

Statistical Classification Based Admission Control

Timothy X Brown

Electrical and Computer Engineering Dept.

University of Colorado
Boulder, CO 80309-0425
timxb@colorado.edu

ABSTRACT

This paper introduces methods based on statistical classification that allow arbitrary measurement features to be incorporated into admission control decisions. Our results show that for high target packet loss rates, nearly any set of features can control the packet loss rate well, while for low target packet loss rates more features provide better control. The methods demonstrate relatively high accuracy in controlling the packet loss rate for both high and low target loss rates and for both memoryless and heavy-tailed traffic distributions. These results represent significant improvements on prior methods and suggest new directions for future research.

1. INTRODUCTION

One approach to meeting quality of service (QoS) is to control access to the network so that network resources are not overloaded. Admission control is the process of deciding what connections to admit and which to reject. Research on admission control has been intense over the past decade and a number of solutions have been proposed. Two broad approaches include model-based and measurement based admission control. As we will show, these solutions typically have a number of deficiencies including difficult to set parameters, sub-optimal utilization, poor scaling to large networks or they are not robust to variations in the traffic.

The purpose of this paper is to present a statistical classification approach to admission control that addresses these deficiencies and provides a unifying framework for model-based and measurement-based admission control. Simulation experiments demonstrate the effectiveness of the method.

2. ADMISSION CONTROL

This section describes the problem model, the goals and challenges in admission control, and then discusses model-based, measurement-based, and statistical classification-based admission control.

2.1. The Problem Model

To make the discussion in this paper concrete, we focus on a specific model for the network, QoS, and traffic. The model, shown in Figure 1, is chosen as the simplest model that demonstrates the key aspects of the QoS methods in this paper. The network is simply a single link connecting two nodes. The link has a fixed bandwidth. A FIFO queue at the head of the link holds excess packets when more traffic arrives than can be sent on the link. The link is loss-less and all sent data is received without error or loss. In ATM, a setup packet allocates resources link by link.²⁴ In RSVP routers allocate bandwidth resources on individual links.²⁶ In weighted fair queueing, each flow over a link is given a virtual queue and bandwidth resources are divided among flows.¹⁰ Therefore, although a single link is quite simple, understanding QoS on a single link is fundamental to more general network architectures.

The traffic consists of many flows that generate packets over time. Flows arrive and depart over time and connections are established via ATM or RSVP. The connection arrival and traffic processes are from an unknown but fixed underlying distribution.

At any time, the network is in a state, \mathbf{x} , defined by the current combination of flows. This state can be very specific, such as a precise description of each flow, or, it can be a simple summary statistic about the traffic such as the total load. It can include additional information such as the measured load, the size of the queue, the time of day, etc. Conceptually, we can treat any state space. The more information the greater the fidelity of the resulting QoS functions.

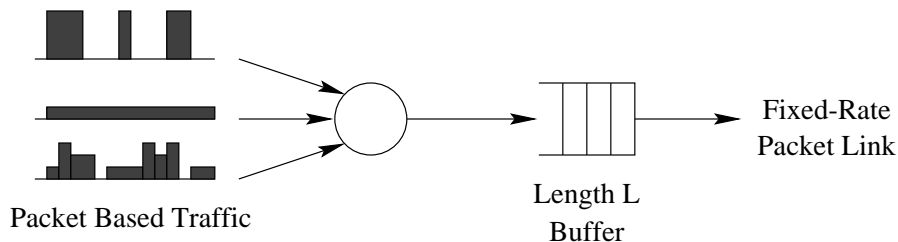


Figure 1. Simplified 1-Link Network Model.

Many QoS metrics can be considered including maximum packet loss rates, average delay, maximum delay, or delay variation (jitter). To demonstrate the main ideas of the paper without bogging down in details, we consider two equivalent forms of QoS. For time-elastic traffic, packets are lost when they arrive to a full buffer. QoS is defined by the rate of packet losses, p . For real-time traffic, a maximum wait time is specified. Since the link rate is fixed, a maximum wait time corresponds to a maximum buffer size. Packets which arrive to a queue which exceeds this maximum will be overdue and are dropped. QoS is defined by the rate of overdue packets, p . Thus, specifying a maximum buffer size, L , yields an equivalent form of QoS for either type of traffic.

The network provides QoS guarantees in the form of a maximum packet loss rate, p^* . To facilitate the admission controller, the network provides monitoring information about the number of packet arrivals and the number of packets lost between each state transition.

In summary, we consider a single link with a fixed bandwidth and QoS is defined by the rate of packet losses, p , due to packets blocked from entering the finite queue. The link and traffic are defined by the buffer length, L , and the distribution of traffic processes.

2.2. Goals and Challenges

When a connection arrives and the network state is \mathbf{x} , the admission control function is to decide whether to accept or reject the connection given \mathbf{x} . An admission controller can have several desirable properties:

Scalable: The controller scales to many connections over a link. The per connection processing and storage is small.

Robust: The controller performs well under many different traffic types.

Efficient: The controller admits the maximum number of connections within the QoS constraints.

Calibrated: The controller parameters should have some meaning to the network operator.

The difficulties in reaching this goal are manifold. The traffic is known to have long-range dependencies that are difficult to model and do not yield simple network control analysis.^{15,21,23} The traffic carried is likely to be a heterogeneous mixture of traffic types including elastic-asynchronous (e.g. e-mail), elastic-interactive (e.g. web browsing), and real-time traffic (e.g. voice).³ Many sources have very poor estimates of their own traffic parameters. For instance, the number of packets sent on an ethernet link varies dramatically from minute to minute and short term measurements are very poor estimates of the overall traffic load.⁷ Different sources often have strong correlations. For example, the voice and video of a video conference may be sent as separate flows while the activity on these flows is highly correlated.

With heterogeneous traffic types, the admission controller must have a policy for deciding how to treat the different traffic types. Finding such policies is a rich research area in itself. For simplicity this paper only considers homogeneous traffic types.

Network complexities also impede a solution. For instance, most network routers are not simple FIFO queues. The server often has a number of queues for different interfaces that share switching resources using strategies that can differ from node to node. They implement queueing protocols such as weighted fair queueing or some form of priority.¹⁰ The queueing and commingling of different traffic types shapes the traffic. Traffic that enters the network at a constant bit rate can leave the network with bursty characteristics.²⁰ In networks that include wireless links to end users, the link bandwidths can vary over time due to noise, interference, and user mobility.

With these traffic and network challenges, we are unlikely to derive simple closed-form solutions that enable robust and efficient admission control decisions. Two categories of admission controllers include model-based and measurement-based admission controllers.

2.3. Model Based Admission Control

In model based admission control the traffic is characterized as coming from some model, and a controller that effectively meets the QoS criteria is derived for this criteria for specific traffic parameters. For example, the trivial peak-rate controller assumes that the traffic is always sent at its peak rate, the traffic parameter is the peak rate, and the admission controller accepts traffic as long as the sum of peak rates is less than the link rate. While robust and scalable, it fails on efficiency. More sophisticated models include assuming the traffic is ON-OFF as described in Section 4 with exponential holding times.^{8,11,16} These are shown to fail when traffic is not from this model so they is not robust.⁷ The traffic features described in the previous section make it unlikely that any simple and robust admission control functions will be found based on traffic modeling. Even if they could, traffic usage patterns are evolving over time and today's model may not apply next month or next year. Conservative approaches such as assuming that each source transmits at its peak rate can be very inefficient when the ratio of sources peak to average rates are large.

2.4. Measurement Based Admission Control

Measurement based admission controllers have been proposed for controlled-load type services since they achieve higher network utilization than methods based on worst-case bounds for traffic that can accept occasional delay or packet loss violations (see ^{4,18} and references therein for more detailed discussion). The methods use measurements of current loads to decide what connections to accept. For our purposes, the methods differ mainly in what information is used to make the admission control decision, and admission decision function on this information. The algorithms use simple per-flow information so that they can scale. The study in⁴ found that overall the different measurement based admission controllers perform similarly well. In particular, they performed well with long-range dependent traffic which has significant load correlations over time that can be exploited. For memoryless traffic, the measurement-based performance is lower than a so-called ideal controller which is substantially similar to the admission controller presented in this paper. None of the measurement based methods are well calibrated and consistently miss target loss rates by orders of magnitude. Since both model-based and measurement-based admission controllers do not meet all the desirable admission controller properties, we seek ways of improving on these methods so that they are consistently robust, efficient, and well calibrated while retaining their scalability.

2.5. Adaptive Framework for Controlling Packet Loss Rate

This section describes a framework for adaptive admission control methods. Packets arrive in the system, are queued, and under congestion are lost. The rate of losses, $p(\mathbf{x})$, depends on the system state, $\mathbf{x} \in R^d$. The state, \mathbf{x} , contains features such as the current flows, the state of the queue, time of day, etc. Monitoring information on the number of loss events, s , out of a total number of events, T , at different conditions, \mathbf{x} , is available from the network in a data set. The data consist of N observations or samples,

$$\mathbf{D} = \{(\mathbf{x}_n, s_n, T_n)\}_{n=1}^N. \tag{1}$$

We use this data in statistical classification. A statistical classifier is a function $C(\mathbf{x}, \mathbf{w})$ that maps from \mathbf{x} to $\{-1, +1\}$. It indicates whether \mathbf{x} has a specific property or not: for instance, whether all the carried connections when the system is in state \mathbf{x} meet their QoS requirements. With this function, the admission controller accepts or rejects a connection if $C(\mathbf{x}, \mathbf{w})$ is $+1$ or -1 respectively. The function is estimated from the data set \mathbf{D} and is parameterized by weights \mathbf{w} .

Classification requires a functional form for the mapping. For instance, the linear threshold classifier uses the form:

$$C^{lin}(\mathbf{x}, \mathbf{w}) = \text{sign}\left(\sum_{i=1}^d w_i x_i + w_0\right). \tag{2}$$

The parameters of $C(\mathbf{x}, \mathbf{w})$ are found by minimizing an objective function, J which we will define later:

$$\mathbf{w}_D = \arg \min_{\mathbf{w}} J(\mathbf{D}, \mathbf{w}). \quad (3)$$

This process is denoted as *training*.

Thus, the adaptive systems that we consider use prior system performance as a function of system state to derive admission control decision functions. The rate of adaptation depends on the application. Data is collected and updates performed over long periods such as days or weeks. The goal is to capture long term changes to traffic distributions and not to track short term traffic variations.

Prior work in this area has focussed on problems where the average number of packets lost is large, $E[s] = pT \gg 1$, e.g.,^{12,17,29} where $E[\cdot]$ stands for expectation. In this case, s/T is a reliable measure of p . As shown in⁶ almost any method works well in this regime. This regime is not practical. The region of interest is where p is very small (e.g., $p = 10^{-6}$), implying the number of samples, T , must be large. While individual sources may be present for long periods, the aggregate set of sources may be changing often, limiting T . For example, in this paper's experiments, the connection holding time is 300 seconds. But, for the aggregate traffic on the link, a new connection arrives or departs twice a second. The packet losses are correlated so that the effective number of independent packets is much smaller than T . Therefore, $pT \ll 1$ with the result that the data has a high variance and in fact most data will have no losses ($s = 0$). The prior methods used s_n/T_n as a proxy for $p(x_n)$ in statistical classification. The resulting decision functions are inconsistent, and when sample sizes are small, make gross errors.

The author has several prior papers on statistical classification applied to admission control.^{5-7,28} These papers first develop the basic method which is summarized in the next section for completeness.^{5,6} They emphasize the role of adaptation in developing accurate admission controllers and shows cases with mis-specified parameters and correlations under which the controller is still accurate.⁷ They go further and look at methods for developing accurate admission control policies under utilization and fairness criteria.²⁸ The paper here builds on this prior work to show statistical classification's merits as a unifying framework for model-based and measurement-based admission control.

2.6. Paper Overview

Using the adaptive framework applied to the one-link model, the next section derives statistical classification methods for admission control. The methods are applied to several scenarios in Sections 4 and 5 where we see that the unified statistical classification approach yields more robust, efficient, and calibrated solutions. Implementation issues are discussed in 6 where we show that the methods are scalable. Conclusions appear in Section 7.

3. STATISTICAL CLASSIFICATION FOR ADMISSION CONTROL

This section derives consistent estimators to admission control decision functions that classify under what states the network will meet or not meet the QoS guarantees. Such a decision function can form the basis of an admission control algorithm. The network gives monitoring data in the form of (1) from which we estimate the decision function. To start, we assume that each packet sent is lost independent of any other so that whether a packet is lost is a Bernoulli trial. This, of course, is not a realistic assumption but we will return to this point in Section 3.2. With this assumption, the losses in sample n are distributed as:

$$\Pr(S = s_n | \mathbf{x}_n, T_n) = \binom{T_n}{s_n} (p(\mathbf{x}_n))^{s_n} (1 - p(\mathbf{x}_n))^{T_n - s_n}, \quad (4)$$

where $p(\mathbf{x}_n)$ is the underlying but unknown probability that a packet is lost when the state is \mathbf{x}_n . We will use this model to develop a loss rate classifier and then show how to relax the independence assumption.

3.1. Classifying Loss Rates

We look at the problem of deriving a function $C(\mathbf{x})$ that classifies whether $p(\mathbf{x})$, the losses rate at \mathbf{x} , is above or below a threshold, p^* . Such a function would enable the network to decide when it can and can not provide QoS guarantees. The ideal decision function has the form:

$$C^*(\mathbf{x}) = \text{sign}(p^* - p(\mathbf{x})) = \begin{cases} +1 & p(\mathbf{x}) < p^* \\ -1 & p(\mathbf{x}) > p^* \end{cases}. \quad (5)$$

We only can observe the data set \mathbf{D} and not the loss rate, $p(\mathbf{x})$. Let $C_{\mathbf{D}}(\mathbf{x})$ be a decision function estimated from the data \mathbf{D} that maps from R^d to $\{-1, +1\}$. The estimator, $C_{\mathbf{D}}(\mathbf{x})$, is consistent if given any \mathbf{x} :

$$\lim_{|\mathbf{D}| \rightarrow \infty} \Pr\{C_{\mathbf{D}}(\mathbf{x}) \neq C^*(\mathbf{x})\} = 0, \quad (6)$$

where the probabilities are over the distribution of \mathbf{D} .

This paper approaches the problem via statistical classification. A statistical classifier is given a training set, $\mathbf{Y} = \{(\mathbf{x}_n, o_n, \alpha_n)\}$, consisting of feature vectors, \mathbf{x}_n , with corresponding desired output classification, $o_n \in \{+1, -1\}$ and positive real-valued sample weight, α_n . For many applications all samples are weighted equally and the weight is disregarded. A classification function, $C(\mathbf{x}, \mathbf{w})$, parameterized by a real-valued vector \mathbf{w} , divides the feature space into positive and negative regions separated by a decision boundary and can be used to classify future feature vectors. Based on the training set, a classifier (i.e., \mathbf{w}) is selected that minimizes some objective function, $J(\mathbf{D}, \mathbf{w})$. An alternative, indirect approach to this problem would be to estimate $p(\mathbf{x})$ first and then derive a loss rate classifier from this estimate. The methods in this section allow us to estimate the classifier directly with fewer parameters than the estimate of $p(\mathbf{x})$.

Thus, the estimator is defined by: mapping from a data set, $\mathbf{D} = \{(\mathbf{x}_n, s_n, T_n)\}$, to a training set, $\mathbf{Y} = \{(\mathbf{x}_n, o_n, \alpha_n)\}$, where $|\mathbf{D}| = |\mathbf{Y}|$; choosing a parameterized decision function model, $C(\mathbf{x}, \mathbf{w})$; and selecting model parameters, \mathbf{w} , by minimizing an objective function, $J(\mathbf{D}, \mathbf{w})$.

A common objective function used in statistical classification is the weighted ℓ -norm objective function ($\ell > 0$):

$$J(\mathbf{D}, \mathbf{w}) = \left(\sum_{n=1}^N \alpha_n |C(\mathbf{x}_n, \mathbf{w}) - o_n|^\ell \right)^{1/\ell}. \quad (7)$$

For instance, $\ell = 2$ is the weighted RMS error between the classifier and desired output. At a single point where $\mathbf{x}_n = \mathbf{x}$, C is fixed and it can be shown that the estimator that minimizes (7) is⁶:

$$C_{\mathbf{D}}(\mathbf{x}) = \text{sign} \left(\sum_{n=1}^N \alpha_n o_n \right). \quad (8)$$

The estimator defined by (8) is consistent (as in (6)) if $\text{sign}(E[\alpha_n o_n]) = \text{sign}(p^* - p(\mathbf{x}_n))$ and $\text{Var}[\alpha_n o_n] < \infty$ where expectations are taken with respect to possible values of α_n and o_n at \mathbf{x}_n . This follows directly from the weak law of large numbers.

This defines our objective function. We next turn to a definition of α_n and o_n (i.e. a mapping from \mathbf{D} to \mathbf{Y}). Given a data set, $\mathbf{D} = \{(\mathbf{x}_n, s_n, T_n)\}$, where $\mathbf{x}_n = \mathbf{x}$ for all n , and assuming the s_n are distributed as in (4), the minimum variance unbiased and efficient estimator of $p(\mathbf{x})$ is given by²⁵:

$$p(\mathbf{x}, \mathbf{w}) = \frac{\sum s_n}{\sum T_n}. \quad (9)$$

We want to know when $p(\mathbf{x}) > p^*$. Using the estimator, (9) this is when

$$\frac{\sum s_n}{\sum T_n} > p^*. \quad (10)$$

This leads to the estimate:

$$C_{\mathbf{D}}(\mathbf{x}) = \text{sign} \left(\sum_{n=1}^N T_n p^* - s_n \right) \quad (11)$$

Comparing with (8), the minimum of (7) is a consistent estimator when:

$$\begin{aligned} \alpha_n &= |T_n p^* - s_n| \\ o_n &= \text{sign}(T_n p^* - s_n) \end{aligned} \quad (12)$$

So far we have defined an objective function and mapping from monitor data to training data that yields a consistent estimator at a single point. The optimal form of the classifier function $C(\mathbf{x}, \mathbf{w})$ is simply a constant. The more important question is whether or not a classification function can be developed that is consistent across all \mathbf{x} . Several considerations are important. First, not all \mathbf{x} are relevant. Some \mathbf{x} may have zero probability associated with them from the underlying sample distribution (they are *unsupported*) so we will only view \mathbf{x} that are supported. Second, the classifier function $C(\mathbf{x}, \mathbf{w})$ may not be able to represent the consistent estimator. For instance, the linear estimator will never yield a consistent estimate if the $C^*(\mathbf{x})$ decision boundary is non-linear. So, we require there exist a parameter set, \mathbf{w}^* , so that $C(\mathbf{x}, \mathbf{w}^*) = C^*(\mathbf{x})$ for all supported \mathbf{x} . Given this condition on the form of the classifier function, minimizing the objective function (7) using the weights and desired outputs in (12) yields a consistent estimator. This was proved in.⁶

The weight vector is found by first computing the training set \mathbf{Y} from \mathbf{D} using (12) and then minimizing the objective function in (7) using $\ell = 2$. To minimize the objective function an iterative gradient descent method is used:

$$\mathbf{w}(m+1) = \mathbf{w}(m) + \eta_m \Delta \mathbf{w}(m), \quad (13)$$

$$\begin{aligned} \Delta \mathbf{w}(m) &= - \left. \frac{\partial J}{\partial \mathbf{w}} \right|_{\mathbf{w}=\mathbf{w}(m)} \\ &= \sum_{n=1}^N \alpha_n(m) e_n(m) \delta_n(m), \end{aligned} \quad (14)$$

where the index m refers to iteration number, η_m is the learning rate parameter, $\alpha_n(m) = \alpha_n$ is defined by \mathbf{Y} , and

$$e_n(m) = o_n - C(\mathbf{x}_n, \mathbf{w}(m)), \quad (15)$$

$$\delta_n(m) = 2 \left. \frac{\partial C(\mathbf{x}_n, \mathbf{w})}{\partial \mathbf{w}} \right|_{\mathbf{w}=\mathbf{w}(m)}. \quad (16)$$

The architecture used in this paper is the linear classifier in (2). The sign function has only zero gradient. For the purposes of training, the $\text{sign}(\cdot)$ function in the linear classifier is replaced with the $\tanh(\cdot)$ function:

$$\tanh(z) = \frac{1 - e^{-2z}}{1 + e^{-2z}} \quad (17)$$

which allows a gradient to be computed in (16).

In this section, we have derived a consistent packet loss rate classifier. Other classifiers, such as classifiers of average packet delay can be similarly derived.

3.2. The Independence Assumption

An assumption in the classification models is that the packet losses within each sample are independent. This is not true since if an arriving packet is lost due to buffer overload, it is likely that packets arriving at nearby times will also be lost. Analysis in⁶ shows that the correlation in packet losses quickly decay as the separation in packet arrival times grows. This applies to the self-similar processes in²¹ where, although the autocorrelation function is not a negative exponential function, it still decreases hyperbolically. This suggests that suitably independent trials can be generated by subsampling the arriving packets at intervals of I so that the subsampled number of trials $T' = T/I$ and the expected number of subsampled losses is $E[s'] = s/I$. Observing (12) for classification and taking expectation over possible subsamples, the effect of using the subsampling is to scale α_n by a constant $1/I$ which has no significant effect on the training. Thus, the method can be applied even with correlations in the packet losses.

4. EXPERIMENTAL FRAMEWORK

The experiments in this section are modeled after the experiments in^{4,18} so that direct comparisons could be made. It is based on the model in Figure 1. We perform two classes of experiments. The first tests whether using more information in making connection access control decisions yields any performance improvement. The second tests whether the statistical classification based admission controller can accurately control the packet loss rate. Before describing these experiments in detail we define this paper's link, traffic, and measurement models.

Table 1. Traffic source parameters for EXP and PAR.

model	$1/\lambda$ (sec)	m_{hold} (sec)	m_{ON}, m_{OFF} (msec)	r (bps)
EXP	0.4	300	325	64k
PAR	0.4	300, $\sigma = 5\text{dB}$	325, $s = 1.1$	64k

4.1. Link and Traffic Models

The link has rate 10Mbps. The sources generate fixed-size packets of length 128 bytes and the buffer is 160 packets long. The buffer discipline is FIFO with tail drop. Any packet arriving to a full buffer is dropped.

Potential connections arrive over time and are accepted or rejected by one of the admission control algorithms. Once accepted the connections generate traffic as an ON/OFF process that alternates between transmitting at rate, r , and not transmitting at all. The ON periods and OFF periods are chosen from specific distributions and the period length is independent of previous periods. The traffic is not subject to policing or traffic shaping.*

We consider two classes of connections, denoted EXP and PAR. In both cases, the arrival process is a Poisson process with inter arrival time $1/\lambda$. The holding time depends on the type of traffic. In the EXP class, the connection holding times are exponential. The ON and OFF periods are both exponential. The ON and OFF periods are identically distributed so that the average rate is half of the peak rate.

Since simple Markov sources, such as EXP, have been shown to not fully capture the impact of traffic on networks, we include a traffic model with heavy-tailed distributions.^{9,15,21,23} The PAR model uses a heavy-tailed, log-normal holding time model based on² with shape parameter σ .[†] The ON and OFF periods are Pareto distributed with shape parameter s .[‡] The actual parameters used are listed in Table 1.

4.2. Measuring the Load

The load measurement procedure is an exponential average. The load is measured over a sample period t_S . Then the sample, l_t^S , is used to update a load estimate:

$$\hat{l}_t = (1 - \gamma) \cdot \hat{l}_{t-1} + \gamma \cdot l_t^S$$

In our experiments, $t_S = 10\text{msec}$ and $\gamma = 0.0007$. This corresponds to an averaging time constant of 10sec. This time is determined as follows.

With the link parameters, and the traffic parameters in Table 1, the simulations show that approximately 300 connections are carried. With a holding time, of 300 sec, on average one of the 300 connections leaves every second. Thus a decision to accept one more connection will increase the load for 1 sec on average and we need enough information to estimate the impact over at least the coming second to make our decision now. Any past measurement will give an unbiased measurement of the future for the exponential ON/OFF traffic. For self-similar traffic, Norros suggest in²² that data from the past t sec be used to estimate the behavior in the next t sec. We conservatively use a $t = 10$ second time scale for all experiments.

*Or, equivalently, the token bucket filter has rate equal to the peak rate, r .

[†]In the log-normal model, the log of the holding times is normally distributed. If m_{hold} is the median holding time, then let Xm_{hold} be a holding time instance where:

$$p(X = x) = \frac{1}{x\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{\ln x}{\sigma}\right)^2}$$

One standard deviation includes from $1/\sigma$ to σ . The shape is sometimes expressed in dB. If σ_{dB} is expressed in dB, then $\sigma = \ln 10\sigma_{\text{dB}}/10$. For $\sigma_{\text{dB}} = 5\text{dB}$, the mean holding time is approximately twice m_{hold} .

[‡]For the Pareto distribution,

$$\text{Prob}\{\text{period} > x\} = \begin{cases} \left(\frac{xs}{m(s-1)}\right)^{-s} & \text{if } x > \frac{m(s-1)}{s} \\ 1 & \text{otherwise} \end{cases},$$

where m is the mean period length.

Table 2. Feature vectors used in statistical classifier.

Case	Features
1	ml
2	pr
3	ql
4	ml,pr
5	ml,ql
6	pr,ql
7	ml,pr,ql

ml = measured load
pr = sum of peak rate
ql = queue length

Newly accepted connections will increase the future load but not impact the load measurements immediately. To conservatively account for the effect of a new connection with peak rate r ,

$$\hat{l}_t := \hat{l}_t + r$$

immediately upon acceptance.

4.3. Performance Frontiers:

The statistical classifier uses the functional form in equation (2). We form a feature vector consisting of the measured load, sum of peak rates (of carried connections plus the new connection), and the queue size at connection time. This is multiplied by a set of weights and compared to a threshold.[§] Different classes of admission controllers can be formed by using different subsets of these features. Table 2 lists the combinations used in this paper.

Several measurement based admission controllers reduce to a simple threshold on the measured load, so they are equivalent to Case 1.¹⁸ ¶ Though some are not thresholds on the measured load, their performance is similar.⁴

For a homogeneous call types, many model-based admission controllers reduce to a threshold on the number of calls.^{16,19} Since the total peak rate, is proportional to the number of carried calls, these are equivalent to Case 2. Therefore, we can compare the statistical classifier based admission control with measurement based admission control via Case 1 and can compare with model based admission control via Case 2.

We first ask whether any setting of parameters for any of the other cases will improve the performance compared to other controllers. To test this we use the notion of a *performance frontier*.

Every set of parameters yields an admission controller. For any given link and traffic model, this admission controller yields a specific link load and loss rate.^{||} In general, more load, at a lower loss rate is better.

The frontier is found by plotting the loss rate vs. load for many combinations of the parameters and observing the combinations that yield the lowest loss rate for a given load. Every data point is derived from a simulation over 100,000 seconds (28.4 hours). Two simulations are made, one for the EXP and one for the PAR data defined in Table 1.

4.4. Statistical Classifier Training and Testing:

The statistical classifier requires a training set (as in (1)) to find the loss rate as a function of load. To generate training data, we first simulate traffic that generates arrivals and we accept all connections. The arrival rate is reduced so as to generate an acceptable load for the accept all policy. At connection acceptance, we record the measured load, the sum of carried peak rates, and the queue length. Since the control variable is the loss rate, the number of packet arrivals and the number lost between acceptance and until the next departure or accepted arrival is recorded. For each traffic type, EXP and PAR, 200,000 samples are collected. Each sample is small with $T = 2000$ packets typical.

[§]In (2), the threshold appears as an offset w_0 and the entire sum is compared to zero. Scaling the parameters has no effect on the decision. Further, we assume that all the parameters are negative, i.e. we are less likely to accept a call as the measured load, sum of peak rates, or the queue size increases. Therefore, without loss of generality, we assume $w_0 = 1.0$.

¶The measurement-based methods are equivalent in the decision mechanism, they differ in how the threshold is found.

^{||}The loads in this paper are the offered loads measured at the input to the node (before any losses). The loss rates are the fraction of offered load that are dropped due to a full buffer.

This data is used to form various training sets from the EXP and PAR data. The training sets differ in the variables that constitute the feature vector, \mathbf{x} . The seven cases in Table 2 are used. The classifier model is the linear threshold in (2). For each training set, a loss rate threshold, p^* , is chosen and parameters are found using the method in Section 3.1. To see the performance over a range of target loss rates, classifiers are trained for $p^* = 10^{-2}$ and $p^* = 10^{-6}$. In total 28 different classifiers are trained, for the combinations of 2 loss rates, 7 feature vectors, and 2 traffic types.

Each classifier is tested against its target traffic type, EXP or PAR. A classifier test consists of simulating traffic for 100,000 secs and accepting or rejecting connections according to the classifier. The loss rate and measured packet load is recorded. The next section describes the results.

5. RESULTS

In this section we describe two sets of results. The first compares the performance frontiers of the different classifiers. The second looks at how well the statistical classifier can control the packet loss rate.

5.1. Performance Frontier Results:

Figure 2 and Figure 3 show the performance frontier data for the seven cases and two traffic types. A note about the figures. Case 1 points (labeled “ml”) are simply samples where the parameter for the peak rate and queue size are both zero. Similarly for the other cases. Thus, Case 7 (“ml,pr,qs”) includes all the points since we always have the option of setting a parameter to zero. similarly, Case 6 (“pr,qs”) includes the Case 2 (“pr”) and Case 3 (“qs”) points, etc.. Thus, we see that Case 7 must be at least as good as any other case. Case 6 must be at least as good as Case 2 and Case 3, etc.. For this reason and for completeness, we plot all the data points even ones that are not on the frontier.

We first observe that for both traffic at high packet loss rates (i.e. $> 10^{-3}$), all feature vectors perform equally well. Only minor differences ($< 30\%$) exist between the best and worst loss rates at a given load. Since the queue size is readily available, requires no monitoring or measurement mechanisms, and requires no connection state information to be stored, it is likely the simplest control mechanism. Unfortunately, the queue size can not control at all at low loss rates. For instance, the most conservative queue size based admission controller is if we set the queue size threshold at zero (only accept calls if the queue is empty.) But, at this level the loss rate is still approximately 10^{-4} for the EXP traffic and 10^{-3} for the PAR traffic.

For low packet loss rates (i.e. $< 10^{-4}$), we see clear differences between the performance frontiers with different admission, controllers and with different traffic. For the EXP traffic, the performance frontier is populated by Cases 2, 4, 6, and 7, i.e. all the cases that include the peak rate feature. This suggest that peak rate is most useful for predicting packet loss with this traffic. But, all features contribute to the performance frontier.

For the PAR traffic, the best performance is Case 4 (“ml,pr”), although at the lowest loss rates, Case 7 performs best. It is interesting to note that with the PAR traffic, unlike the EXP traffic, the peak rate feature alone performs particularly bad while the measured load feature performs better. In summary, at high loss rates, any and all features perform equally well so use whatever feature is available. At low loss rates different sets of features work better under different traffic suggesting that the most robust controller will include all available features and adapt to the traffic at hand.

5.2. Statistical Classification Results:

The statistical classifier admission controller attempts to meet a target packet loss rate. Table 3 shows the performance on the EXP and PAR traffic. For the target loss rate, $p^* = 10^{-2}$, all of the measured loss rates are within a factor of 3 of the target loss rate. For the target loss rate, $p^* = 10^{-6}$, the measured loss rates are within a factor of 5 for all but two cases. The two exceptions are factors of 9 and 30 larger. Case 3 is not included since as noted earlier, the queue size can not control to less than 10^{-4} target loss rate.

To put these numbers in perspective, results on the same experiment for a measurement based admission controller are reported in.⁴ When $p^* = 10^{-2}$, the measured loss rates differed by up to a factor of 50 with 4 being typical. When $p^* = 10^{-6}$, the measured loss rates differed by factors from 19 on EXP traffic to 13,000 on PAR traffic.

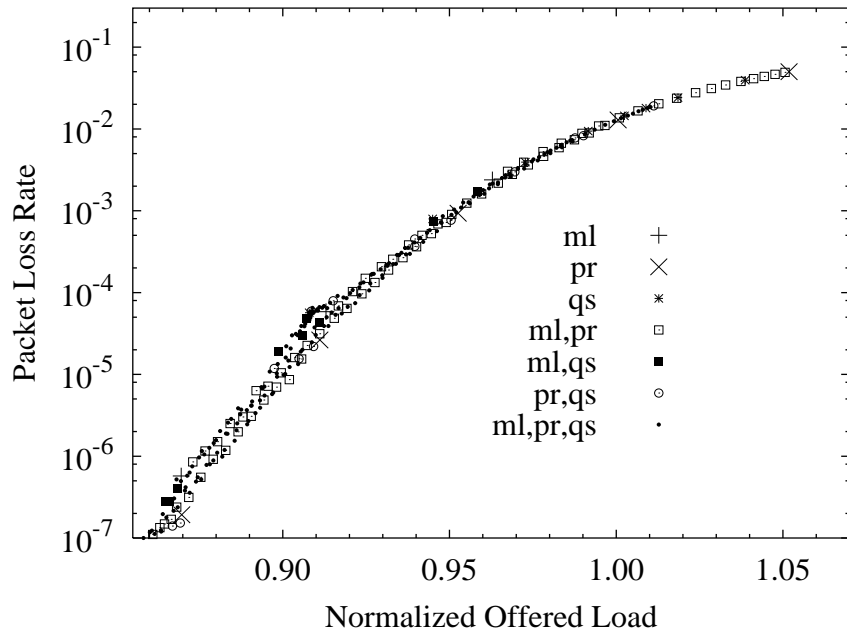


Figure 2. Plot of loss rate as function of load on EXP traffic (Poisson arrivals, exponential holding times, and exponential ON and OFF periods). The admission controllers differ in the combination of features they used to make their decision. ml = measured load, pr = sum of connection peak rates, qs = the size of the queue.

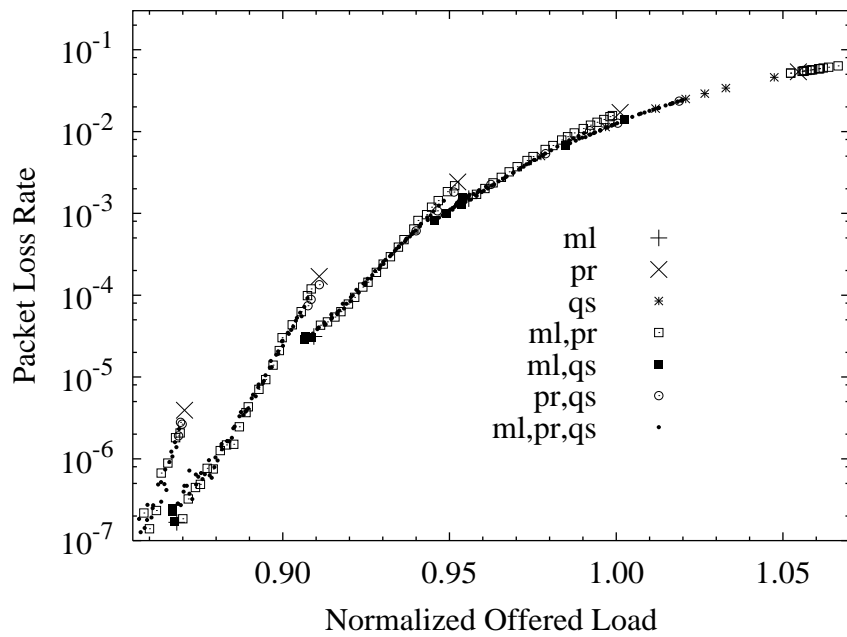


Figure 3. Plot of loss rate as function of load on PAR traffic (Poisson arrivals, log-normal holding times, and Pareto ON and OFF periods). The data striping is an artifact of the parameter sampling.

Table 3. Statistical classifier based admission controller loss rates.

Case	$p^* = 10^{-2}$		$p^* = 10^{-6}$	
	EXP	PAR	EXP	PAR
1 ml	0.75×10^{-2}	1.2×10^{-2}	0.70×10^{-6}	0.87×10^{-6}
2 pr	1.1×10^{-2}	0.98×10^{-2}	0.82×10^{-6}	1.7×10^{-6}
3 qs	2.2×10^{-2}	2.8×10^{-2}	n/a	n/a
4 ml,pr	1.2×10^{-2}	1.2×10^{-2}	0.61×10^{-6}	2.6×10^{-6}
5 ml,qs	1.8×10^{-2}	0.32×10^{-2}	8.9×10^{-6}	0.23×10^{-6}
6 pr,qs	1.6×10^{-2}	2.9×10^{-2}	4.1×10^{-6}	$33. \times 10^{-6}$
7 ml,pr,qs	1.6×10^{-2}	0.63×10^{-2}	4.7×10^{-6}	1.4×10^{-6}

The statistical classifier is guaranteed to asymptotically** generate an admission controller that approaches the optimal loss rate. The differences between target and measured loss rates here are not only due to the finite training set size. They are also due to the fact that the training set data is generated with a slightly different traffic model than the test data. The training sets are generated with an accept all admission control policy for a reduced Poisson arrival rate. The testing is with the learned admission control policy under the heavy arrival rates in Table 1. This may lead to small but significant differences between the optimal and the learned set of parameters.

One final point to emphasize is that a different admission controller is learned for each of the 28 cases in Table 3. For instance, in Case 7, $p^* = 10^{-6}$, the peak rate is weighted higher than the measured load with the EXP traffic, while with the PAR traffic the relative weighting is reversed.

6. IMPLEMENTATION ISSUES

So far the paper has described performance comparisons in a simulated environment. If the statistical classifier based admission control are to be implemented on-line in a network, several major issues would need to be addressed. The main issues are decision resources, training resources, and training strategy.

Few decision resources are needed for the statistical classifier. For instance, classification models such as (2) require storage of less than 10 floating point numbers and require less than 10 floating point operations. More sophisticated classification models, such as a neural network-based model,¹ might require 10's of parameters and a similar number of floating point operations. In any case, this is little computation given the millisecond time-scale of admission control decisions. The measurements and features are collected on similar time scales so that the computation for these would also be small.

Though the decisions can be made quickly with few resources, they require parameters derived from training. The training resources are much larger. The training sets used in this paper consist of 200,000 samples collected over one simulated day for the link. The total storage is several megabytes. The training requires several minutes of CPU on a 300MHz processor. More sophisticated classification models might require hours of CPU time. We argue that these resources are not burdensome since storage is cheap (\ll 1\$ per megabyte) and the training would be performed once per day or even less often.

To be concrete, one implementation would collect data over a day, then in the evening would recompute a new set of parameters to be used in the following day. The data and computation could be assigned to any computer on the network. If for any reason a new set of parameters could not be computed, the old set could be carried over to the next day. We want to re-emphasize here that the decision inputs can include short-term measurements collected at the decision time. But, the decision function adapts its parameters using observations over much longer periods.

The training strategy consists of two parts: finding an initial set of parameters, and evolving the parameters over time. How do we find an initial set of parameters, given an uninitiated statistical classifier? Several models could be used. Off-line traffic simulation, as in this paper, could generate training data for an initial model. Data from another link already on-line could also generate training data. Alternatively, the link could use a known reliable admission control strategy such as accepting only if the sum of peak rates are less than the link bandwidth.

Once an initial admission controller is found, training data is collected over some period, and a new set of parameters are found. Two cases can occur. In the first, the admission controller is accepting too much traffic and

** Asymptotic in the training set size.

the loss rates are above the target loss rate, p^* . If the loss rates are only slightly above p^* , then over the collection period, samples above and below p^* will be collected and the decision boundary modified at the next training period. If the loss rates exceed p^* by a large margin, then this could trigger an early training update period to bring the loss rates in line.

The second case is that the admission controller is conservative and the loss rates are below p^* . In this case, no samples will be collected close to p^* and we have no basis for adapting the parameters. For instance if we initialized the admission controller with the peak rate admission control, then no losses would ever be observed. For this reason so-called exploration is important. Several exploration models exist.²⁷ The simplest computes the admission controller's decision, (accept or reject), and then with some probability ϵ takes the opposite decision. Given the many samples, appropriate values might be $\epsilon = 10^{-3}$. In this way, even if we are conservative, we can learn a correct decision boundary.

Another issue related to training strategies is the long term storage problem. We would like to keep data collected in the past so that we have the most data for classifier training.^{††} One day's worth of data on an active 10Mbps link yields a megabyte of training data. A higher-speed link could overwhelm storage and require subsampling or filtering of the data to reduce its volume. For instance, data far from the decision boundary could be discarded. While the data per collection period can easily be kept manageable, over time this can accumulate to an unmanageable amount. Further, the older the data, the less likely it will be relevant. Several authors have explored this issue.^{12,17} The main idea is that every training period, we randomly discard a fraction of the training samples. For small p^* , the samples without losses far exceed the samples with losses. Thus, some care must be taken to keep examples with losses.

These implementation issues are critical to an efficient and effective statistical classification based admission controller. Future work will explore these in detail.

7. CONCLUSIONS

This paper looks at an enriched model for admission control. In our model, we allow arbitrary features to describe the network state at the time of admission. These features can be used or not used in a parametric admission control decision model as specified by data collected over long time periods. Some admission controllers do use multiple data in their decision, but only in specific ways that may or may not apply to specific traffic models.^{13,14,16} In our approach, the consequence of accepting a new call is measured over many acceptances over long periods to accurately determine the effect. These effects are automatically incorporated into the decision model. The methods have theoretical assurances that over time they can achieve an optimal admission control function in ways not readily captured by other methods.

As a contrived example, if sources are ON-OFF and the ON-OFF transitions are perfectly synchronized for all sources, then allowing any traffic beyond peak rate will result in losses that will appear in monitoring samples. When the statistical decision function trains on these samples, it will automatically derive a peak rate admission controller. Any other proposed method would have to know a priori that the correlations exist.

Our experiments show that high target loss rates can be easily controlled with any feature set. In fact, simply looking at the queue size at the time of admission decision is sufficient for the traffic considered here. More interesting is when the target loss rate is small. Here, a feature's importance depends on the traffic. This suggests including more features and letting the statistical classifier sort out how to weight the different features.

The controller is able to control aggregate performance relatively close to the target values. This suggests the methods here could apply to real-time traffic that requires hard QoS guarantees.

Returning to the goals in Section II-B, the statistical classification based admission control is: scalable, the decision and training resources are small; robust, it accurately controlled the packet loss rate across several traffic types; efficient, asymptotically the statistical classifier will yield the optimal decision function; and calibrated, once the operator specifies the target QoS, the statistical classifier finds the optimal decision function for this QoS.

This paper explored three features, the measured load, the sum of peak rates, and the queue size; one measurement model, exponential averaging; one classifier model, the linear threshold function; one QoS metric, packet loss rate; and two traffic models, the memory-less, EXP model and the heavy-tailed PAR model. The features chosen here are

^{††}Recall that asymptotically as we collect more data the statistical classifier can approach the optimal classifier

intended to be exemplary and no claim is made that they are optimal in any sense. The reader can easily imagine better feature sets. The exponential averaging and linear threshold decision model are chosen for their simplicity, more sophisticated models can easily be included as warranted. The packet loss rate QoS metric is chosen since it is a difficult metric control and thus magnifies any admission controller's weaknesses. The traffic models are intended to be extreme models. The EXP model is known to have tractable properties that have motivated many admission control algorithms, while the PAR model is known to be extremely difficult to control. These models are homogeneous and future work will address heterogeneous models. In this discussion, we are suggesting that future research should focus on finding specific features and classification models that are most efficient for admission control.

ACKNOWLEDGMENTS

This work is funded by NSF CAREER Award NCR-9624791.

REFERENCES

1. Bishop, C., *Neural Networks for Pattern Recognition*, Oxford U. Press, Oxford, 1992. 482p.
2. Bolotin, V.A., "Modeling call holding time distributions for CCS network design and performance analysis," *IEEE J. on Selected Areas in Communications* v. 12, n. 3, pp. 433-438, April 1994.
3. Braden, R., Clark, D., Shenkar, S., "Integrated services in the Internet architecture: an overview." *IETF RFC 1633*, July 1994.
4. Breslau, L., Jamin, S., Shenker, S., "Comments on the Performance of Measurement-Based Admission Control Algorithms," in *Proc. of the Conf. on Computer Communications (IEEE INFOCOM)'00*, v.3 pp. 1233-1242, April, 2000.
5. Brown, T.X., "Adaptive access control applied to Ethernet data," *Advances in NIPS 9*, ed. M. Mozer et al., MIT Press, pp. 932-938, 1997.
6. Brown, T.X., "Classifying Loss Rates in Broadband Networks," in *Proc. of the Conf. on Computer Communications (IEEE INFOCOM)'99* v.1, pp. 361-370, 1999.
7. Brown, T.X., "Adaptive Statistical Multiplexing for Broadband Communications," in ed. Kouvatsos, D. *Performance Evaluation and Application of ATM Networks*, Kluwer, 2000, pp. 51-79.
8. Choudhury, G.L., Lucantoni, D.M., Whitt, W., "On the effectiveness of admission control in ATM networks," in the *14th International Teletraffic Congress in France*, June 6-10, 1994. pp. 411-20.
9. Crovella, M.E., Bestavros, A., "Self-similarity in World Wide Web traffic: evidence and possible causes," *IEEE/ACM T. on Networking*, v. 5, n. 6, pp. 835-845, Dec. 1997.
10. Demers, A., Keshav, S., Shenkar, S., "Analysis and simulation of a fair queueing algorithm," *Proc. of the SIGCOMM'89 Symposium*, pp. 1-12, Sept. 1989.
11. Elwalid, A. I., Mitra, D. "Effective bandwidth of general Markovian traffic sources and admission control of high-speed networks," *IEEE/ACM Trans. on Networking*, v. 1, n. 3, June 1993.
12. Estrella, A.D., Jurado, A., Sandoval, F., "New training pattern selection method for ATM call admission neural control," *Elec. Let.*, Vol. 30, No. 7, pp. 577-579, March 1994.
13. Floyd, S., "Comments on measurement-based admission control for controlled-load services," Technical Report, Lawrence Berkeley Laboratory, July, 1996.
14. Gibbons, R.J., Kelly, F.P., Key, p.B., "Measurement-based connection admission control," *Proc. 15th International Teletraffic Conference*, June 1997.
15. Garrett, M.W., Willinger, W., "Analysis, modeling and generation of self-similar VBR video traffic," in *Proc. of ACM SIGCOMM, 1996*. pp. 269-280.
16. 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.
17. Hiramatsu, A., "ATM communications network control by neural networks," *IEEE Trans. on Neural Networks*, vol. 1, no. 1, pp. 122-130, 1990.
18. Jamin, S., Shenker, S., Danzig, P., "Comparison of measurement-based admission control algorithms for controlled-load service," in *Proc. of the Conf. on Computer Communications (IEEE INFOCOM)'97*, (April 1997).
19. Krishnan, K.R., "Queueing Models for Effective Bandwidth of Markovian On-Off Traffic Sources," Bellcore TM-ARH-024004. Feb. 25, 1994.

20. Lee, D.C., "Worst-case fraction of CBR teletraffic unpunctual due to statistical multiplexing," *IEEE/ACM Tran. on Networking*, vol. 4, no. 1, Feb. 1996. pp. 98–105.
21. Leland, W.E., Taqqu, M.S., Willinger, W., Wilson, D.V., "On the self-similar nature of ethernet traffic," in *Proc. of ACM SIGCOMM 1993*. pp. 183–193.
22. Norros, I., "On the use of fractional Brownian motion in the theory of connectionless networks," *IEEE/ACM T. on Networking*, v. 2, n. 1, pp. 1–15, Feb. 1994.
23. Paxson, V., Floyd, S., "Wide-area traffic: the failure of Poisson Modeling," *Proc. of SIGCOMM 94-8/94*, London, UK, pp. 257–268, 1994.
24. Peterson, L.L., Davie, B.S., *Computer Networks: A Systems Approach, 2nd Ed.*, Morgan Kaufmann, SF, p. 748, 2000.
25. Poor, H.V., *An introduction to signal detection and estimation*, 2nd edition, Springer, 1994.
26. RSVP is described in IETF RFC's 2205–2210 starting with Braden, R., Ed. *RSVP Resource ReSerVation Protocol (RSVP)—Version 1, Functional Specification*, Sept, 1997.
27. Sutton, R.S., Barto, A.G., *Reinforcement Learning: ad Introduction*, MIT Press, Cambridge, Mass., 1998, 322p.
28. Tong, H., Brown, T.X., "Adaptive Call Admission Control under Quality of Service Constraints: a Reinforcement Learning Solution.", *IEEE JSAC*, v. 18, n. 2, pp. 209-221, Feb. 2000.
29. Tran-Gia, P., Gropp, O., "Performance of a neural net used as admission controller in ATM systems," *Proc. Globecom 92*, Orlando, pp. 1303–9.