Centrum voor Wiskunde en Informatica

Performance evaluation of strategies for integration of elastic and stream traffic

R. Núñez Queija, J.L. van den Berg, M.R.H. Mandjes

Probability, Networks and Algorithms (PNA)

PNA-R9903 February 28, 1999

Report PNA-R9903
ISSN 1386-3711

CWI
P.O. Box 94079
1090 GB Amsterdam
The Netherlands

Performance Evaluation of Strategies for Integration of Elastic and Stream Traffic

Rudesindo Núñez Queija
CWI
P.O. Box 94079, 1090 GB Amsterdam, The Netherlands
sindo@cwi.nl

Hans van den Berg and Michel Mandjes*
KPN Research
P.O. Box 421, 2260 AK Leidschendam, The Netherlands
{J.L.vandenBerg,M.R.H.Mandjes}@research.kpn.com

* *Current affiliation: Bell Laboratories/Lucent Technologies,*
600 Mountain Ave., P.O. Box 636, Murray Hill, NJ 07974-0636, USA
michel@research.bell-labs.com

ABSTRACT
This paper deals with the integration of 'stream' traffic and 'elastic' traffic in one single network, e.g. an ATM-based or an IP-based network. Here stream traffic refers to traffic with a certain bandwidth guarantee, whereas elastic traffic flows can adapt their rates to the link bandwidth left over by the stream flows. First, models are developed that describe different strategies for sharing link capacities between the stream and elastic flows. Then we give mathematical methods for obtaining performance measures, in particular call blocking probabilities and file transfer delays. Finally, these methods are used for assessing and comparing the efficiency gains achieved by the integration strategies.

1991 Mathematics Subject Classification: 60K25, 68M20, 90B12, 90B22.
Keywords & Phrases: Integrated services, stream and elastic traffic, real-time and best-effort traffic, file transfer delay, call blocking probability, processor sharing queues.
Note: The work of the first author was carried out in PNA 2.1. as part of the project 'Quality in Future Networks' of the Telematics Institute, and while he visited KPN Research (Leidschendam) on a part-time basis in the period March – June, 1998.

1. Introduction

Two major network concepts have been proposed to support large-scale multiservice networks: ATM (Asynchronous Transfer Mode) and IP (Internet Protocol).
Originally, IP networks (particularly the Internet) were built for data transfer purposes. Consequently, they were not appropriate for supporting real-time services; all traffic was handled on a best effort basis. For that reason, within the Internet society, notably the Internet Engineering Task Force (IETF), considerable effort is put into concepts for introducing Quality of Service (QoS) guarantees for prioritised streams, see for instance Van der Wal et al. [22] and White and Crowcroft [23]. Several proposals have been made, the merits of which are currently investigated, particularly within the IETF working groups *intserv* and *diffserv*.
In the 'telecommunications world', however, there is a strong impetus towards a multiservice network based on ATM, as standardised by ITU and ATM Forum. ATM networks have been designed from the point of view that applications require a strict QoS level. For that reason, ATM is particularly suited for supporting real-time services (having stringent delay requirements). In ATM's original form,

there was no specific facility for handling traffic with relatively low QoS requirements (for instance data transfer), leading to an inefficient use of network resources. In order to cope with this problem, the development of transfer capabilities such as ABR (Available Bit Rate) and UBR (Unspecified Bit Rate) started. As the bandwidth allocated to ABR and UBR strongly depends on the network congestion, there is a strong similarity with IP's best effort class.

From the above description, we see that both network concepts aim at integrating traffic with a certain bandwidth guarantee (or *stream traffic*, cf. Roberts [18, 19]), and *elastic traffic*, that can cope with a non guaranteed, variable, bandwidth. Stream traffic must maintain a so-called *time integrity*; it is generated by interactive applications, like telephony and interactive video. In ATM, this stream mode is supported by the transfer capabilities DBR (Deterministic Bit Rate) or SBR (Statistical Bit Rate), in IP by the guaranteed QoS class, possibly in conjunction with RSVP (ReSerVation Protocol), e.g. White and Crowcroft [23]. Elastic traffic does not exist on its own, in that the rates at which the sources are allowed to send traffic into the network are determined by the level of network congestion. For these elastic flows particularly *semantic integrity* should be preserved. In ABR this integrity is achieved by a feedback loop (reporting the sources on the level of congestion in the network) in conjunction with a large buffer; in IP by the TCP feedback loop together with retransmissions.
This paper aims at shedding light on the merits of the integration of stream traffic and elastic traffic. From the point of view of operational complexity, it is probably preferable to have two (or even more) dedicated networks; but regarding efficient use of resources, integration may be beneficial. It is this efficiency gain (in terms of bandwidth) achieved by integration that we assess.

To get insight into the above issues, network performance analysis is required. Performance studies on networks with elastic traffic can be roughly categorised into two groups:

(1) Detailed studies, mainly at cell/packet level, of the performance of ABR and TCP/IP feedback mechanisms. See e.g Bonomi et al. [5], Ritter [17], and Blondia and Casals [4], who study the performance of various ABR feedback policies. The performance of several variants of TCP/IP is studied in e.g. Lakshman and Madhow [11]. Studies in this category typically investigate buffer requirements (in a bottleneck node), throughputs and the impact of round trip delays on these performance measures. Analytical results are mostly only available for the case of a single elastic traffic source feeding into the network. Most papers do not consider the integration with stream traffic.

(2) Performance studies at call level, in order to study the impact of the interaction between elastic traffic flows and stream traffic flows on throughputs, transfer delays and blocking probabilities. In these call level models, the feedback mechanism is assumed to be 'ideal' (i.e. instantaneous feedback). With this assumption, a network link carrying only elastic traffic flows can be modelled by a *processor sharing queue*. The application of processor sharing queues to study the performance of elastic traffic was identified by e.g. Roberts [18, 19] and Núñez Queija and Boxma [15]. The performance of processor sharing queues has been extensively studied and many results (particularly on the queue length and transfer delay distribution) are available, see e.g. Coffman et al. [6], Ott [16], Schassberger [20], and Yashkov [24], and the survey papers of Yashkov [25, 26]. However, we are particularly interested in the behaviour of *integrated* stream and elastic traffic. In the integrated case, the classical processor sharing queue has to be extended in order to model the impact of the presence of a varying number of stream traffic calls. First rough estimates for the performance of integrated stream and elastic traffic were provided by Lindberger [10]. A more advanced modelling is proposed in Núñez Queija and Boxma [15], Blaabjerg et al. [3], and Altman et al. [1]. These studies underly the methodology applied in the present paper.

The contribution of this paper is twofold. In the first place, we present a mathematical modelling and performance analysis of the integration policies. Secondly, using this method, we present an extensive numerical study in order to get insight into the amount of network resources that can be saved by different integration policies.

We organised this paper as follows. Section 2 further specifies the scope of the paper. In Section 3, the model is described and preliminary results on the relevant performance measures are provided. In Section 4 we analyse the elastic traffic transfer delay in greater detail. Section 5 provides numerical results. We draw conclusions in Section 6.

2. Problem description

We consider a single network link with a certain capacity (bandwidth), that carries both stream traffic and elastic traffic. *Stream traffic* consists of calls requiring a given bandwidth, to be guaranteed by the network (in fact, in case of a variable bit rate stream traffic call, this bandwidth is the *effective* bandwidth). These calls arrive according to some stochastic process, and are cleared after some random time. We assume that an *elastic traffic* call is a file to be transferred; the files (having a random size) arrive according to a stochastic process. The elastic traffic calls share the link bandwidth that is not used by the stream traffic calls.

The (call-level) performance of the stream calls is determined by the fraction of calls being blocked. For elastic traffic, there are two relevant performance measures: (1) The time it takes to transfer a file; we particularly concentrate on the *mean transfer time* of a file of given size. (2) The call blocking probability, in case the elastic calls are guaranteed a minimum bandwidth. For example, in the ATM context the ABR service category provides a Minimum Cell Rate (MCR); in IP networks, minimum throughputs for elastic traffic may be realised by the introduction of packet scheduling mechanisms like weighted fair queueing (WFQ), in conjunction with certain flow admission control schemes, see e.g. Roberts [18].

In this paper we analyse and compare the performance of three different policies/scenarios for handling stream and elastic traffic calls.

- *Segregated scenario.* In the first place, we consider the scenario where stream and elastic traffic are handled by separated resources. One part of the link rate is exclusively dedicated to stream traffic, the other part is exclusively dedicated to elastic traffic (i.e., virtually, two dedicated links are used).
- *Integrated scenario.* In the 'opposite' scenario both traffic types completely share the network resources. The rationale for this scenario is the possibility of achieving a high utilisation. The elastic flows allowed on the link can fully exploit the bandwidth that is not used by the stream flows.
- *Mixed scenario.* In this scenario the link bandwidth is split up into two parts. One part can only be used by elastic traffic flows. The other part of the link bandwidth is to be occupied by the bandwidth requirements of the stream traffic flows. An elastic (respectively stream) flow is blocked when the sum of the guaranteed bandwidths becomes larger than the part of the link bandwidth assigned to elastic (respectively stream) traffic. Note, that in this mixed scenario the bandwidth of the 'stream traffic part' of the link that is not actually used by stream traffic flows can be exploited as excess bandwidth by the elastic flows. The rationale for this scenario is to have the benefit of efficiency gain (as in the integration scenario), but at the same time offering calls of both types a certain 'protection' (at call level) against each other, when calls of one type generate (temporarily) a relatively large load.

3. Model description and preliminary analysis

In this section we present the model that we have developed to describe the three scenarios of Section 2. First, in Section 3.1 we discuss the assumptions that we make in our model. Then, in Sections 3.2, 3.3 and 3.4, we separately treat the three different policies, mentioned in Section 2.

3.1 Modelling assumptions

Throughout this paper, we assume that requests for elastic traffic connections and stream traffic connections occur according to two mutually independent Poisson processes, with intensities λ_e and λ_s calls per second, respectively.
A call of elastic traffic consists of a single file to be transmitted. The mean file length is denoted by f_e. Except in the segregated model of Section 3.2, we assume that the lengths of these files are exponentially distributed (in Section 6 we come back to this assumption). Each file transfer requires a minimum guaranteed transfer rate $r_e^- \geq 0$, during the complete transfer time. Also, the actual transfer rate of an individual file can never exceed the maximum attainable transfer rate r_e^+. Obviously, $r_e^+ \geq r_e^-$. For example, in the context of the ABR service in ATM networks, r_e^+ is called the Peak Cell Rate (PCR) and r_e^- is the Minimum Cell Rate (MCR).
Calls of stream traffic require a fixed transfer rate $r_s > 0$ over the complete duration of their holding times. Again with the exception of Section 3.2, we assume that these holding times are exponentially distributed. We denote the mean holding time by h_s.

The fractions of blocked calls of elastic and stream traffic are denoted by p_e and p_s, respectively. In addition, for elastic traffic, we consider $\mathrm{E}[T_e]$, the mean file transfer time, and $\mathrm{E}[T_e(x)]$, the mean file transfer time of a file of length x. Obviously, for exponentially distributed file lengths,

$$\mathrm{E}[T_e] = \int_{x=0}^{\infty} \mathrm{E}[T_e(x)] \frac{1}{f_e} \mathrm{e}^{-x/f_e} \mathrm{d}x.$$

Before proceeding, we first introduce some further notation. We use the random variable $X_e(t)$, resp. $X_s(t)$, to denote the numbers of elastic, resp. stream, traffic connections at time $t \geq 0$. In steady state we simply use X_e and X_s. The state space S of the process $(X_e(t), X_s(t))$ depends on the model considered. The call admission policy can be formulated as follows: Suppose $(X_e(t), X_s(t)) = (n_e, n_s) \in S$. Then, if a new elastic traffic call arrives at time t, it is accepted if $(n_e + 1, n_s) \in S$, and rejected otherwise. Similarly, a new stream traffic call is accepted iff $(n_e, n_s + 1) \in S$. For notational convenience, we introduce the blocking regions $B_e := \{(n_e, n_s) \in S : (n_e + 1, n_s) \notin S\}$ and $B_s := \{(n_e, n_s) \in S : (n_e, n_s + 1) \notin S\}$. We define the steady-state probabilities, for all possible states $(n_e, n_s) \in S$,

$$\pi_{n_e,n_s} := \mathrm{P}\left\{X_e = n_e, X_s = n_s\right\} = \lim_{t\to\infty} \mathrm{P}\left\{X_e(t) = n_e, X_s(t) = n_s\right\}. \tag{3.1}$$

It will be convenient to order the states (n_e, n_s) lexicographically, i.e. (n_e, n_s) is preceded by all states in the set $\{(n_e', n_s') \in S : n_e' < n_e\} \cup \{(n_e, n_s') \in S : n_s' < n_s\}$. Throughout this paper we use this ordering for the elements of vectors defined on the state space. For example, using this ordering on the corresponding steady-state probabilities π_{n_e,n_s}, we define the steady-state probability vector $\overline{\pi} := (\pi_{n_e,n_s})_{(n_e,n_s)\in S}$.

In Sections 3.2, 3.3 and 3.4, the differences between the three proposed policies are discussed in detail. We do not go into the issue of how to compute the steady-state probabilities efficiently for the integrated and mixed scenario. We only remark that, in both cases, the block tri-diagonal structure of the generator allows for an efficient solution. The steady-state probabilities can for instance be computed using the method of De Nitto Personè and Grassi [8] for generalised Quasi Birth-Death processes (with some minor modifications).

Performance measures

Once the steady-state probabilities π_{n_e,n_s} have been determined, we can compute the blocking probabilities p_s and p_e, and the mean number of elastic traffic connections $\mathrm{E}[X_e]$:

$$\begin{aligned} p_s &= \sum_{(n_e,n_s)\in B_s} \pi_{n_e,n_s}, \\ p_e &= \sum_{(n_e,n_s)\in B_e} \pi_{n_e,n_s}, \\ \mathrm{E}[X_e] &= \sum_{(n_e,n_s)\in S} n_e \pi_{n_e,n_s}. \end{aligned} \tag{3.2}$$

By Little's formula we also have $\mathrm{E}[T_e] = \mathrm{E}[X_e]/\left(\lambda_e(1-p_e)\right)$.
The last performance measure we consider, is $\mathrm{E}[T_e(x)]$. In the segregated model, $\mathrm{E}[T_e(x)]$ is proportional to x, see Section 3.2. We present the analysis of $\mathrm{E}[T_e(x)]$ for both the completely integrated model and the mixed model in Section 4.

3.2 Segregated scenario

In this section, we consider the special case with no interaction between stream traffic and elastic traffic. For this case, the only assumption we make on the distributions of the holding times of stream traffic calls and the lengths of elastic traffic files, is that their first moments exist.
The link capacity C is split into two parts: $C = C_e + C_s$. The capacity C_e is permanently assigned to elastic traffic, and C_s to stream traffic. The state space is therefore given by

$$S^{(\mathrm{seg})} := \left\{(n_e, n_s) \in \mathbb{N}_0 \times \mathbb{N}_0 : n_e r_e^- \le C_e, n_s r_s \le C_s\right\}. \tag{3.3}$$

For stream traffic this results in the Erlang loss model. In particular,

$$p_s = \frac{(\lambda_s h_s)^{K_s}/K_s!}{\sum_{k=0}^{K_s} (\lambda_s h_s)^k/k!}. \tag{3.4}$$

Here, $K_s = \lfloor C_s/r_s \rfloor$ is the maximum number of stream traffic connections.
For elastic traffic, the resulting model is an M/G/1/K queue with so called generalised processor sharing (GPS) service discipline. The elastic traffic connections are served simultaneously, each with speed r_{n_e}, where r_{n_e} depends on the total number of elastic traffic connections n_e. In our case we have for $0 \le n_e \le K_e$,

$$r_{n_e} = \min\left(r_e^+, \frac{C_e}{n_e}\right),$$

with $K_e := \lfloor C_e/r_e^- \rfloor$ the maximum number of elastic traffic connections.
For this queueing model, explicit performance results are available in Cohen [7]. In particular, we obtain the mean file transfer time $\mathrm{E}[T_e(x)]$ for a file of given size x, and the probability p_e that a newly arriving file (elastic traffic flow) is blocked. Let,

$$\phi_n := \prod_{j=1}^{n} \frac{1}{r_j}, \quad n = 1, 2, ..., K_e,$$

and $\phi_0 := 1$. Then

$$p_e = \frac{\frac{(\lambda_e f_e)^{K_e}}{K_e!}\phi_{K_e}}{\sum_{j=0}^{K_e} \frac{(\lambda_e f_e)^j}{j!}\phi_j}, \quad \mathrm{E}[T_e(x)] = (x/C_e)\frac{\sum_{n=0}^{K_e-1} \frac{(\lambda_e f_e)^n}{n!}\phi_{n+1}}{\sum_{j=0}^{K_e} \frac{(\lambda_e f_e)^j}{j!}\phi_j}. \tag{3.5}$$

Formula (3.5) shows that, in the present case without interfering stream traffic, the mean file transfer delay $\mathrm{E}[T_e(x)]$ is proportional to the file size x. Furthermore, the above results for the mean file transfer delay $\mathrm{E}[T_e(x)]$, and the blocking probability p_e depend on the file size distribution only through its mean value: The results are insensitive to higher moments of the distribution.
When we take $r_e^- = 0$ (i.e. really 'best effort' traffic) and $r_e^+ \geq C_e$, our model for elastic traffic becomes the 'standard' M/G/1 processor sharing queue. In that case, the above formula for the mean file transfer delay reduces to the well known M/G/1 processor sharing result (see for instance Kleinrock [9, Formula 4.17]):

$$\mathrm{E}[T_e(x)] = (x/C_e)\frac{\lambda_e f_e}{1-\lambda_e f_e}.$$

For the M/G/1 processor sharing queue, expressions have been found for the Laplace Stieltjes Transform of the distribution of the conditional transfer time $T_e(x)$, see for instance Yashkov [24], Ott [16], Schassberger [20], and Van den Berg and Boxma [2].

3.3 Integrated scenario

In the model with complete integration of the two traffic types, a new call (of any type) is accepted if the guaranteed performance is not violated for any connection. Thus, the state space is given by

$$S = S^{(\mathrm{int})} := \left\{(n_e, n_s) \in \mathbb{N}_0 \times \mathbb{N}_0 : n_e r_e^- + n_s r_s \leq C\right\}.$$

When $n_e > 0$, the transfer rate of each elastic traffic connection is

$$r_{n_e,n_s} := \min\left(r_e^+, \frac{C - n_s r_s}{n_e}\right). \tag{3.6}$$

i.e. the capacity available to elastic traffic is divided equally among all elastic traffic connections, but never exceeding the maximum rate r_e^+ per connection.
Our assumptions on Poisson arrivals and exponentially distributed file lengths (for elastic traffic) and holding times (for stream traffic), ensure that the pair $(X_e(t), X_s(t))$ is a Markov process.
Denote the maximum number of elastic traffic connections and stream traffic connections, by K_e and K_s, respectively. Furthermore, define the maximum number of stream traffic connections when there are n_e elastic traffic connections, by

$$K_s^{(n_e)} := \left\lfloor \frac{C - n_e r_e^-}{r_s} \right\rfloor, \qquad n_e = 0, 1, \ldots, K_e.$$

Obviously, $K_s = K_s^{(0)}$.
With the pairs (n_e, n_s) ordered lexicographically, the generator of the process $(X_e(t), X_s(t))$ is given by

$$\mathcal{Q}^{(\mathrm{int})} := \begin{bmatrix} Q_d^{(0)} & \lambda_e I^{(0)} & 0 & \ldots & \ldots & 0 \\ M^{(1)} & Q_d^{(1)} & \lambda_e I^{(1)} & 0 & \ldots & \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & & M^{(K_e-1)} & Q_d^{(K_e-1)} & \lambda_e I^{(K_e-1)} \\ 0 & \ldots & & 0 & M^{(K_e)} & Q_d^{(K_e)} \end{bmatrix}. \tag{3.7}$$

Here, $\mathcal{Q}^{(\mathrm{int})}$ consists of K_e+1 block rows and block columns. The sizes of the blocks are not fixed. The matrices $I^{(n_e)}$, $n_e = 0, 1, \ldots, K_e - 1$, are of dimension $(K_s^{(n_e)}+1) \times (K_s^{(n_e+1)}+1)$. Its entries are given by $[I^{(n_e)}]_{n_s,n_s} = 1$, $n_s = 0, 1, \ldots, K_s^{(n_e+1)}$, and zero in all other positions. The dimension of $M^{(n_e)}$,

$n_e = 1, 2, \ldots, K_e$, is $(K_s^{(n_e)}+1) \times (K_s^{(n_e-1)}+1)$, with $[M^{(n_e)}]_{n_s,n_s} = n_e r_{n_e,n_s}/f_e$, and all other entries equal to zero. Finally, the matrices $Q_d^{(n_e)}$, $n_e = 0, 1, \ldots, K_e$, are of dimension $(K_s^{(n_e)}+1) \times (K_s^{(n_e)}+1)$. Except for the diagonal elements, $Q_d^{(n_e)}$ is equal to the generator of the queue length process of the $M/M/K_s^{(n_e)}/K_s^{(n_e)}$ model. The diagonal elements are such that each row of $\mathcal{Q}^{(\mathrm{int})}$ sums up to 0.

3.4 Mixed scenario

As in the model with complete segregation, a fixed capacity $C_e > 0$ is exclusively reserved for elastic traffic. The remaining capacity $C_s > 0$ is primarily dedicated to stream traffic, but whenever stream traffic connections do not 'fill' the capacity C_s, elastic traffic may use the spare capacity. However, this capacity is immediately allocated to stream traffic, as soon as a new stream traffic connection is requested. Therefore the capacity C_e should always be sufficient to guarantee the minimum transfer rate r_e^- to each proceeding elastic traffic call. Hence, the state space of the process $(X_e(t), X_s(t))$ is the same as for the segregated model: $S^{(\mathrm{mix})} = S^{(\mathrm{seg})}$, see (3.3). The transfer rate r_{n_e,n_s} of an elastic traffic connection is, as in the integrated model, given by (3.6), with $C = C_e + C_s$. Of course, the process $(X_e(t), X_s(t))$ is again a Markov process.
As in Section 3.3, we denote the maximum numbers of elastic traffic connections and stream traffic connections by K_e and K_s, respectively. The number of states in $S^{(\mathrm{mix})}$ is $(K_e + 1) \times (K_s + 1)$.
Since the elastic traffic does not affect the stream traffic, $X_s(t)$ evolves as the queue length process of the standard Erlang loss model, just as in the segregated model.
The process $(X_e(t), X_s(t))$ is a finite inhomogeneous Quasi Birth Death (QBD) process. Its generator $\mathcal{Q}^{(\mathrm{mix})}$ has the same structure as $\mathcal{Q}^{(\mathrm{int})}$ in (3.7). However, this time the sizes of the blocks are all equal: The matrices $I^{(n_e)}$, $M^{(n_e)}$, and $Q_d^{(n_e)}$, are of dimension $(K_s + 1) \times (K_s + 1)$. The matrices $I^{(n_e)}$, $n_e = 0, 1, \ldots, K_e - 1$ do not depend on n_e, and are equal to the identity matrix. $M^{(n_e)}$, $n_e = 1, 2, \ldots, K_e$ is the diagonal matrix $\frac{n_e}{f_e}\mathrm{diag}[r_{n_e,0}, r_{n_e,1}, \ldots, r_{n_e,K_s}]$. For convenience of notation, we set $M^{(0)}$ equal to the null matrix. Then, for $n_e = 0, 1, \ldots, K_e - 1$, $Q_d^{(n_e)} = Q_s - \lambda_e I - M^{(n_e)}$, and $Q_d^{(K_e)} = Q_s - M^{(K_e)}$, where Q_s is the (tri-diagonal) infinitesimal generator of the queue length process of the standard Erlang loss model.

4. Analysis of the conditional mean transfer time

Once the steady-state probabilities have been determined (e.g. using the method in De Nitto Personè and Grassi [8]), the mean sojourn time $\mathrm{E}[T_e]$ is easily computed, see the remark following (3.2). However, for elastic traffic we are also interested in $\mathrm{E}[T_e(x)]$, the mean transfer time of an *accepted* file with given length x. Recall that, in the segregated model, $\mathrm{E}[T_e(x)]$ is proportional to x, see (3.5). In this section we analyse $\mathrm{E}[T_e(x)]$ in both the integrated and the mixed model. For details on the analysis, for interpretation of various entities, and for full proofs in this section, we refer to Núñez Queija [14].
As in Section 3.1, we denote the state space generically by S. Thus, either $S = S^{(\mathrm{int})}$ or $S = S^{(\mathrm{mix})}$. Let $S^* := \{(n_e, n_s) \in S : n_e > 0\}$. We restrict ourselves to the case where $r_{n_e,n_s} > 0$ for all $(n_e, n_s) \in S^*$. Note that this condition is automatically satisfied when $r_e^- > 0$. For the mixed strategy, the condition is also satisfied when $C_e > 0$. The case with $r_{n_e,n_s} = 0$ for some $(n_e, n_s) \in S^*$, can be treated in a similar way, see Núñez Queija [14].
For $(n_e, n_s) \in S^*$, and $x \geq 0$, we introduce the following conditional expectation:

$\beta_{n_e,n_s}(x) =$ the expected transfer time of a (non-blocked) file of length x, starting with $n_e - 1$ other proceeding elastic traffic connections and n_s stream traffic connections.

Let $\overline{\beta}(x)$ be the vector $(\beta_{n_e,n_s}(x))_{(n_e,n_s)\in S^*}$, where the $\beta_{n_e,n_s}(x)$ are ordered lexicographically. Note that we exclude blocked elastic traffic calls. We may now write

$$\mathrm{E}[T_e(x)] = \frac{1}{1-p_e} \sum_{(n_e,n_s)\in S^*} \pi_{n_e-1,n_s}\beta_{n_e,n_s}(x). \tag{4.1}$$

We now study the functions $\beta_{n_e,n_s}(x)$. First we formulate a system of differential equations and initial conditions, from which we find the $\beta_{n_e,n_s}(x)$. Then we show that these functions converge to a linear function, as $x \to \infty$. Finally, we indicate how the $\beta_{n_e,n_s}(x)$ can be evaluated numerically.

Lemma 4.1 *The functions $\beta_{n_e,n_s}(x)$, $(n_e, n_s) \in S^*$, satisfy the following system of differential equations and initial conditions:*

$$\begin{aligned}
r_{n_e,n_s}\frac{\partial}{\partial x}\beta_{n_e,n_s}(x) \quad &= \quad 1 + \mathbf{1}_{n_e,n_s+1}\lambda_s\beta_{n_e,n_s+1}(x) + \frac{n_s}{h_s}\beta_{n_e,n_s-1}(x) \\
&\quad\ + \mathbf{1}_{n_e+1,n_s}\lambda_e\beta_{n_e+1,n_s}(x) + \frac{n_e-1}{f_e}r_{n_e,n_s}\beta_{n_e-1,n_s}(x) \\
&\quad\ - \left(\mathbf{1}_{n_e+1,n_s}\lambda_e + \frac{n_e-1}{f_e}r_{n_e,n_s} + \mathbf{1}_{n_e,n_s+1}\lambda_s + \frac{n_s}{h_s}\right)\beta_{n_e,n_s}(x). \qquad (4.2)\\
\beta_{n_e,n_s}(0) \quad &:= \quad \lim_{x\downarrow 0}\beta_{n_e,n_s}(x) = 0. \qquad (4.3)
\end{aligned}$$

Here, the indicator function $\mathbf{1}_{n_e,n_s}$ is 1 if $(n_e, n_s) \in S^*$, and 0 otherwise.
Equivalently, we may write in matrix notation:

$$\mathcal{R}\frac{\partial}{\partial x}\overline{\beta}(x) = \overline{\mathrm{e}} + \mathcal{Q}^*\overline{\beta}(x), \qquad \overline{\beta}(0) = \overline{0}. \tag{4.4}$$

In Lemma 4.1, $\overline{\mathrm{e}}$ is a vector with all elements equal to 1. $\mathcal{R}$ is the diagonal matrix, with the diagonal entries being the lexicographically ordered r_{n_e,n_s}. $\mathcal{Q}^*$ is the generator of a Markov process with a similar structure as $\mathcal{Q}^{(\mathrm{int})}$ (or $\mathcal{Q}^{(\mathrm{mix})}$) in (3.7).

Proof of Lemma 4.1
We show the validity of (4.2) for $(n_e, n_s) \in S^*$ such that $(n_e + 1, n_s) \in S^*$ and $(n_e, n_s + 1) \in S^*$. In all other cases, similar arguments can be used. By conditioning on the events that occur in a small time interval of length Δ, we may write, for $\Delta \downarrow 0$:

$$\begin{aligned}
\beta_{n_e,n_s}(x) \quad &= \quad \Delta + \lambda_e\Delta\beta_{n_e+1,n_s}(x - \mathrm{O}(\Delta)) + \frac{n_e-1}{f_e}r_{n_e,n_s}\Delta\beta_{n_e-1,n_s}(x-\mathrm{O}(\Delta)) \\
&\quad\ + \lambda_s\Delta\beta_{n_e,n_s+1}(x-\mathrm{O}(\Delta)) + \frac{n_s}{h_s}\Delta\beta_{n_e,n_s-1}(x-\mathrm{O}(\Delta)) \\
&\quad\ + \left(1 - \lambda_e\Delta - \frac{n_e-1}{f_e}r_{n_e,n_s}\Delta - \lambda_s\Delta - \frac{n_s}{h_s}\Delta\right)\beta_{n_e,n_s}(x - r_{n_e,n_s}\Delta) + \mathrm{o}(\Delta).
\end{aligned}$$

Rearranging terms, and letting $\Delta \downarrow 0$, we have the desired differential equation.
The initial condition follows from the fact that we assumed that $r_{n_e,n_s} > 0$, for all $(n_e, n_s) \in S^*$. Therefore, once an elastic traffic call is accepted, its transfer can start immediately. □
The system of differential equations and initial conditions in Lemma 4.1, uniquely determine the functions $\beta_{n_e,n_s}(x)$, $x \geq 0$. The solution is given in the next theorem, see also Núñez Queija [14].

Theorem 4.1 *Let $\overline{\pi}^* = (\pi^*_{n_e,n_s})_{(n_e,n_s)\in S^*}$ be the steady-state distribution vector corresponding to the generator $\mathcal{Q}^*$: I.e., $\overline{\pi}^*\mathcal{Q}^* = \overline{0}$. Define,*

$$\begin{aligned}
c^* \quad &:= \sum_{(n_e,n_s)\in S^*} n_e r_{n_e,n_s}\pi^*_{n_e,n_s}, \\
p_e^* \quad &:= \sum_{(n_e,n_s)\in S^*:(n_e+1,n_s)\notin S^*} \pi^*_{n_e,n_s}.
\end{aligned}$$

Let $\overline{\gamma} = (\gamma_{n_e,n_s})_{(n_e,n_s)\in S^*}$ *be the unique solution to,*

$$\begin{aligned}-\mathcal{R}^{-1}\mathcal{Q}^*\overline{\gamma} &= \mathcal{R}^{-1}\overline{\mathrm{e}} - \frac{1}{c^* - \lambda_e f_e(1-p_e^*)}\overline{\mathrm{e}},\\ \overline{\pi}^*\mathcal{R}\overline{\gamma} &= 0.\end{aligned}$$

Then the unique solution to (4.4) is given by:

$$\overline{\beta}(x) = \frac{x}{c^* - \lambda_e f_e(1-p_e^*)}\overline{\mathrm{e}} + \left[I - \exp\left\{x\mathcal{R}^{-1}\mathcal{Q}^*\right\}\right]\overline{\gamma}. \tag{4.5}$$

The entities c^* and p_e^* have the following intuitive meaning: In a system with one *permanent* elastic traffic connection, c^* is the average capacity assigned to elastic traffic per time unit; and p_e^* is the blocking probability of new elastic traffic calls.
The existence and uniqueness of $\overline{\gamma}$ is a well known result from Markov decision theory: The numbers γ_{n_e,n_s} can be interpreted as relative costs in a Markov process with generator $\mathcal{R}^{-1}\mathcal{Q}^*$, see for instance Tijms [21, Theorem 3.1.1 and p. 220]. Solution (4.5) can be checked by substitution in (4.4).
Note that, from Theorem 4.1 and Expression (4.1), we have an explicit expression for $\mathrm{E}[T_e(x)]$ in terms of x. At the end of this section we indicate how this expression can be used for computation of $\mathrm{E}[T_e(x)]$. First, however, we establish a relevant limiting result for $\overline{\beta}(x)$ and $\mathrm{E}[T_e(x)]$ as $x \to \infty$.

Corollary 4.2 *For all* $(n_e,n_s) \in S^*$,

$$\lim_{x\to\infty} \beta_{n_e,n_s}(x) - \frac{x}{c^* - \lambda_e f_e(1-p_e^*)} = \gamma_{n_e,n_s},$$

and hence

$$\lim_{x\to\infty} \mathrm{E}[T_e(x)] - \frac{x}{c^* - \lambda_e f_e(1-p_e^*)} = \frac{1}{1-p_e}\sum_{(n_e,n_s)\in S^*} \pi_{n_e-1,n_s}\gamma_{n_e,n_s}.$$

Corollary 4.2 follows from the fact that $\mathcal{R}^{-1}\mathcal{Q}^*$ is the generator of a finite, irreducible Markov process: Its largest eigenvalue is 0, has multiplicity 1 and corresponding left null vector $\overline{\pi}^*$ and right null vector $\overline{\mathrm{e}}$.

Numerical evaluation of the conditional mean transfer time
To compute $\mathrm{E}[T_e(x)]$, one may use Expressions (4.5) and (4.1). The term $\exp\left\{x\mathcal{R}^{-1}\mathcal{Q}^*\right\}\overline{\gamma}$ can be evaluated in a numerically stable way, by using uniformisation: Let $\eta > 0$ be such that $\mathcal{P} := I + \frac{1}{\eta}\mathcal{R}^{-1}\mathcal{Q}^*$ is a non-negative matrix. Then, $\mathcal{P}$ is a stochastic matrix that can be associated with the uniformised jump process of the Markov process governed by $\mathcal{R}^{-1}\mathcal{Q}^*$. Now,

$$\exp\left\{x\mathcal{R}^{-1}\mathcal{Q}^*\right\} = \mathrm{e}^{-\eta x}\exp\left\{\eta x\mathcal{P}\right\} = \mathrm{e}^{-\eta x}\sum_{k=0}^{\infty}\frac{(\eta x)^k}{k!}\mathcal{P}^k,$$

and the terms in this expression only involve non-negative numbers.
As an alternative, we may use (4.4) directly to compute recursively the coefficients of the Taylor series of $\overline{\beta}(x)$ around 0. Again, this should be done using $\mathcal{P}$ instead of $\mathcal{Q}^*$. The advantage of this alternative is that $\overline{\gamma}$ need not be computed. In Experiment 3 of Section 5, we used both methods when computing $\mathrm{E}[T_e(x)]$. In all cases the relative difference between the outcomes of both methods was of the order of 10^{-8}, or smaller.

5. Numerical results

Using the analysis presented in Sections 3 and 4, we performed an extensive numerical study on the integration policies defined in Section 2. In this section, we present some of our results.

Link	C (all models)	155	Mbit/s
	C_e (*not* for integr.)	105–80–55–30–5	Mbit/s
Elastic traffic	f_e	50	Mbit
	r_e^-	0	Mbit/s
	r_e^+	10–50–155	Mbit/s
Stream traffic	h_s	10	sec.
	r_s	5	Mbit/s

Table 1: Parameters in Experiment 1

C_s	50	75	100	125	150
λ_s	0.446118	0.810804	1.203062	1.612456	2.033728

Table 2: Load of stream traffic (in terms of λ_s) for the mixed and segregated strategies; $p_s = 0.01$

It should be emphasised that quite a number of parameters play a role in our model. This, of course, makes it impossible to draw general conclusions over the entire parameter space. In order to cope with that, we fix a number of parameters at a realistic value. The 'guaranteed rate' r_e^- for elastic traffic is taken equal to zero, in the first three experiments. Notice that the $r_e^- = 0$ assumption relates e.g. to the most likely next Internet situation with two traffic classes: high priority (stream) traffic and low priority best effort (elastic) traffic without any bandwidth guarantee. In the fourth experiment $r_e^- > 0$, which relates for example to the situation of an ATM network with ABR connections having an MCR (Minimum Cell Rate) larger than zero, or to a future IP network with appropriate packet scheduling and flow admission control mechanisms in the routers (see e.g. Roberts [18]).

Experiment 1

In our first experiment, we compare the efficiency of the three scenarios (segregated, integrated and mixed). More precisely stated, given certain performance requirements of the two traffic types (mean file transfer time $\mathrm{E}[T_e]$ for elastic traffic and blocking probability p_s for stream traffic calls), we determine the maximum traffic load that can be handled under the three different strategies. Table 1 shows the model parameters.

We have chosen the traffic parameters such that the calls of the two traffic types have the same mean size (i.e. have the same mean number of bits to be transferred): $f_e = h_s r_s$. For various values of the parameters $C_s = C - C_e$ and r_e^+, we evaluated the efficiency of each of the three strategies in the following way: We have chosen λ_s such that the blocking probability p_s of the stream traffic calls for the mixed and segregated strategy equals 0.01 (note that λ_s can be easily computed from the Erlang loss formula), see Table 2. In order to make a fair comparison, in the integrated scenario we have reduced the value of λ_s, such that the amount of *accepted* stream traffic is equal for all three strategies. Then, given a certain load of stream traffic (in terms of λ_s), we determined for each of the three strategies the maximum possible load of elastic traffic (in terms of λ_e), such that $\mathrm{E}[T_e] = h_s = 10$. The results of this first experiment are shown in Figure 1. For $C_e = 5$ the allowed λ_e is smaller than 10^{-5}.

As expected, the mixed and integrated strategies are considerably more efficient than the segregated strategy: apparently, the elastic traffic benefits highly from the fluctuating amount of bandwidth that is left over by the stream traffic. The differences between the mixed strategy and the integrated strategy are very small. In all cases, the mixed strategy is at least as efficient as the integrated strategy. Finally it is noted that the impact of r_e^+ on the efficiency of the strategies is very small. This is due

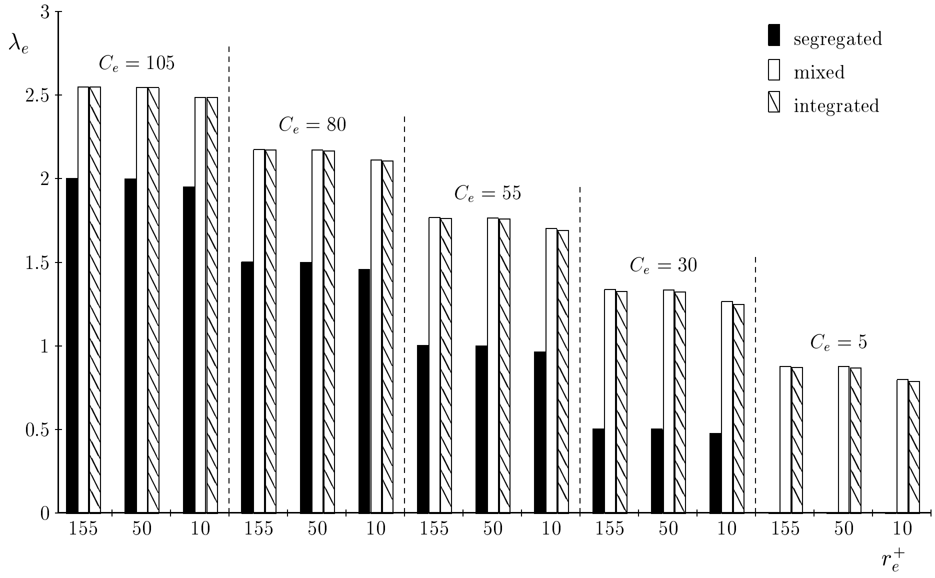

Figure 1: Efficiency of the three strategies (in terms of λ_e)

to the fact that the system is highly loaded: the number of elastic traffic calls simultaneously present in the system is most of the time that large, that each of them receives less than 10 Mbit/s of the total available capacity (hence, it makes no difference whether $r_e^+ = 10, 50$ or 155 Mbit/s).

Experiment 2

In the previous experiment, stream traffic calls and elastic traffic calls arrive/depart at more or less the same time scale. What if this is not the case, i.e. what if the stream traffic fluctuates much faster or much slower than the elastic traffic? To investigate this, we repeated Experiment 1 for the cases $h_s = 1$ (rapidly fluctuating stream traffic) and $h_s = 100$ (slowly fluctuating stream traffic). All other parameters in Table 1 remain unchanged. Note that the values of λ_s in Table 2 are multiplied by a factor 10 (in case $h_s = 1$), and by a factor 0.1 (in case $h_s = 100$), such that p_s remains equal to 0.01 in the mixed and segregated strategies. We observed that in all cases the segregated strategy is the least efficient, and that the mixed strategy outperforms the integrated strategy (particularly when $h_s = 100$). For the mixed strategy, being the most efficient in all cases, the impact of the time scale difference is reported in Figure 2. Intuitively, one expects that when the stream traffic fluctuates very fast, the performance of elastic traffic is the same as for the segregated scenario with C_e equal to the mean available capacity $C - \lambda_s(1 - p_s)h_s r_s$. In Figure 2, also the values of λ_e are given for that case. This phenomenon was already noted in Núñez Queija and Boxma [15] and Altman et al. [1], and formally proved in Núñez Queija [13].
The numerical results show that λ_e increases when the stream traffic fluctuates faster (i.e. h_s becomes smaller). Note that, as expected, the difference between the mixed scenario with $h_s = 1$ (i.e. stream traffic fluctuates relatively fast) and the segregated scenario with $C_e = C - \lambda_s(1 - p_s)h_s r_s$ is negligible. As in the previous experiment, it is seen that the impact of r_e^+ on the efficiency is very small.

Experiment 3

For the mixed strategy, we consider the conditional mean file transfer time $\mathrm{E}[T_e(x)]$ as a function of the file size x. In particular, we are interested in how fast $\mathrm{E}[T_e(x)]$ converges to its linear asymptote (as $x \to \infty$). The parameters f_e, r_e^-, h_s and r_s are fixed at their respective values given in Table 1, and C_e is set equal to 80 (therefore the condition $r_{n_e,n_s} > 0$ in Section 4 is satisfied). The value of

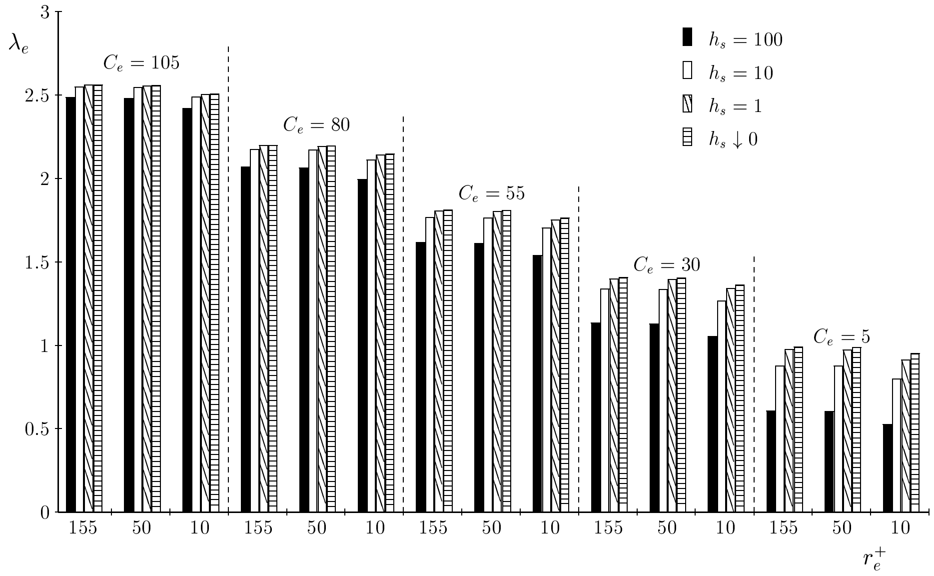

Figure 2: Efficiency of the mixed strategy (in terms of λ_e) on different time scales

λ_s (0.81) is again chosen such that $p_s = 0.01$, and λ_e is fixed at 2.17, which is the value computed in Experiment 1 with $r_e^+ = \infty$. In Figure 3, $\mathrm{E}[T_e(x)]$ is given for the three values of r_e^+. We observe that $\mathrm{E}[T_e(x)]$ is considerably smaller for larger values of r_e^+. We also computed the asymptote of $\mathrm{E}[T_e(x)]$; for $r_e^+ = 10$ the results are shown in Figure 4. For the other two values of r_e^+, we obtained similar figures, the distance between the actual curve and the asymptote being larger for larger r_e^+.
Keeping λ_e fixed, we repeated the above experiment for rapidly fluctuating stream traffic ($h_s = 1$) and for slowly varying stream traffic ($h_s = 100$). As in Experiment 2, the value of λ_s when $h_s = 1$ (resp. $h_s = 100$) is found by multiplication by a factor 10 (resp. 0.1), such that the traffic load of stream traffic (in terms of $\lambda_s h_s$) is the same in all cases. In both cases, the results yield graphs (not shown in this paper) similar to the ones in Figures 3 and 4. However, for 'fast' stream traffic we observed that the distance between $\mathrm{E}[T_e(x)]$ and its asymptote is considerably smaller, and that for 'slow' stream traffic this distance is very large.
The results show that in general the asymptote does not give a useful approximation for $\mathrm{E}[T_e(x)]$. An additional numerical study indicates that a good approximation is provided by the tangent of the curve in the origin. In Figure 4, for values of x smaller than five times the mean file size $f_e = 50$ Mbit, the relative difference between $\mathrm{E}[T_e(x)]$ and the tangent in zero is less than 2.5%. Note that the slope of this tangent line can be easily computed from the steady-state distribution, see Núñez Queija [14].

Experiment 4

In our last experiment we consider the situation that the elastic traffic calls are guaranteed a certain minimum bandwidth r_e^-. For the mixed strategy, we study the impact of C_s on the call blocking probabilities p_e and p_s of the elastic traffic and the stream traffic, respectively. As before, we choose $f_e = 50$ Mbit and $r_s = 5$ Mbit/s. Furthermore, $h_s = 10$ sec., $r_e^- = 5$ Mbit/s (i.e. the transfer time of a file of size x Mbit is bounded by $x/5$ seconds), and $r_e^+ = 155$ Mbit/s. We fix the call arrival intensities at $\lambda_e = 1.90$ and $\lambda_s = 1.15$. These values are chosen such that $p_e = p_s = 0.05$ in the mixed scenario with $C_e = 75$ Mbit/s. The results are shown in Figure 5. It is seen that the call blocking probability for the stream traffic decreases very rapidly when C_s increases, while the blocking probability for the elastic traffic grows only moderately. Note that, as C_s increases, the amount of bandwidth ($C_e = C - C_s$) *reserved* for elastic traffic decreases. A part of this reassigned bandwidth is

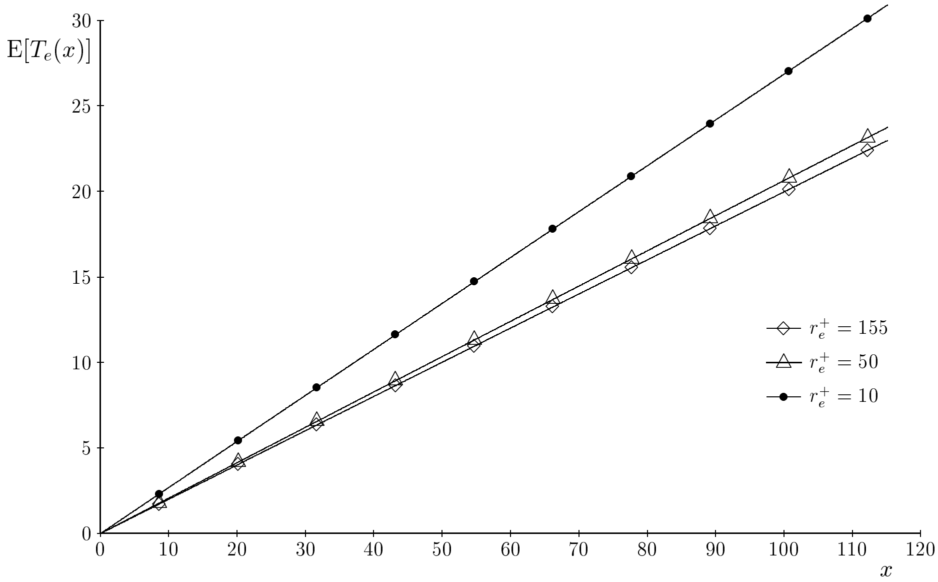

Figure 3: Conditional mean delay for $h_s = 10$, and $r_e^+ = 10, 50, 155$

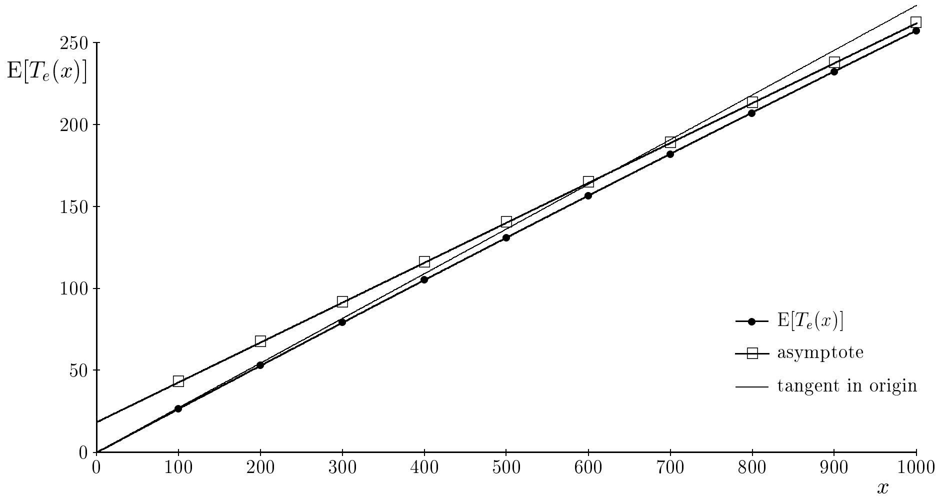

Figure 4: $\mathrm{E}[T_e(x)]$ and its asymptote for $h_s = 10$ and $r_e^+ = 10$

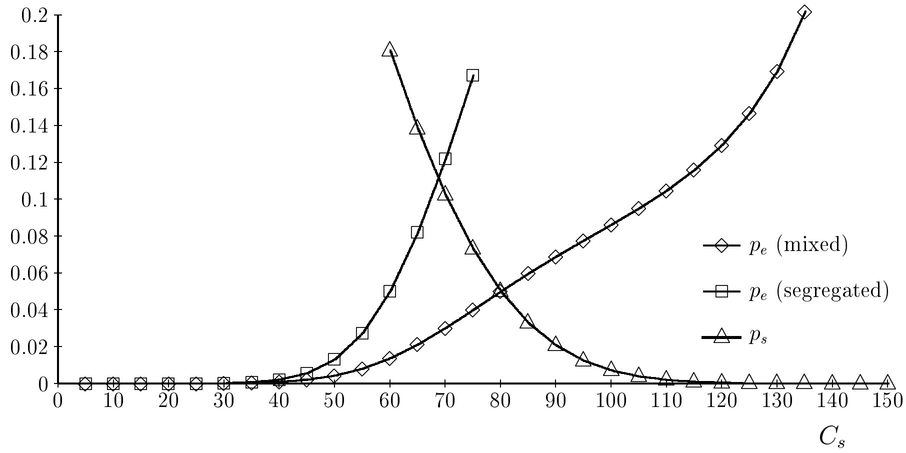

Figure 5: Blocking probabilities for different choices of C_s

however not used by the stream traffic. This amount of bandwidth, $C_s - \lambda_s(1-p_s)h_s r_s$, allocated to, but not used by the stream traffic, is apparently very well exploited by the elastic traffic calls. This is confirmed by the results for the loss probability of elastic traffic calls in the corresponding segregated case, which are also shown in the figure.

6. Conclusions and directions for future research

In this paper we studied the integration of stream traffic and elastic traffic in one single network, e.g. an ATM-based or an IP-based network. First, models were developed describing different integration strategies. Then we presented analytical techniques for obtaining performance measures, in particular call blocking probabilities and file transfer delays. Finally, these methods were used for assessing and comparing the efficiency gains achieved by the integration strategies.

Integration of stream and elastic traffic

The first conclusion is that integration of stream and elastic traffic in one single network is much more efficient (with respect to the use of network resources) than having two dedicated networks for the two traffic types (i.e. segregation). The so-called mixed scenario is slightly more efficient than the integrated scenario, and has the additional advantage of offering calls of both types a certain 'protection' against each other, when calls of one type generate (temporarily) a relatively large load. For other integration schemes – like trunk reservation – the analysis and computation of the performance measures can be done in a similar way; comparison with the integration strategies considered in this paper would be an interesting issue for further research.

Analytical techniques

We demonstrated that the relevant performance measures can be analysed and efficiently calculated in a numerically stable way. In particular, we developed a technique for evaluating the mean transfer time $\mathrm{E}[T_e(x)]$ of an 'elastic' file of given length x. Our numerical study showed that, for values of x up to four or five times the mean file size, a good approximation of $\mathrm{E}[T_e(x)]$ is provided by the tangent line in the origin; the slope of this tangent line can be easily determined from the steady-state distribution.

A possible direction for further research is the following. In the present study file lengths are (mostly) assumed to have an exponential distribution. This assumption allowed for a detailed analysis of the impact of the interaction between both traffic types on their performance. Extension of our analysis to phase type distributions is possible and we expect that similar results hold. However, extension to the case of file size distributions with 'heavy tails', e.g., the Pareto distribution, is not straightforward. It would be useful to be able to compute the relevant performance measures under this modelling assumption, cf. Zwart and Boxma [27] for the case that only elastic traffic calls share the link bandwidth. An interesting question then is whether our conclusions regarding integration/segregation still hold. In particular, can $\mathrm{E}[T_e(x)]$ (for quite large values of x) still be approximated by its tangent in the origin and does it converge to a linear function when x grows to infinity?

Acknowledgement

The authors would like to thank S.C. Borst, O.J. Boxma, R.E. Kooij, and A.P. Zwart for carefully reading previous versions, and for providing comments that led to improvement of the paper.

References

1. E. Altman, D. Artiges, K. Traore [1997]. On the integration of best-effort and guaranteed performance services. *INRIA Research Report*, n. 3222.
2. J.L. van den Berg, O.J. Boxma [1991]. The M/G/1 queue with processor sharing and its relation to a feedback queue. *Queueing Systems*, Vol. 9, 365–401.
3. S. Blaabjerg, G. Fodor, E. Nordstrom [1997]. A partial blocking queueing system with CBR/VBR and ABR/UBR arrival streams. *Proceedings 5th International Conference on Telecommunications, Nashville.*
4. C. Blondia, O. Casals [1996]. Throughput analysis of the Explicit Rate congestion control mechanism in ATM networks. *Proceedings 10th ITC Specialists Seminar, Lund.*
5. F. Bonomi, D. Mitra, J. Seery [1995]. Adaptive algorithms for feedback-based flow control in high speed wide-area networks. *IEEE J. on Selected Areas in Communications*, Vol. 13, 1267–1283.
6. E.G. Coffman, R.R. Muntz, H. Trotter [1970]. Waiting time distributions for processor sharing systems. *J. of the ACM*, Vol. 17, 123–130.
7. J.W. Cohen [1979]. The multiple phase service network with generalized processor sharing. *Acta Informatica*, Vol. 12, 245–284.
8. V. De Nitto Personè, V. Grassi [1996]. Solution of finite QBD processes. *J. Applied Probability*, Vol. 33, 1003–1010.
9. L. Kleinrock [1976]. *Queueing Systems, Vol. II: Computer Applications.* Wiley, New York.
10. K. Lindberger [1997]. Effectiveness and quality of service for ABR traffic. COST 257, TD (97) 25.
11. T.V. Lakshman, U. Madhow [1994]. Performance analysis of window-based flow control using TCP/IP: Effect of high bandwidth-delay products and random loss. *High Performance Networking, V: Proceedings of the IFIP TC6/WG6.4 fifth international conference, Grenoble.* Ed. S. Fdida, Elsevier Science B.V., Amsterdam.
12. M.F. Neuts [1981]. *Matrix-geometric Solutions in Stochastic Models – An Algorithmic Approach.* The Johns Hopkins University Press, Baltimore.
13. R. Núñez Queija [1998]. A queueing model with varying service rate for ABR. *Computer Performance Evaluation — Modelling Techniques and Tools: Proceedings of TOOLS '98, Mallorca.*

Eds. R. Puigjaner, N.N. Savino, and B. Serra, Springer Verlag, Berlin.

14. R. Núñez Queija [1998]. Sojourn times in non-homogeneous QBD processes with processor sharing. *CWI Report,* PNA-R9901.
`http://www.cwi.nl/static/publications/reports/abs/PNA-R9901.html.`
15. R. Núñez Queija, O.J. Boxma [1998]. Analysis of a multi-server queueing model of ABR. *J. Applied Mathematics and Stochastic Analysis*, Vol. 11, 339–354.
16. T.J. Ott [1984]. The sojourn time distributions in the $M/G/1$ queue with processor sharing. *J. Applied Probability*, Vol. 21, 360–378.
17. M. Ritter [1997]. Congestion Detection Methods and their Impact on the Performance of ABR Flow Control. *Teletraffic Contributions for the Information Age: Proceedings of the ITC 15, Washington DC.* Eds. V. Ramaswami and P.E. Wirth, Elsevier Science B.V., Amsterdam.
18. J.W. Roberts [1998]. Quality of service guarantees and charging in multiservice networks. *IEICE Transactions on Communications.*
19. J.W. Roberts [1998]. Realizing quality of service guarantees in multiservice networks. *Proceedings of IFIP Seminar PMCCN '97*, Chapman and Hall.
20. R. Schassberger [1984]. A new approach to the $M/G/1$ processor sharing queue. *Adv. Applied Probability*, Vol. 16, 202–213.
21. H.C. Tijms [1994]. *Stochastic Models – An Algorithmic Approach.* Wiley, Chichester.
22. K. van der Wal, M. Mandjes, H. Bastiaansen [1997]. Delay performance analysis of the new Internet services with guaranteed QoS. *Proceedings of the IEEE*, Vol. 85, 1947–1957.
23. P. White, J. Crowcroft [1997]. The integrated services in the Internet: State of the art. *Proceedings of the IEEE*, Vol. 85, 1934–1946.
24. S.F. Yashkov [1983]. A derivation of response time distribution for a M/G/1 processor sharing queue. *Problems in Control and Information Theory*, Vol. 12, 133–148.
25. S.F. Yashkov [1987]. Processor–sharing queues: Some progress in analysis. *Queueing Systems*, Vol. 2, 1–17.
26. S.F. Yashkov [1992]. Mathematical problems in the theory of processor-sharing queueing systems. *J. Soviet Mathematics*, Vol. 58, 101—147.
27. A.P. Zwart, O.J. Boxma [1998]. Sojourn time asymptotics in the M/G/1 processor sharing queue. *CWI Report,* PNA-R9802.
`http://www.cwi.nl/static/publications/reports/abs/PNA-R9802.html.`