A Measurement-Analytic Approach for QoS Estimation in a Network Based on the Dominant Time Scale

Do Young Eun and Ness B. Shroff, *Senior Member, IEEE*

***Abstract*—In this paper, we describe a measurement-analytic approach for estimating the overflow probability, an important measure of the quality of service (QoS), at a given multiplexing point in the network. A multiplexing point in the network could be a multiplexer or an output port of a switch or router where resources such as bandwidth and buffers are shared. Our approach impinges on using the notion of the *dominant time scale* (DTS), which corresponds to the most probable time scale over which overflow occurs. The DTS provides us with a measurement window for the statistics of the traffic, but is in fact itself defined in terms of the statistics of the traffic over all time. This, in essence, results in a *chicken-and-egg* type of unresolved problem. For the DTS to be useful for on-line measurements, we need to be able to break this chicken-and-egg cycle, and to estimate the DTS with only a bounded window of time over which the statistics of the traffic are to be measured. In this paper, we present a stopping criterion to successfully break this cycle and find a bound on the DTS. Thus, the result has significant implications for network measurements. Our approach is quite different from other works in the literature that require off-line measurements of the entire trace of the traffic. In our case, we need to measure only the statistics of the traffic up to a bound on the DTS. We also investigate the characteristics of this upper bound on the DTS, and provide numerical results to illustrate the utility of our measurement analytic approach.**

***Index Terms*—Dominant time scale (DTS), Gaussian processes, measurements, overflow probability, stopping criterion.**

I. Introduction

IN THIS paper, we develop a *measurement-analytic* approach to estimate the overflow probability at a *multiplexing point* in the network. As shown in Fig. 1, a multiplexing point in the network is where the resources are shared. (It could be a multiplexer or the output port of a switch.) The reason we study the overflow probability is that it is a fundamental measure of network congestion. Hence, accurately estimating it is important both from a quality of service (QoS) and a network optimization perspective.

Traditionally, researchers have pursued the problem of QoS estimation either from a purely analytical or a purely measurement-based approach. *We propose to use methods combining measurement and analysis.* As an illustrative example, assume that a network controller needs to estimate a QoS parameter such as the overflow probability. There are different approaches that one could take to estimate this parameter. One approach would be to directly measure the overflow instants, which would be fine if the instants occurred frequently enough. However, if the QoS metric corresponds to a rare event, e.g., an overflow probability of 10^{-7}, such a purely measurement-based approach may not work because it would take too long—any reasonable time window would have too few samples (if any) to make an accurate estimation. Another technique may be to observe the overflow of a "virtual queue" with a smaller capacity (so that we are sampling at a higher loss rate, perhaps by using a packet marking technique) and make a mapping from this higher overflow probability to the actual overflow probability. The difficulty here is to determine the precise mapping rule to be used.

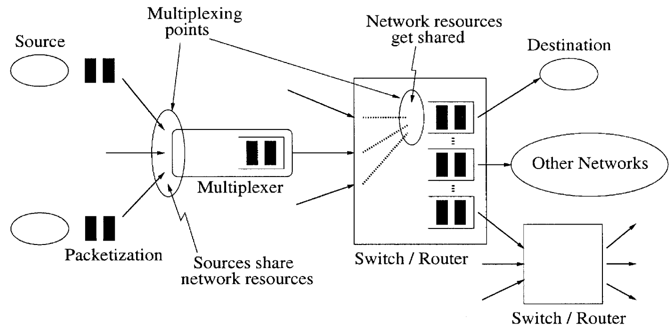


Fig. 1. Typical network.

Our objective is to estimate the statistics of rare events, namely, small values of the buffer overflow probability. If we had an accurate analytical approach to compute these probabilities from the statistics of the arrival process, then we could simply measure the statistics of the traffic arrival process, and based on these measurements compute the overflow probability. This would solve the problem by working around the difficulty of having to measure rare events. Of course, there is still the problem of which statistics of the arrival stream to measure and over what window of time. Further, we need to ensure that the model is relatively simple and parsimonious so that it does not require too many parameters. These issues will be further explored in this paper.

In the last several years, various long-range dependent (LRD) and self-similar models have been found to accurately characterize Internet traffic. These models have been proposed as being more suitable than traditional Markovian models, in that

Manuscript received May 23, 2001; revised October 16, 2001; approved by IEEE/ACM Transactions on Networking Editor S. Low. This work was supported in part by the National Science Foundation under Grant NCR-9624525 and Grant ANI-0099137, and by the Indiana 21st Century Fund under Grant 651-1285-0749.

The authors are with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907 USA (e-mail: shroff@ecn.purdue.edu).

Digital Object Identifier 10.1109/TNET.2003.810316

they are able to capture the burstiness over large time scales using a relatively small number of parameters [2], [3]. There have been in fact a number of arguments for and against the importance of LRD for traffic modeling [4]–[6]. However, recently, the notions of time and space scales have been able to fill in this divergence of opinions [6]–[9]. Depending on the network configuration, there exists a time scale, called the dominant time scale (DTS), over which the traffic should be characterized in order to estimate various QoS measures such as buffer overflow probabilities and delay distributions.

The concept of the DTS is quite useful for queueing analysis and is closely related to the concept embodied in the statement *rare events occur in the most probable way.* To see this more precisely, let $A(s,t)$ be the amount of traffic arrival in the time interval $[s,t)$ and C be the service capacity of the link. If we define $X_t := A(-t,0) - Ct$, then the steady-state overflow probability $\mathbb{P}\{Q > x\}$ is given by

$$\mathbb{P}\{Q > x\} = \mathbb{P}\left\{\sup_{t\geq 0} X_t > x\right\} \tag{1}$$

where x is the buffer level under consideration. Now the above statement says that

$$\mathbb{P}\left\{\sup_{t\geq 0} X_t > x\right\} \approx \sup_{t\geq 0} \mathbb{P}\{X_t > x\} \tag{2}$$

for sufficiently large x. This approximation is, in fact, a lower bound and shown to be quite accurate using large deviations [10], [11] and extreme value [12], [13] theories. Thus, it leads to the notion of the DTS, defined as the time index at which $\mathbb{P}\{X_t > x\}$ attains its maximum. From (2), it is easy to see that the DTS is the most probable duration of a busy period prior to overflow. In the large deviations setup [10], [11], the approximation (2) is obtained in a log-asymptotic way by picking the dominant exponent, or the so-called *principle of the largest term*. The idea is as follows. Assuming t is discrete, we start out by the following simple relation:

$$\begin{aligned}\sup_{t\geq 0} \mathbb{P}\{X_t > x\} \leq& \mathbb{P}\left\{\sup_{t\geq 0} X_t > x\right\}\\ =& \mathbb{P}\{\cup_t (X_t > x)\} \leq \sum_t \mathbb{P}\{X_t > x\}.\end{aligned}$$

Thus, if the probability $\mathbb{P}\{X_t > x\}$ decays fast enough in x, then the principle of the largest term applies, validating (2) on a log scale. Similarly, for a large class of Gaussian processes, the authors [12], [13] have established a result (e.g, see [13, Theorem 3]) involving the DTS, using extreme value theory. Intuitively, it says that if X_t ever exceeds level x, then it exceeds level x within a relatively small interval around the DTS. For further results and explanations on this, see [13] and [14] and references therein.

For fairly general processes, there has recently been a large body of work that employs the notion of time scale based on large deviation theory [9], [15], [16]. This effort has focused on the asymptotic behavior of the buffer overflow probability when the number of sources, the queue length, and the service rate are all proportionally sent to infinity. Let $L(C,B,n)$ be the buffer overflow probability for a buffer level B, service rate C, and source vector $n = (n_1, \ldots, n_J)$ (where n_j is the number of sources of type j). Then the result says that

$$\begin{aligned}\lim_{N\to\infty} \frac{1}{N} \log L(Nc, Nb, Nn) &= -I\\ &= -\inf_t \sup_s \left[s(ct+b) - st \sum_{j=1}^{J} n_j \alpha_j(s,t) \right]\end{aligned} \tag{3}$$

where $\alpha_j(s,t) := \log \mathbb{E}\{\exp(sA_j(0,t))\}/st$ is the effective bandwidth of a source of type j (see [17] for details). This equation is referred to as the *many-sources asymptotic* and has been proven for discrete time in [16] and for continuous time in [15]. From this equation, the overflow probability can be written as

$$\mathbb{P}\{Q > Nb\} = e^{-NI+o(N)} \approx e^{-NI}. \tag{4}$$

In order to use the many-sources asymptotic, we need to calculate the extremizing parameters in (3). Let s^* and t^* be the extremizing parameters over the $\sup$ and $\inf$, respectively. The parameter s^* can be calculated easily due to the convexity of $\alpha_j(s,t)$. Unfortunately, in the literature, there is no general property that we can take advantage of in order to find t^* (the parameter that is related to the DTS) [9], [18].

In this paper, we characterize the input process by a Gaussian process. As in the many-sources asymptotic, the Gaussian setting works especially well when a large number of sources are multiplexed. However, *a priori* modeling the aggregated input traffic as Gaussian allows us to obtain important properties of the DTS. Our goal in this paper will be to develop a measurement-analytic approach based on the DTS to estimate the buffer overflow probability. The DTS provides us with a measurement window for the statistics of the input process, but as will be discussed, is itself defined in terms of the statistics over all time t of the input process. Thus, for the DTS to be useful for on-line measurements, we need to be able to estimate it by measuring the statistics of the input process over only a bounded time window. This is the central problem of the paper and distinguishes our solution from other works in the literature.

The rest of this paper is organized as follows. In Section II, we outline the measurement-analytic approach and the *chicken-and-egg* type of problem that needs to be solved to make it viable. In Section III, we present a stopping criterion for finding the DTS and breaking this chicken-and-egg cycle. In Section IV, we provide an algorithm based on the stopping criterion that can be used for taking on-line measurements and making decisions such as admission control. Using real traffic traces, we show that the overhead is reasonable, as predicted by the asymptotic result in Section III, and that our approach performs quite well at any multiplexing point in the network. In Section V, we extend our approach to a multibuffered queueing system with general work-conserving service disciplines. In Section VI, we discuss the importance of our results and their implications for traffic modeling, as well as on-line measurement aspects of our approach. Finally, we conclude in Section VII.

II. Problem Description and Background

As mentioned in Section I, we will assume that the input process to a multiplexing point in the network (see Fig. 1) is Gaussian. The motivation behind Gaussian traffic characterization is that it is very natural when a large number of sources are multiplexed (motivated by the functional central limit theorem (CLT) [19]), as is expected to be the case in future networks. In fact, extensive numerical results obtained by the authors and by other researchers have demonstrated that the aggregation of even a fairly small number of traffic streams is usually sufficient for the Gaussian characterization of the input process. Further, Gaussian processes are closed under superposition. Therefore, unlike the case of Markovian queueing models (e.g., see [20] and [21] for difficulties with Markovian queueing models), analyzing a queue with a large number of Gaussian sources is no more difficult than analyzing a queue with a single Gaussian source. Also, Gaussian processes are completely specified by their first two moments. This makes Gaussian traffic characterization ideal from a measurement point of view, since measuring statistics beyond the second moment is usually impractical. Also, Gaussian processes can have an arbitrary correlation structure and this includes LRD (when the covariance function is not summable) processes. Hence, Gaussian processes cover all second-order LRD or second-order self-similar processes that have been shown to be good models for characterizing actual traffic [2].

Our multiplexing model constitutes a queueing system with fixed service rate C fed by a Gaussian process $\{A(0,t) : t \geq 0\}$ with stationary increments and mean rate $\lambda = \mathbb{E}\{A(0,t)/t\}$. Further, let $X_t := A(-t,0) - Ct$, $\kappa := -\mathbb{E}\{X_t\}/t = C - \lambda$, and $\sigma^2(t) := \mathrm{Var}\{X_t\}$. If we define $f(t,x,\kappa)$ as

$$f(t,x,\kappa) := \frac{\sigma^2(t)}{(\kappa t + x)^2} \tag{5}$$

then the DTS[1] $\hat{t}(x,\kappa)$ becomes

$$\hat{t}(x,\kappa) = \arg\max_{t\geq 0}\{f(t,x,\kappa)\}. \tag{6}$$

Note that $\mathbb{P}\{X_t > x\} = \Phi^c((\kappa t + x)/\sigma(t))$, where $\Phi^c(\cdot)$ is the complementary CDF of a standard Gaussian random variable. Since $\Phi^c(\cdot)$ is a decreasing function of its argument, $\hat{t}(x,\kappa)$ in (6) is the same time at which $\mathbb{P}\{X_t > x\}$ achieves its maximum value. From [13] and [14], for a large class of Gaussian processes, it has been shown that

$$\log \mathbb{P}\left\{\sup_{t\geq 0} X_t > x\right\} + \frac{m(x,\kappa)}{2} \in O(\log x) \tag{7}$$

where

$$m(x,\kappa) := \frac{(\kappa\hat{t}(x,\kappa) + x)^2}{\sigma^2(\hat{t}(x,\kappa))} = \frac{1}{\sup_{t\geq 0} f(t,x,\kappa)}. \tag{8}$$

This result is often called the maximum variance asymptotic (MVA) result. In [13], it was pointed out that the asymptotic result provided by relationship (7) has a significantly finer resolution than large deviation results. Further, and perhaps more importantly from a networking standpoint, although (7) is an asymptotic relation, the authors have shown through extensive simulations that the approximation

$$\mathbb{P}\{Q > x\} = \mathbb{P}\left\{\sup_{t\geq 0} X_t > x\right\} \approx \exp\left(-\frac{m(x,\kappa)}{2}\right) \tag{9}$$

which naturally follows from (7), provides an accurate estimate of the overflow probability over a wide range of buffer levels and utilizations. Note that the MVA approximation of the buffer overflow probability (9) is completely determined by $\hat{t}(x,\kappa)$ and $\sigma^2(\hat{t}(x,\kappa))$. Hence, once we are able to obtain $\hat{t}(x,\kappa)$ and $\sigma^2(\hat{t}(x,\kappa))$ from measurements, we can immediately calculate $\mathbb{P}\{Q > x\}$ from (9). Here, it is interesting to see that, in the case of Gaussian sources, the parameter t^* in (3) is equivalent to the DTS $\hat{t}(x,\kappa)$, and that (4) gives the same expression as the MVA result (9) when the input process is Gaussian [12], [13], [22].

In the literature, there exists some work to calculate the extremizing point t^* and numerical methods have been proposed. In [18], the entire source traffic was first recorded and then the whole time interval was divided into a sequence of time intervals with fixed step Δ. Then, for each $t = n\Delta$, $n = 1, 2, \ldots$, the authors computed the effective bandwidth at this point and finally performed a full search among those $n\Delta, n = 1, 2, \ldots$ to find t^*. In [23], the authors proposed an iterative method to find (s^*, t^*) using appropriate traffic substitution at each step. (We refer the reader to the paper for details.) However, the rate of convergence (and convergence itself) of this method has not been analytically proven. What is important is that both these techniques require the entire recorded traffic trace, while the method described in this paper will require only a partial amount of measured information.

As mentioned before, the MVA approximation given by (9) is supported theoretically by a variety of asymptotic results and empirically by extensive numerical studies [12]–[14], [24]. Hence, in this scenario we have a good analytical tool to estimate the overflow probability. Further, the MVA result depends only on $\kappa = C - \lambda$, $\hat{t}(x,\kappa)$, and $\sigma^2(\hat{t}(x,\kappa))$. Now, we need to determine the mean rate λ and $\hat{t}(x,\kappa)$ as well as $\sigma^2(\hat{t}(x,\kappa))$, the variance up to the DTS, from measurements to estimate the overflow probability $\mathbb{P}\{Q > x\}$ (or, equivalently, $\mathbb{P}\left\{\sup_{t\geq 0} X_t > x\right\}$). Estimating λ is fairly straightforward since the rate of flow can be measured from traffic traces by counting the number of aggregate packets arrived in time intervals. However, estimating $\hat{t}(x,\kappa)$ and $\sigma^2(\hat{t}(x,\kappa))$ requires that we know *a priori* what the DTS is. But, from the definition of the DTS in (6), $\hat{t}(x,\kappa) = \arg\max_{t\geq 0} \sigma^2(t)/(\kappa t + x)^2$, i.e., the DTS itself requires taking the maximum over all t of the normalized variance. This means that we need knowledge of $\sigma^2(t)$ over all t. Hence, we are faced with what seems to be an unresolvable *chicken-and-egg problem*, i.e., which comes first: the measurement window $\hat{t}(x,\kappa)$ or the variance $\sigma^2(t)$? It turns out that we can indeed break this cycle, as will be shown in Section III, but first we state an important property of the DTS that is also appealing from a measurement viewpoint.

Proposition 1: The DTS $\hat{t}(x,\kappa)$ is an increasing function of x/κ.

Proof: (Reproduced for Completeness From [25]): Since $\hat{t}_x$ is a function of x/κ [see (6)], without loss of generality, we can set $\kappa = 1$. Assume that there exist x and x' such that $x <$

[1] This is often called relevant time scale or critical time scale in similar contexts [6]–[8].

x' and $\hat{t}(x,1) > \hat{t}(x',1)$. From the definition of $\hat{t}(x,1)$ and $\hat{t}(x',1)$, we should then have

$$\frac{\sigma^2(\hat{t}(x',1))}{(\hat{t}(x',1)+x)^2} \leq \frac{\sigma^2(\hat{t}(x,1))}{(\hat{t}(x,1)+x)^2}$$
$$\frac{\sigma^2(\hat{t}(x,1))}{(\hat{t}(x,1)+x')^2} \leq \frac{\sigma^2(\hat{t}(x',1))}{(\hat{t}(x',1)+x')^2}.$$

However, by multiplying the above inequalities side by side, we get

$$\frac{(x'+\hat{t}(x',1))^2}{(x+\hat{t}(x',1))^2} \leq \frac{(x'+\hat{t}(x,1))^2}{(x+\hat{t}(x,1))^2}$$

which cannot be true because $(x'+t)^2/(x+t)^2$ is a monotonically decreasing function of t. Therefore, if $x < x'$, then we should have $\hat{t}(x,1) \leq \hat{t}(x',1)$. ∎

What is important in Proposition 1 is that for any given κ (or, equivalently, utilization $\rho = \lambda/C$), the DTS is increasing in x. Similarly, for any given x, the DTS is increasing in utilization (or decreasing in κ). These two results tell us that we only need to estimate the DTS for the largest utilization and buffer level of interest to us, and then we can immediately have an upper bound on the DTS for all lower buffer values and utilizations. Moreover, the DTS depends on x and κ only by their ratio. This is very important because it reduces the degree of freedom from the DTS being a function of two variables, to being a function of one variable. From a practical point of view, this means not having to take different measurements for each value of buffer size and utilization of interest. We now focus on the aforementioned chicken-and-egg problem in estimating the DTS.

III. Stopping Criterion for Finding the DTS

In this section, we present a method of finding the DTS using only a finite number of observations $t = t_1, t_2, t_3, \ldots$ for $\sigma^2(t)$, thus, breaking the cycle mentioned earlier. Since the DTS itself is defined as a global maximum of a function $f(t,x,\kappa)$, it appears that one may have to estimate $\sigma^2(t)$ for all $t > 0$ in order to calculate the DTS. To overcome this, we would like to find a function such that measuring this function over a *finite* amount of time would enable us to find the global maximum. We show next that there indeed exists such a function and, hence, a stopping criterion for finding the DTS $\hat{t}(x,\kappa)$, telling us that once the stopping criterion is satisfied, we do not have to estimate $\sigma^2(t)$ for larger values of t, and the cycle is broken. To show this result, we first need the following Propositions.

Proposition 2: Let X_t be a stochastic process with stationary increments. Then, for any convex function h, the function $S(t) := t\mathbb{E}\{h(X_t/t)\}$ is subadditive, i.e., $S(s+t) \leq S(s) + S(t)$ for all $s, t \geq 0$.

Proof: Observe that

$$\begin{aligned} h\left(\frac{X_{t+s}}{t+s}\right) &= h\left(\frac{X_t + (X_{t+s} - X_t)}{t+s}\right) \\ &= h\left(\frac{t}{t+s}\left(\frac{X_t}{t}\right) + \frac{s}{t+s}\left(\frac{X_{t+s}-X_t}{s}\right)\right) \\ &\leq \frac{t}{t+s} h\left(\frac{X_t}{t}\right) + \frac{s}{t+s} h\left(\frac{X_{t+s}-X_t}{s}\right). \end{aligned}$$

Thus, by taking expectations, we get the result from stationary increments property. ∎

Proposition 3: Let $\sigma^2(t) = \mathrm{Var}\{X_t\}$, where X_t is any stochastic process with stationary increments. Then, $r(t) := \sigma^2(t)/t$ is subadditive and $R(t) := \sigma^2(t)/t^2$ satisfies the following relation:

$$g(t) := \max_{u \in [t/2, t]} R(u) \geq R(s) \qquad \text{for all } s \geq t. \qquad (10)$$

Proof: See Appendix A. ∎

Note that (10) implies that the function $R(t)$ is likely to decrease. In fact, the function $g(t)$ is nonincreasing, which follows directly from the definition of $g(t)$ and the subadditivity of $r(t)$. Note also that the log–log plot of $R(t)$ is the well-known variance-time plot, which has been used for estimating the Hurst parameter [2]. Now, we present the stopping criterion.

Theorem 4 (Stopping Criterion): Suppose that for a given $x/\kappa > 0$, there exist positive numbers t_s and p satisfying

$$g(t_s) = \max_{u \in [t_s/2, t_s]} R(u) \leq \left(\frac{p}{p+1}\right)^2 R\left(\frac{px}{\kappa}\right) \qquad (11)$$

then $\hat{t}(x,\kappa) \leq t_s$.

Proof: Observe that

$$\kappa^2 f(t,x,\kappa) = \frac{\sigma^2(t)}{\left(t+\frac{x}{\kappa}\right)^2} = R(t)\left(\frac{t}{t+\frac{x}{\kappa}}\right)^2.$$

Then, clearly, $\kappa^2 f(t,x,\kappa) < R(t)$ for all $t, x/\kappa > 0$. For such p and t_s satisfying (11), suppose first that $t_s < px/\kappa$. Then by (11), we have $g(t_s) < R(px/\kappa)$, and this contradicts (10). Thus, we get $t_s \geq px/\kappa$. Now, for all $t \geq t_s$, we have

$$\begin{aligned} \kappa^2 f(t,x,\kappa) &< R(t) \\ &\leq g(t_s) \\ &\leq \left(\frac{p}{p+1}\right)^2 R\left(\frac{px}{\kappa}\right) \\ &= \kappa^2 f\left(\frac{px}{\kappa}, x, \kappa\right) \end{aligned}$$

by Proposition 3. Hence, $\hat{t}(x,\kappa) \leq t_s$. ∎

Note that Theorem 4 holds for any X_t with stationary increments. In fact, since the Hurst parameter H is almost always assumed to be less than one for general stationary processes (the requirement for a stationary process is that $H \leq 1$ and that $H = 1$ corresponds to the pathological case that $X_t = X \cdot t$ for some random variable X), $R(t) \to 0$ as $t \to \infty$. Thus, $\lim_{t\to\infty} g(t) = 0$. Hence, t_s can always be chosen to be finite for any fixed $p, x/\kappa > 0$. Thus, in order to use Theorem 4 for a given buffer level x and utilization (or, equivalently, κ), we first estimate $R(t)$ for several values of $t = t_1, t_2, \ldots, t_n$. We then set positive numbers $p = p_1, p_2, \ldots, p_n$ such that $p_i x/\kappa = t_i$. For example, set $p = p_1$, and compare the right-hand side of (11) with $g(t_i)$ for different values of i. If (11) is satisfied for a certain pair, we immediately obtain an upper bound on $\hat{t}(x,\kappa)$, and the DTS can be found by searching $R(t)$ up to this bound. A specific stopping algorithm using the stopping criterion (Theorem 4) will be presented in Section IV-A.

Now, in order to investigate the tightness of the bound, we consider the following two special cases.

A. Fractional Brownian Motion Case

Suppose that $\sigma^2(t) = Vt^{2H}$ for $t > 0$, i.e., the input process $A(0,t)$ is a fractional Brownian motion process with Hurst parameter $H \in [0.5, 1)$. Then, from (11) and because we can freely choose the positive constant p, the smallest t_s can be found as

$$t_s = 2 \min_{p>0} \left[\left(1 + \frac{1}{p}\right)^{(1/1-H)} p \right] \frac{x}{\kappa}. \tag{12}$$

Direct calculations show that the right-hand side of (12) is minimized when $p = p^* = H/1 - H$ and that the resulting t_s is

$$\begin{aligned} t_s =& 2\left(\frac{1}{H}\right)^{(1/1-H)} \frac{H}{1-H} \frac{x}{\kappa} \\ =& 2\left(\frac{1}{H}\right)^{(1/1-H)} \hat{t}(x,\kappa) = b(H)\hat{t}(x,\kappa), \end{aligned}$$

where $b(H) := 2(1/H)^{(1/1-H)}$. Typical values of H lie in $[0.5, 1)$ and it is easy to see that the function $b(H)$ is monotonically decreasing from 8 to $2e$ in this interval. This means that for an fBm process, in the worst case scenario, the stopping criterion will give a measurement window that is eight times the DTS.

B. Large Buffer Case

In this section, we study the behavior of $t_s/\hat{t}(x,\kappa)$ as x increases to infinity. First, suppose that $\sigma^2(t) \sim Vt^{2H}$, i.e.,

$$\lim_{t\to\infty} \frac{\sigma^2(t)}{Vt^{2H}} = 1 \tag{13}$$

where $0.5 \le H < 1$. Then, we have the following.

Proposition 5: Let $t_s(x,\kappa)$ be the number satisfying (11) with equality,[2] that is

$$g(t_s(x,\kappa)) := \max_{p>0} \left(\frac{p}{p+1}\right)^2 R\left(\frac{px}{\kappa}\right). \tag{14}$$

Then, for any fixed $\kappa > 0$

$$\lim_{x\to\infty} \frac{t_s(x,\kappa)}{\hat{t}(x,\kappa)} = b(H) = 2\left(\frac{1}{H}\right)^{(1/1-H)}. \tag{15}$$

Proof: See Appendix B. ■

From (15), we know that there exists $M > 0$ such that $t_s(x,\kappa)/\hat{t}(x,\kappa) \le M$ for all $x > 0$. Further, as x increases, this function converges to a constant $b(H)$. Although this is an asymptotic result, we see in Section IV using real traces that $t_s(x,\kappa)/\hat{t}(x,\kappa)$ is no larger than 6 for most buffer levels x.

IV. Experimental Results

Using (11), we can find the DTS for different values of x and κ. Since the DTS is a function of x/κ, we will fix κ in the remainder of the paper and focus on the DTS as the buffer level x increases. Similarly, we could apply our result to the case of a fixed buffer level and different utilizations (or κ). In this section, we propose a simple algorithm to find $t_s(x,\kappa)$ and show that $t_s(x,\kappa)/\hat{t}(x,\kappa)$ is usually small, as suggested by Proposition 5, for several different input traffic scenarios. Since the point of this paper is not to develop new estimation schemes for calculating the mean or the variance at a given point (which can be done by using a variety of standard methods), we assume that our estimates of λ and $R(t)$ are accurate. However, later in Section VI-B, we will also look into the effect of estimation errors on the overflow probability. Let the smallest time scale available be normalized to one. For example, if we are measuring traffic that is originally an MPEG-encoded video sequence with 25 frames/s, then one time unit corresponds to 40 ms. Suppose that we measure $R(1), R(2), \ldots$ as time goes on and that we want to find $\hat{t}(x,\kappa)$ for $x = i\Delta$, $i = 0, 1, 2, \ldots$ where Δ is the buffer step size. Then, although Theorem 4 is stated for the continuous-time case, it is straightforward to see that we can still use it for the discrete-time case, as shown next.

A. Stopping Algorithm

We can use (11) in the following form.

Discrete Version of Theorem 4: If there exist positive integers i, j and even number n satisfying

$$g(n) := \max_{n/2 \le u \le n} R(u) \le \left(\frac{j}{\frac{i\Delta}{\kappa} + j}\right)^2 R(j) \tag{16}$$

then $\hat{t}(i\Delta,\kappa) \le n$.

In fact, (16) can be proven by noting that the proof of Proposition 3 is still valid in the integer domain. Now, suppose that we know $R(1), R(2), \ldots, R(n)$. Then, in order to apply our stopping criterion, we first compute $g(n)$ and compare every pair of these values to see if (16) holds. If (16) is satisfied for a certain pair, we immediately get the upper bound. Hence, we can find $\hat{t}(i\Delta,\kappa)$ without knowing all $R(n)$ for $n \ge 1$.

Let $t_s(i,\kappa)$ be the upper bound on $\hat{t}(i\Delta,\kappa)$ from (16) and let $a(i,j) := (j/(i\Delta/\kappa + j))^2$. Fig. 2 shows the flowchart describing the algorithm for finding $t_s(i,\kappa)$ for $i = 1, 2, \ldots, M$ while measuring $R(n)$ as n increases. This algorithm can be explained as follows. Let $X = \{x_{i,j}\}$ be a matrix where $x_{i,j} := a(i,j)R(j)$ for $i = 1, 2, \ldots, M$. Note that $x_{i,j}$ decreases as i increases and that $R(n)$ completely specifies the nth column of X. First, set $i = 1$. Whenever a new value of $g(n)$ is available, we compare it with $x_{i,j}$ for $j = 1, 2, \ldots, n$. If (16) is satisfied for some $\tilde{j}$, $1 \le \tilde{j} \le n$, then we get $t_s(i,\kappa) = n$. In this case, we fix the $\tilde{j}$th column and continue to compare $x_{l,\tilde{j}}$ with $R(n)$ for $l = i+1, i+2, \ldots$, until the condition (16) with i replaced by l is violated. This will give $t_s(l,\kappa) = n$. Once we have found $t_s(i,\kappa)$ for $i = 1, 2, \ldots, M$, the DTS $\hat{t}(i\Delta,\kappa)$ can be obtained by simply searching $x_{i,j}$ for $\hat{t}((i-1)\Delta,\kappa) \le j \le t_s(i,\kappa)$ using Proposition 1. This can be done by any sophisticated searching method, or we can integrate this with the above algorithm.

From now on, unless otherwise noted, we will use precomputed values of λ and $R(n)$, and then apply the algorithm described above in order to illustrate the power of our stopping criterion. *Note that without the stopping criterion, even if we knew $R(n)$ for all n, we would still need to search over an infinite horizon to find the DTS.* In Section VI-B, we will describe

[2] If $\lim_{t\to 0} \sigma^2(t) = 0$, it is easy to show that $R(t)$ is continuous for all $t > 0$ using the Cauchy–Schwarz inequality and the stationary increment property. Hence, $g(t)$ is also continuous and the equality in (11) can be achieved.

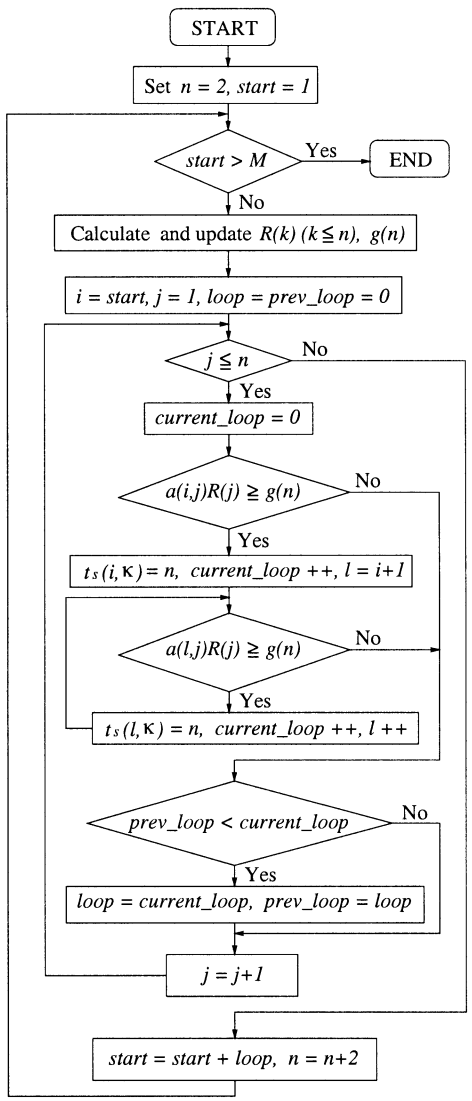


Fig. 2. Algorithm for finding $t_s(i, \kappa)$, the upper bound on the DTS.

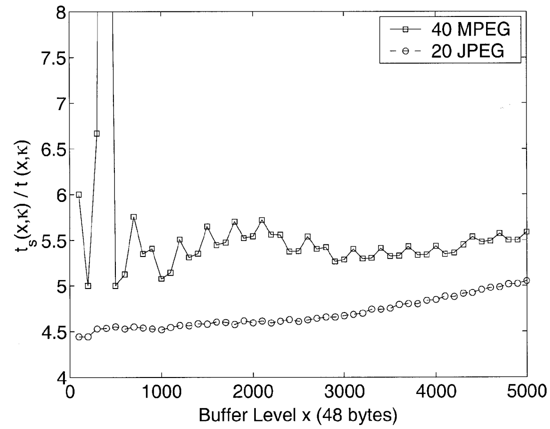


Fig. 3. $t_s(x, \kappa)/\hat{t}(x, \kappa)$ as a function of x for JPEG and MPEG video traffic.

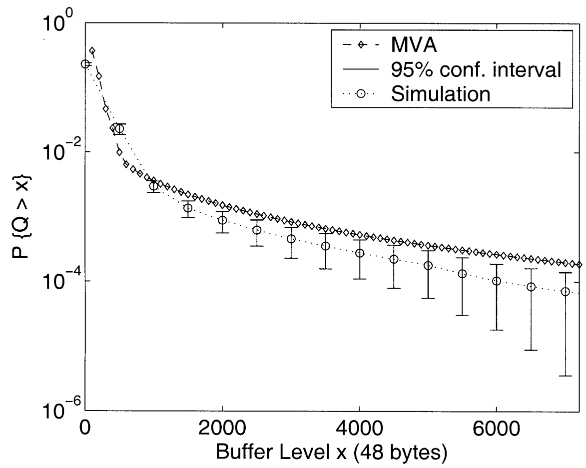


Fig. 4. Buffer overflow probability and its MVA approximation for 40 multiplexed MPEG video traffic. Utilization is set to $\rho = 0.85$.

how to calculate and update the traffic statistics on-line and investigate the effect of estimation error caused by the on-line estimation of the traffic statistics on the QoS estimates.

B. Single Queue

To experimentally validate our stopping criterion and numerically investigate the bound for a single queue, we use MPEG-1 and JPEG encoded *Star Wars* video traffic sequences with a 40-ms frame interval. The first set is 20 multiplexed JPEG sources and the other one is 40 multiplexed MPEG sources. We obtain these multiplexed traces by using a single trace, but with random offset (starting points). Even though the number of multiplexed sources is small, we will see that a Gaussian characterization will yield good results. For the JPEG case, we have observed (after calculating all the $\sigma^2(t)$!) that the function $f(t, x, \kappa)$ in (5) is a unimodal function for all $x > 0$. However, for the MPEG case, due to its periodic nature, the function $f(t, x, \kappa)$ generally has more than one local maximum for all the values of $x > 0$. Thus, without our stopping criterion, one would have to measure $\sigma^2(t)$ for all $t > 0$ to find the global maximum of $f(t, x, \kappa)$.

Fig. 3 shows $t_s(x, \kappa)/\hat{t}(x, \kappa)$ as a function of buffer size under a utilization of 0.9. For the JPEG case, the ratio is well behaved for all buffer sizes under consideration. In contrast, for the MPEG case, the ratio $t_s(x, \kappa)/\hat{t}(x, \kappa)$ is irregular because of the periodic nature of MPEG traffic. For a buffer size of $400 \cdot 48$ bytes, the ratio reaches 17.5 and it is omitted from the figure. However, the upper bound $t_s(x, \kappa)$ is still relatively small at this point (we observe that $\hat{t}(x, \kappa) = 4 \cdot 40$ ms and $t_s(x, \kappa) = 70 \cdot 40$ ms which corresponds to four and 70 frame intervals, respectively). Here, it should be pointed out that our stopping criterion is guaranteed to work with any stationary Gaussian input traffic and that, as predicted from Proposition 5, the ratio $t_s(x, \kappa)/\hat{t}(x, \kappa)$ is no larger than 6 for most of the buffer sizes used in our simulation. This property enables us to apply our stopping criterion to an on-line measurement framework.

Fig. 4 shows the buffer overflow probability for 40 multiplexed MPEG sequences with 95% confidence interval and the MVA approximation [given by (9)] using our stopping criterion for calculating the DTS. For this simulation, we first generate different realizations of the multiplexed streams by choosing different sets of random starting frames. We then get different estimates of the overflow probability for each buffer level. In Fig. 4, we plot the average values of these estimates and calculate the confidence interval using standard techniques, e.g., via the distribution of a standard normal random variable. Since the accuracy of the MVA approximation under a single queue has already been verified for general sets of traffic traces [12], [13], [24], we do not provide further results on this.

From the structure of the MVA approximation and our stopping criterion, we can use Fig. 4 to solve different network problems. Let $C = 620$ Mb/s and let the maximum buffer delay be $d = 15$ ms. Suppose that we want to determine the maximum number of sources n such that the probability of delay violation is no more than $P_D = 10^{-6}$. We assume that the sources are i.i.d and that the 40 multiplexed MPEG sequence represents one aggregate source. Then our problem is to find the largest n $(0 < n < C/\lambda)$ such that

$$\sup_{t\geq 0} \frac{n\sigma^2(t)}{((C-n\lambda)t + Cd))^2} \leq -\frac{1}{2\log P_D} := A \tag{17}$$

where λ and $\sigma^2(t)$ are the mean and the variance function of the source. This type of problem is relevant for admission control. Let n^* be the largest number of such sources that can be admitted without violating the overflow probability constraint. Then, since the left-hand side of (17) increases as n increases, n^* must satisfy (17) with equality, and the resulting equation has a unique solution n^* for any given A.[3] Rewriting (17) gives

$$\sup_{t\geq 0} \frac{\sigma^2(t)}{\left(\kappa t + \frac{\kappa C d}{C - n\lambda}\right)^2} = A\frac{(C-n\lambda)^2}{\kappa^2 n} \tag{18}$$

where $\kappa = (1/\rho - 1)\lambda$ and ρ is the utilization used in Fig. 4, i.e., $\rho = 0.85$. Now, let x and y be defined as

$$x = \frac{\kappa C d}{C - n\lambda} \quad \text{and} \quad y = A\frac{(C-n\lambda)^2}{\kappa^2 n}. \tag{19}$$

Substituting x in terms of n into y gives

$$y = \frac{A\lambda C d^2}{x(x-\kappa d)} \tag{20}$$

where $x > \kappa d$ since $0 < n < C/\lambda$. However, note that x and y above should also satisfy the following equation [from (18)]:

$$y = \sup_{t\geq 0} \frac{\sigma^2(t)}{(\kappa t + x)^2} \tag{21}$$

and that the MVA curve in Fig. 4 is nothing but $\exp(-1/2y)$ as a function of x. Thus, the solution x^* is the point at which (20) and (21) coincide, and the corresponding n^* can be obtained from (19).

Fig. 5 shows how we can find such an n^* using the MVA approximation and the stopping criterion. The solid line corresponds to (21), which can be calculated on-line by the stopping criterion. The remaining two curves correspond to (20) for different service capacities. Note that the meeting points are approximately $x^* = 2800 \cdot 48$ bytes and $x^* = 1350 \cdot 48$ bytes for $C = 620$ and $C = 155$ Mb/s, respectively. Hence, from (19), we obtain $n^* = 40$ for $C = 620$ Mb/s and $n^* = 9$ for $C = 155$ Mb/s. Note that the link utilization $n^*\lambda/C$ increases from 0.92 to 0.96 as the service capacity C increases from 155 to 622 Mb/s, as expected from statistical multiplexing.

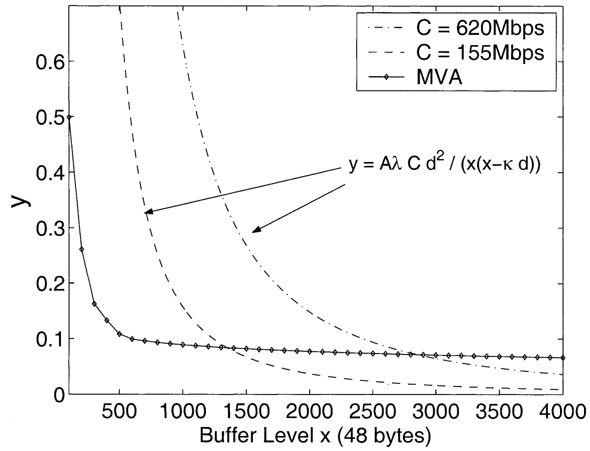


Fig. 5. Calculating the maximum number of sources to be accepted under $P_D \leq 10^{-6}$ and maximum buffer delay = 15 ms for two different service capacities C.

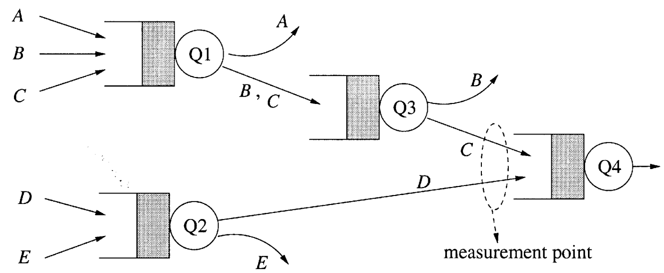


Fig. 6. Illustration of a network of queues. Each queue is being operated at 85% utilization and we are interested in estimating overflow probability at Q4.

TABLE I
DESCRIPTION OF INPUT TRAFFIC FOR FIG. 6

	input traffic	number of sources
A	JPEG (Star wars)	5
B	Ethernet traffic	100
C	MPEG (Star wars)	10
D	MPEG (Jurassic Park)	20
E	Auto Regressive model	n/a

C. Simulating a Network of Queues

In this section, we apply our measurement-analytic method to the network of queues as shown in Fig. 6. Details of the input traffic are summarized in Table I.[4] The traffic is served according to first-in-first-out (FIFO) scheduling discipline at each queue. In Fig. 6, the letter on the arrow represents the corresponding traffic taking that path. For example, source B arriving to Q1 will join Q3 after being served at Q1, and then depart the system after Q3. We set the utilization equal to 0.85 for every queue in this scenario and want to estimate the buffer overflow probability at Q4. In this scenario, we choose the number of each

[3]We assume that n is a real valued number in (17). The actual number then becomes $\lfloor n \rfloor$.

[4]Ethernet trace. [Online.] Available: http://ita.ee.lbl.gov/html/contrib/BC.html. MPEG (Jurassic Park) trace. [Online.] Available: ftp-info3.informatik.uni-wuerzburg.de/pub/MPEG/. See [26] for details.

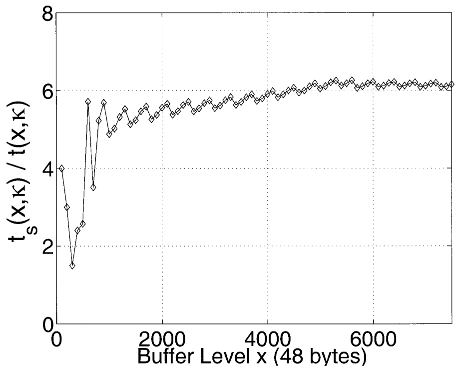


Fig. 7. $t_s(x,\kappa)/\hat{t}(x,\kappa)$ as a function of x for Q4 in the network of queues.

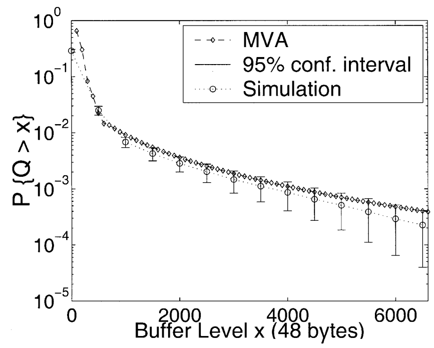


Fig. 8. Buffer overflow probability and its MVA approximation for Q4 in the network of queues.

type of source such that one of them does not dominate at any queue.

Figs. 7 and 8 show the overhead $t_s(x,\kappa)/\hat{t}(x,\kappa)$ and the buffer overflow probability, respectively, at Q4. We see that the overhead is no larger than a factor of six in this case and that the resulting measurement-analytic method accurately estimates the overflow probability. An interesting observation is that although Q4 sees only two independent groups of flows from the previous queues, this does not impact the accuracy of the measurement-analytic scheme. This is consistent with other numerical results that we have conducted and seems to indicate that the Gaussian assumption at interior nodes in the network (where a large number of independent streams may not be multiplexed, e.g., a 16×16 router allows only 16 independent flows to be multiplexed at each output port) does not create significant errors. An explanation for this could be that each of the composite flows at the interior of the network comprises many sources (in this example, sources C and D are composites of ten and 20 different MPEG sources). So, although one cannot assume that the composite flows at the interior of the network are comprised of strictly independent flows (due to the disturbance created by the intermediate queues, e.g., Q1 and Q3 for C and Q2 for D), the fact that they were comprised of independent flows at the exterior of the network appears to help in the Gaussian approximation at the interior of the network.

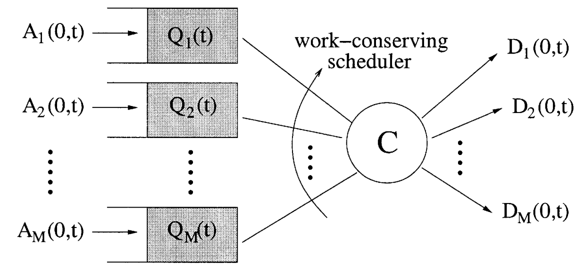


Fig. 9. Multibuffered system with general work-conserving service disciplines.

V. Extension to Multibuffered Queueing System With General Service Disciplines

In this section, we are interested in estimating the overflow probability ($\mathbb{P}\{Q_i > x_i\}$) at each queue i for a multibuffered queue, e.g., the queue shown in Fig. 9. Note that, from the total queue size $\left(Q(t)=\sum_{i=1}^{M} Q_i(t)\right)$ point of view, the system in Fig. 9 is equivalent to that of a single-queue system with service capacity C, regardless of the specific scheduling policy, as long as that policy is work conserving. Thus, if we are interested in the distribution of the total workload, we can use the same technique as in Section IV to estimate the overflow probability $\mathbb{P}\left\{\sum_{i=1}^{M} Q_i > \sum_{i=1}^{M} x_i\right\} = \mathbb{P}\{Q > x\}$. Our focus in this section will, however, be on estimating the overflow probabilities $\mathbb{P}\{Q_i > x_i\}$ associated with each queue i.

We assume here general work-conserving scheduling policies among the different buffers ($Q_i,\ i=1,2,\ldots,M$). Examples of such policies are static priority queueing and generalized processor sharing (GPS). In the literature, the performance analysis of these scheduling policies is difficult and the resultant solutions are not suitable for on-line QoS estimation. We will show that our original measurement-based solution can be employed without significant change and results in simple and accurate estimates of the overflow probability for each queue.

A. Methodology

The multiplexing point in the network can be modeled as a single or multibuffered queue. We only assume that this queue serves traffic according to a work-conserving service discipline (see Fig. 9). By *work conserving*, we mean that as long as one of the queues is not empty, the server keeps working. Let M be the number of different traffic classes being served by a work-conserving scheduler. Traffic from each class is fed into a queue, as shown in Fig. 9. The total service capacity remains fixed while the actual service rate to each queue is time varying, depending on a specific scheduling policy, for example, priority queueing, weighted-fair queueing, earliest deadline first (EDF), etc. Because each queue sees a time-varying service capacity, the analysis of these queueing models is much more complicated compared with a single-queueing model with a constant service rate (see, e.g., [27]–[30]).

Let $A_i(s,t)$ be the amount of class i traffic arriving during $[s,t)$ and $D_i(s,t)$ be the amount of class i traffic that departs during the time period $[s,t)$. We assume that the arrival process of each class of traffic is independent of each other. The maximum service rate of the server is C, and we define

$B_i(s,t)$ as the amount of service that has been allocated to class (or queue) i during $[s,t)$. For example, for a single queue with service capacity C, the function $B_i(s,t)$ simply becomes $B_i(s,t) = C(t-s)$. From the definition of $B_i(s,t)$, we have $D_i(s,t) = B_i(s,t)$ if queue i is backlogged throughout the interval $[s,t)$. Now assume that queue i is empty at time 0, then $Q_i(t)$, the i'th queue size at time t, can be represented as

$$Q_i(t) = \sup_{0 \le s \le t} [A_i(s,t) - B_i(s,t)]. \tag{22}$$

In (22), note that given the arrival $A_i(s,t)$, the function $B_i(s,t)$ that produces the same queue length $Q_i(t)$ may not be unique. To see this, suppose that queue i is empty during $[s',t')$. Then, the queue size will not be changed by increasing the service capacity during the interval $[s',t')$. Motivated by this, we define $B_i(s,t) \Leftrightarrow \tilde{B}_i(s,t)$, if these two functions generate the same $Q_i(t)$ for the same arrival. Thus, from the point of view of the queue size, these two functions are indistinguishable from each other.

We need $A_i(s,t)$ and $B_i(s,t)$ in (22) to estimate $Q_i(t)$. While it is possible to measure $A_i(s,t)$ and $D_i(s,t)$ directly from the traffic stream, it appears that $B_i(s,t)$ may be more difficult to obtain. However, it turns out that there exists the following simple relationship between $B_i(s,t)$ and the departure $D_i(s,t)$ which we can exploit.

Proposition 6: For any queue $i \in 1, 2, \ldots, M$ and under any work-conserving queueing policy, we have

$$B_i(s,t) \Leftrightarrow C(t-s) - \sum_{j \neq i} D_j(s,t) \qquad \text{for all } s < t.$$

Proof: Without loss of generality, it is enough to consider the case that queue i is either backlogged or empty throughout the interval $[s,t)$. If this is not the case, we simply divide the interval $[s,t)$ into smaller intervals such that each of these new intervals is either a busy period or an idle period.

Suppose first that queue i is backlogged throughout the interval $[s,t)$. Then, from the work-conserving property and the definition of $B_i(s,t)$, we have $C(t-s) = \sum_j D_j(s,t)$ and $B_i(s,t) = D_i(s,t)$. Thus, we get $B_i(s,t) = C(t-s) - \sum_{j \neq i} D_j(s,t)$. On the other hand, suppose that the queue is empty during the interval $[s,t)$. Since the service capacity allocated to the queue i cannot be larger than the total service capacity minus what has been used up for the other queues, we have $B_i(s,t) \le C(t-s) - \sum_{j \neq i} D_j(s,t)$. However, this means that $B_i(s,t) \Leftrightarrow C(t-s) - \sum_{j \neq i} D_j(s,t)$ since the queue i is empty during $[s,t)$. ■

From the above result, we can write the buffer overflow probability at time zero as

$$\begin{aligned}\mathbb{P}\{Q_i(0) > x\} &= \mathbb{P}\left\{\sup_{t \ge 0} [A_i(-t,0) - B_i(-t,0)] > x\right\} \\ &= \mathbb{P}\left\{\sup_{t \ge 0} \left[A_i(-t,0) + \sum_{j \neq i} D_j(-t,0) - Ct\right] > x\right\} \\ &= \mathbb{P}\left\{\sup_{t \ge 0} X_i(t) > x\right\}\end{aligned} \tag{23}$$

where

$$X_i(t) := A_i(-t,0) + \sum_{j \neq i} D_j(-t,0) - Ct. \tag{24}$$

Thus, the overflow probability in a general work-conserving queue turns out to be the supremum distribution of $X_i(t)$, which is a function of the traffic (aggregated) arrival and departure, as in (24).

Following our approach earlier, we assume that $X_i(t)$ is Gaussian with stationary increments. For a single queue, we get $X_1(t) = A_1(-t,0) - Ct$ and, hence, this assumption is justified by the CLT type of arguments. However, for a multi-buffered queueing system as shown in Fig. 9, it may not be obvious that this assumption is reasonable, hence, we provide the following justifications. First, the stationary assumption is equivalent to assuming that the rate processes for the arrival and departure are stationary. For the departure, this could be justified if the system started a long time ago and has entered steady state. Second, regarding the Gaussian assumption, note that the departure process may not be Gaussian even if the arrival process is Gaussian. However, as we can see in (24), several different departure processes are superposed in a particular process $X_i(t)$. For example, if there are 16 traffic classes, then 15 different departure processes are being aggregated. Hence, a Gaussian model for $X_i(t)$ seems reasonable, which is a hypothesis that will be further validated via simulations.

Note that all the complexity of general-service disciplines is incorporated into a process $X_i(t)$ as in (23), which is of the same form as the one used in the case of a single queue. Thus, we are able to estimate the overflow probability $\mathbb{P}\{Q_i > x\}$ using exactly the same approach as in Section IV.

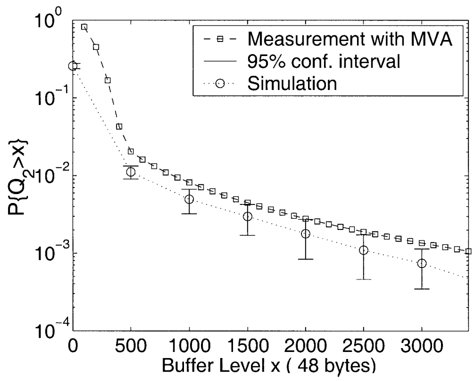
B. Numerical Results

We consider the queueing system shown in Fig. 9. For a particular queue i, let $X_i(t) = \sum_{k=1}^{t} Y_i(k)$. In this case, $Y_i(k)$ will be

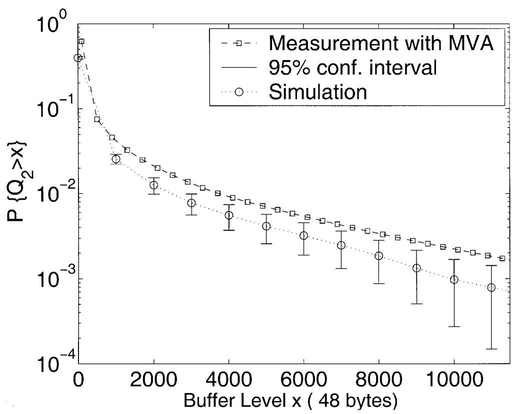
$$Y_i(k) = a_i(k) + \sum_{j \neq i} d_j(k) - C \tag{25}$$

where $a_i(k)$ and $d_i(k)$ are the amount of arrival and departure for class i traffic during the k'th time slot [see (24)]. Thus, we measure the arrival and the sum of the departure traffic less the traffic of the i'th class (equivalently, the total amount of departure traffic minus the amount of departure from the i'th class). Once we have obtained a sequence of $Y_i(k)$, $k = 1, 2, \ldots$, then the mean κ becomes $\kappa = -\mathbb{E}\{Y_i(k)\}$ and the variance function $\sigma_i^2(t)$ becomes $\sigma_i^2(t) = \mathrm{Var}\{\sum_{k=1}^{t} Y_i(k)\}$.

First, we consider a queueing system with a static priority scheduling policy. Since the lower priority queues get service only when all the higher queues are empty, it is sufficient to consider a two-priority queueing system and apply our method to the lower priority queue (LPQ) to estimate its overflow probability. In Fig. 10, we show the resulting overflow probabilities of the LPQ under two different input traffic scenarios. In Fig. 10(a), the HPQ and the LPQ are fed by ten multiplexed MPEG traffic sources and 200 multiplexed Ethernet traffic sources, respectively. The number of multiplexed sources and the total capacity C are chosen such that the mean traffic rate of the HPQ (queue 1)



(a)



(b)

Fig. 10. Buffer overflow probability for the lower priority queue under static priority scheduling policy. (a) $\lambda_1 = 0.6C$, $\lambda_2 = 0.2C$. (b) $\lambda_1 = 0.45C$, $\lambda_2 = 0.4C$.

is equal to $\lambda_1 = 0.6C$ and that of the LPQ (queue 2) to $\lambda_2 = 0.2C$. The process $B_2(0,t)$ for the LPQ then becomes $Ct - D_1(0,t)$, i.e., the service process to the LPQ is now time varying and can be obtained by measuring the departure traffic from the HPQ. In Fig. 10(b), we change the number of sources for each queue such that $\lambda_1 = 0.45C$ and $\lambda_2 = 0.4C$. We see that our approach produces accurate estimates over the whole range of buffer sizes in both cases.

Next, we consider a GPS service discipline. In order to simulate the GPS case, we divide the time axis into fixed time slots. Each class has its own queue with a positive coefficient ϕ_i $(\sum_i \phi_i = 1)$. Thus, at the beginning of each time slot, if all of the queues are backlogged, each traffic class receives service capacity $\phi_i C$. If some of the queues are empty, then the remaining capacity is distributed to the backlogged queues according to their coefficients proportionally. Fig. 11 shows the overflow probability for class-1 traffic (ten multiplexed MPEG traffic) in a GPS queueing system where class-2 traffic is composed of 200 multiplexed Ethernet sources and class-3 traffic is composed of three multiplexed JPEG sources. The coefficients ϕ_i for class i are chosen to be $\phi_i = 0.54, 0.16, 0.3$ and the resulting maximum utilizations $\rho_i = \lambda_i/(\phi_i C)$ of each queue are $\rho_i = 0.82, 0.85, 0.8$ for $i = 1, 2, 3$ respectively, where λ_i is the mean arrival rate of class i traffic. The total capacity then becomes $C = \sum_i \lambda_i/\rho_i$ from $\sum_i \phi_i = 1$. Note that our estimates are accurate over the whole range of the buffer sizes under consideration. While omitted here, we also observe that

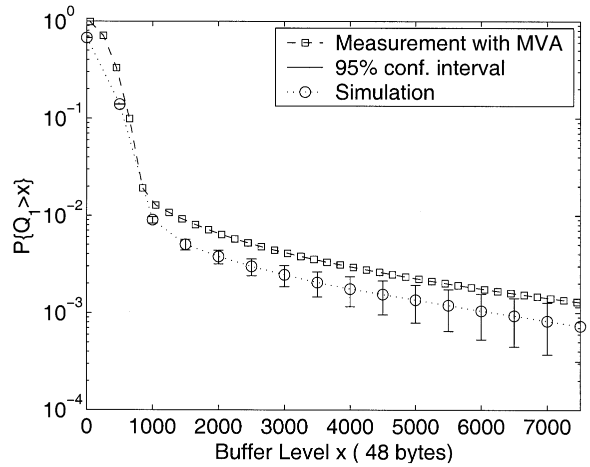


Fig. 11. Buffer overflow probability for class-1 queue under GPS scheduling policy.

the overhead $t_s(x,\kappa)/\hat{t}(x,\kappa)$ is still within a factor of six under these two different service disciplines. Hence, overall, our approach has two appealing attributes for possible deployment in a high-speed network. First, it is suitable for on-line estimation due to the stopping criterion. Second, it is versatile in the sense that it can be applied to any work-conserving service discipline.

VI. Discussion

A. Implication of the DTS and the Stopping Criterion

In this section, we discuss the implications of $\hat{t}(x,\kappa)$ and $t_s(x,\kappa)$ for traffic modeling in more detail. First, we make some observations.

- The value of $\hat{t}(x,\kappa)$ for any fixed x and κ depends only on the shape of the function $R(t)$ within a finite time interval $[0, t_s(x,\kappa)]$, and not on the limiting behavior of $R(t)$ (and, hence, the Hurst parameter).
- In order to produce the same $\mathbb{P}\{Q > x\}$, the model should be able to capture the correlations of lags up to $t_s(x,\kappa)$, not the DTS. Specifically, suppose that we know all the statistics for the Gaussian input process X_t and that, for a given buffer size and utilization, we calculate the DTS $\hat{t}(x,\kappa)$ and its upper bound $t_s(x,\kappa)$. Now, we generate another stationary Gaussian process Y_t whose mean and correlations are the same as those of X_t up to the DTS. Then, in this case, the undetermined region of $R(t)$ for $t > \hat{t}(x,\kappa)$ might change the location of the DTS for the process Y_t. However, if we match the mean and variance of the Y_t process up to the upper bound $t_s(x,\kappa)$ of X_t, instead of the DTS itself, then by Theorem 4, these two processes will give the same DTS $\hat{t}(x,\kappa)$ and $\sigma^2(\hat{t}(x,\kappa))$, no matter how the correlations of lags beyond $t_s(x,\kappa)$ behave.

For a large class of LRD processes, we have

$$R(t) \sim V t^{2H-2}$$

where $0.5 \leq H < 1$ is the Hurst parameter. Qualitatively, we can see that $R(t)$ tends to decrease more slowly as H increases, which makes $\hat{t}(x,\kappa)$ and $t_s(x,\kappa)$ larger. (For instance,

$\hat{t}(x,\kappa) = (H/1-H)(x/\kappa)$ for fBm processes.) Since $t_s(x,\kappa)$ is finite for any given x and κ, the LRD property by itself does not change the buffer distribution. Instead, it exerts its influence on the value of $\hat{t}(x,\kappa)$ by which the buffer behavior is determined. This explains why the overflow probability is so different for a *fixed* short-range dependent (SRD) model and an LRD model. Note that for any given x, we can always find an SRD model that has similar correlations up to $t_s(x,\kappa)$ and, hence, results in a similar overflow probability. In this sense, an LRD model should be considered as merely one of the many ways of modeling real traffic, rather than the only model to compute $\mathbb{P}\{Q > x\}$.

The DTS also has interesting implications for LRD traffic modeling. Previous attempts to find a suitable model for the real LRD traffic and to analyze the buffer based on the model have required estimating the Hurst parameter H, even for simple models such as fractional Brownian motion. However, our stopping criterion suggests that we do not have to estimate H at all in order to calculate the overflow probability for finite buffer levels. Further, note that the "variance-time plot," which has been used for estimating H [2], already contains all the information we need to calculate the overflow probability (i.e., $\mathrm{Var}\{X_t\}$ for all t). Hence, estimating H requires many more computations than all the steps we need for calculating the DTS and the resulting buffer overflow probability, via the stopping criterion.

B. On-Line Estimation of Traffic Statistics

In our methodology, we only need to know the statistics of traffic up to $t_s(x,\kappa)$ to find the DTS. This property reduces the burden from searching the statistics (e.g., $\sigma^2(t)$) over an infinite horizon to searching over a finite horizon. In this section, we will see the effect of on-line estimation of the traffic statistics on our stopping criterion and, hence, on the QoS estimates. Specifically, we will describe the estimation part of the procedure used, i.e., how we calculate and update λ and $R(n)$ as we observe the traffic on-line. Note that estimating such statistics based on observed data is a well-studied discipline and is not the focus of our research here. We only use a standard method which we describe next, however, an improved approach (for example, one that leads to faster or more reliable estimation), if available, could be substituted without changing the rest of the procedure for estimating the DTS.

Let X_i, $i = 1, 2, \ldots$, be the measurement data, i.e., the amount of traffic arrived in time slot i. We then estimate the mean rate using n samples by

$$\bar{\lambda}(n) = \frac{1}{n}\sum_{i=1}^{n} X_i \tag{26}$$

and the covariance $\bar{C}(k)$ using $n+k$ samples by

$$\bar{C}(k) = \frac{1}{n}\sum_{i=1}^{n}(X_i - \bar{\lambda}(n+k))(X_{i+k} - \bar{\lambda}(n+k)). \tag{27}$$

The variance function $\sigma^2(n)$ then can be estimated by

$$\bar{\sigma}^2(n) = n\bar{C}(0) + 2\sum_{k=1}^{n-1}(n-k)\bar{C}(k). \tag{28}$$

We first begin with N samples. As a new sample value is available after each time slot, we update $\bar{\lambda}(N)$ (i.e., replace it by $\bar{\lambda}(N+1)$) and $\bar{C}(0)$, and we calculate $\bar{C}(1)$ for the first time. In this way, as we observe more samples, we update the sample mean into $\bar{\lambda}(N+k)$ (which by ergodicity will converge to the true mean as k increases), and update the sample covariances $\bar{C}(j)$, $(j = 1, 2, \ldots, k-1)$ and newly calculate $\bar{C}(k)$. This is how we calculate and update the mean rate, $R(n) = \sigma^2(n)/n^2$ $(n \le k)$ and $g(k)$ as new samples are available. Note that by doing so, we have estimated the mean rate $\bar{\lambda}(N+k)$, $R(n)$ $(n = 1, 2, \ldots, k)$, and $g(k)$ using only $N+k$ samples. Then from the stopping criterion, we are able to estimate the overflow probability $\mathbb{P}\{Q > x\}$ once we observe $N + t_s(x,\kappa)$ samples.

From this specific procedure for estimating traffic statistics and the stopping criterion, we briefly outline the computational aspect of our algorithm. Given $N+n$ observations $(X_1, X_2, \ldots, X_{N+n})$, we can estimate $\bar{\lambda}(N+n)$ and $\bar{C}(j)$, $j = 1, 2, \ldots, n$ using (26) and (27). This will result in $O(n^2)$ computations for fixed N. To compute $\bar{\sigma}^2(j)$ for $j = 1, 2, \ldots, n$, we rewrite (28) as

$$\bar{\sigma}^2(n+1) = \bar{\sigma}^2(n) + \bar{C}(0) + 2Y(n) \tag{29}$$

where $Y(n) := \sum_{k=1}^{n}\bar{C}(k)$. From this recursion, we note that calculating $\bar{\sigma}^2(j)$ for $j = 1, 2, \ldots, n$ requires only $O(n)$ computations. Hence, with $O(n^2)$ computations, we can estimate the required statistics $\bar{\lambda}(N+n)$ and $\bar{\sigma}^2(j)$ $(j = 1, 2, \ldots, n)$ via (26)–(29). In contrast, we compare $g(n)(:= \max_{n/2\le u\le n} R(u))$ with $a(i,j)R(j)$ for $j = 1, 2, \ldots, n$ to obtain the upper bound on the DTS, and this causes $O(n)$ complexity. One could use standard techniques of estimating the power spectral density to speed up the estimation procedure.[5] For example, using the standard periodogram method [31] and fast Fourier transform to estimate the correlation function would result in $O(n \log n)$ complexity. However, it should be noted that the bottleneck in the procedure is still the actual estimation of the traffic statistics and not the stopping criterion.

Fig. 12 shows the resulting overflow probability estimates for 40 multiplexed MPEG traffic using the MVA approximation (9) for different N, the initial number of samples for estimating λ and $R(n)$. The solid line corresponds to the MVA curve in Fig. 4, where we used entire trace for computing λ and $R(n)$. As N increases, the on-line estimates are getting closer to the "true" value, as desired. For $N = 5000$, the on-line estimation is inaccurate since this number is too small compared with the total number of samples in the entire MPEG trace (about 170 000). Thus, it fails to capture the true statistics of the MPEG trace, which is strongly correlated over multiple time scales. The reason for this is that for small N (e.g., $N = 5000$), the estimated mean from the first N samples is smaller than the overall mean (obtained from entire trace) by approximately 10%. The underestimation of the mean arrival rate results in an underestimation of the overflow probability. This is a typical problem that one could encounter, because future values cannot be predicted with perfect accuracy (without assuming a specific model of the

[5] This was suggested by the anonymous reviewer.

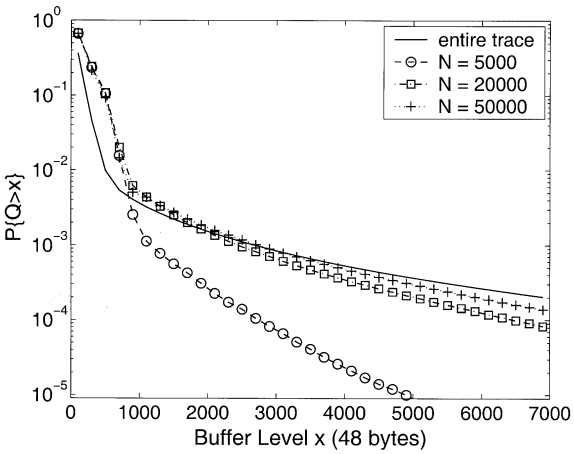


Fig. 12. On-line estimates of the overflow probability for 40 multiplexed MPEG traffic using MVA approximation with different number of samples. N is the number of samples for initial estimates of the mean $\bar{\lambda}$ and the covariance $\bar{C}(k)$ of the input traffic. The capacity is chosen such that the utilization of the queue is 85%.

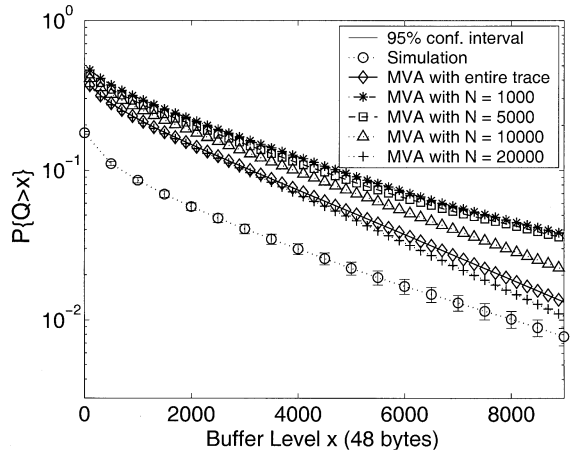


Fig. 13. Buffer overflow probability and its on-line estimates using MVA approximation with different number of samples. Input to the queue is 40 multiplexed AR traffic with first covariance coefficient 0.95. The mean, variance, and length of each AR traffic trace are set to those of the MPEG traffic trace. The capacity is chosen such that the utilization of the queue is 85%.

traffic). The problem could be alleviated by using a larger value of N, or, alternatively, by using a model or *a priori* statistical information about the traffic input. For example, if $N = 20\,000$ or larger, we see that the on-line estimates are reasonably accurate. In particular, we have observed that the estimated $\hat{t}(x, \kappa)$ and $t_s(x, \kappa)$ at $x = 7000 \cdot 48$ bytes, are 189 and 411, respectively. In other words, the whole curve for $N = 20\,000$ in Fig. 12 requires only 20 411 samples (about 12% of the total number of samples).

Next, we investigate the effect of the correlations or long-range dependence of the traffic on on-line measurements. We generate AR-1 traffic traces with the same mean and the same variance as those of the MPEG trace, but with different covariance functions. Fig. 13 shows the overflow probability and its on-line estimates for autoregressive (AR) traffic input with first covariance coefficient $c(1) = 0.95$ (i.e., with covariance function $c(k) = (0.95)^k$, $k \geq 0$). Here, each AR trace has the same number of samples as the MPEG trace (about 170 000). We generate different realizations $Z_1, Z_2, \ldots$, from which the confidence interval is derived. However, note that to obtain the on-line estimates, we just take the first realization Z_1 and the first $N + t_s(x, \kappa)$ samples from Z_1 for measurements, where $t_s(x, \kappa)$ in this case is determined by the stopping criterion. We believe that this is in accordance with the fact that in real world, we only see a single realization of a traffic flow and we can take measurements or any action only from this single realization. Note that this is the worse case situation in our experiments. Since different realizations result in different on-line estimates of $t_s(x, \kappa)$ and $\hat{t}(x, \kappa)$, our estimates will be better if we are allowed to repeat the same on-line measurements for each realization Z_i and then to estimate $t_s(x, \kappa)$ and the DTS by taking the average. In Fig. 13, we see that we now need only about 5% of the total number of samples to get an accurate estimate of the overflow probability. This trace, while highly correlated (first covariance coefficient of 0.95), is not as correlated as the actual MPEG video sequence, which is why we need a smaller fraction of samples to get good estimates of the overflow probability. We then repeat the same experiment except that the first covariance coefficient is now reduced to 0.7, which corresponds to a less bursty traffic flow. The results are plotted in Fig. 14. As we see, less than 1% of the total number of samples is shown to be enough to accurately estimate the overflow probability by utilizing our stopping criterion with on-line measurements of the traffic. We have observed similar results with voice traffic (which is less correlated), where only a very small percentage of the total traffic is required for obtaining good overflow probability estimates.

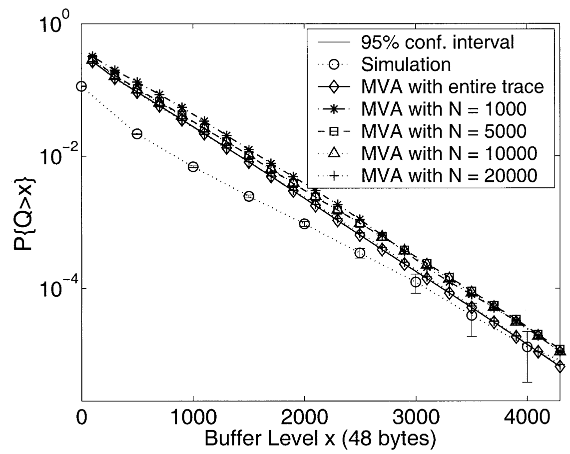


Fig. 14. Buffer overflow probability and its on-line estimates using MVA approximation with different number of samples. Input to the queue is 40 multiplexed AR traffic with first covariance coefficient 0.7. The mean, variance, and length of each AR traffic trace are set to those of the MPEG traffic trace. The capacity is chosen such that the utilization of the queue is 85%.

As mentioned earlier, thus far, we have used standard techniques (26)–(28) to estimate the statistics of the traffic on-line. It should be noted that any sophisticated statistical method to estimate the statistics will work independently from the rest of the procedure, i.e., finding the DTS.

VII. Conclusion

In this paper, we have developed a measurement-analytic approach for estimating the overflow probability at any multiplexing point of interest in the network. Our approach assumes that the input traffic can be characterized by a general class of Gaussian processes and, hence, is applicable to systems where a moderate to large number of traffic sources are multiplexed. Our work is motivated by the fact that, for estimating QoS parameters that correspond to rare events such as the overflow probability, we require a combined measurement and analytical framework. The DTS is useful in our framework because it allows the transformation of the problem of measuring rare events to a problem of measuring the input traffic itself, over a finite time window. However, a seemingly impossible difficulty with this approach is that the DTS is itself defined in terms of the variance of the traffic over all time. We show how to break this chicken-and-egg type of cycle and find a bound on the DTS. We investigate the tightness of this bound and provide numerical examples to illustrate our measurement-analytic approach. We then present a way of extending our approach to a multi-buffered queueing system with work-conserving service disciplines, such as static priority queueing and GPS scheduling, and show that our approach performs well. We also provide some interesting insights gained from using our approach. Since our approach only requires a finite window of measurements (usually small, depending, of course, on the DTS and its bound), it has significant value from an on-line measurement point of view and differs from works in the literature that require knowledge of the entire trace of the traffic.

Appendix A

Proof of Proposition 3

Let $h(\cdot) = (\cdot)^2$. Then, the function $S(t)$ in Proposition 2 becomes $S(t) = r(t) + t(\mathbb{E}\{X_t/t\})$. Since $\mathbb{E}\{X_t/t\}$ is constant for a stochastic process X_t with stationary increments, subadditivity of $r(t)$ follows from Proposition 2. For the second result, let $I(k,t) := [2^{k-1}t, 2^k t]$ where k is a positive integer. Then by definition

$$g(2^k t) = \max_{u \in I(k,t)} R(u).$$

First, note that for any $v \in I(k,t)$, there exist $p, q \in I(k-1,t)$ such that $v = p + q$. Since

$$\begin{aligned} R(v) =& R(p+q) \\ \leq& \frac{p}{p+q} R(p) + \frac{q}{p+q} R(q) \\ \leq& \max\{R(p), R(q)\} \end{aligned}$$

from the subadditivity of $r(t)$, we have

$$\begin{aligned} R(v) &\leq \max_{u \in I(k-1,t)} R(u) \\ &= g(2^{k-1}t). \end{aligned} \tag{30}$$

Since (30) holds for any $v \in I(k,t)$, by taking maximum over the interval $I(k,t)$, we have shown that $g(2^k t)$ is decreasing in $k = 0, 1, 2, \ldots$, for any fixed t.

Now, for any $s \geq t$, we can take $k \geq 1$ such that $s \in I(k,t)$. Thus, finally, we get

$$\begin{aligned} R(s) &\leq \max_{u \in I(k,t)} R(u) \\ &= g(2^k t) \\ &\leq g(2^{k-1}t) \\ &\leq \cdots \leq g(t). \end{aligned}$$

This completes the proof. ∎

Appendix B

Proof of Proposition 5

Clearly, $t_s(x,\kappa) \geq px/\kappa$ for any p that satisfies (11), and note that

$$\left(\frac{p}{p+1}\right)^2 R\left(\frac{px}{\kappa}\right) = \frac{\sigma^2\left(\frac{px}{\kappa}\right)}{\left(\frac{px}{\kappa} + \frac{x}{\kappa}\right)^2}.$$

Let $p^*(x)$ be the maximizer of the right-hand side of (14) for a given x/κ; then we already know that $p^*(x)x/\kappa = \hat{t}(x,\kappa) \sim (H/1-H)(x/\kappa)$ [13]. So, we have

$$\lim_{x\to\infty} p^*(x) = \frac{H}{1-H}. \tag{31}$$

Thus, $t_s(x,\kappa) \uparrow \infty$ as $x \uparrow \infty$. Also, from (13), we know that

$$\lim_{t\to\infty} \frac{R(t)}{Vt^{2H-2}} = 1. \tag{32}$$

In other words, for any given $\epsilon > 0$, we can choose $M > 0$ such that

$$(1-\epsilon)Vt^{2H-2} \leq R(t) \leq (1+\epsilon)Vt^{2H-2}$$

for all $t \geq M$. Thus, by definition of $g(t)$, we have

$$\begin{aligned} (1-\epsilon)V\left(\frac{t}{2}\right)^{2H-2} &\leq g(t) \\ &\leq (1+\epsilon)V\left(\frac{t}{2}\right)^{2H-2} \end{aligned}$$

for all $t \geq 2M$. Hence

$$\lim_{x\to\infty} \frac{g(t_s(x,\kappa))}{V\left(\frac{t_s(x,\kappa)}{2}\right)^{2H-2}} = 1. \tag{33}$$

Now, from (14), $g(t_s(x,\kappa))$ can be represented as

$$g(t_s(x,\kappa)) = \left(\frac{p^*(x)}{p^*(x)+1}\right)^2 R\left(\frac{p^*(x)x}{\kappa}\right).$$

By rewriting this as

$$\frac{g(t_s(x,\kappa))}{V\left(\frac{t_s(x,\kappa)}{2}\right)^{2H-2}} = \left(\frac{p^*(x)}{p^*(x)+1}\right)^2 \frac{R\left(\frac{p^*(x)x}{\kappa}\right)}{V\left(\frac{p^*(x)x}{\kappa}\right)^{2H-2}} \left(\frac{\frac{p^*(x)x}{\kappa}}{\left(\frac{t_s(x,\kappa)}{2}\right)}\right)^{2H-2}$$

we have

$$\lim_{x\to\infty} \frac{t_s(x,\kappa)}{2\hat{t}(x,\kappa)} \frac{\hat{t}(x,\kappa)}{\frac{p^*(x)x}{\kappa}} \left(\frac{p^*(x)}{p^*(x)+1}\right)^{1/H-1} = 1$$

from (32) and (33). Therefore, the result follows from (31). ■

REFERENCES

[1] D. Y. Eun and N. B. Shroff, "A measurement-analytic framework for QoS estimation based on the dominant time scale," in *Proc. IEEE INFOCOM*, vol. 2, Anchorage, AK, 2001, pp. 924–932.

[2] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, "On the self-similar nature of ethernet traffic (Extended Version)," *IEEE/ACM Trans. Networking*, vol. 2, pp. 1–15, Feb. 1994.

[3] J. Beran, R. Sherman, M. S. Taqqu, and W. Willinger, "Long-range dependence in variable-bit-rate video traffic," *IEEE Trans. Commun.*, vol. 43, pp. 1566–1579, Feb. 1995.

[4] A. I. Erramilli, O. Narayan, and W. Willinger, "Experimental queueing analysis with long-range dependent packet traffic," *IEEE/ACM Trans. Networking*, vol. 4, pp. 209–223, Apr. 1996.

[5] D. P. Heyman and T. V. Lakshman, "What are the implications of long-range dependence for VBR video traffic engineering?," *IEEE/ACM Trans. Networking*, vol. 4, pp. 301–317, June 1996.

[6] B. K. Ryu and A. Elwalid, "The importance of long-range dependence of VBR video traffic in ATM traffic engineering: Myths and realities," in *Proc. ACM SIGCOMM*, San Francisco, CA, Aug. 1996, pp. 3–14.

[7] M. Montgomery and G. D. Veciana, "On the relevance of time scales in performance oriented traffic characterization," in *Proc. IEEE INFOCOM*, San Francisco, CA, 1996, pp. 513–520.

[8] M. Grossglauser and J.-C. Bolot, "On the relevance of long-range dependence in network traffic," *IEEE/ACM Trans. Networking*, vol. 7, pp. 629–640, Oct. 1999.

[9] C. Courcoubetis, V. A. Siris, and G. D. Stamoulis, "Application of the many sources asymptotic and effective bandwidths to traffic engineering," *Telecommun. Syst.*, vol. 12, pp. 167–191, 1999.

[10] N. G. Duffield and N. O'Connell, "Large deviations and overflow probabilities for the general single server queue, with application," *Proc. Cambridge Philos. Soc.*, vol. 118, pp. 363–374, 1995.

[11] P. W. Glynn and W. Whitt, "Logarithmic asymptotics for steady-state tail probabilities in a single-server queue," *J. Appl. Probabil.*, pp. 131–155, 1994.

[12] J. Choe and N. B. Shroff, "A central limit theorem based approach for analyzing queue behavior in high-speed networks," *IEEE/ACM Trans. Networking*, vol. 6, pp. 659–671, Oct. 1998.

[13] ——, "Queueing analysis of high-speed multiplexers including long-range dependent arrival processes," in *Proc. IEEE INFOCOM*, vol. 2, New York, NY, Mar. 1999, pp. 617–624.

[14] ——, "Use of supremum distribution of Gaussian processes in queueing analysis with long-range dependence and self-similarity," *Stochast. Models*, vol. 16, no. 2, Feb. 2000.

[15] D. D. Botvich and N. Duffield, "Large deviations, the shape of the loss curve, and economies of scale in large multiplexers," *Queueing Syst.*, vol. 20, pp. 293–320, 1995.

[16] C. Courcoubetis and R. Weber, "Buffer overflow asymptotics for a buffer handling many traffic sources," *J. Appl. Probabil.*, vol. 33, pp. 886–903, 1996.

[17] F. P. Kelly, "Notes on effective bandwidths," in *Stochastic Networks: Theory and Applications*. Oxford, U.K.: Oxford Univ. Press, 1996, pp. 141–168.

[18] C. Courcoubetis and V. A. Siris, "Measurement and analysis of real network traffic," presented at the *7th Hellenic Conf. Informatics (HCI'99)*, Ioannina, Greece, 1999.

[19] J. Jacod and A. N. Shiryaev, *Limit Theorems for Stochastic Processes*. New York: Springer-Verlag, 1987.

[20] G. L. Choudhury, D. M. Lucantoni, and W. Whitt, "Squeezing the most out of ATM," *IEEE Trans. Commun.*, vol. 44, pp. 203–217, Feb. 1996.

[21] N. B. Shroff and M. Schwartz, "Improved loss calculations at an ATM multiplexer," *IEEE/ACM Trans. Networking*, vol. 6, pp. 411–422, Aug. 1998.

[22] J. Choe and N. B. Shroff, "On the supremum distribution of integrated stationary Gaussian processes with negative linear drift," *Advances Appl. Probabil.*, vol. 31, no. 1, pp. 135–157, 1999.

[23] C. Courcoubetis, V. A. Siris, and G. D. Stamoulis, "Traffic equivalence and substitution in a multiplexer," in *Proc. IEEE INFOCOM*, vol. 3, 1999, pp. 1239–1247.

[24] E. W. Knightly and N. B. Shroff, "Admission control for statistical QoS: Theory and practice," *IEEE Network*, vol. 13, no. 2, pp. 20–29, Mar. 1999.

[25] J. Choe and N. B. Shroff, "Efficient Gaussian traffic modeling in high-speed networks," Purdue Univ., West Lafayette, IN, 2000.

[26] O. Rose, "Statistical properties of MPEG video traffic and their impact on traffic modeling in ATM systems," in *Proc. 20th Annu. Conf. Local Computer Networks*, Minneapolis, MN, 1995, pp. 397–406.

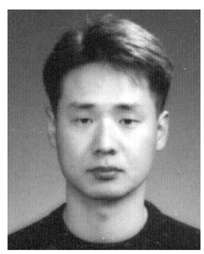
[27] D. Bertsimas, I. Ch. Paschalidis, and J. N. Tsitsiklis, "Large deviations analysis of the generalized processor sharing policy," *Queueing Syst.*, vol. 32, pp. 319–349, 1999.

[28] S. Delas, R. Mazumdar, and C. Rosenberg, "Tail asymptotics for HOL priority queues handling a large number of independent stationary sources," *Queueing Syst.*, vol. 40, pp. 183–204, Mar. 2002.

[29] M. Uitert and S. Borst, "Generalized processor sharing networks fed by heavy-tailed traffic flows," in *Proc. IEEE INFOCOM*, vol. 1, Anchorage, AK, 2001, pp. 269–278.

[30] C. Kotopoulos, N. Likhanov, and R. Mazumdar, "Asymptotic analysis of the GPS system fed by heterogeneous long-tailed sources," in *Proc. IEEE INFOCOM*, vol. 1, Anchorage, AK, 2001, pp. 299–308.

[31] J. G. Proakis and D. G. Manolakis, *Digital Signal Processing: Principles, Algorithms, and Applications*, 2nd ed. New York: Macmillan, 1992.



Do Young Eun received the B.S. and M.S. degrees in electrical engineering from the Korea Advanced Institute of Science and Technology, Taejon, Korea, in 1995 and 1997, respectively. Since 1998, he has been working toward the Ph.D. degree in electrical and computer engineering at Purdue University, West Lafayette, IN.

His research interests include network modeling and analysis, queueing analysis, resource allocation, and network measurement.

Mr. Eun received the Bronze Prize of the Humantech Thesis Prize competition, sponsored by Samsung Electronics, in 1997.

Ness B. Shroff (S'91–M'93–SM'01) received the Ph.D. degree from Columbia University, New York, NY, in 1994.

He is currently an Associate Professor with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN. His research interests include the areas of wireless and wireline communication networks, especially fundamental problems in the design, performance, scheduling, capacity, pricing, and control of these networks.

Dr. Shroff received the National Science Foundation CAREER award in 1996. He is an Editor for IEEE/ACM TRANSACTIONS ON NETWORKING and the *Computer Networks Journal*, and a past Editor of IEEE COMMUNICATIONS LETTERS. He was the Conference Chair for the 14th Annual IEEE Computer Communications Workshop, October 1999, and Program Cochair for the Symposium on High-Speed Networks, IEEE GLOBECOM 2001. He is also the Technical Program Cochair for IEEE INFOCOM 2003 and Panel Cochair for ACM Mobicom 2002.