2

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

Claudio Casetti
Politecnico di Torino
10129, Torino, Italy

Jim Kurose, Don Towsley
University of Massachusetts
Amherst, MA, 01002, USA

Abstract

The purpose of call admission control in Integrated Services Networks is to offer a guarantee that Quality of Service (QoS) bounds are not violated due to the admission of new calls into the network. This is typically accomplished using the declared worst-case traffic descriptors for incoming calls, a solution which results in poor bandwidth utilization. A measurement-based admission scheme is an appealing alternative: not only does it offer adaptivity to changing traffic conditions, it also allows statistical multiplexing gains to be exploited. In this paper, we examine the problem of determining which traffic characterization a measurement-based admission control algorithm should require of sources requesting access. Building on the work of Jamin et al. (1995), we also propose an adaptive measurement-based admission control algorithm that simplifies the estimation process, and show that it can achieve a high level of utilization without violating its delay-based QoS guarantees.

*This work was supported in part by the National Science Foundation under grants NCR-95-08274 and CDA-9502639

W. Dabbous et al. (eds.), *Protocols for High-Speed Networks V*

1 INTRODUCTION

The increasing bandwidths offered by today's high-speed switching and transmission technologies have enabled a new generation of real-time applications that require Quality of Service (QoS) guarantees from the network. As a result, numerous researchers – Braden et al. (1994), Clark et al. (1992), Ferrari et al. (1994), Floyd and Jacobson (1995), Hyman et al. (1991), Kurose (1993), Towsley (1993) and Shenker (1995a) – have advocated extending the Internet service model to accommodate the varying bandwidth, delay bound, and loss tolerance requirements of these applications.

One widely discussed network service model is one which provides "hard" guarantees to an application. In this case, an application is guaranteed a specified bandwidth, maximum end-to-end delay and no packet loss. While such a guaranteed service is appropriate for CBR-like applications with strict loss and delay requirements, it may not be appropriate for variable bit rate, adaptive, real-time applications, described in Clark et al. (1992), that can both tolerate and adapt to certain amounts of loss and/or end-to-end delay. Thus, additional service models that provide "softer" or *statistical* (rather than hard) guarantees have been proposed in Braden et al. (1994), Clark et al. (1992), Jamin et al. (1994.) and Wrockawski (1995). These alternate service models seek to provide such guarantees while still exploiting the advantage of statistical multiplexing at routers. Several recent IETF Internet Drafts, such as Breslau and Shenker (1995), Shenker (1995b), Shenker and Partridge (1995), Shenker, Partridge, Davie and Breslau (1995), lay the foundation for an Integrated Services Internet architecture based on such service models.

Of particular interest to us in this paper is the proposed Integrated Services Internet service class known as *predictive service* – see Braden et al. (1994) and Jamin et al. (1995). With predictive service, as well as with other service classes, such as the so-called "controlled-load service class" defined in Wrockawski (1995), *measurements* of ongoing admitted calls are used together with the declared traffic parameters of a call seeking admission to the network in making the call acceptance/rejection decision. The goal of this admission control process is to admit a call only if there are sufficient resources to satisfy the QoS requirements of the new call, while at the same time not violating QoS guarantees made to existing, already-admitted calls.

In this paper, we propose and evaluate a new measurement-based admission control algorithm. Our work extends existing work on measurement-based admission control, namely Clark et al. (1992) and Jamin et al. (1995), in three important directions:

- We examine the question of how best (from an admission control standpoint) to characterize a source among several "equivalent" leaky bucket source characterizations. We find that among equivalent characterizations, an admission control algorithm that uses a characterization with a smaller token bucket size and a higher token rate can often achieve an overall higher utilization than with other equivalent characterizations, without significantly increasing the percentage of packets experiencing a delay violation.
- We investigate which performance metrics should be measured. We show that measuring rate rather than delay provides us with more stable estimates and results in a simpler admission scheme.
- We consider how to adaptively adjust the length of the window of time over which measure-

ments are made. This obviates the need to choose, *a priori,* a length for a fixed measurement window. We show that our algorithm achieves a high utilization while being able to keep its delay commitments. Our study is chiefly conducted through simulation.

The remainder of this paper is organized as follows. In section 2, we provide a brief overview of past work in the area of measurement-based admission control, and summarize the specific algorithm proposed in Jamin et al. (1995), tailoring the description to the simple network setting we consider. Section 3 addresses the issues of source characterization and suggests that measuring sources' rates, rather than their packet delays, may be a better choice for a measurement-based admission control process. Our adaptive, window-based measurement and admission control algorithm is detailed in section 4. Section 5 presents simulation results of this algorithm under different traffic scenarios. Section 6 concludes the paper.

2 PREVIOUS WORK ON MEASUREMENT-BASED ADMISSION CONTROL

While the problem of admission control to a shared resource has been addressed by many researchers in recent years, the idea of using on-line measurements in the admission control process has been studied in only a few selected works.

Tedijanto and Gün (1993) propose an admission control mechanism based on the on-line estimation of the equivalent bandwidth of an ATM connection using a dynamically adjusted token bucket. Also in an ATM framework, but considering a superposition of sources rather than a single connection, the work by Dziong et al. (1995) employs a Kalman filter to estimate the aggregate effective bandwidth for admission control purposes. Vin et al. (1994) present an observation-based mechanism to estimate the retrieval time of a media block from a multimedia server, using such information to grant access to clients which requested predictive service. Measurement-based admission control for predictive service in Integrated Services Packet Networks was first introduced in Clark et al. (1992) and then further developed in Jamin et al. (1995), where a delay and rate measurement scheme for predictive service was proposed. As our work builds on Jamin et al. (1995), we next describe the admission scheme proposed in that paper. The interested reader is referred to Jamin et al. (1995) for further details.

Since our extensions will be considered in the context of a simpler network model than the one considered in Jamin et al. (1995), we first describe our network model.

2.1 Network Model

We consider the network scenario shown in Fig. 1, consisting of a link of capacity μ connecting two switches. Each switch is assumed to have an infinite buffer, so that packet loss probability is not a factor in our study. This allows us to focus on other QoS metrics such as delay and throughput.

Sources connect to the first switch through links of infinite bandwidth. The first switch concen-

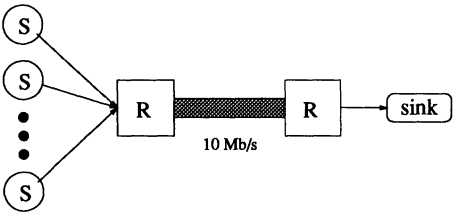

Figure 1 Network Scenario

trates (multiplexes) packets destined to the downstream switch. All traffic from the sources belongs to a single predictive traffic class. While there is no guaranteed-service traffic in our model, we note other traffic classes can be handled using such techniques as Weighted Fair Queueing – see Clark et al. (1992), Demers et al. (1989), Parekh and Gallager (1993) and Golestani (1994) – or class-based queueing as described in Floyd and Jacobson (1995).

2.2 Source Model

New calls arrive to the leftmost switch in Fig. 1 according to a Poisson process with rate r_s calls per second. Call holding times are independent and identically distributed exponential random variables with mean T_s. The traffic generated by an accepted call is modeled by an On/Off Markov-modulated fluid process. While in the On state, a source generates traffic at peak rate r; during the Off period no traffic is generated. In our simulations, unless otherwise noted, we use $\mu = 10Mb/s$, $r_s = 10calls/s$, and $T_s = 135s$.

On and Off periods are exponentially distributed with mean $1/\alpha = 0.3125s$ and $1/\beta = 0.3125s$ respectively. Depending on the scenario we are considering, the peak rates of admitted sources are either fixed (we refer to these as *homogeneous* sources) or exponentially distributed with mean $r = 64kb/s$ (*heterogeneous* sources). Our motivation for choosing fluid sources, rather than a discrete, packetized source model will be explained in section 3. We note that with the choice of the parameters above, a network without any admission control mechanism would achieve a near 100 % utilization.

2.3 The Admission Algorithm

We now describe the measurement-based admission algorithm proposed in Jamin et al. (1995), in the context of the simple network scenario described above. The key to the admission process is that the new call (or "flow" of data) should be accepted only if:

- the current traffic conditions at the switch can 'guarantee' that the call receives the QoS it requests. In the following we will precisely define what we mean by 'guarantee'.
- the acceptance of the new call does not cause QoS guarantees previously made to admitted flows to be violated.

To achieve these goals, a call seeking admission is required to provide a traffic characterization, known as a TSpec – see Shenker and Partridge (1995) and Wrockawski (1995).

Specifically, the call characterizes its traffic with two token bucket parameters: ρ (a token generation rate) and σ (a bucket depth) such that over a period of time T, the traffic generated by the call will not exceed $\rho T + \sigma$. We will address the problem of choosing appropriate values of ρ and σ in section 3.

The measured system quantities are:

- D, the delay experienced by traffic in the switch queue;
- ν, the aggregate traffic rate at the switch.

As we will see, as a result of the measurement process, at any given time the call admission algorithm will have an estimate of these two quantities, $\hat{D}$ and $\hat{\nu}$. Given these estimates, an arriving call is granted admission if the following two conditions hold – see Jamin et al. (1995):

1. Given a current delay estimate, $\hat{D}$, a new delay estimate, $\hat{D}$,' is formed such that $\hat{D}' = \hat{D} + \frac{\sigma}{\mu}$. Note that $\frac{\sigma}{\mu}$ is the amount of time to transmit a full token bucket's worth of data from the new call. This new delay estimate must satisfy:

$$D_{bound} > \hat{D}' = \hat{D} + \frac{\sigma}{\mu} \tag{1}$$

2. If the upper-bound rate declared by the new call, ρ, is added to the current estimate of the aggregate traffic rate, $\hat{\nu}$, the new estimate of the rate, $\hat{\nu}'$, satisfies:

$$v\mu > \hat{\nu}' = \hat{\nu} + \rho \tag{2}$$

the utilization bound, $v\mu$, is referred to as the *utilization target*.

If the new call is admitted, the new estimates for delay and rate are then taken to be $\hat{D}'$ and $\hat{\nu}'$. We now discuss how these values are subsequently updated.

2.4 The estimation process

In the following, we will find it useful to distinguish between "estimates" and "measurements" of delay or rate. "Measurements" are the observed delay values made over time, and "estimates" are constructed using these measurements, as well as the declared traffic parameters of arriving calls.

Delay measurements are collected for a fixed period of time known as the "measurement window." In the simplest case, at the end of a measurement window, an estimate is updated to be the *highest* observed value of that measure during the measurement window. In certain cases, however, the measurement window will be terminated prematurely and the estimates updated in a different manner. Specifically, as in Jamin et al. (1995):

- When a new call is admitted, the delay and rate estimates are updated using equations (1) and (2). It is worth noting that these estimates are *not* updated when a call terminates.
- When a single measurement *exceeds* the current estimate, the estimate is updated to be λ times the sampled value. The factor λ allows us to be more conservative by overestimating the quantity, resulting in more conservative admission control when the measure is close to the bound. In Jamin et al. (1995) it is suggested that $\lambda = 2$ be used for delay and $\lambda = 1$ for rate estimates. We will use these values.

In our fluid-flow simulation model, a single, 'instantaneous' delay measurement is computed by dividing the switch queue length at that time by the queue's service rate μ. Since there is no such thing as an 'instantaneous' rate measure, a small interval S is specified and a rate measurement is taken as the number of bits transmitted on the link during this interval. Jamin et al. (1995) suggest using $S = 500ms$.

It should be pointed out that the quality of the call admission scheme in Jamin et al. (1995) is highly dependent on the values of the so-called 'performance tuning knobs' (target utilization, λ, length of the measurement window). Selection of these values is an open problem which goes beyond the simple task of fine tuning the behavior of the admission scheme. Arguably, the most challenging issue is finding the 'right' length of the measurement window, since the performance of the admission scheme heavily relies on it. As noted in Jamin et al. (1995), too short a window lowers the admission threshold as soon as the traffic thins out, leaving the network dangerously exposed to bursts. On the other hand, long measurement windows lead to an excessively conservative scheme by delaying the time when flows' TSpecs (that give a worst case traffic characterization) are replaced with on-line measurements (that reflect that call's actually generated traffic). Perhaps the most important factor is that without an *a priori* knowledge of the traffic, it is difficult to determine an appropriate length of the measurement window. For these reasons, a measurement window which adapts to network traffic is desirable. We present such an adaptive algorithm in section 4. First we consider the problem of source characterization.

3 SOURCE CHARACTERIZATION AND MEASURED PARAMETERS

3.1 How should we characterize sources?

The admission control process discussed above uses token bucket parameters as a source traffic characterization.

In our simulations, we will model sources as Markov-modulated fluid sources whose smallest unit is the bit. By virtue of this choice, we are not committed to any packet size. However, tokens are representative of whole data units like cells and packets. We assume that a token is equivalent to a 1 kb data unit, as in Jamin et al. (1995).

Following Elwalid and Mitra (1991), we consider bufferless leaky-bucket-constrained Markov-modulated On/Off fluid sources with parameters (σ, ρ). But what values of (σ, ρ) should be

chosen? In Jamin et al. (1995), it is suggested that (σ, ρ) pairs be chosen empirically as the smallest σ and ρ that result in the desired loss rate at the token bucket. Of course, such strategy cannot be employed if the traffic characteristics of the ON/OFF sources have not been evaluated in advance. Using equations (2.18), (2.21) and (8.8) in Elwalid and Mitra (1991), we can derive the following expression relating the bucket depth, σ, the token rate, ρ, the source peak rate, r, and the source's fluid loss probability, π at the leaky bucket:

$$\sigma = -(r-\rho)\frac{ln(1 - \frac{\alpha r}{(\alpha+\beta)\rho\beta\pi} + \frac{\alpha(r-\rho)}{\beta\rho})}{(\alpha+\beta)(1 - \frac{\alpha r}{(\alpha+\beta)\rho})} \tag{3}$$

For a given bucket size, eq. (3) can be used to obtain the theoretical value of the token rate which results in a target loss probability of π at a bufferless token bucket. We chose $\pi = 10^{-5}$. We will refer to (σ, ρ) pairs with the same target loss probability as *equivalent*.

Given the set of equivalent source characterizations satisfying eq. (3), which source characterization should we choose? Ideally, we would like to choose a characterization that, when coupled with the admission control process described in section 2, results in high link utilization at the switch without excessive QoS violations. In order to explore the impact of using different "equivalent" source characterizations, we ran simulations using the call admission algorithm and the homogeneous source model described in section 2. Delay and rate estimates were computed using the procedure outlined above, with a delay bound of $16ms$ and a utilization target of 95 % (which yields a rate bound of 9.5 Mb/s).

Tables 1 and 2 summarize the results we obtained for measurement window lengths of 1 and 10 seconds, respectively. Simulations were run for a total of 10000 seconds of simulated time each, which is at least 3 orders of magnitude longer than the measurement window dynamics. The computed utilization values were such that the width of the 99.8% confidence interval was within 1% of the point estimate; the percentage of late bits showed significantly more variability. Each row in a table shows for an equivalent (σ, ρ) pair,

- the resulting utilization of the switch output link;
- the percentage of bits that violated the delay bound;
- the maximum delay in ms observed during the simulation (i.e. the maximum queue length divided by the service rate);
- the percentage of flows, among those rejected, that were rejected due to *delay* constraint violations (see eq. (1));
- the percentage of flows, among those rejected, that were rejected due to *target utilization* constraint violations (see eq. (2)).

Thus, for example, the first row of Table 2 indicates that among those flows rejected, 3.74 % violated the delay constraint and 99.76 % violated the target rate constraint. Note that the percentages do not add up to 100 %, since there was a portion of flows that violated *both* bounds,

Several observations are in order. Comparing Tables 1 and 2 indicates that a longer mea-

σ	ρ	% Utilization	% Late Bits	Max. Obs. Delay	% Delay Rej.	% Rate Rej.
1	64	74.53	0.0	0.334	0.0	100.00
4	62.8	74.46	0.0	0.0	0.73	99.27
10	61.1	55.35	0.0	0.0	98.28	1.72
20	58.8	32.24	0.0	0.0	100.00	0.00

Table 1 Equivalent token buckets with measurement window = 10 s

σ	ρ	% Utilization	% Late Bits	Max. Obs. Delay	% Delay Rej.	% Rate Rej.
1	64	91.30	0.37	65.32	3.74	99.76
4	62.8	91.25	0.35	41.55	4.12	99.67
10	61.1	91.25	0.29	39.40	6.53	98.54
20	58.8	90.70	0.18	43.15	23.60	86.07

Table 2 Equivalent token buckets with measurement window = 1 s

surement window results in more conservative admission control: with a 10 second window, no late bits were observed, while with a smaller window (1*s*), there was a small amount of delay violation. However, the achieved link utilization is much higher in the case of the small window.

Interesting observations can also be made regarding the different bucket sizes used in the source characterization. The larger the bucket declared by the source, the more likely it is that a flow requesting admission is denied access on the grounds that it might violate the delay bound. This is because a larger σ in (1) provides a higher delay estimate. However, with a large σ, a number of these flows would appear to have been unnecessarily rejected, since none of the admitted flows ever experienced a delay violation for any of the values of σ shown. We observed a similar behavior for other window lengths, as well as with bucket sizes larger than 20. We also observed similar behavior with heterogeneous sources, where only the bucket size is fixed and the token rate is computed on-line for each newly generated call with random peak rate, using eq. (3). We omit those results for reasons of space. Given these observations regarding equivalent token bucket characterizations, for the remainder of this paper we will use $\sigma = 1$, which results in a value of ρ equal to a source's peak rate.

3.2 Which performance measures should be observed?

The admission scheme described above makes use of both delay and rate measures. An adaptive window algorithm would be noticeably simpler if it had to rely on one parameter only.

Tables 1 and 2 show that, especially for small values of σ, the large majority of rejected calls

Window Length [s]	Avg. Delay	Avg. Rate	Delay Variance	Rate Variance
1	9.51	9.24	543.8	1.55
3	0.48	8.49	7.994	1.33
5	0.056	8.01	0.705	1.22
10	0.0371	6.99	0.493	1.12

Table 3 Average and variance of delay and rate measure

are rejected on the basis that they violate the rate criteria (2). An argument for using rate measurements rather than delay measurements is provided by a simulation experiment in which we evaluated the mean and the variance of both delay and rate measures (not estimates) using the window length as a parameter. Our purpose in doing so is to determine which is a more "stable" estimate. The simulation scenario is essentially the same as in the previous section. However, rather than running fixed-length simulations, we used the *batch means* approach in Fishman (1991) as a stopping criteria. The simulations were run until the width of the 90% confidence interval was within 10% of the point estimate.

Table 3 shows the mean and variance estimates for different measurement window sizes, for the case that the admission control algorithm had a utilization target of 100 %. The average delay is in ms, its variance in ms^2, while the average rate is in Mb/s and its variance in $(Mb/s)^2$. Somewhat expectedly, these results discourage the use of delay estimates. Not only is the variance extremely high for short windows, but is also very sensitive to the window length. Also, note the impact of varying measurement window lengths on delay and rate average. While the rate merely shows a 25 % decrease over the range of window sizes considered, the average delay values differ by 2–3 orders of magnitude.

The adaptive measurement-based admission control algorithm we present in the following section thus only uses rate measurements. We note that this also fits well with the IETF specification of the Controlled Load Service in Wrockawski (1995), in that it does not make use of target values for such direct performance measures (e.g., delay or loss) but rather ensures that adequate bandwidth is available to handle the declared traffic rate.

4 A NEW MEASUREMENT-BASED ADMISSION ALGORITHM

In the previous sections we have discussed several aspects of measurement-based call admission control. We enumerate these, and other considerations:

Proposition 1 *A measurement-based admission algorithm should:*

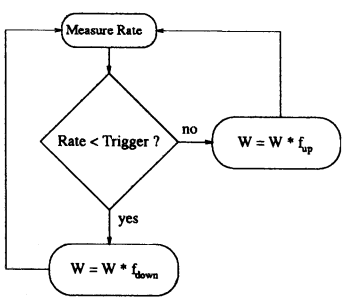

Figure 2 First-level algorithm

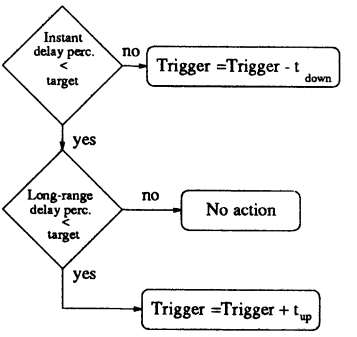

Figure 3 Second-level algorithm

- *avoid using fixed-length measurement windows, on the grounds that the traffic characteristics are usually unknown, or can vary;*
- *adapt to changing traffic conditions, enlarging or shrinking its measurement window so as to realize a more or less conservative admission process;*
- *require sources to characterize their traffic using only the peak rate;*
- *rely upon rate, rather than delay measurements. We note, however, that delay measurements could be used in conjunction with rate measurements.*

The adaptive measurement-based call admission algorithm described below is divided into two "levels". The *first-level algorithm*, which in essence forms the core of the algorithm, uses rate measurements and the declared traffic parameters of accepted calls to adaptively adjust the length of the measurement window. The second-level algorithm adjusts the values of parameters used by the first level algorithm.

The idea behind the first-level algorithm is quite simple. The algorithm continually shrinks the length of the measurement window (resulting in admission control decisions that work towards increasing link utilizations; see Tables 1 and 2, and the discussion in Jamin et al. (1995)) until the amount of traffic generated by accepted calls reaches a *trigger* value. This trigger value is actually a rate value, smaller than the output link capacity, providing an early warning that the system is about to reach a load level where it may be requested that some form of control be implemented (this does not necessarily mean that we are approaching overload, see Wrockawski (1995) for further discussion). The first-level algorithm then reacts by enlarging the measurement window until the measured rate drops below the trigger, at which point the window can be shrunk again. In a sense, the admission control algorithm operates like TCP congestion control – it tries to be more and more aggressive in admitting calls until something "bad" happens. At that point it backs off (i.e., it shrinks the window, resulting in a stricter call admission process) but after backing off again begins to decrease the measurement window.

Fig. 2 illustrates the first level algorithm. The variables used are:

- W, length of the measurement window. In our simulations, we do not allow the measurement window to become smaller than 1 second.
- f_{up}, window enlargement factor
- f_{down}, window shrinking factor

We will refer to f_{down} and f_{up} as *window shaping factors.*

Although the algorithm does away with the "performance tuning knobs" and fixed length window in the original algorithm (see Jamin et al. (1995) and section 2), it appears that we have traded these for equally uncertain parameters: the trigger value and the shaping factors. The second-level algorithm addresses this issue.

Fig. 3 illustrates the second-level algorithm. It lowers and raises the trigger in search of a good operating point, in response to traffic fluctuations. The algorithm reacts to delay bounds violations while cushioning the negative effects of the delay variance. Two delay-related measures are used. The first delay measure is the *instantaneous delay violation percentage;* it provides a warning that a percentage of bits in the switch's queue are violating the delay bounds (thus it is easily implementable). The second delay measure is the *"long term delay violation percentage";* it is the percentage of late bits observed since time 0. We stress that this is only one of the possible solutions: an alternative scheme might choose to address the instantaneous delay alone rather than also including the long term percentage.

The algorithm operates as follows: at time 0 we set a *target* value for the long term delay violation percentage, which we strive to guarantee over n hours (the *target commitment period*). On resetting the measurement window, whenever the first level algorithm attempts to change the window length, it checks whether the instantaneous delay percentage is higher than the target delay percentage. If so, the admission algorithm has been too aggressive, therefore the trigger rate value is lowered, resulting in more conservative admission control. If the instantaneous delay percentage does not exceed the target, the algorithm checks whether the long term delay violation percentage is currently under the target value. If it is, the window trigger value is *raised.* If it is not, no action is performed. This last choice is motivated by the fact that if the instantaneous target delay is not violated, the trigger is sufficiently low and the overall delay only needs more time to settle under the target. In order to lessen the scheme's sensitivity to delay variations, we suggest additive rather than multiplicative trigger increments. Of course, the trigger cannot be higher that the rate bound, i.e., the output link capacity. We will indicate trigger increments by t_{up} and t_{down}.

5 SIMULATION RESULTS

We now evaluate the proposed algorithm under different traffic scenarios. The first scenario uses the source model described in section 3.1. The algorithm's parameters are:

- $f_{up} = 1.2$
- $f_{down} = 0.9$

- t_{up} : $0.01 Mb/s$
- t_{down} : $0.05 Mb/s$

In our first set of results, the target delay percentage was guaranteed over a period of 12 hours, and a delay bound of 16 ms was chosen. Ideally, a 12-hour period would allow network administrators to alternate between more and less stringent commitments depending on the portion of the day when the network is likely to be more congested. The trigger was initially set at $9.0 Mb/s$.

Tables 4 and 5 show the performance of the two-level algorithm using different target values for the long term delay violation percentage, ranging from 0.01% to 2%, for both homogeneous and heterogeneous sources. As expected, as the target (allowable) percentage of delay violations increases, so too does the achieved utilization. Note that both types of sources adhere to the guaranteed target, i.e., the percentage of late bits observed (column 4 in the tables) closely matches the long term delay violation target values shown in column 1. We also observe that homogeneous sources achieve a higher utilization than heterogeneous sources. This is because their fixed rate yields more 'predictable' behavior, and increased predictability accounts for more reliable estimates at the time of admission. This, in turn, yields fewer violations and the opportunity of better exploiting the available bandwidth, thus increasing utilization.

Figures 4 and 5 provide a useful insight into the algorithm. Fig. 4 plots the percentage of late bits since time 0 for three different values of the target long term delay violation percentage (1%, 0.1% and 0.01%) as a function of time for heterogeneous sources. For each curve, we can identify two critical regions in Fig. 4: a *transient* showing a peak and a curve converging to the target value, and a *steady state* region capturing the algorithm's effort to preserve the target percentage. The transient peak is generated by the lack of information available to the network at the beginning of each target commitment period, resulting in the algorithm being unable to find a stable operating point. Whereas both the 1% and the 0.1% case reach the steady state long before the commitment period has expired, the 0.01% target curve can not reach the steady state regime within the time period studied due to the high initial transient peak. Actions that might reduce it include setting a lower starting value of the trigger or initializing the window to several hundred seconds.

The trigger value dynamics for the three curves from Fig. 4 are shown in Fig. 5. It is interesting to observe how trigger peaks are matched to regions where the percentage of late bits is *below* the target (this is particularly evident for the 0.1% case at approximately times 13000, 23000, 32000 and 41000 seconds). Indeed, the algorithm raises the trigger each time the target value is reached.

Thus far, our simulation parameters resulted in a network operating in overload. In the next set of results, we investigate the algorithm's response to traffic fluctuations. Specifically, the call arrival process was modified so that every 100 seconds on average (values were actually determined as independent identically distributed random variables with mean 100), the call arrival rate was switched between 10 sources per second with probability 2/3, and 0.01 sources per second with probability 1/3. Heterogeneous sources were used. Although the ensuing traffic pattern is not meant to model actual network traffic, it is nevertheless interesting to test how the

Target [%]	Avg. window size	% of adm. flows	% of late bits	Utilization [%]
0.01	4.71	25.3	0.0122	85.85
0.05	3.94	25.7	0.0530	87.19
0.10	3.95	26.1	0.1201	86.94
0.20	3.48	25.9	0.2058	88.06
1.00	2.19	27.4	1.0139	91.00
2.00	1.72	27.7	2.0060	92.36

Table 4 Two-level algorithm - homogeneous sources

Target [%]	Avg. window size	% of adm. flows	% of late bits	Utilization [%]
0.01	7.15	21.3	0.0232	74.86
0.05	6.00	22.5	0.0498	77.32
0.10	5.65	22.7	0.1048	78.05
0.20	4.65	23.4	0.2080	80.82
1.00	3.30	24.4	0.9988	84.51
2.00	2.72	25.8	1.9809	86.59

Table 5 Two-level algorithm - heterogeneous sources

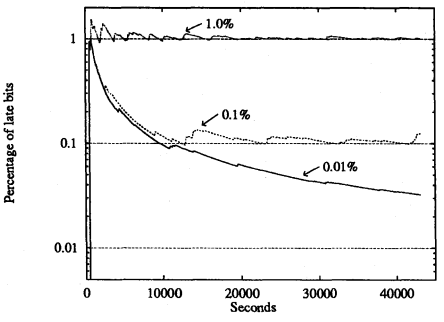

Figure 4 Percentage of overall late bits, as a function of time, for different delay targets - heterogeneous sources

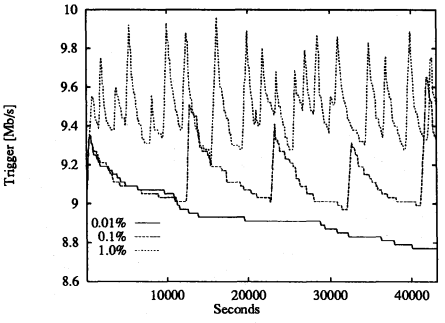

Figure 5 Trigger values as a function of time, for different delay targets - heterogeneous sources

Target [%]	Avg. window size	% of adm. flows	% of late bits	Rel. Util. [%]
0.01	3.25	22.1	0.0708	45.67
0.05	2.97	24.0	0.1076	57.08
0.10	2.50	26.5	0.2923	48.29
0.20	2.00	30.2	0.2344	59.71
1.00	1.51	34.6	1.0142	67.77
2.00	1.40	35.3	2.0171	68.63

Table 6 Two-level algorithm - sensitivity to traffic fluctuations

algorithm handles this rather extremal situation where prolonged underload periods are followed by sharp traffic peaks. The trigger was initialized at $7Mb/s$ to reduce the transient peak.

Using the same algorithm parameters listed above, we obtained the results in Table 6. Fig. 6 plots the number of admitted and active sources during the simulation (to improve readability, only the first 10000 seconds are shown). Note that the algorithm fails to meet the target value of the long term delay violation percentage when the target is smaller that 0.1%. For this case of a non-stationary call arrival process, we show a relative measure of utilization (the ratio between the utilization which could be achieved if no admission control were enforced (approx. 77 %), and the utilization using admission control). Of course, the closer the *relative utilization* is to 100 %, the lesser the impact of admission control on link utilization.

So far, we have taken for granted that our values for the window shaping factors and the target increments were good ones. Fig. 7 gives an indication of how sensitive our algorithm is to the choice of t_{up} and t_{down}. The figure shows the percentage of late bits with a target value of 0.05 % for the long term delay violation percentage, heterogeneous sources and constant overload (i.e. no traffic fluctuations). The increments are expressed in kb/s and range from 1 to 1000. It can be seen that we should take extra care in choosing the values of t_{down}. Also, very low decrements appear to have little effects on keeping the percentage of late bits close to the target, unless paired with equally low increments. We have observed a similar behavior for other scenarios as well, confirming a simple "rule of thumb" – that decrements should be higher than increments. While the results so far have given us an indication as to what range these values should be chosen from, a robust procedure for picking these values remains a topic for future research.

6 CONCLUSIONS AND FUTURE WORK

The issue of measurement-based admission control was the topic of this paper. Through the use of simulation, we have addressed the problem of what source characterization and what measured quantities best suit a measurement-based admission control scheme for predictive service. This enabled us to devise a simple admission algorithm, relying on rate measures only and using only

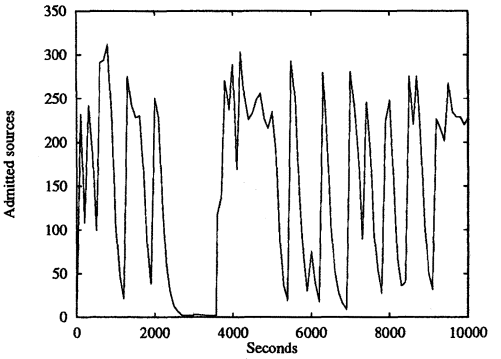

Figure 6 Two-level algorithm - Admitted and active sources as a function of simulated time - heterogeneous sources

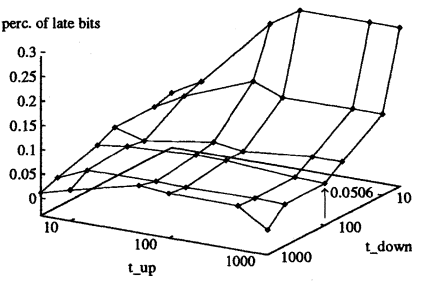

Figure 7 Percentage of overall late bits as a function of trigger increments and decrements - heterogeneous sources

the source peak rate as TSpec. A delay bounding scheme was used together with the admission algorithm to provide delay guarantees over a period of time. The analysis of our results have shown the algorithm and the bounding scheme to perform well under overload conditions, while extremal traffic fluctuation have proved to be more difficult to handle, particularly under tight guarantees.

Our ongoing work is investigating the effect of traffic fluctuations and more thoroughly exploring the algorithm's sensitivity to its tuning parameters. We are also studying the algorithm under empirical traffic patterns, such as video traces.

Acknowledgement

The authors would like to thank Francesco Lo Presti for many valuable discussions on the subject of fluid models and admission control.

REFERENCES

R. Braden, D. Clark, S. Shenker, *Integrated Services in the Internet Architecture: an Overview*, RFC-1633, June 1994.

L. Breslau, S. Shenker, *A Proposal for Accomodating Heterogeneity*, Internet Draft, November 1995, `draft-ietf-intserv-hetero-01.txt`.

D. Clark, S. Shenker, L. Zhang, *Supporting Real-Time Applications in an Integrated Packet Services Network*, SIGCOMM'92, Baltimore, MD, USA, August 1992.

A. Demers, S. Keshav, S. Shenker, *Analysis and Simulation of a Fair Queueing Algorithm*, SIGCOMM'89, Austin, TX, USA, September 1989.

Z. Dziong, B. Shukhman, L. G. Mason, *Estimation of Aggregate Effective Bandwidth for Traffic Admission in ATM Networks*, INFOCOM'95, Boston, MA, USA, May 1995.

A. I. Elwalid and D. Mitra, *Analysis and Design of Rate-based Congestion Control of High Speed Networks, I: Stochastic Fluid Models, Access Regulation*, Queueing Systems, Vol 9, pp. 29–64, October 1991.

D. Ferrari, A. Banerjea, H. Zhang, *Network Support for Multimedia: A Discussion of the Tenet Approach*, Computer Networks and ISDN Systems, Vol. 10, pp. 1267–1280, July 1994.

G. S. Fishman, *Principles of Discrete Event Simulation*, John Wiley & Sons, 1991.

S. Floyd and V. Jacobson, *Link-sharing and Resource Management Models for Packet Networks*, IEEE/ACM Transactions on Networking, Vol. 3, No. 4, pp. 365–386, August 1995.

S. Golestani, *A Self-clocked Fair Queueing Scheme for Broadband Applications*, INFOCOM'94, Toronto, Canada, June 1994.

J. Hyman, A. A. Lazar, G. Pacifici, *Joint Scheduling with Quality of Service Constraints*, IEEE Journal on Selected Areas in Communications, vol. 9, no. 7, pp. 1052–1063, September 1991.

S. Jamin, P. B. Danzig, S. Shenker and L. Zhang, *A Measurement-based Admission Control Algorithm for Integrated Services Packet Network*, SIGCOMM'95, Cambridge, MA, USA, August 1995.

J. Kurose, *Open Issues and Challenges in Providing Quality of Service Guarantees in High Speed Networks*, ACM Computer Communication Review, vol. 23, no. 1, pp. 6–15, January 1993.

A. K. Parekh, R. G. Gallager, *A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Single Node Case*, IEEE/ACM Transactions on Networking, vol. 1, pp. 344—357, June 1993.

S. Shenker, *Fundamental Design Issues for the Future Internet*, IEEE Journal on Selected Areas in Communications, September 1995.

S. Shenker, *Design Notes on the Specification of Services*, Internet Draft, November 1995.

S. Shenker, C. Partridge, *Specification of Guaranteed Quality of Service*, Internet Draft, November 1995, `draft-ietf-intserv-guaranteed-svc-03.txt`.

S. Shenker, C. Partridge, B. Davie, L. Breslau, *Specification of Guaranteed Quality of Service*, Internet Draft, 1995, `draft-ietf-intserv-predictive-svc-01.txt`.

T.E.Tedijanto, L. Gün, *Effectiveness of Dynamic Bandwidth Management Mechanisms in ATM Networks*, INFOCOM'93, San Francisco, CA, USA, March 1993.

D. Towsley, *Providing Quality of Service in Packet Switched Networks*, Performance Evaluation of Computer and Communication Systems, (ed. L. Donatiello and R. Nelson), Springer–Verlag, pp. 560–586, 1993.

H. M. Vin, A. Goyal, A. Goyal, P. Goyal, *An Observation-Based Admission Control Algorithm for Multimedia Servers*, ICMCS'94, Boston, MA, USA, April 1994.

J. Wrockawski, *Specification of the Controlled-Load Network Element Service*, Internet Draft, December 1995, `draft-ietf-intserv-strl-load-svc-01.txt`.