Learning Algorithm for CAC Adjustment in ATM Networks

Z. Dziong, M. Ji, and Y.T. Wang

Performance Analysis Department
Bell Labs, Lucent Technologies
101 Crawfords Corner Rd, Holmdel, NJ 07733, USA
Tel: (1) 732 949-2459, Fax: (1) 732 834-5906
E-mail: zdziong@lucent.com

Abstract

We propose an algorithm for measurement based CAC adjustment in ATM networks. The algorithm was derived under requirement that it is applicable to ATM switches without proprietary solutions. For this reason the algorithm is fed by simple cell loss measurements. Based on this measurements the CAC overbooking factors are updated once a day. The updates are optimized based on the history of the measurements, hence the name *learning algorithm*. The preliminary numerical study shows that the algorithm is robust and delivers high bandwidth utilization while the quality of service constraints are satisfied.

1 Introduction

It is well known that CAC algorithms used in ATM systems can be quite conservative since they are designed for the worst case source behavior and the connections may be not fully active during their life time. Moreover the analytical models applied in these algorithms are also conservative due to difficulties in exact modeling of the connection aggregate process. For these reasons it is desirable to apply some types of measurements to adjust the CAC mechanism in order to achieve good utilization of the network resources. In the literature several models were proposed for adaptive CAC adjustment, e.g. [2, 3]. Most of them try to provide adaptation in the time scale of connection inter-arrival times. Such an approach usually requires sophisticated measurements and is difficult to implement. In this paper we focus on a simple approach which acts in a longer time scale (days) and can be applied even to switches without any modifications.The approach relies on availability of two basic features. The first is the overbooking control "knob" which allows to regulate the CAC overbooking with fine granularity. This feature is already implemented in most of the switches. The second required feature is measurement of lost cells in the aggregate of the controlled traffic on each controlled link in say 15 minute intervals. This feature is also available in many switches since it is consistent with measurements recommended by ATM Forum at M4 interface. The only drawback of this recommendation is that it specifies only counts for the aggregate over all services which in general is not that useful since it mixes the UBR traffic with the loss sensitive traffic.

The proposed algorithm uses the measurements of lost cells to decide whether the overbooking factor should be increased or decreased in a fashion that optimizes the bandwidth utilization and provides required QoS. In a sense the algorithm learns from the effects of its actions, hence in the sequel we refer to it as *learning algorithm*. The efficiency of the learning algorithm is best when it is optimized based on measurements from many controlled links (switches). Therefore the natural location of this algorithm is in a traffic management center which supervises a set of ATM switches.

The paper is organized as follows. After the problem definition, Section 2, we describe the main concept of the algorithm in Section 3. A more detailed description of the algorithm is given in Section 4, including optimization part. Preliminary numerical results are given in Section 5.

2 Problem formulation

Let us consider first an ATM link with capacity L and one service/priority class. For given set of connections the aggregate effective bandwidth, D, can be defined as $D = L'$ where L' is the reduced link speed for which the QoS constraints are tightly met.

In general the aggregate effective bandwidth is calculated by the CAC algorithm based on the connections' policing parameters. This value, D_p^{CAC}, is then used for admission decision concerning a new call. There are several reasons for which D_p^{CAC} can be significantly larger than the actual bandwidth required, D^a:

- CAC model conservatism (error): up to $\sim 15 - 30\%$ [simulations]
- Oversized policing parameters: up to $\sim 5 - 10\%$ [guess]
- Variability of a source traffic profile: up to $\sim 10 - 20\%$ [guess]
- Non-activity of sources: total up to $\sim 400\%$ [reported cases]

In order to increase the bandwidth utilization one can try to adjust the bandwidth allocated to connections based on some measurements of the aggregated cell process. This can be realized by means of the overbooking mechanism controlled by the the over-booking factor defined as

$$\alpha^t = \frac{D_p^{CAC}}{D_t^{CAC}} \tag{1}$$

where D_t^{CAC} is adjusted aggregate bandwidth which should be allocated to the connections. The corresponding over-booking gain is defined as

$$\alpha^{t'} = \frac{D_p^{CAC} - D_t^{CAC}}{D_t^{CAC}} = \alpha^t - 1 \tag{2}$$

One of the key constraints in the considered framework is the assumed 15 minutes measurement interval. Note that there may be several connections leaving and coming during such an interval. Thus, in general, any measurement cannot be related directly to a particular set of connections. This feature, combined with the reduced effectiveness of policing shield resulting from over-booking, causes that the CAC adjustment cannot act in real-time but rather has to be based on a longer history of measured statistics.

2.1 "Busy hours" periods

In general the most relevant measurements are in the periods when all (or almost all) of the link bandwidth is allocated to the connections by CAC algorithm at least for some time. We refer to such a period as "busy hour" period. Note that the "busy hour" periods can occur in different periods of the day. This feature can be used to differentiate CAC tuning in different "busy hour" periods due to a different mix of connections and their parameters. In particular it is possible that on average the permanent connections generated by business customers will be more active during the day-time while in the evening hours there will be more switched connections generated by residential customers. This may allow larger over-booking during evenings. Analogous differentiation can be done for different days of the week or some seasons of the year. The proposed algorithm can be implemented independently for each of the "Busy hours" periods.

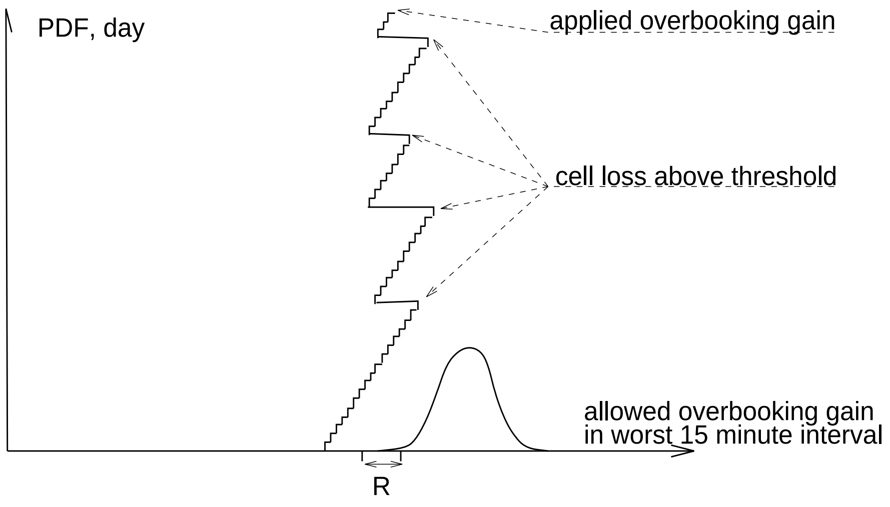
PDF, day
applied overbooking gain
cell loss above threshold
allowed overbooking gain in worst 15 minute interval
R

Figure 1: Learning approach.

2.2 Generic problem formulation

For presentation simplicity we assume that in the worst 15 minute interval of the considered "busy hour" period the level of offered connections provides that whole link bandwidth is allocated to the carried connections (at least for some time). Now let us assume that we know the maximum allowed overbooking, α^a, for this busy hour. This overbooking corresponds to the value for which the maximum actual aggregated effective bandwidth in the considered period is equal to the link capacity. For given CAC algorithm, the maximum allowed overbooking will vary from one day to another due to randomness of source activities and connection mixtures. If we could estimate somehow this overbooking based on measurements then its PDF could be approximated from a histogram of the estimates over many days. An example of such a PDF is given in Figure 1. Based on this PDF and required QoS constraints, the tuned overbooking factor can be evaluated from the following condition:

$$P\{\alpha^t > D^a - R\} = \epsilon \tag{3}$$

where ϵ is determined by QoS requirements and R is added for protection against the PDF estimation error.

3 Learning approach

In the following we assume that only QoS measurements over the 15 minutes intervals are available. The approach is similar to the one applied for power control in the CDMA wireless access networks and many similar dynamic control problems. Namely, at the beginning of subsequent instances of a given "busy hour" period the over-booking gain is increased in small steps δ_{up} until the QoS requirements are met or violated. If the QoS requirements are violated the over-booking gain is immediately decreased in a large step, $\delta_{dn}(\overline{V}_{QoS})$, whose size is a function of the measured QoS violation degree, $\overline{V}_{QoS}$. This procedure is illustrated in Figure 1. The effectiveness of this approach is a function of steps, $\delta_{up}, \delta_{dn}(\overline{V}_{QoS})$. In particular, by reducing δ_{up} and increasing $\delta_{dn}(\overline{V}_{QoS})$ one can reduce the level and frequency of losses. Acting in the opposite direction increases bandwidth utilization. The optimal values of δ_{up} and $\delta_{dn}(\overline{V}_{QoS})$ can be estimated from long term statistics of the system performance as shown in Section 7. Obviously this process should be continuous to adapt to changing environment. Also it may be advantageous to analyze the samples from different ports and switches to extract some features which are independent from a particular port traffic process. In particular one can try to extract information about the value of the over-booking parameter which is safe

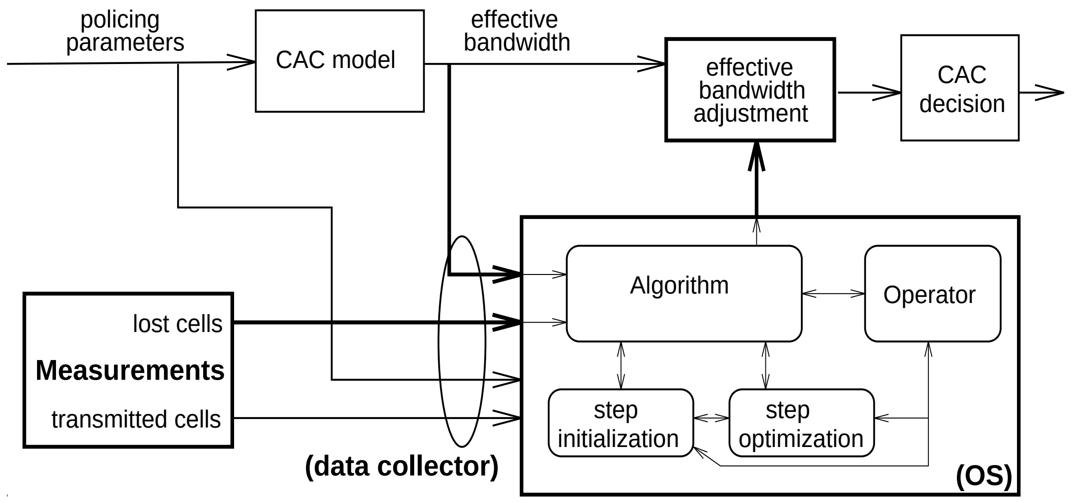

Figure 2: Learning approach architecture.

under any circumstances. Another possibility is to find features of the considered PDFs which are a function of the link speeds. The general architecture of the learning approach is given shown in Figure 2.

4 Learning algorithm

In the following we provide a description of the learning algorithm. The main structure of this algorithm is illustrated in Figure 3.

4.1 Initialization

Initialization consist of four main functions: determination of the "busy hour" periods, selection of VP/VCs, evaluation of initial distribution of the over-booking gain, and evaluation of initial values of the steps. Unless there is some prior knowledge of relevant traffic characteristics, the initialization process is based on some measurements. In this case the initialization phase will require a reasonable amount of time (e.g. one month). Due to space limitation we describe in short only the evaluation of initial distribution of the over-booking gain, since it illustrates at the same time how different priorities are treated in this framework.

4.1.1 Initial distribution of the over-booking gain

The initial distribution of the over-booking gain has to be determined based either on some arbitrary assumptions or on some initial period of measurements where the tuning algorithm will not be activated. In the latter case a simple approach is to observe the average rate, $\overline{m}_i^j$, of the randomly selected VP/VCs (selection should be changed from time to time) and compare these values with the declared sustained rates, S_i^j. Then the applied over-booking gain distribution can be proportional to the observed average ratios of declared and measured values for each service class.

It is important to note that in general the CDV measurements may not be available. This feature, combined with the fact that in some switches there is no active buffer thresholds for real-time traffic (CBR and VBR$_{rt}$), makes it possible that the CDV constraints for CBR and VBR$_{rt}$ can be violated (before the cell losses for these services are visible) when the level of VBR$_{nrt}$ and ABR traffic is below certain low threshold. To protect system against this risk, the over-booking of CBR and VBR$_{rt}$ services should be more conservative than over-booking of VBR$_{nrt}$ and ABR services. In other words the lower priority the more aggressive over-booking

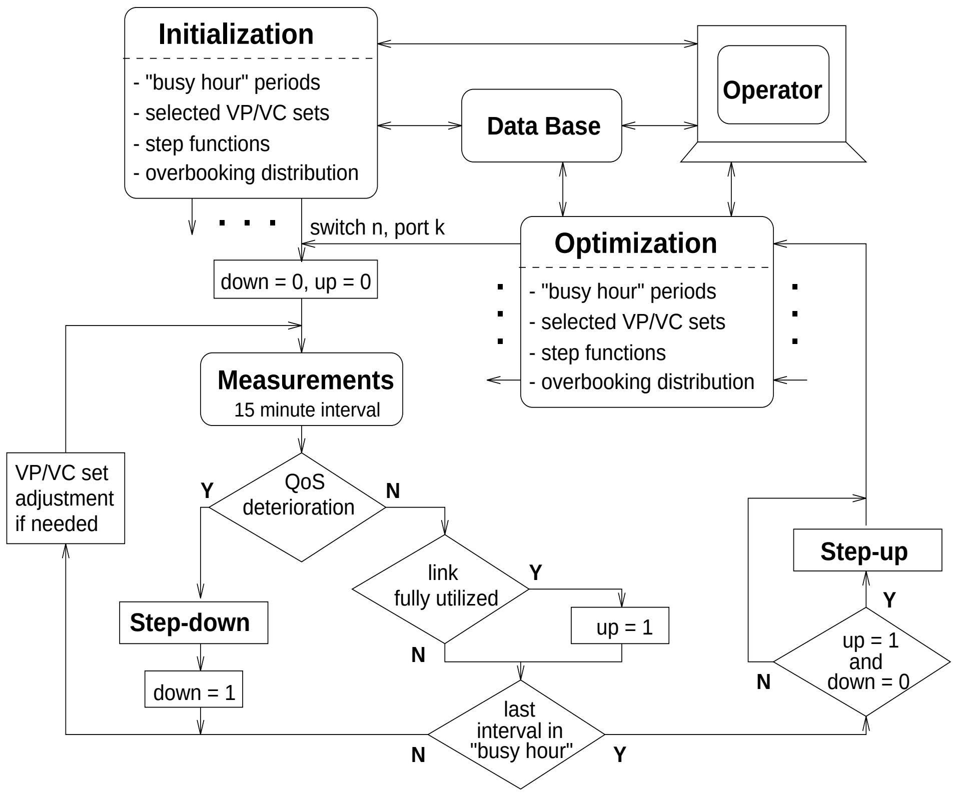

Figure 3: Learning algorithm for CAC adjustment.

should be applied. For example the over-booking distribution factors, γ_j, could be defined as follows:

$$\gamma_j = \kappa_j[\frac{\sum_i S_i^j}{\sum_i \overline{m}_i^j} - 1] \tag{4}$$

where $\kappa_j > 1.0$ for services with aggressive over-booking and $\kappa_j < 1.0$ for services with conservative over-booking. In the following we assume that the over-booking gains for each controlled service class will be proportional to the over-booking distribution factors. In other words one can define the normalized over-booking gain

$$\alpha = \frac{\alpha_j}{\gamma_j} \tag{5}$$

which is the same for all service classes on the link. Conversely we have

$$\alpha_j = \alpha\gamma_j \tag{6}$$

In the remainder of this report α without the service index j will denote the normalized over-booking gain.

4.2 Algorithm optimization

In general the algorithm optimization can be executed during the night hours which allows for extensive calculations.

The list of basic optimization functions is analogous to the list of initialization functions:

- Optimization of "busy hour" periods.
- Optimization of the VP/VC selection.
- Optimization of the over-booking distribution.

- Optimization of the step-up and step-down functions (includes safe, unknown and risky regions).

Optimization of the first three functions is similar to the initialization process based on measurements. In other words these parameters are updated from time to time based on relevant measurements in order to take into account long time scale changes in traffic profiles and source behaviors.

Optimization of the step-up and step-down functions is a key element of the learning approach since the step sizes define the QoS performance of the tuned system. In the next section we focus on this function.

4.3 Optimization of the step-up and step-down functions

4.3.1 Estimation process

Optimization of the steps is based on three types of functions estimated from data histograms:

- $\overline{CLR}_j(\alpha)$; This function relates the normalized over-booking gain, α, to long term average cell loss ratio of service j, $\overline{CLR}_j$, which is experienced for this α.
- $\overline{CLR}_j^{loss}(\alpha)$; This function relates the normalized over-booking gain, α, to average cell loss ratio of service j in 15 minutes intervals with losses, $\overline{CLR}_j^{loss}$, which is experienced for this α.
- $\overline{\alpha}(CLR_j^m)$; This function relates the measured cell loss ratio of service j, CLR_j^m, to $\overline{\alpha}$ defined as average over normalized over-booking gains for which this CLR_j^m is achieved.

Note that apart from functions related to a particular service, one can generate also functions for aggregate traffic over all services with bandwidth reservation. The histograms used for function estimation are build from the pairs $[\alpha, CLR_j^m]_k$ obtained for each instant of busy hour, where k is the port (sub-port) index and CLR_j^m corresponds to cell loss rate in the worst 15 minute interval in the busy hour period. In general for given type of ports one can try to estimate average and individual functions. The average functions are generated using data from all ports at the same time without distinction between different ports. The individual functions are generated for particular ports. Since the data available for a particular port may be insufficient to generate a reliable estimation, one can approximate the individual functions by using both average and individual statistics within the framework of BLUP (Best Linear Unbiased Predictors) [4]. Confidence intervals can be estimated if necessary. Apart from estimation of the defined functions, long term cell loss ratios per service per port (sub-port), $CLR_{j,k}$ should be estimated. This can be done by means of moving average or exponential smoothing.

4.3.2 Step optimization

In general the main objective of the step optimization is to provide that the long term cell loss ratios are within the constraints. Note that the estimated functions and parameters provide only approximate knowledge of the system behavior since the underlying processes include elements of non-stationarity. In particular the source activity and connection mix are factors which can change in time and may influence the algorithm performance. Therefore there is no unique solution to the problem and one can try several options which may provide the main objective but can differ in some other characteristics which may be of importance to the network operator. Specifically the issues of potential risk, convergence speed and the bandwidth utilization are of interest. In the following we describe a framework which we believe is robust but by no means covers all possibilities and several other options can be tried. To simplify presentation we assume that only average functions are available.

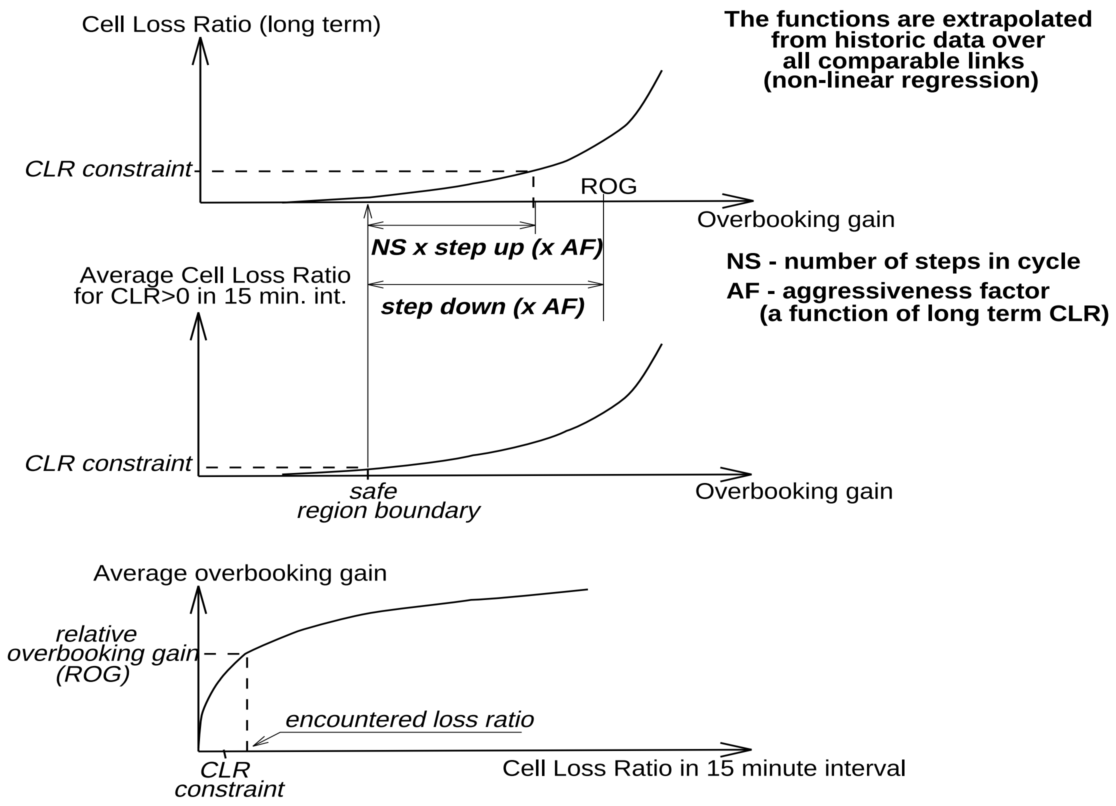
Cell Loss Ratio (long term)
The functions are extrapolated from historic data over all comparable links (non-linear regression)
CLR constraint
ROG
Overbooking gain
NS x step up (x AF)
step down (x AF)
NS - number of steps in cycle
AF - aggressiveness factor (a function of long term CLR)
Average Cell Loss Ratio for CLR>0 in 15 min. int.
CLR constraint
safe region boundary
Overbooking gain
Average overbooking gain
relative overbooking gain (ROG)
encountered loss ratio
CLR constraint
Cell Loss Ratio in 15 minute interval

Figure 4: Illustration of step optimization.

Definition of some parameters:

- Safe region boundary, SF, is defined as maximum normalized over-booking parameter for which $\overline{CLR}_j^{loss}(\alpha) \leq CLR_j^c$ for all services, where CLR_j^c is service j constraint for CLR.
- Target over-booking gain, α^{QoS}, is defined as minimum value of normalized over-booking gain for which $\overline{CLR}_j(\alpha^{QoS}) = CLR_j^c$ for at least one service.
- Relative over-booking gain, α^r, is defined as $\alpha^r = \overline{\alpha}(CLR^{15})$, where CLR^{15} is aggregate cell loss ratio for given 15 minute interval.

Evaluation of steps: Once cell loss ratio of service j at port k in given 15 minute interval exceeds pre-established threshold (e.g. $CLR_{j,k}^{15} > \theta_1 CLR_j^c$ where θ_1 is aggressiveness coefficient with default value 1), the following algorithm is executed:

- Find relative over-booking gain, $\alpha^r = \overline{\alpha}(CLR^{15})$.
- Evaluate step-down:
$$\delta_k^{down} = [\alpha^r - SF]\theta_2(CLR_{j,k} - CLR_j^c) \tag{7}$$
where $\theta_2(CLR_{j,k} - CLR_j^c) \in [0,1]$ (default value 1) is aggressiveness factor being a function of difference between the long term cell loss ratio and its constraint (for service j that encountered the cell losses).
- Evaluate step-up:
  - Linear approach:
$$\delta_{k,1}^{up} = \frac{\alpha^{QoS} - SF}{N_{st}}\theta_2(CLR_{j,k} - CLR_j^c) \tag{8}$$
where N_{st} is an integer factor defining aggressiveness of steps.

– Non-linear approach:
In this case we start from the step $\delta_{k,1}^{up}$ evaluated as in the linear model. Then each subsequent step is evaluated as follows:

$$\delta_{k,i+1}^{up} = \delta_{k,i}^{up} \frac{\overline{CLR}(\alpha_{k,i-1})}{\overline{CLR}(\alpha_{k,i})} \tag{9}$$

where $\alpha_{k,i} = \alpha_{k,i-1} + \delta_{k,i}^{up}$. This way the steps are becoming smaller when the risk of losses is becoming larger. Obviously the factor N_{st} should be smaller in the non-linear approach.

The above procedure is illustrated in Figure 4.

5 Numerical results

We did some preliminary tests of the presented concepts in a simplified simulation environment using the following assumptions:

- all connections are homogeneous, i.e., they have the same traffic descriptors, QoS requirement.
- all connections are deterministic on-off sources.
- only the worst state in busy period is modeled
- actual traffic underutilization and corresponding allowed overbooking vary from day to day around an average value according to uniform distribution within given range.
- large deviation theory is used to generate cell loss ratio
- the steps of the learning algorithm are chosen in advance

Let us define the connection traffic descriptors: P — peak cell rate (PCR), r — sustainable cell rate (SCR), and T_{on} — burst length. The actual connection parameters are modeled as

- $P^a = \beta_1 P$ — Actual peak cell rate (PCR) where β_1 denotes the actual PCR factor.
- $r^a = \beta_2(1+\beta_4\eta)r$ — Actual sustainable cell rate where β_2 is the actual SCR factor and η is uniformly distributed noise between [0 1], β_4 denotes the level of day-to-day variations of allowed overbooking.
- $T_{on}^a = \beta_3 T_{on}$ — Actual burst length where β_3 denotes the actual burst length factor.

In order to evaluate the effectiveness of CAC tuner, we introduce the following notions:

- K_{max} — maximum number of connections based on traffic descriptors assuming no overbooking.
- $K_{practical}$— maximum number of connections based on actual traffic parameters and assuming $r^a = \beta_2(1+\beta_4)r$. This is the best link utilization assuming optimal fixed overbooking.
- K_{ideal}— maximum number of connections based on actual traffic parameters and assuming $r^a = \beta_2(1+\beta_4/2)r$. This is the best link utilization assuming adaptive overbooking and ideal knowledge of the process.

During the simulation, we compute the average number of connections $\overline{K}$. Then the practical and ideal effectiveness of CAC adjustment can be assessed by analyzing the values of the two ratios: $\overline{K}/K_{practical} and \overline{K}/K_{ideal}$.

5.1 Simulation Examples

Example 1: PCR = 6.0 Mb/s, SCR = 0.15 Mb/s, Ton=0.00177 s, $\beta_1 = 1, \beta_2 = 0.5, \beta_3 = 0.5$, $\beta_4 = 0.1$.
The buffer size is 1000 cells and port bandwidth is 45 Mb/sec. QoS Requirement is $CLR = 10^{-9}$. The CAC Tuner parameter are $\delta_{down} = 0.1, \delta_{up} = 0.1/25$ which corresponds to encountering losses once every 25 days on average. The number of iteration in a simulation run is 10000.

The gain in bandwidth utilization under learning algorithm is expressed by average overbooking factor $\overline{\alpha}^t = 2.23$. Concerning QOS, the actual CLR is shown in Figure 5. The instantaneous CLR exceeds the CLR constraint in 3.7300% of days which is close to the expected value of 4%. However, the average CLR in the simulation is $1.91537 * 10^{-10}$, which is quite below the CLR requirement. Obviously by changing the algorithm steps one can increase or decrease the level of CLR. The effectiveness of the learning algorithm in the practical sense is 0.92 while the effectiveness in the ideal sense is 0.86.

Example 2. In this example, we increase the day-to-day variation range of allowed overbooking from 10% to 50% ($\beta_4 = 0.5$). All other parameters are the same as in Example 1. As could be expected the gain in bandwidth utilization is smaller but still significant, $\overline{\alpha}^t = 1.75$. The actual CLR is shown in Figure 6. Once again the average CLR is below the constraint, $\overline{CLR} = 1.8 * 10^{-10}$, The effectiveness of the learning algorithm in practical sense is 0.98 while the effectiveness in the ideal sense is 0.69.

Example 3. In this example, we study effectiveness of the learning algorithm as a function of day-to-day variation range. The results, presented in Figure 7, indicate that the algorithm is robust in the whole range.

6 Summary

In the paper we addressed the issue of CAC adjustment using a simple set of measurements and overbooking control "knobs". The proposed *learning algorithm* is fed by counts of cells lost on the controlled links during say 15 minute intervals. Based on this information the overbooking "knobs" are updated once a day. The algorithm adapts to changing environment and insures that the long term QoS constraints are met. A preliminary study of benefits and performance showed that the algorithm can provide a significant gain in bandwidth utilization. A full version of the algorithm will be tested under more realistic non-stationary conditions.

References

[1] M4 Interface Requirements and Logical MIB: ATM Network Element View. STR-NM-M4NE-REQ-02.00, July 1998.

[2] Gibbens, R.J., Kelly F.P., and Key, P.B. 1995. A decision-theoretic approach to call admission control in ATM networks. *IEEE Journal on Selected Areas in Communication* 13(6):1101–1114.

[3] Dziong, Z. 1997. *ATM Network Resource Management*, McGraw-Hill.

[4] Dziong Z. and Jeske D. 1998. Private communication. Lucent, Holmdel, July.

[5] H. Ji, S. Dravida and R. Nagarajan, 1998. Framework for designing ATM performance management operating support systems, unpublished notes.

[6] Z. Zhang and H. Ji, 1998. Cell loss ratio distribution among on-off sources, Submitted for publication.

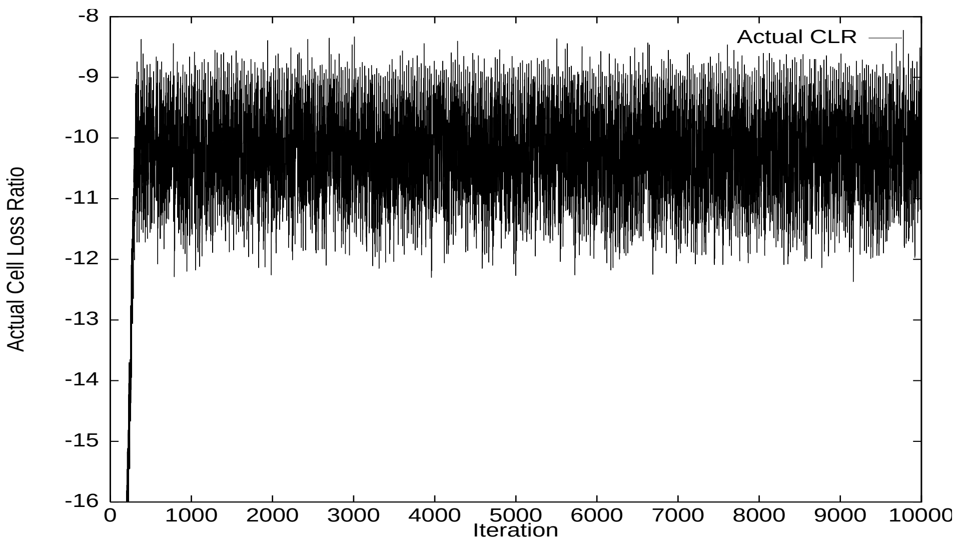

Figure 5: Actual Cell Loss Ratio for 10% day-to-day variation range.

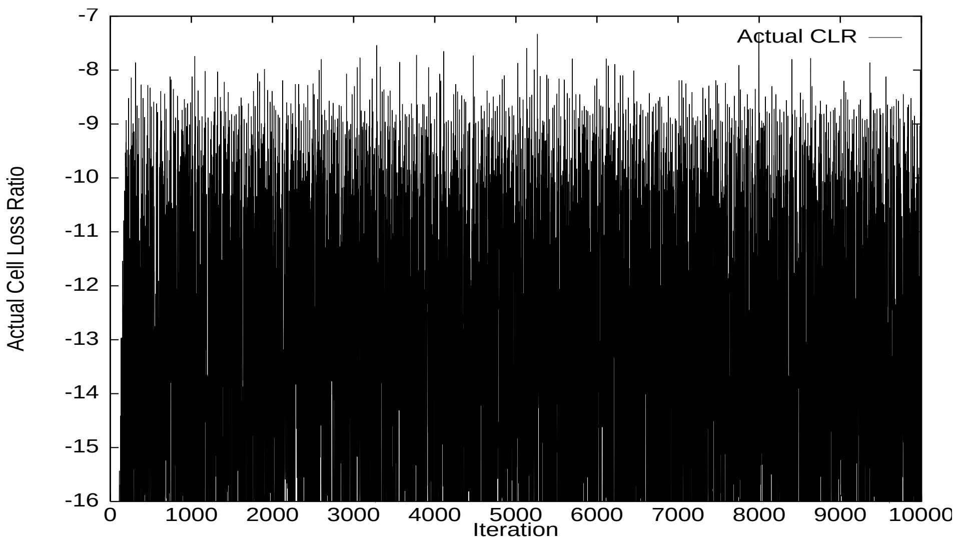

Figure 6: Actual Cell Loss Ratio for 50% day-to-day variation range.

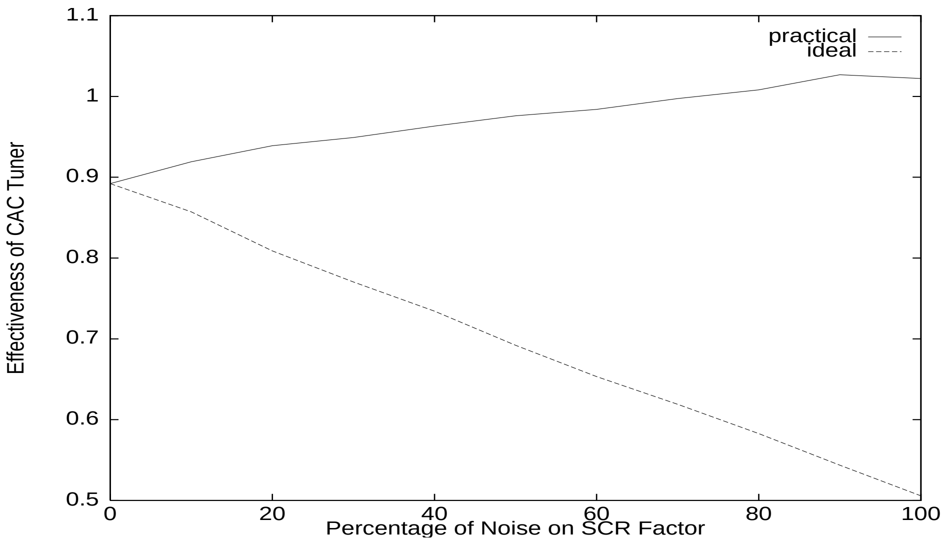

Figure 7: Effectiveness of the learning algorithm vs. day-to-day variation range.