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

Ibrahim Korpeoglu Satish K. Tripathi
Mobile Computing and Multimedia Laboratory
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

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 signaling 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].

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

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

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 signaling 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 signaling 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 signaling, there are also constraints in terms of signaling 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 signaling 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].

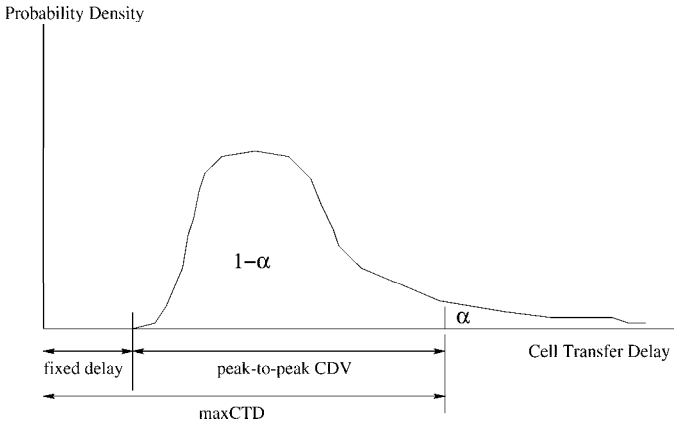

Figure 1: CTD probability density model

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 signaling 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 denote it as $cdv(\alpha)$, when we want to express this dependence.

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. Theoretically, assuming independence of delays in successive switches, 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 due to the signaling constraints, 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 delays

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

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 is characterized only by the queueing delay.

In all our evaluations, we will assume that the delays in successive switches are independent. The delay correlation in successive switches is analyzed in [Y+93], and it is reported that the delay distributions suggest a strong hop-by-hop independence at high link utilizations. It is also reported that at low or moderate link utilizations, we can assume that delays are independent for practical purposes. Hence we will not study the effects of correlation on end-to-end CDV here, but will leave it as a futher study.

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. With the independence assumption of delays, 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 done only for a single α value and only for identical exponential delay distributions. We will extend our comparisons also to Erlang (a special case of gamma) distribution. 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. Our aim in this exercise is to study the effect of different distributions on the relative performance of the methods. We would like to see if the trends are similar in this case too.

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. 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. It is interesting to see that the estimate goes below the actual CDV after certain number of hops. This is undesirable, since underestimating

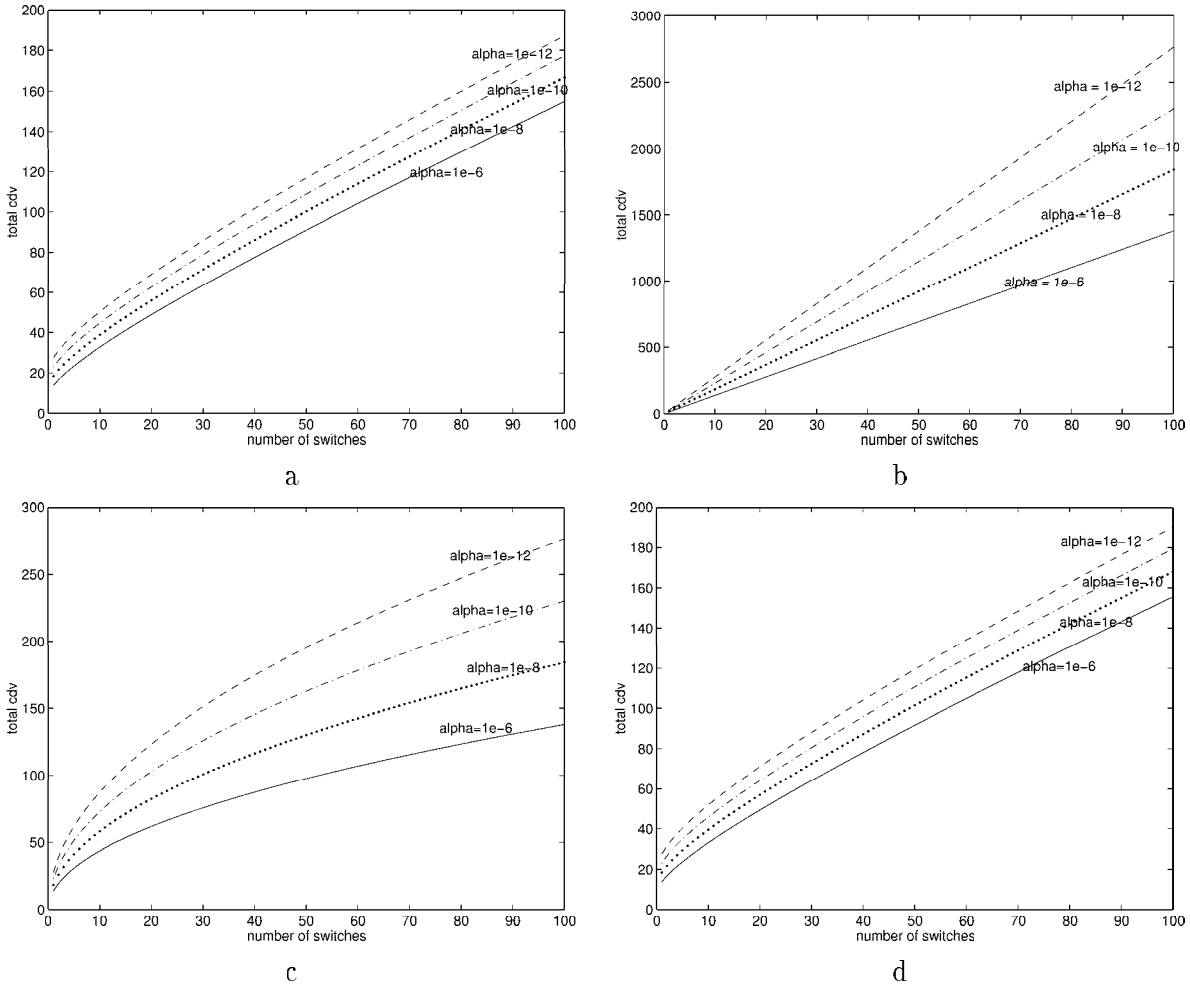

Figure 2: End-to-end CDV versus number of hops. a) Actual end-to-end CDV, b)Simple method estimate, c)Square-root method estimate, d)Asymptotic method estimate.

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.

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 signaling protocols. Both UNI and NNI signaling 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 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 complex and 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, 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. We will also assume that the delays in switches are independent. As we stated earlier, [Y+93] studies end-to-end delay distributions for real time applications in high-speed networks and reports that the delay distributions suggest strong hop-by-hop independence. With independence assumption, 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. 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:

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

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 $cdv_{tot}(\alpha)$:

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