Equivalent Bandwidth Characterization for Real-time CAC in ATM Networks

S. Ramaswamy T. Ono-Tesfaye W. W. Armstrong

P. Gburzynski[1]

Department of Computing Science
615 GSB, University of Alberta, Edmonton, AB, Canada T6G 2H1
tel: (403) 492-2347 (office), (403) 492-1071 (fax)
e-mail: pawel@cs.ualberta.ca URL: http://www.cs.ualberta.ca/~pawel

Keywords: neural networks, regression analysis, ATM networks, call admission, bandwidth characterization

Abstract

Many call admission control schemes for ATM-type networks are based on the concept of *effective bandwidth* (EB). Most such schemes focus on the cell loss rate as the exclusive QoS metric and therefore base their EB on cell-loss rate approximations. We use simulation data to train an adaptive logic network (ALN) to estimate cell loss *and* delay; these estimates can then be used to compute effective bandwidths to satisfy both cell loss and delay. The simulation data is also used to develop regression models for cell loss and delay. We compare the accuracy of the ALN and regression models with that of the well-known fluid-flow model by Anick et al. Results indicate that the ALN and regression models are computationally simple and sufficiently accurate for practical use.

1 Introduction

High-speed ATM-based integrated networks are expected to carry connections with a wide range of characteristics. The call admission control of such networks tries to avoid congestion by limiting the number of connections that are admitted into the network. Many call admission schemes are based on the concept of *effective bandwidth* or equivalent capacity: the effective bandwidth of a traffic source is defined as the bandwidth required to guarantee some specific quality of service (QoS). In such a scheme, a call is admitted by a switch if the call's effective bandwidth is less than or equal to the bandwidth still available on the outgoing link. The effective bandwidth of a source lies between its average and peak cell rates; and the effective bandwidth of N aggregated sources is generally less than N times the effective bandwidth of a single source (this phenomenon is known as the *statistical multiplexing gain*).

[1]Author to whom correspondence should be addressed

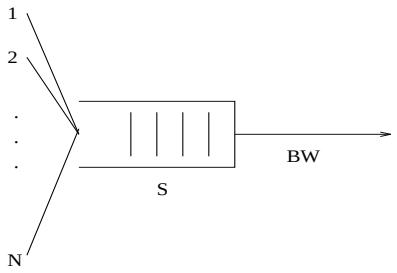

Figure 1: System Model

Most of the effective bandwidth research has focussed on the cell loss rate, CLR, as the exclusive QoS metric. Consider the system shown in figure 1. If the CLR for N sources can be determined as a function of the buffer size S, the link capacity c and possibly some other system parameters,

$$CLR = f(S, c, \ldots) \tag{1}$$

then the effective bandwidth for N sources can be determined by either explicitly inverting the function or by iteration. In most cases, unfortunately, a function like (1) is not explicitly available, and approximations must be used. Anick, Mitra and Sondhi [1] present a well-known *fluid-flow* model (hereafter referred to as the AMS model) for the case of N identical On/Off sources and an infinite buffer. Because of the infinite buffer assumption, the cell loss rate for a finite buffer of size S is approximated by the probability $P(x > S)$ that the occupancy of the infinite buffer exceeds S. Guerin et al. [5] propose a simplified, computationally tractable version of the AMS model. A simple *binomial* approach is described in [8]: the authors' proposal is to approximate the cell loss rate by the probability that the combined cell rate of all N sources exceeds the link capacity (in effect, this assumes zero cell buffers). [10, 13] contain comparisons of these and other methods of obtaining equivalent bandwidths. The limitations of these methods are that

1. They require the QoS requirement to be expressed in terms of the buffer overflow probability, rather than the cell loss probability. For low to moderate buffer sizes, the difference can be significant.

2. They consider only the cell loss rate as a performance metric. In order to limit cell delay as well as cell loss, however, it is necessary to base EB computations not only on estimates of the cell loss rate, but also on estimates of cell delay.

3. They are reasonably accurate only for very low loss probabilities ($< 10^{-6}$). It may not be possible or necessary to meet this requirement, for example, voice traffic is not very demanding in its loss probability requirements (about 10^{-3} according to [9]).

4. The sources studied must be analytically tractable. The traffic model most frequently used is the On/Off bursty source. This approach requires more complex sources to be modeled as a combination of On/Off sources, which may not always be easy to do.

It is important that EB computation schemes capture the non-linear relationship between the total EB and the individual EBs, especially when the statistical multiplexing gain is high. Both [10] and [5] propose admission policies that use a non-linear function of the individual EB to expand the region of application of EB source characterization. The policy proposed by [10] eliminates limitations 1 and 2 above, but still suffers from limitation 3. It also adds complexity by requiring that each call be empirically converted into an equivalent On/Off source feeding a zero-buffer system. The formula developed in [5] is simple; however, it overestimates the effective bandwidth of a source considerably, leading to inefficient link utilization.

In this paper, we consider using regression analysis and an adaptive logic network, ALN, (section 4) approach to the problem of estimating cell loss and delay in the system of fig. 1. These estimates are then used to compute a call's EB. Our goal is to use methods whose complexity is independent of the number of calls in progress. Since calls with widely differing peak rates are unlikely to have the same QoS requirements, equivalent bandwidth schemes are likely to be applied to calls with similar peak rates. Accordingly, we limit ourselves to calls with homogeneous traffic (homogeneous QoS requirements are common to all of the EB schemes described above).

We use simulations to study the factors affecting cell delay and loss. Previously, simulation studies have been aimed at obtaining sets of curves to be used as guidelines in predicting the equivalent capacity of connections. Different sets of curves are stored for calls with widely differing peak rates, even though their other characteristics may be identical (peak to average rate, for example). In this paper, we use the simulation results rather differently. In the first approach, we use multivariate non-linear regression to obtain delay and cell loss as a function of the source characteristics, buffer size, service rate and the number of calls. This equation can be iterated through to obtain the service rate—for a given delay or loss requirement—as a function of the other parameters. All the time-related parameters used in the regression are normalized with respect to the peak rate. Hence, the same regression equation can be used for calls with peak rates very different from the ones used in the simulation, as long as the other characteristics remain the same. Also, because the number of calls is a parameter to the delay and loss models, the computation time for the equivalent bandwidth of a new call is independent of the number of admitted calls. This approach eliminates the problem [14] of the storage space at switches for the bandwidth requirement tables.

In the second approach, we train an ALN to predict the delay or cell loss. The ALN can then be inverted to predict the equivalent bandwidth instead. As before, this trained ALN can be used to predict the EB for calls with peak rates very different from the ones used to collect the simulation data.

Both approaches are shown to predict EB fairly accurately for a wide range of buffer size (1–250 times the burst length), cell loss (0.1–0.0001) and delay (1–25000 cells). This is the non-linear region of interest since analytical methods ([1], [5]) are inaccurate at high cell loss.

We compare the results obtained by the regression and ALN models with that obtained

by the AMS model. Because the AMS model only predicts the CLR (or rather, the overflow probability), we have extended the model to predict the delay as well, and we refer to this as the D-AMS model.

The rest of the paper is organized as follows. In section 3, we describe the simulation experiments that are the basis for the regression and ALN approaches. We define the traffic source model in section 3.1. In sections 4, 5 and 6, we describe the ALN, regression and AMS approaches to delay and cell loss estimation, respectively. Section 7 contains a comparison of the three models with the (exact) simulation results. In section 8, we apply the methods to the computation of effective bandwidths. Our findings are summarized in the concluding section 9.

2 Notation

The following symbols are used throughout the paper:

Clr It represents the cell loss probability at the switch.

Del Represents the queuing delay (excluding processing delay) at the switch.

L_del The logarithm (base 10) of cell delay.

L_clr The logarithm (base 10) of cell loss probability.

t_{ON} The mean duration of the *On* state of an On/Off source.

D-ALN The ALN trained to predict cell delay at the switch.

C-ALN The ALN trained to predict the cell loss at the switch.

D-AMS The AMS model extended to predict cell delay at the switch.

C-AMS The AMS model that predicts the cell loss at the switch.

D-REG The regression model for cell delay at the switch.

C-REG The regression model for cell loss at the switch.

t_{OFF} The mean duration of the *Off* state of an On/Off source.

Bl The mean number of cells generated by the On/Off source when in the *On* state.

Pcr The peak cell rate of a single call.

Scr The average cell rate of a single call.

S The number of cell buffers at the single-port switch.

L_s The logarithm (base 10) of the number of cell buffers, S.

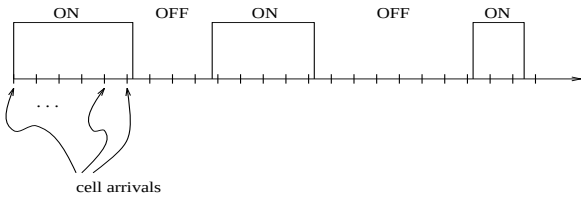

Figure 2: Source Model

N The number of calls being multiplexed at the switch

Bw The cell service rate at the switch.

L We assume that the link rate is 150 Mbps. This is an upper bound on the service rate Bw of the switch.

EB The effective bandwidth per call; it is the service rate Bw required to achieve a given QoS normalized w.r.to Pcr and N.

S/Bl The ratio of buffer size S to the mean burst length Bl. The delay and cell loss probability are assumed to depend on this ratio rather than on S and Bl individually. This has been verified by simulations for a few cases.

L_sbl The logarithm (base 10) of S/Bl.

Av/Pk The ratio of Scr to Pcr.

ALN Adaptive Logic Network—a variant of an artificial neural network (see section 4).

3 Experimental setup

3.1 Models

Each traffic source has an *On* and an *Off* state. When in the *On* state, the source generates cells at a deterministic rate of Pcr cells/s; no cells are generated during the *Off* period. The *On* and *Off* periods are exponentially distributed with means t_{ON} and t_{OFF}, respectively (figure 2). The N sources are independent, but have identical mean *On* and *Off* periods. The mean burst length, Bl, is given by $Pcr \times t_{ON}$, and the average-to-peak-rate ratio, Av/Pk, is given by $\frac{t_{ON}}{t_{ON}+t_{OFF}}$.

3.2 Factors affecting delay and cell loss

Assume that the traffic from a collection of N identical calls is fed into a finite-sized queue and serviced by a single server (figure 1). Then the average delay Del and cell loss Clr are influenced by

- The characteristics of a single call: mean burst length (Bl), average-to-peak-rate ratio (Av/Pk). The peak rate is assumed to be at least 10 times smaller than the link rate[2] and thus only affects the scale of operation without changing the relative effects of the other parameters.

- System parameters: Buffer size S, service rate Bw. As mentioned in section 2, the delay and loss depend on S and Bl only through the ratio S/Bl. However, since our experiments (section 3.2) have been performed for only one value of Bl, we can use S and S/Bl interchangeably. EB is the service rate Bw normalized w.r.t. the peak rate of a call and the number of calls. Its value is between Av/Pk and 1.

- Number of calls, N, being multiplexed at the queue.

The reason that we choose Bl and Av/Pk as the parameters representing the characteristics of a single call—rather than using t_{ON} and t_{OFF} directly—is the following. Suppose that we obtain via a simulation study a formula for EB in terms of Clr, t_{ON}, t_{OFF}, S and N. If we scale the peak rate used in the study by a factor of 10 to find the service rate for another call type with the same characteristics, we will have to scale t_{ON} and t_{OFF} as well. Our formula will no longer apply since it was obtained using a call with different t_{ON} and t_{OFF}. On the other hand, if we choose Bl and Av/Pk as the independent variables, these will remain the same after the peak rate is scaled. Hence we can use the formula exactly as it is.

Of the five factors mentioned previously, we chose three factors for a full-factorial study—S, EB, N. The traffic generated by a single source can be viewed as a video session, with $t_{ON} = 25$ms (representing the mean t_{ON} duration in the video model used by [6]) , $t_{OFF} = 35$ ms and $Pcr = 14150$ cells/s. This results in $Bl = 353$ and $Av/Pk = 0.417$.

3.3 Training set

The levels chosen for the three factors mentioned before are shown in table 1. When the number of calls is large, linear approximations are expected to be accurate (figure 3.3 left) and hence we do not consider more than 25 calls.

The same observation applies when the buffer size is much larger than Bl (figure 3.3 right) and so we only consider a maximum buffer size of 100000. Since Bl is 353 cells, the maximum value of S/Bl is 283. Since we are mainly interested in predicting EB, and its value is between Av/Pk and 1, we used a fine granularity of 0.02. We found that when EB was close to its minimum value of 0.417 (corresponding to Av/Pk), the steady-state values of Clr and Del were quite difficult to obtain accurately, even with very long simulations. Therefore, we opted to use a minimum EB of 0.44.

[2] In [3], it is mentioned that sources with $PCR/L > 10$ will be best served by allocating peak bandwidth; furthermore, the effective bandwidth for such sources is dependent on Bl. By choosing sources that meet this requirement, we can eliminate the necessity for considering Bl in order to obtain EB

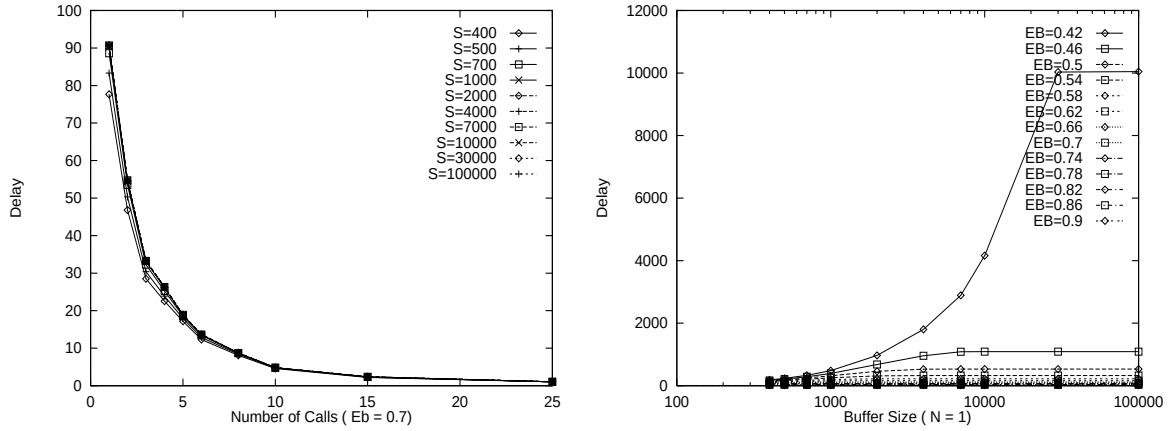

Figure 3: Delay v/s N (left) Delay v/s S (right)

A total of $10 \times 10 \times 24 = 2400$ experiments were run using SMURPH [4], a simulation tool developed at the University of Alberta. This tool allowed us to distribute the experiments over the machines on the local departmental network, utilizing idle cycles. Thus, at the end of 12 hours, the results of all the experiments were available. A few dry runs had been made previously, indicating the length of the runs needed for steady state as the overhead of replicating the experiments with different seeds was deemed to be unacceptable. Approximately 5–25 million cells were simulated in each experiment. The inter-call time was exponentially distributed with mean value equal to t_{ON}, in order to prevent bursts from arriving at the same time at the beginning of the simulation.

3.4 Test set

The levels chosen for the three factors above are shown in table 3.4. The values of *Bl* and *Av/Pk* were the same as in the training set. However, the peak cell rate (*Pcr*) was chosen to be 1000 cells/sec. This was done so that it can be verified that the regression model and the ALN could correctly predict delays when the peak rate was different from that used in the training set.

Figure 4 (left) shows the variation of *Del* with EB for varying N and $S = 1000$ while figure 4 (right) shows *Clr* as a function of EB. The qualitative information in these figures

Factor	*Levels*	*Number of levels*
Number of calls	1–5, 6, 8, 10, 15, 25	10
Buffer size	400, 500, 700, 1000, 2000, 4000, 7000, 10000, 30000, 100000	10
Service rate	0.44–0.9 in steps of 0.02	24

Table 1: Levels of factors for training set

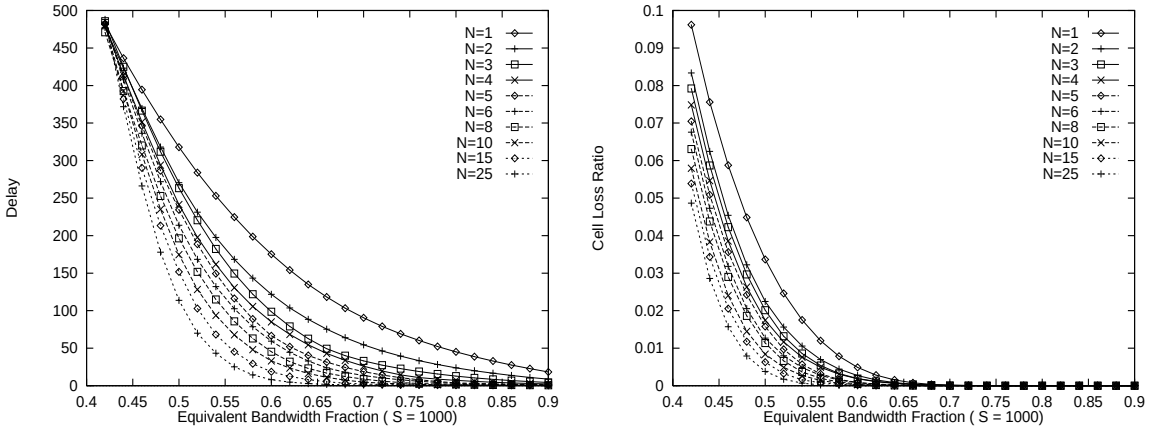

Figure 4: Delay v/s EB (left) CLR v/s EB (right)

is used in deriving the regression and ALN models for delay.

4 Adaptive Logic Networks (ALNs)

An Adaptive Logic Network (ALN) is a type of artificial neural network [2] that maps vectors of real values in Euclidean n-dimensional space to boolean values. The first layer of computing units in the network consists of linear threshold units (perceptrons) that output 1 if an inequality of the form $w_0 + w_1 \times x_1 + w_2 \times x_2 + \ldots + w_n \times x_n >= 0$ is satisfied, otherwise they output 0 . The coefficients w_i of the expression are called the weights of the unit. The boolean outputs of the first layer of units are combined by a tree expression of AND and OR operators of arbitrary fan-in to produce the output of the ALN. An ALN can be viewed as a function $\Re^n \rightarrow \Re$ that produces a 1 as output on or under the graph of the function when given the $n-1$ inputs $x_1, \ldots, x_{n-1}$ and the output x_n that is to be learnt.

A functional computation can be derived from the ALN by taking combinations of linear functions where the combining operators are maxima and minima of functions. If x_n is the output variable of the ALN, then weights of the first layer of units are normalized to have $w_n = -1$ and the inequality of a unit is turned into a function of the form $x_n = w_0 + w_1 \times x_1 + w_2 \times x_2 + \ldots + w_{n-1} \times x_{n-1}$.

Factor	*Levels*	*Number of levels*
Number of calls	1, 2, 3, 7, 9, 12, 20, 30	8
Buffer size	360, 600, 900, 3000, 5000, 8000, 20000, 50000	8
Service rate	0.47, 0.53, 0.57, 0.63, 0.67, 0.73, 0.77, 0.83, 0.87, 0.95	10

Table 2: Levels of factors for test set

The tree of maximum and minimum operators has the same form as the tree of OR and AND operators respectively. The linear functions have weights which are adapted based on training data consisting of vectors $x_1, \ldots, x_n$ that represent the function graph. The algorithm is like least squares fitting of linear pieces to data points, where a linear piece is only active for a subset of the training points. Given $x_1, \ldots, x_{n-1}$, a subtree of a node contains the active linear piece if its value (computed using the maximum and minimum operators according to the subtree) is less (or, respectively, greater) than the value of any other subtree for an AND- (or, respectively, OR-) node.

For ATM traffic characterization, there are several advantages of ALNs over other types of neural networks:

- The normalized weights of linear pieces that are active are the values of the partial derivatives of the output variable with respect to the input variables. Hence they can be directly controlled. For example, forcing a weight to be positive forces the output of the learned function to be strongly monotonic increasing in the corresponding variable.
- If an ALN represents a function $x_n = \mathrm{f}(x_1, \ldots, x_{n-1})$ which is monotonic increasing in some variable x_i, then a functional computation can be derived from that ALN which computes the corresponding function inverse : $x_i = g(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$. This uses the coefficients of the same linear pieces, combined in a different way.
- The ALN does not require the predictor variables to be scaled or normalized. This speeds up the model development process as well the use of the model for prediction.

Note that most performance functions are naturally monotonic. Forcing the learned function to be monotonic or convex makes it difficult for the function learned by an ALN to be influenced by the noise in training points; hence overtraining which prevents good generalization in other neural networks may be avoided in some cases. The use of these advantageous properties is demonstrated in sections 4.2 and 4.3.

4.1 Choosing the epsilon values

Atree 3.0 [2] allows the user to specify a parameter epsilon associated with each variable that expresses the half-length of an interval which has to be covered by each point in a training set. For example if a real value is sampled every 0.8 units, then epsilon might be set to 0.4, half the distance between neighbouring points.

If a function is defined on a domain which is a box of dimension n with sides of lengths $s_1, \ldots, s_n$ (the ranges of the input variables) and there are p points in the training set, then the volume of the space assigned to each data point is $V = (s_1 \times s_2 \times \ldots s_n)/p$, and this volume can be represented by a box whose sides are of length proportional to the ranges of the variables. The half-length of the side in direction i is thus $0.5 \times s_i \times p^{-1/n}$.

In the case of the effective bandwidth problem, we have three input variables: EB, L_s and N. The calculated values for each variable were used as a starting point to obtain a set

of epsilons that permitted a good rate of error reduction on the output variable, in this case, Del. Table 3 compares the calculated values with the values determined by experimentation.

Predictor Variable	Range	Calculated Value	Experimental Value
L_s	5	0.182	0.1
EB	0.6	0.022	0.02
N	24	0.9	1.5

Table 3: Starting epsilons for D-ALN

The same steps were followed when the ALN was trained on Clr. Table 4 shows the calculated and experimental values for the C-ALN. For both the D-ALN and the C-ALN, the calculated and experimental values are reasonably close to each other. This shows that the heuristic chosen to find the starting values of epsilon is useful.

Predictor Variable	Range	Calculated Value	Experimental Value
L_s	2.0	0.11	0.1
EB	0.6	0.035	0.1
N	24	1.35	1.5

Table 4: Starting epsilons for C-ALN

4.2 ALN model for cell delay

The inputs to the D-ALN are L_sbl, EB, N and L_del. Because of the large ranges of S/Bl and Del, we chose to train the D-ALN on L_sbl and L_del, respectively.

Since the ALN is scale invariant, we did not have to normalize the inputs as is usually done with other neural networks. The training was done in stages, progressively lowering the learning rate until the mean residual error fell to 0.006. This implies a mean relative error of $10^{0.006}$ or 1.5% in predicting Del in the training set. Overtraining was prevented by constraining the slope of Del w.r.t. the other variables; in this case, delay was constrained to decrease as EB or N increases and to increase as the L_sbl increases.

Figure 5 compares the estimates of the D-ALN on the training and test sets with simulation results. Since the test set was generated for $Pcr = 1000$ cells/sec while the training set used $Pcr = 14150$ cells/sec, the figure shows that the trained ALN can be used to predict delays for other values of Pcr fairly accurately as long as Bl and Av/Pk remain the same

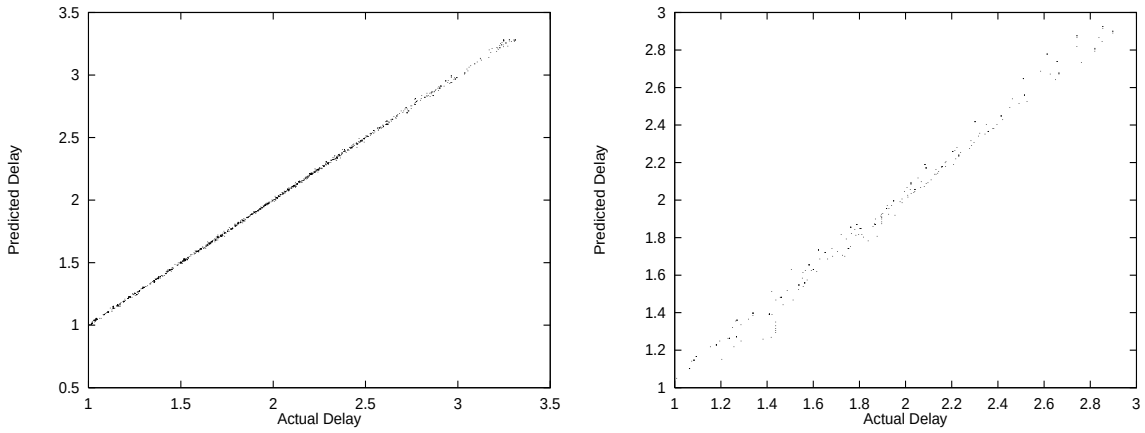

Figure 5: D-ALN model predictions: training set (left) test set (right)

(see section 3). Section 7.1 presents a numerical comparison of the results with the D-AMS and D-REG models.

4.3 ALN model for cell loss

The inputs to the C-ALN are the same as in the case of the D-ALN except that the training is now done on L_clr instead. Since Clr is zero for several points in the training set, these points must be eliminated from the training set. This results in the C-ALN being trained on just 708 data points, still a fairly large number considering that we have only three input variables.

As in the case of the D-ALN, the training was done in stages, progressively lowering the learning rate until the mean residual error on L_clr fell to 0.06. This implies a mean relative error of about $10^{0.06}$ or 15% in predicting the Clr in the training set. Overtraining was prevented by specifying that the Clr decreases as EB, N, or L_sbl increases.

Figure 6 compares the predictions of the C-ALN model on the training and test sets with simulation results. As in the case of Del, the results obtained by evaluating the trained model on the test set show that the ALN can be used to predict Clr for different values of Pcr. Section 7 presents a numerical comparison of the Clr predictions by the C-AMS, C-REG and C-ALN models.

The performance of the C-ALN model on the test set is not as good as the D-ALN model. This is primarily because of the smaller training set due to the missing elements corresponding to zero Clr. The values of $Clr = 0$ in the data sets could be replaced by more accurate values from longer simulation runs in order to obtain better accuracy at high service rates and large buffers. Since this was not done, we see a larger scatter at low loss ($< 10^{-4}$) in figure 6.

Figure 7 (left) compares D-ALN model predictions for $N = 30$ with simulation values. Figure 7 (right) shows the prediction quality of the C-ALN model for $N = 30$, a value not in

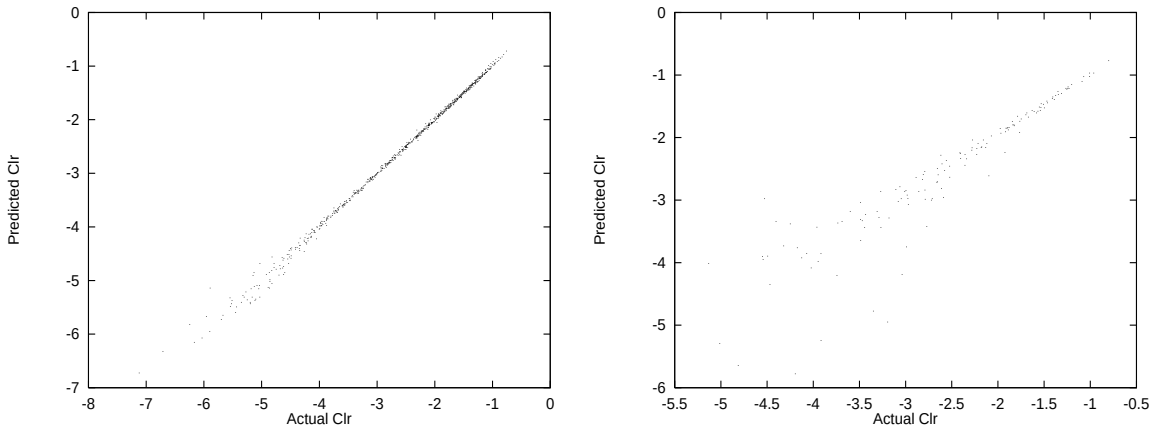

Figure 6: C-ALN model predictions: training set (left) training set (right)

the training set. It can be seen that the extrapolation is fairly accurate.

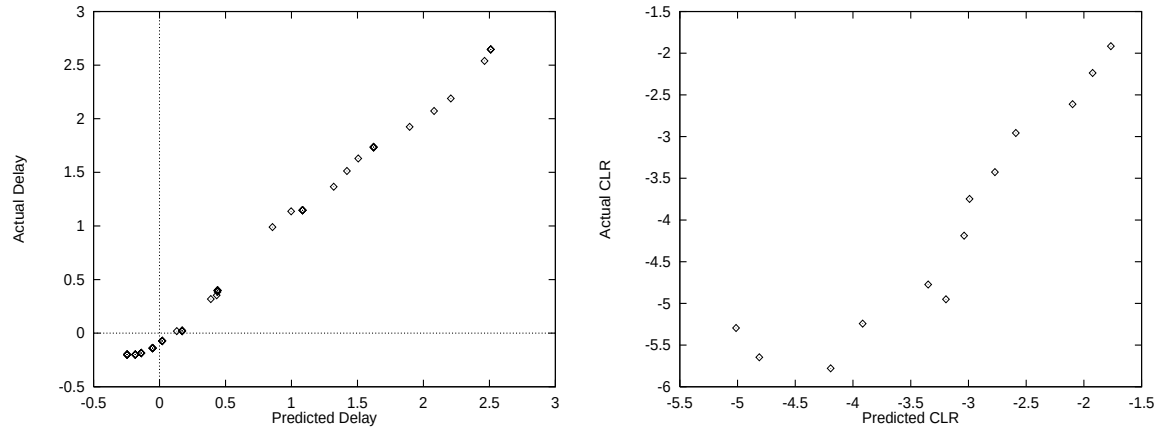

Figure 7: Extrapolation for N=30: D-ALN model (left) C-ALN model (right)

4.4 Residual analysis for ALN models

Since the ALN uses the ordinary least squares (OLS) principle to determine the orientation of the linear pieces, the trained ALN is subject to the assumptions inherent in the use of OLS.

The assumptions made in the case of OLS are [11]

- Predictor variables are nonstochastic and measured without error. Since the predictor variables are controlled inputs to the simulation, this statement is true.
- Model error terms follow a normal probability distribution. This can be seen to be approximately true from the histogram plots of L_del and L_clr residuals (figure 8).

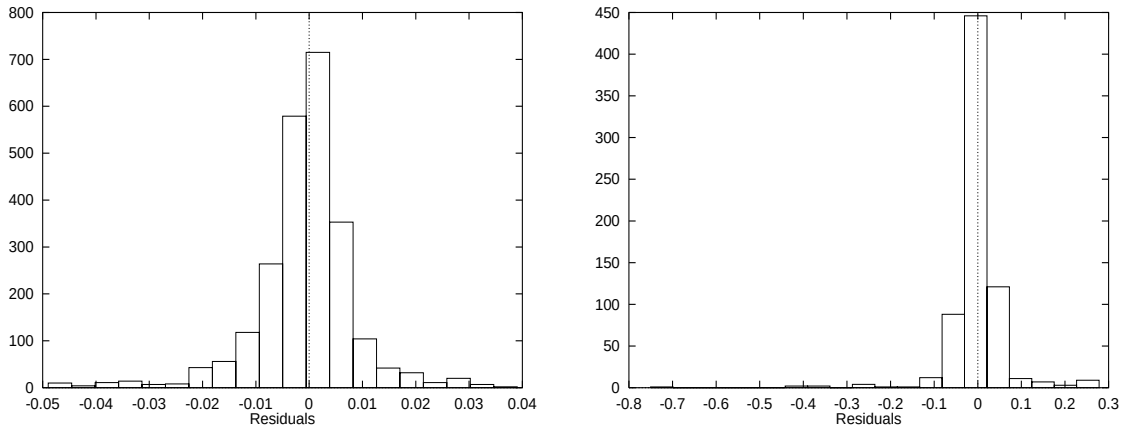

Figure 8: D-ALN histogram of residuals (left) C-ALN histogram of residuals (right)

- Any two errors are independent of each other. The presence of correlation reduces the reliability of the model. To verify this, we plot the residuals against the values obtained by simulation (figure 9).
- Model error terms have zero means, are uncorrelated, and have constant variances. The first of these can be seen to be true by observing that the standard deviation limits appear equidistant from the horizontal axis (figure 9). The second condition is generally true for databases compiled from controlled laboratory experiments. The third assumption is discussed below.

From the L_del-residuals plot, we can see that the errors are randomly distributed on either side of the horizontal axis indicating that there is no systematic error. Since the log function is nonlinear for values of the abcissa < 10, we have eliminated these values from the plot. In any case, we are more interested in delay predictions at higher delays. The residuals are seen to increase in magnitude as L_del increases beyond 2.5, leading us to conclude that there is a small amount of heteroscedasity (unequal variances).

The L_clr-residuals plot shows that the residuals are fairly random for $Clr < 10^{-4}$. For lower CLR, the simulation results themselves are inaccurate. The increased variance of residuals for low CLR is hence not of consequence and the larger values of the residuals can be tolerated.

5 Regression Analysis

Regression analysis is considered indispensable as a data analysis technique in a variety of disciplines. So far, however, the models developed for bandwidth characterization in ATM networks have been analytical models developed from basic principles. Given the variety of traffic expected on ATM networks as well as the complexity of this traffic, it will not always

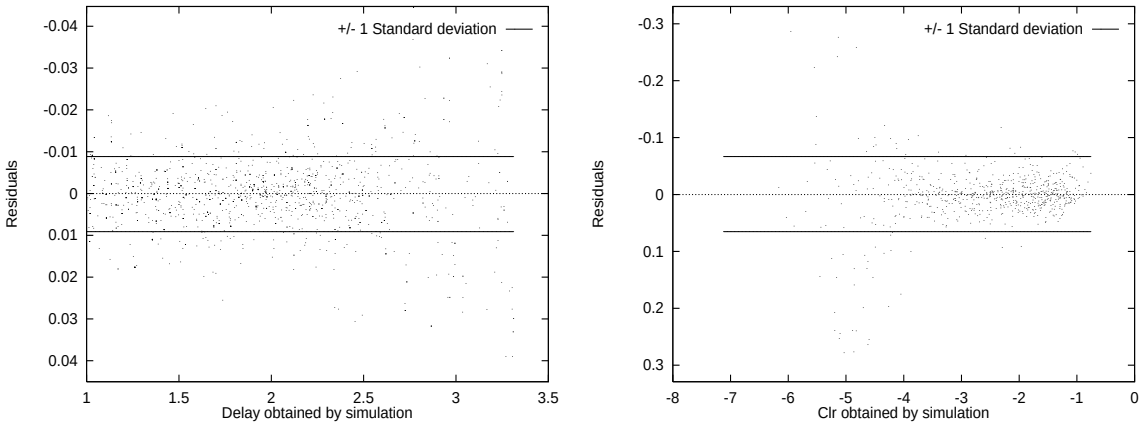

Figure 9: ALN L_del-residuals plot (left) ALN L_clr-residuals plot (right)

be possible to develop such models. Either the development of an analytical model may be time-consuming or the assumptions needed to make the problem tractable may be too limiting.

Moreover, even if a model is obtained by analytical means, it is often too complex to be used for real-time purposes. We feel that a regression model based on simulation data can be developed in a relatively shorter time and yield a simple relationship between the response and predictor variables. This simple relation can be utilised for prediction of the response variable for new values of the predictor variables. In our case, therefore, we can predict Del and Clr for values of $EB, S/Bl$ and N not contained in the simulation data. In addition, we can iterate through the relationships for Del and Clr to find the EB necessary to satisfy a given QoS. Because of the simple relationship between the response and predictor variables, the EB can be found rapidly, making it suitable for real-time call admission control (CAC).

The data upon which the regression analysis is based must be as representative as possible to ensure that inferences or predictions made are accurate. A common misuse of regression methodology is extrapolation—attempting to predict the response variable for values of predictor variables not represented in the database. To reduce the temptation for this, we have tried to make our database as large as is necessary to ensure that most meaningful queries can be answered by the model.

The domains of the predictor variables are as follows

L_s Since S can range from 400 to 100000 cells, L_s can range from 2.6 to 5. This is likely to meet the requirements of real switches.

EB It ranges from 0.44 to about 0.9 of the peak rate. If the service rate requirement of the call is close to 1, it is generally agreed that a peak rate allocation should be made. Also, since the model assumes that the average cell rate per call is 0.417 (section 3.2), a service rate allocation below 0.44 is unlikely.

N The number of calls ranges between 1 and 25. When the number of calls is large,

it is expected that the law of large numbers will apply and there will therefore not be a large change in Clr or Del as N increases further.

5.1 Regression model for delay

In developing the regression model, it is necessary to ensure that the model does not fit the noise in the training data as far as possible. This will enable the model to generalize well to the test set. The coefficient of determination (R^2) is not a good indicator of the goodness of the model as it can be artificially inflated by adding parameters to the model and made arbitrarily close to 1.

Therefore, we need to balance the lower residuals with the cost of adding extra parameters. The CP [7] criterion is widely used for this purpose. It is defined as

$$CP = \frac{sse}{mse\_p_{max}} - (n - 2p) \tag{2}$$

where

- sse is the sum of squares of residuals with p parameters
- mse_p_{max} is the mean sum of squares of residuals for a model that includes as many parameters $p = p_{max}$ as possible; in a sense, the best model possible from the point of view of the smallest residuals. The mean sum of squares of residuals is computed as $mse\_p_{max} = sse\_p_{max}/(n - p_{max})$, where $sse = sse\_p_{max}$ for the model with p_{max} parameters.
- n is the number of training data points.

In order to select the "best" model, we look for models whose CP value is reasonably close to the number of parameters p in the model. The model with the smallest number of parameters is then chosen from this subset of models. Models with a considerable lack of fit will usually have CP values much greater than p. It is possible that a model with fewer parameters may have a smaller CP, but that the larger CP of a model with more parameters is closer to its p (indicating lower bias). In such cases, the choice of a model is largely a matter of judgement.

We use the well-known statistical package SPSS [12]. Because of the large range of Del, we choose L_del as the response variable. Based on the simulation plots, the following observations can be made:

- L_del decreases almost exponentially as a function of EB (figure 4 (left)).
- The variation of L_del as a function of S/Bl has the form

$$a_\infty(EB) \times (1 - e^{-b(EB) \times S/Bl})$$

where $a_\infty(EB)$ is the value of L_del when S/Bl is very large and is only a function of EB (figure 3.3 right).

- L_del decreases exponentially as a function of N (figure 3.3 (left)). The form of the function is

$$c(1) \times e^{-d(EB)\times(N-1)}$$

where coefficient $c(1)$ is the value of L_del when $N = 1$ and is a function of S/Bl and EB.

We do not include S/Bl in the function for N or vice-versa because of an analysis which determined that there is very little interaction between N and S/Bl. We select $N = 1$ as the origin for the number of calls and therefore use $N - 1$ in place of N in the model. The composite model is therefore of the form

$$L\_del = (\alpha \times f(EB)) \times (1 - e^{-\beta(EB)\times S/Bl}) \times (e^{-\gamma(EB)\times(N-1)}) \quad (3)$$

The D-REG model is obtained in three stages. In the first stage, we set $N = 1$ and buffers $S = 100000$ (to ensure a large value of S/Bl). The second and third terms in equation 3 reduce to 1 and so we can obtain a model for L_del in terms of $\alpha \times f(EB)$.

Table 5 shows the CP criterion for the different models tried for $a \times f(EB)$, the first term in equation 3. The model with seven parameters has the smallest residuals and hence it is selected as the baseline for comparing the other models. From equation 2, the CP for this model will be the closest to $p = p_{max} = 7$ because the first term in the equation reduces to $n - p_{max}$. It can be seen that all the models except the last exhibit considerable bias; that is, they lie far away from the CP $= p$ line. Hence we choose the last model for $a \times f(EB)$.

Model	*Number of Parameters (p)*	*CP*
$a \times e^{(-b\times EB+c)}$	3	769.711
$a \times e^{(-b\times EB+c)} + d$	4	718.047
$a \times EB \times e^{(-b\times EB+c)} + d$	4	916.412
$a \times e^{(-b\times EB+c)} \times (d - e \times EB)$	5	771.794
$a \times e^{(-b\times EB+c)} \times (d - e \times EB) + f$	6	722.047
$a \times e^{(-b\times EB+c)} \times (d \times bw - e \times EB^2) + f$	6	796.793
$a \times e^{(-b\times EB+c)} \times (d - e \times EB^2) + f$	6	656.884
$a \times e^{(-b\times EB+c)} \times (d - e \times EB + f \times EB^2)$	6	26.3314
$a \times e^{(-b\times EB+c)} \times (d - e \times EB + f \times EB^2) + g$	7	6.99999

Table 5: CP criterion for D-REG model (stage 1)

In the second stage, we choose $N = 1$ as before, but place no restriction on S. We have obtained the first term in equation 3 and the third term reduces to 1. So we can model

$1 - e^{-\beta(EB)\times S/Bl}$. Table 6 shows the the CP criterion for the different models tried in the second stage. Each of the models in this stage is multiplied by the model chosen in the first stage to obtain the predicted values of *L_del*. We see that only the models with CP = 3.45183 (3 parameters) and CP = 4.99988 (5 parameters) are adequate. The former has lower CP, but the CP of the latter model is closer to its p, indicating smaller bias. We pick the smaller model in this case, opting for simplicity rather than additional accuracy. Our choice is supported by the fact that the values of g and h are close to 1. This indicates that while the four parameter model tracks the simulation data (and noise) better, the model with three parameters is more representative of the real system.

Model	*Number of Parameters (p)*	*CP*
$(1 - e^{(-i\times S/Bl)}$	1	27636.8
$(1 - e^{(-(i-j\times EB)\times S/Bl)}$	2	4732.42
$(g - h \times e^{(-i\times S/Bl)}$	3	24014.5
$(1 - e^{(-(i-j\times EB)\times S/Bl-k\times EB)}$	3	3.45183
$(g - h \times e^{(-i\times S/Bl-j\times EB)}$	4	744.314
$(g - h \times e^{(-(i-j\times EB)\times S/Bl)}$	4	71.6544
$(g - h \times e^{(-(i-j\times EB)\times S/Bl-j\times EB)}$	5	4.99988

Table 6: CP criterion for D-REG model (stage 2)

In the third stage, we place no restriction on N. Table 7 shows the the CP criterion for the different models tried for $e^{-\gamma(EB)\times(N-1)}$, the last product term in equation 3. Interestingly enough, although the single parameter model using $ln(N)$ has a lower *CP* than the model using $N-1$, the two parameter model using $N-1$ has a lower *CP* than the two parameter model using $ln(N)$. Each of the models in this stage is multiplied by the model chosen in the first and second stages to obtain the predicted values of *L_del*. From table 7, we see that only the models with CP = 7.0 (4 parameters) and CP = 5 (5 parameters) lie reasonably close to the CP = p line. We choose the smaller model with $p = 4$.

Hence the final model for *L_del* has the form

$$L\_del = f(EB) \times g(S/Bl) \times h(N-1) \tag{4}$$

where

$$f(EB) = a \times e^{-b\times EB+c} \times (d - e \times EB + f \times EB^2) + g$$

$$g(S/Bl) = (1 - e^{-(i-j\times EB)\times S/Bl-k\times EB})$$

$$h(N\text{-}1) = e^{-(m-n\times EB+o\times EB^2+p\times EB^3)\times(N-1)}$$

Model	*Number of Parameters (p)*	*CP*
$e^{(-m\times(N-1))}$	1	103807.0
$e^{(-m\times ln(N))}$	1	102037.0
$e^{(-(m-n\times EB)\times(N-1))}$	2	12832.5
$e^{(-(m-n\times EB)\times ln(N))}$	2	16276.5
$e^{(-(m-n\times EB+o\times EB^2)\times(N-1))}$	3	501.0
$e^{(-(m-n\times EB+o\times EB^2+p\times EB^3)\times(N-1))}$	4	7.0
$e^{(-(m-n\times EB+o\times EB^2+p\times EB^3+q\times EB^4)\times(N-1))}$	5	5.0

Table 7: CP criterion for D-REG model (stage 3)

We now examine the performance of the delay regression model D-REG. Figure 10 compares values predicted by the D-REG model from the training and test sets against values obtained by simulation. Since the test set was generated for a $Pcr = 1000$ cells/sec while the training set used a $Pcr = 14150$ cells/sec, this shows that the model can be used to predict delays for other values of Pcr as long as Bl and Av/Pk remain the same (see section 3.4). Section 7.1 presents a numerical comparison of the D-REG model with the D-ALN and D-AMS models.

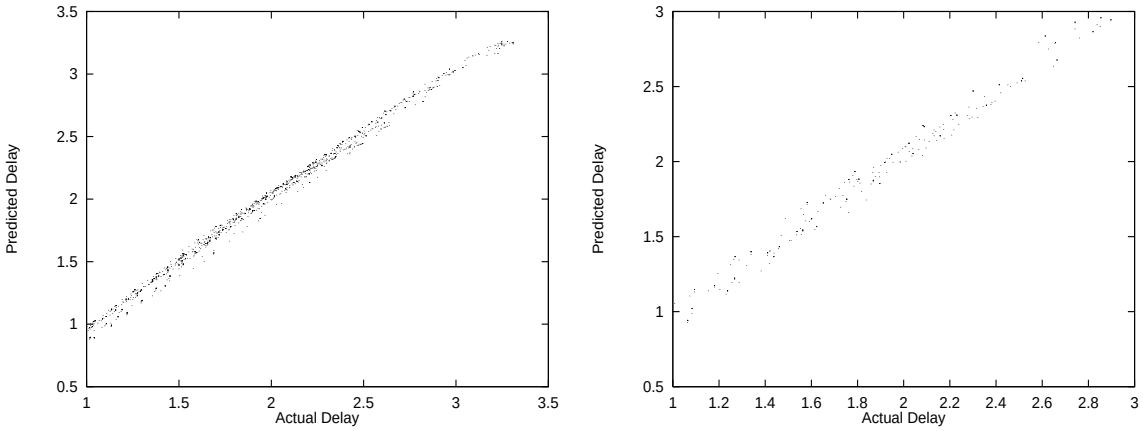

Figure 10: D-REG model predictions: training set (left) test set (right)

5.2 Regression model for cell loss

In developing the regression model for CLR, we follow the same procedure as for the delay model. The CP criterion described in section 5.1 is used to prevent overfitting. Because of the large range of Clr, we choose L_clr as the response variable for the model. Based on the simulation plots, the following observations can be made.

- L_clr decreases almost exponentially as a function of EB (figure 4 right).
- The form of the variation of L_clr with S is

$$a_{\infty}(EB) \times (e^{-b(EB)\times S/Bl})$$

where $a_{\infty}(EB)$ is solely a function of EB and is the value of L_clr when S/Bl is large.

- L_clr decreases exponentially as a function of N. The form of the function is

$$c(1) \times e^{-d(EB)\times(N-1)}$$

where coefficient $c(1)$ is the value of L_clr when $N = 1$ and is a function of S/Bl and EB.

We do not include S/Bl in the function for N or vice-versa for the same reasons as in the D-REG model. We select the smallest value of S (400) in the training set to correspond to the origin for S/Bl and hence replace S/Bl by $S/Bl - 400/353$. Similarly, we select $N = 1$ as the origin for the number of calls. The composite model is therefore of the form

$$L\_clr = (\alpha \times f(EB)) \times (e^{-\beta(EB)\times(S/Bl-400/353)}) \times (e^{-\gamma(EB)\times(N-1)}) \quad (5)$$

The C-REG model is also obtained in three stages. In the first stage, we set $N = 1$ and $S = 400$. The second and third terms in equation 5 reduce to 1 and so we can obtain a model for $\alpha \times f(EB)$. Table 8 shows the CP criterion for the different models tried for $a \times f(EB)$, the first term in equation 5. The model with 5 parameters (CP = 7.03902) and the model with 6 parameters are the possible choices. We pick the model with the lower number of parameters, as usual.

Model	*Number of Parameters (p)*	*CP*
$a + b \times EB$	2	3530.11
$a + b \times EB + c \times EB^2$	3	498.345
$a + b \times EB + c \times EB^2 + d \times EB^3$	4	67.7232
$a + b \times EB + c \times EB^2 + d \times EB^3 + e \times EB^4$	5	7.03902
$b \times EB + c \times EB^2 + d \times EB^3 + e \times EB^4 + f \times EB^5$	5	1.06562
$a + b \times EB + c \times EB^2 + d \times EB^3 + e \times EB^4 + f \times EB^5$	6	5.99999

Table 8: CP criterion for C-REG model (stage 1)

In the second stage, we choose $N = 1$ as before, but place no restriction on S. We already have obtained the first term in equation 5 and the third term reduces to 1. So we can obtain $-\beta(EB) \times (S/Bl - 400/353)$. Table 9 shows the CP criterion for the different models tried

Model	*Number of Parameters (p)*	*CP*
$g \times (S/Bl - 400/353)$	1	6339.87
$(g + h \times EB) \times (S/Bl - 400/353)$	2	401.138
$(g + h \times EB + i \times EB^2) \times (S/Bl - 400/353)$	3	31.5624
$(g + h \times EB + i \times EB^2 + j \times EB^3) \times (S/Bl - 400/353)$	4	4.00003

Table 9: CP criterion for C-REG model (stage 2)

in the second stage. Each of the models in this stage is multiplied by the model chosen in the first stage to obtain the predicted values of L_clr. From table 9, we see that only the model with 4 parameters is adequate.

In the third stage, we place no restriction on N. Table 10 shows the the CP criterion for the different models tried for $-\gamma(EB) \times (N-1)$, the last product term in equation 5. Each of the models in this stage is multiplied by the model chosen in the first and second stages to obtain the predicted values of L_clr. From table 10, we see that only the model with 4 parameters is adequate.

Model	Number of Parameters (p)	CP
$(k + l \times EB) \times (N-1)$	2	1287.76
$(k + l \times (S/Bl - 400/353)) \times (N-1)$	2	6662.78
$(k + l \times EB + m \times (S/Bl - 400/353)) \times (N-1)$	3	1275.32
$(k + l \times EB) \times (N-1) + m \times (N-1)^2$	3	590.717
$(k + l \times EB + m \times (S/Bl - 400/353)) \times (N-1) + n \times (N-1)^2$	4	566.563
$(k + l \times EB) \times (N-1) + (m + n \times EB) \times (N-1)^2$	4	382.839
$(k + l \times EB) \times (N-1) + m \times (N-1)^2 + n \times (N-1)^3$	4	228.313
$(k + l \times EB + m \times (S/Bl - 400/353)) \times (N-1) + (n + o \times EB) \times (N-1)^2$	5	344.301
$(k + l \times EB) \times (N-1) + (m + n \times EB) \times (N-1)^2 + o \times (N-1)^3$	5	126.64
$(k + l \times EB) \times (N-1) + (m + n \times EB) \times (N-1)^2 + (o + p \times EB) \times (N-1)^3$	6	31.2593
$(k + l \times EB + m \times (S/Bl - 400/353)) \times (N-1) + (n + o \times EB) \times (N-1)^2 + p \times (N-1)^3$	6	97.7137
$(k + l \times EB + m \times (S/Bl - 400/353)) \times (N-1) + (n + o \times EB) \times (N-1)^2 + (p + q \times EB) \times (N-1)^3$	7	7.00032

Table 10: CP criterion for C-REG model (stage 3)

Hence the final model for L_clr has the form

$$L\_clr = f(EB) \times g(S/Bl - 400/353) \times h(N-1) \tag{6}$$

where

$f(EB) = a + b \times EB + c \times EB^2 + d \times EB^3 + e \times EB^4$

$g(S/Bl\text{-}400/353) = (g + h \times EB + i \times EB^2 + j \times EB^3) \times (S/Bl - 400/353)$

$$h(N\text{-}1) = (k + l \times EB + m \times (S/Bl - 400/353)) \times (N-1) + (n + o \times EB) \times (N-1)^2 + (p + q \times EB) \times (N-1)^3$$

We now examine the performance of the CLR regression model. Figure 11 compares *L_clr* predicted by the model with the values obtained by simulation. As in the case of cell delay, the results obtained by evaluating the model on the test set show that the model can be used to predict CLR for different values of Pcr. Section 7.2 presents a numerical comparison of the CLR predictions by the C-AMS, C-ALN and C-REG models. As in the case of the C-ALN model, the performance of the C-REG model on the test set is not as good as the D-REG model. The reasons for this are mentioned in section 4.3.

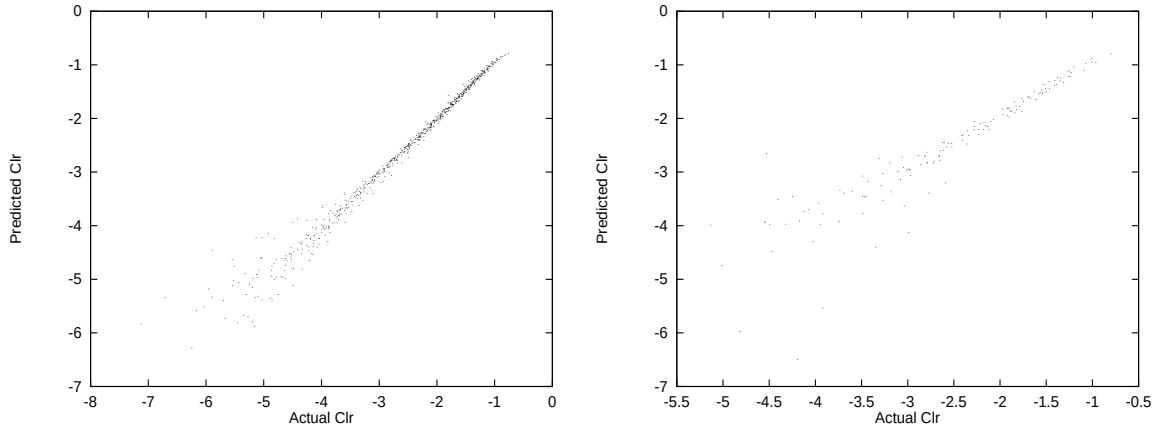

Figure 11: D-REG model predictions: training set (left) test set (right)

Figure 12 (left) compares D-REG model predictions for $N = 30$ with simulation values. Figure 12 (right) shows the prediction quality of the C-REG model on $N = 30$, a value not in the training set. It can be seen that the extrapolation by both models is quite accurate.

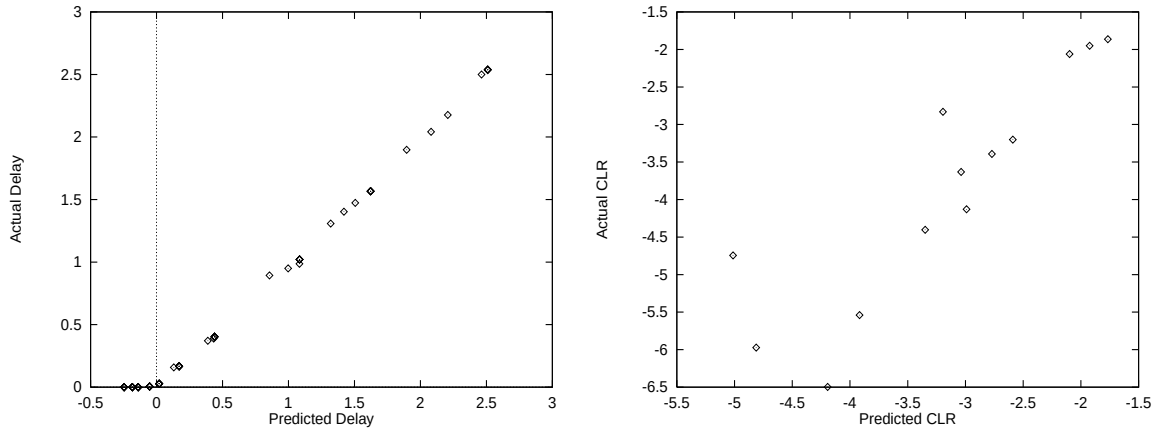

Figure 12: Extrapolation for N=30: D-REG model (left) C-REG model (right)

5.3 Residual analysis for regression models

Since the non-linear regression models use the least squares principle, the assumptions (section 4.4) inherent in the use of least squares apply to the models as well. From figure 13, the *L_del*-residuals plot (left), we can see that the errors are randomly distributed on either side of the horizontal axis, indicating that there is no systematic error. Since the logarithm function is nonlinear for values of the abcissa < 10, we have eliminated these values from the plot. As before, we are more interested in delay predictions at higher delays.

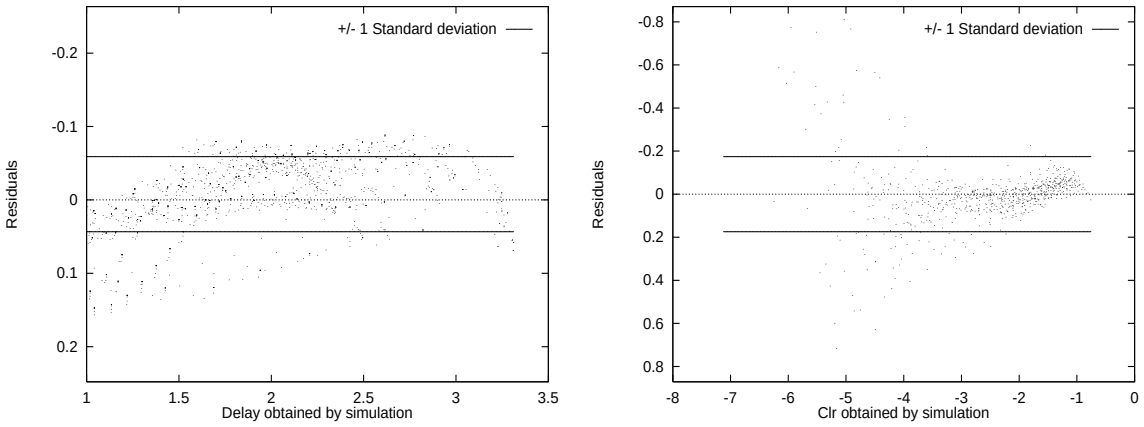

Figure 13: Regression L_del-residual plot (left) L_clr-residual plot (right)

The *L_clr*-residuals plot (right) shows that the residuals are fairly random up to $Clr = 10^{-4}$. For lower CLR, the simulation results themselves are inaccurate and therefore, the larger values of the residuals can be tolerated.

Model error terms are assumed to follow a normal probability distribution. That this assumption is not unjustified can be observed from histogram plots of *L_del* and *L_clr* residuals (figure 14).

6 The Anick-Mitra-Sondhi (AMS) Model

We review the fluid-flow, infinite buffer model developed by Anick, Mitra and Sondhi [1]. This model, although computationally intensive, was shown in [13] to be significantly more accurate than Guerin's or the binomial approach in [8]. Let the j-th component of the equilibrium probability distribution vector $\mathbf{F}(x)$ be the probability that j of the N sources are in the *On* state and that the buffer occupancy does not exceed x. Then,

$$\mathbf{F}(x) = \mathbf{F}(\infty) + \sum_{i=0}^{N-\lfloor \frac{BW}{Pcr} \rfloor - 1} e^{z_i x} a_i \phi_i$$

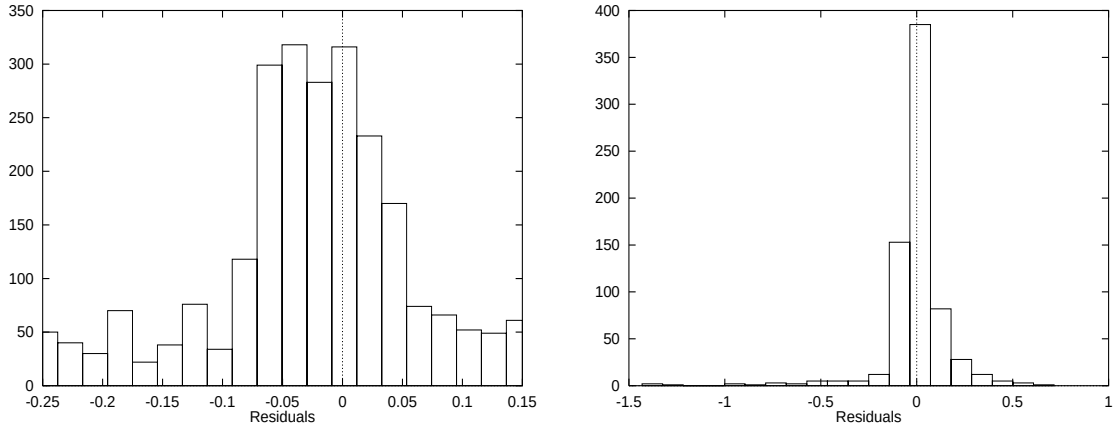

Figure 14: D-REG histogram of residuals (left) C-REG histogram of residuals (right)

where z_i and ϕ_i are the stable (negative) eigenvalues and eigenvectors associated with the differential equation satisfied by $\mathbf{F}(x)$, and the a_i are constants obtained from boundary conditions. [1] gives an explicit procedure for obtaining the z_i, ϕ_i, and a_i.

6.1 AMS cell loss model

As a first approximation, the cell loss rate of a finite system with S buffers can be approximated by the probability that the contents x of the infinite buffer exceed S. This overflow probability is

$$p_S = Pr(x > S) = \sum_{i=0}^{N-\lfloor \frac{BW}{Pcr} \rfloor - 1} e^{z_i S} a_i (\mathbf{1}' \phi_i)$$

This is an overestimate, because the mean occupancy of a finite buffer system is lower than that of the infinite buffer. We found empirically that p_S overestimates the real cell loss rate by at least a factor of 2. We therefore use the approximation

$$Clr = p_S/2 \tag{7}$$

Figure 15 compares the predictions of the C-AMS model on the training and test sets with simulation results.

The reason for the poor performance of the C-AMS model on the test and training sets is two-fold: at high loss rates, the model is inaccurate because the overflow probability used to approximate the cell loss probability is an overestimate. At low loss rates, the model is accurate enough, but the simulation results are not.

Section 7 presents a numerical comparison of the AMS results with the ALN and regression models.

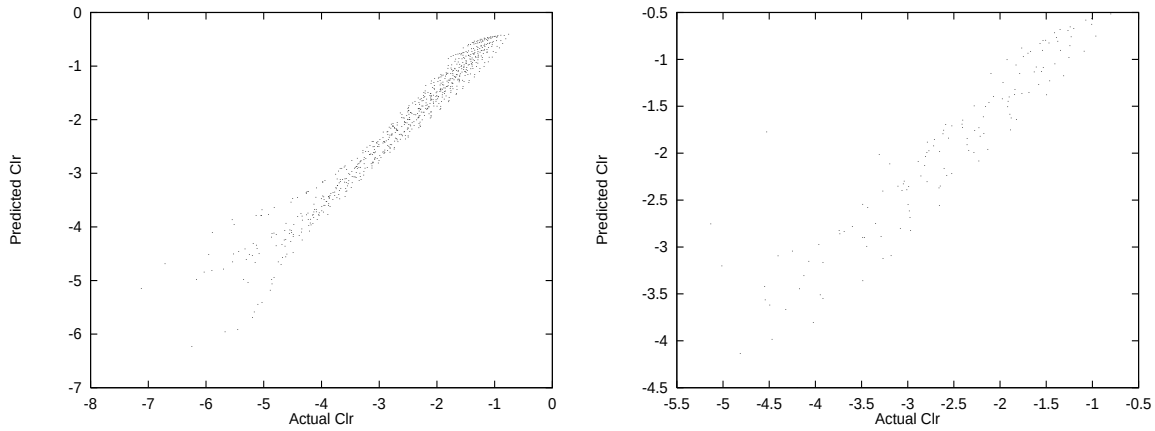

Figure 15: C-AMS predictions : training set (left) test set (right)

6.2 Extended AMS delay model

The mean occupancy of the infinite buffer is given by

$$E[x]_\infty = \sum_{i=0}^{N-\lfloor \frac{BW}{Pcr} \rfloor - 1} \frac{a_i(\mathbf{1}'\phi_i)}{z_i} \tag{8}$$

When there is very high cell loss ($> 10^{-2}$), $E[x]_\infty$ grossly overestimates the mean occupancy of a finite buffer and is often much greater than S. We propose approximating delay by the expectation of the buffer occupancy under the condition that this occupancy is no more than S:

$$Del = E[x]_S = \frac{\sum_{i=0}^{S} i \times Pr(x=i)}{\sum_{i=0}^{S} Pr(x=i)} \tag{9}$$

Figure 16 compares simulation results with the mean delay obtained using equations (8) and (9) for a few typical cases. While both equations give good results for high bandwidths (i.e., when cell loss is low), (9) is significantly more accurate in high cell loss scenarios.

Conditional probability can also be used to approximate jitter (defined as the standard deviation of the cell delay, i.e., the buffer occupancy). Again, the infinite buffer assumption greatly overestimates the variance of a finite buffer, and we approximate $E[x^2]$ by

$$E[x^2]_S = \frac{\sum_{i=0}^{S} i^2 \times Pr(x=i)}{\sum_{i=0}^{S} Pr(x=i)}$$

Then, the jitter is given by

$$jit_S = \sqrt{E[x^2]_S - E[x]_S^2} \tag{10}$$

Again, we compare simulation results with the jitter obtained using equation (10) and with the infinite buffer jitter (fig. 17). As in the case of delay, (10) is more accurate when

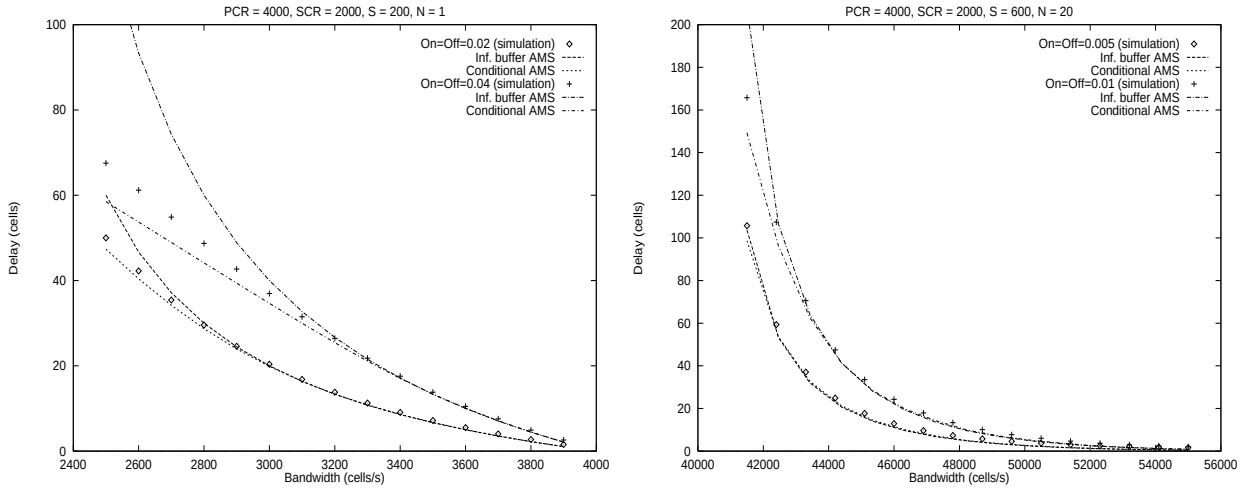

Figure 16: Infinite buffer delay and conditional delay

cell loss is high. Note that for our exponential On/Off sources, the jitter has the same order of magnitude as delay.

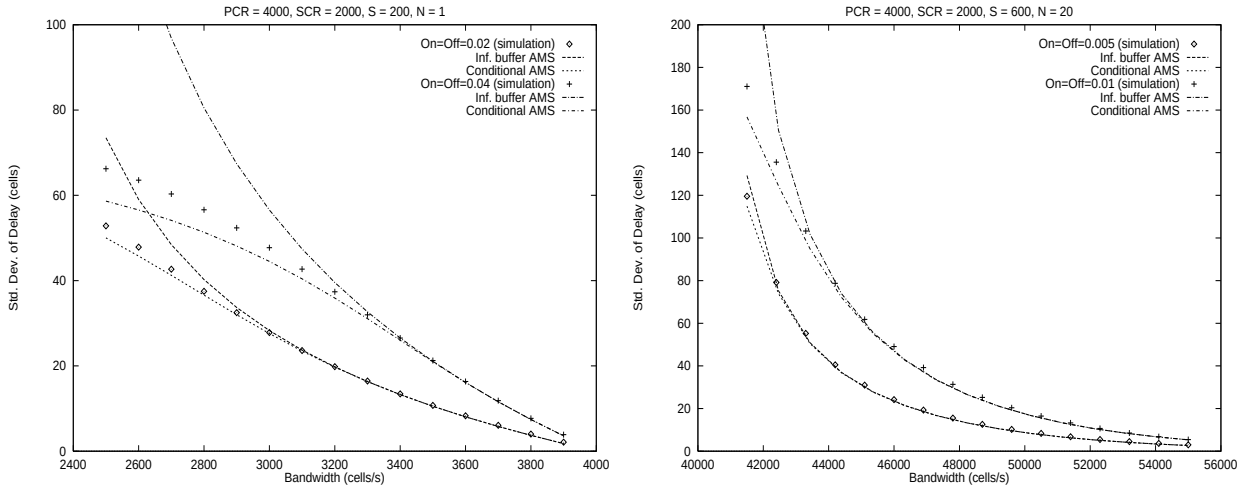

Figure 17: Infinite buffer jitter and conditional jitter

Figure 18 compares the estimates of the D-AMS model on the training and test sets with values obtained by simulation. It can be seen that the predictions are fairly accurate.

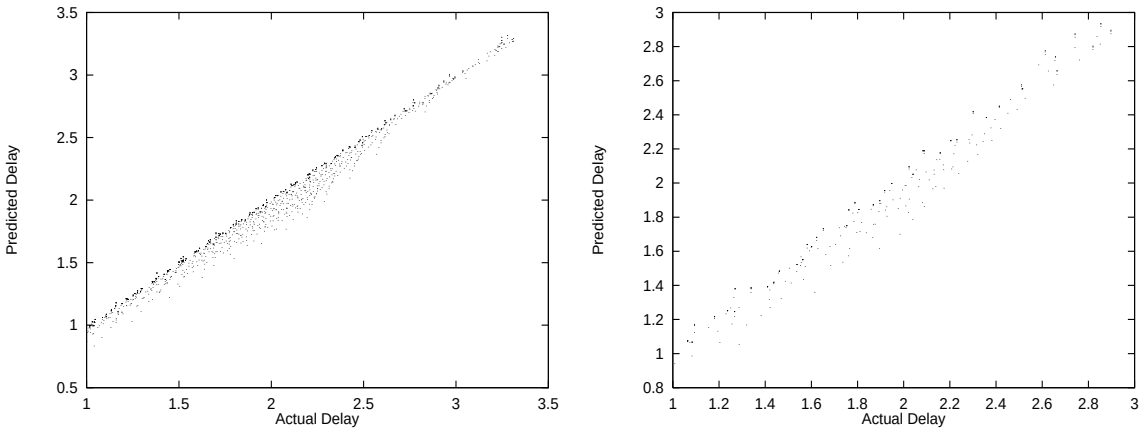

Figure 18: AMS DEL predictions: training set (left) test set (right)

7 Comparison of delay and cell loss predictions

7.1 Delay based comparisons

In this section, we compare delay predictions by the D-AMS, D-ALN and D-REG models against the delay values obtained from the simulation results.

Table 11 ranks the three models based on a visual inspection of the scatter plots for the test set, training set and for the number of calls $N = 30$.

Scatter Plot	D-AMS	D-REG	D-ALN
Training set	2	3	1
Test set	2	3	1
N=30	2	2	1

Table 11: Qualitative comparison of delay predictions

Table 12 makes a numerical comparison based on the average and maximum *relative* errors[3] in the test and training sets. From the table we see that the D-ALN performs the best on both the test and training sets, the D-AMS model comes second and the D-REG model is not far behind. The D-AMS model outperforms the D-REG model on both the test and training sets. However, the average error of the D-REG model on the test set is only 6.1% compared to the average error (3.1%) of the D-ALN model. The gain in computational speed is well worth the small sacrifice in accuracy.

[3]The errors are computed by comparing simulation and predicted values of L_del

Method	*Training set*		*Test set*	
	Maximum error	*Average error*	*Maximum error*	*Average error*
D-AMS	19.9%	2.3%	18.1%	3.1%
D-REG	29.5%	5.5%	44.5%	6.1%
D-ALN	1.67%	0.32%	10.5%	2.6%

Table 12: Numerical comparison of delay predictions

7.2 Cell loss based comparisons

In this section, we compare the CLR predictions by the C-AMS, C-ALN and C-REG models against the CLR values obtained by simulation.

Table 13 ranks the three models based on a visual inspection of the scatter plots for the test set, training set and for number of calls $N = 30$. Table 14 makes a numerical comparison based on the average and maximum *relative* errors[4] in the test and training sets.

Scatter Plot	**C-AMS**	**C-REG**	**C-ALN**
Training set	3	2	1
Test set	2	1	1
N=30	1	2	2

Table 13: Qualitative comparison of CLR Predictions

From table 14, we see that the C-ALN does not outshine the cell loss models—unlike the D-ALN which outperformed the other delay models. The C-REG model does better than the other models, closely followed by the C-ALN model. The average error of the C-REG model, however, on the test set is only 6.8%, indicating that the model is fairly accurate as well as computationally non-intensive. Possibly the large noise in the CLR values, particularly at lower CLR, causes the poor performance of the cell loss models as compared to the delay models. (sections 4.3 and 6.1).

8 Application: Effective Bandwidths

In the preceding section, we examined three methods of estimating the cell loss and mean delay in our system. We now apply those methods to effective bandwidth computation. We compute effective bandwidths that satisfy cell loss requirements, mean delay requirements,

[4]The errors are computed by comparing simulation and predicted values of L_clr

Method	*Training set*		*Test set*	
	Maximum error	*Average error*	*Maximum error*	*Average error*
C-AMS	60.5%	23.7%	60.8%	25.7%
C-REG	24.3%	3.1%	54.9%	6.8%
C-ALN	12.7%	1.2%	55%	7%

Table 14: Numerical comparison of CLR predictions

and both. Two requirement sets are considered: the first requirement set (CLR $\leq 10^{-2}$, delay ≤ 100) corresponds to a situation with stringent delay and jitter requirements but relatively high tolerance for loss (e.g., real-time video); the second requirement set (CLR $\leq 10^{-4}$, delay ≤ 1000) corresponds to the more common case where low loss is required, but delay can be tolerated. Throughout this section, we set $Pcr = 14150$, $t_{ON} = 0.025$, and $t_{OFF} = 0.035$ (this yields an Scr of $0.417 \times Pcr = 5896$). The bandwidths are plotted per source and relative to the Pcr, and are thus in $[0.417, 1]$.

Of the three methods, only the ALN could be explicitly inverted; to obtain effective bandwidths from the regression function, and in the AMS model, we used a simple binary iteration approach. We found that generally, accurate results can be obtained in only 5–10 iterations. The declared range of input values for the ALN was $[0.4, 1]$, and the inverted ALN therefore gave values in $[0.4, 1]$, sometimes returning values slightly below the Scr.

8.1 Effective Bandwidth vs. Number of Calls

We compute effective bandwidths as a function of the number of calls, which we vary from 1 to 20. The buffer size S is fixed at 5000. The results are plotted in figure 19 (requirement set 1 on the left, requirement set 2 on the the right). There are six lines per requirement set: for each of the three approximation methods there is a line for the delay-based effective bandwidth and another for the cell-loss-based effective bandwidth. The effective bandwidth needed to satisfy both delay and cell loss requirements is simply the maximum of the two lines.

We note that the delay requirement is dominant in requirement set 1. The three estimates for delay-based effective bandwidth are very close to each other and show the typical decaying shape (indicating a statistical multiplexing gain). The estimates for the loss-based effective bandwidth are close to the Scr even for $N = 1$ and do not change when N is increased. In this case, it is clear that the ALN underestimates the effective bandwidth, because the estimate is about 5% below the Scr.

For requirement set 2, the loss requirement dominates slightly. As expected, the tighter loss requirements increased the loss-based effective bandwidth, and the less stringent delay requirements reduced the delay-based effective bandwidth.

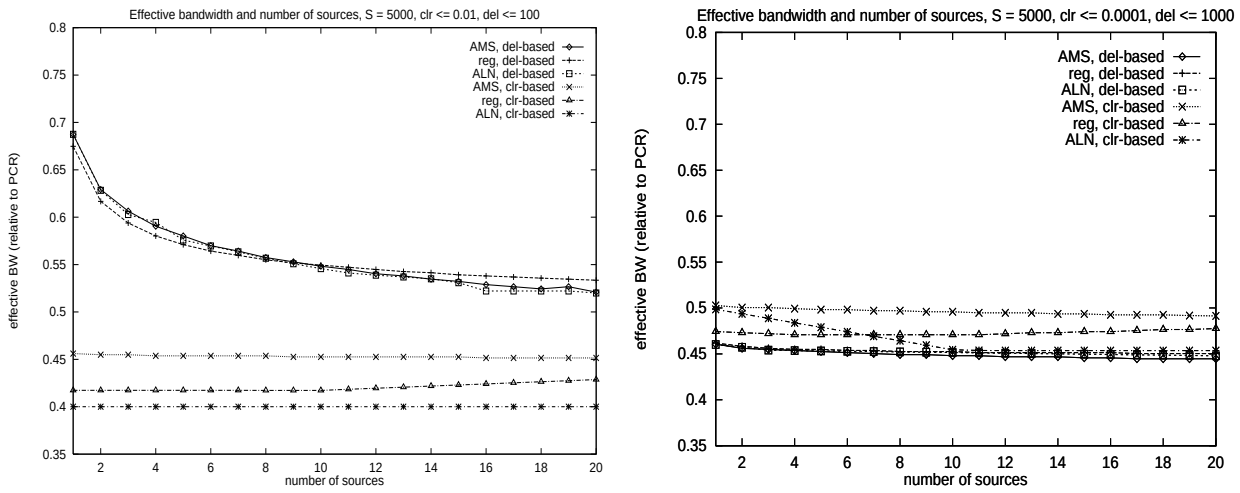

Figure 19: Effective Bandwidth vs. N

8.2 Effective Bandwidth vs. Buffer Size

We compute the effective bandwidths as a function of the buffer size, which we vary from 1 to 10000. The number of calls N is fixed at 10. The results are plotted in figure 20 (requirement set 1 on the left, requirement set 2 on the right). Once again, there are six lines per requirement set: for each of the three approximation methods, there is a line for the delay-based effective bandwidth and another for the cell-loss-based effective bandwidth.

The values for delay-based effective bandwidth computed by the three methods exhibit a characteristic *step* form: the delay requirements can be satisfied by any bandwidth if the buffer size is less than the delay requirement; in our figure, the dots are plotted close to the sustained cell rate line. Once the buffer size is relatively large, adding buffers leaves the mean delay, and therefore the delay-based effective bandwidth, at a constant level. Between these two extremes, there is a narrow range of buffer values where an increase in buffer size leads to an increase in mean delay. The reason for this increase is a simultaneous sharp decrease in the cell loss rate, resulting in the delay of cells that would otherwise be dropped.

Comparing the graphs for loss-based and delay-based effective bandwidth, we note that the cell loss requirement dominates for small buffer sizes and that the delay requirement dominates for a larger buffer sizes. The crossover point between these two buffer size regions depends on the requirements and on the traffic characteristics. There is no increase in statistical multiplexing gain once the cross-over point has been reached.

The three estimates for loss-based effective bandwidth are divergent when effective bandwidth is high (i.e., when the buffer size is small), and somewhat closer when effective bandwidth is low. This can be explained by the fact that the regression function and the ALN are based on simulations of buffer sizes ≥ 400, and are therefore inaccurate when buffer sizes are small. In the case of delay-based effective bandwidths, the three estimates are very close to each other.

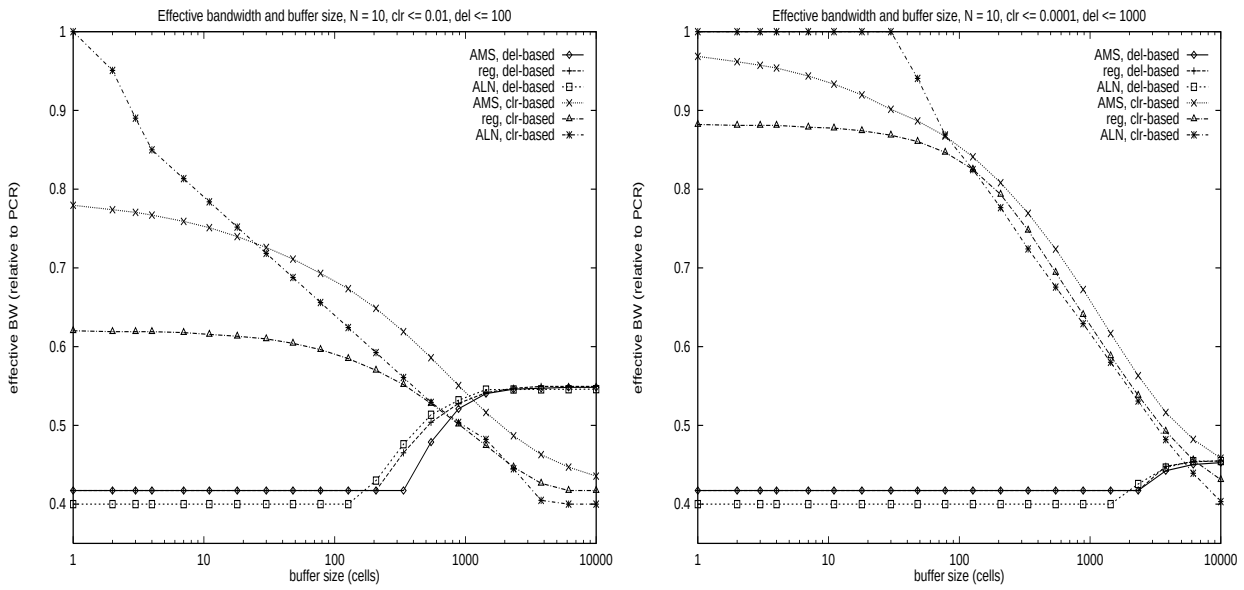

Figure 20: Effective Bandwidth vs. S

9 Conclusions

In this paper we have proposed two simple and reasonably accurate schemes for predicting the delay and cell loss when a number of bursty sources are multiplexed at a link with finite buffer space. We have demonstrated the use of the models in predicting the effective bandwidth requirement of each source as well as the need for considering both delay and cell loss requirements to obtain the effective bandwidth.

Unlike other schemes, these models use the number of sources N as an input, leading to delay and loss computation times that are independent of the number of sources being multiplexed. Though we have used On/Off bursty sources in order to compare the results with other equivalent bandwidth schemes, the techniques presented here can be extended to complex sources that cannot easily be modeled analytically, e.g., aggregate LAN traffic, MPEG traffic. The models can be developed fairly quickly as compared to analytical models, which is particularly valuable given the unpredictable nature of traffic patterns in B-ISDN networks as a result of the diversity of emerging services.

The ALN model has the advantage over other neural networks that the same network that is trained to predict cell loss or delay can be inverted to predict effective bandwidth instead. This means that the ALN can make effective bandwidth predictions as quickly as it can make delay or loss predictions. Since the evaluation time is very small—of the order of milliseconds—ALNs are highly suitable for use in real-time CAC. ALN implementation in hardware is easy because the ALNs are composed of AND gates, OR gates and simple linear threshold elements. This can result in an increase in evaluation speeds by an order of magnitude or more. The regression models, although slower to evaluate than the ALN

models, are much faster than the AMS models. They have the advantage over ALN models in that they are easier to verify for correctness under all possible input combinations. Another advantage of the regression models is that it is possible to obtain an insight into the qualitative behaviour of delay and cell loss by inspection of the models alone—ALN models must be evaluated to retrieve this information and the evaluation must be done for a large number of combinations of the predictor variables before a comprehensive picture can be obtained.

The regression model becomes increasingly more difficult to obtain as the number of input variables increases. The ALN model, however, can adapt more easily to larger dimensions. The reason is that, in the regression model, the interactions between input variables must be specified explicitly. The ALN requires only the relationship between the response variable and each of the inputs to be specified. It learns on its own the relation between the input variables. This markedly simplifies the process of obtaining a good model.

References

[1] D. Anick, D. Mitra, and M. M. Sondhi. Stochastic theory of data–handling system with multiple sources. *The Bell Technical Journal*, 61(8):1871, 1982.

[2] W. W Armstrong and M. M. Thomas. *Handbook of Neural Computation – Adaptive Logic Networks(Section C1.8)*. Oxford University Press, 1996.

[3] M. Decina and T. Toniatti. On bandwidth allocation to bursty virtual connections in ATM networks. *ICC '90*, 3:844–851, 1990.

[4] P. Gburzynski. *Protocol Design for Local and Metropolitan Area Networks*. Prentice–Hall, New Jersey, 1996.

[5] R. Guerin, H. Ahmadi, and M. Naghsineh. Equivalent capacity and its application to bandwidth allocation in high–speed networks. *IEEE Journal on Selected Areas in Communications*, 9(7):968, 1991.

[6] J. Hyman, A. Lazar, and G. Pacifici. Real–time scheduling with quality of service constraints. *IEEE Journal on Selected Areas in Communications*, 9(7):1052–1063, 1991.

[7] C.L Mallows. Some comments on c_p. *Technometrics*, 15:661–675, 1973.

[8] T. Murase, H. Suzuki, S. Sato, and T. Takeuchi. A call admission control scheme for ATM networks using a simple quality estimate. *IEEE Journal on Selected Areas in Communications*, 9(9):1461, 1991.

[9] R. Onvural. *Asynchronous Transfer Mode Networks : Performance Issues*. Artech House Inc., MA, 1994.

[10] K. M. Rege. Equivalent bandwidth and related admission criteria for ATM systems — a performance study. *International Journal of Communication Systems*, 7:181, 1994.

[11] Gunst. R.F and R.L. Mason, editors. *Regression Analysis and its Application — A Data–Oriented Approach.* Marcel Dekker Inc., New York, 1980.

[12] SPSS Inc., Chicago, IL. *SPSS for Windows Advanced Statistics Release 5*, 1992.

[13] E. D. Sykas, K. M. Vlakos, K. P. Tsoukatos, and E. N Protonotarios. Congestion control — effective bandwidth allocation in ATM networks. *IFIP Transactions C [Communication Systems]*, C–14:65, 1993.

[14] F. Vakil. A capacity rule for ATM networks. *GLOBECOM '93*, pages 406–416, 1993.