

Quality of service and flow level admission control in the Internet*

N. Benameur, S. Ben Fredj, S. Oueslati-Boulahia, J. W. Roberts

France Telecom R&D

Tel : (33) 1 45 29 57 01, Fax : (33) 1 45 29 60 69

{nabil.benameur,slim.benfredj, sara.boulahia, james.roberts}@francetelecom.com

Abstract

We propose to apply an integrated admission control scheme to both streaming flows and elastic flows. It is assumed that streaming flow packets are served with priority in network queues so that admission control ensures minimal delay for audio and video applications while preserving the throughput of document transfers. An implicit measurement-based implementation is proposed where admissibility is based on an estimation of the bandwidth a new elastic flow would acquire. An analytical fluid flow model provides insight and guides the choice of admission thresholds. Detailed packet level simulations of TCP and UDP connections show that the proposed algorithms work satisfactorily in the range of admission thresholds predicted by the fluid model.

Key words: Admission control, quality of service, streaming traffic, elastic traffic, flow aware networking.

1 Introduction

Applications expected to produce the bulk of traffic in the future multiservice Internet can be broadly categorized as streaming or elastic according to the nature of the flows they produce. Streaming flows are produced by audio and video applications. Elastic flows, on the other hand, result from the transfer of digital documents (Web pages, files, MP3 tracks,...) using a transport protocol like TCP. The Diffserv network architecture goes some way towards meeting the distinct quality of service requirements of these two types of traffic. However, it remains unclear how this architecture can be used by network providers to offer useful services with meaningful end-to-end guarantees. In this paper we argue that an essential network function is missing: this function is admission control applied to both streaming and elastic flows.

Admission control has been studied mainly in the context of streaming traffic with new flows being rejected if the addition of their traffic would lead to quality degradation for ongoing flows. It has traditionally relied on signalling and explicit resource reservation. The use of admission control for elastic traffic has been proposed more recently with the objective of avoiding unnecessary loss of performance and efficiency in case of demand overload [2, 3, 4]. In contrast with previous studies, which have addressed the issue of admission control either for streaming traffic alone or for elastic traffic alone, in this paper we envisage an integrated network where streaming and elastic flows share the same links and admission control applies to both. Packets of streaming flows have priority in network queues to minimize their delay while elastic flows dynamically share the remaining bandwidth.

*A short version of this paper was presented in the QoS workshop, Coimbra, Portugal, September 2001 [1]

The envisaged admission control is *implicit*: new flows are identified “on the fly” and if a flow must be rejected, its packets are simply discarded. Avoiding the need for signalling and explicit resource reservation considerably simplifies implementation which can be introduced progressively with minimal need for standardization. We propose a measurement-based implementation where admissibility is based on an estimation of the bandwidth which a new elastic flow would acquire. We investigate two alternative criteria for deriving this: the measured rate of a permanent “phantom” connection, and the current packet loss rate experienced by traffic on the considered link or path. The corresponding admission control algorithms were introduced in an earlier work and tested by simulation assuming all traffic is elastic [5]. In this paper, we reexamine these algorithms with the objective of investigating the impact of streaming traffic on their efficiency. To gain insight into their performance in this context and to guide our choice of admission threshold, we develop a fluid model of statistical bandwidth sharing accounting for both classes of flow.

The rest of the paper is organized as follows. We first discuss in Section 2 traffic characteristics of elastic and streaming flows. Section 3 outlines their impact on performance and argues in favour of applying admission control to both types of traffic. Section 4 describes the integrated network architecture we envisage. Section 5 presents the proposed fluid model and its analysis. Results of the simulation of the proposed admission control algorithms are presented in Section 6. Conclusions are drawn in Section 7.

2 Flow level traffic characterization

For the purposes of traffic engineering and control it is most convenient to characterize demand at flow level. While in practice there may arise a certain ambiguity in the definition of what constitutes a flow (an entire Web page, a single element or image in that page or the TCP connection transferring a group of page elements, for example), for present purposes we assume that a flow can be characterized as a sequence of packets having the same identifier transmitted with an inter-packet interval smaller than some threshold (a few seconds, say). Flows may be broadly divided into two categories: elastic and streaming.

2.1 Elastic flows

Elastic flows correspond to the transfer of digital documents (Web page, file, MP3 track,...) and adapt their rate to available capacity. This adaptation is typically achieved by a transport protocol like TCP. An elastic flow is simply characterized by the size of the document to be transferred. The time required to transfer the document depends on the number of ongoing flows on all routes and constitutes the main performance criterion for an elastic flow. In this paper we seek to evaluate statistics of the transfer time (or equivalently, the realized throughput) by considering the random fluctuations of traffic at flow level. We account for the packet level transport protocol only indirectly through the effect it has on flow level bandwidth sharing.

A simple model of elastic traffic at flow level is to assume flows have a size drawn independently from a certain distribution. From the result of measurements, a candidate distribution would have a heavy tail: most flows are very small (so-called “mice”) while the majority of traffic is contained in very long flows (so called “elephants”). The high variability of the size distribution is known to cause the well-known self-similarity of IP traffic at packet level.

For the sake of simplicity, we further assume that flows arrive according to a Poisson process. This assumption is sufficient for present purposes where we aim to illustrate the advantages of admission control and to demonstrate the feasibility of a measurement-based implementation.

Furthermore, it has been shown that performance results derived for this simple model hold under the much more general and realistic assumption that *sessions* arrive as a Poisson process, the flow arrival process within a session being quite arbitrary [6].

2.2 Streaming flows

Streaming flows are produced by audio and video applications and have an intrinsic rate which must be preserved by the network. The essential traffic characteristics of streaming flows are their duration and rate.

The stochastic nature of the particular kind of streaming traffic constituted by telephone calls is well known. Modeling call arrivals as a Poisson process has sound empirical and theoretical foundations. The duration of telephone calls is, like Web document sizes, extremely variable. It can, for example, be modelled by a log-normal distribution [7]. In this paper we assume these modelling assumptions can be carried over to other forms of streaming traffic. We note, however, that a Poisson arrivals assumption may not always be appropriate, in the case of a videoconferencing application, for example.

While certain streaming applications produce constant rate flows, most audio and video applications have variable rate. Variable rate video coding in particular produces extreme rate variations at multiple time scales [8]. Such flows are intrinsically self-similar at packet level.

Although the QoS of streaming traffic is frequently expressed at packet level (i.e., packet end-to-end delay and jitter), it is relevant to consider a flow level model for such traffic. Indeed, it has been shown in [9] that statistical bounds on end-to-end packet delay and jitter can be derived knowing the load induced by streaming traffic, the number of hops traversed and provided that simple engineering rules are enforced (mainly, shaping to a peak rate at ingress routers and non-preemptive priority queuing for streaming traffic). Consequently, by applying (admission) controls at flow level to ensure that the load induced by streaming traffic does not exceed a certain threshold, it is possible to provide statistical performance bounds at packet level.

2.3 Relative demand

Elastic traffic presently constitutes the majority of Internet traffic. Measurements consistently show that TCP connections represent some 80% of IP flows and is responsible for more than 95% of the traffic demand in bytes [10, 11]. These proportions have remained steady over the past 10 years. New streaming applications like voice over IP or video on demand may change these proportions. However, since most audio and video traffic is transferred for local playback using the new peer to peer protocols like Napster, it is likely that elastic traffic will remain dominant.

3 Impact on performance of traffic characteristics

It is important in designing a QoS capable network architecture to fully understand the impact of traffic characteristics on realized performance. In this section we outline the results of mathematical performance models for elastic and streaming traffic and discuss their interaction in an integrated network.

3.1 Elastic traffic performance

Elastic traffic is, by definition, suited to the use of closed-loop control whereby the flow rate is continuously adjusted to make maximal use of available bandwidth. In the present Internet,

elastic flows share bandwidth dynamically under the control of TCP. The generally assumed objective is that bandwidth is shared as fairly as possible between contending flows. The degree of fairness achieved by TCP is variable depending on many factors including the connection round trip time and the maximum window size. The bandwidth achieved by a flow also depends on its size, the throughput of small flows (mice) in particular being severely limited by the slow start algorithm. The bandwidth share acquired by a flow in a network is a complex function of the number of flows on all routes [12]. For present purposes, however, we make a number of simplifying assumptions about the way bandwidth is shared and limit attention to an isolated bottleneck link.

Specifically we assume the bottleneck bandwidth is shared perfectly fairly with instant readjustment whenever new flows begin or existing ones end. Note, however, that detailed packet level simulations of TCP connections are used in Section 6 to validate the concept of measurement-based admission control.

With the assumed traffic model, the shared link behaves like an M/G/1 processor sharing queue [13]. Let the flow arrival intensity be λ flows/sec, the mean flow size σ bits, the link capacity C bits/sec and denote by ρ the link load $\lambda\sigma/C$. Assuming $\rho < 1$, the distribution of the number of flows in progress is geometric:

$$Pr[\text{flows} = n] = \rho^n(1 - \rho), \quad (1)$$

and the expected duration $R(s)$ of a flow of size s is:

$$R(s) = \frac{s}{C(1 - \rho)}. \quad (2)$$

The ratio $s/R(s) = C(1 - \rho)$ constitutes a useful size-independent measure of flow throughput performance. As long as ρ is not too close to 1, throughput is generally satisfactory. In practice, for most shared links of reasonably high capacity, $C(1 - \rho)$ is typically much greater than rate limitations due to causes not considered here (the user's modem, the server, the TCP maximum receive window,...). Such links are thus virtually transparent with respect to their impact on perceived throughput performance. Furthermore, in this stable load regime, performance is *insensitive* to the flow size distribution.

The above results clearly do not apply in a situation of overload with $\rho > 1$. The simple processor sharing model would then be unstable with the number of flows in progress increasing indefinitely while the bandwidth share of each flow tends to zero. In practice when demand exceeds capacity the number of flows in progress does not increase indefinitely. As their bandwidth share diminishes, some flows will be interrupted due to user impatience or aborts triggered by TCP or higher layer protocols.

Results of simulations and mathematical models of an overloaded link show that performance depends significantly on traffic characteristics [14, 15]. Performance tends to improve as the variability of flow sizes increases, for example, since the impact of overload is experienced mainly by the largest flows (the "elephants") which abort leaving sufficient capacity for the majority of small flows (the "mice"). Figure 1 shows results from a simulation of an overloaded link. It shows how the number of ongoing TCP connections increases in time in the absence of any aborts. The upper trace relates to all flows, whereas the lower plot shows the number of mice alone. By mice we mean the smallest 90% of flows¹. We observe that the number of ongoing mice increases much more slowly than the overall number of flows. The larger flows are thus more susceptible to aborts.

¹The precise distribution of flow size used in this simulation is described in Section 6.

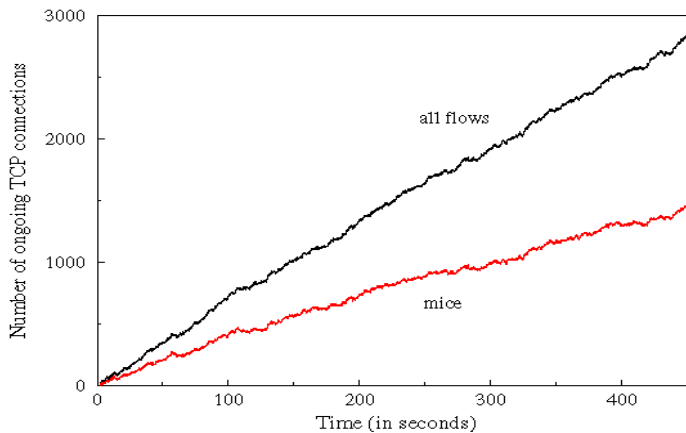


Figure 1: Accumulation of ongoing TCP connections in time on a congested link of 10 Mbps, load=1.4

Realized throughput of all completed flows is uniformly low and link bandwidth is wasted transmitting the first part of the transfers which are eventually aborted. Performance is further degraded when users later reattempt these aborted transfers. To preserve goodput, to ensure adequate flow throughput and to avoid possible propagation of congestion due to the retransmission of discarded packets, it is necessary to perform admission control, i.e., to accept a new flow only if its throughput would be sufficient [14].

The primary reason for controlling the admission of elastic flows is thus not to provide guaranteed minimum throughput although this is certainly a useful consequence. The main reason is to avoid the negative impact of congestion in overload. The admissibility condition should be such that blocking is negligible in normal load ($\rho < 1$) but sufficiently reactive to prevent congestion collapse in overload ($\rho \geq 1$). It turns out that the precise choice of admission criterion is not highly critical. It is perfectly efficient to realize a minimum throughput typically much higher than what might reasonably be considered as acceptable for most applications [5].

3.2 Streaming traffic performance

We assume streaming flows are subject to open-loop control implying the notion of “traffic contract”:

- each flow must declare its traffic characteristics;
- the network performs admission control to ensure that the admission of the flow will not lead to quality degradation;
- if admitted the traffic of the flow is policed to ensure that its traffic characteristics indeed conform to what was declared.

This type of multiplexing has been abundantly studied and the available control options are well known. It is important to distinguish between statistical multiplexing schemes according to whether or not they rely on significant buffering to absorb rate variations [16].

When a sufficiently large buffer is provided, the combined input rate of multiplexed flows can momentarily exceed link capacity with limited loss and packet delays compatible with the QoS requirements of streaming applications. A major problem with this multiplexing scheme,

however, is that performance depends significantly on detailed traffic characteristics which are generally unknown *a priori* and are impossible to police in real time. Implemented admission control schemes relying on worst case traffic assumptions compatible with the simplistic leaky bucket traffic descriptor lead to inefficient resource utilization and are hardly satisfactory.

So-called “bufferless multiplexing” is much simpler to control. Admission control is employed to ensure a negligible probability that the combined input rate of multiplexed flows is greater than link capacity. Bufferless multiplexing is less misleadingly termed rate envelope multiplexing since a small buffer is necessary to absorb delays due to simultaneous arrivals of packets from distinct flows. However, the ensuing delays are small and controllable and the loss probability depends only on the stationary distribution of the input rate and not on any correlation (including self-similarity) in the packet arrival process [9].

Many admission control schemes for rate envelope multiplexing have been proposed in the literature. Measurement-based schemes have the most promise since they allow controlled multiplexing performance with minimal prior specification of flow traffic characteristics. The decision theoretic approach proposed in [17] requires only the flow peak rate to be given and the admission criterion consists in comparing the current overall input rate to an appropriately defined threshold. In other words, the probability of an excess arrival rate and consequent loss is circumscribed by accepting or rejecting new flows according to an instantaneous measure of the current link load.

The efficiency of rate envelope multiplexing depends on the relative value of the flow peak rate compared to the link capacity. When this rate is small (less than 1% of the link capacity, say), the admission load threshold is typically high. An example taken from Figure 7 in [17] shows that, for a peak rate to link capacity ratio of 1%, a load threshold equal to 55% of link capacity produces a loss rate smaller than 10^{-8} . Average utilization then depends on flow characteristics other than their peak rate but is always greater than 50% in the above example (see Figure 11 in [17]).

In the following, we assume that streaming rates are indeed small relative to the link capacity. The assumption that streaming traffic will continue to constitute less than 60% of traffic ensures that rate envelope multiplexing is efficient in the integrated services scheme described next.

3.3 Integrating streaming and elastic flows

We assume streaming flows are handled in network nodes using non-preemptive priority queuing with streaming packets having priority over elastic packets. Note that this is a possible realization of the expedited forwarding (EF) per-hop behaviour of the Diffserv model [18]. Giving priority to streaming packets ensures maximal responsiveness for the underlying audio and video applications. The impact of non-priority traffic has been shown to be negligible in a rather precise sense [9]: streaming flow performance does not suffer from an accumulation of jitter as the flows proceed through the network.

We assume streaming traffic is handled using rate envelope multiplexing, i.e., the combined rate of streaming flows is maintained less than the link capacity. By the assumption that the majority of traffic is elastic, the streaming flows see a link with low effective utilization so that this condition is naturally satisfied. We assume that a maximum peak rate is enforced by shaping all streaming flows at the network ingress. Elastic flows share the remaining bandwidth unused by streaming traffic under the control of TCP.

If the combined expected demand from streaming and elastic traffic is less than link capacity, we would expect from the performance results outlined above that quality of service will be satisfactory for both types of flow: sufficient throughput for elastic flows, negligible delay and

loss for streaming flows. In overload, however, the performance of elastic flows in an uncontrolled system would rapidly deteriorate. Streaming flows would suffer much less, or not at all, in view of their priority in the multiplexer queue. This is not a desirable situation since streaming traffic is not necessarily more valuable to users than elastic traffic. To apply admission control to streaming flows to limit their consumption of shared bandwidth is only a partial solution. Such a scenario is modelled in [19]. It is shown that elastic traffic could suffer serious performance degradation even when overall load is less than capacity due to the occurrence of so-called local instability periods. The paper also underlines the unpredictability of elastic flow performance which here depends on the detailed statistical properties of traffic. These different reasons motivate our proposal of applying admission control to both streaming and elastic flows.

The same admission criterion can be applied to all flows resulting in equal probabilities of blocking. Since we assume the majority of traffic is elastic, the admission criterion is naturally related to the performance of an elastic flow. The proposed scheme thus retains the flexibility and simplicity of elastic flow admission control discussed at the end of Section 3.1. The loss and delay performance of streaming flows is automatically assured given our assumption that streaming traffic is in the minority and flows have a limited peak rate. If these assumptions are not satisfied at some time in the future, it may be necessary to add an admission criterion based on streaming load, as in [17].

4 Integrated admission control scheme

In this section, we describe the salient features of an implicit, flow-based admission control scheme designed for an integrated network where streaming and elastic flows share the same links.

4.1 Implicit admission control

Implementing a flow-based admission control mechanism raises serious scalability issues. The very high flow arrival rate on any network link and the small size and duration of the majority of elastic flows require an implicit admission control procedure that avoids signalling and per-flow resource reservation. Furthermore, the admission criterion for estimating the ability of a link or path to accept a new flow must be instantly accessible as each flow arrives. The most satisfactory approach is to rely on a minimal description of each flow coupled with a measurement-based admissibility condition.

We propose to identify new flows on the fly and to reject them, when necessary, using existing protocol semantics at transport layer and above. A list of flows in progress is maintained containing the flow identity and the arrival epoch of the last packet. The flow identity is determined from certain fields in the packet (IP and TCP) header (e.g., source and destination addresses and port numbers, the flow identity field of IPv6). The flow identity of all arriving packets is systematically compared with this list. If a packet belongs to an existing flow it is forwarded; if not, either the new flow is added to the list if it is accepted, or the packet is simply discarded. The discard of the first packets of a flow is sufficient signal to the source that resources are unavailable². Flows would be overwritten or erased from the table whenever the time since the last packet exceeds a certain threshold.

²The experimental evaluation of this mechanism with the TCP stack implemented on Solaris 2.7 shows that a new flow is definitively rejected upon discard of the first seven SYN packets. In the case of a UDP flow, the number of packets discarded before the source resets the connection is application dependent. A negotiation phase generally precedes the sending of data, which typically will not start if negotiations fail.

Alternative implicit admission control procedures have been proposed in [2, 3]. In these procedures flow identification relies on the assumption that flows use TCP and are delimited by the initial three-way handshake. In the interests of generality, we prefer to avoid depending on the detection of SYN and SYN/ACK packets. This is necessary notably to allow admission control by the same procedure for streaming flows which generally use UDP.

4.2 Measurement-based admissibility condition

The admissibility criterion determines whether or not it is possible to accept a new flow. The appropriate criterion for elastic flows is a threshold on *available bandwidth*, i.e., the bandwidth a new elastic flow would acquire by sharing capacity fairly with the flows already in progress. Given the inherent tolerance of elastic flows to rate fluctuations, a rough estimate of available bandwidth is sufficient to determine whether a new flow can be accepted or not. A significant advantage of this approach is that it applies equally to a single link and to a network path. To be able to test an entire path is useful when admission control decisions are performed in the edge routers of an MPLS domain.

Given the assumptions about streaming traffic volume, available bandwidth is also an appropriate measure for determining the admissibility of a streaming flow. The available bandwidth must clearly be greater than a threshold which is at least equal to an assumed maximum streaming flow peak rate. If streaming traffic were greater in volume than elastic traffic or the flow peak rate greater than a small fraction of the link rate, it would be necessary to define a compound admission criterion including a threshold on the current streaming load. We have not yet explored this eventuality.

Measurement-based admission control is particularly easy in an integrated system where the majority of traffic is elastic. Any imprecision in the algorithm leads at worst to a momentary rate reduction for elastic flows but has a negligible impact on the performance of admitted streaming flows. Under the Poisson arrivals assumption, applying the same admission criterion to all flows equalizes blocking probabilities and preserves the relative proportions of streaming and elastic traffic even in overload³. We investigate the optimal choice of the available bandwidth threshold in the remaining sections of the paper.

4.3 Bandwidth estimation algorithms

We have investigated two possible probing algorithms for evaluating the current available bandwidth. The first consists in emulating a TCP connection over the considered link or path and measuring its instantaneous throughput. The second relies on estimating the current loss rate and deriving a bandwidth estimate from the known relation between packet loss and the throughput of a TCP connection.

The first algorithm uses a “phantom” TCP connection, as first proposed by Afek et al [20]. Its throughput is measured simply by averaging the short term rate of acknowledged packets. The phantom connection sends a continuous stream of dummy packets and reacts to packet loss precisely as would a regular TCP connection. This approach can lead to considerable overhead when a link is lightly loaded. This can be alleviated by imposing a maximum rate (through the advertised receive window for example) or by using short control packets and appropriately modifying the TCP code so that acknowledgements are handled as if they correspond to MTU size data packets.

³This design objective implies that streaming traffic and elastic traffic would be charged in a similar manner.

The second algorithm measures the current packet loss rate p and uses the fact that TCP throughput is a known function of p [21]. In practice, it is not straightforward to estimate the loss rate on the link or path in question. One possibility would be to generate a stream of probe packets and to measure their loss rate. A TCP-based tool for measuring the packet loss rate between a pair of hosts was developed in [22] by measuring the loss rate of a patched TCP connection. It is not strictly necessary to convert the loss rate into a throughput estimate since the loss rate can be used directly as an admission criterion. This is the approach adopted in our ns simulation experiments reported later.

5 An analytical performance model

Consider a single bottleneck link of capacity C bits/s shared by streaming and elastic flows arriving according to Poisson processes of intensity λ_s and λ_e , respectively. These flows also share the capacity of the link with the phantom connection which is modeled as a permanent elastic flow⁴. Let the mean size of elastic flows be $1/\mu_e$ bits and the mean duration of streaming flows $1/\mu_s$ seconds. We assume streaming flows have constant bit rate d_s . Denote by $\rho_e = \lambda_e/(C\mu_e)$ and $\rho_s = \lambda_s d_s/(C\mu_s)$ the link load induced by streaming traffic and elastic traffic, respectively, and write ρ for the overall load $\rho_s + \rho_e$.

The bandwidth of elastic flows varies dynamically depending on the number of flows of both types currently in progress. For present purposes, we assume the link bandwidth unused by streaming traffic is shared perfectly fairly with instant readjustment whenever new flows begin or existing flows end. Note, however, that detailed packet level simulations of TCP and UDP connections are used in Section 6 to validate the proposed mechanisms.

Let d be the threshold on the acceptable throughput of elastic flows. In other words, any newly arriving flow is rejected if the instantaneous flow throughput would otherwise decrease below the threshold d . As discussed in Section 4.2, we assume d is greater than or equal to d_s .

Let Q_s and Q_e represent the number of ongoing streaming flows and elastic flows (excluding the phantom connection), respectively. A new flow can be admitted if

$$\frac{C - Q_s d_s}{Q_e + 1} \geq d \quad (3)$$

In the inequality above, the left term corresponds to the instantaneous throughput of the phantom connection. Strictly speaking, this admission condition assures a minimum throughput to elastic flows equal to $\frac{C}{C+d} \times d$.

The use of a unique admissibility condition for streaming and elastic flows yields equal blocking probabilities for both types and prevents the most tolerant class gaining an unfair advantage in case of heavy traffic.

5.1 Quasi-stationary analysis

Let $N_e(i)$ denote the maximum number of elastic flows (excluding the phantom connection) when there are i ongoing streaming flows:

$$N_e(i) = \left\lfloor \frac{C - i d_s}{d} \right\rfloor,$$

⁴Equivalent formulas are derived in [1] for the system without a permanent phantom connection.

and N_s the maximum number of streaming flows in the absence of elastic traffic:

$$N_s = \left\lfloor \frac{C - d + d_s}{d_s} \right\rfloor.$$

Let $\rho_e(i)$ denote the elastic traffic load when i streaming flows are present:

$$\rho_e(i) = \frac{\lambda_e}{\mu_e(C - id_s)}.$$

Rather than solving this system exactly under Markovian assumptions (this would only lead to a complex algorithmic solution), these quantities are computed below under a quasi-stationary, or QS, assumption: the ratio λ_s/λ_e is assumed small enough so that, in the presence of i ongoing streaming flows, Q_e evolves rapidly with respect to Q_s and attains a stationary regime. The number of elastic flows then behaves like the population of an M/G/1 processor sharing queue which is also serving the permanent phantom connection. The birth and death process describing Q_e in the presence of i streaming flows is as follows:

$$\begin{cases} \forall 1 \leq j \leq N_e(i) & Q_e = j \longrightarrow Q_e = j - 1 & \text{with intensity } \frac{j}{j+1}\mu_e(C - id_s) \\ \forall 0 \leq j \leq N_e(i) - 1 & Q_e = j \longrightarrow Q_e = j + 1 & \text{with intensity } \lambda_e \end{cases}$$

In this QS regime we have:

$$\Pr[Q_e = j | Q_s = i] = \frac{(j+1)(1 - \rho_e(i))^2 \rho_e(i)^j}{1 - (N_e(i) + 2)\rho_e(i)^{N_e(i)+1} + (N_e(i) + 1)\rho_e(i)^{N_e(i)+2}}, \quad \forall j \leq N_e(i). \quad (4)$$

Let B denote the flow blocking probability:

$$B = \sum_i \Pr[Q_s = i] \Pr[Q_e = N_e(i) | Q_s = i]. \quad (5)$$

To compute $\Pr[Q_s = i]$, note that Q_s behaves like the population of an M/G/ ∞ system with state dependent arrival rate and maximum size N_s . The arrival rate when $Q_s = i$, denoted $\lambda_s(i)$, is λ_s thinned by the probability a flow is not blocked (i.e., $Q_e < N_e(i)$):

$$\lambda_s(i) = \lambda_s(1 - \Pr[Q_e = N_e(i) | Q_s = i]), \quad \forall i < N_s.$$

The distribution of Q_s under the QS assumption is thus,

$$\Pr[Q_s = i] = \Pr[Q_s = 0] \prod_{k=0}^{i-1} \frac{\lambda_s(k)}{(k+1)\mu_s}, \quad \forall i \leq N_s. \quad (6)$$

where $\Pr[Q_s = 0]$ is given by the normalization condition $\sum \Pr[Q_s = i] = 1$.

The blocking probability B can thus be derived from (5). To evaluate throughput performance we proceed as follows. The expected response time R of an elastic flow can be deduced using Little's formula with the expected number of elastic flows in progress $E[Q_e]$ derived from the distributions (4) and (6):

$$R = \frac{E[Q_e]}{\lambda_e(1 - B)}.$$

Now, it is known that the response time $R(s)$ of a flow of size s in an M/G/1 processor sharing queue is proportional to s [23]. The constant of proportionality can be deduced on remarking that $R = E[R(s)]$, implying $R(s) = R\mu_e s$. Define $\gamma_e = s/R(s)$ to be the (harmonic) mean throughput of a flow of any size [6]:

$$\gamma_e = \frac{\rho_e(1-B)}{E[Q_e]}C.$$

It is noteworthy that, under the QS assumption, the above results for the performance parameters of interest B and γ_e are *insensitive* to the distributions of elastic flow size and streaming flow duration.

We now compute the average throughput of the phantom connection. Let $\gamma_{ph}(i)$ denote the average throughput realized by the phantom connection when Q_s is in state i . $\gamma_{ph}(i)$ is obtained by averaging over all possible states of Q_e the instantaneous throughput of the phantom connection.

$$\gamma_{ph}(i) = \sum_{j=0}^{N_e(i)} \frac{C - id_s}{j+1} \times \mathbb{P}[Q_e = j | Q_s = i]$$

It follows that

$$\gamma_{ph}(i) = \frac{C - id_s(1 - \rho_e(i))(1 - \rho_e(i)^{N_e(i)+1})}{1 - (N_e(i) + 2)\rho_e(i)^{N_e(i)+1} + (N_e(i) + 1)\rho_e(i)^{N_e(i)+2}}$$

The average throughput of the phantom connection is given by

$$\gamma_{ph} = \sum_{i=0}^{N_s} \mathbb{P}[Q_s = i] \times \gamma_{ph}(i)$$

It can be shown that, when $\rho < 1$ and in the absence of admission control, the average throughput of the phantom connection is exactly twice the average throughput of the admitted elastic flows.

5.2 Validity of the QS assumption

We have simulated the fluid flow system with the objective of validating the analytical model and examining sensitivity with respect to size and duration distributions when the QS assumption is not appropriate. Assumed parameter values are as follows: $C = 10$ Mbit/s, $d_s = 0.0015C$, $1/\mu_e = 200$ Kbits, $1/\mu_s = 60$ s. We consider a class of hyper-exponential distributions for both streaming flow duration and elastic flow size defined as follows:

$$\forall x \geq 0, \Pr[s > x] = \frac{a \exp^{-ax/\sigma} + \exp^{-x/(a\sigma)}}{a+1}$$

where σ is the mean and the parameter a controls the proportions of short and long flows.

Figures 2 and 3 plot the blocking probability B and the normalized average throughput γ_e/C , respectively, against the admission threshold for $\rho = 0.9$ and $\rho = 1.4$. Elastic flow sizes are drawn from the hyper-exponential distribution with $a = 100$. We show simulation results for two different distributions of streaming flow duration, exponential ($a = 1$) and hyper-exponential ($a = 100$). Figure 4 plots the normalized average throughput of the phantom connection.

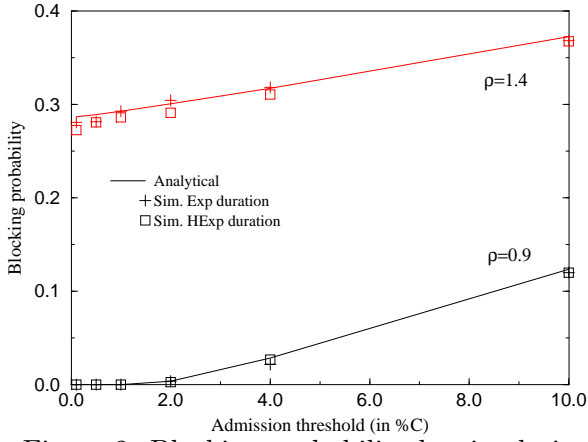


Figure 2: Blocking probability by simulation and analysis of the fluid model

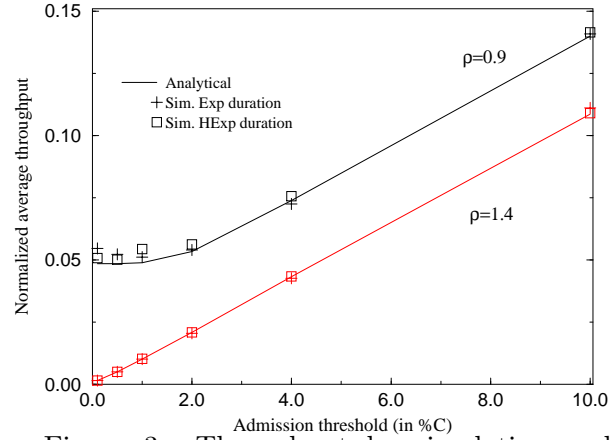


Figure 3: Throughput by simulation and analysis of the fluid model

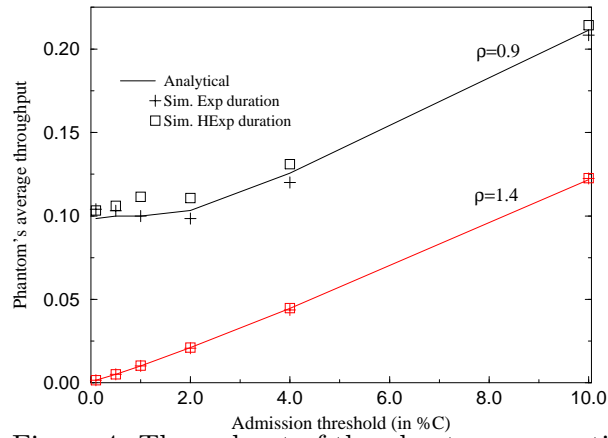


Figure 4: Throughput of the phantom connection by simulation and analysis of the fluid model

We observe that the simulation results are very close to the analytical results in all cases and for all considered duration distributions. In line with the last remark of Section 5.1, the phantom connection grabs twice as much bandwidth as the admitted elastic flows when blocking is negligible. We have also validated the approximation for different elastic flow size distributions and for a shorter mean streaming flow duration of 30 s. The results confirm that the analytical fluid model constitutes a good approximation under rather general and realistic traffic patterns.

5.3 Setting the admissibility threshold

It may readily be verified that when $\rho < 1$ the blocking probability B tends rapidly to zero as the threshold decreases below $0.02C$, e.g., a threshold of $0.01C$ yields $B < 0.001$ for $\rho \leq 0.95$. On the other hand, when $\rho > 1$ we have $B \approx (\rho - 1)/\rho$ whenever the threshold is less than $0.02C$.

When $\rho < 1$, the normalized average throughput is virtually independent of the threshold (and equal to $(1 - \rho)/2$) as soon as the threshold is smaller than $0.02C$. In overload, on the other hand, throughput is roughly proportional to the threshold since the link is almost always saturated.

It is important to note that admitting flows beyond a certain threshold does not reduce the blocking probability in overload and therefore only deteriorates performance. An optimal choice of admissibility threshold should produce negligible blocking in normal load while maintaining sufficiently high throughput in overload. In the light of the above results, a reasonable compromise for the present system is a threshold between 0.5% and 2% of link capacity. The precise value of the admission threshold is not critical. It is therefore not necessary to implement a highly accurate procedure for estimating available bandwidth.

6 Evaluation of the integrated implicit admission control

To evaluate admission control algorithms under realistic traffic conditions we have performed a number of simulation experiments using NS⁵ (Network Simulator, version 2).

6.1 Simulation model

Network model. We considered the simple dumb-bell topology shown in Figure 5. All links have the same fixed delay of 10 ms. Admission control is performed on the 10 Mbit/s bottleneck link. We have verified that the 5 Mbit/s access links do not affect throughput in this configuration. The link buffer has a capacity of 50 packets. Packets of streaming flows are placed at the front of the queue, but have to wait for the complete transmission of the packet currently being served.

Traffic model. TCP connections are generated by each source node according to a Poisson process. Each connection is used to transfer a stream of 1 Kbyte packets representing a document of a certain size and then terminated. The document size is drawn from the following distribution: 90% of documents are so-called “mice” with size uniformly distributed between 1 and 9 packets; the remainder, deemed “elephants” have size uniformly distributed between 10 and 400 packets. This choice is made for the sake of simplicity. Performance is largely independent of the size distribution, as noted in Section 5.

⁵<http://www.mash.cs.berkeley.edu/ns/>

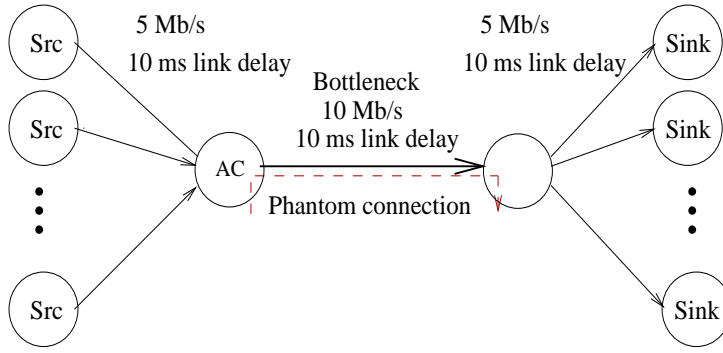


Figure 5: Simulated network topology with a single bottleneck link

Streaming traffic is generated in the form of UDP connections and represents 20% of the offered load. Flow duration is drawn from an exponential distribution with a mean of 60 sec. All UDP packets have a fixed size of 190 bytes. We considered two different traffic models, CBR and on/off. In the CBR model, the UDP connections have a fixed constant rate equal to 15 kbit/s. In the second model, connections are intermittent with the transmission rate in an on-period ten times that of the CBR connections and the probability of being in an on-period equal to 0.1. On- and off-period durations follow an exponential distribution.

6.2 TCP phantom estimator

Measuring bandwidth. The TCP phantom connects the extremities of the bottleneck link. Its goodput, equal to the rate of acknowledged packets, is measured in fixed time intervals of length δ . Let τ_n denote the number of packets acknowledged in interval $((n-1)\delta, n\delta)$ and let B_n be the available bandwidth estimate derived at time $n\delta$. We apply exponential smoothing with parameter α , $0 < \alpha < 1$:

$$B_n = \alpha \times B_{n-1} + (1 - \alpha) \times \tau_n / \delta. \quad (7)$$

The values of δ and α are not highly critical to the accuracy of the method. Following a series of initial experiments we settled on $\delta = 0.1$ seconds (corresponding to 5 times the RTT of the phantom connection) and $\alpha = 0.9$.

Choice of admission threshold. Figure 6 shows how the blocking probability depends on the admission threshold in underload and overload. The figure contrasts the simulation results with the predictions of the theoretical fluid model of Section 5. Figures 7 and 8 plot the throughput realized by large connections of more than 100 packets and by the phantom connection, respectively, as a function of the admission threshold. Table 1 compares the throughput realized by the admitted connections depending on their size.

We observe that the throughput achieved by TCP flows depends on their size, whereas it did not in the fluid model. The throughput of mice is severely limited by TCP slow start even when the number of simultaneously admitted flows is small. Their throughput dominates results for “all connections”. The throughput of large transfers of more than 100 packets depends more on TCP congestion avoidance phase and is more sensitive to the choice of threshold in overload. The permanent phantom has the highest throughput of all. The fluid model provides a rather accurate approximation for the throughput of the phantom connection. It yields optimistic

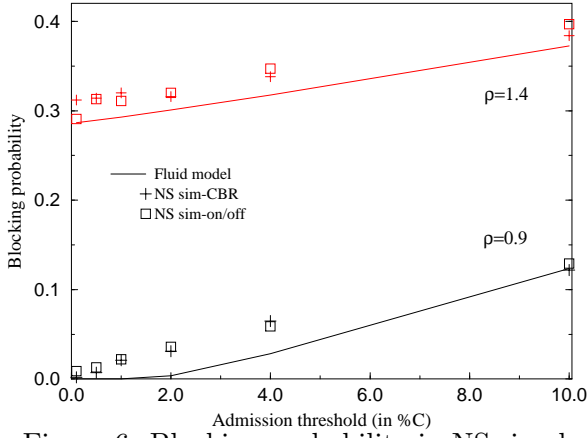


Figure 6: Blocking probability in NS simulations compared with analytical results

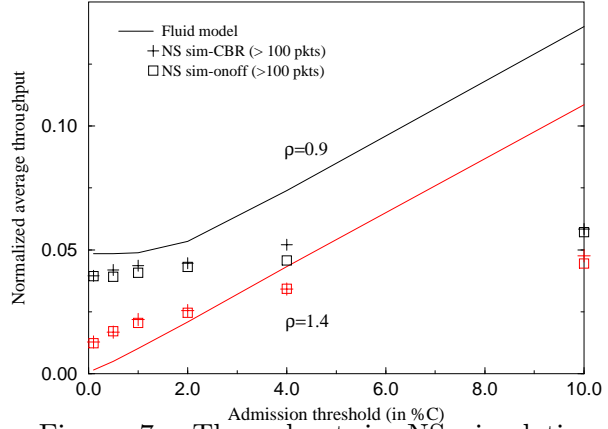


Figure 7: Throughput in NS simulations compared with analytical results

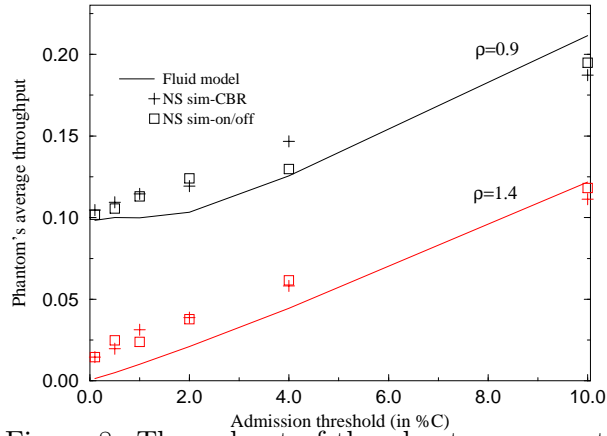


Figure 8: Throughput of the phantom connection in NS simulations compared with analytical results

results, however, for the throughput of admitted connections for a threshold greater than 4%. For such a threshold the admitted connections are not able to completely saturate the link, as assumed in the fluid model.

In line with the predictions of the fluid model results, we observe that there is no advantage in choosing a threshold smaller than 0.5% since for such values blocking is negligible in underload ($\rho = 0.9$) and tends to the fluid limit $(\rho - 1)/\rho$ in overload ($\rho = 1.4$). A threshold lower than 0.5% increases response times and does not reduce the blocking probability. A threshold greater than 4% tends to be inefficient, yielding high blocking and no compensating increase in throughput. In conclusion, simulations confirm that any value between 0.5% and 2% constitutes an acceptable admission threshold.

The results are broadly the same for CBR and on/off streaming flows as illustrated in Table 1, suggesting that TCP adjusts elastic flow rates sufficiently rapidly. Consequently, the same conclusions apply with respect to the choice of threshold.

Threshold		$\rho = 0.9$			$\rho = 1.4$		
		All	> 100 pkts	Phantom	All	> 100 pkts	Phantom
0.5%	CBR	1.8	5.2	10.9	1.3	2.1	2.0
	on/off	1.8	4.9	10.6	1.3	2.2	2.5
2%	CBR	1.9	5.4	11.9	1.5	3.0	3.9
	on/off	1.9	5.3	12.4	1.5	2.9	3.8
4%	CBR	2.0	6.3	14.7	1.7	4.0	5.8
	on/off	1.9	5.5	13.0	1.7	4.1	6.1

Table 1: Impact of the rate threshold on the realized throughput (in %) of connections, $\rho = 0.9, 1.4$

We have also checked the suitability of our choice of threshold for higher proportions of CBR traffic (up to 50%) and for higher UDP connection rates (100 kbps, i.e., 1% of link capacity). The resulting performance for elastic flows is similar to that obtained in the previous scenarios. Hence, values in the range of 1% to 2% constitute a suitable threshold choice (the threshold is necessarily greater than or equal to the peak rate of streaming flows).

6.3 Loss rate estimator

In this section we evaluate the effectiveness of the second admission control approach based on the measured loss rate. In the simulations, we simply measure the loss rate averaged over all TCP packets using the bottleneck link, in 0.1 second time intervals, using exponential smoothing.

It proves difficult to precisely calibrate the observed loss rate with a target available bandwidth. Table 2 gives the average throughput realized by all flows and by large flows (> 100 packets), respectively, together with the blocking rates for different loss thresholds, in underload and in overload conditions. We notice hardly any difference between the results obtained with the CBR traffic model and those relative to the on/off model.

A threshold smaller than 1% is overly conservative and leads to significant blocking in normal load. For thresholds greater than 5% blocking is relatively stable and roughly equal to the fluid limit. Realized throughput decreases as the threshold increases, notably for large transfers.

Loss threshold		$\rho = 0.9$			$\rho = 1.4$		
		All	>100 pkts	Blocking	All	>100 pkts	Blocking
1%	CBR	3.3	13.9	13.7	2.8	11.4	40.8
	on/off	3.3	13.5	14.0	2.9	11.3	42.5
5%	CBR	2.9	10.7	1.2	2.0	5.5	28.8
	on/off	2.8	9.9	1.3	2.0	5.0	30.0
10%	CBR	2.8	10.5	0	1.5	2.8	24.9
	on/off	2.7	9.5	0	1.5	2.52	27.1

Table 2: Impact of the loss threshold on the realized throughput (in %) of connections and on the blocking (in %), $\rho = 0.9, 1.4$

7 Conclusion

We have proposed an integrated admission control scheme applying to both streaming flows and elastic flows. This scheme uses implicit flow rejection and measurement-based admissibility conditions avoiding the need for signalling and per-flow resource reservation. It consists in estimating the rate a new elastic flow would acquire and accepting a new (streaming or elastic) flow only if it would not reduce the throughput of ongoing elastic flows below a certain threshold. Giving priority to packets of streaming flows minimizes loss and delay for audio and video applications without penalizing elastic flows whose minimum throughput is guaranteed.

In order to investigate the choice of an optimal admission threshold, we have developed an analytical fluid model integrating both classes of traffic under the quasi-stationary assumption. The model was shown using simulation to provide a good approximation under rather general and realistic traffic assumptions. The range of optimal admission threshold values predicted by the model was confirmed by simulations taking into account UDP and TCP dynamics at packet level. We found that any threshold in the range of 0.5% to 2% of the link capacity is appropriate. This range of values coincides with that proposed in [5] where only elastic traffic was offered. It appears that streaming traffic in the considered proportion (up to 20%) has little impact in this respect. One of the conclusions that can be drawn from this study is that the choice of threshold is not critical so that the estimation of available bandwidth does not need to be highly accurate.

In order to estimate the available bandwidth, two algorithms were evaluated and shown to work satisfactorily. One of the algorithms relies on a TCP phantom connection and the second is based on measuring the packet loss rate. The inherent tolerance of elastic flows to rate fluctuations and the fact that streaming traffic is admitted at a relatively low load level partly explain the effectiveness of both algorithms.

A remaining critical issue is the feasibility of flow identification on the fly on very high speed interfaces. A test bed is currently being set up to evaluate feasibility and performance in a real network environment. It is also necessary to account for the limited elasticity of all flows on high speed backbone links where the impact of rate limitations in the access network is preponderant.

References

- [1] N. Benameur, S. Ben Fredj, F. Delcoigne, S. Oueslati-Boulahia, and J.W. Roberts. Integrated admission control for streaming and elastic traffic. In *QofIS*, pages 69–81. Second COST 263 International Workshop, Coimbra, Portugal, Springer, September 2001.

- [2] A. Kumar, M. Hegde, S.V.R. Anand, B.N. Bindu, D. Thirumurthy, and A.A. Kherani. Non-intrusive TCP Connection Admission Control for Bandwidth Management of an Internet Access Link. *IEEE Communications Magazine*, 38(5):160–167, May 2000.
- [3] R. Mortier, I. Pratt, C. Clark, and S. Crosby. Implicit Admission Control. *IEEE Journal on Selected Areas in Communications*, 18(12):2629–2639, December 2000.
- [4] J.W. Roberts and S. Oueslati-Boulahia. Quality of Service by Flow Aware Networking. *Phil. Trans. Royal Society London*, 358:2197–2207, 2000.
- [5] S. Ben Fredj, S. Oueslati-Boulahia, and J.W. Roberts. Measurement-based Admission Control for Elastic Traffic. In J. Moreira de Souza, N. L.S. da Fonseca, and E.A. de Souza e Silva, editors, *Teletraffic Engineering in the Internet Era*, pages 161–172. ITC 17, Elsevier, December 2001.
- [6] S. Ben Fredj, T. Bonald, A. Proutière, G. Régnié, and J.W. Roberts. Statistical Bandwidth Sharing: A Study of Congestion at Flow Level. In *ACM SIGCOMM Computer Communication Review*, pages 111–122, 2001.
- [7] V.A. Bolotin. New Subscriber Traffic Variability Patterns for Network Traffic Engineering. In V. Ramaswami and P.E. Wirth, editors, *Teletraffic Contributions for the Information Age*, pages 867–878. ITC 15, Elsevier, 1997.
- [8] M. Grasse, M.R. Fratter, and J.F. Arnold. Origins of Long-Range Dependence in Variable Bit Rate Video Traffic. In V. Ramaswami and P.E. Wirth, editors, *Teletraffic Contributions for the Information Age*, pages 1379–1388. ITC 15, 1997.
- [9] T. Bonald, A. Proutière, and J.W. Roberts. Statistical Performance Guarantees for Streaming Flows using Expediated Forwarding. In *INFOCOM*, pages 1104–1112, 2001.
- [10] K. Thomson, G.J. Miller, and R. Wilder. Wide-Area Internet Traffic Patterns and Characteristics. *IEEE Network*, pages 10–23, Nov/Dec 1997.
- [11] S. McCreary and K. Claffy. Trends in wide area IP traffic patterns - a view from Ames internet exchange. In *13th ITC Specialist Seminar on Internet Traffic Measurement and Modelling*, September 2000. Monterey (CA).
- [12] T. Bonald and L. Massoulié. Impact of Fairness on Internet Performance. In *SIGMETRICS Performance Evaluation Review*, pages 82–91, June 2001.
- [13] J.W. Roberts and L. Massoulié. Bandwidth Sharing and Admission Control for Elastic Traffic. *Telecommunication Systems*, 15:185–201, 2000.
- [14] L. Massoulié and J.W. Roberts. Arguments in Favor of Admission Control for TCP Flows. In P. Key and D. Smith, editors, *Teletraffic Engineering in a Competitive World*, pages 33–44. ITC 16, Elsevier, June 1999.
- [15] T. Bonald and J.W. Roberts. Performance Modeling of Elastic Traffic in Overload. In *SIGMETRICS Performance Evaluation Review*, pages 342–343, June 2001.
- [16] J.W. Roberts, U. Mocci, and J. Virtamo. *Broadband Network Teletraffic, Final Report of European Action COST 242*. Springer, 1996.

- [17] R.J. Gibbens, F.P. Kelly, and P.B. Key. A Decision-Theoretic Approach to Call Admission Control in ATM Networks. *IEEE Journal on Selected Areas in Communications*, 13(6):1101–1114, August 1995.
- [18] S. Blake et al. An architecture for Differentiated Services. RFC 2475, 1998.
- [19] F. Delcoigne, A. Proutière, and G. Régnié. Modelling Integration of Streaming and Data Traffic. Submitted.
- [20] Y. Afek, Y. Mansour, and Z. Ostfeld. Phantom: A Simple and Effective Flow Control Scheme. In *ACM SIGCOMM Computer Communication Review*, pages 169–182, August 1996.
- [21] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose. Modeling TCP Throughput: A Simple Model and its Empirical Validation. In *ACM SIGCOMM Computer Communication Review*, pages 303–314, October 1998.
- [22] Stefan Savage. Sting: a TCP-based Network Measurement Tool. In *USENIX Symposium on Internet Technologies and Systems*, pages 71–79, October 1999.
- [23] J.W. Cohen. The Multiple Phase Service Network with Generalized Processor Sharing. *Acta Informatica*, 12:245–285, 1979.