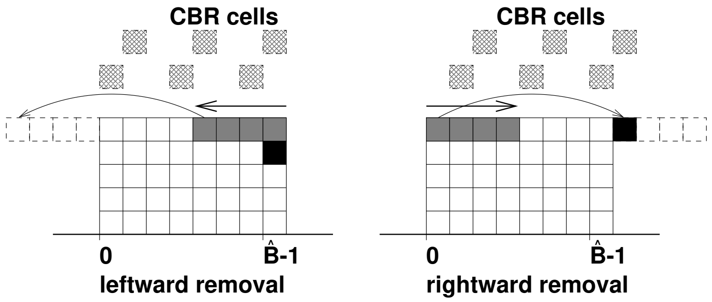

Figure 7: Calculation of d_0 for Case 2.

For rightward removal, we have to check whether or not any CBR cells can fall into slot $\hat{B}_{\mathrm{VBR}}+1$. Forcing the black cell to exceed the threshold, we have:

$$N^p_{C_0}(\tau)\hat{B}_{\mathrm{VBR}} - f + x - \mu\hat{B}_{\mathrm{VBR}} + 1 + yN_{\mathrm{CBR}}\left(1-\frac{\mu}{N_C}\right) > K_m \tag{29}$$

where $y=1$ if $\hat{B}_{\mathrm{VBR}} = lT_{\mathrm{CBR}}$, $l \in \mathcal{Z}^+$ and $y=0$ otherwise. Solving for f, we obtain:

$$f < \hat{B}_{\mathrm{VBR}}(N^p_{C_0}(\tau)-\mu) - K_m + 1 + x + yN_{\mathrm{CBR}}\left(1-\frac{\mu}{N_C}\right) \tag{30}$$

Since $\mu > 1$, the maximum number of arrival slots necessary to cancel the effects of f cells, thus keeping the queue length unchanged, is f. Hence, we set $d_0 = f$, yielding the following:

$$d_0 = \left\lceil \hat{B}_{\mathrm{VBR}}(N^p_{C_0}(\tau)-\mu) - K_m + 1 + N_{\mathrm{CBR}}\cdot\left\lceil\frac{\hat{B}_{\mathrm{VBR}}}{\hat{\lambda}_{\mathrm{VBR}}/\hat{\lambda}_{\mathrm{CBR}}}\right\rceil + yN_{\mathrm{CBR}}\left(1-\frac{\mu}{N_C}\right)\right\rceil - 1 \tag{31}$$

where $y=1$ if $\hat{B}_{\mathrm{VBR}} = lT_{\mathrm{CBR}}$, $l \in \mathcal{Z}^+$ and $y=0$ otherwise.

For leftward removal, we have to check whether or not any CBR cells can fall into slot $\hat{B}_{\mathrm{VBR}}-1$. Again, forcing the black cell to exceed the threshold, we have:

$$N^p_{C_0}(\tau)\hat{B}_{\mathrm{VBR}} - f + x - \mu\hat{B}_{\mathrm{VBR}} + (N_C - (N^p_{C_0}(\tau)-1)+1)\frac{\mu}{N_C} - z\frac{\mu}{N_C} > K_m \tag{32}$$

where $z=1$ if $\hat{B}_{\mathrm{VBR}}-1 = lT_{\mathrm{CBR}}$, $l \in \mathcal{Z}^+$ and $z=0$ otherwise. Solving for f yields

$$f < \hat{B}_{\mathrm{VBR}}(N^p_{C_0}(\tau)-\mu) - K_m + x + (2+N_{\mathrm{CBR}})\frac{\mu}{N_C} - z\frac{\mu}{N_C} \tag{33}$$

We observe that for leftward removal, f is maximized when the CBR connections are aligned such that $z = 0$. By a similar argument as the one above, we have:

$$d_0 = \left\lceil \hat{B}_{\mathrm{VBR}}(N^p_{C_0}(\tau) - \mu) - K_m + (2 + N_{\mathrm{CBR}})\,\frac{\mu}{N_C} + N_{\mathrm{CBR}} \cdot \left\lceil \frac{\hat{B}_{\mathrm{VBR}}}{\hat{\lambda}_{\mathrm{VBR}}/\hat{\lambda}_{\mathrm{CBR}}} \right\rceil \right\rceil - 1 \qquad (34)$$

Combining Eq.'s 31 and 34 and also generalizing to the case of $N_{\mathrm{VBR}} = N^p_{C_0}(\tau) + r$ VBR connections as in Case 1, we find the following:

$$\begin{aligned} d_r \;=\; & \left\lceil \hat{B}_{\mathrm{VBR}}(N^p_{C_0}(\tau) + r - \mu) - K_m + N_{\mathrm{CBR}} \cdot \left\lceil \frac{\hat{B}_{\mathrm{VBR}}}{\hat{\lambda}_{\mathrm{VBR}}/\hat{\lambda}_{\mathrm{CBR}}} \right\rceil + \right. \\ & \left. \max\left\{ 1 + y N_{\mathrm{CBR}} \left(1 - \frac{\mu}{N_C}\right), (2 + N_{\mathrm{CBR}})\,\frac{\mu}{N_C} \right\} \right\rceil - 1 \end{aligned} \qquad (35)$$

where

$$y = \begin{cases} 1, \text{ if } \hat{B}_{\mathrm{VBR}} = l T_{\mathrm{CBR}},\ l \in \mathcal{Z}^+ \\ 0, \text{ otherwise} \end{cases} \qquad (36)$$

4.5 Interval Reduction Algorithm

Given the d-distances, the interval reduction algorithm efficiently samples a vector from the vector space generated by the d-distances and also calculates an IS weight relating to the sampled vector. This algorithm was first developed in [1] and we adapt it here for the case of delay threshold probabilities.

The algorithm iteratively keeps track of the intervals $\hat{t}_i$ over which the starting-slots are sampled, where $\hat{t}_{N_{C_0}(\tau)+r-1} \leq \hat{t}_{N_{C_0}(\tau)+r-2} \leq \cdots \leq \hat{t}_2 \leq \hat{t}_1$. The IS weight is calculated as

$$w^*(\underline{v}^*) = \frac{1}{T^{N_C - 1}} \prod_{k=1}^{N_C - 1} \hat{t}_k \qquad (37)$$

4.6 Distance Shrinking Algorithm

The original d-distances described above are derived by forcing one cell to exceed the threshold. Vectors that correspond to more than one cell exceeding the threshold have connections confined to a distance smaller than the original d-distances, and are thus rarer than vectors that produce one cell that exceeds the threshold. Thus, it is more efficient to target individual bins. Ideally, we would like to identify sets of distances $d_i(h)$ for $1 \leq h \leq D_{max}$ where h specifies the number of cells that exceed the threshold (multinomial bin number). Theoretically computing $d_i(h)$ is not possible because a given distance set $d_i(\cdot)$ may correspond to multiple bins; some bins may have identically zero probability; and there is no way of knowing *a priori* by how much a distance has to be reduced to cause one more cell to exceed the threshold. Note that this distance reduction applies to the connection starting-slots of the VBR source only.

Let the d-distances be specified by an ordered set $\Delta_r = \{d_0, d_1, \cdots, d_r\}$ for $N_C = N_{C_0}(\tau) + r$ connections, where $r \geq 0$. Distance shrinking involves identifying the worst case connection starting-slot configuration such that $N_{C_0}(\tau)$ connections start within d_0 of each other, $N_{C_0}(\tau) + 1$ connections start within d_1 of each other etc. The worst case configuration is identified by bursts producing the maximum congestion conforming to the set of distances Δ_r and not to any other set of distances Δ_r' which contains ordered elements $d_i' < d_i$. There are two configurations depending on whether $\mu < 1$ or $\mu \geq 1$.

The worst case situation conforming to Δ_r for $\mu < 1$ occurs when $N_{C_0}^p(\tau) - 1$ VBR connections start at slot 0, 1 VBR connection starts at each slot d_i for $0 \leq i \leq r$ and the N_{CBR} CBR connections start at slot 0. For $\mu \geq 1$, the worst case situation conforming to Δ_r occurs when 1 VBR connection starts at slot 0, $N_{C_0}^p(\tau) - 1$ VBR connections start at slot d_0, 1 VBR connection starts at each slot d_i for $1 \leq i \leq r$ and N_{CBR} CBR connections start at slot d_0.

The distance-shrinking algorithm consists of two phases. The first involves obtaining the number of cells that exceed the threshold for the worst case configuration by executing a presimulation run, which is equivalent to executing one realization of a vector drawn during a simulation run. The total number of presimulation runs used in this phase brings negligible overhead to the simulation. This process is repeated by shrinking the entries in the set Δ_r by one. The second phase uses the results of the first phase and identifies the ranges of bins to be collected for a given shrink value during simulation.

S	η_s	Bin Range	Lower Bin	Upper Bin
0	η_0	$1 \cdots \eta_0$	1	η_0
1	η_1	$\eta_0 + 1 \cdots \eta_1$	$\min(\eta_0 + 1, D_{max})$	$\min(\max(\eta_0 + 1, \eta_1), D_{max})$
2	η_2	$\eta_1 + 1 \cdots \eta_2$	$\min(\eta_1 + 1, D_{max})$	$\min(\max(\eta_1 + 1, \eta_2), D_{max})$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
S_{max}	D_{max}	$\eta_{S_{max}-1} + 1 \cdots D_{max}$	$\min(\eta_{S_{max}-1} + 1, D_{max})$	D_{max}

Table 1: Table generated by Phase 2 of the distance-shrinking algorithm.

In the following, we present the algorithm for the first phase. Here, S denotes the shrink value, η_S denotes the maximum number of cells that exceed the threshold for the worst case configuration conforming to the set Δ_r shrunk by S and $\underline{\omega}_{\Delta_r}$ denotes the worst case vector with elements $\omega_{\Delta_r}(1), \omega_{\Delta_r}(2) \cdots, \omega_{\Delta_r}(N_C)$ conforming to the set Δ_r shrunk by S:

<u>Phase 1</u> (Assume there are $N_C = N_{C_0}(\tau) + 1$ connections, where $r \geq 0$.)

- set $S = 0$ /* no shrinks initially */
- set $\delta_0 = d_0, \delta_1 = d_1, \cdots, \delta_r = d_r$ /* the original d-distances */
- set $\Delta_r = \{\delta_0, \delta_1, \cdots, \delta_r\}$ /* form the ordered set */
- form $\underline{\omega}_{\Delta_r}$ /* get the first set of worst case starting slots */
- do
  - run for vector $\underline{\omega}_{\Delta_r}$ to get η_s /* get # of cells exceeding τ */
  - for $i = 0$ to r,
    - if the i-th element, $\omega_{\Delta_r}(i) > 0$, $\omega_{\Delta_r}(i) = \omega_{\Delta_r}(i) - 1$ /* shrink */
  - $S = S + 1$ /* update shrink indicator */

  while $\eta_{s-1} \neq D_{max}$ /* phase 1 stopping condition */

Table 1 shows a typical output of the second phase of the algorithm. This table lists the shrink value S, the number of cells η_s that exceed the threshold for the worst case vector for shrink value S, the bin range to be collected for shrink value S and lower and upper bin ranges corrected for overflow conditions.

Let the set of vectors conforming to the ordered distance set generated by shrink value S be denoted by Ω_S for $0 \leq S \leq S_{max}$. Then, we have $\Omega_0 \supset \Omega_1 \supset \cdots \supset \Omega_S$ and $\Omega_0 \subset V$ since V also contains vectors in bin 0 which are not in Ω_0. Due to this relationship between the sets Ω_i, the data collection proceeds as follows: For $S = 0$, hits are recorded for bins 1 to D_{max}. For $S = 1$, hits are recorded for bins $\eta_0 + 1$ to D_{max} etc. Finally for $S = S_{max}$, hits are recorded for bins $\eta_{S-1} + 1$ to D_{max}. The shrink value is advanced when at least 10-100 hits are recorded for all the bins in the shrink range (for the χ^2 assumption [14]) or when a maximum run tolerance is reached.

Note that for a certain shrink S, the bin range hits are recorded for is from $\eta_{S-1} + 1$ to D_{max}. The reason for this is that the space Ω_S produced by restricting to the d-distances shrunk by S no longer contains all the vectors that produce 1 to η_{S-1} cells that exceed the threshold. If we did collect for bins 1 to η_{S-1} at shrink S, we would be violating the Importance Sampling condition.

Note that the sets Ω_i correspond to the sets V_i^* and Ω_0 corresponds to V^*. Any set Ω_i can be written as $\Omega_i = \Omega_{i,1} \times V_2$ since the biasing and distance-shrinking algorithms apply only to the VBR connection starting-slots. The set Ω_0 is generated by shrink value $S = 0$ and hence is defined by the original d-distances.

4.7 Improvement in Simulation Efficiency Using IS

The improvement in simulation efficiency using IS can be calculated by assuming the same estimator variance for both the MC estimate $\hat{P}_{TH}$ and the IS estimate $\hat{P}^*_{TH}$ and taking the ratio of the number of runs required to generate these estimates or, equivalently, by assuming the same number of MC and IS simulation runs and taking the ratio of the resulting estimator variances as follows:

$$R_{net} = \left.\frac{N_{MC}}{N_{IS}}\right|_{\sigma^2(\hat{P}_{TH})=\sigma^2(\hat{P}^*_{TH})} = \left.\frac{\sigma^2(\hat{P}_{TH})}{\sigma^2(\hat{P}^*_{TH})}\right|_{N_{MC}=N_{IS}} \tag{38}$$

Since our Importance Sampling procedure uses a different number of simulation runs for each bin, we use the latter approach in generating improvement factors. For any bin j, the number of IS simulation runs is n_j and the probability estimate is p_{TH_j}. If MC simulation were used for bin j, the estimate formed would involve a sum of Bernoulli trials on the hits on bin j, divided by the number of trials as in Eq. 8. Then, assuming the number of trials

Source	$\hat{\lambda}$, Mbps	$\bar{\lambda}$, Mbps	$\hat{B}$, cells	T, slots
CBR	10	10	1	1
VBR-1	50	5	25	250
VBR-2	500	2	50	12,500

Table 2: Sources for the heterogeneous systems.

System	Sources	$\hat{\mu}$, Mbps	K, cells	T_{SP}, slots	μ, cells/slot
A	CBR, VBR-1	120	200	250	2.4
B	CBR, VBR-2	120	200	12,500	0.24

Table 3: System parameters and derived parameters.

was n_j, an estimate of the variance for bin j is given by $\hat{p}_{TH_j}(1 - \hat{p}_{TH_j})/n_j$. Hence, we estimate the improvement as follows:

$$\hat{R}_{net} = \frac{\hat{\sigma}^2(\hat{P}_{TH})}{\hat{\sigma}^2(\hat{P}^*_{TH})} = \frac{\sum_{j=1}^{D_{max}} j^2 \hat{p}_{TH_j}(1 - \hat{p}_{TH_j})/n_j}{\sum_{j=1}^{D_{max}} \hat{\sigma}^2(\hat{p}^*_{TH_j})} \tag{39}$$

Here, D_{max} is known, n_j and $\hat{p}_{TH_j}$ and $\hat{\sigma}^2(\hat{p}^*_{TH_j})$ are the outputs of the IS simulation, where $\hat{\sigma}^2(\hat{p}^*_{TH_j})$ is calculated as in Eq. 19.

5 Experimental Results

The input parameters for the sources are listed in Table 2. We generated simulation results for two systems. System A consisted of CBR sources and VBR-1 sources. System B consisted of CBR sources and VBR-2 sources. The system parameters and derived parameters for these systems are listed in Table 3.

We first considered System A with three threshold levels, $\tau = 40\%$, $\tau = 45\%$ and $\tau = 50\%$, corresponding to queue lengths of 90, 100 and 110 cells, respectively. The number of VBR sources was fixed at 6 and the number of CBR sources was varied. The stopping condition

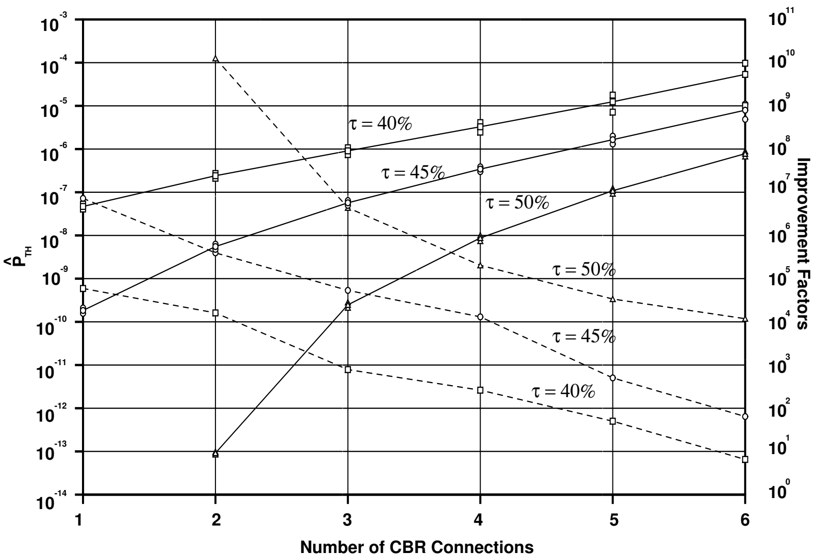

Figure 8: Aggregate delay threshold probability for system A, 6 VBR sources, number of CBR sources varying.

for the simulations was set at 100 hits per bin. In Fig. 8, the aggregate delay threshold probability curves are plotted as solid curves. The improvement factors (or simulation speedup) resulting from using Importance Sampling rather than conventional Monte Carlo simulation are plotted as dashed curves. The 95% confidence intervals are depicted with the same style point as the pertaining delay threshold probability curve.

For 6 VBR sources, one CBR source was sufficient for cells to exceed the thresholds of $\tau = 40\%$ and $\tau = 45\%$. For $\tau = 50\%$, at least two CBR sources were necessary for cells to exceed the threshold. As expected, the delay threshold probability increases when the number of CBR sources is increased and decreases when the delay threshold is increased. The difference between the aggregate delay threshold probability curves decreases as the number of CBR connections is increased. Also, the 95% confidence intervals widen as more CBR sources are added. This is due to the fact that, even though the probability being estimated increases as more CBR sources are added, so does D_{max} and hence more and more hits per bin are necessary to keep the estimator variance constant and throughout the simulations, the stopping conditions were set at 100 hits per bin.

Since the aggregate delay threshold probability increases as more CBR connections are

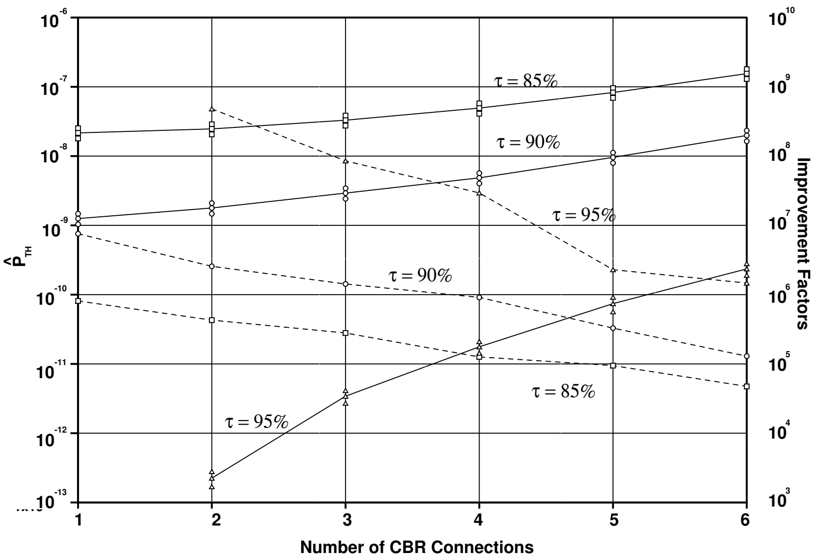

Figure 9: Aggregate delay threshold probability for system B, 4 VBR sources, number of CBR sources varying.

added, so do the improvement factors, as seen in Fig. 8. However, as seen, the improvement factor is inversely proportional to the probability being estimated. The improvement factor approaches 1 as the delay threshold probability exceeds 10^{-5} to 10^{-4}. Monte Carlo simulation is generally feasible for these probabilities, but will typically result in higher confidence intervals since it does not incorporate the distance-shrinking algorithm which targets individual bins.

Next, we considered system B with three threshold levels, $\tau = 85\%$, $\tau = 90\%$ and $\tau = 95\%$, corresponding to queue lengths of 170, 180 and 190 cells, respectively. The number of VBR sources was fixed at 4 and the number of CBR sources was varied. The results are plotted in Fig. 9.

For 4 VBR sources, one CBR source was sufficient for cells to exceed the thresholds of $\tau = 85\%$ and $\tau = 90\%$. However, at least 2 CBR sources were necessary for cells to exceed the threshold at $\tau = 95\%$. Similar behavior in terms of delay threshold probabilities and improvement factors were also observed for system B, where the improvement factors were again inversely proportional to the probability being estimated. However, the increase in delay threshold probability as more CBR sources are added is more gradual due to the large period of system B. Another result of this is that, at a fixed number of connections, the delay threshold probability decreases more and more as the threshold is increased, which can also be observed in Fig. 9.

6 Conclusion

In this paper, we considered the problem of estimating rare aggregate delay threshold probabilities in ATM networks with heterogeneous input traffic consisting of CBR and VBR sources. We used the ATM Forum standardized connection traffic descriptors to characterize the input traffic (operational approach).

For the heterogeneous model, we used a multinomial formulation which effectively removed correlations (due to burstiness) between delay threshold events, and produced a "multiple bin" simulation structure. We presented a three part Importance Sampling procedure based on this structure. The multiple bin structure and the Importance Sampling procedure were previously used for ATM switches with homogeneous input traffic.

For the experimental systems considered, we observed that the improvement in simulation efficiency (speedup over standard Monte Carlo simulation) was inversely proportional to the probability being estimated.

References

[1] J. A. Freebersyser and J. K. Townsend. Efficient Simulation of Cell Loss Probability Using Standardized ATM Connection Traffic Descriptors. In *Proc. IEEE Int. Conf. Commun., ICC '95*, pages 298–303, Seattle, June 1995.

[2] A. A. Akyamac and J. Keith Townsend. Efficient Simulation of Delay Threshold Probabilities in ATM Networks with ON/OFF Arrivals. To appear in the *Proc. IEEE Global Telecom. Conf., GLOBECOM '96*.

[3] J. A. Freebersyser and J. K. Townsend. Efficient Simulation of Cell Loss Probability in ATM Networks with Heterogeneous Connection Traffic Descriptors. In *Proc. IEEE Int. Conf. Commun., ICC '96*, pages 320–327, Dallas, June 1996.

[4] ATM User-Network Interface Specification, Version 3.0. The ATM Forum, September 10, 1993. The ATM Forum.

[5] A. W. Berger and A. E. Eckberg. A B-ISDN/ATM Traffic Descriptor, and its Use In Traffic and Congestion Control. In *Proc. IEEE Global Telecom. Conf., GLOBECOM '91*, Phoenix, AZ, May 1991.

[6] J. Garcia, J.M. Barcelo, and O. Casals. An Exact Model for the Multiplexing of Worst Case Traffic Sources. In *Proc. of the Sixth IFIP WGG.3 Conference on Performance of Computer Networks*, Istanbul, Turkey, 1995.

[7] J. Garcia and O. Casals. A Discrete-Time Queueing Model to Study the Cell Delay Variation in an ATM Network. *Performance Evaluation*, 21(1–2):3–22, 1994.

[8] K. Kvols and S. Blaabjerg. Bounds and Approximations for the Periodic ON/OFF Queue with Applications to ATM Traffic Control. In *Proc. of IEEE INFOCOM 1992*, Florence, Italy, March 1992.

[9] T. Worster. Modeling Deterministic Queues: The Leaky Bucket as an Arrival Process. In *14th International Teletraffic Congress*, Paris, June 1994.

[10] A. Skliros. A Connection Admission Control Algorithm for ATM Traffic Distorted by Cell Delay Variation. In *Proc. 14th Int. Teletraffic Congress, ITC 14*, 1994.

[11] B. T. Doshi. Deterministic Rule Based Traffic Descriptors for Broadband ISDN: Worst Case Behavior and Connection Acceptance Control. In *14th International Teletraffic Congress*, Paris, June 1994.

[12] A. A. Akyamac. Efficient Simulation of Delay Threshold Probabilities in ATM Networks with ON/OFF Arrivals Using Connection Traffic Descriptors. Master's thesis, North Carolina State University, Nov. 1995.

[13] A. A. Akyamac and J. K. Townsend. Formal Proof of Some Heuristics Used in the Importance Sampling Model to Estimate Rare Cell Delay Threshold Pobabilities. Technical Report TR-96/30, Center for Adv. Comput. and Comm., North Carolina State University, Aug. 1996.

[14] R. Z. Gold. Tests Auxiliary to χ^2 Tests in a Markov Chain. *Ann. Math. Statist.*, Vol. 34:56–74, 1963.