Measurement-based Per-flow Statistical Delay Guarantees

Kartik Gopalan and Tzi-cker Chiueh

{kartik,chiueh}@cs.sunysb.edu

Computer Science Department, Stony Brook University, Stony Brook, NY - 11794

ABSTRACT

Admission control algorithms that provide statistical delay guarantees to real-time flows are of considerable interest since they can exploit statistical multiplexing of traffic to achieve higher link utilization. In this paper, we address the issue of providing per-flow statistical delay guarantees to real-time flows that traverse a link serviced by a rate-based scheduler. In such a scenario, each flow can specify a limit on the fraction of traffic that exceeds its delay bound requirements. We present a novel measurement-based admission control technique, called Measured Probability algorithm, that can guarantee a distinct delay bound as well as a delay violation probability to each flow sharing a single link. The algorithm exploits the fact that the actual delay experienced by most packets of a real-time flow is usually far smaller than the worst-case delay bound dictated by the rate assigned to the flow. Flows that can tolerate occasional delay bound violations can be assigned fewer resources than their worst-case requirement. The algorithm dynamically measures the ratio of actual delay experienced to the worst-case delay for each packet traversing the link and constructs a cumulative probability distribution function (CPDF) of this ratio. The CPDF is then used in estimating flow resource requirements whenever a new flow joins the system. Preliminary simulations indicate that, in addition to providing per-flow statistical delay guarantees, our algorithm can potentially improve the number of admitted flows by a factor of 10 *and the link utilization by a factor of* 20 *when delay violation probability as small as* 10^{-6} *is allowed.*

1. INTRODUCTION

Network applications such as streaming media, voice over IP (VoIP) and content distribution generate real-time flows that have tight delay requirements, but can tolerate certain level of missed deadlines or packet losses. Guaranteeing performance to such real-time flows involves provisioning resources during admission control and enforcing the usage of the assigned resources at run-time. A simple approach to this problem is to perform *deterministic* admission control that allocates sufficient resources to each flow such that the flows never encounter any excess delay. However, a deterministic approach pays the price of severely under-utilizing the link's resources due to two reasons. First, the deterministic approach ignores the fact that applications themselves are resilient to small fraction of packets experiencing excess delays. Secondly, statistical multiplexing among traffic from different real-time flows ensures that most of the packets rarely, if ever, approach worst-case delays. Alternatively, one could exploit the above two facts and increase link utilization by providing *statistical delay guarantees*, i.e., by associating a delay violation probability along with each flow's delay bound. A *statistical admission control* algorithm (i.e. one that provides statistical delay guarantees) can thus reduce the amount of resource allocated to each flow and maximize the number of flows admitted with guarantees.

One way to provide statistical guarantees is to use the well known technique of *measurement-based admission control* (MBAC). MBAC algorithms use the dynamically measured traffic characteristics of flows currently traversing the link to determine whether a new flow can be admitted. In this work, we present a novel MBAC algorithm, called *measured probability* (MP) algorithm, that provides statistical delay guarantees to real-time flows on a link serviced by a rate-based packet-by-packet scheduler (such as virtual clock[1] or WFQ[2, 3]). The two main features of our algorithm are (a) it provides a distinct statistical delay guarantee to each real-time flow sharing the link and (b) it exploits statistical multiplexing to increase link utilization and admit more flows in comparison to purely deterministic admission control.

We are interested in rate-based schedulers since, in their case, the relationship between delay bound of a flow and the amount of resource consumed by the flow can be explicitly specified. In contrast, for non rate-based schedulers, such as Earliest Deadline First (EDF), the resource-delay relationship for each flow is difficult to determine (if not impossible) and the admission control process is more complicated. Hence, even though non rate-based schedulers can potentially provide higher link utilization, it is difficult to provide statistical guarantees (such as bounds on loss rate or delay-violation probability) on a per-flow basis. A related problem is to provide end-to-end statistical delay guarantees to flows that traverse multiple links in a network. This problem is not covered by the scope of our paper and has been addressed in [4].

The problem of providing various kinds of statistical performance guarantees has received considerable research attention. Kurose[5] derived probabilistic bounds on delay and buffer occupancy of flows using the concept of stochastic ordering for network nodes that use FIFO scheduling. Reisslein *et al.*[6] have derived statistical delay bounds for traffic flows in a single link and network settings. They ap-

proximate the loss probability at a link using independent Bernoulli random variables. However they consider a fluid traffic model rather than a packetized model. Elwalid and Mitra[7] have proposed a scheme to provide statistical QoS guarantees in the GPS service discipline for two guaranteed traffic classes and one best effort class. Again a fluid traffic model was considered. Schemes for providing statistical QoS in networks using EDF scheduling were proposed by Andrews[8] and Sivaraman[9]. Unlike the rate-based schedulers considered here, EDF decouples rate and delay guarantees at the expense of admission control complexity. Leibeherr *et al.*[10] proposed the notion of effective service curves as a probabilistic bound on service received by a flow.

Several MBAC algorithms address flow QoS requirements along the dimensions of the bandwidth or aggregate loss-rate. However they do not address QoS requirements of real-time flows that need distinct statistical delay guarantees in the context of rate-based schedulers. Breslau *et al.*[11] performed a comparative study of several MBAC algorithms[12–16] and concluded that none of them are capable of accurately achieving loss targets. Qiu and Knightly[12] proposed an MBAC scheme that measures maximal rate envelopes of aggregate traffic flows. Boorstyn *et al.*[17] developed the notion of effective envelopes that capture the upper bounds of multiplexed traffic with a high certainty and can be used to bound the amount of traffic that can be provisioned on a link with statistical guarantees.

The rest of the paper is organized as follows. In section 2, we discuss the traffic and scheduling models underlying our work and propose the MP algorithm. In Section 3, we present preliminary performance results. Section 4 gives a summary of main research results and explores future directions.

2. STATISTICAL ADMISSION CONTROL

We define a *real-time flow* F_i as a packet stream that requires each of its packets to be serviced by the scheduler within a delay bound D_i and with a delay violation probability no greater than P_i. For instance, if $D_i = 30ms$ and $P_i = 10^{-7}$, it means that no more than a fraction 10^{-7} of packets belonging to the flow can experience a delay greater than $30ms$. When a real-time flow F_i arrives, it requests a delay bound D_i and delay violation probability bound P_i from the admission control module. We assume that each flow's incoming traffic is regulated by a token bucket with bucket depth σ_i and token rate ρ_i. The amount of flow F_i traffic arriving at the scheduler in any time interval of length τ is bounded by $(\sigma_i + \rho_i\tau)$. *

*Our algorithm remains essentially unchanged in case of dual/cascaded leaky bucket regulators.

2.1. Scheduling and Worst-case Delay Bound

In this work, we consider the context of a single link with capacity C. Packets arriving at the link are served using virtual clock service discipline[1]. Conceptually, the virtual clock scheduler maintains a sorted priority queue of incoming packets. During admission control each flow F_i is assigned a bandwidth reservation B_i ($B_i \geq \rho_i$) in order to satisfy its delay requirements. Whenever a packet k, belonging to flow F_i and having length L_{ik}, arrives at the scheduler at time t_{ik}, a per-flow state variable V_i is updated as follows.

$$V_i = max\{t_{ik}, V_i\} + L_{ik}/B_i \tag{1}$$

The packet is stamped with value V_i and inserted into the sorted priority queue using the stamped value as the priority. Packets are served by the scheduler according to the increasing order of priority. It can be shown that the worst-case delay experienced by a packet belonging to flow F_i under virtual clock service discipline is given by the following expression[18], where L_{max} is the maximum packet size.

$$D_i^{wc} = \frac{\sigma_i}{B_i} + \frac{L_{max}}{B_i} + \frac{L_{max}}{C} \tag{2}$$

The first component of the delay is the fluid fair queueing delay, the second component is the packetization delay and the third component is scheduler's non-preemption delay. Note that the expression for worst-case delay bound, given in Equation 2, also holds for WFQ service discipline. Hence the algorithm presented here for virtual clock also holds for WFQ. In this paper, we do not include the treatment of WFQ in the interest of brevity, although it would be an interesting exercise to examine its performance in the context of the MP algorithm.

2.2. Measurement-based Delay-Probability-Bandwidth Correlation

Statistical delay guarantees assist in reducing the bandwidth reservation for each flow by exploiting the fact that worst-case delay is rarely the norm for packets traversing a link. Assume that for each link, the system keeps a run-time measurement of the ratio of the actual packet delay experienced to the worst-case delay for each packet. We then have a cumulative probability distribution function (CPDF) $Prob(r)$ of this ratio based on past measurements. $Prob(r)$ gives the probability that the ratio between the actual delay encountered by a packet and its worst-case delay is smaller than or equal to r. Conversely, $Prob^{-1}(p)$ gives the maximum ratio of actual delay to worst-case delay that can be guaranteed with a probability of p. Figure 1 shows an example of the CPDF $Prob(r)$. The CPDF would typically be maintained over a sliding measurement window. Given the measured estimate of functions $Prob(r)$ and $Prob^{-1}(p)$, we can derive the minimum bandwidth B_i^{min} required to

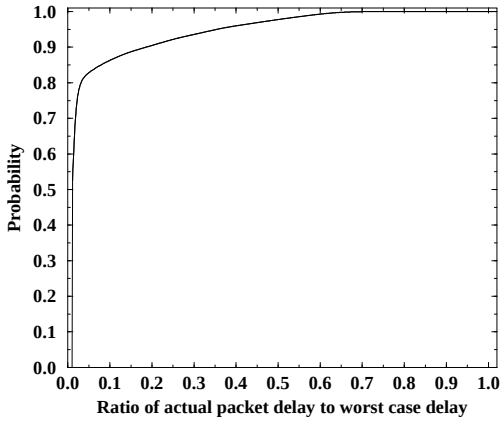

Figure 1. Example of cumulative probability distribution function.

satisfy flow F_i's delay-probability requirement (D_i, P_i) using the following heuristic.

$$D_i = \left(\frac{\sigma_i + L_{max}}{B_i^{min}} + \frac{L_{max}}{C} \right) \times Prob^{-1}(1 - P_i) \quad (3)$$

Equations 3 states that in order to obtain a delay bound of D_i with a delay violation probability bound of P_i, we need to reserve a minimum bandwidth of B_i^{min} which can guarantee a worst-case delay of $D_i / Prob^{-1}(1 - P_i)$.

2.3. Measured Probability Algorithm

With the delay-probability-bandwidth correlation function in place, we now present the MP algorithm in Figure 2. Assume that $N - 1$ flows are currently being served by the scheduler and flow N arrives for admission. The algorithm first calculates B_i^{min} for each of the N flows (including the new one) according to Equation 3. Next, each B_i^{min} is proportionally scaled to $B_i^{'}$ such that the sum of $B_i^{'}$ values is equal to γ times the link capacity C, where $\gamma \geq 1$. The parameter γ is called the *bandwidth over-provisioning factor* and determines the extent to which bandwidth can be reserved in excess of link capacity. This parameter can be used to make the algorithm more or less aggressive in admitting new flows. The new flow is accepted only if none of the flows is projected to have a smaller $B_i^{'}$ value than the long-term average ρ_i. In order to calculate true projected bandwidth share for each flow, the algorithm scales the B_i^{min} values to B_i such that the sum of B_i's exactly fits the link capacity C. The complexity of the algorithm is O(N) where N is the number of flows being considered.

2.4. Discussion

Let us look at a few issues that deserve mention. First issue is that, during initial stages, when there are no active flows at the link, there is no measurement history that can be used to calculate the CPDF $Prob(r)$. Our solution is to admit the initial flows using deterministic admission control. During this time, the measurement history is simultaneouly built up. Once no more flows can be admitted using deterministic approach, we switch to statistical admission control and use the history built up during the deterministic phase.

Input : (a) $(D_i, P_i, \sigma_i, \rho_i)$ for each flow F_i, $1 \leq i \leq N$.
(b)The measured $Prob(r)$ and $Prob^{-1}(p)$ functions.
(c)The link capacity C and current average link utilization U.

for $i = 1$ to N
Calculate B_i^{min} according to Equation 3.

for $i = 1$ to N
$B_i^{'} = \frac{B_i^{min}}{\sum_{j=1}^{N} B_j^{min}} \times C \times \gamma$
if $B_i^{'} < \rho_i$ then reject flow F_N and exit.

/*Flow F_N can be admitted.*/
for $i = 1$ to N
$B_i = \frac{B_i^{min}}{\sum_{j=1}^{N} B_j^{min}} \times C$
Reserve bandwidth B_i for flow F_i.

Figure 2. The MP algorithm to determine whether a new flow F_N can be admitted such that each flow F_i, $1 \leq i \leq N$, can be guaranteed a delay bound D_i and delay violation probability P_i.

Second issue concerns stability of the CPDF $Prob(r)$. If we admit successive flows within short interval of time, the measurement process may not get enough time to stablize and fully reflect the effect of the new flow on the delay distribution of packets. We address this problem by enforcing a time gap between every two successive admissions that allows the measured CPDF to stablize.

A third important issue concerns the fact that when we admit a new flow F_N, we use the CPDF that only reflects the behaviour of already admitted $N-1$ flows and does not account for the future impact of new flow on the CPDF. This might lead to a somewhat over-optimistic algorithm where a new admission might push the delay violation of some already admitted flows above the acceptable limit. Solving this problem involves having an accurate estimate of the delay distribution of the new flow that can be combined with measured CPDF of the already admitted flows before performing admission control. This is difficult to achieve if we do not wish to assume apriori traffic models for the new flow. In practice, the later statistical admission phase will usually have a large enough number of flows that are already admitted during the deterministic phase. Thus the unaccounted impact of a single new flow on the overall CPDF of a much larger set of pre-existing flows will be minimal during the statistical phase. Additionally, whenever the number of admitted flows falls below that permissible by deterministic admission control, we switch over to deterministic phase again and so on.

Source Model	(ON,OFF) Times (secs)	Peak Rate (kbps)	Avg. Rate (kbps)
EXP1	$(.325, .325)$	64	32
EXP2	$(.325, 6.175)$	640	32
PARETO	$(.325, .325)$	64	32

Table 1: ON-OFF source models used in simulation.

3. PERFORMANCE

In this section, we study the performance of the MP algorithm using the *ns-2* simulator. A single link is configured with 10Mbps capacity and packets arriving at the link are served by a virtual clock scheduler as described in Section 2.1. The traffic source models used in our simulation are based on three ON-OFF source models - EXP1, EXP2 and PARETO - which are also used in [11]. EXP1 and EXP2 sources generate packets with exponential inter-arrival times whereas PARETO sources space packets according to Pareto distribution. The details of the three source models are given in Table 1. EXP1 corresponds to voice traffic generated in VoIP applications and EXP2 corresponds to a burstier version of EXP1 traffic. Aggregation of PARETO sources produce the long-range dependence effects normally observed with audio traffic. All packets are 128 bytes long. Traffic from each admitted flow is policed by a token bucket whose parameters depend on flow's peak rate and delay bound requirements.

Each new flow arriving at the link requests a guarantee on delay bound and a delay violation probability. The admission control algorithm decides whether to admit or reject the flow and how much bandwidth to allocate according to algorithm in Figure 2. Each admitted flow is assigned a queue limit of twice its token bucket depth and is permitted to send traffic on the link for the duration of simulation. A flow generator initiates each flow by uniformly selecting one of the three source models. Flow inter-arrival times are exponentially distributed with a mean of 100 seconds. The MP algorithm maintains a stablizing period of at least 100 (simulated) seconds between any two flow admissions so that the measured CPDF is allowed to stablize before being used to admit another flow. The CPDF is maintained over a sliding measurement window of 3000 seconds. Each simulation run lasts for $100,000$ seconds. Each result consists of the average and standard deviation (indicated by vertical error bars in the graphs), over 5 runs using different random number seeds. For preliminary results, we experiment only with synthetic workloads and dynamic flow arrivals. However, it is also important to study the behaviour of admission control using more realistic trace-based workloads under both dynamic flow arrivals and departures. This aspect will be addressed in future work.

First we examine the performance of MP algorithm in comparison to deterministic admission control. The latter assigns bandwidth to each flow assuming a zero delay violation probability. With MP algorithm, the delay violation probability for each flow is 10^{-6}. Figure 3 and 4 plot the number of flows admitted and link utilization, respectively, as delay is varied. Observe that MP algorithm admits roughly 10 times more number of flows and achieves 20 times better utilization than deterministic admission control when delay violation probability as small as 10^{-6} is allowed. The gain comes from the fact that large majority of packets experience just 1% to 2% of the worst-case delay dictated by their assigned bandwidth.

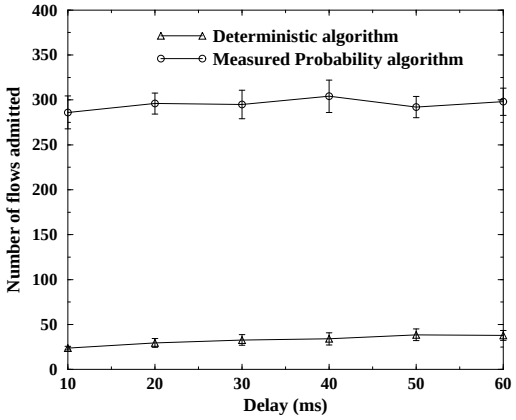

Figure 3. Number of admitted flows vs. delay. Delay violation probability = 10^{-6} and $\gamma = 5$.

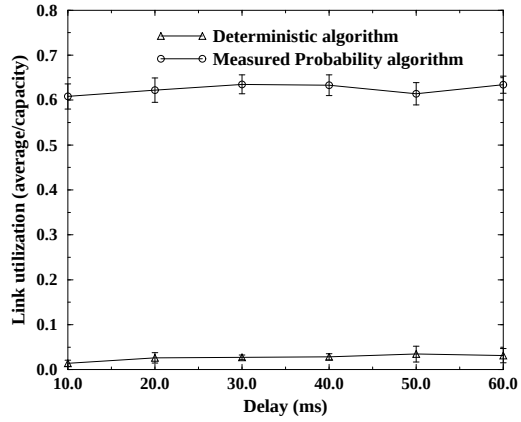

Figure 4. Average utilization vs. delay. Delay violation probability = 10^{-6} and $\gamma = 5$.

Next we show that the MP algorithm indeed provides distinct guarantees on heterogeneous delay violation probabilities for different flows. We examine the scenario in which incoming flows request the same delay bound of 30ms but different guarantees on delay violation probability, which is uniformly distributed among the four values 10^{-4}, 10^{-5}, 10^{-6} and 10^{-7}. Figure 5 plots the actual fraction of pack-

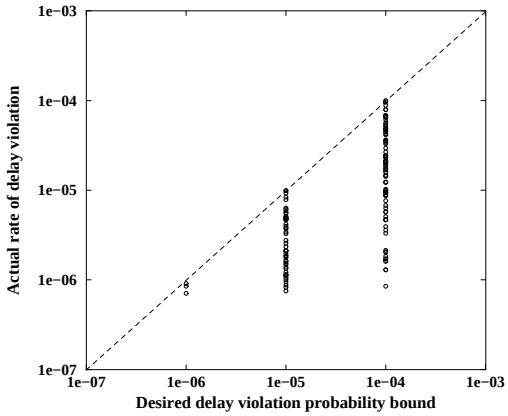

Figure 5. The MP algorithm satisfies distinct per-flow delay violation guarantees. Delay bound for each flow is 30ms and $\gamma = 5$.

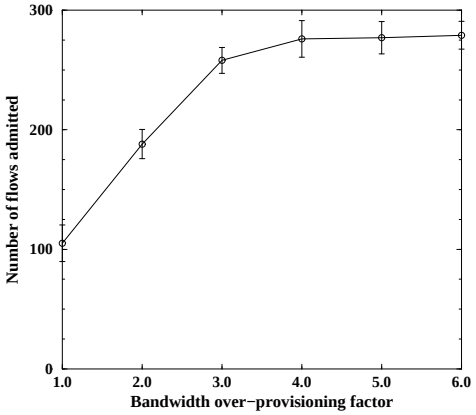

Figure 6. Number of admitted flows vs. bandwidth over-provisioning factor γ. Delay bound for each flow is 30ms and violation probability is chosen uniformly between 10^{-4} and 10^{-7}.

ets exceeding their delay bound against the desired violation probability for each flow experiencing excess delay. The fact that all data points are below the dotted line indicates that the actual delay violation rate is smaller than the maximum permissible for each flow. Furthermore the figure shows that flows that can tolerate higher fraction of delay violations indeed experience a higher rate of violation than flows with lower tolerance. The MP algorithm is able to distinguish among flows in terms of delay violation rates because it assigns service bandwidth to flows in inverse proportion to their tolerance to delay violations. This translates to higher service bandwidth (and hence dynamic priority) for packets belonging to flows with low delay tolerance and vice-versa.

Figures 6 and 7 show that both the number of admitted flows and link utilization increase dramatically as the bandwidth over-provisioning factor γ is increased. As seen earlier, γ controls the extent to which bandwidth assigned to a flow can be smaller than its long term rate. Thus γ can be used as a tuning knob to control the aggressiveness of the MP algorithm.

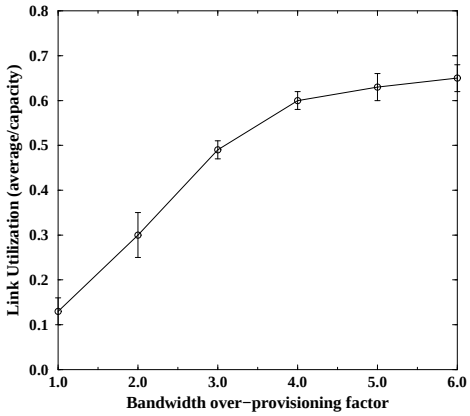

Figure 7. Number of admitted flows vs. bandwidth over-provisioning factor γ. Delay bound for each flow is 30ms and violation probability is chosen uniformly between 10^{-4} and 10^{-7}.

4. SUMMARY AND FUTURE DIRECTIONS.

In this paper, we have proposed a new admission control algorithm, called *measured probability* (MP) algorithm, that provides per-flow statistical delay guarantees (i.e., both delay bound and delay violation probability bound) on links serviced by rate-based schedulers. The MP algorithm uses the concept of measurement-based admission control to exploit statistical multiplexing among flows traversing a link. Our preliminary results show that the algorithm provides significant gain in number of admitted flows and link utilization, while meeting the per-flow delay requirements.

In addition to a more comprehensive performance analysis of our algorithm, we wish to devise simple mechanisms that can pre-estimate an incoming flow's impact on the measured probability estimates. In its current form, the MP algorithm uses the measured probability distribution directly in calculating resource requirements during a new admission and does not account for the future impact of an incoming flow's traffic.

We are also interested in using our algorithm as a building block for more realistic case where flows traverse multiple links served by rate-based schedulers. It is shown in earlier research [6, 7, 19] that it is desirable to smooth a flow's traffic as much as possible at the ingress and perform bufferless multiplexing in the network's interior. Thus, in a packetized environment, a flow's per-link delay budget will be smaller, consisting of only the packetization and non-preemption delay components in Equation 2. This leaves a

smaller margin of error and makes it more challenging to design a robust statistical admission control algorithm.

Acknowledgement : We would like to thank Pradipta De, Srikant Sharma and Ashish Raniwala for insightful discussions and comments that greatly improved the ideas presented in this paper.

REFERENCES

1. L. Zhang, "Virtual Clock: A new traffic control algorithm for packet switching networks," in *Proc. of ACM SIGCOMM'90*,
2. A. Demers, S. Keshav, and S. Shenker, "Analysis and simulation of a fair queuing algorithm," in *Proc. of ACM SIGCOMM'89*, pp. 3–12, 1989.
3. A. Parekh and R. Gallager, "A generalized processor sharing approach to flow control in integrated services networks: The single-node case," *IEEE/ACM Transactions on Networking* **1(3)**, pp. 344–357, June 1993.
4. K. Gopalan and T. Chiueh, *Delay Budget Allocation in Delay Bounded Network Paths*, Technical Report TR-113, Experimental Computer Systems Labs, Stony Brook University, Stony Brook, NY, June 2002.
5. J. Kurose, "On computing per-session performance bounds in high-speed multi-hop computer networks," in *Proc. of ACM Sigmetrics'92*, pp. 128–139, 1992.
6. M. Reisslein, K. Ross, and S. Rajagopal, "A framework for guaranteeing statistical qos," *IEEE/ACM Transactions on Networking* **10(1)**, pp. 27–42, February 2002.
7. A. Elwalid and D. Mitra, "Design of generalized processor sharing schedulers which statistically multiplex heterogeneous qos classes," in *Proc. of IEEE INFOCOM'99*, pp. 1220–1230, March 1999.
8. M. Andrews, "Probabilistic end-to-end delay bounds for earliest deadline first scheduling," in *Proc. of IEEE INFOCOM 2000*, March 2000.
9. V. Sivaraman and F. Chiussi, "Providing end-to-end statistical delay guarantees with earliest deadline first scheduling and per-hop traffic shaping," in *Proc. of IEEE INFOCOM 2000*, March 2000.
10. J. Liebeherr, S. Patek, and A. Burchard, *A calculus for end-to-end statistical service guarantees*, Technical Report CS-2001-19, University of Virginia, August 2001.
11. L. Breslau, S. Jamin, and S. Shenker, "Comments on performance of measurement-based admission control algorithms," in *Proc. of IEEE INFOCOM 2000*, March 2000.
12. J. Qiu and E. Knightly, "Measurement-based admission control with aggregate traffic envelopes," *IEEE/ACM Transactions on Networking* **9(2)**, pp. 199–210, April 2001.
13. S. Jamin, P. Danzig, S. Shenker, and L. Zhang, "A measurement-based admission control algorithm for integrated services packet networks," *IEEE/ACM Transactions on Networking* **5(1)**, pp. 56–70, February 1997.
14. S. Floyd, *Comments on measurement-based admission control for controlled load services*, Technical Report, Lawrence Berkeley Laboratory, July 1996.
15. R. Gibbens and F. Kelly, "Measurement-based connection admission control," in *Proc. of 15th Intl. Teletraffic Conference*, June 1997.
16. S. Crosby, I. Leslie, B. McGurk, J. Lewis, R. Russell, and F. Toomey, "Statistical properties of a near-optimal measurement-based admission CAC algorithm," in *Proc. of IEEE ATM'97*, June 1997.
17. J. L. R. Boorstyn, A. Burchard and C. Oottamakorn, "Statistical service assurances for traffic scheduling algorithms," *IEEE Journal on Selected Areas in Communications* **18(13)**, pp. 2651–2664, December 2000.
18. H. Zhang, "Service disciplines for guaranteed performance service in packet-switching networks," *Proc. of IEEE* **83(10)**, pp. 1374–1396, October 1995.
19. L. Georgiadis, R. Guerin, V. Peris, and K. N. Sivarajan, "Efficient network QoS provisioning based on per node traffic shaping," *IEEE/ACM Transactions on Networking* **4(4)**, pp. 482–501, August 1996.