Estimating End-to-End Cell Delay Variation in ATM Networks *

Ibrahim Korpeoglu Satish K. Tripathi
Institute for Advanced Computing Studies
Department of Computer Science
University of Maryland
College Park, MD 20742
{korpe,tripathi}@cs.umd.edu

Xiaoqiang Chen
ATM Networks Research Department
Bell Laboratories
Lucent Technologies
Holmdel, NJ 07733
xchen@bell-labs.com

CS-TR-3767, UMIACS-TR-97-27

Abstract

Cell delay variation (CDV) is one of the quality of service parameters that can be negotiated between applications and an ATM network. The network should check during connection setup, as part of call admission control, whether it can satisfy the requested CDV of an application. For this comparison, the network should estimate the end-to-end CDV that it can support, by using local information about cell delays and delay variations in switches. An accurate estimation of the end-to-end CDV is important for decreasing call-blocking probability and increasing network utilization. In this article, we will first describe, evaluate, and identify the short-comings of three proposed methods for end-to-end CDV estimation. Then we will present a new method based on Chernoff bound and compare it to the other methods. The Chernoff method is promising since it has good accuracy and applicability under current signalling support for ATM networks.

1 Introduction

ATM networks support a wide variety of applications by providing them with different service classes. The applications may range from reliable, best-effort data transfer applications, to delay-sensitive, bandwidth-guarantee-demanding real-time audio and video applications. The network can provide quality of service (QoS) to these applications in terms of loss rate, delay, and delay variation. The quality of service level that an application requires is expressed by setting up the corresponding parameters in the SETUP message of the UNI signaling protocol [UNI96].

*This work is supported by Lucent Technologies, Bell Laboratories.

The ATM Forum specifies three quality of service parameters that can be negotiated between the applications and an ATM network: *maximum cell transfer delay* (maxCTD), *peak-to-peak cell delay variation* (CDV) and *cell loss ratio* (CLR) [TM96]. Depending on the service class that an application wants to use, a combination of these parameters needs to be set by the application. The values of these parameters depend on the level of quality of service that an application desires. These parameters are then propagated hop-by-hop towards the destination by using UNI and NNI signalling during connection setup [UNI96] [PNN96] [BIC94]. Each network switch on the connection path checks whether the requested level of service can be supported by the network resources up to that switch. If it determines that the service can be supported, then it passes the connection request to the next switch; otherwise it initiates the release of the connection, and the request is rejected. The application may try to re-negotiate with the network by lowering the level of requested quality of service to establish a new connection.

The network switches should have knowledge about the level of quality of service that they can support. During call setup, by using the local information from switches, the network should estimate the supportable quality of service level over multiple switches and finally the end-to-end quality of service level from source to destination. Thereby, a comparison can be made to decide whether or not to accept the connection. The quality of service values over multiple switches can also be called *accumulated* quality of service values, since they can be considered as the accumulation of local quality of service values along a connection path.

Estimation of end-to-end quality of service values is not only used by signalling and call admission control. It can also be used by QoS routing algorithms [PNN96] in order to select the best path to a destination.

In this article, we will focus on the estimation of one important quality of service parameter: *peak-to-peak cell delay variation (CDV)*. CDV is an important parameter for continuous stream applications like audio and video, which require bounded cell inter-arrival times for a good quality perception by users.

An accurate estimation of end-to-end CDV is important for decreasing call-blocking probability and increasing network utilization [HB95]. Overestimation is not desirable, since it will increase call-blocking probability and decrease the network utilization. Since accumulation is done as part of signalling, there are also constraints in terms of signalling parameters that can be used for accumulation [UNI96] [BIC94]. Complex algorithms are not desirable, since they may cause an increase in call setup time.

Three methods have been proposed for estimating end-to-end CDV: the simple method [TM96], the square-root method [BIC94], and the asymptotic method [Bor95]. As shown in section 4, none of these provide a satisfactory solution in terms of correctness, accuracy, feasibility, and simplicity. In this article, we propose a new method based on the Chernoff bound that has a high accuracy, simple implementation, and is feasible to implement under current signalling support.

The outline of the paper is as follows: Section 2 gives the formal definition of peak-to-peak CDV and the requirements for a good CDV accumulation method. Section 3 describes three proposed CDV accumulation methods. In section 4, these three methods are evaluated and their shortcomings are identified. In section 5, a new method based on the Chernoff bound is given, and its performance is evaluated against the other methods in section 6. Finally, section 7 gives a summary of evaluations and comparisons.

2 Definitions and Requirements

Cell transfer delay (CTD) is the total delay that a cell experiences along a connection path from source to destination. If the path consists of only one switch, then it is the total delay experienced through that switch. CTD has 2 parts: fixed delay and queueing delay. Fixed delay consists of propagation delays in links, switching delays, and transmission delays. Figure 1 gives a model for the probability density function of the CTD [TM96].

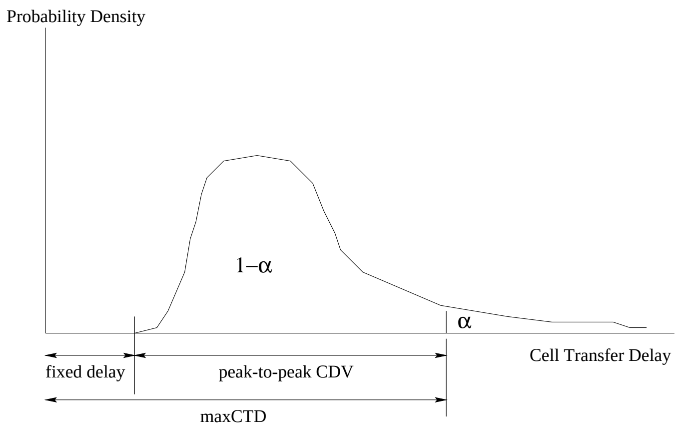


Figure 1: CTD probability density model

The ATM Forum traffic management specification defines the peak-to-peak CDV as follows [TM96]:

Definition: *Peak-to-peak CDV is the* $(1-\alpha)$ *quantile* [1] *of the cell transfer delay (CTD) minus the fixed CTD, that could be experienced by any delivered cell on a connection during the entire connection holding time. The term "peak-to-peak" refers to the difference between the best and worst case of CTD, where the best case is equal to the fixed delay, and the worst case is equal to a value likely to be exceeded with probability no greater than* α.

The value of α is expected to be a very small number in the range $[10^{-4}, 10^{-12}]$. Currently, the ATM Forum has no approved specification about how to determine the value of α in a switch and in multiple switches on a connection path. We will take the value of α to be the same for all switches on a path. We assume that switches either will use a predetermined value for α or will derive it from QoS parameters, like cell loss ratio. Currently there is no signalling support to explicitly set the value of α by the user at connection initiation time and propagate it through the switches.

Throughout the next sections, we will refer to peak-to-peak CDV simply as *CDV*. Since its value is dependent on α, we will also denote it as $cdv(\alpha)$.

Definition: *Local CDV is the the peak-to-peak CDV in one particular switch.*

Definition: *End-to-end CDV is the peak-to-peak CDV along a connection path from source to destination. If the connection is over one switch, then it is the same as the local CDV on that switch.*

For end-to-end CDV over multiple hops, CTD denotes the end-to-end cell transfer delay. The-

[1] $(1-\alpha)$ quantile of a random variable X is the value x, such that $P(X \geq x) = \alpha$.

oretically, the probability density function of the end-to-end CTD can be obtained by taking the convolution of individual CTD probability density functions in switches. But this is not a feasible method to derive the end-to-end CDV. It is also not practical to take end-to-end measurements for cell delay to obtain a histogram and derive the end-to-end CDV. This is because the CTD probability density function is time varying and α is too small, requiring many samples to be collected frequently enough for an accurate estimation.

Therefore, it is better and simpler to estimate the end-to-end CDV by using local knowledge about delay, delay variation, and peak-to-peak CDV in switches. We will call the end-to-end CDV estimate from this local information also as *accumulated CDV*, and the methods to be used for estimations as *CDV accumulation methods*. The derivation of the local information about cell delay and distributions is itself an important problem [Rop96, CPB93, SSD93], which we will not address here.

A good CDV accumulation method should satisfy the following properties:

- The accumulation method for end-to-end CDV should never underestimate the actual value. Otherwise, the negotiated quality of service for connections cannot be satisfied.

- It is also not desirable to overestimate the end-to-end CDV. The buffer requirement at the destination is determined by the estimation of the end-to-end CDV. If it is overestimated then the buffers will be unnecessarily large. The buffer requirement is directly proportional to CDV [AKRS94]. Also, overestimation causes connections with valid quality of service requirements to be rejected. This is unfair for a connection request that has greater CDV requirement than the actual value but less than the estimated value. Denying connections with supportable quality of service requirements increases the call blocking probability for the network and decreases the network utilization [HB95], [Wri95].

- CDV accumulation is used at connection setup time as part of the call admission control algorithm and the P-NNI route selection algorithm. It should be simple enough so that it will not increase the call setup time.

- CDV is accumulated from source to destination using UNI and NNI signalling. Therefore, the accumulated CDV value is passed from one switch to another as a parameter in the SETUP message of the signalling protocol. The accumulation algorithm may need other parameters to be signaled in order to estimate the end-to-end CDV accurately, and these should be supported by signalling protocols.

3 Related Work

In this section, we will describe three CDV accumulation methods that are proposed to the ATM Forum. These methods are the simple method [TM96], the square-root method [BIC94], and asymptotic method [Bor95].

Simple Method: The simple method estimates the end-to-end CDV as the sum of individual local CDVs along the path from source to destination. If there are N switches along the path, then the total end-to-end CDV over these N switches is estimated by:

$$cdv_{tot}(\alpha) = \sum_{i=1}^{N} cdv_i(\alpha) \tag{1}$$

Here, $cdv_i(\alpha)$ denotes the local CDV in switch i. The simple method is not a theoretical upper-bound for end-to-end CDV. It is only a heuristic that works for common distributions like normal, gamma etc. [Bor95] shows an example CTD pdf, for which the simple accumulation method does not bound the end-to-end CDV.

Square-root Method: The intuition behind the square-root method is the assumption that local CDVs in switches equal some constant times the standard deviation of CTD, where the constant is the same for all switches. The standard deviation of the sum of N independent random variables is the square-root of the sum of the squares of individual standard deviations. With this assumption, end-to-end CDV can be computed in the same way. Assuming the delays in switches are independent, then the total end-to-end CDV is estimated as:

$$cdv_{tot}(\alpha) = \sqrt{\sum_{i=1}^{N} cdv_i(\alpha)^2} \tag{2}$$

The square-root method can also accommodate the correlation between the delays in switches [Noo95]. If the correlation coefficient between the delay in the upstream nodes and the current node is ρ, then the end-to-end CDV from source to node i (including node i) is estimated by:

$$cdv_{tot_i}(\alpha)^2 = cdv_{tot_{i-1}}(\alpha)^2 + cdv_i(\alpha)^2 + 2 \times \rho \times cdv_{tot_{i-1}}(\alpha) \times cdv_i(\alpha) \tag{3}$$

Here, cdv_{tot_i} denotes the estimated CDV up to node i. However, it is not easy to estimate the correlation factor between the delay in upstream switches and the delay in the current switch.

CDV may not be only related to standard deviation of the delay but also to the mean of the delay and to the distribution of the delay. Therefore, the constant to be multiplied by standard deviation may not be the same for all switches. Hence, it is not enough to consider only the variance of delay for an accurate and correct estimate of CDV.

Asymptotic Method: The asymptotic method, proposed by [Bor95], uses both mean and variance of cell transfer delay and actual local CDV in each switch to estimate the end-to-end CDV. The basis of the asymptotic method is the central limit theorem: whatever the distributions of individual random variables are, sum of N of them tends to be normally distributed as N gets large. The error due to finite value of N is compensated by adding the maximum difference between the estimated and actual CDV in the switches along the path. The end-to-end CDV, assuming independent delays in switches, is estimated by:

$$cdv_{tot}(\alpha) = \sum_{i=1}^{N} \mu_i + \sqrt{\sum_{i=1}^{N} {\sigma_i}^2} \times Q^{-1}(\alpha) + \max_{1 \leq i \leq N} \{cdv_i(\alpha) - (\mu_i + \sigma_i \times Q^{-1}(\alpha))\} \tag{4}$$

Here, μ_i is the mean and ${\sigma_i}^2$ is the variance of the delay in switch i. $Q^{-1}(\alpha)$ is the (1-α) quantile of the standard normal distribution. Since this is fixed, it can be stored as a table for different α values. The term $\{cdv_i(\alpha) - (\mu_i - \sigma_i \times Q^{-1}(\alpha))\}$ is the discrepancy in a switch: the difference between the actual CDV and the asymptotic method estimate of CDV. The heuristic for error compensation is taking the maximum of these discrepancies among all the nodes.

4 Evaluation of Proposed Methods

In this section, we evaluate the three proposed methods in terms of how accurately they estimate the actual end-to-end CDV, their complexity, and their feasibility under current signalling support.

In our experiments we will make assumptions about the queueing delay distributions in switches. This will allow us to compare the performance of the estimation methods numerically. Although this is not sufficient for a complete evaluation, it will give an insight about the relative performances of the methods.

In the experiments, the local CDV in a switch is computed by taking the $(1-\alpha)$ quantile of the assumed queueing delay distribution. We do not need to consider the fixed delay in this case, since the distribution characterizes only the queueing delay. The actual end-to-end CDV will be computed by first finding the end-to-end queueing delay distribution, and then taking its $(1-\alpha)$ quantile. The end-to-end queueing delay distribution is obtained by the convolution of individual delay distributions. Matlab [Mat94] is used for all the numerical analysis.

A similar performance evaluation of these methods is presented in [Bor95], but it is calculated only for a single α value and only for identical exponential delay distributions. We will extend our comparison also to Erlang (a special case of gamma) distribution in switches. As shown in [NLG96] by simulation, the delay and delay variance for CBR traffic exhibits a gamma distribution. Hence, gamma is a good model for delay distributions.

We start our evaluation by assuming identical exponential distributions in switches. Although this is far from being a practical assumption, it provides a rough picture of how well the methods perform. We further assume that the delays are uncorrelated (independent delay distributions). The density function of an exponential distribution is given by: $f(x) = \lambda e^{-\lambda x}$. We assume that the scale parameter (λ) is equal to 1 for each switch. The value of λ and the unit of the delay is irrelevant for comparison purposes, since they will scale both the actual CDV and the estimates, as proved in [Bor95]. Therefore, we will not use any unit in our computations and comparisons.

The $(1-\alpha)$ quantile of an exponential distribution, which gives the local CDV, is given by $-\ln(\alpha)/\lambda$. The convolution of N exponential distributions is an Erlang distribution. By taking the $(1-\alpha)$ quantile of the resulting Erlang distribution, we obtain the actual end-to-end CDV over N hops.

We compute the actual end-to-end CDV values and their estimates. The results are shown in figure 2. Figure 2-a shows the actual end-to-end CDV for different number of hops; Figures 2-b, 2-c, and 2-d show the simple, square-root, and asymptotic method estimations. As α decreases, both the actual CDVs and their estimates increase. This is because α is the measure of the guarantee that the cell delay will not exceed the CDV threshold. If we have a smaller α (stricter guarantee), then the CDV threshold should be large enough to accommodate it (see figure 1). The scales of the four graphs are quite different. The simple method estimate is much higher than the square-root and asymptotic method estimates. The difference between the estimates and the actual end-to-end CDV can be seen more clearly in figure 3. Figure 3-a compares the actual CDV with the simple method estimate. Simple method estimates are much higher than the actual values. Figure 3-b compares square-root estimate with the actual CDV. As also shown in [Bor95], it is interesting to see that the estimate goes below the actual CDV after certain number of hops. This is undesirable, since underestimating the actual CDV may cause violation of quality of service. In figure 3-c, the asymptotic method estimate is compared with the actual CDV. The Asymptotic method estimates are very close to the actual CDV.

The relative errors of the estimates with respect to the actual CDV are shown in figure 4 for different α values. The relative error of the simple method rapidly increases as the number of hops increases. The relative error of the square-root method becomes negative after some number of hops. The relative error of the asymptotic method decreases as the number of hops increases, although initially it increases.

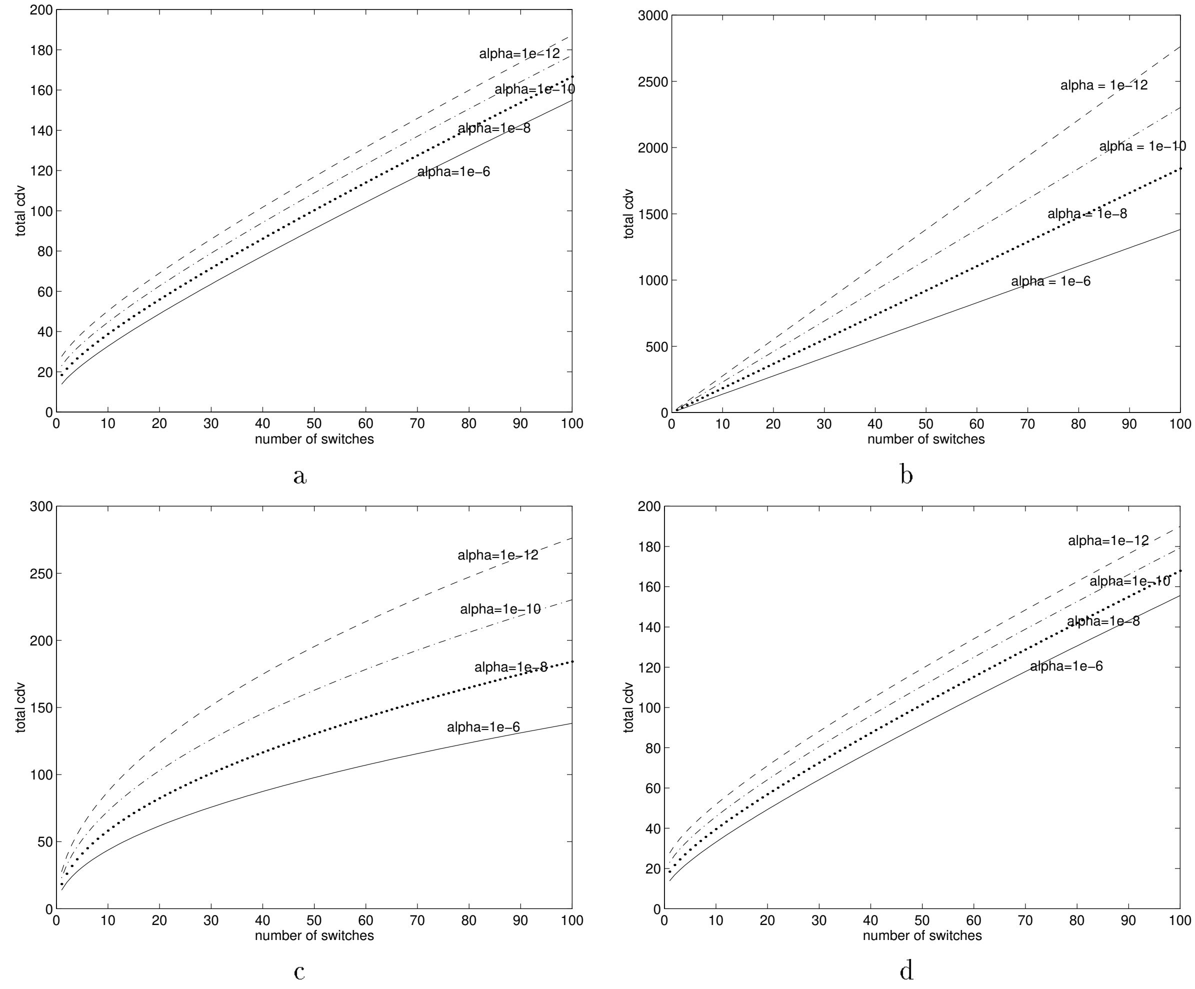
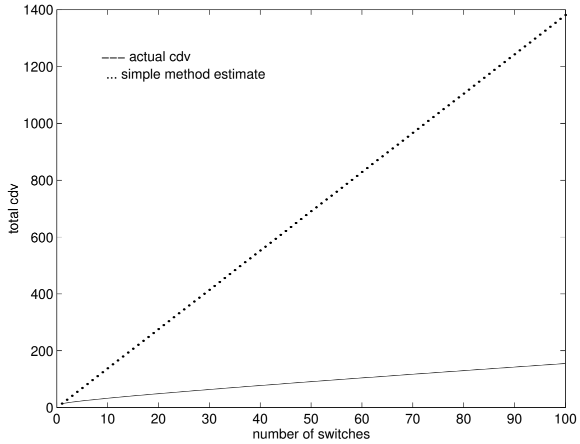
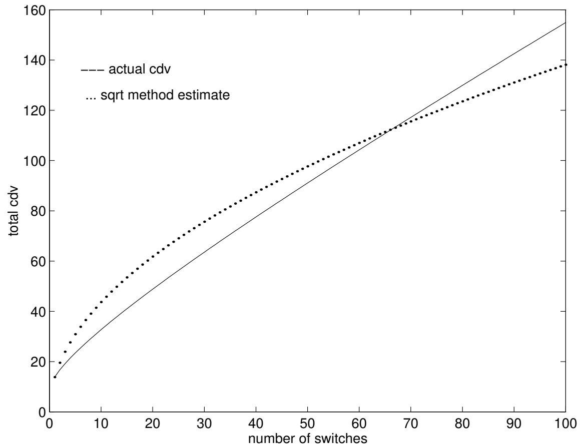


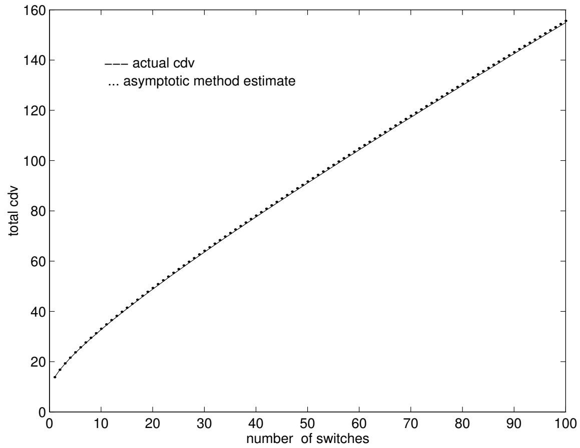
Figure 2: End-to-end CDV versus number of hops. a) Actual end-to-end CDV, b)Simple method estimate, c)Square-root method estimate, d)Asymptotic method estimate.



a)

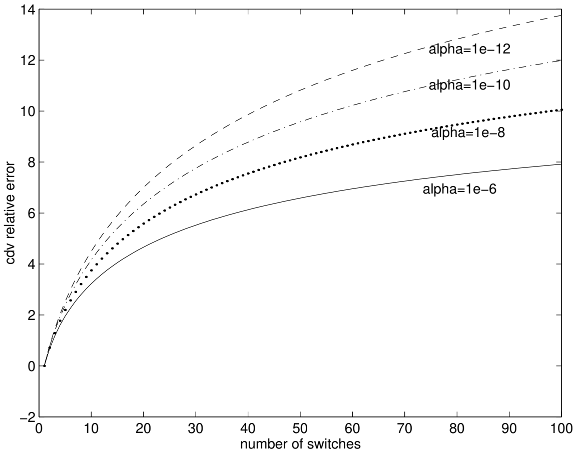


b)

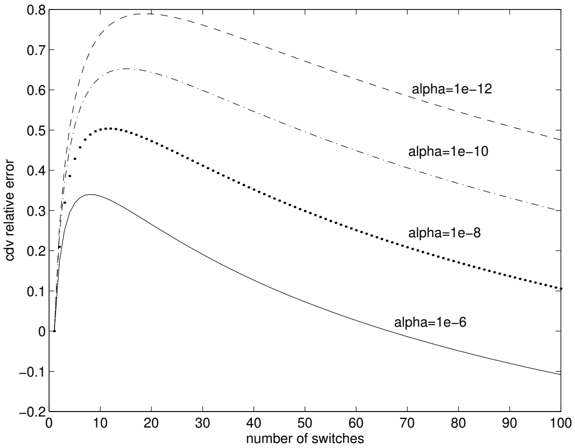


c)

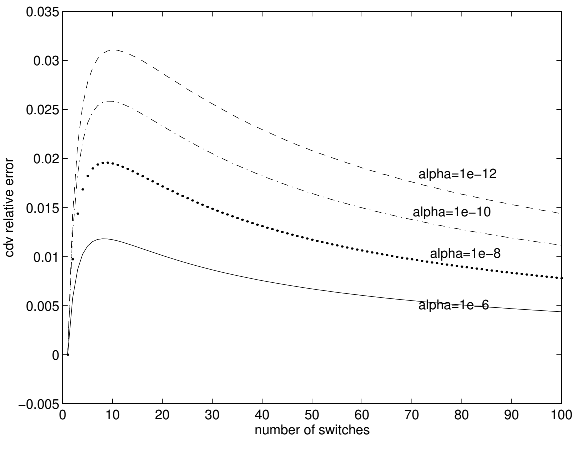
Figure 3: a) Actual end-to-end CDV versus simple method estimate, b)Actual CDV versus square-root methods estimate, c)Actual CDV versus asymptotic method estimate



a)



b)



c)

Figure 4: a) Relative error of simple method, b) Relative error of square-root method, c) Relative error of asymptotic method.

Having studied a simple network model and gained insight about the relative performances of the methods, we will now consider the case where the delays in switches are not exactly the same. We assume that the delays in switches have Erlang distributions, but with different shape parameters. The Erlang distribution is a special case of gamma distribution and has two parameters: a scale parameter (λ) and a shape parameter (r). We fix the scale parameter to 0.05 and vary the shape parameter between 2 and 5. Again choices of the parameter values do not convey any delay information about real-world but serve for comparison purposes. As stated earlier, the unit of the delay is irrelevant for the comparisons.

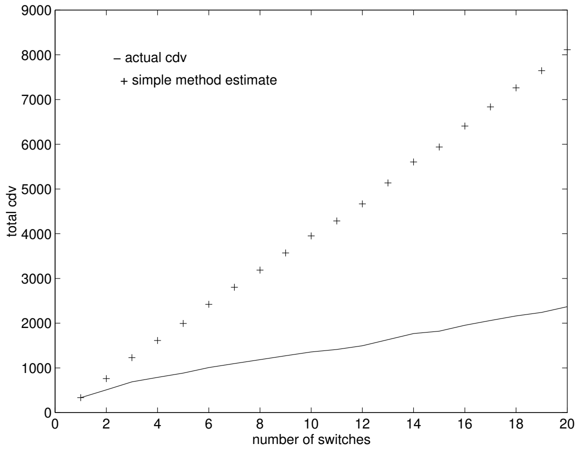
Figure 5 shows the actual CDV and the estimates. The simple method again grossly overestimates the actual CDV, whereas the asymptotic method is very close. The square-root method starts underestimating much more quickly than in figure 3-c. This is because distributions in each switch have a shape parameter greater than 1, as opposed to an exponential distribution, for which it is 1. This makes the accumulation of shape parameters over multiple hops quickly exceed the cut-off shape parameter value, after which the underestimation starts. Figure 6 shows the relative errors of estimates. The absolute values of relative errors in this case are different than the ones in figure 4, but the qualitative information provided is the same.

So far, we have evaluated the three methods in terms of how accurately they estimate the end-to-end CDV. The simple method grossly overestimates the end-to-end CDV, which can increase the call blocking probability and decrease the network utilization. The square-root method starts underestimating the CDV after some number of hops, which may cause violation of QoS provision. For this reason, the square-root method is not promising for use in ATM networks. ATM networks are designed to be very scalable; a connection path may have arbitrarily large number of switches. Therefore, we will not include the square-root method in comparisons with our proposal in section 6.

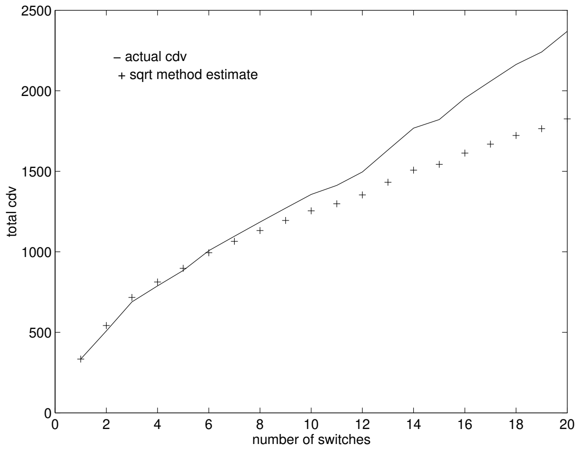
The asymptotic method performs best in terms of accuracy. The problem with the asymptotic method is that it is not feasible under current signalling protocols. Both UNI and NNI signalling supports only one parameter for CDV accumulation. Only one value can be passed from switch to switch and accumulate the end-to-end CDV. The asymptotic method requires at least three parameters to be signaled: the total mean of the delays, the total variance of the delays, and the maximum discrepancy factor (see equation 4). For this reason, the asymptotic method is not practical. The asymptotic method also can not handle the correlation of the delays in switches. If there is a correlation, the end-to-end delay variance is not the addition of individual delay variances, as used in the asymptotic method. The computation overhead of the asymptotic method is also significant. In requires the estimation of mean and variance of the delay in a switch and also the estimation of the local CDV. It also requires a table lookup for the quantiles of standard normal distribution.

Accurate estimation of local CDV is a significant problem and is more complex than estimating the mean and variance of the delay. This is because α is a very small number and the estimation of the $(1-\alpha)$ quantile of CTD requires a large number of samples to be collected. This should also be done frequently enough to capture the time-dependent variation in CTD pdf. The CTD pdf changes when new calls are added and dropped and is also dependent on the current traffic activity of existing calls. Both the asymptotic method and the simple method rely on the accurate estimation of local CDVs. Therefore, the asymptotic method is a quite complex method and is hard to realize.

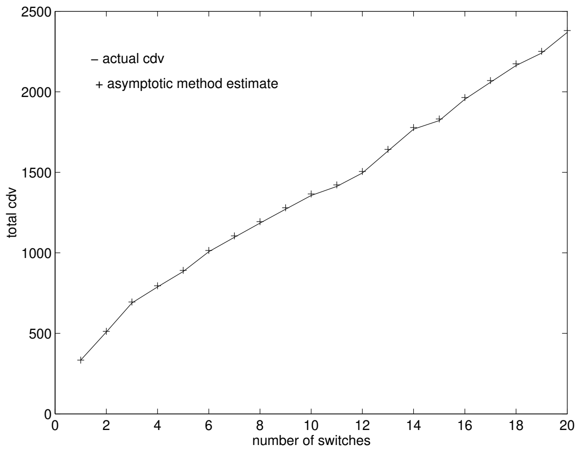
Since the simple method grossly overestimates the total CDV and the asymptotic method is too complex, there is a need for a simpler but also accurate accumulation method. In the next section, we provide a new accumulation method based on the Chernoff bound. As will be shown,



a)

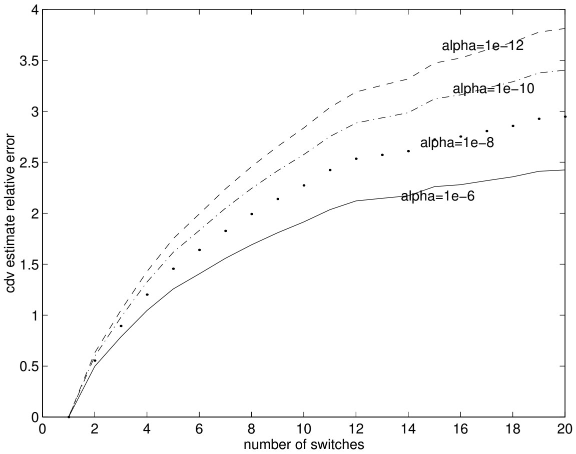


b)

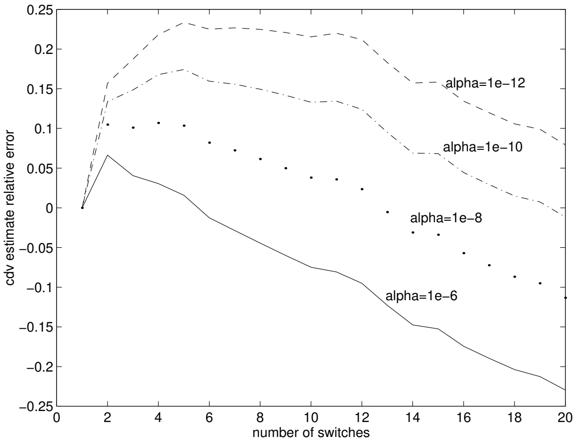


c)

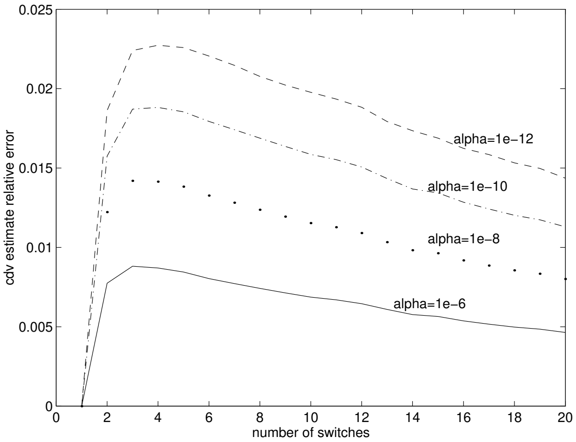
Figure 5: Comparison of the three methods against actual CDV. Each switch has Erlang distributed queueing delay with different shape parameters.



a)



b)



c)

Figure 6: Relative errors of estimates. a) Simple method, b) Square-root method, c) Asymptotic method.

the Chernoff method estimates CDV better than the simple method and is less complex and more applicable than the asymptotic method.

5 Chernoff Method

So far, all the methods rely on the accurate estimation of the local CDV in switches, in order to estimate the end-to-end CDV. However, as we argued earlier, estimation of local CDV by direct measurements is a difficult and costly operation, which requires lots of samples to be collected frequently. This makes a model based CDV estimation method, based on a delay distribution model in switches, preferable.

[NLG96] shows, by simulation study of the CBR traffic, that a gamma distribution adequately models cell delay in an ATM switch. Our method is inspired by this observation and will be based on the assumption that the queueing delay in a switch is gamma distributed. Then we can find the moment generating function (MGF) of the end-to-end delay distribution, apply the Chernoff bound to it, and obtain the end-to-end CDV.

If the delays are gamma distributed, then we can find the parameters of the gamma distribution from the mean (μ) and variance (σ^2) of the delay and compute the $(1-\alpha)$ quantile of it (the local CDV). A gamma distribution has two parameters: a scale parameter (λ) and a shape parameter (r). These parameters are related to the mean and variance of the delay with the following equations:

$$\lambda = \mu/\sigma^2, \qquad r = \mu^2/\sigma^2 \tag{5}$$

The MGF, $\Phi(s)$, of a gamma distribution with parameters λ and r is given as:

$$\Phi(s) = (\lambda/(\lambda - s))^r \tag{6}$$

Assuming the delays are independent, the MGF of the end-to-end delay distribution over N hops, denoted with $\Phi_N(s)$, is the product of the MGFs of the individual delay distributions in the switches:

$$\Phi_N(s) = \prod_{i=1}^{N} (\lambda_i/(\lambda_i - s))^{r_i} \tag{7}$$

The Chernoff bound gives a bound on the probability distribution of a random variable related to its MGF [Pap91]. If X is a random variable with MGF of $\Phi(s)$, then the Chernoff bound states that $P(X \geq x)$ *is less than or equal to* $e^{-sx}\Phi(s)$, where s is greater than zero. Then the probability distribution of end-to-end CTD over N hops (CTD_N) is bounded as follows:

$$P(CTD_N \geq t) \;\leq\; e^{-st}\Phi_N(s), \quad s > 0 \tag{8}$$

From the equations 5, 7 and 8 (by substitution and manipulation), we get the following bound for the end-to-end CDV, denoted with $cdv_{tot}(\alpha)$:

$$cdv_{tot}(\alpha) \leq -\ln(\alpha)/s + 1/s \sum_{i=1}^{N} r_i \ln(\lambda_i/(\lambda_i - s)), \quad where \quad 0 < s < \min_{1\leq i\leq N} \lambda_i \tag{9}$$

In this inequality, the CDV bound changes with the value of s. We can solve for an optimal value of s, say s^*, if we fix the parameters (λ_i, r_i) for each switch, such that $cdv_{tot}(\alpha)$ is minimized. This value of s is in the range $(0, \min_{1\leq i\leq N}\{\lambda_i\})$. However, this is not feasible for implementation, since all the parameters can not be signaled. Therefore, we suggest an alternative approach to find the end-to-end CDV, which is not optimal, but good enough for practical purposes.

The summation term in equation 9 can be accumulated hop-by-hop. But, we also need to set s to a value that is greater than zero, but smaller then the minimum λ_i along the path. Therefore two things need to be signaled:

1) A minimum value of λ_i over the path.

2) An accumulated value of the term inside the summation.

Each switch chooses a value of s to compute the bound and compares it with the requested end-to-end CDV. If the estimate is less than the requested end-to-end CDV, then the term $(r_i \ln(\lambda_i/(\lambda_i - s)))$ is accumulated in the forward pass (with the call setup request). Let us denote the chosen value of s in switch i as s_i. Then the following constraint should be satisfied: $0 < s_i < \min_{0<j<=i}\{\lambda_j\}$.

We choose s_i to be equal to $(\min\{\lambda_i\})/2$, since $\min\{\lambda_i\}$ is already a signaled parameter. This is not exact, because neither is the condition $0 < s_i < \min_{0<j<=i} \lambda_j$ met in the strict sense, nor is our heuristic choice of $s_i = \lambda_i/2$ optimal (s^* is optimal). However, the effect is in the conservative direction, i.e. causing overestimation. The value of $\min\{\lambda_i\}$ is known accurately at the destination (or at the last switch). Therefore, upon connection acceptance, in the backward pass (with the call accept message), the correct value of $\min\{\lambda_i\}$ can be used to recalculate the local contribution of each switch, and the slack can be used to re-adjust the resource allocations.

To re-state our approach, each switch signals the minimum λ_i and the accumulated sum. It computes the estimated peak-to-peak CDV as $-\ln(\alpha)/s$ plus the accumulated sum divided by s. If this exceeds the requested peak-to-peak CDV, the call is rejected. Otherwise the minimum λ_i and the accumulated sum is forwarded to the next hop where the process is repeated. If the call is accepted, each switch in the backward pass has a chance to recompute its local contribution using the correct value of minimum λ_i. As we stated, the direction was conservative in the forward pass. Hence, it is likely that there is some slack available to re-adjust resource allocation in the backward pass and optimize the use of resources.

The pseudo-code of the algorithm that will run in each switch is shown in figure 7. The parameters to the function are the signaled parameters obtained from the previous switch. They will be modified in the function and the new values will be signaled to the next switch.

6 Evaluation of Chernoff Method

We evaluate the performance of our method in terms of how accurately it estimates the actual end-to-end CDV; and we compare its performance with that of the simple method and the asymptotic method.

As explained in section 4, the actual end-to-end CDV is computed by taking the $(1-\alpha)$ quantile of the end-to-end CTD distribution, which is found by the convolution of individual switch queueing delay distributions. The end-to-end CTD distribution can have a closed formula for some individual delay distributions like exponential, Erlang, or gamma with one of the parameters fixed. Therefore, we will take the delay distributions in switches to be Erlang distributions, with one of the parameters fixed (Erlang is a special case of gamma, having integer shape parameters). We fix α to be 10^{-6}. While fixing and assigning values to these parameters, we are using exemplary values, which do not convey any information from the real-world. But this does not harm our purpose, which is to compare the methods and obtain an insight about their relative performance. We are not doing a performance analysis in absolute terms, which would show the delay and delay variance values exactly.

Input:

$CDV_{requested}$: The requested CDV level of the user.
α : The degree of the delay guarantee.

Input/Output:

$AccSum$: Accumulated sum in the upstream nodes.
λ_{min}: Minimum λ in the upstream nodes.

Local Variables:

λ_{local}: The scale (λ) parameter of the local delay distribution (gamma).
r_{local} : The shape (r) parameter of the local delay distribution (gamma).
s_{local} : The local s value in the switch.
CDV_{total} : The total CDV estimate including the upstream nodes and this node.
μ_{delay} : The mean of the cell queueing delay in the switch.
σ_{delay} : The standard deviation of the cell queueing delay in the switch.

Algorithm:

get μ_{delay} and σ_{delay}; (assuming switch is computing them in the background)
$\lambda_{local} \longleftarrow \mu_{delay}/{\sigma_{delay}}^2$;
$r_{local} \longleftarrow {\mu_{delay}}^2/{\sigma_{delay}}^2$;
if λ_{local} is less than λ_{min} **then**
    $\lambda_{min} \longleftarrow \lambda_{local}$;
end if
$s_{local} \longleftarrow \lambda_{min}/2$;
$AccSum \longleftarrow AccSum + (r_{local} \times \ln(\lambda_{local}/(\lambda_{local} - s_{local}))$;
$CDV_{total} \longleftarrow (-1 \times \ln\alpha/s_{local}) + ((1/s_{local}) \times AccSum)$;
if CDV_{total} is less than or equal to $CDV_{requested}$ **then**
    signal $AccSum$ and λ_{min} to the next switch;
else
    reject the connection request;
end if

Figure 7: CDV accumulation method based on Chernoff bound

First, we compare the methods for a tandem of switches, each having Erlang delay distribution with the same scale (λ) parameter, but with different shape (r) parameters. The convolution of Erlang distributions with the same scale parameter is an Erlang distribution with the shape parameter equal to the sum of the shape parameters of convolved Erlang distributions. The shape parameters are assigned randomly to the switches.

Figure 8 shows the actual CDVs and their estimates for varying number of hops. The estimates of both the Chernoff method and the asymptotic method are very close to the actual CDV values, but the simple method estimates are much higher.

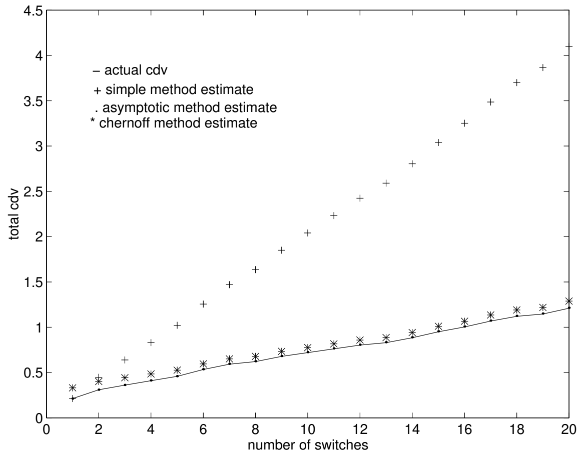


Figure 8: Comparison of estimates and actual end-to-end CDV. Distributions are Erlang with fixed scale parameter and different shape parameters.

For a second case, we fix the shape parameter in the distributions and vary the scale parameter randomly. We fix the shape parameter to 1 (hence distributions are exponential - a special case of Erlang). The convolution of exponential distributions is an hypoexponential distribution, for which we have a closed form.

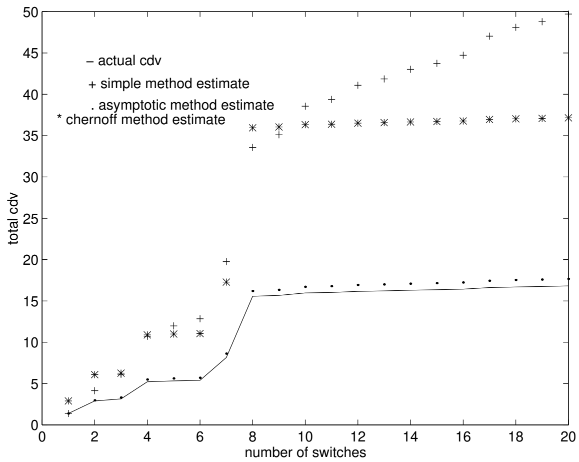


Figure 9: Comparison of estimates and actual end-to-end CDV. Distributions are exponential (special case of Erlang with $r = 1$) with random λ values in successive switches.

Figure 9 shows the comparison of methods. Although in this case, the Chernoff method is not performing as well as asymptotic method, it is much better than the simple method. After a certain point (switch 8 in this case), the difference between simple method estimates and Chernoff method estimates are increasing. This is because switch 8 has the minimum λ value and our algorithm does not find out it before, due to signalling constraints. Before that point, the estimate is computed sub-optimally, without knowing the exact minimum. This shows the sensitivity of our algorithm to how quickly it discovers the minimum λ on the path. The sooner it finds the minimum, the better it performs. However the reverse still performs much better than simple method, as will be shown in figure 11.

Since algorithm performance is dependent on how quickly it encounters the minimum λ, we analyze it for two extreme cases: minimum λ is found in the first switch and minimum λ is found in the last switch. Figure 10 shows the estimates of the method for a case where λ is consistently increasing along a path. In this case, after a few hops, the Chernoff method again estimates consistently better than the simple method.

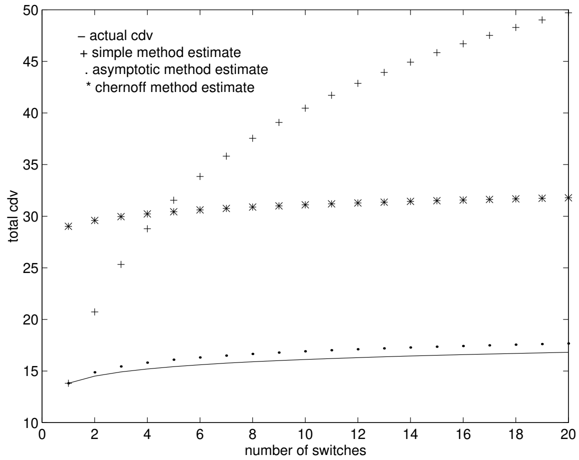


Figure 10: Comparison of estimates and actual end-to-end CDV. Distributions are exponential with increasing λ values in successive switches.

Figure 11 shows the estimates of the methods for the case where λ is consistently decreasing along a connection path. Again, the Chernoff method performs much better than the simple method and close to the asymptotic method's performance.

It is clear that the Chernoff method estimates end-to-end CDV much more accurately than the simple method. The accuracy is also close to the asymptotic method's accuracy in most cases. The advantage of the Chernoff method over the asymptotic method is two fold. 1) It requires fewer parameters to be signaled. The algorithm described requires two parameters, whereas the asymptotic method requires three. It may also be possible to enhance our algorithm so that it would require only one parameter. This can be achieved if the minimum and maximum average delays and delay variances in switches can be constrained, so that we can fix the value of s. 2) It does not require the accurate estimation of the local CDV, which is a costly operation. It only requires the estimation of mean delay and delay variance, which are much easier to measure and estimate. Therefore, we see our method to be feasible and accurate enough to be applicable in ATM networks.

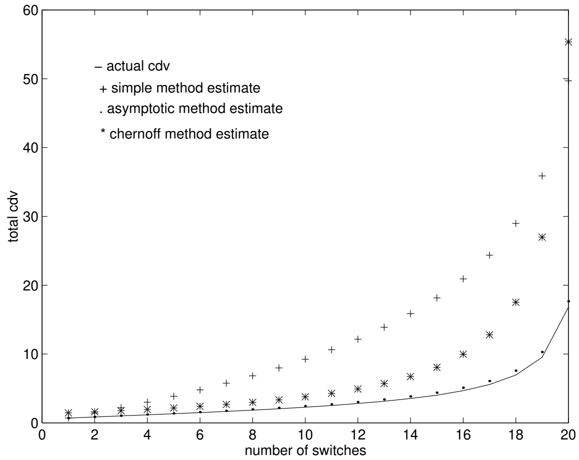


Figure 11: Comparison of estimates and actual end-to-end CDV. Distributions are exponential with decreasing λ values in successive switches.

7 Conclusion

In the first part of the article, we evaluated and compared three CDV accumulation methods which are proposed to ATM Forum. Square-root method is underestimating the end-to-end CDV after some number of hops, hence is not suitable for use in ATM networks, which may have arbitrarily large number of hops on a path. The simple method grossly overestimates the actual total CDV, although computationally it has less overhead. The asymptotic method, which quite accurately estimates total CDV, needs three parameters in the signalling SETUP message, hence is not applicable now. It is also computationally expensive.

We proposed a new method which is based on the Chernoff bound and on the assumption that the delays in switches are gamma distributed. Table 1 summarizes the evaluation of all the four CDV accumulation methods.

From the comparisons made for some sample distributions we found that the Chernoff method performs much better than the simple method and is close to the asymptotic method with much less complexity. Under practical conditions, where switch loads are not extremely unbalanced, the Chernoff method performs very close to the asymptotic method.

The Chernoff method is based on the assumption that the delays are gamma distributed. As shown in [NLG96], this is not an unrealistic assumption. An important advantage of this method is that it allows the estimation of the total CDV using only the mean and variance of the delay in switches without requiring measurement and computation of the $(1-\alpha)$ quantile of CTD, whereas both the simple and asymptotic methods require. This is important because the measurement and computation of mean and variance of delay is much simpler. Measurement and computation of (1-α) quantile of CTD requires on the order of $1/\alpha$ delay samples to be collected to obtain a histogram and to derive the $(1-\alpha)$ quantile thereof. Since α is specified to be too small (between 10^{-4} and 10^{-12}), this requires a large number of delay samples to be collected frequently, which is a costly operation. This motivates the need for a realistic assumption about the delay distributions and use it as a model to derive the local CDV. We used the gamma distribution with appropriate scale and shape parameters to model the delay distribution.

The Chernoff method also does not require as many parameters to be signaled as the asymptotic

CRITERIA	Accumulation Method			
	Simple Method	Square-root Method	Asymptotic Method	Chernoff Method
Correctness	No theory behind. Can give incorrect results for some distributions.	Uses only the variance of the delay to estimate CDV, which is not enough.	Has sound theory based on central limit theorem.	Has sound theory based on Chernoff bound.
Accuracy	Grossly overestimates the actual end-to-end CDV.	Reasonable, but underestimates after some number of hops.	Close to actual end-to-end CDV.	Close to actual end-to-end CDV in most cases.
Signalling overhead	Needs 1 parameter to be signaled.	Needs 1 parameter to be signaled.	Needs 3 parameters to be signaled.	Needs 2 parameters to be signaled.
Complexity	Simple. Needs local CDV measurements.	Simple, but requires square-root operation. Needs local CDV measurements.	Complex. Needs local CDV, mean delay and delay variance measurements .	Reasonable. Needs only mean delay and delay variance measurements.

Table 1: Summary of evaluation of CDV accumulation methods

method. Moreover, the Chernoff method has the advantage of being implementable under current signalling specifications, provided that we can set an appropriate value for s. If we have an upper-bound on the mean and variance of the delay in a switch, an appropriate value of s can be chosen. After having more experimental results about the traffic and delay characteristics in ATM switches, a bound can be derived.

Acknowledgments

We thank Lucent Technologies for supporting this work. We also thank Cynthia Rais for her comments on earlier drafts of this paper.

References

[AKRS94] C. Aras, J. Kurose, D. S. Reeves, and H. Schulzrinne. Real-Time Communication in Packet Switched Networks. In *Proc. of the IEEE*, volume 82, pages 122–139, January 1994.

[BIC94] ATM Forum. *B-ISDN Inter-Carrier Interface (BICI) Specification Version 1.1*, September 1994.

[Bor95] M. Borden. Properties of CDV and Its Accumulation. ATM Forum Contribution, August 1995.

[CPB93] K. C. Claffy, G. C. Pluyzos, and H. W. Braun. Applications of Sampling Methodologies to Network Traffic Characterization. In *Proceeding of SIGCOMM Conference*, 1993.

[HB95] D. Hushes and M. Borden. Effect of CDV Accumulation Method on Utilization. ATM Forum Contribution, November 1995.

[Mat94] The MathWorks Inc. *MATLAB version 4.2c*, 1994.

[NLG96] H. Naser and A. Leon-Garcia. A Simulation Study of Delay and Delay Variation in ATM Networks, Part I: CBR Traffic. In *INFOCOMM Conference Proceedings*, 1996.

[Noo95] M. Noorchasm. On the Subject of CDV Accumulation. ATM Forum Contribution, June 1995.

[Pap91] A. Papoulis. *Probability, Random Variables, and Stochastic Processes.* McGraw-Hill, 3rd edition, 1991.

[PNN96] ATM Forum. *Private Network-Network Interface Specification Version 1.0*, March 1996.

[Rop96] C. Roppel. Estimating Cell Transfer Delay in ATM Networks Using In-Service Monitoring Methods. In *Proceedings of GLOBECOMM*, 1996.

[SSD93] P. Skelly, M. Schwartz, and S. Dixit. A Histogram-Based Model for Video Traffic Behaviour in an ATM Multiplexer. *IEEE/ACM Transactions on Networking*, 1(4), 1993.

[TM96] ATM Forum. *ATM Traffic Management Specification Version 4.0*, April 1996.

[UNI96] ATM Forum. *User Network Interface Version 4.0*, 1996.

[Wri95] S. Wright. CDV Accumulation Between Switching Stages. ATM Forum Contribution, April 1995.