A Measurement-based Admission Control Algorithm for Integrated Services Packet Networks

Sugih Jamin, Peter B. Danzig
Computer Science Department
University of Southern California
Los Angeles, California 90089-0781
{jamin,danzig}@usc.edu

Scott Shenker, Lixia Zhang
Palo Alto Research Center
Xerox Corporation
Palo Alto, California 94304-1314
{shenker,lixia}@parc.xerox.com

Abstract: Many designs for integrated service 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 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, at least for the scenarios studied here, the 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 the delay bound.

1 Bounded Delay Services and Predictive Service

There have been many proposals for supporting real-time applications in packet networks; see [OON88, FV90, GAN91] for a few representative examples. Most of these proposals provide some form of bounded delay packet delivery service. The CSZ proposal [CSZ92] introduced *predictive* service which is designed to achieve higher utilizations than more traditional bounded delay services by allowing occasional delay bound violations. The ability of bounded delay services to achieve high utilizations and also meet their service commitments depends crucially on the admission control algorithm. Conversely, the ability of an admission control algorithm to increase network utilization is ultimately constrained by the service commitments the network makes. In this section we first look at different service models and their admission control algorithms. We then describe the implementation and use of predictive service in greater detail.

Sugih 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 is supported by AFOSR award number F49620-93-1-0082, by the NSF small-scale infrastructure grant, award number CDA-9216321, and by equipment loan from Sun Microsystems, Inc. At PARC, this research was supported in part by the Advanced Research Projects Agency, monitored by Fort Huachuca under contract DABT63-94-C-0073. The views expressed here do not reflect the position or policy of the U.S. government.

Traditional real-time service provides a hard or absolute bound on the delay of every packet; in the literature, this service model is called *guaranteed* service. 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, references [CSZ92, SCZ95] proposed predictive service, which offers a fairly, but not absolutely, reliable bound on packet delivery times. It is important to note that the service definition itself does not specify the 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 [NK92]. 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 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.[1] This ability to occasionally incur delay violations gives admission control a great deal more flexibility, and is the chief advantage of predictive 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 [FV90] or a filter like a token bucket [OON88]; these descriptions provide upper bounds on the traffic that can be generated by the source. Traditional approaches to admission control, like those used 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. Calculating the worst-case delays may be very complex, but the underlying admission control principle is conceptually simple: does granting a new request for service cause the worst-case behavior of the network to violate any delay bound? (See [FV90] for an example of this approach.) Network utilization under this model is usually acceptable when flows are smooth; when flows are bursty, however, their traffic characterizations necessarily must be quite loose, in that the average behavior of the flows is significantly less than the upper bound of the traffic descriptions, and guaranteed service inevitably results in low utilization [ZF94].

There are many other approaches to admission control that attempt to achieve higher utilizations by weakening the degree of

[1] 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; our point is that the provider cannot realistically assure that such excessive violations will never occur.

reliability of the delay bound. For instance, the probabilistic delay bound service described in [ZK94] does not provide for the worst-case scenario, instead it guarantees a bound on the probability of lost/late packets based on statistical characterization of traffic ([VPV88]). In such an approach, each flow is allotted an effective bandwidth that is larger than its average rate but less than its peak rate; network utilization is thus increased. In most cases the *a priori* characterization of flows is based on a statistical model [Hui88, SS91] or on a fluid flow approximation [GAN91, Kel91]).[2] We think it will be quite difficult, if not impossible, to provide accurate and tight statistical models for each individual flow. Therefore the *a priori* traffic characterizations handed to admission control will inevitably be fairly loose upper bounds. For instance, the average bit rate produced by a given codec in a teleconference will depend on the participant's body movements. This can't possibly be predicted in advance with any degree of accuracy.

For this reason, we think that *measurement based* admission control will play a key role in achieving high network utilizations. The measurement based admission control approach advocated in [CSZ92, JSZC92] 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. For instance, if a source describes itself as conforming to a token bucket with a rate of 5 Mbps, but typically sends at an average rate of 1 Mbps, the measurement-based admission control approach does not indefinitely continue to set aside 5 Mbps for this flow, unlike the more traditional forms of admission control. Because it relies on measurements, and source behavior is not always static, 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, because of the unpredictable variations in individual flows, these measurement based approaches must be very conservative, by using some worst-case calculation for example, when there are only a few flows present—and thus have significant gain in utilization only when there is a high degree of multiplexing.

Predictive service differs in two important ways from traditional guaranteed service: first, the service commitment is somewhat less reliable, and second, 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. In this paper we describe a measurement based admission control algorithm for predictive service. The admission control criteria presented here are superficially similar to a preliminary version presented in the extended abstract [JSZC92]. The equations underlying the admission control algorithm in [JSZC92] were somewhat ad hoc, whereas here we motivate the present equations with a more formal and controlled approximation. In addition, we present substantially more simulations in this work, including some driven by traffic source models that exhibit long-range dependence, both in themselves and in their aggregation. We demonstrate affirmative answers to the following two questions. First, can one provide reliable delay bounds with a measurement-based admission control algorithm? Second, if one does indeed achieve reliable delay bounds, does offering predictive service increase network utilization?

[2] We refer the interested readers to [Jam95] for a more comprehensive overview and bibliography of admission control algorithms.

The authors of [HLP93] use measurements to determine admission control, but the admission decisions are pre-computed 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. The possibility of using ongoing measurements of load in making admission decisions was suggested, but not fully developed nor explored, in [OON88, GAN91, AS94]. Several recent papers, such as [SS91, Hir91, CLG95] presented measurement-based admission control algorithms. In general, these references do not report extensive simulations of their approaches; in addition, the authors of [AS94, GAN91, SS91] assume certain stationary bit rate distributions and use measurements to obtain the parameters of the distributions. We do not make assumptions on traffic characteristics. Incidentally, our work reported in this paper has been extended in [DKPS95] to support advance reservations. The authors of [DKPS95] have also replicated some of our results on their independently developed network simulator.

1.1 Implementation and Use of Predictive Service

We now describe the implementation of predictive service in more detail. In [CSZ92], the authors described an unified scheduling algorithm for providing both guaranteed and predictive services. Guaranteed service is provided with the weighted fair queueing (WFQ) scheduling discipline described in [DKS89, Par92]. Predictive service is provided with the priority queueing discipline. The priority queue providing predictive service consists of several levels of priority and is served by the bandwidth left over from serving guaranteed service. A switch that implements the CSZ scheme offers several discrete levels of predictive service, each associated with a different delay bound.

In our scheme, the admission control algorithm 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. An end system could, for example, use adaptive source routing, such as the one in [Bre95], to select a route that satisfies its end-to-end requirements. We also assume the existence of a reservation protocol, such as the one in [Z^+93], which the end systems could use to communicate their resource requirements to the network.

Sources requesting service must characterize the worst-case behavior of their flow. In [CSZ92] 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, i.e. no more than b tokens can be accumulated. 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.

For constant bit rate sources, one can set the token rate, r, to the peak traffic generation rate and let the bucket depth, b, be 1. In this case, the token-bucket filter precisely characterizes the traffic coming out of the sender. It is difficult, however, to precisely characterize sources with non-constant bit rate. While the token bucket filter can be used to capture the worst-case behavior, we believe the average behavior is typically not known in advance and must be measured on-line.

Our approach, in contrast to that used for guaranteed service, depends less on the accuracy of a flow's *a priori* characterization. Only when processing a new request, before the system has any

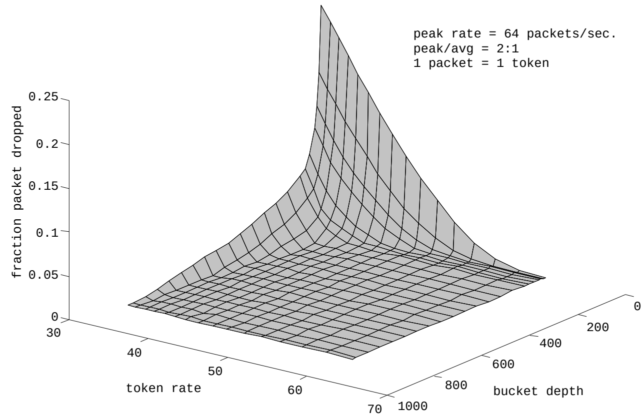

Figure 1: Fraction of packets dropped by token bucket filter as a function of r and b.

observed history of the new flow, are the user-specified parameters used in the decision process. Once a flow starts sending, the system uses measured values to characterize network load in making future admission decisions. The user, then, should pick a token bucket filter which looks like a reasonable upper bound on its behavior. Many non-constant bit rate sources do not naturally conform to a token bucket filter with token rate less than their peak rates. In order to transform the packet stream into one that can conform to a somewhat reasonable token bucket filter, users can choose to either drop, or queue for later transmission, packets that arrive at an empty token bucket (see also Figure 3 in Section 4.2). In general, to assign a token bucket filter to a source, one first generates a plot similar to Fig. 1, then picks a $(r,\ b)$ that gives the desired token-bucket loss/late rate. Figure 1 plots the fraction of packets dropped, as a function of r and b, for the traffic generated with the EXP1 source model described in Section 4.2, if the user elects to drop packets that arrive at an empty token bucket. It is conceivable that future real-time applications will have a module that can, over time, learn a suitable r and b to upper-bound their traffic.

2 Measurement-based Admission Control for ISPN

Our admission control algorithm consists of two logically distinct aspects. The first aspect is the set of admission control criteria for when 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 3. In this section we present the analytical underpinnings of our admission control criteria.

When admitting a new flow, the admission control algorithm not only must decide if that 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 admission control algorithm 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 delays of predictive flows by replacing the worst-case parameters in the analytical models with measured quantities. We call this approximation the *equivalent token bucket filter*. This approximation yields a series of expressions for the expected maximal delays that would results from the admission of a new flow. Recall that in CSZ, switches serve guaranteed traffic with 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 the details of the admission control algorithm based on the equivalent token bucket filter approximations.

2.1 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 [Cru91], derived a tight bound for the worst-case delay, D_j^*, of priority queue level j. Our derivation follows Parekh's [Par92], 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. Further, we assume that the aggregate rate, aggregated over all traffic classes, is within the link capacity ($\sum r_j \leq \mu$).

Theorem 1 *Parekh [Par92]: The worst-case class j delay, with* FIFO *discipline within the class and assuming infinite peak rates for the sources, is*

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

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 were simultaneously dumped into the switch and all classes 1 through $j-1$ sources continued to send at their reserved rates.

We now use Eq. 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 $(\nu_j,\ b_j)$ 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 $(r_k^\alpha,\ b_k^\alpha)$ are the token bucket parameters of

[3] For a proof of Theorem 1, we refer the interested readers to [Par92] or [Jam95].

the new flow:

$$D_k^{*'} = \frac{\sum_{i=1}^{k-1} b_i}{\mu - \nu_G - \sum_{i=1}^{k-1} \nu_i} + \frac{b_k + b_k^\alpha}{\mu - \nu_G - \sum_{i=1}^{k-1} \nu_i}$$
$$= D_k^* + \frac{b_k^\alpha}{\mu - \nu_G - \sum_{i=1}^{k-1} \nu_i}. \quad (2)$$

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 Eq. 1, adding in the bucket depth b_k^α and reserved rate r_k^α of flow α.

$$D_j^{*'} = \frac{\sum_{i=1}^{k-1} b_i + b_k + b_k^\alpha + \sum_{i=k+1}^{j} b_i}{\mu - \nu_G - \sum_{i=1}^{k-1} \nu_i - \nu_k - r_k^\alpha - \sum_{i=k+1}^{j-1} \nu_i}$$
$$= D_j^* \frac{\mu - \nu_G - \sum_{i=1}^{j-1} \nu_i}{\mu - \nu_G - \sum_{i=1}^{j-1} \nu_i - r_k^\alpha} +$$
$$\frac{b_k^\alpha}{\mu - \nu_G - \sum_{i=1}^{j-1} \nu_i - r_k^\alpha}, \quad k < j \leq K, \quad (3)$$

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 effects 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 Eq. 1, adding in the reserved rate, r_G^α, of flow α.

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

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 Eq. 3 is equivalent to Eq. 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. We have observed this phenomenon predicted by the analytical models in our simulations.

2.2 Worst-case Delay: Guaranteed Service

In reference [Par92], Parekh proved 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 admission control algorithm only needs to assure that (1) the new flow will not cause predictive flows to miss *their* delay bound (see Eq. 4 above), and that (2) it will not over-subscribe the link: $\nu_G + r_G^\alpha \leq \upsilon\mu$, where μ is the link bandwidth and υ is the utilization target (see Section 3.2 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 and we need not concern ourselves with the effect of adding a new predictive flow on existing guaranteed traffic.

2.3 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, $\widehat{\nu}_j$, and the experienced packet queueing delay for that class, $\widehat{D}_j$. For guaranteed service, we count the sum of all reserved rates, ν_G, and we measure the actual bandwidth utilization, $\widehat{\nu}_G$, of all guaranteed flows. Our approximation is based on substituting, in the above equations, the measured rates $\widehat{\nu}_j$ and $\widehat{\nu}_G$ for the reserved rates, and substituting the measured delays $\widehat{D}_j$ for the maximal delays. We now use the previous computations and these measured values to formulate an admission control algorithm.

2.4 The Admission Control Algorithm

New Predictive Flow. If an incoming flow α requests service at predictive class k, the admission control algorithm:

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:
$$\upsilon\mu > r_k^\alpha + \widehat{\nu}_G + \sum_{i=1}^{N} \widehat{\nu}_i, \quad (5)$$
2. Denies the request if admitting the new flow could violate the delay bound, D_k, of the same priority level:
$$D_k > \widehat{D}_k + \frac{b_k^\alpha}{\mu - \widehat{\nu}_G - \sum_{i=1}^{k-1} \widehat{\nu}_i}, \quad (6)$$
or could cause violation of lower priority classes' delay bound, D_j:
$$D_j > \widehat{D}_j \frac{\mu - \widehat{\nu}_G - \sum_{i=1}^{j-1} \widehat{\nu}_i}{\mu - \widehat{\nu}_G - \sum_{i=1}^{j-1} \widehat{\nu}_i - r_k^\alpha} +$$
$$\frac{b_k^\alpha}{\mu - \widehat{\nu}_G - \sum_{i=1}^{j-1} \widehat{\nu}_i - r_k^\alpha}, \quad k < j \leq K. \quad (7)$$

New Guaranteed Flow. If an incoming flow α requests guaranteed service, the admission control algorithm:

1. Denies the request if either the bandwidth check in Eq. 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:

$$D_j > \widehat{D}_j \frac{\mu - \widehat{\nu}_G - \sum_{i=1}^{j-1} \widehat{\nu}_i}{\mu - \widehat{\nu}_G - \sum_{i=1}^{j-1} \widehat{\nu}_i - r_G^\alpha}, \qquad 1 \le j \le K. \tag{9}$$

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

3 A Simple Time-window Measurement Mechanism

The formulae described in the previous section rely on the measured values $\widehat{D}_j$, $\widehat{\nu}_G$, and $\widehat{\nu}_j$ 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 mechanism is extremely simple and could be replaced by a number of other approaches. We consider the simplicity of our approach an advantage in our study because it helps us isolate properties inherent in our admission control criteria from those induced by the measurement mechanism. We expect that many other measurement processes would be suitable, and we plan to experiment with alternate approaches in the future. 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.

3.1 Measurement Process

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

Measuring delay. The measurement variable $\widehat{D}_j$ tracks the estimated maximum queueing delay for class j. We use a measurement window of length T as our basic measurement block. The value of $\widehat{D}_j$ is updated on three occasions. At the end of the measurement block, we update $\widehat{D}_j$ 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 $\widehat{D}_j$ to be λ times this sampled delay. The parameter λ allows us to be more conservative by increasing $\widehat{D}_j$ to a value higher than the actual sampled delay. Finally, we update $\widehat{D}_j$ whenever a new flow is admitted, to the value of projected delay from our admission control equations. More precisely, the updating of $\widehat{D_j}$ is as follows:

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

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

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

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

When a flow leaves the network, we do not explicitly tear down its reservation; 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.

3.2 Performance Tuning Knobs

We now look at the constants used in the algorithm.

v: In a simple $M/M/1$ queue, the variance in delay diverges as the system approaches full utilization. We have observed a similar phenomenon in our simulations. A measurement-based approach is doomed to fail when delay variations are exceedingly large, which will occur at very high utilizations. It is thus necessary to identify an *utilization target* and require that the admission control algorithm 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. Sources with big, long bursts will require a lower link utilization target. In this paper, we set utilization target at 90% capacity.

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

S: The averaging period S in Eq. 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. One way to determine the "right" value of S is by using the *batch mean* method as explained in [Jai91]. In this paper we use S of at least 500 packet transmission times.

T: Once $\widehat{D}$ or $\widehat{\nu}$ 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 of each other. When we admit a new flow and add its worst case effect to the measured values, we also restart the measurement window to give the measurement mechanism one whole window to gather information on the new flow.

Of the four performance knobs, υ, λ, S, and T, tuning T provides the most pronounced effect on experienced delay and link utilization. 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. Varying T has two related effects on the admission control algorithm. First, since T is the length of the measurement block used to determine maximal packet delays and sampled utilizations, increasing T makes these estimates more conservative, which in turn makes the admission control algorithm 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 delays and utilizations 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 $\overline{r}$ is the average flow reservation rate, and μ the link bandwidth (and assuming $\upsilon = 1$ for convenience), we will admit at most $\mu/\overline{r}$ 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 $A = \mu/\overline{r}$ number of flows admitted every T period. Thus at the end of its average lifetime, $\overline{L}$, an average flow would have seen approximately $F = A * \overline{L}/T$ number of flows. If the average rate of an average flow is $\widehat{\overline{r}}$, ideally we want $F * \widehat{\overline{r}}$ to be near μ as a link's stable utilization level.

However, flows also depart from the network. The expected number of 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 $F * \widehat{\overline{r}}$ becomes *much* smaller than μ. For the same average reservation rate, $\overline{r}$, and a given T, the size of the stable F is determined by the average flow duration, $\overline{L}$. A shorter average flow duration means more departure per T. In the long run, we aim for $F * \widehat{\overline{r}} \approx \mu$, or equivalently, $\overline{L}/T \approx \overline{r}/\widehat{\overline{r}}$. If all flows use exactly what they reserved, we have $\overline{L}/T = 1$, meaning that we should not try to give away the flows' reservations. We present further illustrative simulation results on the importance of the $\overline{L}/T$ ratio in Section 4.5.

To deploy our admission control algorithm in the wider world, we would need some form of learning algorithm which could, over longer time scales than discussed (and simulated) here, determine the appropriate values for T, and the other parameters, given the observed traffic patterns. We have not yet produced such a 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.

4 Simulations

Admission control algorithms for guaranteed service can be verified by formal proof. Measurement based admission control algorithms 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 results from our simulations, we first present the topologies and source models used in them.

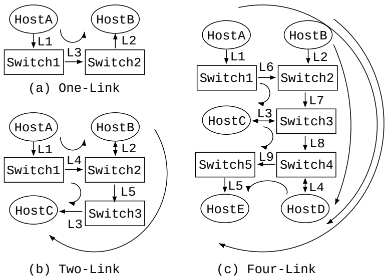

Figure 2: Simulated topologies

4.1 Simulated Topologies

For this paper, we ran our simulations on three topologies: the ONE-LINK, TWO-LINK, and FOUR-LINK topologies depicted in Figures 2(a), (b), and (c) respectively. In all our topologies, each host is connected to a switch by an infinite bandwidth link. The connection between switches are all 10 Mbps 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. Finally, 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 Figure 2, the host pairs and the paths their packets traverse are indicated by the directed curve lines.[4]

4.2 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 rather standard two-state Markov process used widely in the literature. Recent studies ([LTWW94, DMRW94, PF94, KM94, GW94, BSTW95]) 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. Reference [PF94] further called attention to the effect long-range dependent traffic might have on the feasibility of measurement-based admission control. To investigate this and other LRD related questions, we augmented our simulation study with two LRD source models.

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/sec. Let I milliseconds be the average of the exponentially distributed OFF times, then the average packet generation rate a is given by $1/a = I/N + 1/p$. From traffic traces, we observed that current implementations of *vat* and *nv* generate fixed rate traffic that can be well modeled with the EXP model. In particular, the EXP1 model described in the next section is a model for packetized voice encoded using ADPCM at 32 Kbps.

LRD: Pareto-ON/OFF. Our next model is an ON/OFF process with Pareto distributed ON time and exponentially distributed OFF time [Flo94, CI90] (for ease of reference, we call this the *Pareto-*

[4] We have results for larger networks in an extended version of this paper.

ON/OFF model). During each ON period, a Pareto distributed number of packets, with mean N and Pareto shape parameter β, are generated at some peak rate p packet/sec. Pareto shape parameter less than 1 gives data with infinite mean; β less than 2 results in data with infinite variance. The Pareto location parameter is $N * (\beta - 1)/\beta$. The OFF times are exponentially distributed with average I milliseconds. Each *Pareto*-ON/OFF source by itself does not generate LRD series. However, the aggregation of them (of degree greater than 10 [Wil95]) does [LTWW94, PF94, KM94].

LRD: Fractional ARIMA. We use each number generated by the *fractional autoregressive integrated moving average* (fARIMA) process ([HR89]) as the number of fixed-size packets to be sent back to back in the next ON period. Interarrival 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 an 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 [GW94] 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 [GW94] indicated that aggregate 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 [BJ76, Hos84] for details). 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 adding the absolute value of its minimum to every number in the series. This way of obtaining non-negative series is also used in [AM95]. Note that this shifting constrains the maximum value of the generated series to be always twice its average. The Whittle maximum likelihood estimator [Ber94] confirms that our shifting, cropping, and overlaying of the fARIMA generated series does not destroy its long-range dependence.

To ease discussion on the effect of different source models on traffic characteristics, it is useful to define the following additional concepts [OON88]: ρ is a source's *density* (the ratio of its average to peak rate, a/p), R a source's *grain size* (the ratio of its peak rate to link bandwidth, p/μ). The burstiness of a source is measured by $1/\rho$. Figure 3 shows a packet-arrival depiction of the ON/OFF model 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 0, packets that arrive at an empty token bucket are immediately dropped.

In addition to each source's characteristics of density and grain size, network traffic dynamics is also shaped 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 use exponentially distributed CHT for the EXP model, following [Mol27]. The CHTs for

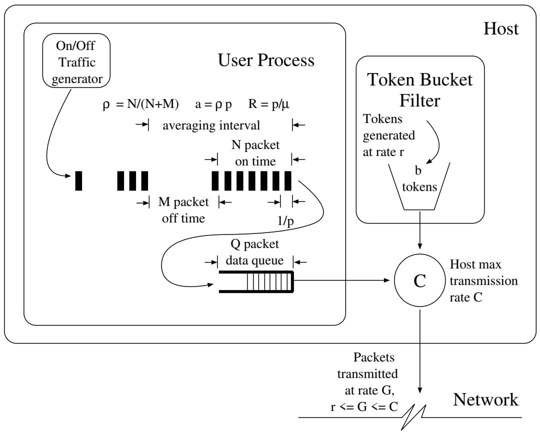

Figure 3: ON/OFF traffic model with token-bucket filter

the LRD models, however, are taken from a lognormal distribution, following [Bol94, DMRW94]. The interarrival times of all flows are exponentially distributed [PF94].[5]

4.3 Parameter Choices

We chose six instantiations of the above three source models, as summarized in Table 1. In the table, $p = \infty$ means that after each OFF time, packets for the next ON period are transmitted back to back. In the same table, we also list the settings of the token bucket parameters assigned to each source. Column 8 of the table, 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 0 (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. In order to preserve the tail of the ON-time distribution, which is crucial to the *Pareto*-ON/OFF model, 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 Kbits of data, which is also our packet size.

Column 10 of the table, labeled D^*, lists the guaranteed delay bound for each source given its assigned token bucket filter. 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 [Par92]. 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 msecs. means first class predictive service, 160 msecs. 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 service are the same. This facilitates the 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 LRD2 and fARIMA cases, the utilization comparison is less meaningful. In the LRD2 case, for example, the predictive delay bound is smaller, 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

[5] Source models used in this paper will be included in a future release of our *tcplib* traffic generator [DJ91]. A fuller description and evaluation of the models are also forthcoming.

Table 1: Six Instantiations of the Three Source Models

Model Name	Model's Parameters				Token Bucket Parameters				Bound (ms)	
	p pkt/ sec	I msec	N pkts	p/a	r tkn/ sec	b tkns	cut rate	max qlen	D^*	D_j
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						
LRD1	64	325	20	1.2	64	1	0	0	16	16
LRD2	256	360	10	1.9	192	4	3.8e-2	125	21	16
				s						
fARIMA ({0.75}, 0.15, -)	∞	125	8	13	1024	100	1.1e-2	34	100	160

autoregressive process with weight 0.75 means our fARIMA traffic also has strong short-range dependence, while maintaining stationarity ([BJ76], 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.

In this paper, flow interarrival times are exponentially distributed with an average of 400 milliseconds. The average holding time of all EXP sources is 300 seconds. The LRD sources, including the fARIMA source, have lognormal distributed holding times with median 300 seconds and shape parameter 2.5.

We ran all our simulations for 3000 seconds 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. (We also ran some of the simulations for 5.5 hours simulated time and confirmed zero delay bound violation.)

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 multi-hop 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 illustrative simulation results showing the effect of the $\overline{L}/T$ ratio on network performance, as discussed in Section 4.2. 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 up this section with a discussion of some general allocation properties of admission control algorithms when not all flows are equivalent; we believe these properties to be inherent in all admission control algorithms whose only admission criterion is to avoid service commitment violations.

4.4 On the Viability of Predictive Service

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

Table 2: Single-hop Homogeneous Sources Simulation Results

Model Name	Guaranteed		Predictive			
	%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
LRD1	33	144	84	364	12	60
LRD2	21	48	77	174	15	30
fARIMA	55	9	81	13	72	60

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 Figure 2(a). For each source, we ran two simulations. The first simulation has all sources requesting guaranteed service. The second one has all sources requesting predictive service. The results of the simulations are shown in Table 2. The column labeled "%Util" contains the link utilization of the bottleneck link, L3. The "#Actv" column contains the average number of active flows concurrently running on that bottleneck link. The "$\lceil d_j \rceil$" column contains the maximum experienced delay of packet of predictive class j. The "$\overline{L}/T$" column lists the ratio of average flow duration to measurement window used with each source model.

As mentioned in Section 4.2, we consider the performance of our admission control algorithm "good" if there is *no* delay bound violation during a simulation run. Even with this very restrictive metric, one can see from Table 2 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 Kbps, corresponding to the token bucket rate at the sources. Under guaranteed service, only 18 flows can be admitted to the 10 Mbps bottleneck link (with 90% utilization target). The actual link utilization is only 2%.[6] Under predictive service, 466 flows are served on the average, resulting in actual link utilization of 62%.

[6] Non-measurement based admission control algorithms may not need to set an utilization target and thus can achieve a somewhat higher utilization; for the scenario

Table 3: Multi-hop Homogeneous Sources Link Utilization

Topology	Link Name	Model Name	Guaranteed %Util	Predictive %Util	Predictive $\lceil d_j \rceil$
TWO-LINK	L4	EXP1	45	67	2
		EXP3	2	44	20
		LRD2	21	74	10
	L5	EXP1	46	78	3
		EXP3	2	58	30
		LRD2	21	79	14
FOUR-LINK	L6	EXP2	17	42	6
		LRD1	22	46	1
		fARIMA	38	54	36
	L7	EXP2	28	71	31
		LRD1	33	81	3
		fARIMA	55	77	40
	L8	EXP2	28	72	24
		LRD1	33	81	3
		fARIMA	53	74	29
	L9	EXP2	28	71	31
		LRD1	34	80	2
		fARIMA	53	80	44

In this homogeneous scenario with only one class of predictive service and constantly oversubscribed link, our measurement-based admission control algorithm easily adapts to LRD traffic between the coming and going of flows. The utilization increased from 33% to 84% and from 21% to 77% for the LRD1 and LRD2 sources. 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. (This is an artifact of the shifting we do, as discussed in Section 4.2, to obtain non-negative values from the fARIMA generated series.) In all cases, we were able to achieve high levels of utilization without incurring delay violations. Thus, at least for these scenarios, we see no reason to conclude that LRD traffic poses special challenges to our measurement-based approach. We will continue to study the effect of LRD traffic on our measurement-based algorithm under different network operating conditions.

Homogeneous Sources: The Multi-hop Case. Next we ran simulations on multi-hop topologies depicted in Figures 2(b) and (c). The top half of Table 3 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 LRD2 models, one per simulation. The bottom half of Table 3 shows the results from simulating source models EXP2, LRD1, and fARIMA on the FOUR-LINK topology. For each source model, we again ran one simulation where all sources requested guaranteed service, and another one where all sources requested *one* class of predictive service. The $\overline{L}/T$ ratio used with each source model was the same one as in the single-hop case.

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.

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 1

simulated here, two more guaranteed flows could have been admitted.

Table 4: Single-hop, Single Source Model, Multiple Predictive Services Link Utilization

Model	PP	GP	GPP
EXP1	77	77	–
EXP2	71	70	–
EXP3	31	31	–
LRD1	82	83	81
LRD2	75	75	–
fARIMA	79	79	78

Table 5: Single-hop, Multiple Source Models, Single Predictive Service Link Utilization

Service	EXP1–LRD1	EXP1–LRD2	EXP2–EXP3	EXP2–fARIMA	EXP3–fARIMA	LRD1–LRD2
Guaranteed	36	35	21	38	18	29
Predictive	83	81	70	79	81	83

and, when requesting predictive service, requested 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 4 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 requested the predictive class "assigned" to the source model under study (see Table 1). In the other cases, both predictive classes, of bounds 16 and 160 msecs. were requested. Compare the numbers in each column of Table 4 with those in the "%Util" column of Table 2 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 5 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.

Finally in Table 6 we show utilization numbers for flows with multiple source models requesting multiple service models. The first row shows the utilization achieved when each source requested a predictive service suitable for its characteristics (see Table 1). The second row shows the utilization when half of the flows requested guaranteed service and the other half requested the predictive service suitable for its characteristics. And the last row shows the utilization achieved when all flows asked only for guaranteed service.

Heterogeneous Sources: The Multi-hop Case. We next ran simulations with four source models, EXP1, EXP2, LRD2, and fARIMA, on all our topologies. In Table 7 we show the utilization level of the bottleneck links of the different topologies. Again, contrast the utilizations achieved under guaranteed service alone with those under both guaranteed and predictive services. The observed low utiliza-

Table 6: Single-hop, Multiple Source Models, Multiple Predictive Services Simulation Results

Service	EXP1–EXP2	EXP1–fARIMA	EXP2–LRD2	EXP3–LRD1	LRD2–fARIMA
Predictive	75	78	76	77	81
Guar./Pred.	73	74	71	80	74
Guaranteed	43	50	22	31	43

Table 7: Single- and Multi-hop, Multiple 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	41	69	15	61
TWO-LINK	L4	38	71	10	73
	L5	38	75	5	81
FOUR-LINK	L6	17	40	1	23
	L7	38	73	7	54
	L8	38	71	13	64
	L9	39	73	10	56

tion 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 multi-hop flows by later hops, as we will explain in Section 4.6.

Our results so far indicate that a measurement-based admission control algorithm can provide reasonable reliability and significant utilization gains. These conclusions appear to hold not just for single hop topologies and smooth traffic sources, but also for multi-hop 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.

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. In particular, we need to address how to cope with low degrees of multiplexing and very large-grain flows. This is a subject of future research. Also, we should note that our measurement-based approach is vulnerable when sources are correlated. If all flows burst at once then delay violations will result. We relied on the uncorrelated nature of flows, high degree of multiplexing, and flow grain sizes less than one tenth of link bandwidth to render this possibility a very unlikely event. However, in the presence of some naturally correlated event (say many simultaneous broadcasts of the same speech) then there might be problems. We are not aware of any way to prevent this, since the network cannot detect such correlations beforehand. Finally, given the assumption made by predictive service that applications can tolerate occasional delay violations, it would be interesting to see how much more utilization gain can be achieved if one operated the network at a delay violation rate acceptable to the applications; we will do this study in the future.

4.5 On the Appropriate Value of T

In Section 4.2 we showed that T has two related effects on the admission control algorithm: (1) too small a T results in more delay violations and lower link utilization, (2) too long a T depresses utilization by keeping the artificially heighten 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.

Table 8: Effect of T and $\overline{L}$

(a)

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

(b)

$\overline{L}$	T = 1e4 %Util	T = 1e4 $\lceil d_j \rceil$	T = 1e5 %Util	T = 1e5 $\lceil d_j \rceil$
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

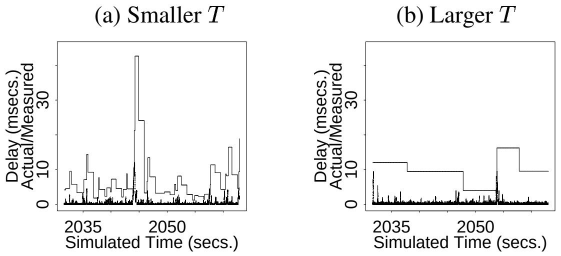

Figure 4: Effect of T on Experienced Delay

In Table 8(a) we show the average link utilization and maximum experienced delay from simulations of LRD2 source model on the ONE-LINK topology with average flow duration of 3000 seconds. 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 8(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 seconds (practically infinite, given our simulation duration of the same length) to 30 seconds. Notice how longer lasting flows allow higher achieved link utilizations while larger measurement periods yield lower link utilizations. 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 Figures 4 and 5 we provide sample path snapshots showing the effect of T on delay and link utilization. Since the figures are meant to be canonical illustrations on the effect of T on the admission control algorithm, we do not provide the details of the simulations from which they are obtained. We note however, a T that yields artificially low utilizations when used in conjunction with one source model may yield appropriate utilizations when used with burstier sources or sources with longer burst time.

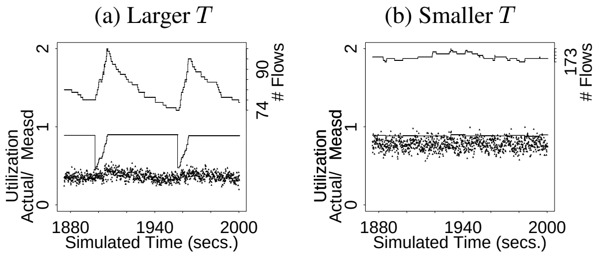

Figure 5: Effect of T on Link Utilization

4.6 Flow Dynamics

Almost all admission control algorithms 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 admission control algorithms based on the *violation prevention* paradigm is 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, admission control algorithms based purely on violation prevention can sometimes produce equilibria with some categories of flows very under represented. In particular we identify two cases of under representation: First, as expected, when the network is as loaded as in our simulations, multi-hop 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 3, more than 75% of the 700 new EXP1 sources admitted under guaranteed service after the warmup period 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 LRD2 or the fARIMA sources. And 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 observed two causes to this: (1) switches that carry only multi-hop flows could be starved by admission rejections at downstream switches. The utilization of link L6 in both Tables 3 and 7 is 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 multi-hop flows being rejected by downstream switches. (2) non-consummated 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 values in anticipation of the new flow's traffic, traffic that never came. 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 utilizations at the two bottleneck links of the TWO-LINK topology as reported in Table 3. Note, however, even with the presence of this phenomenon, the utilization achieved under predictive service with our measurement-based admission control algorithm still outperforms those achieved under guaranteed service.

The second case of under representation occurs when sources have different characteristics. Sources that request a smaller rate can prevent those requesting larger rate from getting into the network. For example, in the simulation using the EXP2–EXP3 source pair reported in Table 5, 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 admission control algorithm 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 higher priority class. Because the delay of the lower priority class is affected by *both* the rate and bucket size of the higher priority flow (as explained in Section 2.1), the admission control algorithm is more likely to reject flows with large bucket sizes and high priority than those with smaller bucket size or low priority. We see this phenomenon in the simulation of source model EXP3 reported in Table 4. When all sources requested 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 requested guaranteed or second class predictive service, only 8% of the 1137 new flows ended 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 requested only class 2 predictive service (see Table 2), but still order of magnitude higher than the 2% achieved when all flows requested only guaranteed service (again, see Table 2).

We consider the under representation 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 under representation phenomenon arises in *all* admission control algorithms based on the *violation prevention* paradigm when the links are as oversubscribed as in our simulations. In fact, our data shows that these uneven allocations occur—actually, 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.

5 Conclusion

In this paper we presented a measurement based admission control algorithm. The admission control algorithm 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 utilizations significantly higher than those achievable under guaranteed service.

Appreciations

We thank Walter Willinger, Vern Paxson, Sally Floyd, Amarnath Mukherjee, and Mark Garrett for explaining the details of self-similar traffic. Walter Willinger graciously provided us with Jan Beran's S code for the Whittle's estimator [Ber94]. We enjoyed fruitful discussions with Lee Breslau, Ron Frederick, Deborah Estrin, and John A. Silvester. Sugih Jamin further thanks Allison Mankin, Allyn Romanow, and Uniforum for supporting his research. And we thank the anonymous referees for their suggestions.

References

[AM95] A. Adas and A. Mukherjee. "On Resource Management and QoS Guarantees for Long Range Dependent Traffic. *Proc. of*

IEEE INFOCOM 1995, Apr. 1995.

[AS94] S. Abe and T. Soumiya. "A Traffic Control Method for Service Quality Assurance in an ATM Network". *IEEE Journal of Selected Areas in Communication*, 12(2):322–331, Feb. 1994.

[Ber94] J. Beran. *Statistics for Long-Memory Processes*. New York: Chapman & Hall, 1994.

[BJ76] G.E.P. Box and G.M. Jenkins. *Time Series Analysis: Forecasting and Control*. New Jersey: Prentice Hall, 1976.

[Bol94] V.A. Bolotin. "Modeling Call Holding Time Distributions for CCS Network Design and Performance Analysis". *IEEE Journal of Selected Areas in Communication*, 12(3):433–438, Apr. 1994.

[Bre95] L. Breslau. *Adaptive Source Routing of Real-Time Traffic in Integrated Services Networks*. PhD thesis, USC, 1995.

[BSTW95] J. Beran, R. Sherman, M.S. Taqqu, and W. Willinger. "Long-range Dependence in Variable-Bit-Rate Video Traffic". *IEEE Transactions on Communications*, 43:1566–1579, 1995.

[CI90] D. Cox and V. Isham. *Point Processes*. New York: Chapman & Hall, 1990.

[CLG95] S. Chong, S-Q. Li, and J. Ghosh. "Predictive Dynamic Bandwidth Allocation for Efficient Transport of Real-Time VBR Video over ATM". *IEEE Journal of Selected Areas in Communication*, 13(1):12–23, Jan. 1995.

[Cru91] R.L. Cruz. "A Calculus for Network Delay, Part I: Network Elements in Isolation". *IEEE Transactions on Information Theory*, 37(1):114–131, Jan. 1991.

[CSZ92] D.D. Clark, S.J. Shenker, and L. Zhang. "Supporting Real-Time Applications in an Integrated Services Packet Network: Architecture and Mechanism". *Proc. of ACM SIGCOMM '92*, pages 14–26, Aug. 1992. **URL** ftp://parcftp.parc.xerox.com/pub/net-research/csz_sigcomm92.ps.

[DJ91] P.B. Danzig and S. Jamin. *tcplib: A Library of TCP Internetwork Traffic Characteristics*. Technical Report 91-495, USC, CS Dept., 1991. **URL** http://netweb.usc.edu/jamin/tcplib.

[DKPS95] M. Degermark, T. Köhler, S. Pink, and O. Schelén. "Advance Reservations for Predicted Service". *Proc. 5th Int'l Network and Operating Systems Support for Digial Audio and Video Workshop*, pages 3–14, Apr. 1995.

[DKS89] A. Demers, S. Keshav, and S.J. Shenker. "Analysis and Simulation of a Fair Queueing Algorithm". *Proc. of ACM SIGCOMM '89*, pages 1–12, Sept. 1989.

[DMRW94] D.E. Duffy, A.A. McIntosh, M. Rosenstein, and W. Willinger. "Statistical Analysis of CCSN/SS7 Traffic Data from Working CCS Subnetworks". *IEEE Journal of Selected Areas in Communication*, 12(3):544–551, Apr. 1994.

[Flo94] S. Floyd. Personal communication. Phone conversation, 1994.

[FV90] D. Ferrari and D.C. Verma. "A Scheme for Real-Time Channel Establishment in Wide-Area Networks". *IEEE Journal of Selected Areas in Communication*, 8(3):368–379, 1990.

[GAN91] R. Guérin, H. Ahmadi, and M. Naghshineh. "Equivalent Capacity and Its Application to Bandwidth Allocation in High-Speed Networks". *IEEE Journal of Selected Areas in Communication*, 9(7):968–981, Sept. 1991.

[GW94] M. Garrett and W. Willinger. "Analysis, Modeling and Generation of Self-Similar VBR Video Traffic". *Proc. of ACM SIGCOMM '94*, pages 269–279, Sept. 1994.

[Hir91] A. Hiramatsu. "Integration of ATM Call Admission Control and Link Capacity Control by Distributed Neural Network". *IEEE Journal of Selected Areas in Communication*, 9(7):1131–1138, Sept. 1991.

[HLP93] J.M. Hyman, A.A. Lazar, and G. Pacifici. "A Separation Principle Between Scheduling and Admission Control for Broadband Switching". *IEEE Journal of Selected Areas in Communication*, 11(4):605–616, May 1993.

[Hos84] J.R.M. Hosking. "Modeling Persistence in Hydrological Time Series Using Fractional Differencing. *Water Resources Research*, 20(12):1898–1908, Dec. 1984.

[HR89] J. Haslett and A.E. Raftery. "Space-time Modelling with Long-memory Dependence: Assessing Ireland's Wind Power Resource". *Applied Statistics*, 38(1):1–50, 1989.

[Hui88] J.Y. Hui. "Resource Allocation for Broadband Networks". *IEEE Journal of Selected Areas in Communication*, 6(9):1598–1608, Dec. 1988.

[Jai91] R. Jain. *The Art of Computer Systems Performance Analysis*. New York: John Wiley & Sons, Inc., 1991.

[Jam95] S. Jamin. *A Measurement-based Admission Control Algorithm for Integrated Services Packet Network*. Ph.D. Dissertation Proposal Excerpts. Technical Report USC-CS-95-617, Univ. of Southern California, CS Dept., 1995. **URL** http://netweb.usc.edu/jamin/admctl/quals-excerpts.ps.Z.

[JSZC92] S. Jamin, S.J. Shenker, L. Zhang, and D.D. Clark. "An Admission Control Algorithm for Predictive Real-Time Service (Extended Abstract)". *Proc. 3rd Int'l Network and Operating Systems Support for Digial Audio and Video Workshop*, Nov. 1992.
URL http://netweb.usc.edu/jamin/admctl/nossdav92.ps.Z.

[Kel91] F.P. Kelly. "Effective Bandwidths at Multi-Class Queues". *Queueing Systems*, 9:5–16, 1991.

[KM94] S.M. Klivansky and A. Mukherjee. On long-range dependence in nsfnet traffic. Technical Report GIT-CC-94-61, Gerogia Insititue of Technology, December 1994.

[LTWW94] W.E. Leland, M.S. Taqqu, W. Willinger, and D.V. Wilson. "On the Self-Similar Nature of Ethernet Traffic (Extended Version)". *ACM/IEEE Transactions on Networking*, 2(1):1–15, Feb. 1994.

[Mol27] E.C Molina. "Application of the Theory of Probability to Telephone Trunking Problems". *The Bell System Technical Journal*, 6:461–494, 1927.

[NK92] R. Nagarajan and J. Kurose. "On Defining, Computing, and Guaranteeing Quality-of-Service in High-Speed Networks". *Proc. of IEEE INFOCOM '92*, 1992.

[OON88] H. Ohnishi, T. Okada, and K. Noguchi. "Flow Control Schemes and Delay/Loss Tradeoff in ATM Networks". *IEEE Journal of Selected Areas in Communication*, 6(9):1609–1616, Dec. 1988.

[Par92] A.K. Parekh. *A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks*. PhD thesis, MIT, Lab. for Information and Decision Systems, Tech. Report LIDS-TR-2089 1992. Parts of this thesis were also published with R.G. Gallager in the *ACM/IEEE Transactions on Networking*, 1(3):344-357 and 2(2):137-150.

[PF94] V. Paxson and S. Floyd. "Wide-Area Traffic: The Failure of Poisson Modeling". *Proc. of ACM SIGCOMM '94*, pages 257–268, Aug, 1994. An extended version of this paper is available as **URL** ftp://ftp.ee.lbl.gov/papers/poisson.ps.Z.

[SCZ95] 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, 1995.

[SS91] H. Saito and K. Shiomoto. "Dynamic Call Admission Control in ATM Networks". *IEEE Journal of Selected Areas in Communication*, 9(7):982–989, Sept. 1991.

[VPV88] W. Verbiest, L. Pinnoo, and B. Voeten. "The Impact of the ATM Concept on Video Coding". *IEEE Journal of Selected Areas in Communication*, 6(9):1623–1632, Dec. 1988.

[Wil95] W. Willinger. Private communication. E-mail, 1995.

[Z+93] L. Zhang et al. *Resource ReSerVation Protocol (RSVP)* Internet-Draft. **URL** ftp://ds.internic.net/internet-drafts/, Oct. 1993.

[ZF94] H. Zhang and D. Ferrari. "Improving Utilization for Deterministic Service in Multimedia Communication". *IEEE International Conference on Multimedia Computing and Systems*, 1994.

[ZK94] H. Zhang and E.W. Knightly. "Providing End-to-End Statistical Performance Guarantee with Bounding Interval Dependent Stochastic Models". *Proc. of ACM SIGMETRICS'94*, pages 211–220, May. 1994.