Adaptive ATM Access Control

Timothy X Brown
Josua Alspector
Bell Communications Research
October, 1994

1. Introduction

ATM is an emerging standard in broadband communication that allows traffic of different rates, both continuous and bursty, onto a single channel similar to high-speed protocols such as ethernet or the internet. Unlike these *best-effort* services, ATM *guarantees* customers a given quality of service in terms of different combinations of loss-rate, delay, and delay jitter.

One problem is how to admit the maximal number of these disparate users while maintaining service quality. Simply allocating users bandwidth equal to their maximal (peak) transmission rate, while straightforward, would be uneconomical for bursty users who may have an average rate at a small fraction of their peak rate. Methods that can *statistically multiplex* many bursty users would provide significantly greater network utilization. For example, a source with many short bursts and 5% utilization could potentially be multiplexed with close to 20 times more users on the channel than with peak rate allocation while maintaining the same quality of service level. But in order to guarantee quality of service, any statistical multiplexing technique must be robust to a range of call statistics. With this motivation, we more fully describe the problem.

1.1 The Control Problem

The access control problem is pictured in Figure 1. A user wants to access a given link on which other users with varying traffic types already have access. The users are multiplexed either directly or by the switching of different inputs to one of the switch’s outputs. A buffer in the mux protects against momentary overloads. When a new user arrives they must first request access from the *access controller*. The user gives certain call parameters, and then the controller accepts

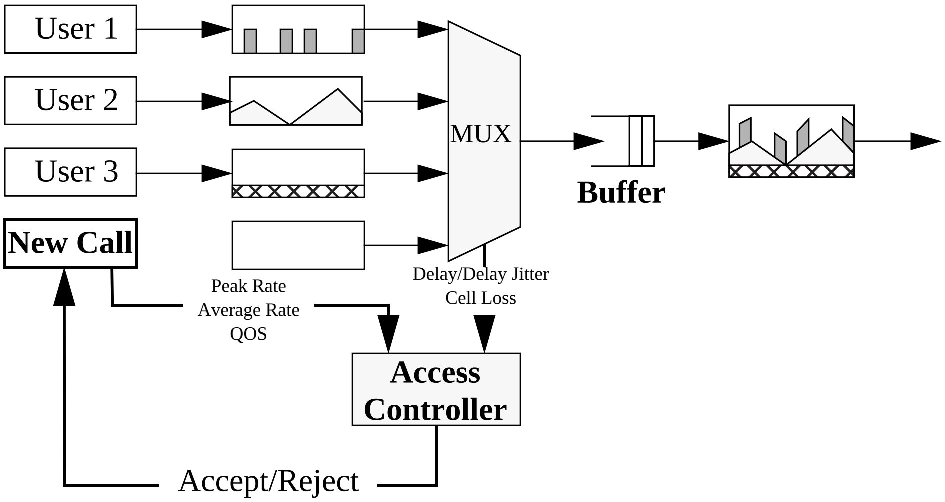


Figure 1 The access control model.

or rejects the request based on factors such as the current status of the mux and the number and type of already accepted calls. Resources for calls are released only when the user notifies that he is disconnecting. In this manner calls are made and disconnected over time.

The parameters required for evaluating a connection request are the desired quality of service (QOS), peak and average transmission rates, and the burst period [ATM93]. The latter has precise meaning for certain types of sources, but can be interpreted as the time scale over which a source must be observed to measure the average rate. Quality of service is expressed in terms of factors such as loss rate, cell delay, and delay jitter (variance).

Cells are lost when the link is overloaded and the buffer overflows. Overloads, if small enough, can be absorbed by the buffer. For this reason traffic with bursts much shorter than the buffer length are less likely to generate a sustained overload and cell loss than traffic with bursts much longer than the buffer length. We will observe this phenomena later. Delay, delay jitter, and cell loss for typical traffic are related in the sense that cells that violate QOS constraints can be considered lost. For this reason we will focus mainly on cell loss.

1.2 The Role of Different Traffic

Constant rate calls can in general be guaranteed good quality of service by reserving bandwidth on the channel equal to its transmission rate. The problem of access control revolves around variable rate or bursty sources. The archetypical and worst-case bursty source is one that alternates between transmitting at its peak rate and not transmitting at all. Such *ON/OFF* sources are distinguished by three factors: the peak rate, the distribution of the ON period length and the distribution of the OFF period length. From these factors, statistics necessary for ATM protocol such as peak rate, average rate and ON/OFF time scale are determined.

1.3 Other Control Approaches

1.3.1 Peak Rate

We compare our method with three other approaches. The first is *peak rate* allocation in which users are accepted so long as the sum of the peak transmission rates remains below the channel capacity. Given a large enough buffer to handle simultaneous arrivals from the different inputs, no cells can be lost. This desirable latter property is offset by the under utilized resources that result. We shall see that utilization can easily be a small fraction of what is possible.

1.3.2 Zero Buffer

A method that is both safe, like peak rate allocation, and has the potential for allowing higher utilization is the stationary or *zero buffer* approximation[Gue91]. This assigns a capacity to a set of calls so that the probability that the aggregate transmission rate exceeds this capacity is less than the loss ratio:

$$C_{\mathrm{ST}} = \sum_i m_i + \sqrt{-(2\ln(\varepsilon) + \ln(2\pi))\sum_i \sigma_i^2} \qquad (1)$$

where

ε = loss rate upper bound
m_i = mean transmission rate of call i
σ_i^2 = transmission rate variance of call i.

This works best with many low-rate and high peak to average ratio sources, otherwise peak rate allocation can be used. The rate variance is easy to calculate for ON/OFF sources with peak rate, P, and mean rate, m: $\sigma^2 = m(P - m)$. Assuming no buffer, it, like peak rate, suffers from reduced utilization.

1.3.3 Guerin Equivalent Bandwidth

The previous approaches are robust in the sense that their performance is dependent only on simple aggregate statistics. A reasonable extension would be to use all the call’s traffic statistics. Mitra [Mit88] analyzed for Maarkov source models (implying exponentially distributed ON and OFF period lengths). This method is able to make precise predictions, but at the expense of a complex and time-consuming calculation.

One simplification of this technique, the *Guerin equivalent bandwidth*[Gue91], is practical while still efficient compared to other techniques. Access control is reduced to assigning each source an effective or equivalent bandwidth and making sure that the sum of these equivalent bandwidths is less than the channel capacity. A calls equivalent bandwidth is assigned according to the formula:

$$C_{\text{equiv}} = R_{\text{peak}}\left(\frac{1}{2} - z + \sqrt{\left(\frac{1}{2} - z\right)^2 + 2\rho z}\right) \tag{2}$$

where

R_{peak} = peak transmission rate
$z = \dfrac{Q}{2\ln(1/\varepsilon) b R_{\text{peak}}(1-\rho)}$
Q = mux/switch buffer size
ε = loss rate upper bound
b = mean burst ON period
ρ = utilization /mean rate.

Based on simulation experiments using this model, it always meets its desired loss rate bounds. In fact, it has been criticized for allocating too much bandwidth [Cho94] since it ignores statistical multiplexing. In any case it is a useful benchmark.

1.4 An Access Controller Taxonomy

To understand the relationship between these controllers we develop a simple taxonomy for classifying access controllers.

Treating arbitrary traffic processes is difficult, leading—as in the Guerin equivalent bandwidth technique—to assumptions about the traffic model. As we will see, these *model-based* techniques fail when traffic does not follow the model.

Equivalent bandwidth techniques are *linear* in that the necessary bandwidth is a linear function of the number of each type of source. To be robust, the assigned bandwidth must be enough so that, once accepted, the call always will “fit” regardless of what calls connect or disconnect later; ignoring much of the statistical multiplexing potential. For identical sources a practical sub-linear allocation along the lines of the Guerin equivalent bandwidth is described in [Kri94], but is not discussed further here since it is based on assuming a particular source model.

Several *adaptive* techniques have been proposed for accepting or rejecting users based on models that observe and adjust with the control performance, [Hir90] [Che92] and [Est94] being representative examples. These have been shown to produce near ideal control models in a variety of situations. While outlining many of the important issues, these earlier attempts are applied to simpler models than treated in this paper and do not appear viable in the complex many-dimensional traffic models expected in practice.

Table 1 summarizes the access control features. A taxonomy of the controllers described appears in Table 2. The two neural network (NN) controllers are the subject of this paper. Since the nonlinear techniques promise the greatest efficiency, we will concentrate on the NN Gestalt approach. Table 2 also points out some as yet unexplored classes of controllers that may provide a desirable mix of features.

1.5 Simulator

We have developed a general purpose cell-by-cell software simulator tool for simulating ATM traffic based on the model in Figure 1. It provides a uniform test bed for evaluating different

Table 1: Three Dichotomies for Comparing Access Controllers

Feature	Advantages	Disadvantages
Linear Rule	Linear properties ease computations.	Ignores multiplexing gain with more users.
Non-Linear	Can get full statistical multiplexing gain.	May be unwieldy.
Model Free	Maximum applicability, robustness.	May waste bandwidth.
Model Based	Can lead to tractable closed-form solutions.	Model may not apply to given data.
Adaptive	Can squeeze most from available BW.	Requires feedback mechanism.
Non-Adaptive	Lower complexity.	May waste bandwidth or be brittle.

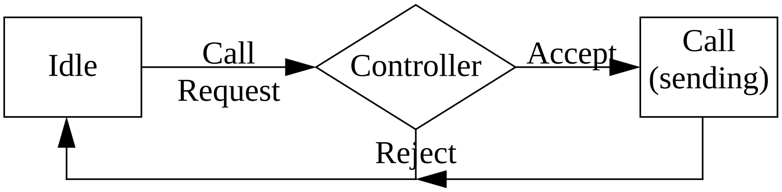


Figure 2 Source Model. A separate process generates cells within a call.

switching strategies. The tool simulates a source as in Figure 2 and can simulate 1000’s of combinations of idle length, call length and cell generation processes. It also can implement a variety of access control schemes. The program, since it is tuned to this model, can simulate N sources for T x10^7 time slots in about NT minutes on a Sun Sparc workstation—fast enough to be able to generate enough data to observe the tail behavior of the network. For generality, all traffic rates are normalized by the link capacity to a maximum of one cell per time slot.

1.6 Disclaimers

This paper ignores many important issues such as traffic shaping, policing of traffic sent versus traffic requested, fair queueing policies, bandwidth reservation, routing, discrepancies between the peak, average, and burstiness definitions here and in the ATM Forum, and transmission errors. These are issues that can be addressed one by one in future work. While the access control work here is centered on ATM, it applies to many other domains.

1.7 Paper Overview

We first show that the conventional controllers fail on an important class of traffic. The flaw is that they assume a model for which there is no guarantee that the traffic may fit. We next describe an adaptive scheme based on neural networks. This scheme is then shown to be better than the conventional approaches both when given ideal data, and troublesome data. We finish with a discussion of system requirements and a business case.

2. What’s Wrong with Model-Based Approaches?

Treating arbitrary source models is difficult, so often assumptions are made about the distribution of source statistics. These can produce very good results. Using the example at the introduction

Table 2: Taxonomy of controllers. We consider in detail the controllers in bold.

		Linear	Non-Linear
Model Free	Adaptive	NN EqBW	**NN Gestalt**
	Non-Adaptive	**Peak Rate**	**Zero Buffer**
Model Based	Adaptive	??	??
	Non-Adaptive	**Guerin EqBW**	Krishnan

with bursts 1% of the buffer size; utilization 5%; and loss 10^{-6}, we find from (2) that we can safely admit 17 times as many users as would be indicated by assigning $C_{\text{equiv}} = R_{\text{peak}}$.

2.1 Importance of Burst-Length Distributions

These results assume that the distribution of ON and OFF periods is exponential (See Table 3). This is argued to be worse-case since the exponential distribution has a heavier tail then many other distributions, that is, it is more likely to have large deviations from the mean than other distributions. Recent work on monitoring ethernet traffic indicates that data traffic is even worse than many of these presumed worst-case assumptions[Lel93]. They suggest that such traffic has burst-lengths that are distributed more as root exponentials or even an infinite variance distribution like the Pareto. These have much greater probability of buffer-filling long ON periods as seen in Table 3. The one in a million bursts (last column) for the exponential distribution will be only ten times the mean, for the root exponential 100 times, and for the Pareto, can be many 1000's of times.

In the simulations, we use the geometric distribution instead of the exponential, and for the others use continuous distributions as approximations for the discrete burst lengths. The effect of these heavy tailed distributions is explored in the next section.

2.2 Experimental Results

To see the effects of these heavy tail distributions on the equivalent bandwidth controller, we simulated 220 million cells (10 minutes at 155Mb/sec) of the different traffic types. For two different ON/OFF period lengths, peak/average ratio of 2 and buffer size of 1000, we chose four equal sources with the largest acceptable load according to (2). The results are in Table 4. For the zero buffer and peak rate allocation no cells were lost under any distribution but the normalized load was always 1.0, a significantly smaller utilization.

The heavy tailed distributions clearly overwhelm this exponential-distribution based techniques, even in the long burst experiments in which they allow only 3% more load than with peak rate allocation. The next section describes a technique that allocates peak rate when the traffic dictates, and allows more traffic when it safely can.

Table 3: Distributions for periods of mean length *m*.

Distribution	P(period length > x)	Variance	x: P=10^{-6}
Exponential	$e^{-x/m}$	m^2	14 m
Root Exponential	$e^{-\sqrt{2x/m}}$	$6m^2$	95 m
Pareto	$\left(\frac{x(1+\varepsilon)}{m\varepsilon}\right)^{-(1+\varepsilon)}, x > \frac{m\varepsilon}{1+\varepsilon}$	∞ $(\varepsilon < 1)$	26,000 m $(\varepsilon = 0.1)$

3. Neural Approach

As seen in the previous section, by choosing simple, tractable models, the model-based approaches can not guarantee QOS bounds when given real data. Two alternatives are to either choose more complex models or to choose the very simple peak rate. The latter has the clear potential for wasting bandwidth, making many services too costly. The former risks never capturing every possible service type that may come along, or being too computationally expensive to allow real-time access control.

We propose a data driven technique that adapts to the actually observed data sources. The controller system would observe the traffic generated, and store examples of source combinations and their achieved quality of service. These can then be processed off-line (for example on a daily basis) to modify the controller function. We explore neural networks as the adaptive element for this control scheme.

3.1 Why Neural Networks?

A neural network can be thought of as a black box that accepts a large number of observations of input/output pairs from a given function and produces an approximation to the function. With enough observations, a neural network can approximate bounded continuous functions to arbitrary accuracy[Sti90]. Experimentally, neural networks have been shown to do at least as well as other statistical modeling techniques on finite data sets[Ng91]. Beyond simply performance, the neural network has the advantages of an iterative update capability, and efficient representations, both in terms of memory size, and computational efficiency. If necessary, specialized hardware developed at Bellcore[Jay92], can provide additional speed.

As another advantage, the neural network can use its “data fusion” capabilities, to consider all available information and all aspects of the quality of service including cell loss, delay and delay jitter. This is important since recent work has suggested that the standard ATM parameters (peak rate, average rate and burst time scale) may not be measurable let alone sufficient for describing a traffic stream [Wil94]. Other parameters such as burst size variance, Hurst parameters, and traffic class may be necessary.

Table 4: 10 Minute Simulation using Equivalent Bandwidth Allocation

Source Distribution	Mean Burst Period	Load/Load w/Peak Rate	Loss Rate (10^{-6} target)
Exponential	100	1.75	0
Root Exponential	100	1.75	3×10^{-3}
Pareto ($\varepsilon = 0.1$)	100	1.75	4×10^{-2}
Exponential	10000	1.03	0
Root Exponential	10000	1.03	3×10^{-5}
Pareto ($\varepsilon = 0.1$)	10000	1.03	2×10^{-4}

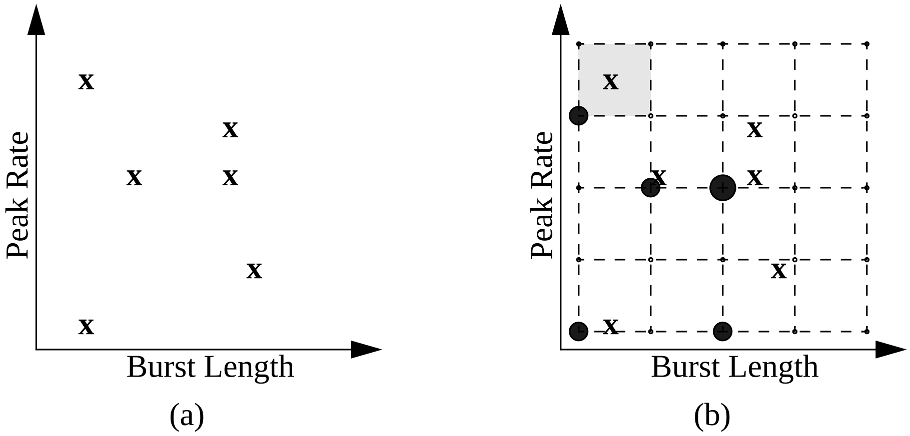


Figure 3 Representing a configuration of calls (a). Binning and weighting to get a finite representation (b).

3.2 Neural Approach: Equivalent Bandwidth

A bandwidth allocation technique depends on a function that maps the call characteristics into a bandwidth to allocate for the call. As a function approximator, the neural network can approximate an ideal bandwidth allocation function by allocating bandwidth to calls and then observing the quality of service when the call completes; continuously adapting and refining its control with the actual data.

At first glance, an equivalent bandwidth approach is alluring in that a single relatively simple function needs to be developed for assigning bandwidth to each call. But as with all linear equivalent bandwidth techniques, it would not take advantage of statistical multiplexing.

3.3 Neural Approach: Gestalt

The approach we pursue considers the gestalt of all calls that access the link. The controller considers all calls including the new call request simultaneously and then decides if all calls will meet their quality of service or not. Based on this, the controller will accept or reject the new call.

Critical to this approach is developing a tractable representation for all of the calls. The simplest approach would classify each call into one of a finite number of classes, and then represent a combination of calls by a vector that counts the number in each call class. In practice this may require 1000’s of classes to adequately represent the many combinations of call type, peak rate, utilization, and burstiness; and with every new user yet another class may have to be developed.

Instead, we plot each call in the space of call parameters as shown for two dimensions in Figure 3a. The controller only needs to recognize patterns of calls that will meet or not meet QOS. To translate this continuous representation into a finite representation, but also having the fidelity to distinguish many different call types, we first divide the space into a grid and then count not the number of calls in each bin, but the total load in each bin. In Figure 3b this is represented by allocating each grid box’s load to the node in the lower left corner. As a further refinement we distribute a call’s load to all the box’s corners in proportion to how close it is to that corner. The real vector of the node loads represents the set of calls. This could be fed to any of a a variety of classifiers to determine whether to accept or reject.

The gestalt controller with this representation has the advantage that it reduces the control problem to the well understood classification problem. The equivalent bandwidth approach does not. To see the significance, suppose we could observe a set of simultaneous calls for a long period of time and decide whether QOS is met or not. With the equivalent bandwidth approach, we would now only know that the bandwidth allocated to each call by the current model in the current combination needs to be increased or decreased. For the gestalt approach, we now have an absolute exemplar that could be saved for making future decisions.

Note that if the classifier is a linear decision function (i.e. a threshold on the weighted sum of the contents of each bin) then this is equivalent to an equivalent bandwidth approach. This implies non-linear classifiers such as neural networks are essential for doing more than equivalent bandwidth.

3.4 Neural Network Experiments

Using the simulator we tried four types of experiments. One was similar to the experiment in Table 4, the next was an experiment using 24 sources divided equally into the 6 classes listed in Table 5 the next used a measured video data trace, and the last used ethernet data traces. In all cases the target loss rate is 10^{-6}, the mux buffer is 1000 cells, and the rates are normalized by the link capacity. The first experiment is a stringent test of a controller's ability to predict QOS, the next is a measure of how much load the controller will accept, and it's ability to meet QOS requirements when calls are continuously arriving and departing, the last two measure the controller's performance on real data and appear in the following sections.

The results are shown in Tables 6 and 7. Comparing Tables 6 and 4, the gestalt controller carries similar loads for the same level of service in geometrically distributed traffic, but defaults to nearly peak rate for the heavy tail distributions. This is reflected in Table 7 where for the mixed model with geometric period lengths the neural network and Guerin approaches carry similar loads that are significantly more traffic than peak rate (50%) while maintaining QOS. With Pareto distributed period lengths, the carried traffic was (10%) higher than peak rate, but less then the Guerin equivalent bandwidth, but unlike this latter technique, the neural network maintained quality of service. For this low peak to average ratio, zero buffer allocation reduced to peak rate

These results show that the model based technique is unreliable, and that more traffic can be carried than peak rate without violating quality of service using the Gestalt approach. To see the performance with larger peak to average ratios, we use a mixed simulation with the 10 to 1 traffic in Table 8 where 20 of each source type were used. The results are in Table 9. These show that with

Table 9: Controller Accepted Load and Loss in the 10-1 Mixed Simulation

Controller	Geometric: Load(Loss)	Pareto: Load(Loss)
Peak Rate	0.0877(0)	0.0964(0)
Zero Buffer	0.1805(0)	0.2167(0)
Guerin Eq. BW	0.3639(0)	0.3897(2e-3)
NN Gestalt	0.4908(0)	0.2699(6e-7)

the neural network controller, the carried traffic was more than five times peak, three times zero buffer, and a third more then Guerin with geometric distributions. For Pareto distributions the carried load is 2.5 times peak and 25% more than zero buffer, whereas the equivalent bandwidth fails. This shows that even with the Pareto distribution we can do significantly better than peak rate allocation.

Table 5: Source Classes Used in 2-1 Mixed Simulation

Peak Rate	Mean Period ON/OFF	Peak to Avg. Rate	Mean Duration Call/Inter-Call	Offered Load
0.9	$10^2/10^2$	2	$10^5/10^5$	0.225
0.3	$10^2/10^2$	2	$10^5/10^5$	0.075
0.1	$10^2/10^2$	2	$10^5/10^5$	0.025
0.9	$10^4/10^4$	2	$10^5/10^5$	0.225
0.3	$10^4/10^4$	2	$10^5/10^5$	0.075
0.1	$10^4/10^4$	2	$10^5/10^5$	0.025

Table 6: 10 Minute Simulation using Neural Network Gestalt

Source Distribution	Mean Burst Period	Load/Load w/Peak Rate	Loss Rate (10^{-6} target)
Exponential	100	1.75	0
Pareto ($\varepsilon = 0.1$)	100	1.00	0
Exponential	10000	1.04	0
Pareto ($\varepsilon = 0.1$)	10000	1.00	0

Table 7: Controller Accepted Load and Loss in the 2-1 Mixed Simulation

Controller	Geometric: Load(Loss)	Pareto: Load(Loss)
Peak Rate	0.3819(0)	0.3792(0)
Zero Buffer	0.3819(0)	0.3792(0)
Guerin Eq. BW	0.5648(0)	0.5591(2e-3)
NN Gestalt	0.5851(0)	0.4377(0)

3.5 Video Data Experiments

3.5.1 The Data

We obtained a trace from 2 hours of a variable rate video source[Bel94]. It is described in detail in [Gar93]. The data is in terms of bytes per 1.39 msec data *slice*, with 30 slices per frame, 2850 frames per data file, and 60 two-minute files total.

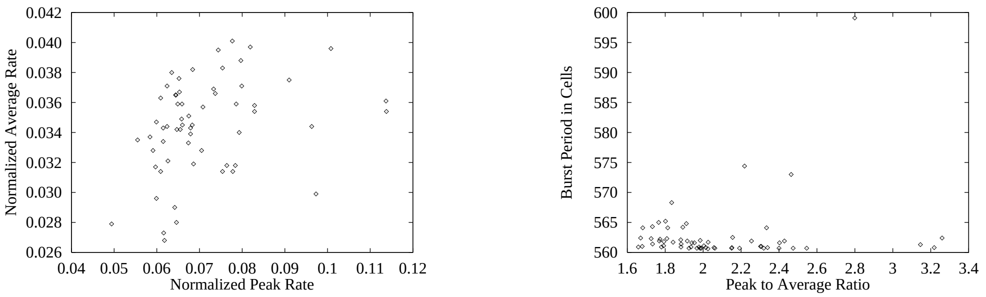
Our method was to treat each file as a separate data source and calculate traffic descriptors separately for each. The average rate was determined assuming transmission rate of 560.6 cells/slice period (i.e. 155 Mb/sec) with 48 bytes per cell. The peak and burst period were less straight forward since the slice time was somewhat arbitrary. The peak was determined by finding the largest average over 11.1 msec's (8 slices). Once the peak was determined, we modeled the source as a deterministic server queue which sends data 48 bytes at a time at the peak cell rate and on which a slice of data is added at the beginning of each slice period. With this model, an ON burst was defined as the time the queue was not empty and the average ON and OFF burst periods were determined. Note that the sum of the average ON and OFF periods could never be less than 1 slice time.

To see the data characteristics, Figure 4a plots the peak vs. average rate normalized by a 155 Mb/sec channel capacity for the 60 data segments. The peak rates are small indicating that at least 10 can be multiplexed onto the channel. Figure 4b plots the peak to average ratio vs. the burst period. We see that the ratio varies from about 1.6 to 3.4. Because many are multiplexed together, we would expect that we can be carrying loads close to the channel capacity.

One relevant and interesting characteristic of this data [Gar93] is the distribution and periodicity of the bursts. In typical segments, the data in frames during a scene are small relative to frames at scene boundaries which are many times larger. This tends to produce a multimodal distribution. The slices that make up a frame are highly correlated from frame to frame so that if the *n*th slice of one frame is large then the *n*th slice of preceding and subsequent frames are large also. The non-monotonic distribution, and the periodic behavior imply that assuming the burst are from a simple geometrically distributed ON-OFF process is folly.

Table 8: Sources Used in 10-1 Mixed Simulation.

Peak Rate	Mean Period ON/OFF*	Peak to Avg. Rate	Mean Duration Call/Inter-Call*	Offered Load
0.9	$10^2/9x10^2$	10	$10^6/10^6$	0.045
0.3	$10^2/9x10^2$	10	$10^6/10^6$	0.015
0.1	$10^2/9x10^2$	10	$10^6/10^6$	0.005
0.9	$10^4/9x10^4$	10	$10^6/10^6$	0.045
0.3	$10^4/9x10^4$	10	$10^6/10^6$	0.015
0.1	$10^4/9x10^4$	10	$10^6/10^6$	0.005



(a) (b)

Figure 4 Plot of normalized peak rate vs. average rate (a) and peak to average ratio vs. burst time period (b)

3.5.2 Experiment and Results

We simulated an ATM link carrying random combinations of the data segments, each combination consisted of 0–2 randomly offset copies of the odd numbered segments. The results of this were used to train the adaptive controller. All of the controllers were tested on a mixed-style simulation consisting of 60 sources, 2 for each of the even numbered segments, with geometrically distributed call and inter-call periods of mean length 9x10^6 and 10^6 cells respectively. The target loss rate was again 10^{-6}. The net load is about 1.9 times the capacity. Note that the test segments are different then (but not necessarily independent of) the training segments.

The results are in Table 10. The equivalent bandwidth technique loses too much data. For the peak to average ratios in this data, the zero buffer controller reduced to peak rate. The neural network increases the capacity by 50% over peak rate, while still maintaining quality of service.

3.6 Ethernet Data

3.6.1 The Data

We used data collected from a single ethernet link at Bellcore. The data is described in [Lel93] as the August 89 busy hour and contains traffic ranging from busy file-servers/routers down to active users on down to users with just a handful of packets. The ethernet transport protocol allows an essentially arbitrary number of users to use a single bus. Users are either idle, or transmitting data packets at the bus's full rate. Characteristics of the data are in Table 11. Note that the packet sizes are bounded to about a kilobyte. This data is still heavy tailed, since the distribution of the interar-

Table 10: Controller Accepted Load and Loss with Video Data.

Controller	Load(Loss)
Peak Rate	0.5007(0)
Zero Buffer	0.5007(0)
Guerin Eq. BW	0.9231(4e-5)
NN Gestalt	0.7576(0)

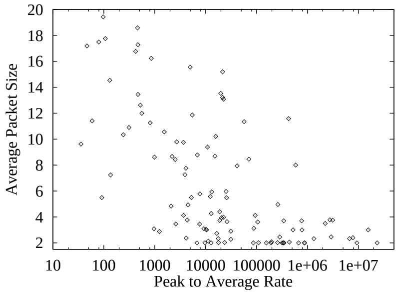


Figure 5 Plot of peak to average ratio vs. burst time period for ethernet data.

rival times is heavy tailed. Recall that this means a few very long periods offset by many short periods. So, even though the individual packets are bounded they tend to come in trains of packets one after the other followed by long “inter-train” periods.

The detailed data set records every packet’s: arrival time (to the nearest 20μsec), size, plus source and destination tags. This can be used to break out individual users traffic streams. For the purpose of our simulations, we are trying to understand the ability to multiplex individual users. Ethernet contains a collision protocol that prevents bandwidth overload, so by definition all of the traffic can fit on to a 10 Mb/s line. If instead we decorrelate the users via random offsets, we can understand the process of multiplexing many independent data sources. Since the peak rate is fixed, we can get a good idea of the wide range of traffic characteristics from the plot of peak to average ratio vs. average burst size in Figure 5.

3.6.2 Experiment and Results

We designed an experiment similar to the ones for the video data. Because of the low data rates, we assumed a model where a portion (5Mb/sec) of the ATM link’s capacity is isolated from the other traffic types—using a scheme such as fair queueing [Dem90]—and allocated to this service. We divided the 108 sources into two groups of 54 “users” each. Because of the 6 orders of magnitude variation in source loads, we were careful to make sure that each group had similar total load and load distribution by first ordering them by decreasing load, and then dividing them roughly into odd and even sets ({1,4,6,8,...} and {2,3,5,7,...}). As with the video data we simulated an ATM link carrying random combinations of the sources, each combination consisted of 0–4 randomly offset copies of the odd numbered segments. The results of this were used to train the adaptive controller. All of the controllers were tested on a mixed-style simulation consisting of 108 sources, 2 for each of the even numbered segments, with geometrically distributed call and inter-

Table 11: Characteristics of Aug89.busyhour Ethernet Data.

Data Rate	10 Mega bits/sec
Cell Rate	26042 Cells/sec
Average Load	15.1%
Number of Packets	1,404,444
Smallest, Largest Packet	64, 1518 bytes
Number of Sources	108

call periods of mean length $8x10^6$ and $2x10^6$ cells respectively. The target loss rate was again 10^{-6}. The net load is only 24% of the capacity, while the total peak rate is 108 times capacity.

The results are in Table 12. The equivalent bandwidth technique again loses too much data. The zero buffer controller accepts significantly more than peak rate. The neural network increases the capacity by 75% over peak rate, while still maintaining quality of service. Table 12 lists the maximum number of simultaneously accepted calls, illuminating the source of peak rate's poor performance.

4. Implementation Issues

To apply the neural network access controller to a real system, several issues must be addressed, including rate of adaptation, decision speed, memory requirements, availability of necessary information and insertion point in the system that would have the necessary information. We envision the system receiving updates on a long time scale on the order of every evening. Shorter time scales risk over fitting to extremely short term changes, longer scales will not react quickly to new usage patterns. The gestalt approach would typically use less than 1000 floating point operations for each decision. The neural network builds a compact representation that can be stored in less than a ~1000 bytes of memory. The software itself is compact, but if monitoring equipment is generating exemplars for off-line training of the neural networks mapping, disc storage on the order of a megabyte may be necessary. These requirements may or may not be limiting depending on the capabilities of the switch and switch software with which it is implemented.

The controller, to be adaptive, needs feedback on its decisions. In the scenarios that I simulated, this feedback was in the form of number of cells lost between two periods of time. The ForeSwitch has counters and query points that make this information available. It is not clear though whether this is available for all switches, although some newer switch designs provide this information down to the connection level. Switches conforming to [GR94] would have this information and more.

Target insertion points for the controller include the fabric control module of Bellcore's Q-Port software, Bellcore's BroadNet/BNMS software, and test beds such as the Aurora network. Discussions with Q-port developers indicate that alternative controllers can be incorporated into their software design. Discussions with the BroadNet designers indicate that the ideas here could be used for setting/recommending the *Booking Factor*, or statistical multiplexing level. We are working with Aurora's developer here in Bellcore to see when and how they could use the adaptive scheme. Much of the hardware necessary and software interfaces is already in place.

Table 12: Controller Accepted Load and Loss with Ethernet Data.

Controller	Load(Loss)	Max. Simul. Calls
Peak Rate	0.0029(0)	1
Zero Buffer	0.0420(0)	81
Guerin Eq. BW	0.2423(8e-4)	98
NN Gestalt	0.0604(0)	86

5. Economic Impact

5.1 Assumptions

Reference [Mil93] has market and economic analysis of a few architectural options for implementing ATM networks in a range of RBOC LATAs. With their data we can estimate the economic impact of using our adaptive admission control scheme for Bellcore customers. This assumes that Bellcore network control software contains our adaptive call admission control scheme and that the alternative for comparison is either of the simple but safe zero buffer or peak rate allocation.

Furthermore, we assume that our control software can make twice the bandwidth available as these schemes with the same hardware facilities. This is an estimate based on the mixture of call types in previous sections that gave as little as a factor of 1.5 and as much as a factor of 3. Since the other alternative analytic (non-adaptive) equivalent bandwidth schemes can not be relied on, we will assume that a customer’s only reasonable alternative is to use peak or zero buffer allocation.

5.2 Architecture and LATA Choice

In [Mil93] the authors compared a few large switches located at LATA hubs with many small switches located at customer premises and central offices. This latter ‘edge’ architecture was cheaper to install than the ‘hub’ architecture, so we chose this for analysis to be conservative. They also compared urban, rural, and suburban LATAs. We choose their urban LATA which was from a medium size city. Their customers were chosen from existing high-speed data customer using T1 and T3 facilities.

5.3 Installed First Costs for a SONET Ring

In a discussion with the authors, we devised a scenario to provision a LATA to begin ATM service. In the edge architecture, it is assumed that there are SONET rings with 3 service nodes (including an ATM switch) per ring. A fourth service node in the central office interconnects rings. It is assumed that each service node serves 3 customers which allows for the inexpensive 4 by 4 ATM switch at that node (3 for customers, one for the unidirectional ring). Such a switch costs $40,000.

In addition to the switch costs, there is a start-up cost of $16,200 per service node for SONET service plus $1,150 per DS3 connection plus some fiber costs. The total cost per service node is about $60,000. Therefore a ring with 4 service nodes costs about $240,000 to provision plus a small cost for fiber.

5.4 Increased Costs Due to the Need to Double Capacity

Let us now assume that the ring is filled to capacity because of increased traffic load. The network providers are faced with a variety of options. They can leave the existing ATM switches in place and put some of their customers on new switches. If they duplicate each switch at each service node, this would cost another $40,000 per service node. A cheaper alternative would be to upgrade to an 8 by 8 ATM switch. When provisioned for the same 3 customers, this switch costs $70,000 or an additional $30,000 per service node above the original 4 by 4 switch. So the cost of a service node increases from $60,000 to $90,000 if the traffic doubles. This represents an increase in costs of 50 percent for a doubling of traffic. This ignores additional costs for labor in replacing the switch, the costs of an upgraded operating system, the cost of renewed testing, etc.

5.5 Costs and Revenue per LATA

In the medium sized city used in the special report, there are 35 rings. The total cost for provisioning this LATA in the study came to $10 million which is slightly more than 35 multiplied by $240,000 due to costs we have ignored. However, our estimate of a 50 percent increase in costs to reprovision for a doubling of the traffic seems reasonable. Therefore, we can expect that it will cost an additional $5 million per LATA if we have an increase in traffic demands by a factor of two.

We do not know the exact revenue expected from high speed data customers, but have estimated it at $3,000 to $4,000 per month per cell relay customer. This gives revenue of about $10 million per year per LATA. Currently, many of these customers have flat rate service based on input line capacity. If traffic doubles without an increase in customers, the increase in revenue would be the additional premium for higher rate lines. If there is an increase in customers, then revenue could double.

5.6 Costs and Revenues with Capacity Increase Due to Adaptive Admission

If the traffic doubles without an increase in customers, then the network as originally provisioned would be able to handle this increase if the adaptive admission control algorithm were part of the network control software. In this way, a Bellcore client would save about $5 million per LATA in reprovisioning costs. Even with the flat rate service, this represents a substantial cost saving, and a significant competitive advantage for Bellcore software.

Depending on the premium for the higher rate service, up to an additional $10 million per year in revenue would be generated without any new facilities, new operating systems, and without rejecting customer requests for increased service.

If the increase in traffic demands is due to new customers, rather than increased traffic, then there is revenue increase, however, the advantage of the adaptive admission control algorithm is not so clear cut. Each customers fills one slot on the ATM switch. Even with our algorithm, the switches would still have to be upgraded to provide for new customers leading to no advantage. However, because of the flexibility implied by delaying upgrading on some switches which are limited by traffic (not customers), there would still be an expected savings. Perhaps it wouldn’t be the 50 percent number, but we guess it might be about 25 percent.

There are significant unknowns in this analysis. We don’t include cost savings other than ATM switch provisioning, we don’t know the rate schedules. Nevertheless, we feel that the adaptive admission control algorithm would provide for a 25 to 50 percent reduction in provisioning costs for a doubling of traffic. Alternatively, one could increase revenue with the same facilities if this algorithm were implemented in the control software. This seems to us an important strategic and competitive advantage when a network provider makes a decision to purchase control software.

6. Related Problems

We have carefully examined one aspect of the ATM network control problem. The techniques developed here are to varying degrees applicable to other aspects of the problem.

In routing, a fundamental operation is the determination of whether a given link in a potential route has sufficient capacity to accept the current call. A connection over several links is acceptable only if all of the separate links can accept the connection's traffic. In standard circuit switched routing, this is a matter of counting available circuits. As we have seen in ATM the acceptance of a new call is dependant on the detailed characteristics of all of the already carried calls. Switching nodes could keep track of the gestalt representation of other links and their decision function model in order to search for available links.

In ATM, where services may be transported over heterogeneous providers' networks with proprietary acceptance algorithms, a method for learning probability of acceptance as a function of advertised usage parameters would be an alternative to having an exact state of a link. This model can be built with the same techniques described in this paper.

In provisioning, the traffic engineer must decide where and when capacity must be added. Standard measures such as utilization can be nearly meaningless since utilization may be small to accommodate an erratic or bursty user. The controller models built with the techniques from this paper can be queried to see what additional traffic can be carried.

The general form of this problem—classifying whether a set of calls is acceptable or not based on observing past visits to the state—can be expanded to include not just when the customer observes bad service, but also when the provider observes bad service, e.g. for deciding if a set of calls may exhaust a network resource which may not have a direct impact on QOS, but may impair monitoring/billing/maintenance/reliability/etc. functions.

7. Conclusions

We have described an access control technique and shown that it admits calls onto a link while maintaining quality of service despite non-standard call processes and realistic data that cause other controllers to fail. For bursty traffic it can admit significantly more traffic than simple peak rate, carrying 5 times the load in a mixed 10-to-1 peak-to-average source simulation. Compared with alternative controllers, it carries three times more than with the zero buffer controller, and 30% more than with a proposed equivalent bandwidth technique—the latter of which fails when given non-standard traffic and thus is not a viable alternative.

The technique was also tested on data gathered from real traffic sources. On variable rate video data and ethernet data it carried 50% more load than with the best reliable alternative. In both cases, the equivalent bandwidth failed by more than an order of magnitude to meet quality of service levels. With the peak rate controller and ethernet data, the carried load was less than 5% of that with the adaptive controller. Thus we have shown that the adaptive controller can accept significantly more load than with peak rate or other controllers, while still maintaining quality of service despite the variety of complex source types.

Since it is based on directly observing the traffic, it can be argued that this controller is near optimal. More significantly it is robust to different real data traffic types, adapting to both new types of users and changes in signalling protocol. The signalling protocol has additional information such as class of service that unlike these other techniques can be additional information for the adaptive access controller. Also, unlike these other techniques, the adaptive control technique is based on satisfying quality of service no matter how it is defined, whether it is cell loss, delay, delay variation, or any combination.

A path also exists for implementing this on a variety of software and hardware prototype products. Economic analysis shows that the adaptive controller here could reduce the provisioning cost of doubling a network's capacity by 25%.

8. Acknowledgments

Wewould like to thank the many people who reviewed this work and gave useful suggestions, and the many others who donated their computers for the countless simulations.

9. References

[ATM93] *ATM User-Network Interface Specification*, eds. Breault, R., Grace, J., Jaeger, J., Wojnarski, L., ATM Forum, Version 2.4 Aug. 5, 1993.

[Cho94] Choudhury, G.L., Lucantoni, D.M., Whitt, W., "Squeezing the Most Out of ATM," to be presented at the 14th International Teletraffic Congress in France, to be held June 6-10, 1994.

[Bel93] The data can be obtained by ftp from `thumper@bellcore.com`, in the directory `vbr.video.trace`.

[Che92] Chen, X., Leslie, I.M., "Neural Adaptive Congestion Control for Broadband ATM," *IEE Proc.-I*, vol. 139, no. 3, pp. 233–240, 1992.

[Dem90]Demers, A., Keshav, S., Shenker, S., "Analysis and Simulation of a Fair-Queueing Algorithm," in *Internetworking: Research and Experience, Vol. 1,* John Wiley and Sons, 1990. pp. 3–26

[Est94] Estrella, A.D., Jurado, A., Sandoval, F., "New Training Pattern Selection Method for ATM Call Admission Neural Control," *Electronic Letters*, vol. 30, no. 7, pp. 577–579, 1994.

[Gar93] \Garrett, M.W., *Contributions Toward Real-Time Services on Packet Networks*, Ph.D. Dissertation, Columbia University, May 1993.

[GR94] *Generic Requirements for Operations of ATM NEs: Performance Management Requirements*, Bellcore GR–1248–CORE, Issue 1, August 1994.

[Gue91] Guerin, R., Ahmadi, H., Naghshineh, M., "Equivalent Capacity and its Application to Bandwidth Allocation in High-Speed Networks," *IEEE JSAC*, vol. 9, no. 7, pp. 968–981, 1991.

[Hir90] Hiramatsu, A., "ATM Communications Network Control by Neural Networks," *IEEE Trans. on Neural Networks*, vol. 1, no. 1, pp. 122–130, 1990.

[Jay92] Jayakumar, A., Alspector, J., "A Cascadable Neural Network Chip Set with On-Chip Learning using Noise and Gain Annealing," in the *Proc. of the 1992 IEEE Custom Integrated Circuits Conf.* pp. 19.5.1–19.5.4.

[Kri94] Krishnan, K.R., "Queueing Models for Effective Bandwidth of Markovian On-Off Traffic Sources," Bellcore TM-ARH-024004. Feb. 25, 1994.

[Lel93] Leland, W.E., Taqqu, M.S., Willinger, W., Wilson, D.V., "On the Self-Similar Nature of Ethernet Traffic," in Proceedings of ACM SIGCOMM 1993. pp. 183–193.

[Mil93] Milton, J., "Broadband ATM Small Switch vs. Large Switch Analysis," Bellcore special report SR-TSV-002913 Dec., 1993

[Mit88] Mitra, D., “Stochastic Theory of a Fluid Model of Producers and Consumers Coupled by a Buffer,” *Advance. Appl. Prob.*, vol. 20, pp. 646–676, 1988.

[Ng91] Ng, K., Lippmann, R.P., “A Comparative Study of the Practical Characteristics of Neural Network and Conventional Pattern Classifiers,” in *Neural Information Processing Systems 3*, ed. Lippmann, R.P., Moody, J.E., Touretzky, D.S., Morgan Kauffman, San Mateo CA, 1991, pp. 970–976.

[Sti90] Stinchcombe, M., White, H., “Approximating and Learning Unknown Mappings Using Multilayer Feedforward Networks with Bounded Weights,” in *Proceedings of International Joint Conference on Neural Networks*, IEEE, 1990, pp. III-7–16.

[Wil94] Willinger, W., Private Communication 2/94.

Timothy X Brown
Adaptive Sytsems Research Group

Joshua Alspector
Adaptive Systems Research Group