A Measurement-Based Admission Control Algorithm for Integrated Service Packet Networks

Sugih Jamin, *Member, IEEE*, Peter B. Danzig, *Member, IEEE*, Scott J. Shenker, *Senior Member, IEEE*, and Lixia Zhang, *Senior Member, IEEE*

***Abstract*—Many designs for integrated services networks offer a bounded delay packet delivery service to support real-time applications. To provide bounded delay service, networks must use admission control to regulate their load. Previous work on admission control mainly focused on algorithms that compute the worst case theoretical queueing delay to guarantee an absolute delay bound for all packets. In this paper, we describe a *measurement-based* admission control algorithm (ACA) for *predictive* service, which allows occasional delay violations. We have tested our algorithm through simulations on a wide variety of network topologies and driven with various source models, including some that exhibit long-range dependence, both in themselves and in their aggregation. Our simulation results suggest that measurement-based approach combined with the relaxed service commitment of predictive service enables us to achieve a high level of network utilization while still reliably meeting delay bound.**

***Index Terms*—Quality-of-service guarantee, predictive service, real-time traffic.**

I. Bounded Delay Services and Predictive Service

THERE HAVE been many proposals for supporting real-time applications in packet networks by providing some form of bounded delay packet delivery service. When a flow requests real-time service, it must characterize its traffic so that the network can make its admission control decision. Typically, sources are described by either peak and average rates [15] or a filter like a token bucket [33]; these descriptions provide upper bounds on the traffic that can be generated by the source. The traditional real-time service provides a hard or absolute bound on the delay of every packet; in [10] and [15], this service model is called *guaranteed* service. Admission control algorithms (ACA's) for guaranteed service use the *a priori* characterizations of sources to calculate the worst-case behavior of all the existing flows in addition to the incoming one. Network utilization under this model is usually acceptable when flows are smooth; when flows are bursty, however, guaranteed service inevitably results in low utilization [40].

Higher network utilization can be achieved by weakening the reliability of the delay bound. For instance, the *probabilistic* service described in [41] does not provide for the worst-case scenario, instead it guarantees a bound on the rate of lost/late packets based on statistical characterization of traffic. In this approach, each flow is allotted an effective bandwidth that is larger than its average rate but less than its peak rate. In most cases, the equivalent bandwidth is computed based on a statistical model [23], [37] or on a fluid flow approximation [16], [27] of traffic.[1] If one can precisely characterize traffic *a priori*, this approach will increase network utilization. However, we think it will be quite difficult, if not impossible, to provide accurate and tight statistical models for each individual flow. For instance, the average bit rate produced by a given codec in a teleconference will depend on the participant's body movements, which can't possibly be predicted in advance with any degree of accuracy. Therefore, the *a priori* traffic characterizations handed to admission control will inevitably be fairly loose upper bounds.

Many real-time applications, such as *vat, nv*, and *vic*, have recently been developed for packet-switched networks. These applications adapt to actual packet delays and are thus rather tolerant of occasional delay bound violations; they do not need an absolutely reliable bound. For these *tolerant* applications, [10] and [36] proposed *predictive* service, which offers a fairly, but not absolutely, reliable bound on packet delivery times. The ability to occasionally incur delay violations gives admission control a great deal more flexibility, and is the chief advantage of predictive service. The measurement-based admission control approach advocated in [10] and [26] uses the *a priori* source characterizations only for incoming flows (and those very recently admitted); it uses measurements to characterize those flows that have been in place for a reasonable duration. Therefore, network utilization does not suffer significantly if the traffic descriptions are not tight. Because it relies on measurements, and source behavior is

Manuscript received October 1995; revised September 1996; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor J. Kurose. S. Jamin was supported in part by the Uniforum Research Award and by the Office of Naval Research Laboratory under Contract N00173-94-P-1205. At USC, this research was supported by AFOSR Award F49620-93-1-0082, by the National Science Foundation Small-Scale Infrastructure Grant CDA-9216321, and by an equipment loan from Sun Microsystems, Inc. At PARC, this work was supported in part by the Advanced Research Projects Agency, monitored by Fort Huachuca under Contract DABT63-94-C-0073. This paper was presented at the ACM/SIGCOMM'95 Conference on Applications, Technologies, Architectures and Protocols for Computer Communication, August 28–September 1, 1995, Cambridge, MA.

S. Jamin is with the CSE Division, Department of Electrical Engineering and Computer Science, The University of Michigan, Ann Arbor, MI 48109-2122 USA (email: jamin@eecs.umich.edu).

P. B. Danzig is with the Computer Science Department, University of Southern California, Los Angeles, CA 90089-0781 USA.

L. Zhang is with the Computer Science Department, University of California at Los Angeles, Los Angeles, CA 90095 USA.

S. J. Shenker is with the Xerox Palo Alto Research Center, Palo Alto, CA 94304-1314 USA.

Publisher Item Identifier S 1063-6692(97)01637-3.

[1] We refer the interested readers to [24] for a more comprehensive overview and bibliography of ACA's.

not static in general, the measurement-based approach to admission control can never provide the completely reliable delay bounds needed for guaranteed, or even probabilistic, service; thus, measurement-based approaches to admission control can only be used in the context of predictive service and other more relaxed service commitments. Furthermore, when there are only a few flows present, the unpredictability of individual flow's behavior dictates that these measurement-based approaches must be very conservative—by using some worst-case calculation for example. Thus, a measurement-based ACA can deliver significant gain in utilization only when there is a high degree of statistical multiplexing.

In summary, predictive service differs in two important ways from traditional guaranteed service: 1) the service commitment is somewhat less reliable and 2) while sources are characterized by token bucket filters at admission time, the behavior of existing flows is determined by measurement rather than by *a priori* characterizations. It is important to keep these two differences distinct because while the first is commonplace, the second, i.e., the use of *measurement-based* admission control, is more novel. On the reliability of service commitment, we note that the definition of predictive service itself does not specify an acceptable level of delay violations. This is for two reasons. First, it is not particularly meaningful to specify a failure rate to a flow with a short duration [32]. Second, reliably ensuring that the failure rate never exceeds a particular level leads to the same worst-case calculations that predictive service was designed to avoid. Instead, the CSZ approach [10] proposes that the level of reliability be a contractual matter between a network provider and its customers—not something specified on a per-flow basis. We presume that these contracts would only specify the level of violations over some macroscopic time scale (e.g., days or weeks) rather than over a few hundred packet times.[2] In this paper, we describe a measurement-based ACA for predictive service. We demonstrate affirmative answers to the following two questions. First, can one provide reliable delay bounds with a measurement-based ACA? Second, if one does indeed achieve reliable delay bounds, does offering predictive service increase network utilization? Earlier versions of this work have been published in [25] and [26]. Incidentally, the work reported in this paper has been extended in [12] to support advance reservations. Degermark *et al.* [12] have also replicated some of our results on their independently developed network simulator.

The authors of [17] and [20] use measurements to determine admission control, but the admission decisions are precomputed based on the assumption that all sources are exactly described by one of a finite set of source models. This approach is clearly not applicable to a large and heterogeneous application base, and is very different from our approach to admission control that is based on ongoing measurements. In [2] and [37], the authors use measurement to learn the parameters of certain assumed traffic distributions. The authors of [11] and [14] use measurement of existing traffic in their calculation of equivalent bandwidth, providing load, but not delay, bound. In [8] and [19], a neural network is used for dynamic bandwidth allocation. In [29], the authors use precomputed low frequency of flows to renegotiate bandwidth allocation. Hardware implementation of measurement mechanisms are studied in [7] and [38].

II. Measurement-Based Admission Control for ISPN

Our ACA consists of two logically distinct aspects. The first aspect is the set of criteria controlling whether to admit a new flow; these are based on an approximate model of traffic flows and use measured quantities as inputs. The second aspect is the measurement process itself, which we will describe in Section III. In this section, we present the analytical underpinnings of our admission control criteria.

Sources requesting service must characterize the worst-case behavior of their flow. In [10], this characterization is done with a token bucket filter. A token bucket filter for a flow has two parameters: its token generation rate, r, and the depth of its bucket, b. Each token represents a single bit; sending a packet consumes as many tokens as there are bits in the packet. Without loss of generality, in this paper we assume packets are of fixed size and that each token is worth a packet; sending a packet consumes one token. A flow is said to conform to its token bucket filter if no packet arrives when the token bucket is empty. When the flow is idle or transmitting at a lower rate, tokens are accumulated up to b tokens. Thus flows that have been idle for a sufficiently long period of time can dump a whole bucket full of data back to back. Many nonconstant bit rate sources do not naturally conform to a token bucket filter with token rate less than their peak rates. It is conceivable that future real-time applications will have a module that can, over time, learn a suitable r and b to bound their traffic.

We have studied the behavior of our ACA mostly under the CSZ scheduling discipline [10]. Under the CSZ scheduling discipline, a switch can support multiple levels of predictive service, with per-level delay bounds that are order of magnitude different from each other. The ACA at each switch enforces the queueing delay bound at that switch. We leave the satisfaction of end-to-end delay requirements to the end systems. We also assume the existence of a reservation protocol which the end systems could use to communicate their resource requirements to the network.

When admitting a new flow, not only must the ACA decide whether the flow can get the service requested, but it must also decide if admitting the flow will prevent the network from keeping its prior commitments. Let us assume, for the moment, that admission control cannot allow *any* delay violations. Then, the ACA must analyze the worst-case impact of the newly arriving flow on existing flows' queueing delay. However, with bursty sources, where the token bucket parameters are very conservative estimates of the average traffic, delays rarely approach these worst-case bounds. To achieve a fairly reliable bound that is less conservative, we approximate the maximal delay of predictive flows by replacing the worst-case parameters in the analytical models with measured quantities.

[2] A network provider might promise to give its customers their money back if the violations exceed some level over the duration of their flow, no matter how short the flow; however, we contend that the provider cannot realistically assure that excessive violations will never occur.

We call this approximation the *equivalent token bucket filter*. This approximation yields a series of expressions for the expected maximal delay that would result from the admission of a new flow. In CSZ, switches serve guaranteed traffic with the weighted fair queueing scheduling discipline (WFQ) and serve different classes of predictive traffic with priority queueing. Hence, the computation of worst-case queueing delay is different for guaranteed and predictive services. In this section, we will first look at the worst-case delay computation of predictive service, then that of guaranteed service. Following the worst-case delay computations, we present the equivalent token bucket filter. We close this section by presenting details of the ACA based on the equivalent token bucket filter approximations.

A. Worst-Case Delay: Predictive Service

To compute the effect of a new flow on existing predictive traffic, we first need a model for the worst-case delay of priority queues. Cruz, in [9], derived a tight bound for the worst-case delay, D_j^*, of priority queue level j. Our derivation follows Parekh's [34], which is a simpler, but looser, bound for D_j^* that assumes small packet sizes, i.e., the transmission time of each packet is sufficiently small (as compared to other delays) and hence can be ignored. This assumption of small packet sizes further allows us to ignore delays caused by the lack of preemption. Furthermore, we assume that the aggregate rate, aggregated over all traffic classes, is within the link capacity $(\Sigma r_j \leq \mu)$.

Theorem 1–Parekh [34]: The worst-case class j delay, with first in first out (FIFO) discipline within the class and assuming infinite peak rates for the sources, is

$$D_j^* = \frac{\sum b_i}{\mu - \sum_{i=1}^{j-1} r_i} \tag{1}$$

for each class j. Further, this delay is achieved for a strict priority service discipline under which class j has the least priority.[3]

The theorem says that the delay bound for class j is the one-time delay burst that accrues if the aggregate bucket of all classes 1 through j flows are simultaneously dumped into the switch and all classes 1 through $j-1$ sources continue to send at their reserved rates.

We now use (1) as the base equation to model the effect of admitting a new flow α on existing predictive traffic. First we approximate the traffic from all flows belonging to a predictive class j as a single flow conforming to a (ν_j, b_j) token bucket filter. A conservative value for ν_j would be the aggregate reserved rate of all flows belonging to class j. Next, we recognize that there are three instances when the computed worst-case delay of a predictive class can change: 1) when a flow of the same class is admitted, 2) when a flow of a higher priority class is admitted, and 3) when a guaranteed flow is admitted. The switch priority scheduling isolates higher priority $(<k)$ classes from a new flow of class k, so their worst-case delay need not be re-evaluated when admitting a flow of class k. In the remainder of this section, we compute each of the three effects on predictive traffic individually. At the end of these computations, we will observe that admitting a higher priority predictive flow "does more harm" to lower priority predictive traffic than admitting either a guaranteed flow or a predictive flow of the same priority.

In the equations below, we denote newly computed delay bound by $D^{*'}$. We denote the sum of guaranteed flows' reservation by ν_G. The link bandwidth available for serving predictive traffic is the nominal link bandwidth minus those reserved by guaranteed flows: $\mu - \nu_G$.

1) Effect of New Predictive Flow α on Same Priority Traffic: We can model the effect of admitting a new flow α of predictive class k by changing the class's token bucket parameters to $(\nu_k + r_k^\alpha, b_k + b_k^\alpha)$, where (r_k^α, b_k^α) are the token bucket parameters of the new flow

$$\begin{aligned} D_k^{*'} &= \frac{\sum_{i=1}^{k-1} b_i}{\mu - \nu_G - \sum_{i=1}^{k-1} \nu_i} + \frac{b_k + b_k^\alpha}{\mu - \nu_G - \sum_{i=1}^{k-1} \nu_i} \\ &= D_k^* + \frac{b_k^\alpha}{\mu - \nu_G \sum_{i=1}^{k-1} \nu_i}. \end{aligned} \tag{2}$$

We see that the delay of class k grows by a term that is proportional to flow α's bucket size.

2) Effect of Predictive Flow α on Lower Priority Traffic: We compute the new delay bound for class j, where j is greater than the requested class, k, directly from (1), adding in the bucket depth b_k^α and reserved rate r_k^α of flow α

$$\begin{aligned} D_j^{*'} &= \frac{\sum_{i=1}^{k-1} b_i + b_k + b_k^\alpha + \sum_{i=k+1}^{j} b_i}{\mu - \nu_G \sum_{i=1}^{k-1} \nu_i - \nu_k - r_k^\alpha - \sum_{i=k+1}^{j-1} \nu_i} \\ &= D_j^* \frac{\mu - \nu_G - \sum_{i=1}^{j-1} \nu_i}{\mu - \nu_G - \sum_{i=1}^{j-1} \nu_i - r_k^\alpha} + \frac{b_k^\alpha}{\mu - \nu_G - \sum_{i=1}^{j-1} \nu_i - r_k^\alpha}, \\ & k < j \leq K \end{aligned} \tag{3}$$

where K is the number of predictive classes. The first term reflects a *squeezing* of the pipe, in that the additional bandwidth required by the new flow reduces the bandwidth available for lower priority flows. The second term is similar to the delay calculated above, and reflects the effect of the new flow's burstiness.

3) Effect of Guaranteed Flow α on Predictive Traffic: Again, we compute the new delay bound $D^{*'}$ for *all* predictive classes directly from (1), adding in the reserved

[3] For a proof of Theorem 1, we refer interested readers to [34, Theorem 2.4], or [24, Theorem 1].

rate, r_G^α, of flow α

$$D_j^{*'} = \frac{\sum_{i=1}^{j} b_i}{\mu - \nu_G - \sum_{i=1}^{j-1} \nu_i - r_G^\alpha} = D_j^* \frac{\mu - \nu_G - \sum_{i=1}^{j-1} \nu_i}{\mu - \nu_G - \sum_{i=1}^{j-1} \nu_i - r_G^\alpha}, \quad 1 \le j \le K. \tag{4}$$

Notice how the new guaranteed flow simply squeezes the pipe, reducing the available bandwidth for predictive flows; new guaranteed flows do not contribute any delay due to their buckets because the WFQ scheduling algorithm smooths out their bursts. Also observe that the first term of (3) is equivalent to (4): the impact of a new guaranteed flow is like adding a zero-size bucket, higher priority, predictive flow.

Contrasting these three equations, we see that the experienced delay of lower priority predictive traffic increases more when a higher priority predictive flow is admitted than when a guaranteed flow or a same-priority predictive flow is admitted. The WFQ scheduler isolates predictive flows from attempts by guaranteed flows to dump their buckets into the network as bursts. In contrast, lower priority predictive traffic sees both the rates *and* buckets of higher priority predictive flows. A higher priority predictive flow not only squeezes the pipe available to lower priority traffic, but also preempts it.

B. Worst-Case Delay: Guaranteed Service

In [34], the author proves that in a network with arbitrary topology, the WFQ scheduling discipline provides guaranteed delay bounds that depend only on flows' reserved rates and bucket depths. Under WFQ, each guaranteed flow is isolated from the others. This isolation means that, as long as the total reserved rate of guaranteed flows is below the link bandwidth, new guaranteed flows cannot cause existing ones to miss their delay bounds. Hence, when accepting a new guaranteed flow, our ACA only needs to assure that 1) the new flow will not cause predictive flows to miss *their* delay bound [see (4)], and that 2) it will not oversubscribe the link: $\nu_G + r_G^\alpha \le v\mu$, where μ is the link bandwidth and v is the utilization target (see Section III-B for a discussion on utilization target). In addition to protecting guaranteed flows from each other, WFQ also isolates (protects) guaranteed flows from all predictive traffic.

C. Equivalent Token Bucket Filter

The equations above describe the aggregate traffic of each predictive class with a single token bucket filter. How do we determine a class's token bucket parameters? A completely conservative approach would be to make them the sum of the parameters of all the constituent flows; when data sources are bursty and flows declare conservative parameters that cover their worst-case bursts, using the sum of declared parameters will result in low link utilization. Our algorithm is approximate and optimistic: we take advantage of statistical multiplexing by using measured values, instead of providing for the worst possible case, to gain higher utilization, risking that some packets may occasionally miss their delay bounds. In essence, we describe existing aggregate traffic of each predictive class with an *equivalent token bucket filter* with parameters determined from traffic measurement.

The equations above can be equally described in terms of current delays and usage rates as in bucket depths and usage rates. Since it is easier to measure delays than to measure bucket depths, we do the former. Thus, the measured values for a predictive class j are the aggregate bandwidth utilization of the class, $\hat{\nu}_j$, and the experienced packet queueing delay for that class, $\hat{D}_j$. For guaranteed service, we count the sum of all reserved rates, ν_G, and we measure the actual bandwidth utilization, $\hat{\nu}_G$, of all guaranteed flows. Our approximation is based on substituting, in the above equations, the measured rates $\hat{\nu}_j$ and $\hat{\nu}_G$ for the reserved rates, and substituting the measured delays $\hat{D}_j, j = 1, \cdots, K$ for the maximal delays. We now use the previous computations and these measured values to formulate an ACA.

D. The Admission Control Algorithm

1) New Predictive Flow: If an incoming flow α requests service at predictive class k, the ACA:

1) denies the request if the sum of the flow's requested rate, r_k^α, and current usage would exceed the targeted link utilization level

$$v\mu > r_k^\alpha + \hat{\nu}_G + \sum_{i=1}^{K} \hat{\nu}_i \tag{5}$$

2) denies the request if admitting the new flow could violate the delay bound, D_k, of the same priority level

$$D_k > \hat{D}_k + \frac{b_k^\alpha}{\mu - \hat{\nu}_G \sum_{i=1}^{k-1} \hat{\nu}_i} \tag{6}$$

or could cause violation of lower priority classes' delay bound, D_j

$$D_j > \hat{D}_j \frac{\mu - \hat{\nu}_G - \sum_{i=1}^{j-1} \hat{\nu}_i}{\mu - \hat{\nu}_G - \sum_{i=1}^{j-1} \hat{\nu}_i - r_k^\alpha} + \frac{b_k^\alpha}{\mu - \hat{\nu}_G - \sum_{i=1}^{j-1} \hat{\nu}_i - r_k^\alpha}, \quad k < j \le K. \tag{7}$$

2) New Guaranteed Flow: If an incoming flow α requests guaranteed service, the ACA:

1) denies the request if either the bandwidth check in (5) fails or if the reserved bandwidth of all guaranteed flows exceeds the targeted link utilization level

$$v\mu > r_G^\alpha + \nu_G; \tag{8}$$

2) denies the request if the delay bounds of predictive classes can be violated when the bandwidth available for predictive service is decreased by the new request

$$D_j > \hat{D}_j \frac{\mu - \hat{\nu}_G - \sum_{i=1}^{j-1} \hat{\nu}_i}{\mu - \hat{\nu}_G - \sum_{i=1}^{j-1} \hat{\nu}_i - r_G^{\alpha}}, \qquad 1 \leq j \leq K. \qquad (9)$$

If the request satisfies all of these inequalities, the new flow is admitted.

III. A Simple Time-Window Measurement Mechanism

The formulas described in the previous section rely on the measured values $\hat{D}_j, \hat{\nu}_G$, and $\hat{\nu}_j$ as inputs. We describe in this section the time-window measurement mechanism we use to measure these quantities. While we believe our admission control equations to have some fundamental principles underlying them, we make no such claim for the measurement process. Our measurement process uses the constants λ, S, and T; discussion of their roles as performance tuning knobs follows our description of the measurement process.

A. Measurement Process

We take two measurements: experienced delay and utilization. To estimate delays, we measure the queueing delay $\hat{d}$ of every packet. To estimate utilization, we sample the usage rate of guaranteed service, $\hat{\nu}_G^S$, and of each predictive class $j, \hat{\nu}_j^S$, over a sampling period of length S packet transmission units. In the following, we describe how these measurements are used to compute the estimated maximal delay $\hat{D}_j$ and the estimated utilization $\hat{\nu}_G$ and $\hat{\nu}_j$.

1) Measuring Delay: The measurement variable $\hat{D}_j$ tracks the estimated maximum queueing delay for class j. We use a measurement window of T packet transmission units as our basic measurement block. The value of $\hat{D}_j$ is updated on three occasions. At the end of the measurement block, we update $\hat{D}_j$ to reflect the maximal packet delay seen in the previous block. Whenever an individual delay measurement exceeds this estimated maximum queueing delay, we know our estimate is wrong and immediately update $\hat{D}_j$ to be λ times this sampled delay. The parameter λ allows us to be more conservative by increasing $\hat{D}_j$ to a value higher than the actual sampled delay. Finally, we update $\hat{D}_j$ whenever a new flow is admitted, to the value of projected delay from our admission control equations. Algebraically, the updating of $\widehat{D_j}$ is as follows:

$$\hat{D}_j' = \begin{cases} \text{MAX}\,(\hat{d}), & \text{of past T measurement window} \\ \lambda \hat{d}, & \text{if } \hat{d} > \hat{D}_j \\ \text{Right side of} & \text{when adding a new flow} \\ \text{eq. (6), (7), or (9),} & \text{depending on the service and} \\ & \text{class requested by the flow.} \end{cases} \qquad (10)$$

2) Measuring Rate: The measurement variables $\hat{\nu}_G$ and $\hat{\nu}_j$ track the highest sampled aggregate rate of guaranteed flows and each predictive class respectively (heretofore, we will use "$\hat{\nu}$" as a shorthand for "$\hat{\nu}_G$ and/or $\hat{\nu}_j$," and "$\hat{\nu}^S$" for "$\hat{\nu}_G^S$ and/or $\hat{\nu}_j^S$.") The value of $\hat{\nu}$ is updated on three occasions. At the end of the measurement block, we update $\hat{\nu}$ to reflect the maximal sampled utilization seen in the previous block. Whenever an individual utilization measurement exceeds $\hat{\nu}$, we immediately update $\hat{\nu}$ with the new sampled value. Finally, we update $\hat{\nu}$ whenever a new flow is admitted. Algebraically, the updating of $\hat{\nu}$ is as follows:

$$\hat{\nu}' = \begin{cases} \text{MAX}\,(\hat{\nu}^S), & \text{of past } T \text{ measurement window} \\ \hat{\nu}^S, & \text{if } \hat{\nu}^S > \hat{\nu}, \text{ where } \hat{\nu}^S \text{ is the average} \\ & \text{rate over } S \text{ averaging period} \\ \hat{\nu} + r^{\alpha}, & \text{when adding a new flow } \alpha. \end{cases} \qquad (11)$$

The measured rate of guaranteed traffic is capped at the sum of guaranteed reserved rate ($\hat{\nu}_G' = \text{MIN}\,(\hat{\nu}_G, \nu_G)$).

When a flow leaves the network, we do not explicitly adjust the measured values; instead we allow the measurement mechanism to adapt to the observed traffic automatically. We do, however, subtract the reserved rate of a departing guaranteed flow from the sum of all guaranteed reserved rate, ν_G.

B. Performance Tuning Knobs

We now look at the constants used in the algorithm.

1) v: In a simple $M/M/1$ queue, the variance in delay diverges as the system approaches full utilization. A measurement-based approach is doomed to fail when delay variations are exceedingly large, which will occur at very high utilization. It is thus necessary to identify a *utilization target* and require that the ACA strive to keep link utilization below this level.

The appropriate utilization target of any given link depends on the characteristics of the traffic flowing through it. If each source's rate is small compared to link capacity (small grain size) and bursts are short, the link's utilization target can be set higher. Bursty sources with big, long bursts or long idle periods will require a lower link utilization target. In this paper, we set utilization target at 90% capacity.

2) λ: In our simulations, a single instance of packet delay above the current estimate typically indicates that subsequent delays are likely to be even larger; so when a packet's queueing delay, $\hat{d}$, is higher than its class's estimated maximal delay $\widehat{D_j}$, we back off our delay estimate to a much larger value, $\lambda \hat{d}$. In this paper, we use $\lambda = 2$.

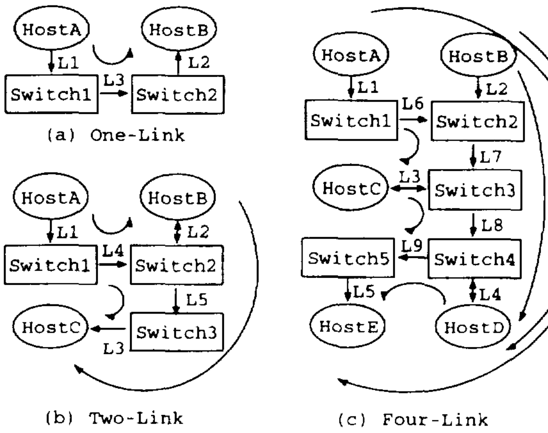
3) S: The averaging period S in (11) controls the sensitivity of our rate measurement. The smaller the averaging period, the more sensitive we are to bursts; the larger the averaging period, the smoother traffic appears. An S that captures individual bursts may make the admission control more conservative than desired. In this paper we use S of at least 100 packet transmission times.

4) T: Once $\hat{D}$ or $\hat{\nu}$ is increased, their values stay high until the end of their respective measurement window T. The size of T controls the adaptability of our measurement mechanism to drops in traffic load. Smaller T means more adaptability, but larger T results in greater stability. The window size for utilization measurement should allow for enough utilization samples, i.e., T should be several times S. The measurement windows of the load and the delay can be maintained independently. When we admit a new flow and add its worst case effect to the measured values, we also restart the measurement windows to give the measurement mechanism one whole window to gather information on the new flow.

Of the four performance knobs, v, λ, S, and T, tuning T provides the most pronounced effect on experienced delay and link utilization. Varying T has two related effects on the ACA. First, since T is the length of the measurement block used to determine how long we keep the previous maximal packet delay and sampled utilization, increasing T makes these estimates more conservative, which in turn makes the ACA itself more conservative. Thus, larger T means fewer delay violations and lower link utilization. Second, T also controls how long we continue to use our calculated estimate of the delay and utilization induced by a newly admitted flow. Recall that whenever a new flow is admitted, we artificially increase the measured values to reflect the worst-case expectations, and then restart the measurement window. Thus, we are using the calculated effects of new flows rather than the measured effects until we survive an entire T period without any new flow arrival. This means that if $\bar{r}$ is the average flow reservation rate, and μ the link bandwidth (and assuming $v = 1$ for convenience), we will admit at most $\mu/\bar{r}$ number of flows and then not admit anymore flow until the end of a T period. During its lifetime, L, a flow will see approximately $A = \mu/\bar{r}$ number of flows admitted every T period. Thus at the end of its average lifetime, $\bar{L}$, an average flow would have seen approximately $F = A * \bar{L}/T$ number of flows. If the average rate of an average flow is $\hat{\bar{r}}$, ideally we want $F * \hat{\bar{r}}$, a link's stable utilization level, to be near μ. However, flows also depart from the network. The expected number of admitted flow departures during the period T depends on the number of flows and their duration. If this number of departures is significant, a flow will see a much smaller number of flows during its lifetime, i.e., the stable $F * \hat{\bar{r}}$ becomes *much* smaller than μ. For the same average reservation rate, $\bar{r}$, and a given T, the size of the stable F is determined by the average flow duration, $\bar{L}$. A shorter average flow duration means more departure per T. In the long run, we aim for $F * \hat{\bar{r}} \approx \mu$, or equivalently, $\bar{L}/T \approx \bar{r}/\hat{\bar{r}}$. If all flows use exactly what they reserved, we have $\bar{L}/T = 1$, meaning that we should not try to give away the flows' reservations. We present further illustrative simulation results on the importance of the $\bar{L}/T$ ratio in Section IV-E. Note that when T is infinite, we only use our computed values, which are conservative bounds, and ignore the measurements entirely. That is, we will never suffer any delay violations at a given hop if we use an infinite value for T. Thus, the parameter T always provides us with a region of reliability.

IV. Simulations

Admission control algorithms for guaranteed service can be verified by formal proof. Measurement-based ACAs can only be verified through experiments on either real networks or a simulator. We have tested our algorithm through simulations on a wide variety of network topologies and driven with various source models; we describe a few of these simulations in this paper. In each case, we were able to achieve a reasonable degree of utilization (when compared to guaranteed service) and a low delay bound violation rate (we try to be very conservative here and always aim for *no* delay bound violation over the course of all our simulations). Before we present the results from our simulations, we first present the topologies and source models used in these simulations.



(a) One-Link

(b) Two-Link

(c) Four-Link

Fig. 1. The one-link, two-link, and four-link topologies.

A. Simulated Topologies

For this paper, we ran our simulations on four topologies: the One-Link, Two-Link, Four-Link, and TBone topologies depicted in Fig. 1(a)–(c) and Fig. 2, respectively. In the first three topologies, each host is connected to a switch by an infinite bandwidth link. The connection between switches in these three topologies are all 10 Mb/s links, with infinite buffers. In the One-Link topology, traffic flows from HostA to HostB. In the Two-Link case, traffic flows between three host pairs (in source–destination order): HostA–HostB, HostB–HostC, HostA–HostC. Flows are assigned to one of these three host pairs with uniform probability. In the Four-Link topologies, traffic flows between six host pairs: HostA–HostC, HostB–HostD, HostC–HostE, HostA–HostD, HostB–HostE, HostD–HostE; again, flows are distributed among the six host pairs with uniform probability. In Fig. 1, these host pairs and the paths their packets traverse are indicated by the directed curve lines.

The TBone topology consists of 10, 45, and 100 Mb/s links, as depicted in Fig. 2(a). Traffic flows between 45 host-pairs following four major "currents" as shown in Fig. 2(b): the numbers 1, 2, 3, 4 next to each directed edge in the figure denote the "current" present on that edge. The 45 host-pairs are listed in Table I. Flows between these host-pairs ride on only one current, for example flows from host H1 to H26 rides on current 4. In Fig. 2(a), a checkered box on a switch indicates that we have instrumented the switch to study traffic flowing out of that switch onto the link adjacent to the checkered box.

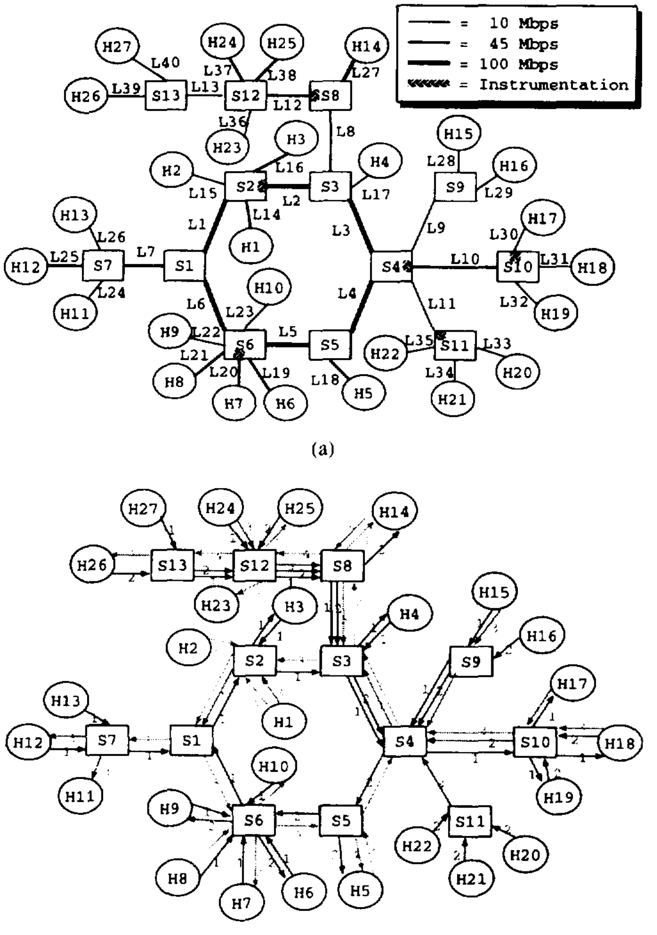


Fig. 2. The TBONE topology. (a) TBONE topology. (b) Four traffic "currents" on TBONE.

TABLE I
FORTY-FIVE HOST PAIRS ON TBONE

Source	Destination(s)	Source	Destination(s)
H1	H1 H5, H7, H11, H12, H14, and H26	H14	H23 and H25
H2	H10 and H25	H15	H11 and H17
H3	H4 and H19	H16	H5 and H9
H4	H18	H17	H12
H5	H14 and H25	H19	H5
H6	H18	H20	H5
H7	H17	H21	H9
H8	H4, H5, H26	H22	H6
H9	H3 and H19	H24	H12 and H17
H10	H3 and H18	H25	H6 and H14
H12	H4	H26	H9 and H14
H13	H17	H27	H4

B. Source Models

We currently use three kinds of source model in our simulations. All of them are ON/OFF processes. They differ in the distribution of their ON time and call holding time (CHT, which we will also call "flow duration" or "flow lifetime"). One of these is the two-state Markov process used widely in the literature. Recent studies ([6], [13], [18], [28], [30], [35]) have shown that network traffic often exhibits long-range dependence (LRD), with the implications that congested periods can be quite long and a slight increase in number of active connections can result in large increase in packet loss rate [35]. Paxson and Floyd [35] further called attention to the possibly damaging effect long-range dependent traffic might have on measurement-based ACA's. To investigate this and other LRD-related questions, we augmented our simulation study with two LRD source models.

1) EXP Model: Our first model is an ON/OFF model with exponentially distributed ON and OFF times. During each ON period, an exponentially distributed random number of packets, with average N, are generated at fixed rate p packet/s. Let I ms be the average of the exponentially distributed OFF times, then the average packet generation rate a is given by $1/a = I/N + 1/p$. The EXP1 model described in the next section is a model for packetized voice encoded using ADPCM at 32 Kb/s.

2) LRD—Pareto-ON/OFF: Our next model is an ON/OFF process with Pareto distributed ON and OFF times (for ease of reference, we call this the *Pareto*-ON/OFF model). During each ON period, a Pareto distributed number of packets, with mean N and Pareto shape parameter β, are generated at peak rate p packet/s. The OFF times are also Pareto distributed with mean I ms and shape parameter γ. Pareto shape parameter less than 1 gives data with infinite mean; shape parameter less than two results in data with infinite variance. The Pareto location parameter is $mean * (shape - 1)/shape$. Each *Pareto*-ON/OFF source by itself does not generate LRD series. However, the aggregation of them does [39].

3) LRD—Fractional ARIMA: We use each number generated by the *fractional autoregressive integrated moving average* (fARIMA) process [22] as the number of fixed-size packets to be sent back to back in each ON period. Interarrivals of ON periods are of fixed length. For practical programming reasons, we generate a series of 15 000 fARIMA data points at the beginning of each simulation. Each fARIMA source then picks a uniformly distributed number between 1 and 15 000 to be used as its index into that series. On reaching the end of the series, the source wraps around to the beginning. This method is similar to the one used by the authors of [18] to simulate data from several sources using one variable bit rate (VBR) video trace.

The fractional ARIMA model generates long-range dependent series. However, the marginal distribution of fARIMA generated series is Gaussian, whereas VBR video traces exhibit low average with high peaks; thus we can *not* use the fARIMA output to model traffic from a single VBR video source. Nevertheless, simulation results in [18] indicated that aggregation of fARIMA generated series may well model aggregate VBR video traffic—such as that coming from a subnetwork. The fARIMA model takes three parameters: the autoregressive process order with the corresponding set of weights, the degree of integration, and the moving average process order with the corresponding set of weights, it also requires an innovation with a Gaussian marginal distribution (see, e.g., [4], [21]). We first generate a normally distributed innovation with mean N and standard deviation s packets. If the minimum of the fARIMA output is less than zero, we shift the whole series by

TABLE II
SIX INSTANCES OF THE THREE SOURCE MODELS

Model Name	Model's Parameters				Token Bucket Parameters				Bound (ms)	
	p pkt/s	I ms	N pkts	p/a	r tkn/s	b tkns	cut rate	max qlen	D^*	D_f
EXP1	64	325	20	2	64	1	0	0	16	16
EXP2	1024	90	10	10	320	50	2.1e-3	17	160	160
EXP3	∞	684	9	∞	512	80	9.4e-5	1	160	160
				β						
POO1	64	2925	20	1.2	64	1	0	0	16	16
POO2	256	360	10	1.9	240	60	4.5e-5	220	256	160
				s						
fARIMA ({0.75}, 0.15, -)	∞	125	8	13	1024	100	1.1e-2	34	100	160

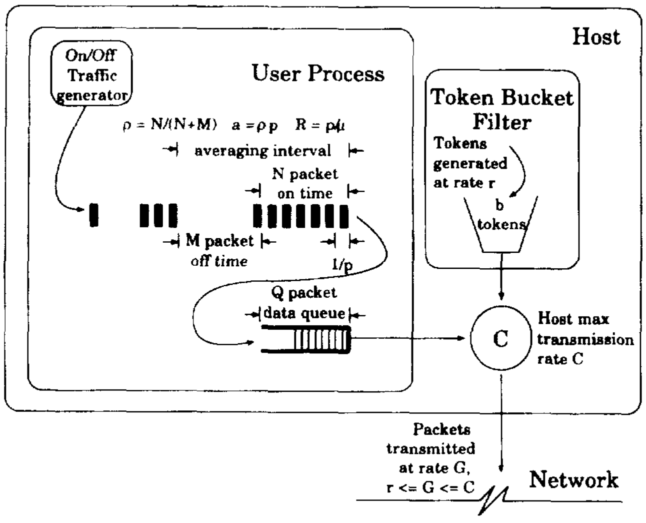


Fig. 3. ON/OFF traffic model with token-bucket filter.

adding the absolute value of its minimum to every number in the series. This way of obtaining nonnegative series is also used in [1]. Note that this shifting process constrains the maximum value of the generated series to be always twice its average. The Whittle maximum likelihood estimator [3] confirms that our shifting, cropping, and overlaying of the fARIMA generated series does not destroy its long-range dependence.

Fig. 3 shows a packet-arrival depiction of an ON/OFF source in the context of a host with token-bucket filter. To make a given traffic generation source conform to a particular token bucket filter, a host can queue packets arriving at an empty bucket until more tokens are available. If the data queue length (B) is zero, packets that arrive at an empty token bucket are immediately dropped.

In addition to each source's burstiness, network traffic dynamics is also effected by the arrival pattern and duration of flows. Our simulator allows us to drive each simulation with a number of flow generators; for each generator, we can specify its start and stop times, the average flow interarrival time, the maximum number of concurrently active flows, and the mix of transport protocol, source model, token bucket filter, and service request ascribed to each flow. We ascribe exponentially distributed lifetimes to the EXP model, following [31]. The duration of for LRD sources, however, are taken from a lognormal distribution, following [5] and [13]. The interarrival times of all flows are exponentially distributed [35].

C. Parameter Choices

We chose six instantiations of the above three source models, as summarized in Table II. In the table, $p = \infty$ means that after each OFF time, packets for the next ON period are transmitted back to back. (On real networks, packets are sent back to back when the applications generate traffic faster than the network can transmit it.) In the same table, we also list the settings of the token bucket parameters assigned to each source. Column 8, labeled *cut rate*, indicates the average number of packets that would have been dropped by each flow's token bucket filter *over* the total number of packets sent by the flow, had the data queue length been zero (i.e., packets are immediately dropped upon arriving at an empty token bucket). Column 9, labeled *max qlen*, shows the maximum data queue length a flow can expect to see if the data queue has infinite length. We assign each flow a data queue with infinite length in all our simulations (i.e., packets that arrive at an empty token bucket are always queued, and the queue never overflows). Recall that in this paper we use fixed packet size and each of our token is worth 1 Kb of data, which is also our packet size.

The shape parameter of the Pareto distributed ON time (β) of the Pareto-ON/OFF sources are selected following the observations in [39]. According to the same reference, the shape parameter of the Pareto distributed OFF time (γ) stays mostly below 1.5; in this paper, we use γ of 1.1 for all POO sources. For the POO1 model, we use a token bucket rate equals to the source's peak rate such that the token bucket filter does not reshape the traffic. For the POO2 model, some of the generated packets were queued; this means during some of the source's alleged "OFF" times, it may actually still be draining its data queue onto the network. Thus for the POO2 model, the traffic seen on the wire may not be Pareto-ON/OFF.

When a flow with token bucket parameters (r, b) requests guaranteed service, the maximal queueing delay (ignoring terms proportional to a single packet time) is given by b/r [34]. Column 10, labeled D^*, lists the guaranteed delay bound for each source given its assigned token bucket filter. Column 11, labeled D_j, lists the predictive delay bound assigned to each source. We simulate only two classes of predictive service. A predictive bound of 16 ms means first class predictive service, 160-ms second class. We have chosen the token bucket parameters so that, in most cases, the delay bounds given to a flow by predictive and guaranteed services are the same. This facilitates comparison between the utilization levels achieved with predictive and guaranteed services. In the few cases where the delays are not the same, such as in the POO2 and fARIMA cases, the utilization comparison is less meaningful. In the POO2 case, for example, the predictive delay bound is smaller than the guaranteed bound, so the utilization gain we find here understates the true gain.

For the fARIMA source, we use an autoregressive process of order 1 (with weight 0.75) and degree of integration 0.15 (resulting in a generated series with Hurst parameter 0.65). The first order autoregressive process with weight 0.75 means our fARIMA traffic also has strong short-range dependence, while maintaining stationarity [4, p. 53]. The interarrival time between ON periods is 1/8th of a second. The Gaussian innovation fed to the fARIMA process has a mean of 8 packets with standard deviation 13.

Except for simulations on the TBONE topology, flow interarrival times are exponentially distributed with an average of 400 ms. Because of system memory limitation, we set the average flow interarrivals of simulations on the TBONE topology to 5 s. The average holding time of all EXP sources is 300 s. The POO and fARIMA sources have lognormal distributed holding times with median 300 s and shape parameter 2.5.

We ran most of our simulations for 3000 s simulated time. The data presented are obtained from the later half of each simulation. By visual inspection, we determined that 1500 simulated seconds is sufficient time for the simulation to warm up. However, simulations with long-range dependent sources requesting predictive service requires a longer warmup period. We ran all simulation involving such sources for 5.5 h simulation time, with reported data taken from the later 2.7 h.

We divide the remainder of this section up into three subsections. First, we show that predictive service indeed yields higher level of link utilization than guaranteed service does. We provide supporting evidence from results of simulations with both homogeneous and heterogeneous traffic sources, on both single-hop and multihop networks. Depending on traffic burstiness, the utilization gain ranges from twice to order of magnitude. This is the basic conclusion of this paper.

Second, we provide some simulation results to illustrate the effect of the $\overline{L}/T$ ratio on network performance, as discussed in Section III-B. We show that a larger $\overline{L}/T$ ratio yields higher utilization but less reliable delay bound, while a smaller one provides more stable delay estimate at lower utilization. We also present a few sample path snapshots illustrating the effect of T.

Finally, we close this section with a discussion of some general allocation properties of ACA's when flows are not equivalent; we believe these properties to be inherent in all ACA's whose only admission criterion is to avoid service commitment violations.

TABLE III
SINGLE-HOP HOMOGENEOUS SOURCES SIMULATION RESULTS

Model	Guaranteed		Predictive			
Name	%Util	#Actv	%Util	#Actv	$\lceil d_j \rceil$	$\overline{L}/T$
EXP1	46	144	80	250	3	60
EXP2	28	28	76	75	42	300
EXP3	2	18	62	466	33	600
POO1	7	144	74	1637	5	60
POO2	3	38	64	951	8	60
fARIMA	55	9	81	13	72	60

D. On the Viability of Predictive Service

We considered six different source models, four different network topologies (one single hop and three multihop), and several different traffic mixes. In particular, some of our traffic loads consisted of identical source models requesting the same service (the homogeneous case), and others had either different source models and/or different levels of service (the heterogeneous case). The organization of our presentation in this section is: 1) homogeneous sources, single hop, 2) homogeneous sources, multihop, 3) heterogeneous sources, single hop, and 4) heterogeneous sources, multihop.

1) Homogeneous Sources—The Single-Hop Case: By homogeneous sources, we mean sources that not only employ just one kind of traffic model, but also ask for only one kind of service. For this and all subsequent single-hop simulations, we use the topology depicted in Fig. 1(a). For each source, we ran two kinds of simulation. The first has all sources requesting guaranteed service. The second has all sources requesting predictive service. The results of the simulations are shown in Table III. The column labeled "%Util" contains the link utilization of the bottleneck link, L3. The "#Actv" column contains a snapshot of the average number of active flows concurrently running on that bottleneck link. The "$\lceil d_j \rceil$" column contains the maximum experienced delay of predictive class j packets. The "$\overline{L}/T$" column lists the ratio of average flow duration to measurement window used with each source model.

We repeated the predictive service simulations nine times, each time with a different random seed, to obtain confidence intervals. We found the confidence interval for the all the numbers to be very tight. For example, the utilization level of POO1 sources under predictive service has a 99% confidence interval of (74.01, 74.19); the 99% confidence interval for the maximum experience delay is (4.41, 4.84) (the number reported in the table is the ceiling of the observed maximum).

As mentioned in Section IV-B, we consider the performance of our ACA "good" if there is *no* delay bound violation during a simulation run. Even with this very restrictive requirement, one can see from Table III that predictive service consistently allows the network to achieve higher level of utilization than guaranteed service does. The utilization gain is not large when sources are smooth. For instance, the source model EXP1 has a peak rate that is only twice its average rate. Consequently, the data only shows an increase in utilization from 46% to 80%. (One can argue that the theoretical upper bound in the utilization increase is the peak to average ratio.) In contrast, bursty sources allow predictive service to achieve several orders of magnitude higher utilization compared to that achievable under guaranteed service. Source model EXP3, for example, is a very bursty source; it has an infinite peak rate (i.e. sends out packets back to back) and has a token bucket of size 80. The EXP3 flows request reservations of 512 Kb/s, corresponding to the token bucket rate at the sources. Under guaranteed service, only 18 flows can be admitted to the 10 Mb/s bottleneck link (with 90% utilization target). The actual link utilization is only 2%.[4] Under predictive service, 466 flows are served on the average, resulting in actual link utilization of 62%.

In this homogeneous scenario with only one class of predictive service and constantly oversubscribed link, our measurement-based ACA easily adapts to LRD traffic between the coming and going of flows. The utilization increased from 7% to 74% and from 3% to 64% for the POO1 and POO2 sources, respectively. The utilization gain for the fARIMA sources was more modest, from 55% to 81%. This is most probably because the source's maximum ON time is at most twice its average (an artifact of the shifting we do, as discussed in Section IV-B, to obtain nonnegative values from the fARIMA generated series). In all cases, we were able to achieve high levels of utilization without incurring delay violations. To further test the effect of long OFF times on our measurement-based algorithm, we simulated POO1 sources with infinite duration. With utilization target of 90% link capacity, we did see a rather high percentage of packets missing their delay bound. Lowering the utilization target to 70%, however, provided us enough room to accommodate traffic bursts. Thus for these scenarios, we see no reason to conclude that LRD traffic poses special challenges to our measurement-based approach.

2) Homogeneous Sources—The Multihop Case: Next, we ran simulations on multihop topologies depicted in Fig. 1(b) and 1(c). The top half of Table IV shows results from simulations on the TWO-LINK topology. The utilization numbers are those of the two links connecting the switches in the topology. The source models employed here are the EXP1, EXP3, and POO2 models, one per simulation. The bottom half of Table IV shows the results from simulating source models EXP2, POO1, and fARIMA on the FOUR-LINK topology. For each source model, we again ran one simulation where all sources request guaranteed service, and another one where all sources request *one* class of predictive service.

The most important result to note is that, once again, predictive service yielded reasonable levels of utilization without incurring any delay violations. The utilization levels, and the utilization gains compared to guaranteed service, are roughly comparable to those achieved in the single hop case.

TABLE IV
MULTIHOP HOMOGENEOUS SOURCES LINK UTILIZATION

Topology	Link name	Model name	Guaranteed	Predictive	
			%Util	%Util	$\lceil d_j \rceil$
TWO-LINK	L4	EXP1	45	67	2
		EXP3	2	44	20
		POO2	3	59	7
	L5	EXP1	46	78	3
		EXP3	2	58	30
		POO2	3	70	17
FOUR-LINK	L6	EXP2	17	42	6
		POO1	4	31	1
		fARIMA	38	54	36
	L7	EXP2	28	71	31
		POO1	7	66	2
		fARIMA	55	77	40
	L8	EXP2	28	72	24
		POO1	8	75	7
		fARIMA	53	74	29
	L9	EXP2	28	71	31
		POO1	8	59	2
		fARIMA	53	80	44

TABLE V
SINGLE-HOP, SINGLE SOURCE MODEL, MULTIPLE PREDICTIVE SERVICES LINK UTILIZATION

Model	PP	GP	GPP
EXP1	77	77	—
EXP2	71	70	—
EXP3	31	31	—
POO1	70	69	69
POO2	60	57	—
fARIMA	79	79	78

3) Heterogeneous Sources—The Single-hop Case: We now look at simulations with heterogeneous sources. For each of the simulation, we used two of our six source model instantiations. Each source was given the same token bucket as listed in Table II and, when requesting predictive service, requests the same delay bound as listed in the said table. We ran three kinds of simulation with heterogeneous sources: 1) single source model requesting multiple levels of predictive service, 2) multiple source models requesting a single class of predictive service, and 3) multiple source models requesting multiple levels of predictive service. In all cases, we compared the achieved utilization with those achieved under guaranteed service. For the first and third cases, we also experimented with sources that request both guaranteed and predictive services. When multiple source and/or service models were involved, each model was given an equal probability of being assigned to the next new flow. In all these simulations, the experienced delays were all within their respective bounds.

Table V shows the utilization achieved when flows with the same source model requested: two classes of predictive service (PP), guaranteed and one predictive class (GP), and guaranteed and two predictive classes (GPP). In the GP case, flows request the predictive class "assigned" to the source model under study (see Table II). In the other cases, both predictive classes, of bounds 16 and 160 ms were requested. Compare the numbers in each column of Table V with those in the "%Util" column of Table III under guaranteed service. The presence of predictive traffic invariably increases network utilization.

Next, we look at the simulation results of multiple source models requesting a single service model. Table VI shows the utilization achieved for selected pairings of the models. The column headings name the source model pairs. The first row shows the utilization achieved with guaranteed service, the second predictive service. We let the numbers speak for themselves.

[4] Parameter-based ACA's may not need to set a utilization target and thus can achieve a somewhat higher utilization; for the scenario simulated here, two more guaranteed flows could have been admitted.

TABLE VI
SINGLE-HOP, MULTIPLE SOURCE MODELS, SINGLE SERVICE LINK UTILIZATION

Service	EXP1–POO1	EXP2–EXP3	EXP2–POO2	EXP2–fARIMA	EXP3–fARIMA	POO2–fARIMA
Guaranteed	15	21	5	38	18	32
Predictive	75	70	63	79	81	69

TABLE VII
SINGLE-HOP, MULTIPLE SOURCE MODELS,
MULTIPLE PREDICTIVE SERVICES LINK UTILIZATION

Service	EXP1–EXP2	EXP1–fARIMA	EXP1–POO2	EXP2–POO1	EXP3–POO1	POO1–fARIMA
Guaranteed	43	50	29	10	7	23
Guar./Pred.	73	74	65	61	51	65
Predictive	75	78	65	62	60	65

TABLE VIII
SINGLE- AND MULTIHOP, ALL SOURCE MODELS, ALL SERVICES LINK UTILIZATION

Topology name	Link name	Guaranteed %Util	Guaranteed and Predictive		
			%Util	$\lceil d_1 \rceil$	$\lceil d_2 \rceil$
ONE-LINK	L3	24	66	3.	45.
TWO-LINK	L4	15	72	2.	54.
	L5	21	72	2.	41.
FOUR-LINK	L6	19	47	1.	36.
	L7	24	70	2.	46.
	L8	20	72	2.	49.
	L9	18	75	1.	53.
TBONE	L2	9	14	0.02	0.15
	L10	17	31	0.15	5.35
	L11	27	32	0.37	21.9
	L12	22	23	0.1	5.84
	L20	8	21	0.22	16.6
	L30	32	52	0.49	34.7

TABLE IX
EFFECT OF T AND $\overline{L}$

T	%Util	$\lceil d_j \rceil$
1e4	82	25
5e4	81	22
1e5	77	15
2e5	75	13
5e5	68	5

(a)

$\overline{L}$	T			
	1e4		1e5	
	%Util	$\lceil d_j \rceil$	%Util	$\lceil d_j \rceil$
3000	86	48	82	24
900	84	32	80	16
300	82	25	77	15
100	81	21	76	11
30	78	15	69	7

(b)

Finally, in Table VII we show utilization numbers for flows with multiple source models requesting multiple service models. The first row shows the utilization achieved when all flows asked only for guaranteed service. The second row shows the utilization when half of the flows requests guaranteed service and the other half requests the predictive service suitable for its characteristics (see Table II). And the last row shows the utilization achieved when each source requests a predictive service suitable for its characteristics.

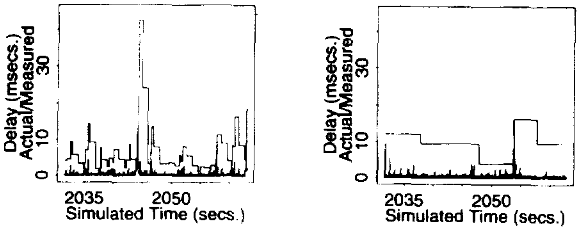
4) Heterogeneous Sources—The Multihop Case: We next ran simulations with all six source models on all our topologies. In Table VIII we show the utilization level of the bottleneck links of the different topologies. Again, contrast the utilization achieved under guaranteed service alone with those under both guaranteed and predictive services. The observed low predictive service utilization on link L6 is not due to any constraint enforced by its *own* admission decisions, but rather is due to lack of traffic flows caused by rejection of multihop flows by later hops, as we will explain in Section IV-F. Utilization gains on the TBONE topology are not so pronounced as on the other topologies. This is partly because we are limited by our simulation resources and cannot drive the simulations with higher offered load. Recall that flow interarrivals on simulations using the TBONE topology have an average of 5 s, which is an order of magnitude larger than the 400 ms used on the other topologies.

Our results so far indicate that a measurement-based ACA can provide reasonable reliability at significant utilization gains. These conclusions appear to hold not just for single hop topologies and smooth traffic sources, but also for multihop configurations and long-range dependent traffic as we have tested. We cannot, within reasonable time, verify our approach in an exhaustive and comprehensive way, but our simulation results are encouraging.

E. On the Appropriate Value of T

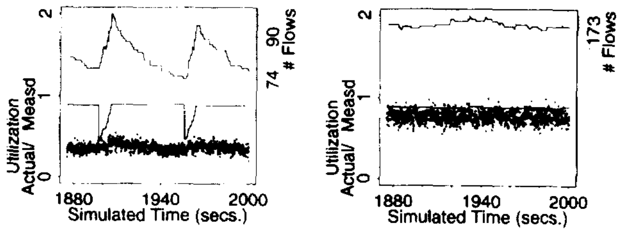
In Section III-B, we showed that T has two related effects on the ACA: 1) too small a T results in more delay violations and lower link utilization and 2) too long a T depresses utilization by keeping the artificially heightened measured values for longer than necessary. While the first effect is linked to flow duration only if the flow exhibits long-range dependence, the second effect is closely linked to the average flow duration in general. The results in this section are meant to be canonical illustrations on the effect of T on the ACA, thus we do not provide the full details of the simulations from which they are obtained.

In Table IX(a), we show the average link utilization and maximum experienced delay from simulations of flows with average duration of 300 s. We varied the measurement window, T, from $1e4$ packet times to $5e5$ packet times. Notice how smaller T yields higher utilization at higher experienced delay and larger T keeps more reliable delay bounds at the expense of utilization level. Next, we fixed T and varied the average flow duration. Table IX(b) shows the average link utilization and maximum experienced delay for different values of average flow duration with T fixed at $1e4$ and $1e5$. We varied the average flow duration from 3000 s (practically infinite, given our simulation duration of the same length) to 30 s. Notice how longer lasting flows allow higher achieved link utilization while larger measurement periods yield lower



(a) (b)

Fig. 4. Effect of T on experienced delay. (a) Smaller T. (b) Larger T.



(a) (b)

Fig. 5. Effect of T on link utilization. (a) Larger T. (b) Smaller T.

link utilization. Link utilization is at its highest when the $\overline{L}/T$ ratio is the largest and at its lowest when this ratio is the smallest. On the other hand, the smaller $\overline{L}/T$ ratio means lower experienced delay and larger $\overline{L}/T$ means the opposite—thus lowering the $\overline{L}/T$ ratio is one way to decrease delay violation rate.

In Figs. 4 and 5, we provide sample path snapshots showing the effect of T on delay and link utilization. We note, however, a T that yields artificially low utilization when used in conjunction with one source model may yield appropriate utilization when used with burstier sources or sources with longer burst time.

F. On Unequal Flow Rejection Rates

Almost all ACA's in the literature are based on the *violation prevention* paradigm: each switch decides to admit a flow if and only if the switch can still meet all of its service commitments. In other words, the *only* criteria considered by ACA's based on the violation prevention paradigm are whether any service commitments will be violated as a result of a new admission. In this section we discuss some policy or allocation issues that arise when not all flows are completely equivalent. When flows with different characteristics—either different service requests, different holding times, or different path lengths—compete for admission, ACA's based purely on violation prevention can sometimes produce equilibria with some categories of flows experiencing higher rejection rate than other categories do. In particular, we identify two causes of unequal rejection rate: 1) flows traversing a larger number of hops have a higher chance of being rejected by the network and 2) flows requesting more resources are more likely to be rejected by the network.

1) Effect of Hop Count on Flow Rejection Rates: As expected, when the network is as loaded as in our simulations, multihop flows face an increased chance of being denied service by the network. For example, in our simulation with homogeneous sources on the TWO-LINK network, as reported in Table IV, more than 75% of the 700 new EXP1 sources admitted under guaranteed service are single-hop flows. This is true for both of the bottleneck links. A somewhat smaller percentage of the more than 1000 flows admitted under predictive service are single-hop flows. This effect is even more pronounced for sources that request larger amount of resources, e.g., the POO2 or the FARIMA sources. Furthermore, it is exacerbated by sources with longer lifetimes: with fewer departures from the network, new flows see an even higher rejection rate.

Aside from disparity in the kinds of flow present on the link, this phenomenon also affects link utilization; upstream switches (switches closer to source hosts) could yield lower utilization than downstream switches. We observe two causes to this: 1) switches that carry only multihop flows could be starved by admission rejections at downstream switches. The utilization numbers of link L6 in both Tables IV and VIII are consistently lower than the utilization of the other links in the FOUR-LINK topology. Notice that we set these simulations up with no single hop flow on link L6. The low utilization is thus not due to the constraint put on by link L6's *own* admission decisions, but rather is due to multihop flows being rejected by downstream switches. 2) Nonconsummated reservations depress utilization at upstream switches; to illustrate: a flow admitted by an upstream switch is later rejected by a downstream switch; meanwhile, the upstream switch has increased its measurement estimates in anticipation of the new flow's traffic, traffic that never come. It takes time (to the expiration of the current measurement window) for the increased values to come back down. During this time, the switch cannot give the reserved resources away to other flows. We can see this effect by comparing the utilization at the two bottleneck links of the TWO-LINK topology as reported in Table IV. Note, however, even with the presence of this phenomenon, the utilization achieved under predictive service with our measurement-based ACA still outperforms those achieved under guaranteed service.

2) Effect of Resource Requirements on Flow Rejection Rates: Sources that request smaller amount of resources can prevent those requesting larger amount of resources from entering the network. For example, in the simulation using the EXP2–EXP3 source pair reported in Table VI, 80% of the 577 new guaranteed flows admitted after the simulation warmup period were EXP2 flows, which are less resource demanding. In contrast, 40% of flows admitted under predictive service with our measurement-based ACA were the more resource demanding EXP3 flows. Another manifestation of this case is when there are sources with large bucket sizes trying to get into a high priority class. Because the delay of a lower priority class is affected by *both* the rate and bucket size of the higher priority flow (as explained in Section II-A), the ACA is more likely to reject flows with a large bucket size and high priority than those with a smaller bucket size or low priority. We see this phenomenon in the simulation of source model EXP3 reported in Table V. When all sources request either of the two classes of predictive service with equal probability, of the 1162 flows admitted after the simulation warmup period, 83%

were of class 2. When sources request guaranteed or second class predictive service, only 8% of the 1137 new flows ends up being guaranteed flows. In both of these scenarios, the link utilization achieved is 31%, which is lower than the 62% achieved when all flows request only class 2 predictive service (see Table III), but still order of magnitude higher than the 2% achieved when all flows request only guaranteed service (again, see Table III).

We consider the unequal rejection rate phenomenon a policy issue (or rather, several policy issues) because there is no delay violations and the network is still meeting all its service commitments (which is the original purpose of admission control); the resulting allocation of bandwidth is, however, very uneven and might not meet some policy requirements of the network. We want to stress that this unequal rejection rate phenomenon arises in *all* ACA's based on the *violation prevention* paradigm. In fact, our data show that these uneven allocations occur in sharper contrast when all flows request guaranteed service, when admission control is a simple bandwidth check. Clearly, when possible service commitment violations is the only admission control criteria, one cannot ensure that policy goals will be met. Our purpose in showing these policy issues is to highlight their existence. However, we do not offer any mechanisms to implement various policy choices; that is the subject of future research and is quite orthogonal to our focus on measurement-based admission control.

V. Miscellaneous Practical Deployment Considerations

We have not yet addressed the issue of how to adjust the level of conservatism (through T) automatically, and this will be crucial before such measurement-based approaches can be widely deployed. The appropriate values of T, and the other parameters, must be determined from observed traffic over longer time scales than discussed (and simulated) here. We have not yet produced such an higher order control algorithm. In the simulations presented in this paper, we chose a value of T for each simulation that yielded *no* delay bound violation over the course of the simulation at "acceptable" level of utilization.

We should also note that our measurement-based approach is vulnerable to spontaneous correlation of sources, such as when all the TV channels air coverage of a major event. If all flows suddenly burst at the same time, delay violations will result. We are not aware of any way to prevent this kind of delay violation, since the network cannot predict such correlations beforehand. Instead, we rely on the uncorrelated nature of statistically multiplexed flows to render this possibility a very unlikely event.

As we mentioned earlier, when there are only a few flows present, or when a few large-grain flows dominate the link bandwidth, the unpredictability of individual flow's behavior dictates that a measurement-based ACA must be very conservative. One may need to rely less on measurements and more on the worst-case parameters furnished by the source, and perform the following bandwidth check instead of (5)

$$v\mu > \tilde{\nu}_G + \sum_{i=1}^{K} \hat{\nu}_i \tag{12}$$

where

$$\begin{aligned}\tilde{\nu}_G &= \hat{\nu}_G + \kappa(\mathrm{MAX}\,(0, \nu_G - \hat{\nu}_G))\\ \tilde{\nu}_j &= \hat{\nu}_j + \kappa(\mathrm{MAX}\,(0, \nu_j - \hat{\nu}_j)), \qquad j = 1 \cdots K\end{aligned}$$

ν_G is the sum of all reserved guaranteed rates, ν_j is the sum of all reserved rates in class j, K is number of predictive classes, and κ is a fraction between zero and one. For $\kappa = 1$, we have the completely conservative case. Similarly, one could do the following delay check:

$$D_j = \frac{\sum_{i=1}^{j} \kappa \sum_{\alpha \in \{i\}} b_i^{\alpha}}{\mu - \kappa\nu_G - \kappa\sum_{i=1}^{j-1} \nu_i}. \tag{13}$$

for every predictive class j for which one needs to do a delay check as determined in Section II-D.

VI. Conclusion

In this paper, we presented a measurement-based ACA that consists of two logically distinct pieces, the *criteria* and the *estimator*. The admission control criteria are based on an equivalent token bucket filter model, where each predictive class aggregate traffic is modeled as conforming to a single token bucket filter. This enables us to calculate worst case delays in a straightforward manner. The estimator produces measured values we use in the equations representing our admission control criteria. We have shown that even with the most simple measurement estimator, it is possible to provide a reliable delay bound for predictive service. Thus we conclude that predictive service is a viable alternative to guaranteed service for those applications willing to tolerate occasional delay violations. For bursty sources, in particular, predictive service provides fairly reliable delay bounds at network utilization significantly higher than those achievable under guaranteed service.

Acknowledgment

This extended version of the ACM SIGCOMM'95 paper [25] has benefited from discussions with S. Floyd, S. Keshav, and W. Willinger; it has also been improved by incorporating suggestions from the anonymous referees. The authors thank them.

References

[1] A. Adas and A. Mukherjee, "On resource management and QoS guarantees for long range dependent traffic," in *Proc. IEEE INFOCOM 1995*.

[2] S. Abe and T. Soumiya, "A traffic control method for service quality assurance in an ATM network," *IEEE J. Select. Areas Commun.*, vol. 12, no. 2, pp. 322–331, Feb. 1994.

[3] J. Beran, *Statistics for Long-Memory Processes*. New York: Chapman & Hall, 1994.

[4] G. E. P. Box and G. M. Jenkins, *Time Series Analysis: Forecasting and Control*. Englewood Cliffs, NJ: Prentice-Hall, 1976.

[5] V. A. Bolotin, "Modeling call holding time distributions for CCS network design and performance analysis," *IEEE J. Select. Areas Commun.*, vol. 12, no. 3, pp. 433–438, Apr. 1994.

[6] 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, 1995.

[7] M. Conti *et al*, "Interconnection of dual bus MANs: architecture and algorithms for bandwidth allocation," *J. Internetworking: Res. Exper.*, vol. 2, no. 1, pp. 1–22, Mar. 1991.

[8] S. Chong, S-Q. Li, and J. Ghosh, "Predictive dynamic bandwidth allocation for efficient transport of real-time VBR video over ATM," *IEEE J. Select. Areas Commun.*, vol. 13, no. 1, pp. 12–23, Jan. 1995.

[9] R. L. Cruz, "A calculus for network delay, Part I: network elements in isolation," *IEEE Trans. Inform. Theory*, vol. 37, no. 1, pp. 114–131, Jan. 1991.

[10] D. D. Clark, S. J. Shenker, and L. Zhang, "Supporting real-time applications in an integrated services packet network: architecture and mechanism," in *Proc. ACM SIGCOMM '92*, pp. 14–26.

[11] Z. Dziong, M. Juda, and L. G. Mason, "A framework for bandwidth management in ATM networks—aggregate equivalent bandwidth estimation approach," *IEEE/ACM Trans. Networking*, submitted for publication.

[12] M. Degermark, T. Köhler, S. Pink, and O. Schelén, "Advance reservations for predicted service," in *Proc. 5th Int. Network and Operating Systems Support for Digial Audio and Video Workshop*, 1995, pp. 3–14.

[13] D. E. Duffy, A. A. McIntosh, M. Rosenstein, and W. Willinger, "Statistical analysis of CCSN/SS7 traffic data from working CCS subnetworks," *IEEE J. Select. Areas Commun.*, vol. 12, no. 3, pp. 544–551, Apr. 1994.

[14] S. Floyd, "Comments on measurement-based admissions control for controlled-load service," *Computer Commun. Rev.*, submitted for publication, 1996.

[15] D. Ferrari and D. C. Verma, "A scheme for real-time channel establishment in wide-area networks," *IEEE J. Select. Areas Commun.*, vol. 8, no. 3, pp. 368–379, 1990.

[16] R. Guérin, H. Ahmadi, and M. Naghshineh, "Equivalent capacity and its application to bandwidth allocation in high-speed networks," *IEEE J. Select. Areas Commun.*, vol. 9, no. 7, pp. 968–981, Sept. 1991.

[17] R. J. Gibbens, F. P. Kelly, and P. B. Key, "A decision-theoretic approach to call admission control in ATM Networks," *IEEE J. Select. Areas Commun.*, vol. 13, no. 6, pp. 1101–1114, Aug. 1995.

[18] M. Garrett and W. Willinger, "Analysis, modeling and generation of self-similar VBR video traffic," in *Proc. ACM SIGCOMM '94*, pp. 269–279.

[19] A. Hiramatsu, "Integration of ATM call admission control and link capacity control by distributed neural network," *IEEE J. Select. Areas Commun.*, vol. 9, no. 7, pp. 1131–1138, Sept. 1991.

[20] J. M. Hyman, A. A. Lazar, and G. Pacifici, "A separation principle between scheduling and admission control for broadband switching," *IEEE J. Select. Areas Commun.*, vol. 11, no. 4, pp. 605–616, May 1993.

[21] J. R. M. Hosking, "Modeling persistence in hydrological time series using fractional differencing," *Water Resources Res.*, vol. 20, no. 12, pp. 1898–1908, Dec. 1984.

[22] J. Haslett and A. E. Raftery, "Space-time modeling with long-memory dependence: assessing ireland's wind power resource," *Appl. Stat.*, vol. 38, no. 1, pp. 1–50, 1989.

[23] J. Y. Hui, "Resource allocation for broadband networks," *IEEE J. Select. Areas Commun.*, vol. 6, no. 9, pp. 1598–1608, Dec. 1988.

[24] S. Jamin, "A measurement-based admission control algorithm for integrated services packet network," Ph.D. dissertation Proposal Excerpts, Computer Science Dept., Univ. of Southern California, Tech. Rep. USC-CS-95-617, 1995.

[25] S. Jamin, P. B. Danzig, S. J. Shenker, and L. Zhang, "A measurement-based admission control algorithm for integrated services packet networks," in *Proc. ACM SIGCOMM '95*, pp. 2–13.

[26] S. Jamin, S. J. Shenker, L. Zhang, and D. D. Clark, "An admission control algorithm for predictive real-time service (extended abstract)," in *Proc. 3rd Int. Network and Operating Systems Support for Digial Audio and Video Workshop*, Nov. 1992.

[27] F. P. Kelly, "Effective bandwidths at multi-class queues," *Queueing Syst.*, vol. 9, pp. 5–16, 1991.

[28] S. M. Klivansky and A. Mukherjee, "On long-range dependence in nsfnet traffic," Gerogia Inst. of Technol., Atlanta, Tech. Rep. GIT-CC-94-61, Dec. 1994.

[29] S-Q. Li, S. Chong, and C-L. Hwang, "Link capacity allocation and network control by filtered input rate in high-speed networks," *ACM/IEEE Trans. Networking*, vol. 3, no. 1, pp. 10–25, Feb. 1995.

[30] 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, no. 1, pp. 1–15, Feb. 1994.

[31] E. C Molina, "Application of the theory of probability to telephone trunking problems," *Bell Syst. Tech. J.*, vol. 6, pp. 461–494, 1927.

[32] R. Nagarajan and J. Kurose, "On defining, computing, and guaranteeing quality-of-service in high-speed networks," in *Proc. IEEE INFOCOM'92*.

[33] H. Ohnishi, T. Okada, and K. Noguchi, "Flow control schemes and delay/loss tradeoff in ATM networks," *IEEE J. Select. Areas Commun.*, vol. 6, no. 9, pp. 1609–1616, Dec. 1988.

[34] A. K. Parekh, "A generalized processor sharing approach to flow control in integrated services networks," Ph.D. dissertation, Lab. for Information and Decision Systems, MIT, Cambridge, MA, Tech. Rep. LIDS-TR-208, 9 1992.

[35] V. Paxson and S. Floyd, "Wide-area traffic: the failure of poisson modeling," in *Proc. ACM SIGCOMM'94*, pp. 257–268.

[36] S. J. Shenker, D. D. Clark, and L. Zhang, *A Scheduling Service Model and a Scheduling Architecture for an Integrated Services Packet Network*. URL ftp://parcftp.parc.xerox.com/pub/net-research/archfin.ps, 1993.

[37] H. Saito and K. Shiomoto, "Dynamic call admission control in ATM networks," *IEEE J. Select. Areas Commun.*, vol. 9, no. 7, pp. 982–989, Sept. 1991.

[38] R. Warfield, S. Chan, A. Konheim, and A. Guillaume, "Real-time traffic esitmation in ATM networks," *Int. Teletraffic Congr.*, June 1994.

[39] W. Willinger, M. S. Taqqu, R. Sherman, and D. V. Wilson, "Self-similarity through high-variability: statistical analysis of Ethernet LAN traffic at the source level," in *Proc. ACM SIGCOMM '95*, pp. 100–113.

[40] H. Zhang and D. Ferrari, "Improving utilization for deterministic service in multimedia communication," presented at the IEEE Int. Conf. Multimedia Computing and Systems, 1994.

[41] H. Zhang and E.W. Knightly, "Providing end-to-end statistical performance guarantee with bounding interval dependent stochastic models," in *Proc. ACM SIGMETRICS'94*, pp. 211–220.

Sugih Jamin (S'89–M'96) received the B.A. degree in rhetoric and applied mathematics from the University of California, Berkeley, in 1989, and the M.Sc. and Ph.D. degrees in computer science from the University of Southern California, Los Angeles, in 1996.

He is currently an Assistant Professor at the University of Michigan, Ann Arbor, where he joined the faculty in August of 1996. He spent parts of the 1992–1993 academic year at the Xerox Palo Alto Research Center. His research interests include measurement-based admission control, network congestion control, network traffic characterization, multicast transport protocols, performance evaluation, web caching, distributed data replication, and network agents. He maintains a home site at http://irl.eecs.umich.edu/jamin, which lists his current contact information.

Dr. Jamin received the Best Student Paper Award from ACM SIGCOMM in 1995.

Peter B. Danzig (S'84–M'85) received the B.S. degree in applied physics from the University of California, Davis, in 1982, and the Ph.D. degree in computer science from the University of California, Berkeley, in 1989.

He is currently an Associate Professor at the Univerity of Southern California, Los Angeles. His research addresses both building scalable Internet information systems and flow, congestion, and admission control algorithms for the Internet. He has served on several ACM SIGCOMM and ACM SIGMETRICS program committees and is an Associate Editor of *Internetworking: Research and Experience*.

Scott J. Shenker (S'87–SM'95) received the Sc.B. degree in from Brown University, in 1978, and the Ph.D. degree in theoretical physics from the University of Chicago, in 1983.

He is currently a Principal Scientist at the Xerox Palo Alto Research Center. He spent the 1983–1994 academic year at Cornell University, Ithaca, NY, as a Post-Doctoral Associate. His most recent computer science research focuses on the design of integrated service packet networks and the related issues of service models, scheduling algorithms, and reservation protocols. His recent economic research addresses incentive compatibility and fairness in various cost sharing mechanisms. Besides computer networks and theoretical economics, his other research interests include chaos in nonlinear systems, critical phenomena, distributed algorithms, conservative garbage collection, and performance analysis.

Lixia Zhang (S'81–M'86–SM'94) received the Ph.D. degree in computer sciences from the Massachusetts Institute of Technology, Cambridge, in 1989.

She is an Associate Professor of Computer Science at the University of California, Los Angeles, where she joined the faculty in January of 1996. Prior to that, she was a member of the Research Staff at Xerox PARC, engaged in research on advanced networking technologies, including high performance transport protocols, reliable multicast, and integrated services support over the Internet. She is the Co-Chair of the IETF RSVP Working Group, a member of the IETF Transport Area Directorate, and wa a member of the Internet Architecture Board from 1994 to 1996.