Statistical Properties of a Near-Optimal Measurement-Based CAC Algorithm

Simon Crosby, Ian Leslie[1]
Brian McGurk, John T. Lewis, Raymond Russell, Fergal Toomey[2]

Abstract: Our algorithm, called Mosquito [3], allows sources to be ignorant of their statistics but offers near-optimal utilisation of the network. Our approach is based on Large Deviation Theory : the large deviation rate-function (entropy) of bursty ATM traffic can be estimated from measurements of traffic activity. The entropy can be used to determine the bandwidth requirement of the traffic. In this paper, we explain the basic ideas behind the algorithm and describe its implementation. We present some results of a statistical investigation of the performance of the Mosquito CAC algorithm comparing it with that of various modifications of the algorithm.

1 Introduction

The appeal of ATM is that it allows a wide range of services, with different traffic characteristics, to be statistically multiplexed onto a single bearer. Statistical multiplexing is attractive because it increases utilisation of network resources, but it can only be achieved at the cost of buffering traffic at switches and multiplexing points during periods of overload. Buffers must be sized to ensure that the queueing delay is not excessive; given the buffer size, the system must then ensure that the admission of new connections to the multiplex will not violate the Quality of Service (QoS) constraints of existing calls by increasing the probability of buffer overflow.

The problem of deciding whether or not to accept a new connection is called Connection Admission Control (CAC); it has received much attention in the literature [1, 2, 3, 4, 5, 6]. Perhaps the most difficult aspect of CAC is determining the *impact* of a new connection of a given type on the current multiplexed load. In recent years, the notion of the *effective bandwidth* of a traffic source has been developed to describe the effective resource requirement of a source. Although different definitions of the term exist, it has been widely used to express the trade-off between instantaneous bandwidth requirements and buffer space for a source (or collection of sources), given its traffic characteristics and QoS constraints [1, 7, 8, 9, 6].

The effective bandwidth of a source depends strongly on the statistical properties of its traffic as well as its QoS requirements. It is thus essential to have an accurate characterisation of each source which contains precisely the information required to compute its effective bandwidth. Two approaches to obtaining this information are possible. The first, which has been adopted to date by the majority of publications in the area, insists that the source should provide the network with a full statistical characterisation of its behaviour. Typically this is achieved by constructing and analysing a parametric model of the source, based on precise or statistical descriptions of its behaviour. Parameters of the model may be fitted from actual measurements of the traffic, made in isolation [10, 11, 12].

The second approach, adopted in this paper, argues that it is unreasonable to expect traffic sources to provide these parameters. Instead, we argue that the network is capable of deriving the traffic characteristics in a concise and efficient way. This approach, based on on-line measurements of traffic activity, was pioneered by Courcoubetis et al. [13]. Our work builds on recent theoretical developments in the field of large deviations [14, 15] and results from the practical application of this theory to real ATM traffic [16, 17, 18, 19]. Our theoretical developments have been made in the ESPRIT *Measure* project.

[1]University of Cambridge Computer Laboratory, New Museums Site, Pembroke Street, Cambridge CB2 3QG, UK; phone: +44 1223 334416, fax: +44 1223 334678; email: Simon.Crosby@cl.cam.ac.uk. The authors gratefully acknowledge the funding of the European Commission under the ESPRIT Measure project, LTR 20113

[2]Dublin Institute for Advanced Studies, 10 Burlington Road, Dublin 4, Ireland. phone: +353 1 6680748; fax: +353 1 6680561; email: toomey@stp.dias.ie

[3]The algorithms described in this paper are the subject of a current patent application

In order to justify our approach, let us consider what properties are required of a good traffic characterisation scheme. First, the traffic descriptors should contain *precisely* that information needed to describe the likely effect of the traffic on the network and no more. Second, the scheme should work both with forseeable traffic types such as voice, video and data, and also with unforeseeable future applications. Third, it is desirable that the scheme should not be overly complex or make use of descriptors that are difficult to determine in practice.

It is clear that parametric schemes based on source modelling do not satisfy any of these criteria. It would be impossible to submit each new video codec, network interface, protocol stack and application to a process of measurement and modelling in order to derive its traffic parameters. Moreover, the process of deriving these parameters would be time-consuming and complex, and the resulting traffic characterisation would contain excess information. It would also depend on the *content* of the data, because the effective bandwidth of compressed video depends on the degree of compression attained, which is content-specific. It is thus impossible for users to provide the network with anything other than a conservative bound on their transmission rates. If these bounds, typically formulated as a statement of the source peak rate and/or a set of leaky- or token-bucket parameters [20], are used for admission control, the resulting utilisation of the network may be unacceptably low [21].

It is imperative that we make an appropriate tradeoff between efficiency of resource usage in the network, and the cost of characterising the sources. Our approach imposes minimal requirements on the sources whilst permitting a high degree of statistical multiplexing gain in the network. These arguments have been recognised by several other authors, for example Jamin [22, 21], and Floyd [23]. Jamin et al., proposing a measurement based admission control algorithm for the IETF Predictive service, proposes substituting conservative delay bounds, computed from the worst-case traffic which could be transmitted through a user-parameterised token-bucket, with estimated bounds computed from simple measurements of the source activity. Floyd's approach [23] uses Hoeffding bounds to compute the equivalent capacity of the sources, viewed as a sum of random variables; based on this, she proposes a simple admission control procedure for the IETF Controlled Load service. The algorithm relies on policed source peak rates, together with an exponentially weighted moving average estimation of the source mean rates. Both approaches differ significantly from ours in two major respects: first, the IETF traffic classes at which they are aimed are likely to be tolerant of errors in the estimation procedures whereas we view loss and delay bounds as being hard; second, their estimation procedures themselves are significantly less sophisticated than ours. In contrast, we focus on accurate estimation procedures as the fudamental basis for measurement-based admission control. Consequently, our estimators may be useful in frameworks other than those which we describe in this paper, and indeed in those of the above authors. A significant body of literature on estimation for resource allocation is now emerging.

Consider a network carrying a multiplex of traffic connections, and receiving requests to set up new connections. These will have some simple user declared parameters associated with them (even if this is only the peak rate). Our CAC algorithm combines estimation of the bandwidth requirement of the existing load using on-line measurement with a prediction of the bandwidth requirement of the offered traffic based on its declared parameters. If the sum of these bandwidths exceeds the line rate, the offered traffic is rejected; otherwise, it is accepted. Both the estimators and the predictors we have devised make essential use of mathematical properties of the *entropy* of the relevant traffic.

In this paper, we concentrate on two versions of our CAC algorithm; both make use of the declared peak-rate as predictor (a more sophisticated predictor, making use of several declared parameters, will be described elsewhere).

2 Theoretical Framework

We first establish the terminology that we will use. Our central concern is the loss of cells due to overflow at a buffer. Consider an ATM traffic stream arriving at a buffer which has finite storage capacity b; here b may be sized according to the delay constraint of the traffic. Cells are removed from the buffer at fixed rate s, the *line-rate*. Each traffic stream has a finite duration, as might be expected for calls of finite length. We refer to a traffic stream as a *trace*. Associated with each trace is a *cell-loss ratio* between zero and one; we denote the cell-loss ratio for a buffer-size b and a line-rate s by $\mathrm{CLR}(b, s)$. Experience with a wide variety of traffic-sources shows that the the logarithm of $\mathrm{CLR}(b, s)$ is asymptotically linear

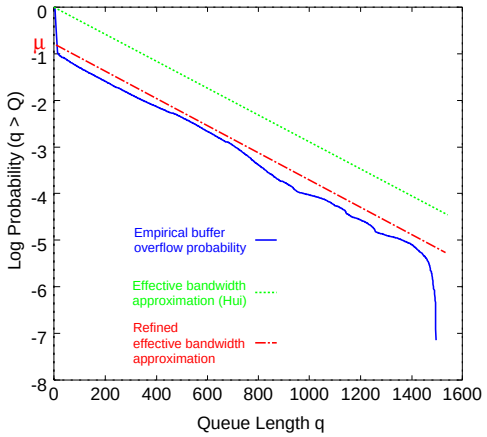

Figure 1: Empirical loss probability for 18 streams of JPEG coded video

in the buffer-size b for practical buffer sizes[4], and for fixed line-rate s. A typical example is shown in Figure 2 which plots, for a set of motion JPEG sources and fixed line-rate s on the Fairisle ATM network at Cambridge [26], the logarithm of the observed cell-loss ratio[5] against buffer-size b.

We define the *bandwidth requirement* of a trace to be the minimum line-rate at which a target cell-loss ratio c is not exceeded in a buffer of storage capacity b:

$$\mathrm{BWR}(b,c) := \min\{s : \mathrm{CLR}(b,s) \leq c\}. \tag{1}$$

Notice that this is an operational definition which does not involve any statistical theory; for a given trace, it can be determined empirically by trial and error. In practice of course it is neccessary to estimate the bandwidth requirement of a trace rapidly.

The general features of a plot of the logarithm of cell-loss ratio against buffer-size for fixed line-rate are explained by queuing-theory. The trace is identified with a portion of a sample-path of a stationary stochastic process $\{A_t\}$, the *arrivals process*; here A_t denotes the total number of cells which have arrived up to time t. It can be shown, using large deviation theory, that if the arrivals process has a rate-function $I(a)$ on the scale of t, such that

$$\mathbb{P}(A_t/t > a) \asymp \exp(-tI(a)),$$

then the cell-loss ratio in a buffer of size b with fixed line-rate s decays exponentially in b when b is large:

$$\mathrm{CLR}(b,s) \asymp \exp(-b\delta).$$

Furthermore, the decay-rate δ is determined by the rate-function $I(a)$:

$$\delta = \min\{I(a+s)/a : \ a > 0\}.$$

What we refer to as the *thermodynamic entropy* of a traffic stream is just the rate function $I(a)$. These theoretical results, valid for a wide class of stochastic processes, offer an explanation of the observed asymptotic behaviour of the logarithm of the cell-loss ratio as a function of buffer-size and relate the slope of its linear asymptote to the rate-function of a stochastic process representing the traffic.

The behaviour of the cell-loss ratio for small buffer sizes is less well understood, but theoretical studies of model stochastic processes have thrown some light on this. A simple two-state Markov model is capable of capturing the general features of ATM traffic in a crude way. For such models, Buffet and Duffield [27] showed that the large deviation estimate $\mathbb{P}(Q > q) \asymp \exp(-q\delta)$ can be improved by the introduction of a prefactor $e^{-\varphi}$ so that we have a bound valid for all values of q:

$$\mathbb{P}(Q > q) \leq \exp(-\varphi - q\delta).$$

[4]We do not intend to fuel the self-similarity debate [24, 25] further – we simply report our observations based on experience with real ATM traffic using practical buffer sizes and time-scales.

[5]Observe that the sharp drop-off at the tail of the distribution is due to the finite observation period of the experiment.

This has motivated us to use a straight-line upper bound to the graph of the logarithm of the cell-loss ratio against buffer-size:

$$\log \mathrm{CLR}(b,s) \leq -\mu(s) - \delta(s)b, \tag{2}$$

at least for buffer sizes above a small threshold b_0. The parameters $\mu(s)$ and $\delta(s)$ are functions of the line-rate s. Suppose we have estimates $\hat{\mu}(s)$ and $\hat{\delta}(s)$ of the parameters; we define the *estimated bandwidth requirement* by

$$\widehat{\mathrm{BWR}}(b,c) := \min\{s : -\hat{\mu}(s) - \hat{\delta}(s)b \leq \log c\}. \tag{3}$$

This is used in two ways in our CAC algorithms: for the existing multiplexed traffic, we use on-line estimation of the parameters μ and δ to determine its estimated bandwidth requirement; for a proposed new connection, we set μ equal to zero and estimate δ using declared parameters.

The on-line estimation of the slope parameter δ for multiplexed traffic is the characteristic feature of the CAC algorithms introduced in this paper. The on-line estimation of the intercept parameter μ is comparatively straight-forward, and based on the frequency of occurrence of small interarrival times; we describe it in more detail in [18]. The estimation of δ is based on the formula

$$\delta(s) = \min\{I(a+s)/a : a \geq 0\},$$

relating it to the large deviation rate-function $I(a)$ of the arrivals process. This formula has an equivalent version which is more suited to estimation:

$$\delta(s) = \max\{\theta > 0 : \lambda(\theta) < s\theta\},$$

where $\lambda(\theta)$ is the *scaled cumulant generating function* (SCGF) of the arrivals process, defined by

$$\lambda(\theta) = \lim_{t\to\infty} (1/t) \log \mathbb{E}(\exp(\theta A_t)). \tag{4}$$

It is of crucial importance to observe that the SCGF is automatically convex.

A Note on Effective Bandwidth. Before describing the CAC algorithms, we briefly clarify our use of the term effective bandwidth, or equivalent capacity. The definition which most closely fits our approach originates with Hui [1], and is the value of s in equation (2), which describes a straight line of slope δ through the origin in the graph of the empirical CLR of Figure 2. This gives rise directly to the *effective bandwidth approximation*, which is the function

$$\hat{s}(b,c) = -b\frac{\lambda(-\log c/b)}{log(c)}. \tag{5}$$

The effective bandwidth approximation is an upper bound on the rate at which a buffer of size b must be drained in order to meet the target CLR constraint c. It does not include any refinement involving the intercept μ, shown in Figure 2. We have termed our refined approximation the estimated bandwidth requirement, as defined in equation (3). However, we shall make use of the effective bandwidth approximation of equation (5) in situations where we have no way to estimate μ, for example, when presented with a new call attempt for which we have no measurements.

3 The Measure CAC Algorithm

In this section we show how the the theoretical formulation of our approach can be implemented in practice. In all cases, parameters supplied by a new call request are passed to the predictor which attempts to predict the SCGF, and thence the bandwidth requirement, of the new call. This is combined with the estimated bandwidth requirement of the current multiplex, produced by the *estimator*, to decide whether or not to admit the call.

Figure 2 shows the behaviour of the *Measure* CAC algorithm. Given a current multiplex of calls, the system estimates the bandwidth requirement of the multiplex from the estimated SCGF using the entropy estimator (depicted here by a thermometer). The predictor used in the *Measure* algorithm requires a new

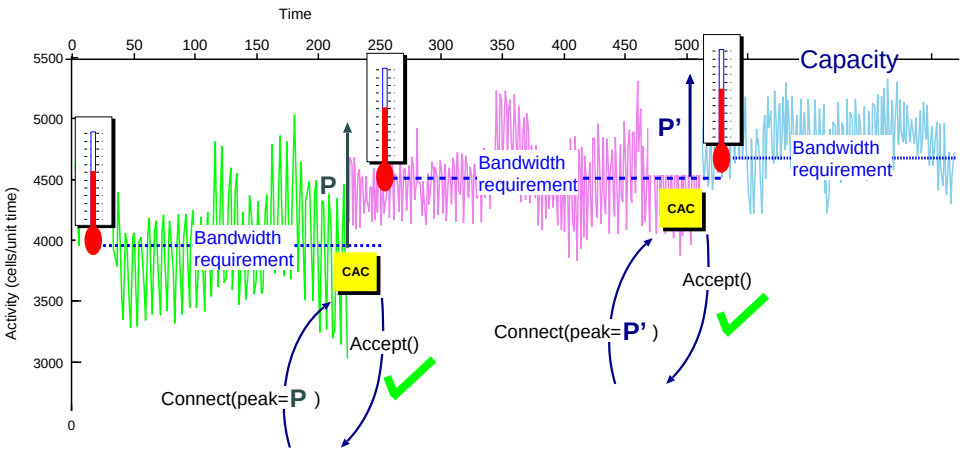

Figure 2: Operation of the Measure CAC algorithm

call attempt to declare only its peak rate P; this is used as a pessimistic assessment of the call's bandwidth requirement. The CAC algorithm sums the peak rate P, and the estimated bandwidth requirement of the multiplex; if the total is less than the link capacity, then the connection can be accepted without violating the QoS of any calls. As soon as the new call commences, the estimator uses measurements of the new multiplex to revise its estimate of the current bandwidth requirement. When the next call attempt arrives, the procedure is repeated, as shown. If a new call attempt arrives before the algorithm has developed an accurate estimate of the new bandwidth requirement, as shown in Figure 3, the algorithm acts *conservatively*. It uses the most recent stable estimate of the bandwidth requirement, plus the sum of the peak rates of all subsequently admitted calls. Thus, in Figure 3, the second call will be rejected, because the sum of the two peaks $P + P'$ and the first bandwidth requirement estimate exceeds the link capacity.

The *Measure* algorithm uses the following formulation as the estimator for the SCGF λ of the current multiplex:

$$\lambda(\theta) := \lim_{t\to\infty} \frac{1}{t} \log \mathbb{E}(e^{\theta A_t}), \tag{6}$$

An obvious way of estimating the SCGF from observations of A_t over a finite time interval T is to use empirical averages to estimate the expectation: we break the observations into K blocks of length B in time and let $\tilde{X}_k$ be the total arrivals in the k^{th} block. The SCGF is then readily estimated as follows:

$$\hat{\lambda}(\theta) := \frac{1}{B} \log \frac{1}{K} \sum_{k=1}^{K} e^{\theta \tilde{X}_k}. \tag{7}$$

The *Measure* CAC algorithm is simple, yet remarkably effective, as demonstrated by the results of Section 5.

4 The *Mosquito* Estimator and CAC Algorithm

The *Measure* estimator of equation (7) gives a very reliable estimate of λ for small values of θ, where it is dominated by the mean activity. For higher values of θ, the value of the SCGF is dominated by the probabilities of rare events; since these are inherently difficult to measure accurately, $\hat{\lambda}$ is less reliable. This can cause the *Measure* algorithm to admit more calls than it should.

If, however, we know the peak-rate p of a source, then we can improve our estimator $\hat{\lambda}$ by incorporating p into it. We call the revised formulation of the SCGF the *Mosquito* estimator. It is based upon two

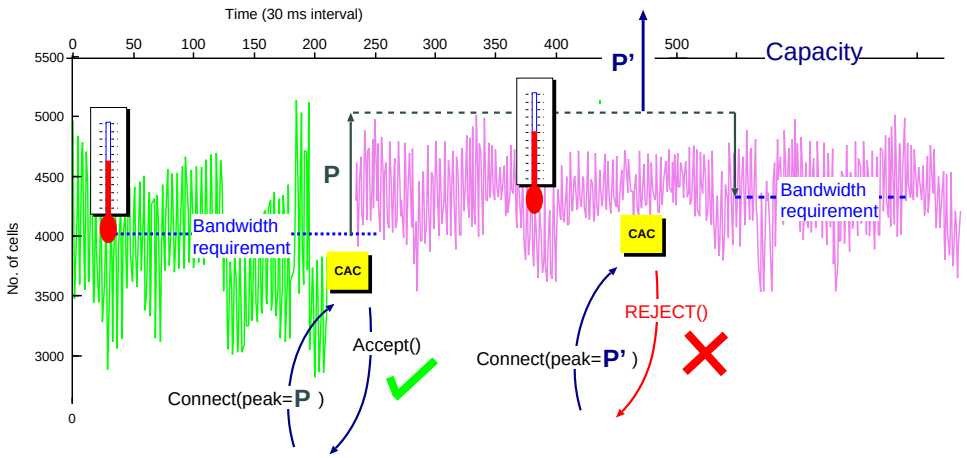

Figure 3: Conservative CAC using the Measure CAC algorithm

crucial observations, namely that the peak-rate p is the asymptotic slope of the true SCGF λ and that $\lambda(\theta)$ is convex. Thus if we take any point θ_0 and define a new function $\lambda^{'}$ by

$$\lambda^{'}(\theta) := \begin{cases} \lambda(\theta) & \theta \leq \theta_0 \\ \lambda(\theta_0) + p(\theta - \theta_0) & \theta > \theta_0, \end{cases}$$

then we get a convex upper bound on λ as shown in Figure 4. We can boost the tail of our estimate $\hat{\lambda}$ in the same way:

$$\lambda^{'}_{\text{mosquito}}(\theta) := \begin{cases} \hat{\lambda}(\theta) & \theta \leq \theta_0 \\ \hat{\lambda}(\theta_0) + p(\theta - \theta_0) & \theta > \theta_0, \end{cases}$$

It remains to choose an appropriate value of θ_0; it should be the maximum value of θ for which we have confidence in the estimate $\hat{\lambda}$ and as such should reflect the accuracy with which we can measure the probability of rare events. Since we are ultimately interested in applications to queuing systems, we can use a heuristic based on the single-server queue: applying the effective bandwidth approximation to a buffer of size b, what is the largest decay-rate δ that will still give us an observable cell-loss? The minimum observable cell-loss ratio over a trace of C cells is $1/C$ and, according to the effective bandwidth approximation, this is attained in a buffer of size b at a decay-rate δ given by

$$\delta = \frac{-\log C}{b}.$$

This value of δ can be used as the "cut-off" θ_0 above.

The *Mosquito* CAC algorithm uses the estimated bandwidth requirement of the multiplex from the *Mosquito* estimator, together with a pessimistic predictor which assumes that the bandwidth requirement of a new connection is its declared peak rate.

5 Simulation Results

This section presents the results of simulation experiments using the CAC algorithms of Sections 3 and 4. The aim of the experiments was to evaluate the performance of our approach with respect to several criteria which we discuss below.

Firstly, we are interested in the performance of each algorithm in terms of the resource utilisation it achieves. A *pessimistic* CAC algorithm which allocates resources using the declared peak rate of each source can guarantee that the loss constraints will always be met. Our algorithms, however, attempt

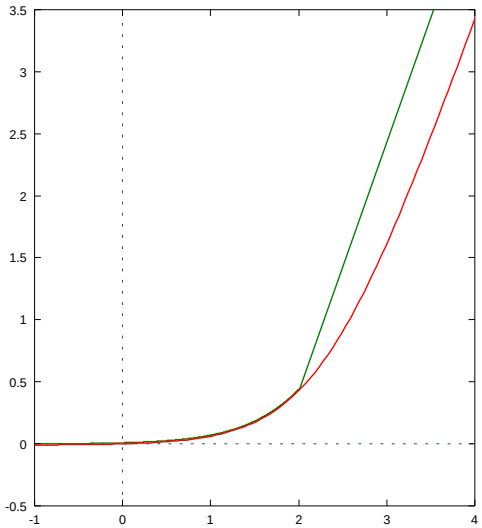

Figure 4: Convex upper bound on λ

to increase the link utilisation by admitting as many calls as possible whilst still maintaining the QoS guarantee. We thus need to compare our approach with the pessimistic system, and with a system which is *optimal* in the sense that the CAC algorithm is assumed to have complete knowledge of the statistical properties of every connection requesting admission. The optimal CAC should achieve maximal link utilisation while ensuring that QoS constraints are met. Secondly, we are interested in the effectiveness of each CAC algorithm in terms of its ability to guarantee the QoS constraints of the traffic.

Simulation Model. In each of our simulations we model a single output buffer and transmission link of an ATM switch. The link speed used is 100 Mb/s, which corresponds to the TAXI transmission rate for the Fairisle ATM network at Cambridge.

We present results for simulations using two traffic types. The first is a source which has been widely used in performance studies of ATM systems in the literature, namely a trace of the activity of the Star Wars movie. The Star Wars data set was produced by Garrett and Vetterli at Bellcore, and has been studied in some detail, for example in [24, 28] and [16]. It comprises the information content in bytes per frame for about 2 hours of the film, as transmitted by a DCT based codec similar to JPEG. The byte count per frame is broken down into "slices" of 16 video lines. With 30 slices per frame and 24 frames per second, one slice represents the information transmitted in about 1.4 milliseconds. The Star Wars traffic has been shown to exhibit signs of long-range dependent behaviour, making it potentially difficult for a measurement based CAC to cope with.

The Star Wars traces were constructed by encoding each slice as a single AAL5 PDU which was transmitted (and traced) over Fairisle; cells from each slice were transmitted at a CBR rate equal to the slice rate, reducing the peak rate from the line rate to about 24.2 Mb/s for the worst slice. The mean rate is about 5.3 Mb/s. The buffer is sized according to the requirement that no frame would be delayed more than a frame time; this results in a maximum buffer size of 500 cells. We used a CLR constraint of 10^{-4} for all of the results presented in this paper.

The second traffic type is a two-state Markovian source which emits cells at a constant (peak) rate of about 11.1 Mb/s (28.8Kcells/s) in the ON state, and is silent in the OFF state.

Call Model. We study a scenario in which calls of a particular traffic type arrive according to an exponential inter-arrival time distribution, an assumption which appears to be well founded [29]. In the absence of real-world data we have used call lengths which are exponentially distributed. We present results for *long* and *short* calls. In both cases calls arrive at a high (Poisson) rate with mean 5 calls/s. Blocked calls are lost, but the high arrival rate means that the system is continually faced with new call attempts. We thus expect the system to remain close to maximum utilisation.

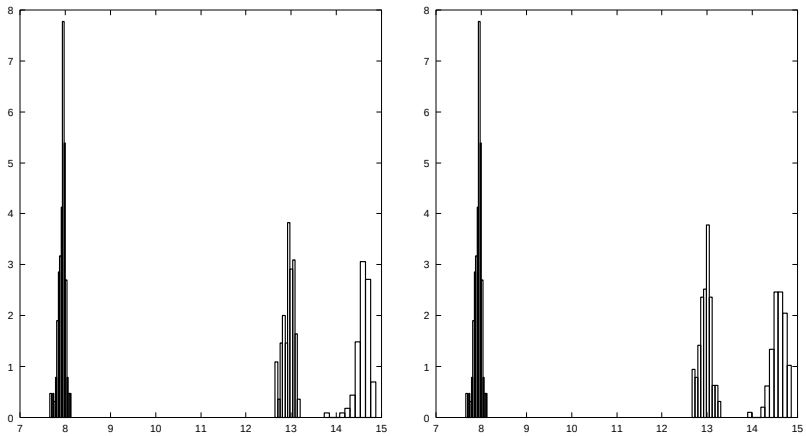

Figure 5: Histograms of the average number of calls in progress over a simulation run. The plot on the left refers to the Measure algorithm; that on the right, to Mosquito. The left-most histogram in each plot was made using peak-rate admission control; the central histogram was made using admission control based on the simple effective bandwidth approximation (in conjunction with a peak-rate bound in the case of Mosquito); and for the right-most histogram, admission control based on the refined effective bandwidth approximation was used. The average fraction of calls whose QoS requirement was violated was 0% for peak-rate admission, and approx. 5% for Measure and Mosquito.

Calls have an exponentially distributed length; long calls have a mean length of 60 seconds, and short calls a mean of 10 seconds. Each accepted call transmits an independent *trace*. In the case of Star Wars, the trace is derived by randomly selecting a start point in the traffic trace. For the ON-OFF source, each call chooses its parameters randomly: the mean (as a fraction of the peak) is chosen from 0.1, 0.3, 0.5, 0.7, and 0.9 with probabilities 1/16, 4/16, 6/16, 4/16, and 1/16 respectively. The burstiness is chosen from 0.3, 0.5, 0.7, and 0.9 with probabilities 1/8, 3/8, 3/8, and 1/8 respectively. Calls are not correlated in any way.

For the results which compare the performance of the different algorithms, the *same random seed* was used with each different algorithm, resulting in precisely the same call arrivals process in each case. This allows us to compare not only the statistical properties of the algorithms, but also their dynamic behaviour.

Results. Figure 5 shows histograms of the average number of connections in progress during a simulation run, for each of the CAC algorithms discussed. The plot on the left refers to the *Measure* algorithm, and that on the right to *Mosquito*. The left-most histogram in each plot was made using peak-rate admission control; this allows for an average of approximately just 8 connections in the system. The central histogram was made using the *Measure* (respectively, *Mosquito*) algorithm in conjunction with the simple effective bandwidth approximation. The advantage gained by exploiting statistical multiplexing is clearly apparent, as the average number of calls in progress is now approximately 13. The refined effective bandwdith approximation, which makes use of the intercept μ as described in section 2, was used to make the right-most histograms in each plot. This produces a clear improvement in performance, with the average number of calls in progress going up to about 14.5. The fraction of calls experiencing cell-loss ratios in excess of their requirement was in each case approximately 5%.

The performance of Measure and Mosquito may be similar, but their internal workings are not. This may be illustrated by considering the variability of the queue length under each algorithm. Figure ddist presents histograms of δ values recorded at regular intervals during a single simulation run, in which the average rate of connection requests was low. (Recall that δ is the decay parameter of the tail of the queue length distribution). There is significantly greater variability in the value of δ over time under the Measure algorithm; this is due to the fact that the tail of the SCGF is poorly estimated giving rise to poor estimates of δ at lower loads. Under the Mosquito algorithm, the estimate of the tail of the SCGF

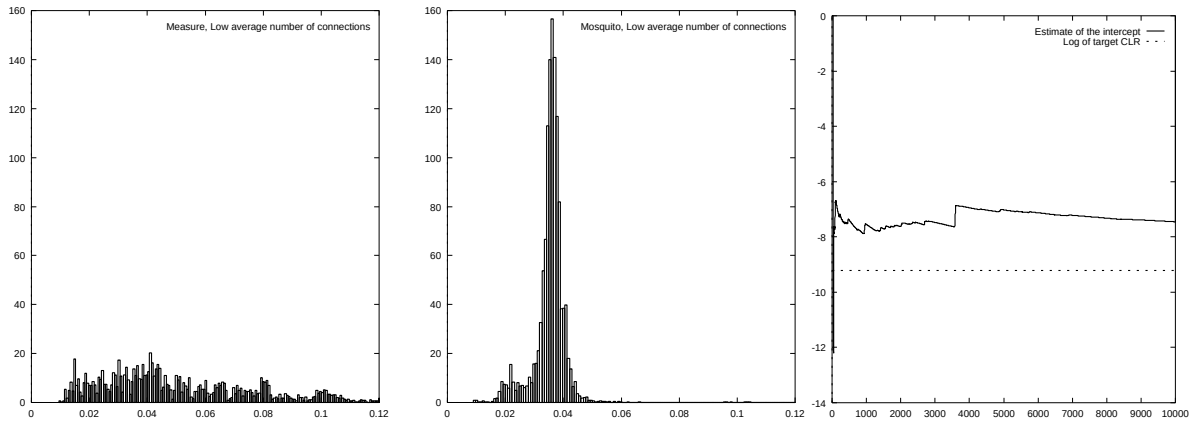

Figure 6: The two plots on the left are histograms of the δ values used by the Measure and Mosquito algorithms to estimate required bandwidth, recorded at regular intervals over a single simulation run. The plot on the right depicts the intercept estimate μ used in the refined effective bandwidth approximation, plotted against time over a single simulation run.

is replaced with a more solid piece of information, the declared peak-rate of the source, yielding stable δ estimates.

As the rate of connection requests is increased the distributions of δ values under the two algorithms become more similar, with the average value of δ decreasing in both cases. This is to be expected, for if the average number of connections is high, then the system is operating under a high load and the estimates of δ are being formed from the estimated SCGF near the origin, where the Measure and Mosquito algorithms use the same estimator.

Figure 5 also shows the value of the estimate of μ against time for one simulation run. It is seen that although there is initially a lot of variability, the estimate quickly settles down to a stable value. The estimates are made by measuring the cell-loss ratio in a buffer of size zero over the total current length of the simulation and this contributes to their stability. It is also possible to use a moving average estimate, formed from the cell-loss ratio over an interval of fixed length in the past; estimates made this way are found to be similarly stable over time, provided the measurement interval is long enough.

References

[1] Joseph Y. Hui. Resource allocation for broadband networks. *IEEE JSAC*, 6(9):1598–1608, December 1988.

[2] N.M. Mitrou and D.E. Pendarakis. Cell level statistical multiplexing in ATM networks: Analysis dimensioning and call acceptance control w.r.t QOS criteria. In *Broadband Technologies*, New Jersey, October 1990. Proceedings Seventh ITC Specialist Seminar.

[3] James Appleton. Modelling a connection acceptance strategy for ATM networks. In *Broadband Technologies*, New Jersey, October 1990. Proceedings Seventh ITC Specialist Seminar.

[4] R.J. Gibbens, F.P. Kelly, and P.B. Key. A decision theoretic approach to call admission control in ATM networks. *IEEE Journal on Selected Areas in Communications, Special issue on advances in the fundamentals of networking – part 1*, 13(6), August 1995.

[5] Peter Key. Connection admission control in ATM networks. *BT Technology Journal*, 13(3), July 1995.

[6] Frank P. Kelly. Notes on effective bandwidths. In Frank P. Kelly, S. Zachary, and I. B. Ziedins, editors, *Stochastic Networks: Theory and Applications*, volume 4 of *Royal Statistical Society Lecture Note Series*, pages 141–168. Oxford University Press, 1996.

[7] F.P. Kelly. Effective bandwidths at multi-class queues. *Queueing Systems*, 9:5–16, 1991.

[8] R. Guerin, H. Ahmadi, and Naghshineh. Equivalent capacity and its application to bandwidth allocation in high speed networks. *IEEE JSAC*, 9:968–981, 1991.

[9] R. J. Gibbons and P. J. Hunt. Effective bandwidths for the multi-type UAS channel. *Queueing Systems*, 9:17–28, 1991.

[10] A.I. Elwalid, D. Mitra, and T.E. Stern. Statistical multiplexing of Markov modulated sources: Theory and computational algorithms. In *Proceedings of the 13th International Teletraffic Congress*, Copenhagen, June 1991.

[11] B. Maglaris, D. Anastassiou, P. Sen, and J.D. Roberts. Performance models of statistical multiplexing in packet video communications. *IEEE Trans. Commun.*, 36:834–844, July 1988.

[12] M. Frater, J.F. Arnold, and P. Tan. A new statistical model for traffic generated by VBR coders for television in the broadband ISDN. *Proceedings IEEE CSVT*, December 1994.

[13] C. Courcoubetis, G. Kesidis, A. Ridder, J. Walrand, and R. Weber. Admission control and routing in ATM networks using inferences from measured buffer occupancy. Technical report, University of California, Berkley, EECS Department, UCB, California, CA94720, 1991.

[14] N. G. Duffield, J. T. Lewis, Neil O'Connell, Raymond Russell, and Fergal Toomey. The entropy of an arrivals process: a tool for estimating QoS parameters of ATM traffic. In *Proceedings of the 11th UK Teletraffic Symposium*, Cambridge, March 1994.

[15] N.G. Duffield, J.T. Lewis, N. O'Connell, R. Russell, and F. Toomey. Entropy of ATM traffic streams. *IEEE Journal on Selected Areas in Communications, Special issue on advances in the fundamentals of networking – part 1*, 13(6), August 1995.

[16] Simon Crosby, Ian Leslie, John Lewis, Neil O'Connell, Raymond Russell, and Fergal Toomey. Bypassing modelling: an investigation of entropy as a traffic descriptor in the Fairisle ATM network. In *Proceedings Twelfth UK Teletraffic Symposium*, Windsor, England, March 1995. IEE, Springer Verlag.

[17] Simon Crosby, Meriel Huggard, Ian Leslie, John Lewis, Fergal Toomey, and Cormac Walsh. Bypassing modelling: Further investigations of entropy as a traffic descriptor in the Fairisle ATM network. In *Proceedings WATM'95 First Workshop on ATM Traffic Management*, Ecole Nationale Superieure des Telecommunications, Paris, December 1995. IFIP W.G. 6.2.

[18] Simon Crosby, Ian Leslie, John Lewis, Raymond Russell, Fergal Toomey, and Matt Davey. Exploring the queueing behaviour of atm switches. In *Proceedings Thirteenth UK Teletraffic Symposium*, Strathclyde University, Glasgow, March 1996. IEE.

[19] Nils Bjorkman, Alexander Latour, Aziz Miah, Simon Crosby, Ian Leslie, John Lewis, Raymond Russell, and Fergal Toomey. Exploring the queueing behaviour of ATM switches. In *Performance '96*, EPFL, Lausanne, Switzerland, October 1996. IFIP WG 7.3.

[20] ATM Forum. *Traffic Management Specification, Version 4.* ATM Forum/95-0013R8, October 1995.

[21] Sugih Jamin, Peter Danzig, Scott Shenker, and Lixia Zhang. A measurement based admission control algorithm for integrated service packet networks. *IEEE Transactions on Networking*, 5(1):56–70, February 1997.

[22] Sugih Jamin, Peter Danzig, Scott Shenker, and Lixia Zhang. A measurement-based admission control algorithm for integrated services packet networks. In *Proceedings ACM Sigcomm*, Cambridge MA, October 1995.

[23] Sally Floyd. Comments on measurement based admission control for controlled-load services. Submitted to CCR, July 1996.

[24] M. W. Garrett and W. Willinger. Analysis, modeling and generation of self-similar vbr video traffic. In *Proceedings ACM SIGCOMM 94*, pages 269–280, London, UK, August 1994.

[25] Matthias Grossglauser and Jean-Chrysostome Bolot. On the relevance of long-range dependence in network traffic. In *Proceedings ACM Sigcomm*, Stanford University CA, August 1996.

[26] Richard Black, Ian Leslie, and Derek McAuley. Experiences of building an ATM switch for the Local Area. In *Proceedings ACM SIGCOMM*, volume 24(4), September 1994.

[27] E. Buffet and N.G. Duffield. Exponential upper bounds via martingales for multiplexers with markovian arrivals. *J. Appl. Prob.*, 31:1049–1061, 1992.

[28] Song Chong, San-qi Li, and Joydeep Ghosh. Dynamic bandwidth allocation for efficient transport of real-time VBR video over ATM. In *Proceedings IEEE INFOCOM*, pages 81–90, Toronto, Ontario, Canada, June 1994.

[29] V. Paxson and S. Floyd. Wide-area traffic: The failure of poisson modeling. In *Proceedings ACM SIGCOMM 94*, pages 257–268, London, UK, August 1994.