

# On-line Measurement of QoS for Call Admission Control

Matthew Siler and Jean Walrand  
Department of Electrical Engineering and Computer Science  
University of California at Berkeley  
siler@eecs.berkeley.edu, wlr@eecs.berkeley.edu

February 27, 1998

## Abstract

On-line measurement of traffic can improve the quality of network monitoring and call admission control algorithms by providing reliable estimates of QoS parameters. We propose to estimate QoS by fitting an appropriate function to the buffer occupancy distribution of a single-server first-come-first-served queue. We collect samples of the buffer occupancy distribution just after packet arrivals in the queue. Then, we fit an approximating function, a positive sum of exponentials, using minimum chi-squared estimation techniques. The QoS parameters can then be inferred from the fitted distribution function. We consider three types of minimum chi-squared estimators: the first uses the sample covariance matrix, while the last two use biased approximations that are considerably easier to compute. Furthermore, we extend these algorithms to virtual queuing systems to estimate the impact of additional connections being admitted to the queue. We provide algorithms and criteria for all three estimators, along with simulation results. These simulations show that we can obtain reliable estimates of QoS parameters within minutes.

## 1 Introduction

Providing guaranteed Quality of Service (QoS) on the Internet requires that the network be able

to measure traffic and negotiate access to network resources. As QoS-demanding applications are more widely used, the network will eventually need to provide users with statistical guarantees of QoS, such as the loss rate, average packet delay, and delay jitter. In this paper, we present a collection of estimators that use on-line measurements of user traffic to provide call admission for a single-server first-come-first-served (FCFS) queue. Our approach infers the QoS parameters from measurements of the buffer occupancy at packet arrival times. A nonlinear optimization routine is used to fit these measurements to a buffer occupancy distribution, from which we can infer the loss rate and delay statistics. Furthermore, we extend these estimators to call admission by combining our fitting algorithm with virtual queuing systems to estimate the additional capacity within the queue. As a result, we are able to improve the performance of call admission algorithms by providing fast and reliable estimators of QoS.

To motivate our approach, consider the simple network in Figure 1. Suppose that we have a collection of users who are multiplexed onto a single bottleneck connection to the Internet. Such a scenario could easily describe local subscriber access for cable modems or ADSL technology. Traffic is divided into classes, with high-priority traffic scheduled before low-priority traffic in the multiplexer. Although users are free to send “best-effort” traffic to the low-priority

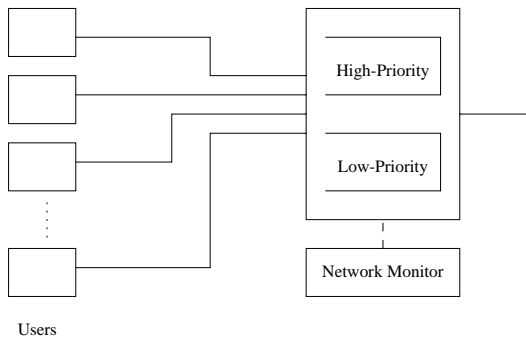


Figure 1: Multiple users sharing a high-priority and low-priority queue.

queue at any time, high-priority connections must be admitted beforehand. Assuming that high-priority traffic is rate-controlled and properly policed by the network, QoS is controlled through call admission. A call is blocked if the multiplexer estimates that sufficient resources are not available to handle the additional traffic. To provide statistical guarantees of service in the high-priority queue, a Network Monitor is used to measure the traffic and predict the amount of available resources.

Direct measurement and estimation of QoS parameters is difficult, usually because of the large variance of such estimators. For example, if we are interested in maintaining a loss rate of  $10^{-7}$ , then a suitable 95% confidence interval of the loss rate requires over 154 million packets, which may take days to collect.

We propose indirect estimators of QoS parameters which reduce the convergence time to minutes. These estimators are natural extensions to estimation techniques presented in [3]. Let  $W_\tau$  be the occupancy of the queue, in bytes, immediately after a typical packet arrival. Then, the distribution  $F$  of  $W_\tau$  determines the QoS provided by the queue. To estimate the QoS parameters, our approach consists of the following steps. First, we collect a histogram of  $W_\tau$ . Second, we fit a distribution  $G(\theta)$  to the collected histogram, so that  $G(\hat{\theta})$ , for some  $\hat{\theta} \in \Theta$ , is the “best” fit with respect to some discrepancy measure in  $\Theta$ . Third, we then use  $G(\hat{\theta})$  to infer the loss rate and delay statistics. Specifically, we

collect measurements during busy cycles. Under certain assumptions, the histogram bins are asymptotically multivariate normal. As a result, we consider three discrepancy measures which perform minimum chi-squared fitting. The *Maximum Likelihood Estimator (MLE)* uses an unbiased estimator of the covariance matrix in addition to the histogram. The other two estimators, the *Neyman IID Chi-Squared Estimator (NCE)* and *Pearson IID Chi-Squared Estimator (PCE)*, use the same fitting algorithms, except they compute a biased estimator of the covariance matrix that considerably reduces complexity. In all three cases, we compute suitable criteria, or estimators of the expected discrepancy, that may be used in evaluating the quality of the fitting algorithm. If used in conjunction with a virtual queue, then these estimators can also be used to predict the impact of adding additional traffic.

In Section 2, we present our queuing model and the relevant QoS parameters. Section 3 provides an introduction to model selection techniques, and presents suitable selection criteria for the three types of estimators we consider. Section 4 demonstrates how these algorithms may be incorporated into virtual queuing systems for predicting the effect of additional traffic on the queue. We briefly consider implementation issues in Section 5. To show the performance of our system, we provide simulation results in Section 6. Our concluding remarks, including future directions, are presented in Section 7.

## 2 Queue Model

We model the high-priority queue as a single-server FCFS queue with rate  $R$  bytes per second and a maximum buffer size of  $B$  bytes, as shown in Figure 2. Let  $\{A_t : t \geq 0\}$  be the packet arrival process to the queue, where  $A_t$  is the number of packet arrivals for the aggregate connection up to time  $t$ , for  $t \in [0, \infty)$ . If an arriving packet exceeds the available buffer capacity, then the entire packet is dropped by the queue. Define  $\tau_n$  to be the  $n$ -th packet arrival time, so that the jumps of  $A_t$  are precisely the times in the set

$\{\tau_n : n \geq 1\}$ . Denote the corresponding packet lengths by the random variables  $\{S_n : n \geq 1\}$ , where  $S_n \geq 1$  is the length, in bytes, of the  $n$ -th packet. Finally, let  $W_t$  be the buffer occupancy, in bytes, at time  $t \in [0, \infty)$ . Therefore,  $W_{\tau_n}$  is the number of bytes in the queue immediately after the  $n$ -th packet arrives.

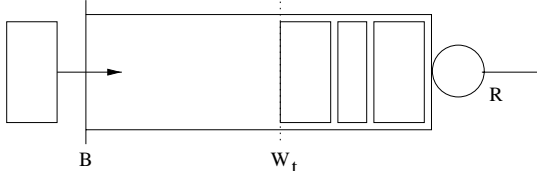


Figure 2: Single-server FCFS queue with rate  $R$  and buffer size  $B$ .

The perceived QoS by a typical arriving packet depends on the occupancy of the buffer at the time of arrival. If the arrival process is time-stationary and ergodic, and the queue is stable, then we can define  $W_\tau$  to be the occupancy of the buffer at a typical arrival time  $\tau$ . The QoS depends on the stationary distribution  $F$  of  $W_\tau$ . From now on, we will assume that the traffic is stationary. We consider extensions to these algorithms for time-varying traffic in Section 7.

Under this model, we may compute all of our QoS performance measures directly from  $F$ . The loss rate, i.e., the fraction of packets that will be dropped by the queue because of overflow, is defined as

$$L \equiv \Pr(W_\tau > B) = 1 - F(B).$$

Furthermore, the moments of the packet delay,  $D$ , may be computed directly from  $F$ . In particular, the average packet delay,  $E[D]$ , and the delay jitter,  $Var[D]$ , may be computed from

$$E[D] = \frac{1}{R} \int_0^\infty (1 - F(x)) dx$$

$$Var[D] = \frac{1}{R^2} \int_0^\infty 2x(1 - F(x)) dx - E[D]^2.$$

Of course,  $F$  is an unknown distribution function. In the next section, we will see how these

QoS measures may be estimated by recomputing the statistics above using a suitable distribution function to approximate  $F$ .

### 3 Estimating the Buffer Occupancy Distribution

To estimate  $F$ , we rely on *model selection* techniques [6]. First, we define a set of parameterized distribution functions that we will use to approximate  $F$ . Second, we present an overview of chi-squared estimation techniques. Third, we present a collection of three estimation algorithms, based on chi-squared fitting, that are used to fit an approximate distribution to  $F$ . Lastly, we suggest a suitable goodness-of-fit test based on the chi-squared distribution.

#### 3.1 Approximating Family of Distributions

To approximate the distribution  $F$ , we consider a collection of distribution functions on  $[0, \infty)$  of the form

$$G(\theta; x) = 1 - \sum_{j=1}^J \alpha_j e^{-\beta_j x}, \quad x \geq 0$$

where  $\theta \in \Theta$  is a vector of real, non-negative parameters given by

$$\theta = (\alpha_1, \dots, \alpha_{J-1}, \beta_1, \dots, \beta_J)^T$$

and  $\alpha_J \equiv 1 - \sum_{j=1}^{J-1} \alpha_j$ . We restrict  $\alpha_j \in [0, 1]$  and  $\beta_j \in [0, \beta_{max}]$ , for some  $\beta_{max} > 0$ , so that  $\Theta$  is a compact subset of  $\mathfrak{R}^{2J-1}$ . Moreover,  $G(\theta; x)$  is continuous and infinitely differentiable in  $\theta$ . Let  $\mathcal{G}_J = \{G(\theta; x) : \theta \in \Theta\}$  be the set of all approximating distribution functions of  $F$ , for a given  $J$ .

The set  $\mathcal{G}_J$  is an appropriate collection of distribution functions to consider. In fact, if we model the traffic process into the queue, given by  $(A_t, S_n)$ , as a  $p$ -state Markov-modulated fluid source, then the distribution of  $W_\tau$  is precisely a

distribution function from  $\mathcal{G}_{p-1}$  [11]. Furthermore, large deviation results state that, for a large number of users, the decay rate of the tail distribution of  $W_\tau$  is approximately exponential [2]. This suggests that a single exponential ( $J = 1$ ) will be sufficient for estimating the loss rate of a large capacity FCFS queue. However, the head of the queue may be quite dependent on additional terms, so multiple exponentials may be useful in estimating the delay statistics.

### 3.2 Chi-Squared Estimation

Given a set of  $n$  observations  $\{W_{\tau_i} : 1 \leq i \leq n\}$ , we want to select the distribution function from  $\mathcal{G}_J$  that best approximates the empirical distribution  $\hat{F}$  of these observations. That is, we find  $G \in \mathcal{G}_J$  that minimizes  $\Delta(G, \hat{F})$ , where  $\Delta(\cdot, \cdot)$  quantifies the disparity between two distribution functions.

Instead of fitting to  $\hat{F}$ , we approximate  $\hat{F}$  with a histogram  $H$ , with  $K$  bins, of the observations. Thus,  $H$  is a quantized approximation of  $\hat{F}$  that reduces the complexity of our estimators. In fact, the fitting of  $H$  involves only  $K$  equations, rather than the number of observations  $n$ . We lose some information when computing  $H$  from  $\hat{F}$ , and an important question is how to choose  $K$ , and the locations of the  $K$  bins, that define  $H$ .

The histogram is constructed as follows. Let  $K \geq 2J$  be the number of bin thresholds we choose to fit. The bin thresholds are chosen beforehand, with the constraint that  $b_0 \equiv 0$ ,  $b_K \equiv B$ , and  $b_0 < b_1 < \dots < b_{K-1} < b_K$ . Note that the last bin counts the number of actual losses from the queue. Typically, we will refer to these bin thresholds in vector form:

$$b = (b_1, b_2, \dots, b_K)^T.$$

Furthermore, let  $\{X_n; n \geq 1\}$  be a sequence of  $\{0, 1\}^K$ -valued random vectors, such that

$$X_n = (X_{n,1}, X_{n,2}, \dots, X_{n,K})^T$$

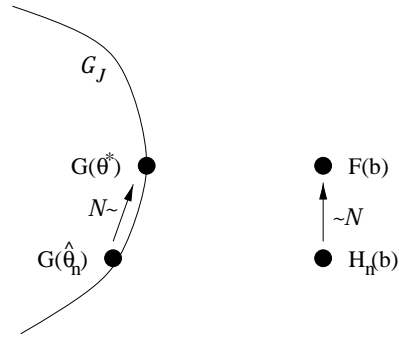


Figure 3: Convergence of distribution functions.

and

$$X_{n,k} = \begin{cases} 1 & W_{\tau_n} \leq b_k \\ 0 & \text{otherwise} \end{cases}.$$

Then,  $X_{n,k}$  is the indicator of the event that the number of bytes in the queue, after the  $n$ -th packet arrival, is below the threshold  $b_k$ . The histogram,  $H_n(b)$ , is simply the fraction of the first  $n$  packets that arrive before each threshold. Therefore, we may define the  $k$ -th element of  $H_n(b)$  as

$$H_n(b_k) = \frac{1}{n} \sum_{i=1}^n X_{i,k}.$$

This is also  $\hat{F}$  evaluated at the threshold  $b_k$ . The random vectors  $\{X_n : n \geq 1\}$  are not independent. However, the number of packets that arrive during distinct busy cycles are i.i.d. for each particular bin. Using this property, one can show that  $H_n(b) \xrightarrow{a.s.} F(b)$ , and that

$$\sqrt{n} (H_n(b) - F(b)) \xrightarrow{d} N(0, C)$$

for some covariance matrix  $C \geq 0$ . Therefore, the histogram  $H_n(b)$  is asymptotically multivariate normal, with mean vector  $F(b)$  and covariance matrix  $\frac{1}{n}C \rightarrow 0$ . Suppose that the closest  $G(\theta)$  to  $F$  is  $G(\theta^*)$ , and that the closest  $G(\theta)$  to  $H_n$  is  $G(\hat{\theta}_n)$ . Since  $H_n(b) - F(b)$  is approximately normal, we can expect  $G(\hat{\theta}_n; b) - G(\theta^*; b)$

and  $\hat{\theta}_n - \theta^*$  to be approximately normal as well (see Figure 3). Since the error is normally distributed, chi-squared estimation should be used to define  $\theta^*$  and to estimate  $\hat{\theta}_n$ . Specifically, we use the discrepancy measure  $\Delta(G(\theta), F) = \Delta(\theta)$ , where

$$\Delta(\theta) = (G(\theta; b) - F(b))^T \tilde{C}^{-1} (G(\theta; b) - F(b)).$$

Since  $C$  is positive semi-definite, the inverse may not exist, so  $\tilde{C}^{-1}$  is an appropriate pseudo-inverse of  $C$ . Furthermore,  $F$  and  $C$  are unknown, so  $\Delta(\theta)$  must be estimated with the empirical discrepancy measure

$$\Delta_n(\theta) = (G(\theta; b) - H_n(b))^T \tilde{C}_n^{-1} (G(\theta; b) - H_n(b)).$$

Therefore, the *minimum chi-squared estimator* of  $\theta^*$ , based on the first  $n$  observations, is the measurable function  $\hat{\theta}_n$ , where

$$\hat{\theta}_n = \arg \min_{\theta \in \Theta} \Delta_n(\theta). \quad (1)$$

Once we compute this estimate, the minimum discrepancy estimate of  $F$  is then  $G(\hat{\theta}_n)$ .

By the compactness of  $\Theta$ , and the continuity and almost sure uniform convergence of  $\Delta_n(\theta)$  on  $\Theta$ , the minimum discrepancy estimators  $\{\hat{\theta}_n : n \geq 1\}$  exist, and

$$\begin{aligned} \Delta_n(\hat{\theta}_n) &\xrightarrow{a.s.} \Delta(\theta^*) \\ \hat{\theta}_n &\xrightarrow{a.s.} \theta^* \end{aligned}$$

as  $n \rightarrow \infty$  [6] [5].

### 3.3 Estimation Algorithms and Criteria

To compute  $\Delta_n(\theta)$ , we need to compute  $C_n$ , and we present three methods for doing this in this section. The MLE approach computes  $C_n$  from the sample covariance matrix. This provides an unbiased estimate of  $C$ , but requires more computation time and complexity within the measurement device. To avoid this additional complexity, we present two other estimators which

only require the histogram. However, they introduce a bias into the fit. In the next three sections, we present each of these estimators, and derive a suitable criterion for each.

#### 3.3.1 Maximum Likelihood Estimation

The MLE approach uses an unbiased estimate of the covariance matrix to estimate  $C_n$  in (1). Therefore, in addition to the histogram, we also require the computation of  $C_{MLE,n}$ , the sample covariance matrix of  $H_n(b)$ . Since busy cycles are i.i.d., the sample covariance matrix may be computed from  $\{X_i : 1 \leq i \leq n\}$ , provided that the boundaries between busy cycles are known. Suppose that the  $n$ -th packet is the last one to arrive in the  $m$ -th busy cycle of the queue. Then, we can compute

$$C_{MLE,n} = \left(\frac{n}{m}\right)^{-1} C_{Y,m}.$$

$C_{Y,m}$  is the (unbiased) sample covariance matrix of the random vectors  $Y_1, Y_2, \dots, Y_m$ , where

$$Y_i = (Y_{i,1}, Y_{i,2}, \dots, Y_{i,K})^T$$

and  $Y_{i,k}$  is the number of packets in the  $i$ -th busy cycle such that  $W_{\tau_n} \leq b_k$ .

To develop a suitable criterion for the MLE approach, we need to estimate  $E_F[\Delta(\hat{\theta}_n)]$ , the expected discrepancy under the distribution  $F$ . If we assume that  $G(\theta^*) \approx F$ , then we may approximate this with

$$E_F[\Delta(\hat{\theta}_n)] \approx E_F[\Delta_n(\hat{\theta}_n)] + \frac{2J(J-1)}{n}.$$

That is, the expected discrepancy between  $G(\hat{\theta}_n)$  and  $F$  is approximately the expectation of the empirical discrepancy plus an error term due to the variance of the empirical discrepancy. This error term only depends upon the number of independent parameters and  $n$ . If one cannot assume that  $G(\theta^*) \approx F$ , then the set  $\mathcal{G}_J$  may be

biasing the estimates in some other way. Heuristically, this shows the penalty for increasing the number of parameters, and suggests that, for a small number of samples, a distribution from  $\mathcal{G}_1$  will provide the best fit for most distributions.

In this case, a suitable criterion is therefore

$$n\Delta_n(\hat{\theta}_n) + 2(2J - 1). \quad (2)$$

where  $\Delta_n(\hat{\theta}_n)$  is the discrepancy at time  $n$ . This value is used to compare the quality of the fit with respect to similar estimators.

### 3.3.2 Neyman IID Chi-Squared Estimation

Unlike the MLE approach, the NCE uses a biased estimate of  $C$  in its discrepancy measure. If we make the assumption that the random vectors  $\{X_n : n \geq 1\}$  are i.i.d., then we may approximate  $C_n$  with the covariance matrix

$$C_{NCE,n} = \left\{ H_n(b_{\min(k,l)}) - H_n(b_k)H_n(b_l) \right. \\ \left. : k, l = 1, \dots, K \right\}.$$

Since the true correlation between bins over time is positive, the asymptotic distribution of  $\Delta(\theta)$  is biased. However, our simulations show that the bias is not significant, and tends to estimate a loss rate that is higher than the true one.

In this case, a precise criterion is difficult to find since the bias due to correlation between samples is not known. Therefore, we use the same criterion as in (2), except with the appropriate replacement of  $C_{NCE,n}$  for  $C_n$ . For the traffic sources we simulated, the NCE approach performed well enough to justify the reduction in complexity. However, we expect that this difference may be more profound for burstier sources.

In many situations, where the resource and computation power of the measurement device is limited, this bias is a small price to pay for the reduced complexity of implementation. The histogram may be used to compute all the necessary statistics for the fit. Furthermore, the in-

verse can be computed analytically as a tridiagonal matrix, which reduces the computation time considerably.

### 3.3.3 Pearson IID Chi-Squared Estimation

The PCE is similar to the NCE in its approach, except that  $C_n$  is computed from the previously fitted distribution, rather than from the histogram directly. The covariance matrix in this case is

$$C_{PCE,n} = \left\{ G(\hat{\theta}_{n-1}; b_{\min(k,l)}) - G(\hat{\theta}_{n-1}; b_k)G(\hat{\theta}_{n-1}; b_l) \right. \\ \left. : k, l = 1, \dots, K \right\}. \quad (3)$$

Again, we introduce a bias that is similar to the NCE, and the criterion is the same. For measurement devices with limited resources, this estimator is as efficient as the NCE. However, since (3) involves the previously fitted distribution model, the estimates tend to be more “smoothed” than the other two approaches. That is, abrupt changes in traffic do not disrupt the estimates as significantly as in the other two methods.

## 3.4 Testing Goodness of Fit

With the MLE, NCE, and PLE approaches, we compute criteria based on chi-squared fitting. In fact, we can also use the criterion in each case as a chi-squared statistic to test the validity of our fit. With each estimator,

$$\Delta_n(\hat{\theta}_n) \approx \chi_r^2$$

where  $\chi_r^2$  is a chi-squared random variable with  $r = K - (2J - 1)$  degrees of freedom. For large enough  $n$ , we can compute  $\Delta_n(\hat{\theta}_n)$  and compare it to  $\chi_r^2$  using a lookup table.

For the MLE approach, this should provide a good indicator of the quality of the fit. However, since the NCE and PCE approaches involve some bias, the value for  $\Delta_n(\hat{\theta}_n)$  will be *stochastically smaller* than in the MLE case [7] [4]. That is, if we use a  $\chi_r^2$  test with the NCE or PCE approach, then we may be lead to believe that the

fit is better than it truly is. In any case, the  $\chi_r^2$  statistic that we compute is only an approximation, and so other verification techniques should be considered as well.

## 4 Call Admission

In the previous section, we considered algorithms for estimating the QoS parameters of the actual traffic from measurements of the buffer occupancy. For call admission, it is necessary to estimate the amount of additional resources available in the high-priority queue, so that new connections may be added without violating the guaranteed QoS. Since the arrival process is given by the aggregate connection  $(A_t, S_n)$ , it is not possible to compute in advance how  $F$  changes as new connections are added. However, it is possible to estimate the impact of adding a few more connections by using a virtual queuing system [3].

Suppose we are interested in estimating the impact of  $\epsilon$  additional traffic, i.e., the new distribution of  $W_\tau$  if we were to simultaneously replace each user with  $1 + \epsilon$  users. Then, if  $F = F(0, B, R)$  is our original FCFS queuing system with zero additional customers, then, for  $\epsilon$  small,

$$F(\epsilon, B, R) \approx F(0, B, R/(1 + \epsilon)).$$

To estimate  $F(\epsilon, B, R)$ , we can instead estimate  $F(0, B, R/(1 + \epsilon))$  with a virtual queuing system as shown in Figure 4. This is simply a queue that runs in parallel with the actual queue, receives the same traffic, but has a reduced transmission rate of  $R/(1 + \epsilon)$  to simulate the effect of fewer available resources. Any of the estimation algorithms from the previous section may be used to compute a suitable approximation for  $F(0, B, R/(1 + \epsilon))$ , and then we can estimate the QoS parameters for  $\epsilon$  additional traffic from the fitted distribution.

For reliable estimates of QoS, it is important that  $\epsilon$  be kept small. However, it is also equally important that  $\epsilon$  be large enough that it upper

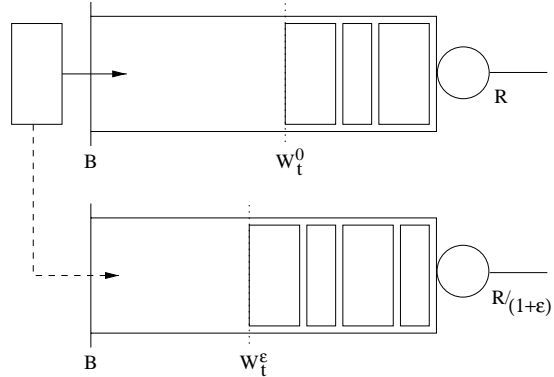


Figure 4: Virtual queuing system for  $\epsilon$  additional traffic.

bounds the fraction of additional traffic introduced by the new connection. If one is unsure of which  $\epsilon$  to use, then a suitable call admission region may be estimated using  $n > 0$  virtual queuing systems, with  $0 < \epsilon_1 < \epsilon_2 < \dots < \epsilon_n$ .

## 5 Implementation Issues

Implementation of these algorithms requires additional hardware and software within the Network Monitor.

### 5.1 Hardware Requirements

For  $K$  thresholds, precisely  $K + 1$  counters, one for each bin and one to count the total number of packets, are required to compute the histogram. We can compute the necessary statistics using these counters so that precisely one counter must be incremented after each packet arrival. Since this must be done for each packet, the time it takes to compute  $\max\{b_k : W_{\tau_n} \leq b_k\}$ , and then to increment the appropriate counter, must be much less than the packet interarrival time. For the MLE approach, an unbiased estimate of  $C$  must also be computed. This requires an additional  $K(K + 1)/2$  counters that must all be updated after each busy cycle in the queue.

## 5.2 Software Requirements

Assuming that we have collected a histogram  $H_n(b)$ , and in the MLE approach  $C_{MLE,n}$ , then we may compute  $\hat{\theta}_n$  using software within the Network Monitor. To compute  $\hat{\theta}_n$ , we require a weighted nonlinear least-squares optimization algorithm to minimize  $\Delta_n(\theta)$ . The weighting depends on the particular estimation technique, but generally involves the computation of a pseudo-inverse of  $C_n$ . We compute the pseudo-inverse for two important reasons. First,  $C_n$  is only positive semi-definite, so the inverse may not exist. Second, even if it does exist,  $C_n$  may be ill-conditioned in the sense that one or more singular values are close to zero. To solve both these problems, we compute the singular value decomposition (SVD) [8] of  $C_n$  as

$$C_n = USU^T,$$

where  $U$  is a matrix of orthonormal basis functions, and  $S$  is a diagonal matrix, with the singular values on the diagonal. Then, the pseudo-inverse  $\tilde{C}_n^{-1}$  is

$$\tilde{C}_n^{-1} = U\tilde{S}U^T,$$

where  $\tilde{S}_{ii} = 1/S_{ii}$  provided that  $S_{ii} \geq s_{tol}$ , and  $\tilde{S}_{ii} = 0$  otherwise. The value of  $s_{tol}$  is the minimum tolerated singular value. For the MLE approach, this requires that the SVD be computed before every fit. However, for the NCE and PCE approaches, we may exploit the structure of the covariance matrix and compute  $\tilde{C}_n^{-1}$  analytically, so only the singular values need to be checked.

Since  $G(\theta)$  is linear in  $\alpha = (\alpha_1, \dots, \alpha_{J-1})^T$  and nonlinear in  $\beta = (\beta_1, \dots, \beta_J)^T$ , we require an optimization algorithm that can minimize over both types of parameters. Specifically, we use separable regression, where a nonlinear optimization routine iterates between fitting  $\alpha$  and  $\beta$ . For  $J > 1$ , the linear fit is accomplished by keeping  $\beta$  fixed, and using a weighted linear least-squares algorithm to compute  $\alpha$ . The nonlinear

optimization routine is considerably more difficult. Traditional optimization routines typically fail for our particular choice of  $\mathcal{G}_J$  when  $J > 1$ . This is because the gradient matrix tends to be ill-conditioned when two choices of  $\beta_j$  are close together [9]. To avoid this, we use a Simplex Search algorithm developed by Nelder and Mead [8] which does not require the first derivative.

To find an appropriate  $J$  to use in the fit, we start with  $J = 1$ , compute the best fit, and then try  $J = 2$ , and so on. Since  $J = 1$  will always succeed in finding a best fit, a solution will always be available. Although increasing  $J$  indefinitely may lead to a tighter fit, it does not necessarily lead to a better estimate. With each increment of  $J$ , the number of parameters to fit increases by two. Using (2), it is possible to pre-determine whether larger values of  $J$  are likely to provide a better fit. In the end, the best fit according to (2) is used to compute the QoS parameters of interest.

## 6 Simulations

To test our approach, we have implemented all three estimators in software. We present simulations for the single exponential case for estimating the QoS parameters and an  $\epsilon = 5\%$  call acceptance region, which are presented below.

### 6.1 Estimating QoS Parameters

Let  $A_t$  have a Poisson distribution with rate  $\lambda$ , and  $S_n$  be exponential with rate  $\mu$ , for each  $n$ . Then,

$$P(W_\tau \leq x) \approx 1 - e^{-\mu(1-\rho)x}, \quad 0 \leq x \leq B$$

where  $\rho = \frac{\lambda}{\mu R}$ . This requires that the loss rate be  $L \approx 0$  since  $B < \infty$  means that some packets will eventually be dropped from the queue. Therefore, the QoS parameters in this case are

$$\begin{aligned} \log L &\approx -\mu(1-\rho)B \\ \log E[D] &\approx -\log(\mu R - \lambda). \end{aligned}$$

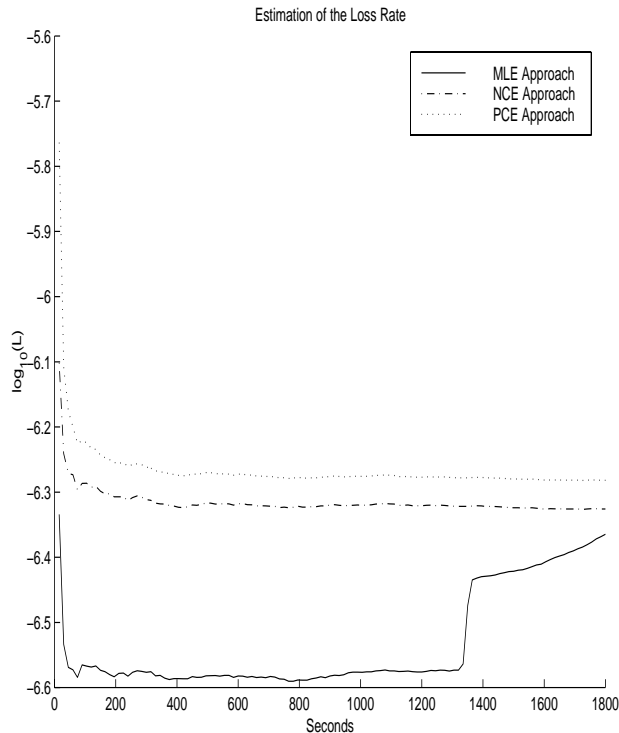


Figure 5: Estimated loss rate for the MLE, NCE, and PCE approaches.

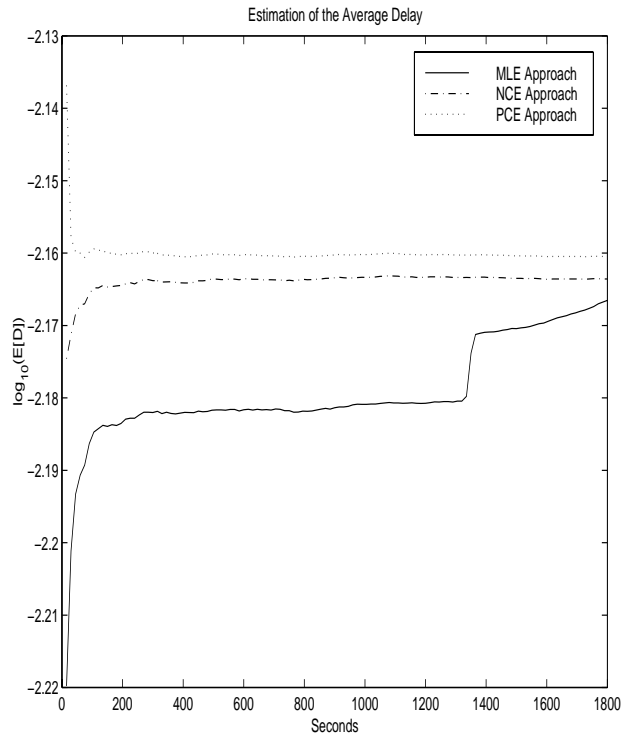


Figure 6: Estimated average delay for the MLE, NCE, and PCE approaches.

In the simulations below, we use  $\lambda = 840$  packets/sec,  $1/\mu = 100$  bytes/packet,  $B = 10,000$  bytes, and  $R = 100,000$  bytes/sec. We chose  $K = 8$  thresholds, where

$$b = (300, 600, 1000, 1500, 2250, 3000, 5000, 10000)^T$$

Figure 5 and Figure 6 show the estimated loss rate and average delay for the MLE, NCE, and PCE approaches, averaged over 200 simulation runs. Estimates were taken over 30 minutes, at 15 second intervals. In this case, the actual QoS parameters are

$$\begin{aligned} \log L &\approx -6.95 \\ \log E[D] &\approx -2.2 \end{aligned}$$

From the simulations, the estimates converge very quickly, within a matter of a few minutes, to values that are very close to the true ones. That we

can estimate the QoS parameters reliably within this amount of time is remarkable considering that no packets are ever dropped, and the buffer is almost never more than half full. Furthermore, at least in this example, the MLE performs only slightly better than the NCE and PCE approaches, and is certainly not worth the extra complexity to implement.

For the histogram thresholds, we chose  $b$  somewhat arbitrarily. As one would expect, the speed and accuracy of the estimates change with  $K$  and with the choice of  $b$ . This suggests that dynamic thresholding, i.e., adjusting the location of the thresholds based on the estimate, may improve these algorithms.

## 6.2 Call Admission

To demonstrate the performance of our algorithms for call admission, we also estimated the impact of  $\epsilon = 5\%$  additional traffic through the use of

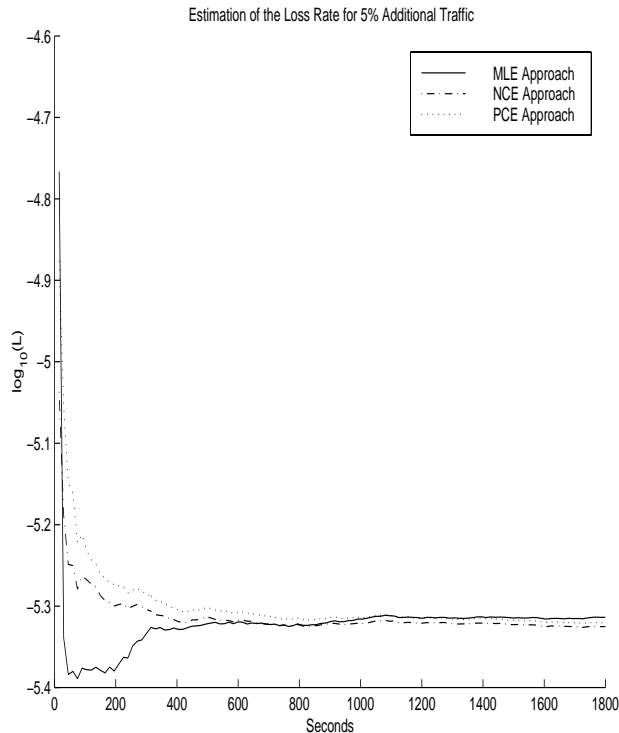


Figure 7: Estimated loss rate for 5% additional traffic using the MLE, NCE, and PCE approaches.

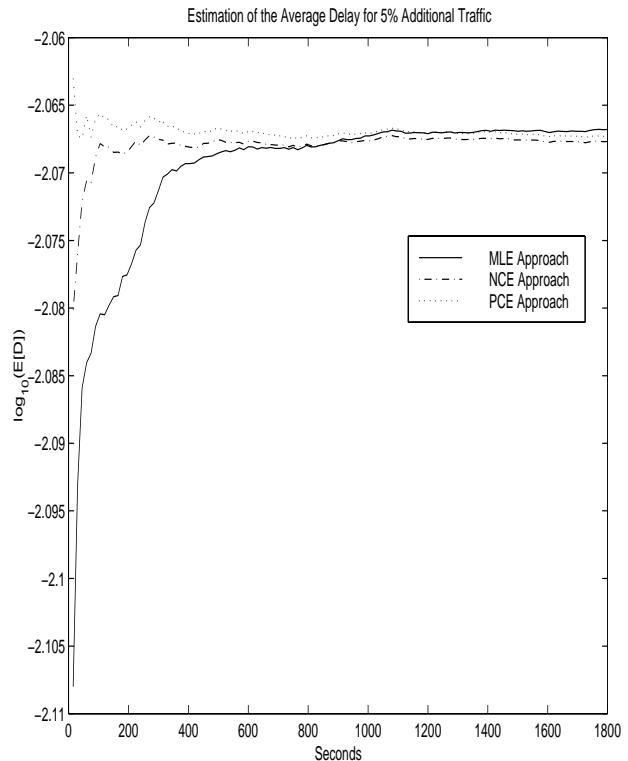


Figure 8: Estimated average delay for 5% additional traffic using the MLE, NCE, and PCE approaches.

a virtual queue. Figure 7 and Figure 8 show the estimated loss rate and average delay for the MLE, NCE, and PCE approaches. In our example, 5% additional traffic corresponds to the rate of  $A_t$  being increased to  $(1 + \epsilon)\lambda$ . Therefore, the actual loss rate and average delay would be

$$\begin{aligned} \log L &\approx -5.12 \\ \log E[D] &\approx -2.07 \end{aligned}$$

if the traffic were run through the actual queue.

In our simulation, the algorithms tend to underestimate the actual loss rate with  $10^{-5.3}$  instead of  $10^{-5.12}$ , but this result is certainly acceptable for a call admission algorithm. If not, better resolution could be gained by estimating the QoS parameters for several choices of  $\epsilon$ , so that the call admission algorithm can verify estimates from multiple sources. On the other hand,

the estimate of average delay is quite accurate, and would certainly be sufficient for call admission.

## 7 Remarks

We have presented a collection of algorithms which use on-line measurements to quickly and reliably estimate QoS parameters for a call admission algorithm. These algorithms overcome the problems of direct measurement through indirect estimation. Specifically, we show that fitting a histogram of the buffer occupancy to a sum of exponentials can decrease the estimation time of QoS parameters to within minutes.

Our analytic results rely on the stationarity of the traffic. However, actual network traffic is clearly not so. Therefore, we are currently investigating extensions to our algorithms for dealing

with time-varying traffic. Most likely, collective learning, where the Network Monitor associates traffic patterns with certain behavior, as well as filtering of estimates will be instrumental in our approach. Furthermore, our simulations suggest that we can improve the quality and convergence rate of our estimates by dynamically choosing thresholds.

## References

- [1] Patrick Billingsley. *Probability and Measure*. Wiley Series in Probability and Mathematical Statistics. John Wiley & Sons, Inc., 1995.
- [2] J. Bucklew. *Large Deviation Techniques in Decision, Simulation, and Estimation*. Wiley Series in Probability and Mathematical Statistics. John Wiley & Sons, 1990.
- [3] C. Courcoubetis, G. Kesidis, A. Ridder, J. Walrand, and R. Weber. Admission control and routing in ATM networks using inferences from measured buffer occupancy. *IEEE Transactions on Communications*, 43(4):1778–84, April 1995.
- [4] L. Gleser and D. Moore. The effect of dependence on chi-squared and empiric distribution tests of fit. *The Annals of Statistics*, 11(4):1100–1108, 1983.
- [5] R. Jennrich. Asymptotic properties of nonlinear least squares estimators. *The Annals of Mathematical Statistics*, 40(2):633–643, 1969.
- [6] H. Linhart and W. Zucchini. *Model Selection*. Wiley Series in Probability and Mathematical Statistics. John Wiley & Sons, 1986.
- [7] D. Moore. The effect of dependence on chi-squared tests of fit. *The Annals of Statistics*, 10(4):1163–1171, 1982.
- [8] W. Press, B. Flannery, S. Teukolsky, and W. Vetterling. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, 1988.
- [9] Axel Ruhe. Fitting empirical data by positive sums of exponentials. *SIAM Journal on Scientific and Statistical Computing*, 1(4):481–498, December 1980.
- [10] R. J. Serfling. *Approximation Theorems of Mathematical Statistics*. Wiley Series in Probability and Mathematical Statistics. John Wiley & Sons, 1980.
- [11] J. Walrand and P. Varaiya. *High-Performance Communication Networks*. Morgan Kaufmann Publishers, 1996.
- [12] H. Zhu and V. S. Frost. In-service monitoring for cell loss quality of service violations in ATM networks. *IEEE/ACM Transactions on Networking*, 4(2):240–8, April 1996.