Efficient Simulation of Delay Threshold Probabilities in ATM Networks with ON/OFF Arrivals Using Connection Traffic Descriptors

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

Abstract

We consider the probability that the cell transfer delay exceeds a given threshold (delay threshold probability) for ATM networks. 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 based on closed form analysis. The multinomial formulation effectively removes the correlations associated with the bursty events. We then develop and demonstrate a three part Importance Sampling procedure to increase the efficiency of the simulation. For the experimental systems considered, we observe that the improvement in simulation efficiency is inversely proportional to the probability being estimated.
1 Introduction

In this paper, we consider the probability that the cell transfer delay exceeds a given threshold (delay threshold probability) in ATM networks. This measure has also been called the delay survivor function, the queue survivor function, the tail probability and the remote quantile in the literature. Another important QoS measure is the cell loss probability, which is not considered here (see [1]). These probabilities are rare and thus simulations are required to form estimates of the QoS measures.

One motivation for obtaining QoS measures as a function of input traffic and system parameters is that the technique presented here can be used to test connection admission control (CAC) algorithms.

Rather than using statistical models to characterize the input traffic, in this paper we use the connection traffic descriptors standardized by the ATM Forum. 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 [2].

Before the standardization of the ATM Forum traffic parameters, approaches considering traffic descriptors similar to those used here were used in [3], [4] and [5]. In [6], [7] and [8] the ATM descriptors were mentioned but only the first two were utilized. The three standardized descriptors (triplet) were used in [9], [10], [11], [12], [13], [14] [15] and [1].

The problem of calculating a delay threshold probability using infinite buffers and the operational approach incorporating the connection traffic descriptors has been considered before [6], [12], [10], [15]. However, [6] is restricted to traffic for which \( \hat{\lambda} = \bar{\lambda} \) and \( \hat{B} = 1 \). An exact solution is given in [12] for the classical multiplexing case in which \( N_C \hat{\lambda} < \hat{\mu} \) and conservative upper bounds are generated with the assumption that \( \hat{\mu} \) is an integer multiple of \( \hat{\lambda} \). In [10], an exact solution is found for the case in which \( \hat{\lambda} = \bar{\lambda} \). A general exact solution is found in [15], but restricted to only 2 connections.

In this paper, we develop a simulation model to estimate the delay threshold probability resulting from a single stage ATM switch with homogeneous input traffic. Since the probabilities involved are rare, we use Importance Sampling (IS) as a means of generating efficient
simulations. We estimate the delay threshold probability exactly for an infinite buffer situation and generate an upper bound for the case of finite buffers [16]. Our model incorporates all three ATM Forum standardized connection traffic descriptors and does not place any restrictions on the input traffic other than the fact that it has to be UPC compliant. Cell loss probability using this model is covered in [1].

This paper is organized as follows: Section 2 gives a description of the ATM switch we use in this study. We also describe the input traffic and present the slotted time simulation model we use in our estimations. In Section 3, we present an exhaustive formulation of the problem and develop a Monte Carlo (MC) simulation framework. The IS method that we use to speed up our simulations is described in Section 4. We give 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 \(N_C = N_P\). Each connection has the same triplet \((\lambda, \bar{\lambda}, \overline{B})\) (homogeneous traffic) and is routed uniformly to one of the output ports through a nonblocking shared bus that operates at a speed of \(N_P \bar{\lambda}\). We analyze a single “tagged” output buffer to represent the performance of all output buffers. The output
buffers have a finite size of $K$ cells, a service rate of $\mu$ 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 traffic. We consider worst case UPC-compliant traffic to be the periodic greedy ON/OFF traffic pattern. In the ON period, the source generates traffic at a peak rate of $\hat{\lambda}$ for a burst duration of $\hat{B}$, and the OFF period is required to average out to the mean rate $\bar{\lambda}$. 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 [10], [11], [12], [13], [14] and [1]. This traffic has been shown to cause more congestion in the case of finite buffers [13] and has been demonstrated to yield results very similar to the alternate worst cases [10], [15] under infinite buffers.

For our simulations, we use a slotted time model resulting from normalization with respect to the peak rate. With this normalization, the equivalent period in slots is $T = \hat{B}\hat{\lambda}/\bar{\lambda}$ arrival slots and the equivalent service rate is $\mu = \hat{\mu}/\bar{\lambda}$ cells/arrival slot. This approach is more general than normalization with respect to the service rate since we do not require $\hat{\mu}$ to be an integer multiple of $\hat{\lambda}$. An example of the resulting slotted-time pattern is plotted in Fig. 2 for $\hat{B} = 4$ and $T = 16$. The randomness in the simulation model is found in the connection starting-slot positions. If we look at a fixed window of size $T$ arrival slots, then each of the $N_C$ connections is characterized by a discrete-time random variable, uniform on $[0, T-1]$, representing the slot in which that connection starts its $\hat{B}$-length burst. We represent the starting-slots as an $N_C$-dimensional vector $\underline{v}$. For simplicity, we fix one 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. 3. 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 $\mu/N_C$ cells can be serviced. Cell losses occur when there is no space left in the output buffer to accommodate the incoming cell. For a cell that is not lost, the delay threshold $\tau$ 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 $\tau$ and $K_m$ interchangeably in this paper.

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. As a solution, we consider an arbitrary connection starting-slot vector $\mathbf{v}$ that corresponds to $l$ cell losses and $k$ nonlost cells which exceed the given threshold $\tau$ in steady state. For an infinite buffer, $k$ cells will exceed the threshold, the $l$ additional cells that would normally be lost will exceed the threshold and $x$ residual cells will exceed the threshold due to the fact that the $l$ cells not lost add to the overall congestion by remaining in the buffer. So, if we denote the number of cells that exceed the threshold by $c$, then $c_{\text{finite}} = k$ and $c_{\infty} = k + l + x$, $x \geq 0$. Hence, we have $c_{\text{finite}} \leq c_{\infty} - l$. Thus, for the delay threshold probability $\Pr(D > \tau)$, we have:

$$\Pr(D_K > \tau) \leq \Pr(D_{\infty} > \tau) - \Pr(\text{cell loss})$$ (1)
where the subscripts $K$ and $\infty$ refer to finite and infinite buffers, respectively.

The expression in (3.1) 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 simplicity. The cell loss estimates have been considered in a similar context in [1] and the results found there will be used to generate the overall finite buffer upper bound estimate.

### 3.2 Exhaustive Solution to $P_{TH}$

Let $D_{\text{max}}$ be the maximum number of cells that can exceed the given delay threshold for the infinite buffer. The worst case that can occur is when all connections begin their $\hat{B}$-length bursts in the same arrival slot, which will result in the maximum number of cells that exceed the threshold. Due to the burst at slot 0, this situation corresponds to the aligned-at-zero (AAZ) case (Shown in Fig. 4). Note that the AAZ case is not the only situation that will produce $D_{\text{max}}$ cells that exceed the threshold, but it is the only case for which $D_{\text{max}}$ cells are guaranteed to exceed the threshold. The value of $D_{\text{max}}$ can be found by computing the number of cells that exceed the threshold for the AAZ case. Immediately after $N_A = [(K_m - 1)/(1 - \mu/N_C)] + 1$ cell arrivals, the queue length $Q$ will be $K_m - 1 < Q(N_A) \leq K_m$. All subsequent cells will cause the queue length to exceed the threshold. Hence, $D_{\text{max}}$ is given by:

$$D_{\text{max}}(N_C) = N_C \hat{B} \left\lceil \frac{K_m - 1}{1 - \frac{\mu}{N_C}} \right\rceil - 1$$

(2)

where $\lceil \cdot \rceil$ denotes the ceiling operator. Note the dependence of $D_{\text{max}}$ on $N_C$. 

---

Figure 4: AAZ case for $N_C(\tau)$ connections in terms of arrival slots. The number of arrivals is $N_C = N_C(\tau)$. In this figure there are 5 connections with $\hat{B} = 8$. 

---
Let \( V \) be the set of all connection starting-slot vectors and \( n_{TH} \) be the number of connection starting-slot vectors that map to exactly \( j \) cells that exceed the threshold \( \tau \) in steady state. Here \( |V| = T^{N_C-1} \), where \( | \cdot | \) denotes the cardinality of the set. Thus, 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, the delay threshold probability for the infinite buffer is given by:

\[
P_{TH} = \sum_{j=0}^{D_{max}} \left( \frac{j}{N_C B} \right) p_{TH_j}
\]

As seen, the delay threshold probability is expressed as a weighted sum of multinomial probabilities. This multinomial framework removes correlation and allows us to compute confidence intervals for the estimates.

The exhaustive solution to this problem is obtained by summing important events over the entire set \( V \) as follows:

\[
P_{TH} = \frac{1}{|V| N_C B} \sum_{i_2=0}^{T-1} \sum_{i_3=0}^{T-1} \cdots \sum_{i_{N_C}=0}^{T_{SS}-1} \sum_{j=0}^{N_C} \sum_{n=1}^{N_C} I_{TH}(0, i_2, i_3, \cdots , i_{N_C}, j, n)
\]

where \( I_{TH} \) is the binary indicator function of the event that a cell exceeds the threshold \( \tau \) in steady state and \( T_{SS} \) is a steady state period. This exhaustive expression has a complexity of \( O(N_C T^{N_C}) \) which quickly becomes intractable for realistic systems.

### 3.3 Monte Carlo Simulation

Using Monte Carlo simulation, we obtain an unbiased estimate for \( P_{TH} \) by first individually estimating the probabilities \( p_{TH_j} \) and then forming the overall estimate \( \hat{P}_{TH} \):

\[
\hat{P}_{TH} = \frac{1}{N_C B} \sum_{j=1}^{D_{max}} \sum_{j=1}^{D_{max}} \hat{p}_{TH_j}
\]

We use a multinomial framework instead of the binomial framework in (4) to obtain independence between important events. The binary indicator function results in correlated
distributions (see [17]) and we do not have an \textit{a priori} method of determining this correlation. Estimating this correlation via simulation is computationally prohibitive.

The individual estimates $\hat{P}_{TH_j}$ of $P_{TH_j}$ are found by running $N_{MC}$ simulation runs, with one connection starting-slot vector drawn in each run uniformly from the sample space $V$. We have:

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

where $I_j(s)$ is the indicator function for $j$ cells exceeding the threshold in steady state for vector $s$. We will refer to the original probability structure $\hat{P}_{TH_j}$ as being composed of bins numbered 0 to $D_{max}$, where bin $j$ corresponds to $j$ cells that exceed the threshold. Note that even though $\hat{P}_{TH}$ has a multinomial form, the individual estimates $\hat{p}_{TH_j}$ are generated using binomial indicator functions. The probabilities $p_{TH_j}$ are binomial in that either $j$ cells exceed the threshold or not and are not necessarily uncorrelated. But, since realistic ATM systems have small delay threshold probabilities the “majority” of the vectors will be located in bin 0. So, if $j$ cells do not exceed the threshold, then most of the time 0 cells will and hence we consider the bins to be independent since bin 0 is not included in our estimates.

The actual variance of the MC estimate $\hat{P}_{TH}$ is given by:

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

(7)

We do not know the probabilities $p_{TH_j}$ \textit{a priori} and hence we form an unbiased estimate of the estimator variance of $\hat{P}_{TH}$ is as follows:

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

(8)

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

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

(9)

For the confidence intervals we use the result that the confidence interval of a weighted sum of individual multinomial probabilities follows a $\chi^2$ distribution [18], stating that the probability that $P_{TH}$ satisfies $\hat{P}_{TH} - Z\hat{\sigma}(\hat{P}_{TH}) \leq P_{TH} \leq \hat{P}_{TH} + Z\hat{\sigma}(\hat{P}_{TH})$ is at least $1 - \alpha$ where
$Z$ is the positive square root of the upper $(1 - \alpha)$th percentage point of the $\chi^2$ distribution with $D_{\text{max}}$ degrees of freedom. Using the $\chi^2$ 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 $N_C$

There are two types of situations for which the end-to-end delay of cells can exceed the given threshold. We will analyze these situations to generate a proper range for $N_C$ for simulations.

The first type occurs when individual connections start so close together that the server can no longer keep up with the arrivals resulting in congestion which leads to cells exceeding the threshold. For a given threshold $\tau$ (corresponding to a queue length of $K_m$), there is a minimum number of connections $N_{C_0}(\tau)$ required to cause at least one cell to exceed the threshold. This minimum number is found by observing the AAZ case at $N_{C_0}(\tau)$ connections and forcing the last cell arrival to exceed the threshold (the shaded cell in Fig. 4). We have $B(N_{C_0}(\tau) - \mu) + \mu/N_{C_0}(\tau) > K_m$. Solving the quadratic inequality for the first type, we find

$$N_{C_0}(\tau) = \left[ \frac{\mu + K_m}{B} + \sqrt{(\mu + \frac{K_m}{B})^2 - \frac{4\mu}{B}} \right] + 1$$(10)

The second type occurs when there are so many connections into the system that a cell exceeding the threshold is guaranteed for all connection starting-slot vectors. This occurs when the sum of the average rates is greater than the service rate, $\bar{\lambda}N_C > \bar{\mu}$, meaning that all cells will eventually exceed the threshold in steady state. Therefore for the second type, $\bar{\lambda}N_C \leq \bar{\mu}$ (or equivalently the utilization $\rho = N_C\bar{\lambda}/\bar{\mu} \leq 1$).

So, the useful range for the number of connections is given by $N_{C_0}(\tau) \leq N_C \leq \lceil \bar{\mu}/\bar{\lambda} \rceil$.

### 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) \( f_V(v) \) to \( f^*_V(v^*) \) such that the estimate is formed with this new pdf. 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.

To keep \( \hat{P}_{TH} \) unbiased, each important event (identified by cells that exceed the threshold) must be appropriately weighted or “unbiased”. Additionally, we require that the biased pdf \( f^*_V(v^*) > 0 \) whenever the original pdf \( f_V(v) > 0 \).

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.

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 in \( V_{TH} \) results in at least one cell exceeding 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:

$$\hat{P}_{TH}^* = \sum_{j=1}^{D_{max}} \left( \frac{j}{N_C B} \right) \hat{p}_{TH,j}^*$$

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

$$\hat{p}_{TH,j}^* = \frac{1}{n_j} \sum_{s=1}^{n_j} I_j^*(s) w_j^*(s)$$

where $I_j^*(s)$ is the indicator function of a cell exceeding the threshold under the new pdf $f_{V,j}^*(\cdot)$ and $w_j^*(s)$ is a vector-dependent weight function used to unbiased the estimate. For run $s$, the weight for vector $\ux_j^*(s)$ is given by $w_j^*(s) = f_V(\ux_j^*(s))/f_{V,j}^*(\ux_j^*(s))$ 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:

$$\hat{\sigma}^2(\hat{P}_{TH}^*) = \frac{1}{(N_C B)^2} \sum_{j=1}^{D_{max}} j^2 \hat{\sigma}^2(\hat{p}_{TH,j}^*)$$

where

$$\hat{\sigma}^2(\hat{p}_{TH,j}^*) = \frac{1}{n_j(n_j - 1)} \sum_{s=1}^{n_j} (I_j^*(s)w_j^*(s) - \hat{p}_{TH,j}^*)^2$$
4.2 Distance Calculations for the First Bin

The function of the distance calculation algorithm is to identify sets that contain connection starting-slot vectors that cause important events, i.e. the set $V^*$. In this first algorithm, we generate distances for the first bin (bin 1).

At least $N_{C_0}(\tau)$ connections are necessary so that a cell can exceed the given threshold $\tau$ (corresponding to a queue length of $K_m$). For $N_{C_0}(\tau)$ connections, we can determine the distance within which all connections must start their bursts so that a cell can exceed the threshold since a certain amount of congestion is required to fill the queue. We denote this distance as $d_0$ where the subscript emphasizes that there are $N_{C_0}(\tau)$ connections. Unless all connections start their bursts within $d_0$ slots of each other, no cells will exceed the threshold.

The value of $d_0$ depends on the relative values of $N_{C_0}(\tau)$ and $\mu$ and is calculated by observing configurations that will just cause one cell to exceed the threshold. The relative values of $N_{C_0}(\tau)$ and $\mu$ generate three different cases. These cases differ in the amount of service the server can provide in an arrival slot:

1. $\mu < 1$, one cell in an arrival slot can increase the queue length for that arrival slot.

2. $\mu \geq 1$, $N_{C_0}(\tau) - 1 \geq \lceil \mu \rceil + 1$, one cell in an arrival slot can no longer increase the queue length for that arrival slot, but $N_{C_0}(\tau) - 1$ cells can.

3. $\mu \geq 1$, $N_{C_0}(\tau) - 1 \leq \lfloor \mu \rfloor$, at least $N_{C_0}(\tau)$ cells in an arrival slot are required to increase the queue length for that arrival slot.

For the three cases, we have identified the extreme starting-slot configurations to cause one cell to exceed the threshold. For $N_{C_0}(\tau)$ connections, these configurations are shown in Fig's. 6, 7 and 8 for Cases 1, 2 and 3, respectively. In these figures, the upper axis indicates the extreme burst configuration generated by $N_{C_0}(\tau)$ connections and the lower axis indicates the change in queue length due to the burst configuration in the upper axis. As seen, the extreme configurations for all three cases are identified by a block of $N_{C_0}(\tau) - 1$ bursts and a single burst offset by a distance of $d_0$ from this block. The shaded cell in this
Figure 6: Extreme configuration for $\mu < 1$, $N_{C_0}(\tau)$ connections.

Figure 7: Extreme configuration for $\mu \geq 1$, $N_{C_0}(\tau) - 1 \geq \lfloor \mu \rfloor + 1$, $N_{C_0}(\tau)$ connections.

Figure 8: Extreme configuration for $\mu \geq 1$, $N_{C_0}(\tau) - 1 \leq \lfloor \mu \rfloor$, $N_{C_0}(\tau)$ connections.
offset burst is the cell forced to exceed the threshold such that if the single burst were offset by one more slot, no cells would exceed the threshold.

For example, for Case 1, the queue length just after the arrival of the shaded cell must satisfy \( \hat{B}(N_{C_0}(\tau) - 1 - \mu) - \mu (d_0 - \hat{B}) + (1 - \mu)(\hat{B} - 1) + 1 > K_m \). Hence, we obtain an inequality for \( d_0 \) and since \( d_0 \) is an integer, we have:

\[
d_0 = \left\lceil \frac{\hat{B}(N_{C_0}(\tau) - \mu) - K_m}{\mu} + 1 \right\rceil - 1
\]  

(15)

The value of \( d_0 \) for the other two cases is found similarly and the results are given in Table 1.

As more connections are added to the system, we can identify other distances \( d_0, d_1, \ldots, d_r \) such that important events can be caused in \( r \) ways. These are identified by either \( N_{C_0}(\tau) \) connections starting within \( d_0 \) slots of each other or \( N_{C_0}(\tau) + 1 \) connections starting within \( d_1 \) slots of each other etc. Finally, important events can be caused by \( N_{C_0}(\tau) + r \) connections starting within \( d_r \) slots of each other. If the connection starting-slots cannot be confined in any way to the above \( r \) distances, then no important events will occur. The subsequent distances are found by adding more offset single bursts to the right of the rightmost burst to cause extreme configurations similar to the previous ones. For Case 1, we can calculate \( d_0, d_1, \ldots, d_r \) exactly and we observe that the subsequent distances are found by adding equidistant offset bursts. However, for Cases 2 and 3, we can no longer find analytic expressions for the subsequent distances due to the fact that more than one cell can be serviced in each arrival slot. We can upper bound the subsequent distances by adding \( \hat{B} \) to the previous distance. The subsequent distance values are also shown in Table 1.
If at any point in the distance calculation procedure a distance $d$ exceeds half the period $T/2$, IS is no longer beneficial. However such cases correspond to high probabilities for which conventional MC simulation is feasible.

At a given number of connections $N_C = N_{C_0}(\tau) + r$, $V^*$ is the set of all vectors for which the starting-slots can be confined to any of the $r$ distances $d_0, d_1, \cdots, d_r$. Any vector for which the starting-slots cannot be confined to these $r$ distances is excluded from $V^*$.

### 4.3 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.

For $N_C = N_{C_0}(\tau) + r$ connections, the interval reduction algorithm starts with the burst at slot 0 and initially randomly samples $r$ connection starting-slot vectors over the entire interval. It then generates valid intervals around the existing connections (according to the $d$-distances) that can potentially result in cells that exceed the threshold with the addition of the next sampled connection starting-slot. The algorithm keeps track of the intervals $\hat{t}_i$ over which the connection starting-slots are samples and continues iteratively until all connection starting-slots have been sampled. At the end of each iteration, the total support for the subsequent connection starting-slot is reduced, i.e. $\hat{t}_{N_{C_0}(\tau)+r-1} \leq \hat{t}_{N_{C_0}(\tau)+r-2} \leq \cdots \leq \hat{t}_2 \leq \hat{t}_1$.

For a vector $v^*$ produced by the interval reduction method, the IS weight is given as the ratio $f_{\tilde{v}}(v) / f_{\tilde{v}^*}(v^*)$. Since the distributions are uniform, the weight will be the ratio of the product of supports used by the IS scheme to the ones used by the MC scheme, which samples from the whole period for each connection. Thus, the IS weight is calculated as:

$$w^*(v^*) = \frac{1}{T} \prod_{k=1}^{N_{C_0} - 1} \hat{t}_k$$

### 4.4 Distance Shrinking Algorithm

The original $d$-distances described above are derived by forcing one cell to exceed the threshold. In the multiple bin structure of the multinomial approach, vectors that correspond...
to more than one cell exceeding the threshold will invariably have connections confined (in
terms of starting-slots) to a distance smaller than the original $d$-distances. In fact, in most
cases, the vectors produced by the original $d$-distances will pertain to bins 0 and 1 (i.e., result
in either no cells or 1 cell exceeding the threshold). Thus, simulating for the higher bins
with the original distances could prove to be very inefficient since multiple hits are required
for all bins to accurately determine statistics of the multinomial.

Ideally, we would like to identify sets of distances $d_i(h)$ for $1 \leq h \leq D_{\text{max}}$ where $h$
specifies the number of cells that exceed the threshold (multinomial bin number). We have
observed that theoretically computing $d_i(h)$ is not possible for all cases. This is 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.

Let the $d$-distances be specified by an ordered set $\Delta_r = \{d_0, d_1, \ldots, d_r\}$ 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 config-
uration is identified by bursts producing the maximum congestion conforming to the set
of distances $\Delta_r$ and not to any other set of distances $\Delta'_r$ which contains ordered elements
$d'_i < d_i$. This configuration depends on the value of $\mu$.

The worst case configurations for $\mu < 1$ and $\mu \geq 1$ are shown in Fig’s. 9 and 10
respectively. As seen, the worst case situation conforming to $\Delta_r$ for $\mu < 1$ occurs when
$N_C - r - 1 = N_{C_0}(\tau) - 1$ connections start at slot 0 and one connection starts at each slot
$d_i$ for $0 \leq i \leq r$. For $\mu \geq 1$, the worst case situation conforming to $\Delta_r$ occurs when 1
connection starts at slot 0, $N_C - r - 1 = N_{C_0}(\tau) - 1$ connections start at slot $d_0$ and one
connection starts at each slot $d_i$ for $1 \leq i \leq r$.

The distance-shrinking method consists of two phases. The first of these involves obtaining
the number of cells that exceed the threshold for the worst case configuration by running
a presimulation algorithm. This step is equivalent to executing one realization of a vector
Figure 9: Worst case configuration conforming to $\Delta_r$ for $\mu < 1$, $N_C = N_{C_0}(\tau) + r$ connections.

Figure 10: Worst case configuration conforming to $\Delta_r$ for $\mu \geq 1$, $N_C = N_{C_0}(\tau) + r$ connections.

drawn during a simulation run. This process is repeated by shrinking the entries in the set $\Delta_r$ 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. The second phase of the algorithm also has to keep track of overflows for bin ranges. These can occur if the upper range value exceeds $D_{max}$ or the lower range value exceeds the upper range value.

The algorithm for the first phase follows, where $S$ denotes the shrink value, $\eta_S$ denotes the maximum number of cells that exceed the threshold for the worst case configuration conforming to the set $\Delta_r$ shrunk by $S$ and $\omega_{\Delta_r}$ 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 $\Delta_r$ shrunk by $S$:

**Phase 1** (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 $\omega_{\Delta_r}$ /* get the first set of worst case starting slots */
- do
  - run for vector $\omega_{\Delta_r}$ to get $\eta_S$ /* get # of cells exceeding $\tau$ */
  - for $i = 0$ to $r$, ...
Table 2: Table generated by Phase 2 of the distance-shrinking algorithm.

<table>
<thead>
<tr>
<th>$S$</th>
<th>$\eta_s$</th>
<th>Bin Range</th>
<th>Lower Bin</th>
<th>Upper Bin</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>$\eta_0$</td>
<td>$0 \cdots \eta_0$</td>
<td>$\min(\eta_0 + 1, D_{max})$</td>
<td>$\max(\eta_0 + 1, \eta_1), D_{max}$</td>
</tr>
<tr>
<td>1</td>
<td>$\eta_1$</td>
<td>$\eta_0 + 1 \cdots \eta_1$</td>
<td>$\min(\eta_1 + 1, D_{max})$</td>
<td>$\max(\eta_1 + 1, \eta_2), D_{max}$</td>
</tr>
<tr>
<td>2</td>
<td>$\eta_2$</td>
<td>$\eta_1 + 1 \cdots \eta_2$</td>
<td>$\ldots$</td>
<td>$\ldots$</td>
</tr>
<tr>
<td>\cdots</td>
<td>\cdots</td>
<td>\cdots</td>
<td>\cdots</td>
<td>\cdots</td>
</tr>
<tr>
<td>$S_{max}$</td>
<td>$D_{max}$</td>
<td>$\eta_{S_{max} - 1} + 1 \cdots D_{max}$</td>
<td>$\min(\eta_{S_{max} - 1} + 1, D_{max})$</td>
<td>$D_{max}$</td>
</tr>
</tbody>
</table>

* if the $i$-th element, $\omega_{\Delta_+}(i) > 0$, $\omega_{\Delta_+}(i) = \omega_{\Delta_+}(i) - 1$  /* shrink */
- $S = S + 1$  /* update shrink indicator */
while $\eta_{s-1} \neq D_{max}$  /* phase 1 stopping condition */

Note that the presimulation runs in the first phase bring negligible overhead to the overall simulation. The second phase produces results as in Table 2. Listed in this table are the shrink value $S$, the number of cells $\eta_s$ 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. If more than one shrink contains the same bin range, then the data is collected for that bin range with the maximum shrink.

Let the set of vectors conforming to the ordered distance set generated by shrink value $S$ be denoted by $\Omega_S$ 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 $\Omega_0$. Due to this relationship between the sets $\Omega_i$, the data collection proceeds as follows: For $S = 0$, data is collected for bins 1 to $D_{max}$ and the stopping conditions are specified with respect to bins 1 to $\eta_0$. For $S = 1$, data is collected for bins $\eta_0 + 1$ to $D_{max}$ and the stopping conditions are specified with respect to bins $\eta_0 + 1$ to $\eta_2$ etc. Finally for $S = S_{max}$, data is collected for and stopping conditions are specified with respect to bins $\eta_{S-1} + 1$ to $D_{max}$.

The distance shrinking algorithm was developed for the following reason. For two sets $\Omega_i \subset \Omega_j$ it is possible that the set $\Omega_j$ contains all vectors for a bin $b$ and the set $\Omega_i$ does not. Since it is required that $f_{\Delta_+}^{*}(\bar{v}^*) > 0$ whenever $f_{\Delta_+}(\bar{v}) > 0$, hits for bin $b$ should not be collected from the set $\Omega_i$. 

17
Note that the sets $\Omega_i$ correspond to the sets $V_i^{*}$ and $\Omega_0$ corresponds to $V^{*}$. The set $\Omega_0$ is generated by shrink value $S = 0$ and hence is defined by the original $d$-distances.

There are two types of stopping conditions. The first of these is a minimum number of hits per bin. This number is typically 5 to 10, which makes the $\chi^2$ assumption valid [18]. The second of these is a maximum tolerance on the number of runs per shrink or a maximum number of hits for all nonzero bins in a given shrink. This is to ensure the algorithm does not get stuck in ranges for which there are bins with zero probability.

4.5 Improvement in Simulation Efficiency Using IS

We calculate the improvement in simulation efficiency using IS 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:

$$R_{net} = \frac{N_{MC}}{N_{IS}}$$

(17)

Exchanging the terms $\sigma^2(\hat{P}_{TH})$ and $N_{MC}$ in the MC variance expression in (3.3) and replacing $\sigma^2(\hat{P}_{TH})$ with $\sigma^2(\hat{P}_{TH}^{*})$ and subsequently replacing $p_{TH,j}$ and $\sigma^2(\hat{P}_{TH}^{*})$ with their respective estimates $\hat{p}_{TH,j}$ and $\hat{\sigma}^2(\hat{P}_{TH}^{*})$, we obtain an expression for the estimate of the improvement:

$$\hat{R}_{net} = \frac{N_{MC}}{N_{IS}} = \frac{\sum_{j=1}^{D_{max}} \left( \frac{j}{N_{C}} \right)^2 \hat{p}_{TH,j} \left( 1 - \hat{p}_{TH,j} \right)}{\hat{\sigma}^2(\hat{P}_{TH}^{*}) N_{IS}}$$

(18)

Here, $N_{C}$, $B$ and $D_{max}$ are known a priori and $\hat{p}_{TH,j}$, $\hat{\sigma}^2(\hat{P}_{TH}^{*})$ and $N_{IS}$ are all outputs of the IS simulation.

5 Experimental Results

We first consider two simple examples to demonstrate our IS method. We then consider a variety of experimental systems with realistic system parameters.
Table 3: Parameters for the simple systems.

<table>
<thead>
<tr>
<th>System</th>
<th>Input Parameters</th>
<th>Derived System Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>λ, Mbps</td>
<td>ʋ, Mbps</td>
</tr>
<tr>
<td>A1</td>
<td>500</td>
<td>30</td>
</tr>
<tr>
<td>A2</td>
<td>8</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 4: Comparison of exhaustive results and IS simulation results for system A1, 3 connections.

<table>
<thead>
<tr>
<th>Shrink S</th>
<th>Bin j</th>
<th>Exhaustive Results</th>
<th>IS Simulation Results</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td># of vectors/bin</td>
<td>P_{TH_j}</td>
</tr>
<tr>
<td>N/A</td>
<td>0</td>
<td>381</td>
<td>0.9625</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>1</td>
<td>0.0075</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>2</td>
<td>0.0375</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>3</td>
<td>0.0025</td>
</tr>
<tr>
<td>Overall</td>
<td></td>
<td>P_{TH} = 1.300 × 10^{-2}</td>
<td>Overall</td>
</tr>
<tr>
<td></td>
<td></td>
<td>95% CI = [1.495 × 10^{-2}, 1.503 × 10^{-2}]</td>
<td>R_{net} = 4.870 × 10^{1}</td>
</tr>
</tbody>
</table>

5.1 Simple Examples

We consider two simple systems A1 and A2 with input parameters and derived system parameters given in Table 1. These systems are such that the space \( V \) is tractable and exhaustive results are feasible to determine. We set the delay threshold at \( K_m = 0.75K \) and \( K_m = 0.8K \) for systems A1 and A2, respectively. All these threshold levels, \( N_{C_0}(\tau) = 3 \) for both systems. Also, \( d_0 = 2, d_1 = 5 \) for system A1 and \( d_0 = 2, d_1 = 6 \) for system A2.

The stopping conditions were set to 100 hits per bin and a maximum of 1000 hits for each nonzero bin.

The exhaustive results and the IS simulation results are shown in Tables 4, 5, 6 and 7 for both systems. In each table, the first section lists the shrink value for each bin, the number of vectors per bin, the exact probabilities of the bins and the overall probability \( P_{TH} \). The second section lists the total number of IS runs for each bin, the estimated probability for each bin, the overall estimate \( \hat{P}_{TH} \), the 95% confidence interval and the estimated improvement over standard MC simulation. Note that data is not collected for bin 0 during simulation.

The following observations can be made from the results: An increase in the number of connections results in a higher \( D_{max} \) and a higher delay threshold probability. The number
### Table 5: Comparison of exhaustive results and IS simulation results for system A1, 4 connections.

<table>
<thead>
<tr>
<th>Shrink S</th>
<th>Bin</th>
<th># of vectors/bin</th>
<th>( p_{TH} )</th>
<th>IS Simulation Results</th>
<th>( n_i )</th>
<th># of hits</th>
<th>( p_{TH} )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>6.000 \times 10^9</td>
<td>data not collected for bin 0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0</td>
<td>6.000 \times 10^9</td>
<td>1.000 \times 10^{10}</td>
<td>100</td>
<td>1.172 \times 10^{-2}</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>6.000 \times 10^9</td>
<td>1000 \times 10^{10}</td>
<td>2917</td>
<td>3.510 \times 10^{-2}</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0</td>
<td>6.000 \times 10^9</td>
<td>1605 \times 10^{10}</td>
<td>2349</td>
<td>2.349 \times 10^{-2}</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0</td>
<td>6.000 \times 10^9</td>
<td>517 \times 10^{10}</td>
<td>3830</td>
<td>3.830 \times 10^{-2}</td>
<td></td>
</tr>
<tr>
<td>Overall</td>
<td></td>
<td></td>
<td></td>
<td>( P_{TH} = 2.018 \times 10^{-2} )</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 6: Comparison of exhaustive results and IS simulation results for system A2, 3 connections.

<table>
<thead>
<tr>
<th>Shrink S</th>
<th>Bin</th>
<th># of vectors/bin</th>
<th>( p_{TH} )</th>
<th>IS Simulation Results</th>
<th>( n_i )</th>
<th># of hits</th>
<th>( p_{TH} )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>6.000 \times 10^9</td>
<td>data not collected for bin 0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0</td>
<td>6.000 \times 10^9</td>
<td>1.000 \times 10^{10}</td>
<td>100</td>
<td>1.303 \times 10^{-2}</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>6.000 \times 10^9</td>
<td>1000 \times 10^{10}</td>
<td>217</td>
<td>2.324 \times 10^{-2}</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>6.000 \times 10^9</td>
<td>665 \times 10^{10}</td>
<td>389</td>
<td>5.580 \times 10^{-2}</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>6.000 \times 10^9</td>
<td>389 \times 10^{10}</td>
<td>5560</td>
<td>5.560 \times 10^{-2}</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>6.000 \times 10^9</td>
<td>120 \times 10^{10}</td>
<td>120</td>
<td>2.444 \times 10^{-2}</td>
<td></td>
</tr>
<tr>
<td>Overall</td>
<td></td>
<td></td>
<td></td>
<td>( P_{TH} = 8.023 \times 10^{-2} )</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 7: Comparison of exhaustive results and IS simulation results for system A2, 4 connections.

<table>
<thead>
<tr>
<th>Shrink S</th>
<th>Bin</th>
<th># of vectors/bin</th>
<th>( p_{TH} )</th>
<th>IS Simulation Results</th>
<th>( n_i )</th>
<th># of hits</th>
<th>( p_{TH} )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>6.000 \times 10^9</td>
<td>data not collected for bin 0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0</td>
<td>6.000 \times 10^9</td>
<td>1.000 \times 10^{10}</td>
<td>100</td>
<td>1.303 \times 10^{-2}</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>6.000 \times 10^9</td>
<td>1000 \times 10^{10}</td>
<td>217</td>
<td>2.324 \times 10^{-2}</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>6.000 \times 10^9</td>
<td>665 \times 10^{10}</td>
<td>389</td>
<td>5.580 \times 10^{-2}</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>6.000 \times 10^9</td>
<td>389 \times 10^{10}</td>
<td>5560</td>
<td>5.560 \times 10^{-2}</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>6.000 \times 10^9</td>
<td>120 \times 10^{10}</td>
<td>120</td>
<td>2.444 \times 10^{-2}</td>
<td></td>
</tr>
<tr>
<td>Overall</td>
<td></td>
<td></td>
<td></td>
<td>( P_{TH} = 8.028 \times 10^{-2} )</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

95% CI = [7.984 \times 10^{-2}, 8.071 \times 10^{-2}] 
\( R_{net} = 1.503 \times 10^6 \)
of IS runs required to obtain the same number of hits per bin increases as \( D_{\text{max}} \) is increased. So, more IS runs are required to obtain the same accuracy for a higher estimate. This is due to the multinomial structure of the problem. An increased number of bins means that more runs are necessary to obtain a given number of hits per bin. The improvement of IS over MC also decreases as more connections are added to the system. Specifically for system A2 at 5 connections, \( d_2 = 10 \) and IS simulation breaks down to MC simulation, hence the improvement becomes unity. This is not the case for system A1 for which \( d_2 = 8 \), but at 5 connections, the sample space \( V \) is intractably large for exhaustive simulation. As seen from Table 4, bin 1 has identically 0 probability for system A2 at 3 connections.

### 5.2 Realistic Systems

The input parameters and derived simulation parameters for the realistic systems are listed in Table 8. We generated simulation results by varying \( \tau, \tilde{B}, \lambda \) and \( \tilde{\lambda} \).

We considered system F with two threshold levels \( \tau_1 = 90\% \) and \( \tau_2 = 95\% \) which correspond to queue lengths of 180 and 190 slots, respectively. Here, \( N_{C_0}(\tau) = 8 \) connections for both threshold levels. The simulation results are plotted in Fig. 11 and listed in Table 9. We observe an increase in the delay threshold probability as the threshold is decreased and as more connections are added. We also observe that the IS improvement (or speedup) over MC decreases and that the confidence intervals widen as more connections are added to the system. For \( \tau_1 = 90\% \), IS lost its efficiency at 9 and 10 connections and MC was used. For the IS cases, the speedup factors are inversely proportional to the probability being estimated.

<table>
<thead>
<tr>
<th>System</th>
<th>Input Parameters</th>
<th>Derived System Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( \lambda, \text{Mbps} )</td>
<td>( \lambda, \text{Mbps} )</td>
</tr>
<tr>
<td>B</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>C</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>D</td>
<td>500</td>
<td>50</td>
</tr>
<tr>
<td>E</td>
<td>500</td>
<td>50</td>
</tr>
<tr>
<td>F</td>
<td>500</td>
<td>50</td>
</tr>
<tr>
<td>G</td>
<td>500</td>
<td>50</td>
</tr>
<tr>
<td>H</td>
<td>400</td>
<td>50</td>
</tr>
</tbody>
</table>

Table 8: Parameters for the Experimental Systems.
Figure 11: Delay threshold probability as a function of $N_C$ for system F, $\tau$ varying.

<table>
<thead>
<tr>
<th>$N_C$</th>
<th>$P_{TH}$</th>
<th>$\tau_1 = 90%$</th>
<th>$95%$ CI</th>
<th>$R_{est}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>$5.596 \times 10^{-7}$</td>
<td>$(3.414 \times 10^{-7}, 7.968 \times 10^{-7})$</td>
<td>$4.140 \times 10^7$</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>$3.938 \times 10^{-8}$</td>
<td>$[0, 2.233 \times 10^{-1}]$</td>
<td>[MC]</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>$2.838 \times 10^{-9}$</td>
<td>$[0, 1.041 \times 10^{-6}]$</td>
<td>[MC]</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$N_C$</th>
<th>$P_{TH}$</th>
<th>$\tau_2 = 95%$</th>
<th>$95%$ CI</th>
<th>$R_{est}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>$5.144 \times 10^{-7}$</td>
<td>$[1.650 \times 10^{-7}, 8.638 \times 10^{-7}]$</td>
<td>$1.055 \times 10^7$</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>$1.886 \times 10^{-8}$</td>
<td>$[2.687 \times 10^{-8}, 2.503 \times 10^{-8}]$</td>
<td>$3.029 \times 10^6$</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>$4.967 \times 10^{-9}$</td>
<td>$[0, 1.377 \times 10^{-6}]$</td>
<td>$7.331 \times 10^5$</td>
<td></td>
</tr>
</tbody>
</table>

Table 9: Simulation results for system F, $\tau_1 = 90\%$, $\tau_2 = 95\%$, varying number of connections.
Figure 12: Delay threshold probability as a function of $N_C$ for systems B and C ($\hat{B}$ varying), $\tau = 90\%$.

The confidence intervals are not very tight due to the $\chi^2$ assumption for the distribution. The discontinuity at 8 connections implies that no cells would exceed the threshold if there were less than 8 connections. This fact is illustrated by a downward pointing arrow in Fig. 11.

To observe the effect of burstiness, we considered systems B and C, with burst lengths of 25 and 20, respectively. The results are given in Fig. 12 and Table 10. The threshold was set at 90% and $N_{C_0}(\tau) = 6$ and 7 for systems B and C, respectively. As expected, increasing the burst length increases the delay threshold probability. Similar behavior as before is observed for the speedup and confidence intervals.

For system B at 6 connections, $d_0 = 0$ and the only vector combination is the AAZ case. Thus, the exact delay threshold probability can be computed as $D_{\text{max}}(N_{C_0}(\tau))/(N_C\hat{B}[V])$.

Systems G and H depicted in Fig. 13 and Table 11 have peak rates of 500 Mbps and 400 Mbps, respectively. The threshold was set at 90% and $N_{C_0}(\tau) = 4$ for both systems. The system with the higher peak rate has a higher delay threshold probability, the effect of which decreases as more connections are added to the system. For these systems, IS was used for all points and the speedups were inversely proportional to the estimated probabilities.
Table 10: Simulation results for systems B and C ($\hat{B}$ varying), $\tau = 90\%$.

<table>
<thead>
<tr>
<th>$N_C$</th>
<th>$P_{TH}$</th>
<th>95% CI</th>
<th>$R_{net}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>$6.827 \times 10^{-4}$</td>
<td>exact</td>
<td>exact</td>
</tr>
<tr>
<td>7</td>
<td>$3.387 \times 10^{-3}$</td>
<td>$[1.026 \times 10^{-3}, 5.748 \times 10^{-4}]$</td>
<td>$1.977 \times 10^{-4}$</td>
</tr>
<tr>
<td>8</td>
<td>$2.123 \times 10^{-2}$</td>
<td>$[0.9216 \times 10^{-3}]$</td>
<td>$1.257 \times 10^{-1}$</td>
</tr>
<tr>
<td>9</td>
<td>$1.606 \times 10^{-1}$</td>
<td>$[0.3.877 \times 10^{-2}]$</td>
<td>[MC]</td>
</tr>
<tr>
<td>10</td>
<td>$7.314 \times 10^{-1}$</td>
<td>$[0.3.009 \times 10^{-1}]$</td>
<td>[MC]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$N_C$</th>
<th>$P_{TH}$</th>
<th>95% CI</th>
<th>$R_{net}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>0</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>7</td>
<td>$1.057 \times 10^{-3}$</td>
<td>$[3.250 \times 10^{-3}, 1.889 \times 10^{-3}]$</td>
<td>$1.806 \times 10^{-1}$</td>
</tr>
<tr>
<td>8</td>
<td>$2.827 \times 10^{-2}$</td>
<td>$[0.5.846 \times 10^{-2}]$</td>
<td>$1.306 \times 10^{2}$</td>
</tr>
<tr>
<td>9</td>
<td>$5.444 \times 10^{-1}$</td>
<td>$[0.2.172 \times 10^{-1}]$</td>
<td>$8.876 \times 10^{4}$</td>
</tr>
<tr>
<td>10</td>
<td>$2.082 \times 10^{-1}$</td>
<td>$[0.1.077 \times 10^{-1}]$</td>
<td>[MC]</td>
</tr>
</tbody>
</table>

Figure 13: Delay threshold probability as a function of $N_C$ for systems G and H ($\hat{\lambda}$ varying), $\tau = 90\%$. 

24
The threshold was set at \( \lambda /5/0/% \) and points for which the cell loss probability is at least \( /1 \) or \( /2 \) orders of magnitude smaller than the delay threshold probability /.

Table 11: Simulation results for systems G and H (\( \lambda \) varying), \( \tau = 90\% \).

<table>
<thead>
<tr>
<th>( N_C )</th>
<th>( P_{TH} )</th>
<th>95% CI</th>
<th>( R_{net} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>1.359 ( \times 10^{-6} )</td>
<td>( [6.254 \times 10^{-8}, 2.172 \times 10^{-6}] )</td>
<td>6.663 ( \times 10^{4} )</td>
</tr>
<tr>
<td>5</td>
<td>5.407 ( \times 10^{-8} )</td>
<td>( [0, 1.482 \times 10^{-7}] )</td>
<td>3.443 ( \times 10^{6} )</td>
</tr>
<tr>
<td>6</td>
<td>9.196 ( \times 10^{-8} )</td>
<td>( [0, 2.167 \times 10^{-7}] )</td>
<td>4.574 ( \times 10^{4} )</td>
</tr>
</tbody>
</table>

Table 12: Simulation results for systems D and E (\( \lambda \) varying), \( \tau = 50\% \).

<table>
<thead>
<tr>
<th>( N_C )</th>
<th>( P_{TH} )</th>
<th>95% CI</th>
<th>( R_{net} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>2.862 ( \times 10^{-16} )</td>
<td>( [1.014 \times 10^{-16}, 3.706 \times 10^{-16}] )</td>
<td>1.145 ( \times 10^{4} )</td>
</tr>
<tr>
<td>5</td>
<td>3.037 ( \times 10^{-16} )</td>
<td>( [4.464 \times 10^{-17}, 5.628 \times 10^{-16}] )</td>
<td>2.430 ( \times 10^{4} )</td>
</tr>
<tr>
<td>6</td>
<td>9.080 ( \times 10^{-18} )</td>
<td>( [0, 3.037 \times 10^{-17}] )</td>
<td>1.307 ( \times 10^{4} )</td>
</tr>
</tbody>
</table>

Next, we considered systems D and E with mean rates of 2 Mbps and 4 Mbps, respectively. The threshold was set at 50\% and \( N_{C0}(\tau) = 11 \) for both systems. The results given in Fig. 14 and Table 12 show that at a given number of connections, the delay threshold probability for system E is twice as that of system D. This is the same as the ratio of the mean rates. Similar speedup factors as above were observed.

Finally, we demonstrate how we obtain the upper bound for the delay threshold probability for a finite buffer system F. Table 13 lists \( \hat{P}_{TH} \), \( \hat{P}_{CL} \) (from [1]) and the resulting upper bound \( \Pr(D_K > \tau) \), which is plotted in Fig. 15. The resulting upper bound will be tight at points for which the cell loss probability is at least 1 or 2 orders of magnitude smaller than the delay threshold probability.

Next, we considered systems D and E with mean rates of 2 Mbps and 4 Mbps, respectively. The threshold was set at 50\% and \( N_{C0}(\tau) = 11 \) for both systems. The results given in Fig. 14 and Table 12 show that at a given number of connections, the delay threshold probability for system E is twice as that of system D. This is the same as the ratio of the mean rates. Similar speedup factors as above were observed.

Finally, we demonstrate how we obtain the upper bound for the delay threshold probability for a finite buffer system F. Table 13 lists \( \hat{P}_{TH} \), \( \hat{P}_{CL} \) (from [1]) and the resulting upper bound \( \Pr(D_K > \tau) \), which is plotted in Fig. 15. The resulting upper bound will be tight at points for which the cell loss probability is at least 1 or 2 orders of magnitude smaller than the delay threshold probability.
Figure 14: Delay threshold probability as a function of $N_C$ for systems D and E ($\bar{\lambda}$ varying), $\tau = 50\%$.

Table 13: Upper bound delay threshold probability estimates for system F, $\tau$ varying.
Figure 15: Upper bound delay threshold probability estimates for system F, $\tau$ varying.

6 Conclusion

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

We developed a multinomial formulation which effectively removed correlations (due to burstiness) between delay threshold events, which produced a “multiple bin” simulation structure. We developed and demonstrated a three part Importance Sampling procedure based on this structure.

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


