

Efficient Simulation of Rare Jitter Probabilities in ATM Switches

Ahmet A. Akyamaç

J. Keith Townsend

Center for Advanced Computing and Communication

North Carolina State University

Raleigh, N.C.

Abstract

The ATM Forum has defined two different types of cell delay variation (CDV) measures: The 1-Point CDV, which is a measure of jitter, is typically specified for constant bit rate (CBR) sources (eg. real-time video transfer) and the 2-Point CDV can be specified for both CBR and variable bit rate (VBR) sources (eg. compressed video). Analytical results are not available and the events associated with large jitter in ATM networks are typically rare ($< 10^{-6}$), hence Monte Carlo simulation is not feasible. In this paper, we model traffic using the ATM Forum standardized traffic descriptors and consider dispersion as a measure of jitter. We extend our previous work on remote delay quantiles for heterogeneous systems to generate efficient simulation techniques using Importance Sampling (IS) to estimate the 2-Point CDV for mixed CBR and VBR sources. Subsequently, we present a novel IS simulation methodology to estimate the 1-Point CDV for CBR sources in the presence of background VBR traffic. For both cases, we observe from experimental results that the improvement in simulation efficiency is inversely proportional to the probability being estimated ¹.

¹This 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

Cell Delay Variation (CDV), or jitter, is an important Quality of Service (QoS) measure that will be provided in Asynchronous Transfer Mode (ATM) Networks. The ATM Forum [1] has defined two different types of CDV measures: The 1-Point CDV is typically specified for constant bit rate (CBR) sources (eg. real-time video transfer) and the 2-Point CDV can be specified for both CBR and variable bit rate (VBR) sources (eg. compressed video). The events associated with large jitter in ATM networks are typically rare, on the order of 10^{-9} and below.

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].

Due to the low probabilities involved, large buffer sizes and the operational definition of the input traffic, accurate analytical solutions to CDV are unavailable. Thus, simulation is required to estimate the CDV probabilities.

Various forms of cell delay variation have been considered before in the literature. In [3], the waiting time distributions for VBR sources are analyzed at the output of an ATM switch, where the service time is assumed to be identical to the transmission time of a packet. The departure process of VBR cells is also considered in the context of $MMPP/D/1$ and $M/D/1$ queues in [4]. But, neither of these works considers CBR traffic with VBR background traffic. In [5], jitter is defined in terms of tail probabilities of the queueing delay and also in terms of the squared coefficient of variation of the intercell delay at the output. The CBR traffic is assumed to be multiplexed with Poisson background traffic. The traffic model in [6] is the same as the one on [5] and it is assumed that the sequence of random delays for the CBR cells constitutes a Markov process. The Poisson background traffic assumption is not realistic in the case of bursty VBR sources, which are present in ATM networks. In [7], the VBR traffic is modeled as a superposition of 2-state Markov processes and the CBR traffic is modeled as a special case of a Markovian $D - BMAP$ process. Approximate solution techniques are given for the tail probabilities of the cell delays. The ATM Forum traffic descriptors are used in [8] to generate tail probabilities, but the service time is assumed to be identical to the transmission time of a cell. The tail distribution described above corresponds to the 2-Point CDV definition. In [9], the 1-Point CDV definition is considered using comparisons between the interdeparture times and the CBR period. The background VBR traffic is in the form of independent and identically distributed batches of customers.

In this paper, we consider cell delay variation through a single stage ATM switch with CBR traffic multiplexed with background VBR traffic. We use the ATM Forum connection traffic descriptors to generate 1-Point CDV results for CBR traffic in the presence of background VBR traffic and 2-Point CDV results for mixed CBR and VBR traffic. We do not restrict the service rate and make no assumption of the background VBR traffic other than the fact that it has to be worst-case UPC compliant. Due to the rare nature of the jitter

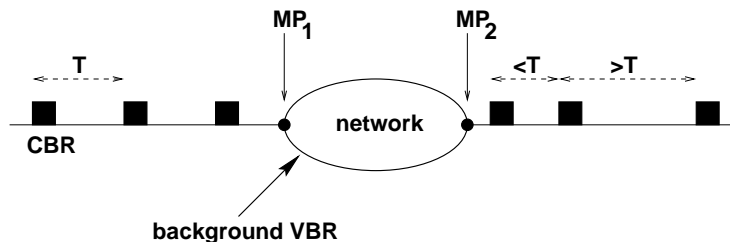


Figure 1: Cell Delay Variation through a network.

probabilities, we use Importance Sampling (IS) as a means of generating efficient simulations. For the 2-Point CDV results, we extend the techniques we used previously in [10] for the case of aggregate delay threshold probabilities with heterogeneous traffic. For the 1-Point CDV results, we present, for the first time, a new IS procedure to generate efficient simulations.

Cell delay variation for CBR traffic in ATM networks can manifest itself as either cell clumping or cell dispersion [11]. In this paper, we focus on cell dispersion as a quality of service measure. Cell clumping is bounded by the period of the CBR source, whereas cell dispersion can occur with significantly higher probability than cell clumping. At the expense of large buffers and induced delay, cell clumping can be effectively reduced by the use of jitter removal buffers [12]. However, it may not be possible to cancel the effects of very large delays in the network, in which case cells which are very late would be dropped. For this reason, it is important to characterize probabilities for cell dispersion.

This paper is organized as follows: Section 2 gives a description of the two ATM Forum CDV definitions. The system description and simulation setup are presented in Section 3. The efficient simulation methodology for the 2-Point CDV is described in Section 4. A detailed development of the IS simulation method for the 1-Point CDV follows in Section 5. Experimental results are presented in Section 6, followed by conclusions in Section 8.

2 The ATM Forum CDV Definitions

For CBR traffic that is not distorted by cell delay variation, the interarrival times of all CBR cells are constant and are given by T_{CBR} . However, when these CBR cells pass through a network (which contains switches, buffers and other background traffic) as in Fig. 1, the interdeparture times are no longer constant and may vary significantly from T_{CBR} . Interdeparture times less than T_{CBR} refer to clumping and those greater than T_{CBR} refer to dispersion. Cell delay variation, or jitter, is a measure of how significant a change is induced on the instantaneous interdeparture times of the CBR cells. In essence, this change will result in interdeparture times different than T_{CBR} . This, in turn, may have severe effects depending on the application using the CBR cells. For example, for a 1.5 Mbps MPEG NTSC video source, a delay variation of more than 11 ms. is unacceptable [13].

The ATM Forum defines two different measures for cell delay variation [1]. The one-point

CDV measures jitter at a single measurement point (MP). In Fig. 1, MP_2 would be such a point. The two-point CDV measures jitter between two MP 's, such as MP_1 and MP_2 in Fig. 1.

Following the notation in [1], the one-point CDV for cell k , y_k , at a measurement point, say MP_2 , is defined as the difference between the cell's reference arrival time, c_k , and the cell's actual arrival time at MP_2 , a_k , and is thus given by $y_k = c_k - a_k$. In this formulation, the reference arrival times are computed as follows:

$$\begin{aligned} c_0 &= a_0 = 0 \\ c_{k+1} &= \begin{cases} c_k + T_{\text{CBR}} & \text{if } c_k \geq a_k \\ a_k + T_{\text{CBR}} & \text{otherwise} \end{cases} \end{aligned} \quad (1)$$

Positive values of CDV correspond to cell clumping (interdeparture times less than T_{CBR}) and negative values of CDV correspond to cell dispersion (interdeparture times greater than T_{CBR}). The one-point CDV is typically used for CBR sources.

We mentioned in Section 1 that we focus on dispersion as a CDV measure. Thus, the important events are characterized by cells for which $y_k < -\delta$, where δ is the target dispersion value, which we specify as a percentage of the CBR period T_{CBR} .

The two-point CDV for cell k , v_k , between two measurement points MP_1 and MP_2 is defined as the difference between the absolute cell transfer delay of cell k , x_k , between the two reference points MP_1 and MP_2 and a reference cell transfer delay, $d_{1,2}$ between MP_1 and MP_2 and is thus given by $v_k = x_k - d_{1,2}$ [1]. The reference cell transfer delay is the absolute delay between MP_1 and MP_2 experienced by a reference cell. Note that the absolute cell transfer delay is nothing but the Cell Transfer Delay defined in [1] and hence can be used for both CBR and VBR sources. The important events for the two-point CDV are characterized by cells for which $v_k > \tau$, where τ is the target delay value.

3 Simulation Setup

3.1 System Description

The model we use for the ATM switch is given in Fig. 2. The switch has N_P input ports and N_P output ports. We assume that the number of connections is $N_C = N_P$. Each connection is routed uniformly and instantaneously to one of the output ports through the nonblocking bus. Due to the uniform nature of the traffic, we analyze a single *tagged* output buffer to represent the performance of all output buffers. Also, for a worst case analysis, we assume that all N_C connections are routed to the tagged output buffer. Even though we specify the output buffers to have a length of K , we will assume that they have infinite capacity and we will use K as a parameter for the two-point CDV calculations. The buffers are served at a constant rate of $\hat{\mu}$ using a FIFO queueing discipline. We place no restriction on the value of $\hat{\mu}$ or the relative values of $\hat{\mu}$ and the traffic rates.

We assume there are two classes of traffic: CBR traffic has the triplet $(\hat{\lambda}_{\text{CBR}}, \bar{\lambda}_{\text{CBR}}, \hat{B}_{\text{CBR}})$ and VBR traffic has the triplet $(\hat{\lambda}_{\text{VBR}}, \bar{\lambda}_{\text{VBR}}, \hat{B}_{\text{VBR}})$. By definition of the CBR traffic, $\hat{\lambda}_{\text{CBR}} =$

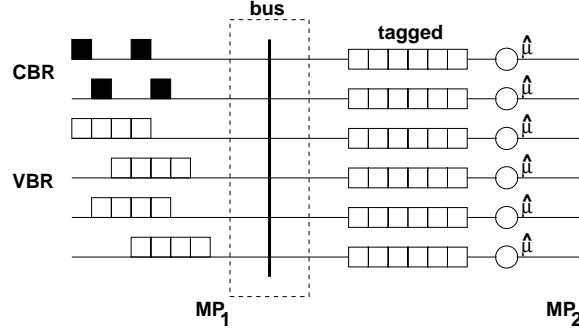


Figure 2: The ATM Switch Model.

$\bar{\lambda}_{\text{CBR}}$ and $\hat{B}_{\text{CBR}} = 1$. We assume that $\hat{\lambda}_{\text{VBR}} = k\hat{\lambda}_{\text{CBR}}$, where $k \in \mathcal{Z}^+$ and $k \geq 2$. For the VBR class, we make the worst case assumption that the traffic arrives according to the *greedy* pattern. In the ON period, the VBR source generates traffic at the peak rate of $\hat{\lambda}_{\text{VBR}}$ for a burst duration of \hat{B}_{VBR} , and the OFF period is required to average out to the mean rate of $\bar{\lambda}_{\text{VBR}}$.

3.2 Simulation Procedure

We use a slotted-time simulation model resulting from the normalization with respect to the peak rate of the VBR source, $\hat{\lambda}_{\text{VBR}}$. In the normalized model, one slot corresponds to the transmission of a single 53-byte ATM cell at the rate of $\hat{\lambda}_{\text{VBR}}$. After normalization, the equivalent arrival rate of the VBR source is 1 cell/slot and that of the CBR source is $\hat{\lambda}_{\text{CBR}}/\hat{\lambda}_{\text{VBR}} = 1/k$ cells/slot, where k was defined above. Due to the greedy traffic assumption, the VBR source is periodic. By definition, the CBR source is also periodic. After normalization, the periods are given by:

$$T_{\text{VBR}} = \frac{\hat{\lambda}_{\text{VBR}}\hat{B}_{\text{VBR}}}{\bar{\lambda}_{\text{VBR}}} \quad (2)$$

$$T_{\text{CBR}} = \frac{\hat{\lambda}_{\text{VBR}}\hat{B}_{\text{CBR}}}{\bar{\lambda}_{\text{CBR}}} = \frac{k\hat{\lambda}_{\text{CBR}}}{\bar{\lambda}_{\text{CBR}}} = k \quad (3)$$

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

The shared bus switch architecture results in *service slots* within an arrival slot, as shown in Fig. 4. At an arrival slot, there can be up to N_C cell arrivals and thus the number of service slots per arrival slot is N_C . We make a worst case assumption that if at an arrival slot the number of cell arrivals is $N < N_C$, we assume these cells occupy the first N service

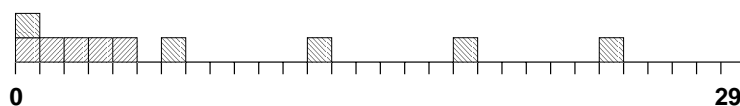


Figure 3: Slotted-time example pattern for CBR/VBR traffic.

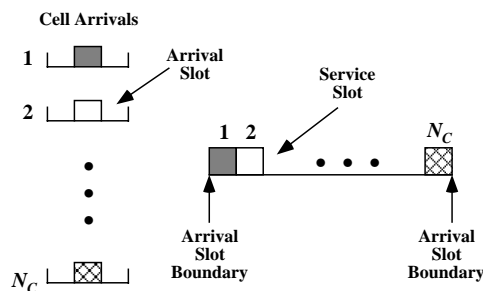


Figure 4: Arrival slot and service slot relationship.

slots. In each service slot, one cell can be loaded into the output buffer and μ/N_C cells can be serviced.

The cells exit the buffer in slotted-time, where one departure slot is equivalent to one arrival slot. If cell x arrives at arrival slot a_x at line s_x (which is, by definition, the service slot) and encounters a queue length of q_x upon arrival, then the departure slot for cell x is given by:

$$d_x = a_x + \left\lceil \frac{\frac{q_x+1}{\mu/N_C} + s_x - 1}{N_C} \right\rceil - 1 \quad (4)$$

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

$$\underline{v} = (\underline{v}_{\text{VBR}} | \underline{v}_{\text{CBR}}) \quad (5)$$

4 Efficient Simulation of the Two-Point CDV

We mentioned in Section 2 that the two-point CDV involves the absolute cell transfer delay v_k for cell k and that we focus on the events for which $v_k > \tau$, where τ is a reference absolute delay. In [10], we developed efficient simulation methods to estimate the aggregate delay threshold probability for heterogeneous traffic consisting of two classes, namely CBR and VBR, using the same ATM switch model shown in Fig. 2. We defined the delay threshold

probability to be the probability that the end to end delay through the switch exceeds a given threshold τ , $\Pr(D > \tau)$, where τ represented a percentage of the specified buffer length K .

Here, we assume that MP_1 is the entry point into the switch and MP_2 is the exit point from the switch. Thus, in this context, the two-point CDV is identical to the delay threshold probability considered in [10]. Since we assumed that the routing within the switch is instantaneous, the delay between points MP_1 and MP_2 is a result of the queueing. We use the techniques developed in [10] to generate the two-point CDV for the individual classes. Note that we consider the two-point CDV for both the CBR and VBR classes. For consistency, we use the notation in [10] whereby we represent the two-point CDV (or delay threshold probability) using P_{TH} .

Following the development in [10], 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, $|V| = (T_{VBR})^{N_{VBR}} \cdot (T_{CBR})^{N_{CBR}}$, where $|\cdot|$ denotes the cardinality of the set. The probability that j cells exceed the threshold in steady state is given by $p_{TH_j} = n_{TH_j}/|V|$ for $j = 0, \dots, D_{max}$, where D_{max} is the maximum number of cells that can exceed the threshold. 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:

$$P_{TH} = \sum_{j=0}^{D_{max}} \left(\frac{j}{N_{cells}} \right) p_{TH_j} \quad (6)$$

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

$$N_{cells} = N_{VBR} \hat{B}_{VBR} + N_{CBR} \cdot \frac{T_{SP}}{\hat{\lambda}_{VBR}/\hat{\lambda}_{CBR}} \quad (7)$$

The above formulation expresses P_{TH} as a weighted sum of multinomial probabilities and thus creates a “multiple bin” probability structure composed of bins numbered 0 to D_{max} , where bin j corresponds to j cells that exceed the threshold. The value of D_{max} can be found using a single simulation run [10].

Calculating the exact delay threshold probability has a complexity of $O((T_{VBR})^{N_{VBR}} \cdot (T_{CBR})^{N_{CBR}})$ and is thus intractable for realistic systems. Monte Carlo simulation is not feasible due to the very low probabilities involved (typically in the range of 10^{-9} to 10^{-12}). In [14, 10], we developed a three-part Importance Sampling simulation procedure to efficiently estimate P_{TH} . The first two parts identified a set V^* which contained all of the vectors that caused a cell to exceed the delay threshold and generated the related IS weights. The third part took advantage of the multiple bin structure to further enhance the simulation efficiency.

Using the simulation procedure in [10], the aggregate delay threshold probability is estimated as follows:

$$\hat{P}_{TH}^* = \sum_{j=1}^{D_{max}} \left(\frac{j}{N_{cells}} \right) \hat{p}_{TH_j}^* \quad (8)$$

where $\hat{p}_{TH_j}^*$ are the estimates for the individual bins and are given by:

$$\hat{p}_{TH_j}^* = \frac{1}{n_j} \sum_{s=1}^{n_j} I_j^*(s) w_j^*(s) \quad (9)$$

where $I_j^*(s)$ is the indicator function for j cells exceeding the threshold at run s , $w_j^*(s)$ is a vector dependent IS weight function and n_j is the number of simulation runs for bin j . We estimate the variance of the IS estimator as follows:

$$\hat{\sigma}^2(\hat{p}_{TH_j}^*) = \frac{1}{n_j(n_j - 1)} \sum_{s=1}^{n_j} (I_j^*(s) w_j^*(s) - \hat{p}_{TH_j}^*)^2 \quad (10)$$

and following [14, 10], we generate confidence intervals using the result that the confidence interval of a weighted sum of individual multinomial probabilities follows a χ^2 distribution. The improvement in simulation efficiency is given by

$$\hat{R}_{net} = \frac{\sum_{j=1}^{D_{max}} j^2 \hat{p}_{TH_j}^* (1 - \hat{p}_{TH_j}^*) / n_j}{\sum_{j=1}^{D_{max}} j^2 \hat{\sigma}^2(\hat{p}_{TH_j}^*)} \quad (11)$$

Note that the two-point CDV, or delay threshold probabilities for the individual classes, P_{CBR} and P_{VBR} are given by:

$$P_{CBR} = \frac{1}{C_{CBR}} \sum_{j=1}^{D_{max}} p_{CBR_j} \quad (12)$$

$$P_{VBR} = \frac{1}{C_{VBR}} \sum_{j=1}^{D_{max}} p_{VBR_j} \quad (13)$$

where $C_{CBR} = N_{CBR} T_{SP} \hat{\lambda}_{CBR} / \hat{\lambda}_{VBR}$ is the total number of CBR cells and $C_{VBR} = N_{VBR} \hat{B}_{VBR}$ is the total number of VBR cells. Here, p_{CBR_j} is the expected number of CBR cells that exceed the threshold conditioned on a total of j cells that exceed the threshold and p_{VBR_j} is the expected number of VBR cells that exceed the threshold conditioned on a total of j cells that exceed the threshold. It is very difficult to identify sets of vectors that will result in exactly k CBR and l VBR cells that exceed the threshold. For this reason, we use the simulation biasing procedure in [10] to identify the set V^* that contains all vectors which cause at least one cell to exceed the threshold. Hence, all vectors which cause CBR and VBR cells to exceed the threshold will also be included in this set.

For the CBR class, the estimate of the two-point CDV, \hat{P}_{CBR}^* , is given by:

$$\hat{P}_{CBR}^* = \frac{1}{C_{CBR}} \sum_{j=1}^{D_{max}} \hat{p}_{CBR_j}^* \quad (14)$$

where $\hat{p}_{CBR_j}^*$ is generated as follows:

$$\hat{p}_{CBR_j}^* = \frac{1}{n_j} \sum_{s=1}^{n_j} M_s^j(\text{CBR}) w_j^*(s) \quad (15)$$

Here, $w_j^*(s)$ is the IS weight defined earlier and $M_s^j(\text{CBR})$ is the number of CBR cells that exceed the threshold at run s , given that at total of j cells exceed the threshold. Similarly, the estimate of the two-pint CDV for the VBR class, \hat{P}_{VBR}^* , is given by:

$$\hat{P}_{\text{VBR}}^* = \frac{1}{C_{\text{CBR}}} \sum_{j=1}^{D_{\max}} \hat{p}_{\text{VBR}_j}^* \quad (16)$$

where $\hat{p}_{\text{VBR}_j}^*$ is generated as follows:

$$\hat{p}_{\text{VBR}_j}^* = \frac{1}{n_j} \sum_{s=1}^{n_j} M_s^j(\text{VBR}) w_j^*(s) \quad (17)$$

For this case, $M_s^j(\text{VBR})$ is the number of VBR cells that exceed the threshold at run s , given that at total of j cells exceed the threshold.

Hence, to estimate the individual delay threshold probabilities (two-point CDV), we run IS simulations as in [10] and collect the aggregate statistics as well as individual CBR and VBR statistics. Since the biasing is performed according to the aggregate probability, confidence intervals and improvement factors are only calculated for the aggregate estimate. Note that if the CBR and VBR classes were targeted individually, the individual improvements would be only slightly higher. But, as mentioned before, it is very difficult to target the IS biasing for individual classes. Furthermore, since the set V^* contains *all* the vectors that case at least one cell to exceed the threshold, the biasing procedure is valid for the individual classes and the individual estimates are unbiased. Experimental results for the two-point CDV are reported in Section 6.

5 Efficient Simulation of the One-Point CDV

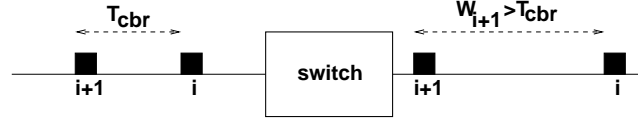
5.1 Motivation

For the one-point CDV, we assume that the measurement point is at the output of the ATM switch, labeled MP_2 in Fig. 2. We consider the one-point CDV for the CBR class only. We assume that there is a single CBR source at the first input line. All other input lines have VBR sources, constituting the background traffic.

For the CBR source, dispersion between two cells i and $i+1$ is illustrated in Fig. 5. At the input of the switch, the interarrival time of cells i and $i+1$ is T_{CBR} slots, whereas the interdeparture time is $W_{i+1} > T_{\text{CBR}}$.

For the cell $i+1$ shown in Fig. 5, the ATM Forum one-point CDV definition presented in Section 2 will result in $y_{i+1} = T_{\text{CBR}} - W_{i+1}$. In Section 2, we noted that important (dispersive) events are characterized by $y_{i+1} < -\delta$. This is thus equivalent to saying $W_{i+1} > T_{\text{CBR}} + \delta$. Hence, the tail probability of dispersion in the one-point CDV sense is given by the following:

$$\Pr(W > T_{\text{CBR}} + \delta) \quad (18)$$

Figure 5: Illustration of dispersion between two cells i and $i + 1$.

where W is a random variable denoting the interdeparture times of cells. However, W is a parameter defined at the output of the switch. To translate W to a parameter internal to the switch, we note that the queuing delay experienced by cell i upon arrival to the output buffer, D_i , is given by:

$$\begin{aligned} D_0 &= 0 \\ D_i &= \sum_{k=0}^i W_k - iT_{\text{CBR}}, \quad i \geq 1 \end{aligned} \quad (19)$$

Hence, the interdeparture time between cells i and $i + 1$ is given by:

$$W_{i+1} = D_{i+1} - D_i + T_{\text{CBR}} \quad (20)$$

Thus, the probability of dispersion can also be expressed as:

$$\Pr(D_{i+1} - D_i > \delta) \quad (21)$$

For conciseness, the probability of dispersion will also be represented by P_{DP} in the remainder of this paper.

5.2 Multinomial Formulation

In a period T_{SP} , the total number of pairs of CBR cells is given by:

$$N_{\text{pairs}} = \frac{T_{\text{SP}}}{T_{\text{CBR}}} \quad (22)$$

For a given set of parameters for the CBR and VBR classes and a given dispersion value δ , let J_{max} be the maximum number of CBR pairs that are dispersed by more than δ (ie. that have an interdeparture time greater than $T_{\text{CBR}} + \delta$). Since there is only one CBR source and also due to the periodicity of the system, we can assume, without loss of generality, that the CBR source starts at slot 0. Thus, the starting slot vector \underline{v} is given by:

$$\underline{v} = (\underline{v}_{\text{VBR}}|0) \quad (23)$$

As before, V is the set of all possible connection starting-slot vectors, where for this case, $|V| = (N_{\text{VBR}})^{T_{\text{SP}}}$. Let n_{DP_j} be the number of connection starting-slot vectors that map to exactly j pairs that are dispersed by more than δ in steady state. Then, the probability that

j pairs are dispersed by more than δ is given by $p_{DP_j} = n_{DP_j}/|V|$ for $j = 0, \dots, J_{max}$. Note that $\sum_{j=0}^{J_{max}} p_{DP_j} = 1$. Then, the dispersion probability is given by:

$$P_{DP} = \sum_{j=0}^{J_{max}} \left(\frac{j}{N_{\text{pairs}}} \right) p_{DP_j} \quad (24)$$

This multinomial formulation is very similar to that of the delay threshold probability given in Eq. 6. We again have a multiple bin probability structure, where bin j corresponds to j pairs that are dispersed by more than δ in steady state. The exhaustive solution to Eq. 24 requires $(N_{\text{VBR}})^{T_{\text{SP}}}$ runs and is also infeasible for realistic systems.

5.3 Monte Carlo Simulation

Using Monte Carlo simulation and an approach similar to the ones in [14, 10], we form an unbiased estimate of P_{DP} as follows:

$$\hat{P}_{DP} = \frac{1}{N_{\text{pairs}}} \sum_{j=1}^{J_{max}} j \hat{p}_{DP_j} \quad (25)$$

where \hat{p}_{DP_j} are estimates of the individual probabilities p_{DP_j} and are given by:

$$\hat{p}_{DP_j} = \frac{1}{N_{MC}} \sum_{s=1}^{N_{MC}} \Gamma_j(s) \quad (26)$$

where N_{MC} is the total number of Monte Carlo simulation runs and $\Gamma_j(s)$ is an indicator function for j cells dispersing by more than δ at run s . The estimate of the estimator variance is given by:

$$\hat{\sigma}^2(\hat{P}_{DP}) = \frac{1}{(N_{\text{pairs}})^2} \sum_{j=1}^{J_{max}} j^2 \hat{\sigma}^2(\hat{p}_{DP_j}) \quad (27)$$

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

$$\hat{\sigma}^2(\hat{p}_{DP_j}) = \frac{1}{N_{MC}(N_{MC} - 1)} \sum_{s=1}^{N_{MC}} (\Gamma_j(s) - \hat{p}_{DP_j})^2 \quad (28)$$

Following [14, 10], we generate confidence intervals using the result that the confidence interval of a weighted sum of individual multinomial probabilities follows a χ^2 distribution.

5.4 Importance Sampling Method

In ATM networks, the probability of high dispersion should be kept very low, typically below 10^{-9} . As mentioned before, Monte Carlo simulation is not feasible at these probabilities for sufficiently accurate estimates. Hence, we use Importance Sampling (IS) to increase the efficiency of the simulations. Using IS, we 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}_{DP}^* . 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 original 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 we have a fixed CBR source, this can further be reduced to $f_{\underline{V}_1, \underline{V}_2}(\underline{v}_1, \underline{v}_2) = f_{\underline{V}_1}(\underline{v}_1)$. Furthermore, since we only bias the VBR source, $f_{\underline{V}_1^*, \underline{V}_2^*}(\underline{v}_1^*, \underline{v}_2^*) = f_{\underline{V}_1^*}(\underline{v}_1^*)$. To keep \hat{P}_{DP} unbiased, each important event must be appropriately weighted or “unbiased”. This weight is given by $w(\underline{v}_1^*)$. Additionally, we require that $f_{\underline{V}_1^*}(\underline{v}_1^*) > 0$ whenever $f_{\underline{V}_1}(\underline{v}_1) > 0$.

The important region can be identified by a subset V_{DP} of V such that all vectors in V_{DP} result in at least one CBR pair dispersed by more than δ in steady state. For realistic systems, $|V_{DP}| \ll |V|$. Monte Carlo simulation is inefficient since it samples from V . We do not know V_{DP} *a priori* and instead sample from a set V^* which is known to contain V_{DP} and for which $|V^*| \ll |V|$. Note that V^* may contain vectors that cause no dispersion. We form the unbiased estimate \hat{P}_{DP}^* as a linear combination of individual estimates $\hat{p}_{DP_j}^*$ as before:

$$\hat{P}_{DP}^* = \sum_{j=1}^{J_{max}} \left(\frac{j}{N_{pairs}} \right) \hat{p}_{DP_j}^* \quad (29)$$

For the IS case, the individual estimates $\hat{p}_{DP_j}^*$ are as follows:

$$\hat{p}_{DP_j}^* = \frac{1}{n_j} \sum_{s=1}^{n_j} \Gamma_j^*(s) w_j^*(s) \quad (30)$$

where $\Gamma_j^*(s)$ is the indicator function for pairs dispersed by more than δ for run s under the biased pdf $f_{\underline{V}_1^*}(\underline{v}_1^*)$ and $w_j^*(s) = f_V(\underline{v}^*(s)) / f_{\underline{V}_1^*}(\underline{v}_1^*(s))$ is the IS weight for the vector drawn in run s . The variance of the IS estimate is found as in Eq. 27, where the individual estimates $\hat{\sigma}^2(\hat{p}_{DP_j}^*)$ are calculated as follows:

$$\hat{\sigma}^2(\hat{p}_{DP_j}^*) = \frac{1}{n_j(n_j - 1)} \sum_{s=1}^{n_j} (\Gamma_j^*(s) w_j^*(s) - \hat{p}_{DP_j}^*)^2 \quad (31)$$

The improvement factors are calculated as in Section 4 as follows:

$$\hat{R}_{net} = \frac{\sum_{j=1}^{J_{max}} j^2 \hat{p}_{DP_j}^* (1 - \hat{p}_{DP_j}^*) / n_j}{\sum_{j=1}^{J_{max}} j^2 \hat{\sigma}^2(\hat{p}_{DP_j}^*)} \quad (32)$$

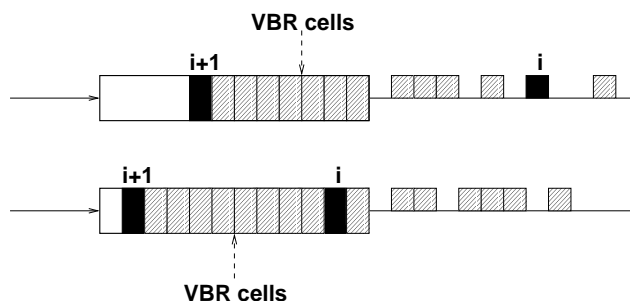


Figure 6: Two different scenarios in which dispersion can occur.

5.5 IS Biasing Procedure

The function of the biasing procedure is to generate the space V^* from which the VBR connection starting-slots are chosen. The simulation algorithm, to be presented in Section 5.6, involves selection of the starting-slots of the VBR sources from V^* so as to cause a dispersion of more than δ for at least one CBR pair. The simulation algorithm uses the results of the biasing procedure to drive the simulation.

Once one CBR pair is forced to disperse by more than δ , the remaining (if any) VBR sources are selected so as to capture all cases for which there are multiple pairs that disperse by more than δ .

As shown in Eq. 21, dispersion occurs when the difference of the departure times of two consecutive CBR cells, $D_{i+1} - D_i$ exceeds the specified jitter target δ . If no VBR cells arrive between the arrival of the two CBR cells i and $i+1$, then there will be no dispersion. Hence, dispersion is caused by the arrival of background VBR cells in between two CBR cells. There are two different ways in which two CBR cells i and $i+1$ can disperse, as illustrated in Fig. 6. In the first case, the queue empties after the departure of cell i , but before the arrival of cell $i+1$, as shown in the top half of Fig. 6. In this case, dispersion is caused by VBR cells that arrive after CBR cell i has departed from the queue, but before CBR cell $i+1$ has departed. In the second case, CBR cells i and $i+1$ are both inside the queue for a nonzero amount of time, as shown in the bottom half of Fig. 6. In this case, dispersion is caused by all VBR cells that arrive after the arrival of CBR cell i , but before the arrival of CBR cell $i+1$. For Cases 1 and 2, a different number of VBR cell arrivals are necessary to cause dispersion. We subsequently analyze these cases separately.

5.5.1 Case 1

For this case, CBR cell i departs from the queue before the arrival of cell $i+1$. An illustration of this case is shown in Fig. 7 in slotted-time, where, without loss of generality, we assume that cell i arrives at slot 0. Also, assume initially that $Q_0 = 0$ before the arrival of cell i . The CBR cell i exits the buffer in $\epsilon = \lceil 1/\mu \rceil$ slots, as shown in Fig. 7. Then, for case 1, all VBR cells must arrive within a support l , illustrated in Fig. 7, where $1 \leq l \leq T_{\text{CBR}} - \lceil 1/\mu \rceil$. Since the CBR source was assumed to arrive at line 1, VBR cells that arrive at slot T_{CBR} will

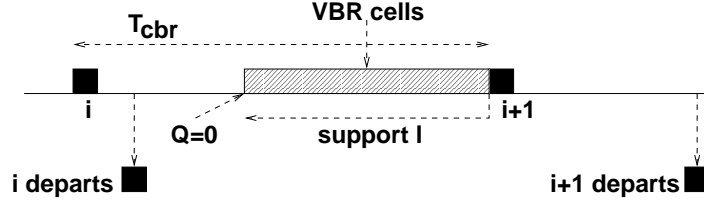


Figure 7: An illustration of Case 1 in slotted-time.

have no effect on the dispersion. Also note that, by definition, the queue length cannot drop to 0 during a given support l since otherwise, the starting-slot configuration would conform to a different support $l' < l$.

The queue length before the arrival of CBR cell $i + 1$ is given by:

$$Q = V_{\text{cells}} - l\mu \quad (33)$$

where V_{cells} is the total number of VBR cells that arrive within the support l . Hence, CBR cell $i + 1$ exits the buffer in $\lceil (Q + 1)/\mu \rceil$ slots. To cause a dispersion of more than δ , we require that:

$$\left\lceil \frac{Q + 1}{\mu} \right\rceil - e > \delta \quad (34)$$

Using the relation that $\lceil a \rceil > b \Rightarrow a > b$ and combining Eq.'s 33 and 34, we find that the number of VBR cells in the support of l must satisfy:

$$V_{\text{cells}} > \mu(\delta + e) - 1 + l\mu \quad (35)$$

Thus, in a support of l where $1 \leq l \leq T_{\text{CBR}} - \lceil 1/\mu \rceil$, the minimum number of VBR cells required to cause a dispersion of more than δ is given by:

$$V_{\text{min}} = \lfloor \mu(\delta + e) - 1 + l\mu \rfloor + 1, \text{ for } 1 \leq l \leq T_{\text{CBR}} - \lceil 1/\mu \rceil \quad (36)$$

where $e = \lceil 1/\mu \rceil$ slots.

In the above, we assumed that $Q_0 = 0$. If $Q_0 > 0$, then we would have that $e' = \lceil (Q_0 + 1)/\mu \rceil > e$ and from Eq. 36, we would have that $V'_{\text{min}} > V_{\text{min}}$. This means that stating V_{min} as in Eq. 36 encompasses all cases, including the ones for which $Q_0 > 0$.

5.5.2 Case 2

For this case, there is a nonzero amount of time for which CBR cells i and $i + 1$ are simultaneously inside the queue. An illustration of this case is shown in Fig. 8 in slotted-time, where we again assume that cell i arrives at slot 0. Again, we assume initially that $Q_0 = 0$ before the arrival of cell i . Hence, we have $e = \lceil 1/\mu \rceil$. Here, the queue length does not drop to zero between the arrivals of the two CBR cells. The support l satisfies $T_{\text{CBR}} - e + 1 \leq l \leq T_{\text{CBR}} + \hat{B}_{\text{VBR}} - 1$, since bursts arriving to the left of CBR cell $i + 1$

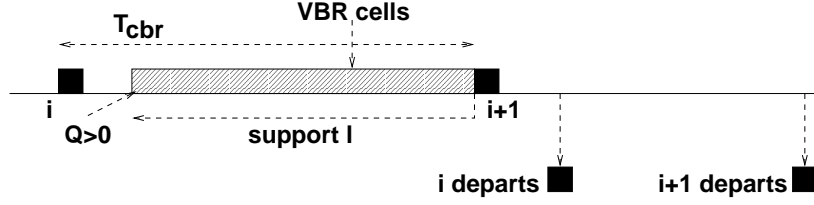


Figure 8: An illustration of Case 2 in slotted-time.

can also contribute to dispersion. For $l \leq T_{\text{CBR}}$, the total amount of service between the two cells is $T_{\text{CBR}}\mu$ regardless of the support and the cell i also contributes by $1/\mu$ to the dispersion. Proceeding as in Case 1, we obtain:

$$V_{\text{cells}} > \mu \left(\delta + e - \frac{1}{\mu} \right) - 1 + T_{\text{CBR}}\mu \quad (37)$$

Note that for Case 2, V_{cells} is independent of the support. If $Q_0 > 0$, we would have $e' = \lceil (Q_0 + 1)/\mu \rceil$ and would replace $1/\mu$ by $(Q_0 + 1)/\mu$ in the above expression. Thus, for V_{cells} to encompass all dispersive cases, we make the following modification:

$$V'_{\text{cells}} = \mu \left(\delta + e - \frac{1}{\mu} - 1 \right) - 1 + T_{\text{CBR}}\mu \quad (38)$$

where $e = \lceil 1/\mu \rceil$. When the support is $l > T_{\text{CBR}}$, V_{cells} is augmented by $l - T_{\text{CBR}}$ to compensate for the cells that remain to the left of CBR cell i and hence that have no effect on the dispersion. Thus, for Case 2, V_{min} is given by:

$$V_{\text{min}} = \begin{cases} \left\lfloor \mu \left(\delta + e - \frac{1}{\mu} - 1 \right) - 1 + T_{\text{CBR}}\mu \right\rfloor + 1 & , T_{\text{CBR}} - e + 1 \leq l \leq T_{\text{CBR}} \\ \left\lfloor \mu \left(\delta + e - \frac{1}{\mu} - 1 \right) - 1 + T_{\text{CBR}}\mu \right\rfloor + 1 + l - T_{\text{CBR}} & , T_{\text{CBR}} < l \leq T_{\text{CBR}} + \hat{B}_{\text{VBR}} - 1 \end{cases} \quad (39)$$

For Cases 1 and 2, Eq.'s 36 and 39 specify the minimum number of VBR cells necessary for a support of l . The minimum number of VBR connections to provide V_{min} cells is then given by:

$$N_{\text{min}} = \left\lceil \frac{V_{\text{min}}}{\min\{l, \hat{B}_{\text{VBR}}\}} \right\rceil \quad (40)$$

The maximum number of VBR cells available between CBR cells i and $i + 1$ is given by:

$$V_{\text{max}} = \begin{cases} N_{\text{VBR}} \cdot \min\{l, \hat{B}_{\text{VBR}}\} & , 1 \leq l \leq T_{\text{CBR}} \\ N_{\text{VBR}} \cdot \min\{l, \hat{B}_{\text{VBR}}\} - (l - T_{\text{CBR}}) & , T_{\text{CBR}} < l \leq T_{\text{CBR}} + \hat{B}_{\text{VBR}} - 1 \end{cases} \quad (41)$$

For a support l , unless $V_{\text{max}} \geq V_{\text{min}}$, there will be no cases which cause a dispersion of more than δ . Hence, the minimum number of VBR sources required to cause at least one CBR pair to disperse by more than δ is given by:

$$V_{\text{min}}^{\text{vbr}} = \min_{1 \leq l \leq T_{\text{CBR}} + \hat{B}_{\text{VBR}} - 1} \{N_{\text{min}}\} \quad (42)$$

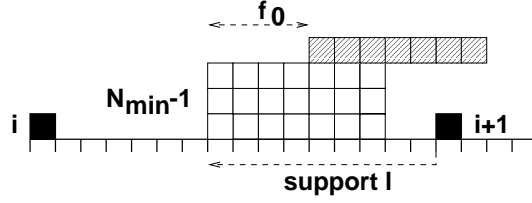


Figure 9: Calculation of the distance f_0 .

For a support of l , one burst is always fixed at slot $T_{\text{CBR}} - l$. This is also the leftmost burst. The number of additional VBR cells required in the support l to cause dispersion is given by:

$$V_{\text{extra}} = V_{\text{min}} - \min\{l, \hat{B}_{\text{VBR}}\} \quad (43)$$

Given a support l and a number of VBR bursts $N_{\text{VBR}} = N_{\text{min}}$, unless all VBR bursts arrive within a certain number of slots of the fixed burst, there will not be enough cells to cause dispersion. Let f_0 be the maximum distance from the fixed burst within which all VBR bursts must arrive so that at least V_{min} cells arrive in the support l . The maximum value of f_0 can be found by placing $N_{\text{min}} - 1$ VBR bursts at slot $\hat{B}_{\text{CBR}} - l$ (the beginning of the burst) and finding the maximum distance for burst N_{min} , as seen shaded diagonally in Fig. 9. The resulting distance is given as follows:

$$f_0 = s - \left[V_{\text{extra}} - (N_{\text{min}} - 1) \min\{l, \hat{B}_{\text{VBR}}\} \right] \quad (44)$$

So, at $N_{\text{VBR}} = N_{\text{min}}$ VBR connections and at the support l , unless all VBR bursts start within f_0 slots of each other, the CBR pairs i and $i + 1$ will not be dispersed.

At $N_{\text{VBR}} = N_{\text{min}} + r$ connections, where $r > 0$, each one of the extra VBR bursts can compensate for one move of the diagonally shaded burst in Fig. 9 to the right. If $r \geq \hat{B}_{\text{VBR}}$, the r extra bursts compensate for the entire diagonally shaded burst and hence no more than $N_{\text{min}} + r$ bursts are necessary. We have:

$$f_{r,r>0} = \begin{cases} s & , \text{ if } f_0 \geq s - r \\ s - 1 & , \text{ otherwise} \end{cases} \quad (45)$$

Thus, at $N_{\text{VBR}} = N_{\text{min}} + r$ connections and at the support l , unless N_{min} VBR bursts start within f_0 slots of each other or $N_{\text{min}} + 1$ VBR bursts start within f_1 slots of each other, etc., the CBR pairs i and $i + 1$ will not be dispersed.

For each support l for a given CBR pair, the biasing procedure identifies the values N_{min} and f_r , $0 \leq r \leq N_{\text{VBR}} - N_{\text{min}}$. It also identifies $V_{\text{min}}^{\text{vbr}}$, the minimum number of VBR connections necessary to cause a dispersion for any pair.

5.6 Simulation Algorithm

The simulation algorithm uses the results of the IS biasing procedure to drive the simulation runs. The relationship between the algorithms is shown in Fig. 10.

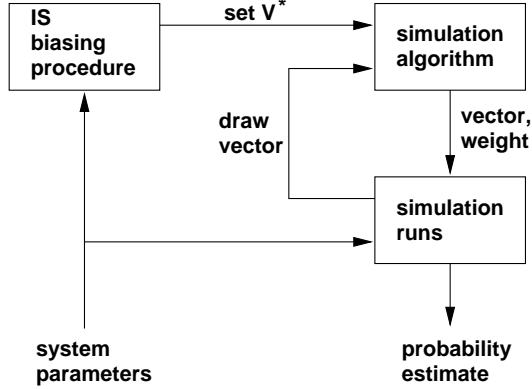


Figure 10: Relationship between the simulation procedures.

If $V_{\min}^{\text{vbr}} > N_{\text{VBR}}$, the algorithm exits. Otherwise, it first calculates a value $r_{\text{over}} = N_{\text{VBR}} - V_{\min}^{\text{vbr}}$, which specifies the number of connections that can be sampled uniformly from the entire period at the beginning since only V_{\min}^{vbr} bursts are necessary to cause a pair to be dispersed. The IS weight is initialized to 1. Subsequently, the algorithm goes through all pairs and all supports to determine which pairs can potentially disperse and which supports within the pairs can potentially cause dispersion. Pairs and supports that can cause dispersion are marked, the others are unmarked. For marked pairs and supports, this results in *min* and *max* ranges for which identify the starting slots within which the next burst must be chosen to potentially cause a dispersion. The overall range is given by:

$$(MIN, MAX) = \bigcup_{(\text{marked pairs } p)} \bigcup_{(\text{marked supports } l)} (min, max) \quad (46)$$

This concludes the update phase. Subsequently, the next burst is sampled within (MIN, MAX) and the IS weight is multiplied by $(MAX - MIN)/T_{\text{SP}}$. The algorithm proceeds with update and sample phases until all bursts for the given vector are sampled. If at a point in the algorithm it is observed that for a given pair and a given support that N_{min} bursts arrive within f_0 or $N_{\text{min}} + 1$ bursts arrive within f_1 etc., a Monte Carlo flag is set, meaning that a potential dispersion has already occurred and that the remaining bursts should be sampled uniformly from T_{SP} . This facet enables the algorithm to capture all vectors that cause multiple pairs to disperse as well.

A pseudocode of the simulation algorithm follows.

simulation_procedure:

- $r_{\text{over}} \leftarrow N_{\text{VBR}} - V_{\text{min}}^{\text{vbr}}$.
- if ($r_{\text{over}} < 0$) exit. /* not enough VBR connections */
- $w = 1$. /* initialize IS weight */
- $v = (\text{zeros}(N_{\text{VBR}})|0)$. /* initialize the vector */
- unmark MC. /* initialize Monte Carlo flag */
- for (burst=1 ; burst \leq r_{over} ; burst++) /* the first r bursts are MC */
 - sample $v(\text{burst})$ uniformly over $[0, T_{\text{SP}} - 1]$.
- call *update_procedure* (v, MIN, MAX) /* this generates MIN and MAX */
- for (burst= $r+1$; burst \leq N_{VBR} ; burst++) /* sample other bursts */
 - sample $v(\text{burst})$ uniformly over $[MIN, MAX]$.
 - $w \leftarrow w \times (MAX - MIN)/T_{\text{SP}}$.
 - call *update_procedure* (v, MIN, MAX) /* update after every burst */
- return v, w . /* return the vector and IS weight */

update_procedure(v, MIN, MAX):

- for (pair=1 ; pair \leq $T_{\text{SP}}/T_{\text{CBR}}$; pair++) /* all pairs */
 - for (support=1 ; support \leq $T_{\text{CBR}} + \hat{B}_{\text{VBR}} - 1$; support++) /* supports */
 - * if (pair can be dispersed at support) mark (pair@support).
 - * else unmark (pair@support)
 - * update *min* and *max* from v .
 - * for ($r=0$; $r \leq N_{\text{VBR}} - N_{\text{min}}$; r++) /* check for MC */
 - if ($N_{\text{min}} + r$ bursts are within f_r) mark MC.
 - else unmark MC.
 - * if (MC) (min, max) \leftarrow $(0, T_{\text{SP}} - 1)$
- (MIN, MAX) \leftarrow $\cup(min, max)$ /* update MIN and MAX */
- return MIN, MAX .

5.7 Specification of Dispersion

As mentioned in Section 2, we specify the dispersion target δ in terms of a percentage of the CBR period T_{CBR} . There is a maximum dispersion that can be specified for a given system and a given number of background VBR sources.

Due to the physical constraints of the system, for there to be dispersion in steady state, all cells must exit within a superperiod T_{SP} . Otherwise, all of the CBR cells will be clumped at departure. The physical constraint implies the following:

$$\left\lceil \frac{1}{\mu} \right\rceil + \delta + (N_{\text{pairs}} - 2) \left\lceil \frac{1}{\mu} \right\rceil \leq T_{\text{SP}} - T_{\text{CBR}} \quad (47)$$

which results in the following physical upper bound for dispersion:

$$\delta_{\text{phys}} = T_{\text{SP}} - (N_{\text{pairs}} - 1) \left\lceil \frac{1}{\mu} \right\rceil - T_{\text{CBR}} \quad (48)$$

We also know from Section 5.5 that given N_{VBR} VBR connections, the dispersion for a given support l satisfies:

$$\delta_{\text{max}}(l) \leq \frac{1}{\mu} \left(N_{\text{VBR}} \min\{l, \hat{B}_{\text{VBR}}\} + 1 - l\mu \right) - \left\lceil \frac{1}{\mu} \right\rceil \quad (49)$$

Hence, an upper bound for the dispersion in terms of the system parameters is given by:

$$\delta_{\text{sys}} = \max_l \delta_l(l) \quad (50)$$

Hence, for the simulation runs, the target dispersion value must satisfy:

$$\delta_{\text{max}} = \min\{\delta_{\text{phys}}, \delta_{\text{sys}}\} \quad (51)$$

No dispersion value above δ_{max} is possible.

6 Experimental Results

6.1 Two-Point CDV Results

For the two-point cell delay variation, we consider the same systems considered in [10]. The input parameters for the sources are listed in Table 1. System A consists of CBR sources and VBR-1 sources. System B consists of CBR sources and VBR-2 sources. The system parameters and derived parameters for these systems are listed in Table 2.

Here, we made the assumption that the two classes of connections occupied consecutive lines and that the CBR connections occupied the first lines. Hence, the CBR connections arrive at lines 1 through N_{CBR} and the VBR connections arrive at lines $N_{\text{CBR}} + 1$ through $N_{\text{CBR}} + N_{\text{VBR}}$. In effect, this gives the CBR sources *line* priority (not to be confused with

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

Table 1: Sources for the 2-Point CDV.

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

Table 2: System parameters and derived parameters for the 2-Point CDV.

HOL priority which would be implemented at the servers). The simulations were run by fixing the number of VBR sources to the minimum required for dispersion when mixed with a single CBR source. We then varied the number of CBR sources. Had we varied the number of VBR sources, we would have obtained curved as in [14] due to the dominant nature of the VBR class.

We first considered System A with two threshold levels $\tau = 40\%$ and $\tau = 50\%$, corresponding to queue lengths of 90 and 110 cells, respectively. The number of VBR sources was set at 6 and the number of CBR sources was varied. The simulation stopping condition was set at 100 hits per bin. The dispersion estimates are plotted in Fig. 11. The aggregate 2-Point CDV estimates (increasing curves) and the improvement factors (decreasing curves) are plotted as solid lines. The individual estimates for the CBR and VBR traffic are plotted as dotted and dashed lines, respectively. The threshold levels $\tau = 40\%$ and $\tau = 50\%$ are represented by circle and square points, respectively, and the confidence intervals are depicted with the same type point as the corresponding threshold.

For $\tau = 50\%$, at least 2 CBR sources were necessary for there to be dispersion. The confidence intervals and improvement results for the aggregate dispersion were discussed in [10]. We observe from Fig. 11 that the improvement is inversely proportional to the probability being estimated.

We observe from Fig. 11 that the 2-Point CDV for the CBR traffic is always lower than that of the aggregate traffic and vice-versa for the VBR traffic. This is due to the fact that the CBR sources occupy the first lines at the input of the switch, and hence have *line* priority. The aggregate 2-Point CDV is always in between the individual 2-Point CDV's. However, the difference between the individual 2-Point CDV's and the aggregate 2-Point CDV decreases as more CBR connections are added. This is expected since as more connections are added,

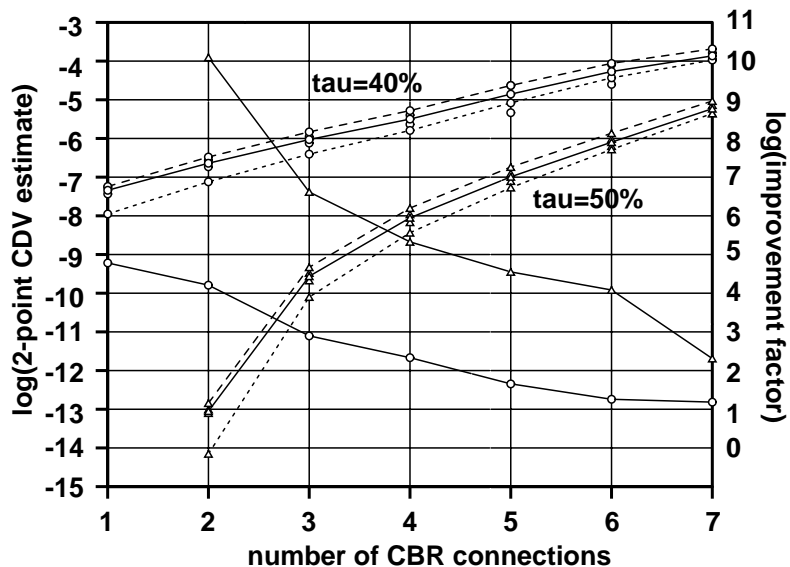


Figure 11: 2-Point CDV estimates for system A, 6 VBR sources, number of CBR sources varying (increasing solid lines are aggregate 2-Point CDV's, decreasing solid lines are improvement factors, dashed lines are for VBR, dotted lines are for CBR. \circ is for $\tau = 40\%$, \triangle is for $\tau = 50\%$).

the average queue length will increase and the effect of the *line* priority will diminish.

Next, we considered System B with two threshold levels $\tau = 85\%$ and $\tau = 95\%$, corresponding to queue lengths of 170 and 190 cells, respectively. The number of VBR sources was set at 4 and the number of CBR sources was varied. The dispersion estimates are plotted in Fig. 12. The threshold levels $\tau = 85\%$ and $\tau = 95\%$ are represented by circle and square points, respectively, and the confidence intervals are depicted with the same type point as the corresponding threshold. For $\tau = 95\%$, at least 2 CBR connections were required to cause dispersion.

The results obtained for System B are similar to that of System A. Again, the improvement factors are inversely proportional to the probability being estimated. Also, the 2-Point CDV for the CBR traffic is always lower than that of the aggregate traffic and vice-versa for the VBR traffic. However, due to the large period of System B, the addition of CBR sources does not have as large an effect on the dispersion. For the same reason, convergence of the three curves is not as apparent for System B as it was for System A.

7 One-Point CDV Results

For the one-point cell delay variation, we consider the sources for which the input parameters are listed in Table 3. We generated results for 8 systems, labeled C through J. The system parameters and derived parameters for these systems are listed in Table 4. Systems D, E

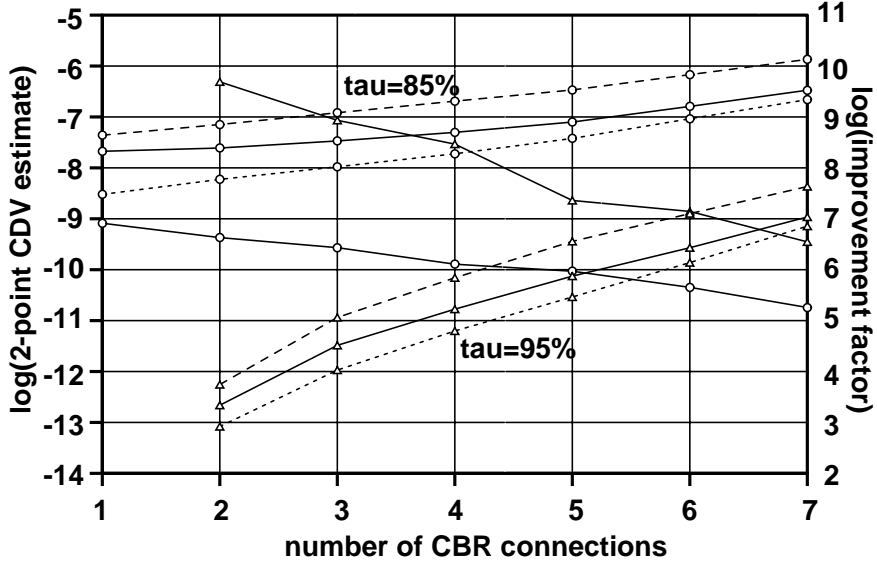


Figure 12: 2-Point CDV estimates for system B, 4 VBR sources, number of CBR sources varying (increasing solid lines are aggregate 2-Point CDV's, decreasing solid lines are improvement factors, dashed lines are for VBR, dotted lines are for CBR. \circ is for $\tau = 85\%$, \triangle is for $\tau = 95\%$).

and F are generated by varying a single VBR parameter of System C and Systems H, I and J are generated by varying a single VBR parameter of System G. System C has $\mu < 1$ and System G has $\mu > 1$. The CBR source is identical for all cases.

Here, we assumed that there is a single CBR source at line 1 and the VBR sources occupy the remaining lines. In this manner, the CBR source is the observed source and the VBR sources constitute the background traffic. The simulations were run by fixing the number of VBR sources and varying the target 1-Point dispersion parameter δ from 100% of T_{CBR} to δ_{max} . For systems C-F, $N_{\text{VBR}} = 10$ and for Systems G-J, $N_{\text{VBR}} = 14$. The simulation stopping condition was set at 200 hits per bin.

The simulation results for Systems C and D, which vary in the mean rate of the VBR source, are plotted in Fig. 13. The mean VBR rates are given by 10 Mbps and 5 Mbps for Systems C and D, respectively. In Fig. 13, the decreasing curves are the 1-Point CDV estimates and the increasing curves are the improvement factors. the curves for System C are solid and marked with circular points, whereas those for System D are dashed and marked with triangular points. For each case, the confidence intervals are marked with the same style point as the corresponding 1-Point CDV curve. The 1-Point CDV curves are initially almost linear (in log scale) and then exhibit an asymptotic behavior until the maximum dispersion value δ_{max} . A dispersion value above δ_{max} is not possible. The points enclosed in circles at the rightmost δ value of the CDV curves represent the fact that for that given δ and for a given pair, there is only one valid support and for that support $f_0 = 0$. Thus, the IS algorithm effectively identifies only the N_{pairs} vectors that cause dispersion. Then, the

Source	$\hat{\lambda}$, Mbps	$\bar{\lambda}$, Mbps	\hat{B} , cells	T , slots
CBR-2	5	5	1	1
VBR-3	500	10	20	1000
VBR-4	500	5	20	2000
VBR-5	250	10	20	500
VBR-6	500	10	10	500
VBR-7	100	5	10	200
VBR-8	100	2	10	500
VBR-9	200	5	10	400
VBR-10	100	5	8	160

Table 3: Sources for the 1-Point CDV.

System	Sources	$\hat{\mu}$, Mbps	K , cells	T_{SP} , slots	μ , cells/slot
C	CBR-2, VBR-3	120	200	1000	0.24
D	CBR-2, VBR-4	120	200	2000	0.24
E	CBR-2, VBR-5	120	200	500	0.48
F	CBR-2, VBR-6	120	200	500	0.24
G	CBR-2, VBR-7	120	200	200	1.2
H	CBR-2, VBR-8	120	200	500	1.2
I	CBR-2, VBR-9	120	200	400	0.6
J	CBR-2, VBR-10	120	200	160	1.2

Table 4: System parameters and derived parameters for the 1-Point CDV.

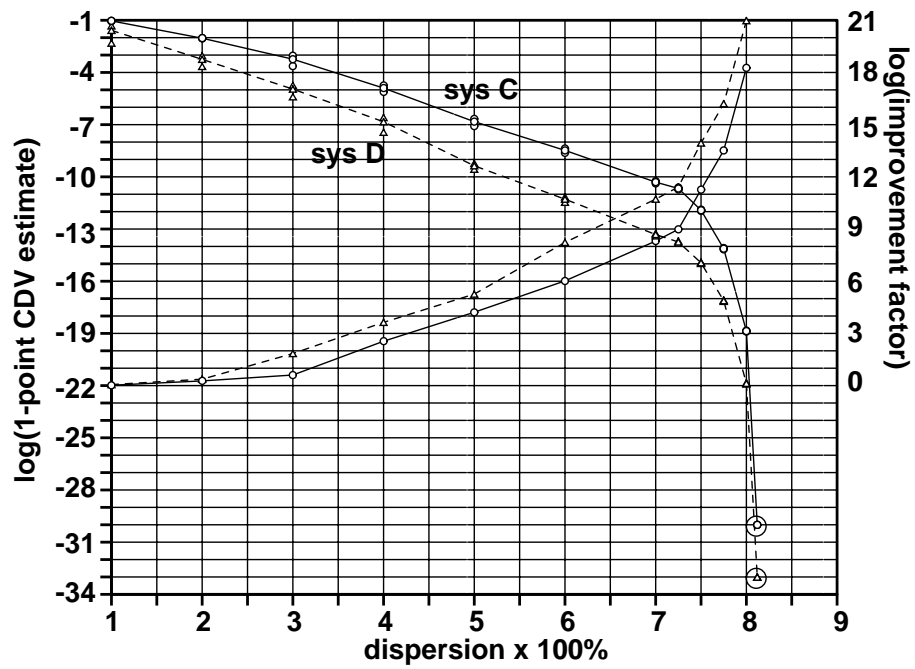


Figure 13: 1-Point CDV estimates for Systems C and D, 10 VBR sources w/ $\bar{\lambda}$ varying (decreasing lines are 1-Point CDV's, increasing lines are improvement factors. \circ and solid lines are for System C, \triangle and dashed lines are for System D).

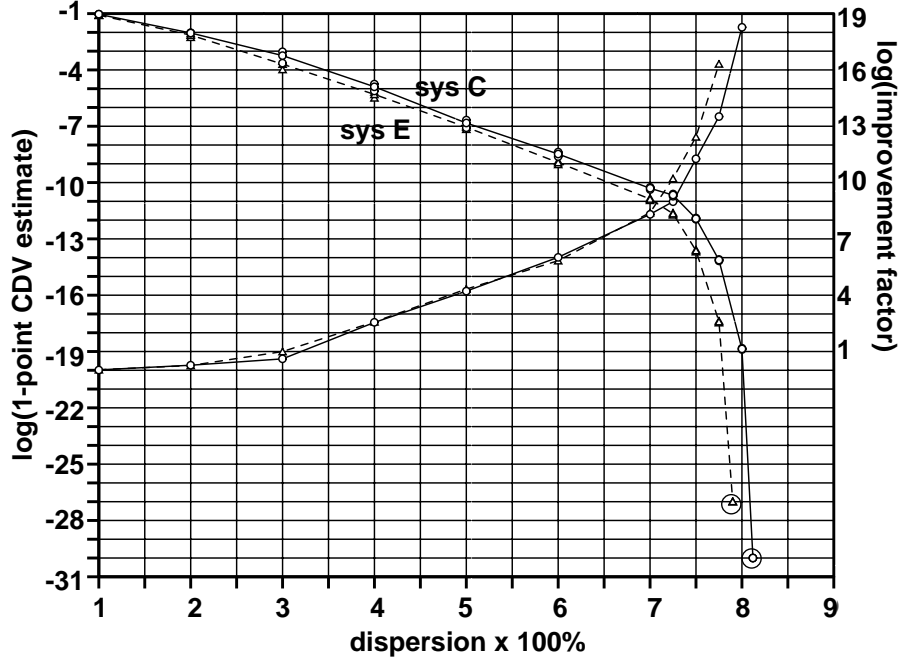


Figure 14: 1-Point CDV estimates for Systems C and E, 10 VBR sources w/ $\hat{\lambda}$ varying (decreasing lines are 1-Point CDV's, increasing lines are improvement factors. \circ and solid lines are for System C, \triangle and dashed lines are for System E).

1-Point CDV probability can be found exactly using:

$$P_{DP} = \frac{1}{(T_{SP})^{N_{VBR}}} \quad (52)$$

For the exact points, the improvement generated is many orders of magnitude larger than for the other points and hence is not shown in Fig. 13.

As seen from Fig. 13, the improvement factors are again inversely proportional to the 1-Point CDV probability being estimated. The 1-Point CDV is higher for System C, since System C has a higher mean VBR rate. The difference between the curves increases as δ is increased from 100% until it becomes constant as the asymptote is approached.

Systems C and E vary in the VBR peak rate and Systems C and F vary in the VBR burst length. The simulation results for these systems are plotted in Fig.'s 14 and 15, respectively.

For both cases, we once again observe that the improvement in simulation efficiency is inversely proportional to the probability being estimated. We also observe similar properties for the 1-Point CDV curves.

For System C, $\hat{\lambda}_{VBR} = 500$ Mbps, compared to $\hat{\lambda}_{VBR} = 250$ Mbps for System E. From Fig. 14, we observe that the effect of the peak VBR rate on the 1-Point CDV is not very significant, which is an expected result. However, as seen in Fig. 15, the burst length of the VBR source can severely affect the 1-Point CDV. Compared to System C with $\hat{B}_{VBR} = 20$

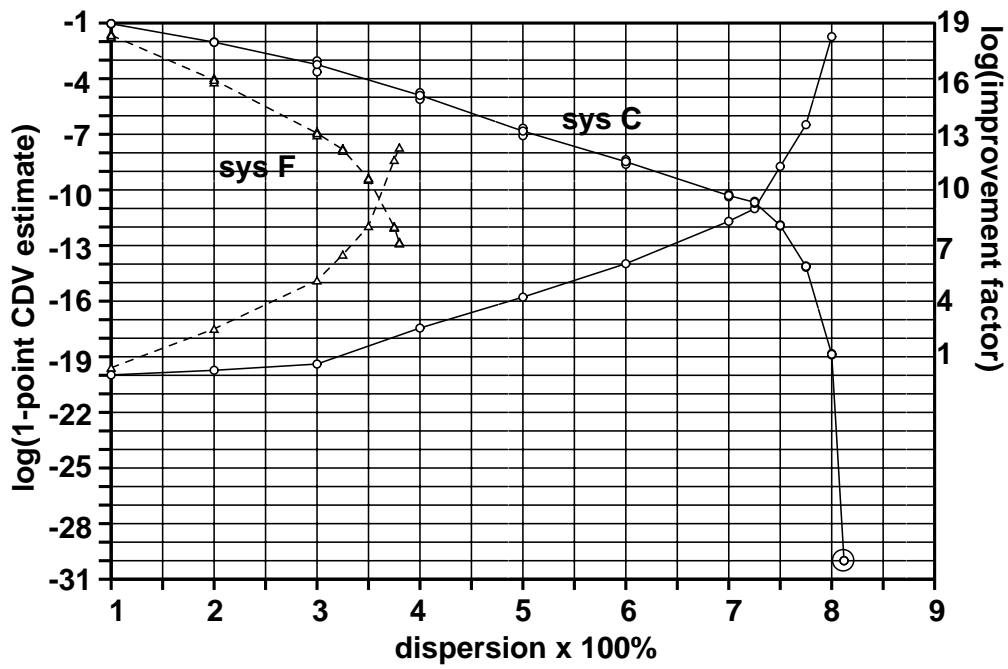


Figure 15: 1-Point CDV estimates for Systems C and F, 10 VBR sources w/ \hat{B} varying (decreasing lines are 1-Point CDV's, increasing lines are improvement factors. \circ and solid lines are for System C, \triangle and dashed lines are for System F).

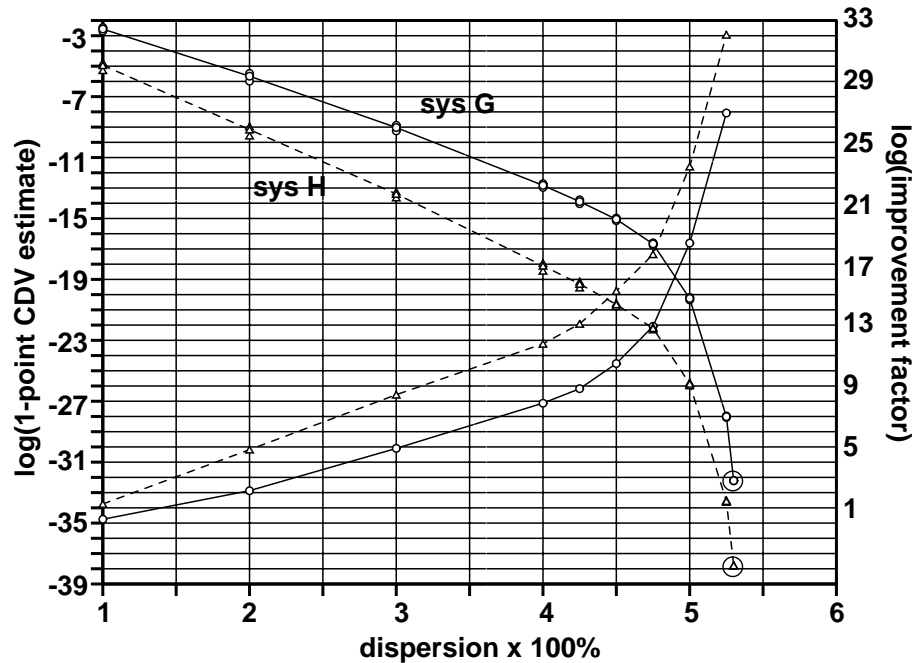


Figure 16: 1-Point CDV estimates for Systems G and H, 14 VBR sources w/ $\bar{\lambda}$ varying (decreasing lines are 1-Point CDV's, increasing lines are improvement factors. \circ and solid lines are for System G, \triangle and dashed lines are for System H).

cells, System F had $\hat{B}_{\text{VBR}} = 10$ cells. The 1-Point CDV for System F was significantly lower than that for System C. Also, the δ_{max} for System F was less than half of that for System C.

The 1-Point CDV probability estimates and improvement factors for Systems G and H, G and I, G and J are plotted in Fig.'s 16-18. For Systems G-J, the service rate μ is greater than that for Systems C-F. The 1-Point CDV curves exhibit the same asymptotic behavior and the improvements are again inversely proportional to the probability being estimated.

However, when μ is increased, more VBR sources are necessary to cause dispersion. Hence, we have 14 VBR sources for Systems G-J. Also, δ_{max} and the 1-Point CDV probabilities decrease as μ is increased. Since the probabilities decrease, we also observe an increase in the improvement factors.

The general behavior of the curves when the VBR parameters are varied is similar to those of Systems C-F. However, for Systems G-J, greater changes in the given parameters are required to cause changes similar to the ones in Systems C-F for the 1-Point CDV probabilities.

8 Conclusion

In this paper, we considered the problem of estimating the 1-Point CDV for CBR sources with background VBR traffic and the 2-Point CDV for mixed CBR and VBR sources in ATM

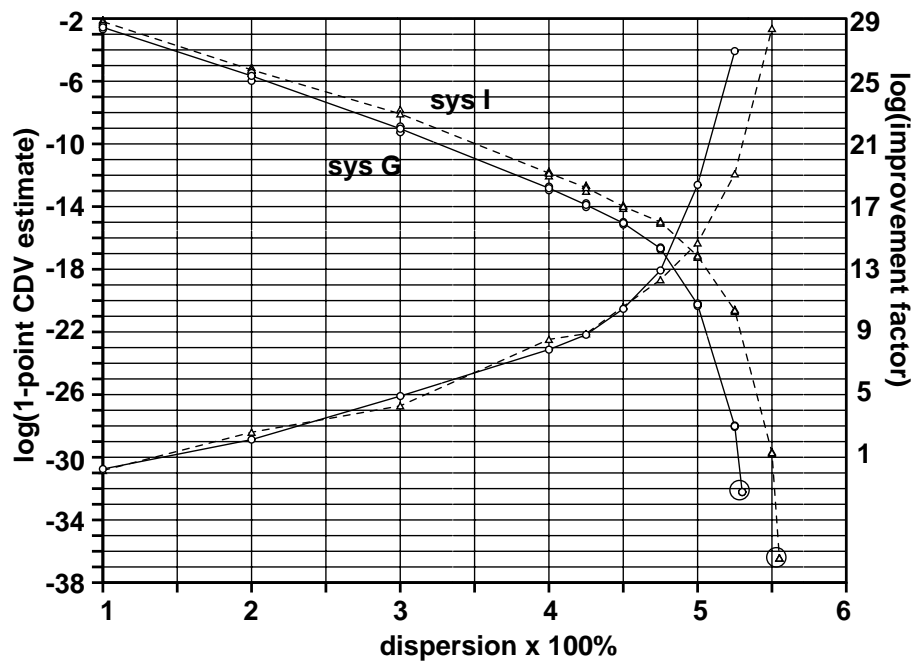


Figure 17: 1-Point CDV estimates for Systems G and I, 14 VBR sources w/ $\hat{\lambda}$ varying (decreasing lines are 1-Point CDV's, increasing lines are improvement factors. \circ and solid lines are for System G, \triangle and dashed lines are for System I).

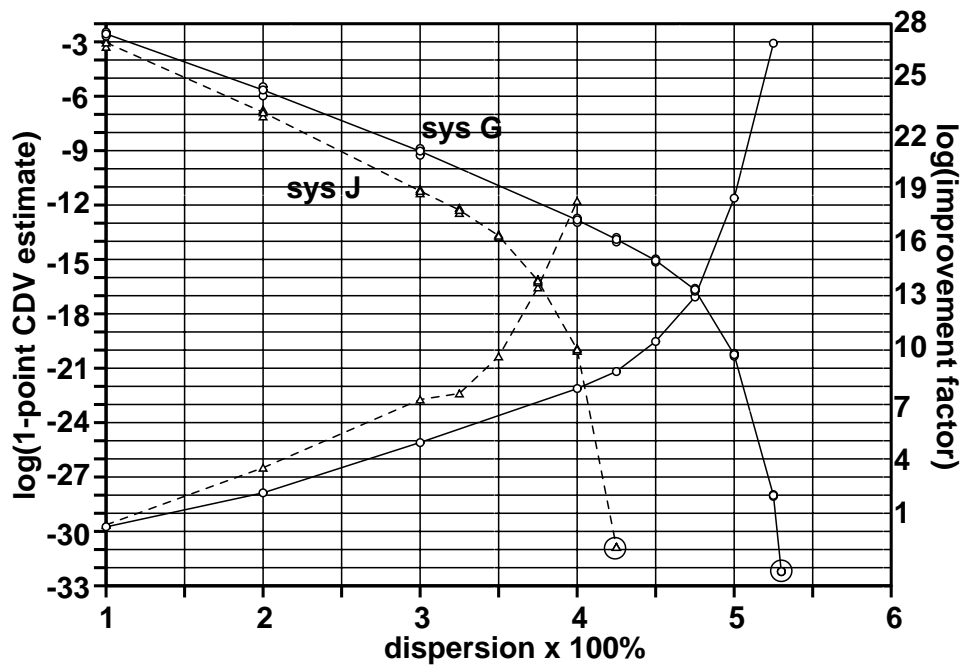


Figure 18: 1-Point CDV estimates for Systems G and J, 14 VBR sources w/ \hat{B} varying (decreasing lines are 1-Point CDV's, increasing lines are improvement factors. \circ and solid lines are for System G, \triangle and dashed lines are for System J).

networks. We used the ATM Forum standardized connection traffic descriptors to characterize the input traffic (operational approach) and we considered dispersion as a measure of CDV.

We developed efficient simulation methods to estimate the rare tail dispersion probabilities for both cases. In each case, we developed multinomial formulations to effectively remove correlations (due to bursty traffic) between important events. We extended our previous work utilizing Importance Sampling in the case of delay threshold probabilities for heterogeneous traffic to estimate the 2-Point CDV probabilities. Subsequently we developed a new efficient simulation method using Importance Sampling to estimate the 1-Point CDV probabilities.

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

References

- [1] ATM User-Network Interface Specification, Version 3.0. The ATM Forum, September 10, 1993. The ATM Forum.
- [2] 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.
- [3] P. Boyer, A. Dupuis, A. Gravey, and J. Pitie. The Output Process of the Single Server Queue with Periodic Arrival Process and Deterministic Service Time. *F. Baccelli and G. Fayolle (eds.), Lecture Notes in Control and Information Sciences, vol. 60*, pages 408–438, 1984.
- [4] H. Saito. *Teletraffic Technologies in ATM Networks*. Boston: Artech House, 1994.
- [5] J. W. Roberts and F. Guillemin. Jitter in ATM Networks and its Impact on Peak Rate Enforcement. *Performance Evaluation*, 16(1–3):35–48, 1992.
- [6] S. Blaabjerg. Cell Delay Variation in a FIFO Queue: A Diffusion Approach. *H. G. Perros and Y. Viniotis (eds.), IFIP Trans. C-21: High Speed Networks and Their Performance*, pages 237–256, 1994.
- [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] 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.

- [9] C. Bisdikian, W. Matragi, and K. Sohraby. A Framework for Jitter Analysis in Cell Based Multiplexers. *Performance Evaluation*, 22(3):257–277, 1995.
- [10] A. A. Akyamac and J. Keith Townsend. Efficient Simulation of Delay Threshold Probabilities in ATM Switches with Heterogeneous Traffic. To appear in the *Proc. of the Int. Conf. Commun., ICC '97*.
- [11] A. Gravey and P. Boyer. Cell Delay Variation Specification in ATM Networks. In *Proc. Proc. Modelling and Performance Evaluation of ATM Technology*, La Martinique, 1988.
- [12] R. O. Onvural. *Asynchronous Transfer Mode Networks: Performance Issues, 2nd. Ed.* Boston: Artech House, 1995.
- [13] J. D. Russell. Multimedia Networking Performance Requirements. *Y. Viniotis and R. O. Onvural (eds.), Asynchronous Transfer Mode Networks*, pages 187–197, 1993.
- [14] A. A. Akyamac and J. Keith Townsend. Efficient Simulation of Delay Threshold Probabilities in ATM Networks with ON/OFF Arrivals. In *Proc. IEEE Global Telecom. Conf., GLOBECOM '96*, pages 612–616, London, Nov. 1996.