Real Time Connection Admission Control with Multistate Traffic Sources

Robert Engel, Jonathan S. Turner

Abstract

This paper develops a practical method for connection admission control in ATM networks. The method is based on a virtual cell loss probability criterion, is designed to handle heterogeneous traffic types and allows each traffic source to be described by an individual finite-state model with as many states as are needed to describe the source traffic. To make connection admission decisions with respect to individual links, an aggregate finite state model is computed from the individual models and used to estimate the virtual cell loss probabilities. To reduce the computational requirements for maintaining the aggregate traffic model, the aggregate model uses quantized data rates and is maintained incrementally using direct numerical convolution. The approximations required by the quantization process can be done in a strictly conservative way.

1.0 Introduction

Asynchronous Transfer Mode (ATM) networks allow multiplexing all types of traffic together on the same link. A crucial part for the success of these networks is, that promised guarantees to each connection can be kept while utilizing the link as efficiently as possible. Connection Admission Control (CAC) plays an important part in this traffic management. It has to decide whether a new connection can be multiplexed on a link without violating the traffic contract of already existing connections or the new one. While different Quality of Service (QoS) parameters can be used for a CAC algorithm we think that the cell loss probability is the most important.

In our approach we use the Virtual Cell Loss Probability [3] as our CAC parameter. The aggregate bandwidth is calculated by convolving [2] the declared traffic rates of the multiplexed sources. As convolution can be time consuming we quantize the link rate in a fixed number of bins. Instead of using a two state traffic source model characterized by average and maximum cell rate (as e.g. in [1],[4]), we use a general, multistate traffic source model because the traffic of many applications is too complicated to be described precisely enough with a two state (on-off) model. Since the number of states of a traffic source is critical to the processing time we show how any many-state source can be approximated with a reduced set of states. We prove that such an approximation is strictly conservative and we give an estimator to make the approximation as tight as possible. We use world wide web traffic and video traffic data to evaluate our CAC algorithm. These results show that it is very fast (<1ms per connection add/drop) and that the maximum number of accepted connections is very close to the theoretical limit.

Although many CAC algorithms based on the cell loss rate already exist, they all consider the maximum and average rate and therefore the worst-case and most bursty two state model of a traffic. In a conventional approach like [5],[6] the buffer overflow probability is calculated. The calculation is based on an assumed arrival model and burst length distribution, while our approach is independent of any traffic model and is independent of the burst size. The equivalent capacity method [7],[8] consists in determining a capacity with which a connection can be represented. While this approach is precise for few connections, it uses a stationary approach, based on a two state model characterized with mean and average rates if there are many connections. Especially in this case we think that our approach is more precise. Dynamic CACs [9],[10] measure the effective traffic and allow new connections to be multiplexed if the measured traffic is not too big. However, if a connection sends less traffic than declared, additional connections will be multiplexed. When the connection goes back to its full rate, extensive cell loss is the result. Recently, neural networks [11],[12] have been proposed for acceptance control. If all connections can be described with a fixed set of classes then the neural network can be trained properly. However, it is unknown how to train a neural network, so that it can perform CAC when connections are described with a set of parameters that can take any value.

2.0 Incremental Link Rate Distribution Algorithm

Let us assume every traffic is described as a list of pairs:

$$(\lambda_1, p_1)\ ;\ (\lambda_2, p_2)\ ;\ (\lambda_3, p_3)\ ;\ \dots\ ;\ (\lambda_r, p_r) \qquad \text{where } 0 \le p_i \le 1 \text{ and } \sum_{i=1}^{r} p_i = 1$$

This list is interpreted to mean that the connection with probability p_i carries data at the normalized rate λ_i in this state. We assume that the λ_i are ordered in increasing size.

When several connections are multiplexed on a link, for every possible combination of the rates of the connections a probability can be calculated. We use an incremental algorithm to calculate the aggregate probability distribution. It is a generalization of the 2-state model used in [1] to a multistate model. It is based on the simplification that all rates of an individual traffic descriptor are a multiple of some basic rate (e.g. 1% of the link rate). We show in section 4.0 how a general traffic descriptor can be simplified. In (1) and (2) we show how to update the aggregate offered traffic descriptor when a new *r*-state connection is added or dropped.

$$C^*[i] = \sum_{k=1}^{r} C[i - \lambda_k]\, p_k \qquad C[i]=0 \text{ if } i<0 \tag{1}$$

$$C[i] = \frac{C^*[i + \lambda_1] - \sum_{k=2}^{r} C[i + \lambda_1 - \lambda_k]\, p_k}{p_1} \qquad C[i]=0 \text{ if } i<0 \tag{2}$$

$C[i]$ is an aggregate descriptor coefficient without the new connection and $C^*[i]$ with the connection. It represents the probability that the offered aggregate rate is *i*. If $C[i]$ are represented with an array we simply calculate the new values starting at the top of the array (high *i*) for (1) and at the low end of the array for (2).

3.0 Virtual Cell Loss Algorithm

To predict the cell loss of the aggregate traffic descriptor and therefore the QoS we use a logically bufferless fluid flow model [3] that measures the virtual cell loss probability which provides an upper bound for the real cell loss probability. The virtual cell loss is a good estimate for the actual cell loss when peak virtual circuit rates are small relative to the link rate or when the actual buffer size is comparable in size to typical data bursts or smaller. The virtual cell loss probability is defined in (3).

$$P_v = \frac{overflowBandwidth}{offeredBandwidth} = \frac{\sum_{i>R} C[i]\,(i-R)}{\sum_{i=1}^{\infty} C[i]\, i} \tag{3}$$

Here, *R* stands for the link rate. As we can see, direct computation from this equation is time-consuming since the number of terms $C[i]$ for which $i>R$ grows with the number of connections that are multiplexed on the link. Some simple mathematical transformations yield.

$$P_v = \frac{\sum_{i>R} C[i]\, i - \sum_{i>R} C[i]\, R}{\sum_{i=1}^{\infty} C[i]\, i} = \frac{\mu - \sum_{i\le R} C[i]\, i - R\left(1 - \sum_{i \le R} C[i]\right)}{\mu} \qquad \text{where} \quad \mu = \sum_{allConnections} \mu_{connection} \tag{4}$$

Here μ stands for the offered average traffic rate, which is the sum of the average traffic rates of all connections.

The total processing time when connections are added/dropped is therefore the time needed to update the parameter (adding and dropping) plus the time needed to calculate the virtual cell loss probability (only adding). If we limit the maximum number of states of a connection the CAC algorithm runs in constant time since the number of $C[i]$

between 0 and R is also constant. In section 4.0 we show how to reduce this number of states while still guaranteeing a conservative and precise traffic description.

4.0 Approximating Traffic Descriptors With Reduced State Models

The calculation time of our CAC algorithm depends on the number of states of the individual traffic descriptors and the number of bins with which the aggregate traffic is represented. In the following we show how we can transform a general r+1-state traffic descriptor into a reduced, a+1-state traffic descriptor that is guaranteed not to underestimate the original descriptor. Underestimating here means that the calculated cell loss probability of any aggregate traffic distribution together with the approximated descriptor may not be smaller than the same aggregate traffic together with the original descriptor. Thus, we have an upper bound on the original descriptor. We also show an estimator, that allows us to tighten this bound.

The same approach can then be used to quantize the reduced state traffic descriptor to the given bin sizes.

4.1 Conservative Approximation

Let us consider the following traffic that we want to approximate. For the sake of simplicity and without losing any generality we assume that for every rate in the range [0..r] a probability is associated (of course some of these probabilities can be equal to zero). We want to approximate this original descriptor $(0, p_0)$, $(1, p_1)$, $(2, p_2)$, ... , (r, p_r) with (σ_0, q_0), (σ_1, q_1), (σ_2, q_2), ... , (σ_a, q_a). We use the following strategy as illustrated in Figure 1. Let us divide the range of bandwidth of the real descriptor into a regions, ranging from rate 1 to r. Each rate/probability of the approximated descriptor approximates a region of the original descriptor so that the approximated rate is equal to the top rate in the region and the approximated probability is chosen so that the average original bandwidth of the region is equal to the approximated average bandwidth. A region s covering the range of bandwidth $(k, m\,]$ is therefore approximated:

$$\sigma_s = m \tag{5}$$

$$q_s = \frac{\sum_{j > k}^{m} j p_j}{\sigma_i} \text{ if (s>0)}, \quad q_0 = 1 - \sum_{j=1}^{a} q_j \tag{6}$$

First of all, note that the average bandwidth of the approximated and the original traffic descriptor are the same ($\mu_a = \mu_r$), and that the sum of all approximated probabilities is one.

Let us now prove that when we multiplex the approximated descriptor with any aggregate descriptor we are guaranteed that the cell loss probability (or the bandwidth overflow) is at least as big as when we do the same with the original descriptor. Let us suppose the aggregate traffic without the new descriptor is given by the coefficients $C[i]$. From (4) and (1) we have the difference between approximated and real cell loss probability which must be non negative.

$$\hat{P}_v - P_v = \frac{\sum_{l > R} \sum_{i=0}^{a} plus(l + \sigma_i - R)\, C[l]\, q_i}{\mu + \mu_a} - \frac{\sum_{l > R} \sum_{i=0}^{r} plus(l + i - R)\, C[l]\, p_i}{\mu + \mu_r} \tag{7}$$

$plus(i) = i$ if i>0 and 0 otherwise. To prove that (7) is greater or equal to zero we first note that the denominators are equal. Then we split up the sum in terms for the different regions and prove that every tuple (σ_i, p_i) is an overestimation of the region it approximates, independent of $C[i]$.

$$\left(\sum_{i=0}^{a} plus(l + \sigma_i - R)\, C[l]\, q_i\right) - \left(\sum_{i=0}^{r} plus(l + i - R)\, C[l]\, p_i\right) \tag{8}$$

Let us drop the coefficient $C[l]$, substitute $j = R - l$ and split up the two sums in the regions we used for the approximation. We prove that for every value j and for every region s the approximation error is non-negative and therefore the total approximation error must be, too.

$$E_s^j = \left(\sum_{i=0}^{a} plus(\sigma_i - j)\, q_i\right) - \left(\sum_{i=0}^{r} plus(i - j)\, p_i\right) \quad \text{should be non-negative for all } s \text{ and } j \tag{9}$$

We see from (10) that we need to consider only the case where $j > 0$ as in the other case the difference is exactly 0.

$$E_s^j = \sum_{i=0}^{a} (-j)\, q_i + \sum_{i=0}^{a} \sigma_i q_i - \left(\sum_{i=0}^{r} (-j)\, p_i + \sum_{i=0}^{r} i p_i\right) = (-j + \mu_a - (-j + \mu_r)) = 0 \quad \text{for } j \le 0 \tag{10}$$

There are three cases to consider: Either j is to the right, to the left or within the region. The following equations show the three cases.

$$E_s^j = 0 \quad \text{for } j > m \tag{11}$$

$$E_s^j = \mu_{as} - j q_s - \mu_{rs} + \sum_{i>k}^{m} j p_i = j \sum_{i>k}^{l} p_i \left(1 - \frac{i}{\sigma_s}\right) \ge 0 \quad \text{for } j \le k \tag{12}$$

$$E_s^j = \sum_{i>k}^{j} i p_i + j\left(\sum_{i>j}^{m} p_i - \sum_{i>k}^{m} \frac{i p_i}{\sigma_s}\right) = \sum_{i>k}^{j} i p_i \left(1 - \frac{j}{\sigma_s}\right) + j\left(\sum_{i>j}^{m} p_i \left(1 - \frac{i}{\sigma_s}\right)\right) \ge 0 \quad \text{for } k < j \le m \tag{13}$$

Since the approximated bandwidth in a region is always the maximum bandwidth of the region our strategy is clearly strictly conservative.

Considering (12) and (13) we see that we can improve our strategy while still guaranteeing a completely conservative approximation. The new strategy is the same for the top region. Seen from a lower region the top region approximation clearly overestimates the probability distribution (the greater the term in (12) the larger the overestimation). In a lower region we can therefore compensate for this overestimation by choosing the approximation bandwidth smaller than the top of the region. We choose it as small as possible while still guaranteeing that the total approximation error represented by one term from (12) and possibly several terms from (13) is still positive. Figure 1 shows a real traffic descriptor (black dots) of which each region is approximated with a single rate/probability at the top of the region (white square) and the shift to the left of the approximated descriptors when the overestimation is compensated. The lowest region (with rate zero) is not considered as a regular region. It is only used to ensure that the sum of the probabilities is one.

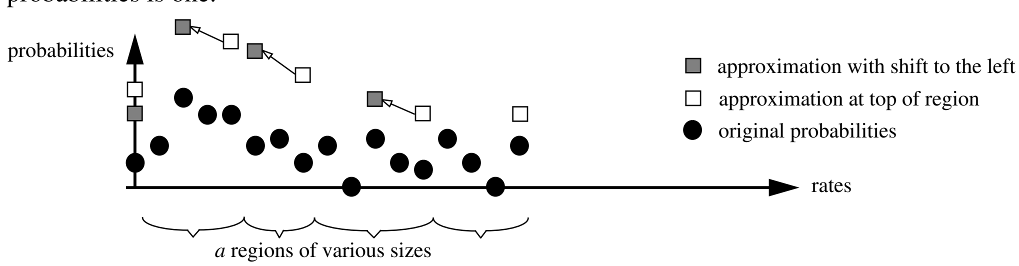

FIGURE 1. Approximating a real traffic descriptor by choosing a rate/probability for each region

4.2 Determining Optimal Regions

The preceding section does not give any hint how the regions have to be chosen in order to have an optimal approximation. We define the following estimator E of the quality of the approximation

$$E = \sum_{j>0}^{r} \sum_{s=1}^{a} E_s^j \tag{14}$$

where E_s^j is defined by (11), (12), and (13).

E is in so far only an estimator for the quality and not a correct value as we sum up over all the possible values of *j*. As soon as we have different values of *j* we should also consider the different values of $C[l]$, which are coupled with *j*. We can not do that, though, because we do not know the distribution of $C[l]$. In our estimator therefore we make the assumption that all the $C[l]$ are equal.

In fact the distribution of the $C[l]$ is generally even more favorable than that. Normally we operate in a range of $C[l]$ where for $m<n$ and $C[m]>0$ and $C[n]>0$, $C[m]>C[n]$. In addition to that, $C[l]=0$ if $l>maxRate$ where *maxRate* is the sum of the largest rates of all descriptors. These two general properties of the distribution mean that the approximation error for high values of *j* (small values of *l*) are weighted more strongly. Since for high values of *j* only the upper regions contribute to an approximation error and we tend to overestimate the upper regions and underestimate the lower regions, we are on the safe side.

The finding of optimal regions consists in calculating *E* for all possible region combinations. The lowest value of *E* represents the best region distribution. Of course this calculation has to be done off-line, when the traffic descriptor for a certain application is determined. The CAC can then directly be fed with the approximate traffic descriptor.

4.3 Relaxed Approximation

We can use the estimator of the previous section for an even tighter approximation of a bandwidth descriptor provided our assumption about the distribution of the *C[l]* is correct.

Instead of guaranteeing that the approximation error is positive for every single value of *j* we relax (9) as follows

$$E_s^{j+} = \sum_{i=j}^{r} \sum_{k=s}^{a} E_s^j \geq 0 \quad \forall j \ \forall s \tag{15}$$

Intuitively we compensate for the underestimation of a current region and a certain value of *j* with the overestimation of a higher region. Again this is based on the assumption that due to the distribution of $C[l]$ the higher region's (over)estimation error has a larger weight than the lower region's (under)estimation error.

To find out the optimum region distribution we have to calculate this approximation error for all possible region sizes. The region with the lowest approximation error which is still greater or equal than zero is to be picked. In Figure 2 we see how big the difference in terms of cell loss probability can be between the different strategies of traffic approximation. Note that even with the conservative four state model, the number of multiplexed sources for which the virtual cell loss probability is 10^{-6} is only about 3% less than the number of sources computed with the 11-state model.

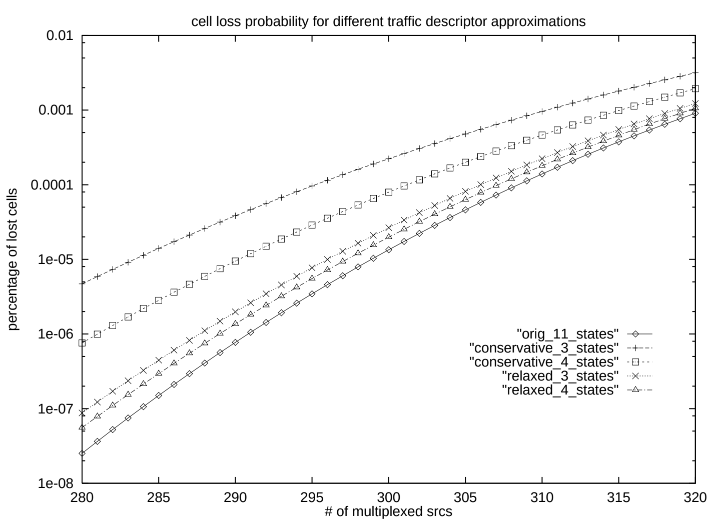

FIGURE 2. Approximating an 11-state traffic descriptor {(0,0.1) (1,0.2) (2,0.3) (3,0.09) (4,0.08) (5,0.07) (6,0.06) (7,0.04) (8,0.03) (9,0.02) (10,0.01)} with different strategies. Link rate = 1000

4.4 Quantization

While the reduction of states is done off-line, the quantization of the rates has to be performed by the CAC. Either of the previous methods can be used. Since the goal is not to reduce the number of states we only have to decide if a rate σ_i has to be quantized into the lower or the upper bin, depending on how $\sigma_{i+1}, \ldots, \sigma_a$ are quantized.

A special case is where several rates are so close together that they are quantized into the same bin. In this case the number of states is reduced by the quantization. If all rates are within the smallest bin we either represent the traffic as an on off traffic with the lowest peak rate possible or as a constant bit rate traffic with the rate being the mean rate of the actual descriptor plus a compensation/security factor (e.g. 10%). Though the 'burstiness' in this case is not correct the rate is so small compared to the link rate, that we have to multiplex a lot of these small rate traffic descriptors in order for them to use up a significant portion of the link. And this large number of connections (although they might be very bursty in this small bandwidth range) reduces the error of the mean rate representation significantly. Intuitively, this is very clear: An on-off connection with an on rate of 10 Megabit/s is very bursty for a 100 Megabit/s link and should be represented accurately. However if the same connection is multiplexed on a Gigabit/s link a representation with the mean rate is fully sufficient.

5.0 Applying our CAC to Real Traffic Descriptors.

A lot of CAC algorithms might work well in theory, but with real traffic examples their performance suddenly drops. In this section we show how well our CAC performs on two real and very different examples of traffic:

The first traffic is generated by a JPEG video source[14]. We obtained a traffic descriptor by measuring the ATM traffic at the output of an ATM JPEG CODEC using a Q factor of 30. This gave us a 52-rate traffic descriptor with rates in the range from 0 to 21.6 Mbit/s and a mean rate of 11.3 Mbit/s.

The second traffic is the traffic we expect from an average world wide web user being connected to the www-server at Washington University. Making the assumption that every file has to be transmitted in 0.5 seconds while limiting the maximum rate to 10 Mb/s we came up with a typical world wide web user profile for this server. The profile contains 101 rates ranging from 0 to 10 Mb/s with an average of 60.2 Kb/s. More details can be found in [13].

In the following figures we show how the choice of the number of quantization bins and the number of states to which we reduce any many state traffic influences the accuracy of our CAC algorithm. For all cases we depict the real cell loss rate (called orig_without_quantification) and the real traffic modeled by an on off source approach, as it is done in most other CAC algorithms. In Figure 3 and Figure 5 we fix the number of quantization bins to 500 and vary the reduced set of rates of the approximated descriptor; in Figure 4 and Figure 6 we fix the number of reduced states of the approximated traffic to four, and vary the number of quantization bins, for homogeneous multiplexing of video traffic and www traffic, respectively. In Figure 7 we multiplex video and www traffic together, starting with 1 video connection, then 19 www connections and so on. Again note that our approach is far more precise (with any set of parameters) than the classical on off approach.

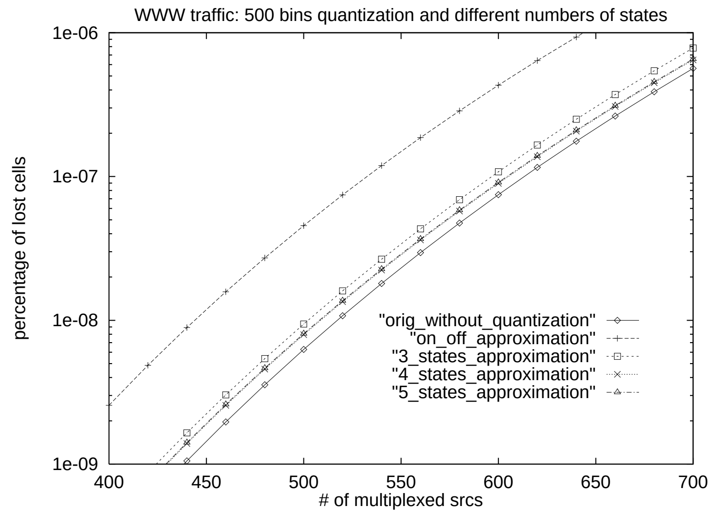

FIGURE 3. Cell loss rates for www traffic multiplexing. Link rate=150Mb/s, the approximations use 500 bins quantization and different numbers of states

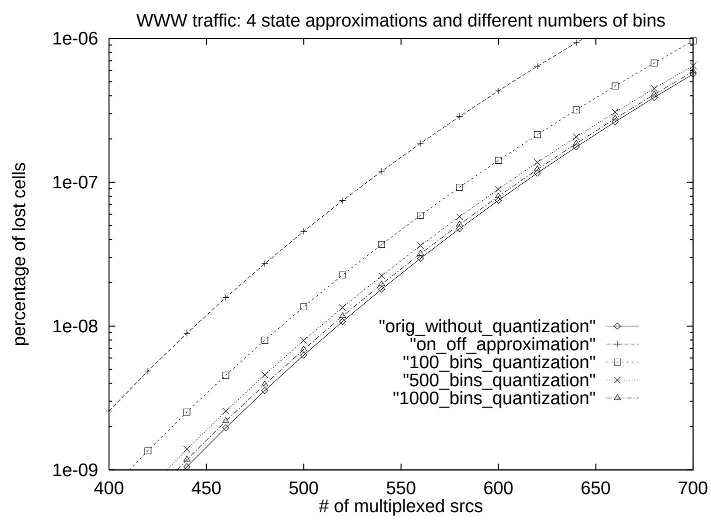

FIGURE 4. Cell loss rates for www traffic multiplexing. Link rate=150Mb/s, the approximations use 4 states and different numbers of quantization bins

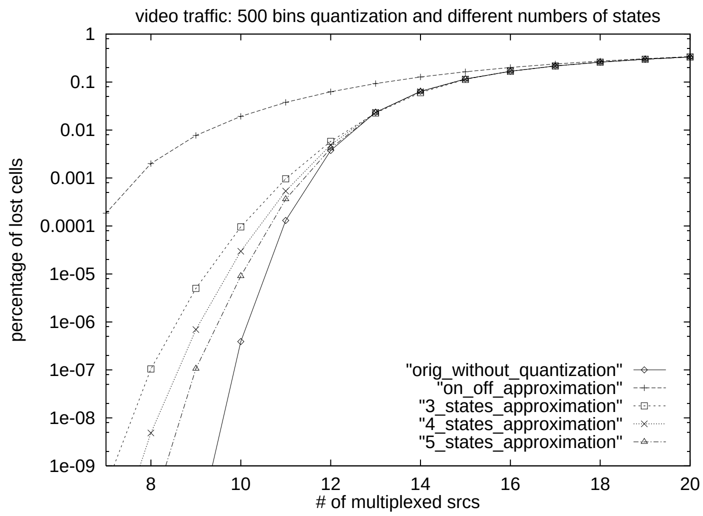

FIGURE 5. Cell loss rates for video traffic multiplexing. Link rate=150Mb/s, the approximations use 500 quantization bins and 4 states

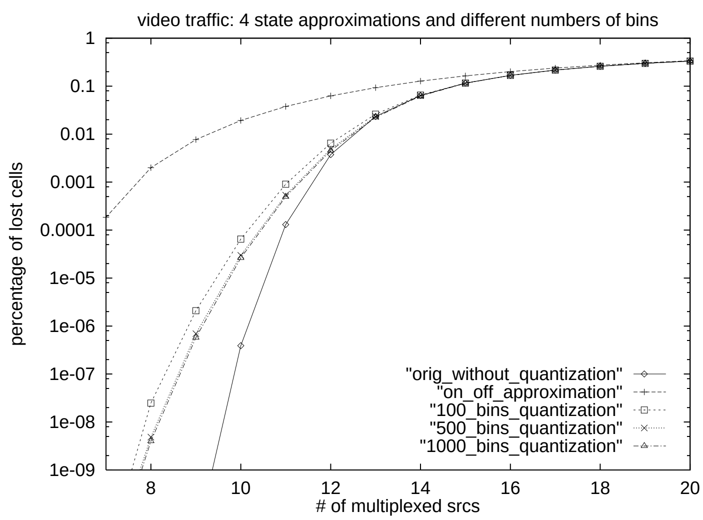

FIGURE 6. Cell loss rates for video traffic multiplexing. Link rate=150Mb/s, the approximations use 4 states and different numbers of quantization bins

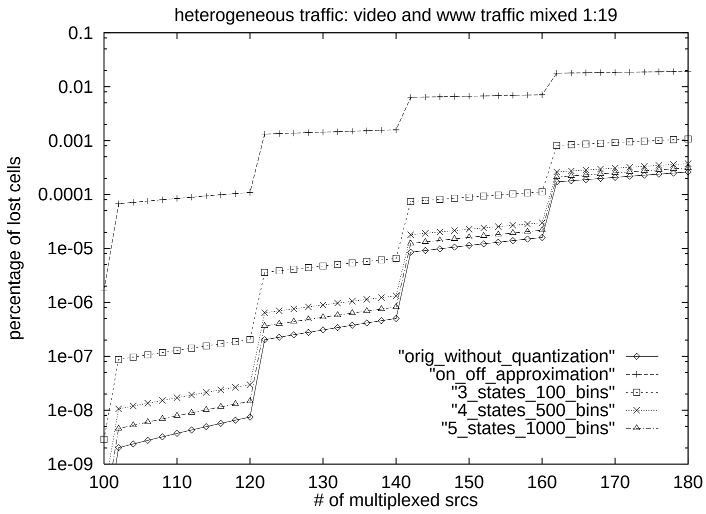

FIGURE 7. Cell loss rates for heterogeneous traffic multiplexing: Video and www traffic are multiplexed so that every 20th connection is a video connection.

6.0 Conclusions

The cell loss rate is one of the most important parameters in a CAC. We have shown that the aggregate bandwidth can be described with a compact model whose coefficients are adapted incrementally when connections are added or dropped. A general multistate source traffic model allows us to calculate this aggregate bandwidth distribution better than current models which consider only the mean and the peak rate. Our traffic model does not need classes and does not make any assumptions with respect to the holding times in various states.

The time to calculate whether a new connection can be accepted is independent of the number of multiplexed connections. It depends only on the number of probability coefficients we use and the number of states of a traffic source. Since traffic sources can potentially have a large number of states we have shown an algorithm to approximate them with a reduced state model that is guaranteed to be conservative while still being sufficiently accurate.

7.0 References

[1] Jonathan S. Turner, “Managing Bandwidth in ATM Networks with Bursty Traffic”, IEEE Network, September 1992

[2] Hiroshi Saito, Kohei Shiomoto “Call Admission Control in an ATM Network Using Upper Bound of Cell Loss Probability”, IEEE Transactions on Communications, Vol. 40, No. 9, September 1992.

[3] Totomu Murase et alt., “A Call Admission Control Scheme for ATM Networks Using a Simple Quality Estimate”, IEEE Journal on Selected Areas In Communications, Vol. 9, No 9, December 1991.

[4] Tsern-Huei Lee et alt., “Real Time Call Admission Control for ATM Networks with Heterogeneous Bursty Traffic”, IEEE Supercomm/ICC Proceedings, 1994.

[5] Tao Yang, Danny H. K. Tsang, “A Novel Approach to EStimating the Cell Loss Probability in an ATM Multiplexer Loaded with Homogeneous On-Off Sources”, IEEE Transactions on Communications, Vol. 43, No. 1, January 1995.

[6] Jimmy H. S. Chan, Danny H. K. Tsang, “Bandwidth Allocation of Multiple QOS Classes in ATM Environment”, IEEE Infocom Proceedings, 1994.

[7] Zbigniew Dziong, Boris Shukhman, Lorne G. Mason, "Estimation of Aggregate Effective Bandwidth for Traffic Admission in ATM Networks", IEEE Infocom Proceedings, 1995

[8] Roch Guerin, Hamid Ahmadi, Mahmoud Naghshineh, "Equivalent Capacity and Its Application to Bandwidth Allocation in High-speed Networks", IEEE Journal on Selected Areas in Communications, Vol. 9, No 7, September 1991.

[9] Sugih Jamin, Peter B. Danzig, Scott Shenker, Lixia Zhang, "A Measurement-Based Admission Control Algorithm for Integrated Services Packet Networks", IEEE/ACM Transactions on Networking, 1996.

[10] Hiroshi Saito, Kohei Shiomoto, "Dynamic Call Admission Control in ATM Networks", IEEE Journal on Selected Areas in Communications, Vol. 9, No. 7, September 1991.

[11] Ray-Guang Cheng, Chung-Ju Chang, "A Neural-Net Based Fuzzy Admission Controller for an ATM Network", IEEE Infocom Proceedings, 1996.

[12] Richard G. Ogier, Nina T. Plotkin, Irfan Khan, "Neural Network Methods with Traffic Descriptor Compression for Call Admission Control", IEEE Infocom Proceedings, 1996.

[13] Robert Engel, "Design and Implementation of a New Connection Admission Control Algorithm Using a Multistate Traffic Source Model", Master's Project, Washington University in St. Louis, 1996.

[14] Nilesh R. Gohel, "Medical Video Transmission Via Bandwidth Constrained ATM Networks", Master's Thesis, Computer Science Department Washington University in St. Louis, 1994.