Available at
www.ComputerScienceWeb.com
POWERED BY SCIENCE @ DIRECT®

ELSEVIER

Computer Networks 42 (2003) 521–536

COMPUTER NETWORKS

www.elsevier.com/locate/comnet

Congestion at flow level and the impact of user behaviour

T. Bonald, J.W. Roberts *

France Telecom R&D, DAC/OAT, 38-40 Rue du general Leclerc, 92794 Issy les Moulineaux Cedex, France

Received 31 July 2002; accepted 18 January 2003

Responsible Editor: Z.-L. Zhang

Abstract

So-called elastic flows, corresponding to document transfers of various types, constitute the bulk of Internet traffic. This paper presents models of a single bottleneck link handling elastic traffic, accounting for random flow arrivals. The transport protocol and packet scheduling are taken into account approximately by assuming perfectly realized bandwidth sharing objectives.

We refer to the demand as the product of the flow arrival rate and the average flow size. It is shown that per-flow throughput performance is generally satisfactory as long as demand is only slightly less than capacity. In overload, on the other hand, some flows must be abandoned. A fraction of link bandwidth is then wasted and performance critically depends on user behaviour. The models are useful in appraising the effectiveness of proposed schemes for Internet service differentiation.

© 2003 Elsevier Science B.V. All rights reserved.

Keywords: Flow-level modelling; Elastic traffic; Overload; Service differentiation

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 for both real time communication and playback of stored sequences. Their quality of service is mainly determined by the degree of transparency of the communication path with respect to the integrity of the initial signal. Elastic flows, on the other hand, result from the transfer of digital documents (Web pages, MP3, emails,...) and their transmission rate is adaptable depending on current traffic levels. Rate and duration are thus measures of quality of service rather than intrinsic flow characteristics. In this paper we consider elastic flows exclusively, ignoring for the sake of simplicity the impact on available capacity of streaming traffic.

Flows using the core network are typically generated by a very large population of users independently communicating with an equivalently large population of servers and correspondents for a variety of different applications. The net result of

[*] Corresponding author. Tel.: +33-1-45-29-5701; fax: +33-1-45-29-6069.

E-mail address: james.roberts@rd.francetelecom.com (J.W. Roberts).

1389-1286/03/$ - see front matter © 2003 Elsevier Science B.V. All rights reserved.
doi:10.1016/S1389-1286(03)00200-7

all this activity is a traffic process which can only be described in statistical terms. While for streaming flows it is common practice to study traffic as a stochastic process, when evaluating the performance of adapted admission control and routing strategies, for example, this approach has only recently been developed for elastic traffic (see [2] and cited works). In this paper we evaluate a number of simple flow level models for elastic traffic which illustrate some fundamental performance characteristics, notably with respect to the potential for QoS discrimination in a Diffserv architecture.

For present purposes, each elastic flow is assumed to correspond to the transfer of a single document, characterized simply as a volume of data in bits. Such a flow is not materialized in a datagram network. However, it makes sense to model traffic in terms of flows since QoS, measured in terms of response time, is experienced at this level. It is also the case that all the packets of a given flow generally follow the same network path and response time is largely determined by the number and characteristics of other flows with which it shares the resources of that path.

Response time depends on a large number of factors including signalling overhead (DNS, connection establishment,...), user equipment limitations, access bandwidth, server load and performance, as well as network congestion. A reasonable objective for a backbone network provider would be to provide sufficient capacity to make the response time largely independent of network delays. This means that the throughput of elastic flows is limited essentially by the above mentioned factors which are outside the provider's control. We contend that this is possible and can indeed be achieved relatively simply by maintaining the network in a stable operating regime where expected demand is somewhat less than available capacity.

There is indeed a huge difference in performance according to whether the network is stable or in an overloaded state. In the latter case, the arrival rate of new flows on a saturated link tends to be greater than the departure rate so that the number of flows continues to grow while their throughput tends to zero. In practice, the latter regime necessarily stabilizes due to aborted transfers. Potential for service differentiation and, conversely, danger from unwanted discrimination due for example to different round trip times (RTTs), depends critically on whether the network is stable or overloaded. In particular, if the network provider fulfils the objective of ensuring transparency, there is little scope for "gold", "silver" and "bronze" service classes, as envisaged in the Diffserv model.

This paper presents models for evaluating and qualifying the above performance characteristics. In the next section, we introduce our flow level model of elastic traffic as a random process. The model is used in Section 3 to evaluate the performance of a bottleneck link in the stable regime. We notably show that a high capacity backbone link is transparent when individual flows are subject to realistic external rate limitations, due to the access line speed, maximum TCP window or server performance, for instance. In Section 4 we study the overload regime and explain how transfers aborted due to impatience ensure stability but maintain an operating point where QoS is consistently poor, except potentially for users with a short RTT or an aggressively tuned transport protocol. Section 5 compares the discrimination realized by different scheduling mechanisms envisaged for service differentiation in both stable and overload regimes.

2. Modelling elastic traffic

We model elastic traffic in terms of *flows* where, for the purpose of this paper, a flow is defined as the sequence of packets pertaining to one instance of some application. The flow might correspond to a TCP connection established for the transfer of one element of a Web document or an entire page if this can be identified as a single entity. For the sake of simplicity, we suppose all flows are elastic and ignore the impact of streaming traffic on bandwidth availability. This simplification does not detract from our conclusions which are primarily of a qualitative nature.

2.1. Flows vs. packets

We adopt a fluid traffic model where the rate of a flow is always well defined and varies depending on the number of flows currently sharing link bandwidth on its path. We therefore ignore the imprecision in meeting sharing objectives introduced by packet level protocols (e.g., slow start and congestion avoidance algorithms of TCP) in order to derive simpler results on QoS at the higher level. In the following, a flow is simply characterized by an arrival time and a volume of data to be transmitted on a network path. It may additionally be qualified by parameters such as the RTT or other external factors affecting the bandwidth it obtains on a shared link.

A key reason for choosing to model flows rather than packets is that quality of service is experienced by users at this level. Users of elastic applications are generally *not* sensitive to the end-to-end delay of each packet, but to the time necessary to transfer an entire document, equal to the response time of the associated flow. This perceived QoS depends critically on the number of flows sharing the same resources and on the way this number varies as new flows begin and others terminate. The objective of our models is to study these flow level dynamics.

Flows are generally not isolated but are generated within sessions. A session may be generically defined as an alternating series of flows and "think-times". The statistical properties of a session, including flow size distributions and correlations between successive flows and think-times, can be complex and clearly depend on the underlying application. However, these statistical properties are independent from one session to another [2]. This independence assumption naturally leads to a Poisson session arrival process when the number of users is large and no one user generates a significant proportion of the overall traffic.

The flow arrival process, on the other hand, tends to be bursty and has indeed been shown to be self-similar in certain cases [8,12]. A plausible explanation for this behaviour is that the number of flows per session has a heavy-tailed distribution [4]. It may nevertheless be appropriate in certain circumstances to suppose flows arrive according to a Poisson process. This would be the case, for example, when flows correspond to a large number of sessions and the spacing of flows within a session is large compared to the average inter-flow interval. Furthermore, results derived for Poisson flow arrivals are exactly or approximately applicable under the relaxed assumption of Poisson session arrivals [2]. We denote the flow arrival rate by λ.

Measurements of the size of documents such as Web pages and FTP files show that their distribution has a heavy tail [6,12]. A consequence of this is that the large majority of flows are very small while most of the traffic in bytes is contained in large flows. The precise distribution clearly depends on the type of document considered. In the models we develop we use a number of different document size distributions either to illustrate the impact of this traffic characteristic on performance or to facilitate tractability when solutions are only available for Markovian systems. Document size is denoted by the random variable σ. A reasonable fit to the form of the heavy tail observed in practice is provided by the Pareto distribution:

$$\Pr[\sigma > x] = \left(\frac{b}{x}\right)^a \quad \text{for } x > b,$$

where $a > 1$ is a fixed parameter and b is the minimum document size, equal to $(a-1)E[\sigma]/a$. Note that the associated variance is infinite as soon as $a \leqslant 2$. We take $a = 1.4$ for all numerical applications in the paper.

2.2. An isolated bottleneck link

We consider an isolated bottleneck link of capacity C. Traffic demand, expressed as a bit rate, is the product of the flow arrival rate λ and the average flow size $E[\sigma]$. The *load* offered to the link is then defined by the ratio:

$$\rho = \frac{\lambda E[\sigma]}{C}. \tag{1}$$

It is worth noting that while the load on network links is usually less than 1, no control mechanism is currently present in the Internet to prevent traffic demand from exceeding link capacity. This can occur notably in case of failure when traffic must be diverted from its usual route. One of the key objectives of this paper is to evaluate network performance in overload, i.e., when $\rho > 1$.

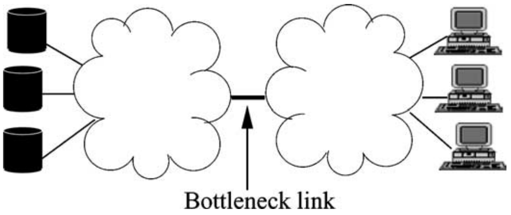

Fig. 1. An isolated bottleneck link.

Users perceive QoS through the time necessary to transfer a given document or, equivalently, by the average throughput realized during a transfer. A useful measure of throughput is the ratio:

$$\gamma = \frac{E[\sigma]}{E[T]}, \tag{2}$$

where T is the response time of an arbitrary flow. We will refer to γ as the *flow throughput*. This parameter is generally easier to calculate than the expected throughput $E[\sigma/T]$ but is equally useful as a measure of performance. Note that $0 \leqslant \gamma \leqslant C$.

We study the bottleneck link as a queueing system. It is important to note, however, that the queue in question is not materialized as such. It is distributed over the different sources which are currently transferring documents across the link to a certain population of users (see Fig. 1). Packet level queues are not included in the model. The effect of packet loss and congestion avoidance algorithms is taken into account through the assumed bandwidth sharing objectives (typically realized by TCP).

In the following, we evaluate the impact of the offered load ρ on the QoS parameter γ. In the next section, we consider the stable case $\rho < 1$. The unstable case $\rho > 1$ is considered in Section 4.

3. Stability and transparency

In this section, we show that when the traffic offered to a network link is less than its capacity, the link is virtually transparent. The QoS perceived by users depends much more on external throughput constraints such as a limited access rate or the performance of Web servers, for instance.

3.1. Fair sharing

Consider first the simplest case of perfect and fair bandwidth sharing, i.e., when n flows are in progress, each flow receives a fraction $1/n$ of the link capacity C. The associated model is then a processor sharing queue [10]. Provided the offered load ρ is less than 1, the number of flows in progress N has a geometric distribution in the steady state:

$$\forall n \geqslant 0, \quad \Pr[N = n] = \rho^n(1-\rho).$$

In particular, the average number of flows in progress is given by

$$E[N] = \frac{\rho}{1-\rho}.$$

By Little's law, we derive the flow throughput:

$$\gamma = \frac{\rho C}{E[N]} = C(1-\rho). \tag{3}$$

In this system, it is known that expected response time is proportional to the flow size so that γ has an alternative interpretation as an expected per-flow throughput: $\gamma = \sigma/E[T|\sigma]$. All these results are insensitive to *all* traffic characteristics except the overall load ρ. The only assumption is that *sessions* arrive as a Poisson process [2].

In practice, the throughput of the flows on a backbone link is subject to external limitations. To simplify discussion, we assimilate all these limitations to the rate of the user's access line. For instance, a user connected to the Internet by an ADSL access of 1.5 Mbits/s can never occupy more than 1% of a 155 Mbits/s OC3 backbone link. Assume for simplicity that all users have the same access rate $r < C$, where C/r is a given integer m. The associated model then corresponds to a multi-server processor sharing queue [5]. The stationary distribution of the number of flows in progress remains insensitive to all traffic characteristics and we have

$$\Pr[N = n] = (1-\rho)f(\rho)\begin{cases} \frac{m!}{n!}(\rho m)^{n-m}, & \text{for } n < m, \\ \rho^{n-m}, & \text{for } n \geqslant m, \end{cases}$$

where $f(\rho)$ is the probability the link bandwidth is fully utilized:

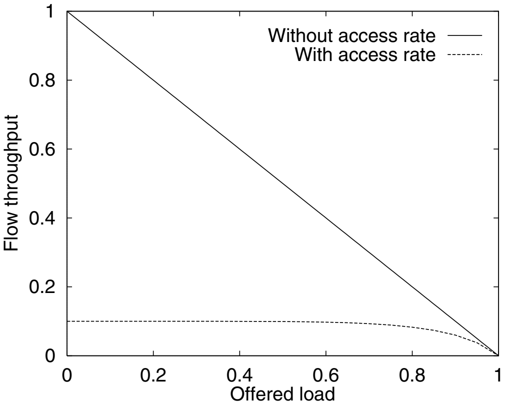

Fig. 2. Normalized flow throughput against offered load in case of fair sharing.

$$f(\rho)=\frac{\frac{(\rho m)^m}{m!}}{\frac{(\rho m)^m}{m!}+(1-\rho)\sum_{n=0}^{m-1}\frac{(\rho m)^n}{n!}}.$$

The flow throughput is now:

$$\gamma=\frac{\rho C}{E[N]}=\frac{C(1-\rho)}{C(1-\rho)/r+f(\rho)}. \tag{4}$$

Response time is still proportional to flow size σ so that γ again has a more significant interpretation as size-independent expected per-flow throughput. When $r/C \ll 1$, we obtain the simpler approximate expression:

$$\gamma \approx \min(r, C(1-\rho)).$$

We conclude that for small access rates r, flow throughput is approximately insensitive to offered load, provided $\rho < 1$. In other words, the link is virtually transparent to the users, whose perceived QoS depends much more on the access rate r. This is illustrated by Fig. 2, which gives the flow throughput as a fraction of the link capacity for an access rate $r/C = 0.1$. Note that this access rate ratio is rather high for a backbone link and is chosen mainly to make the figure clearer. A more realistic value of this ratio makes the flow throughput completely insensitive to the offered load as long as this is less than $1 - r/C$.

3.2. Unequal sharing

We now consider the case where bandwidth sharing is not perfectly fair but depends on a parameter associated with each flow. This unfairness might represent the impact of different RTTs on the throughput of TCP, for example, or different versions of the protocol which are more or less responsive to congestion. For simplicity, we consider just two types of flow with associated arrival rates λ_1 and λ_2, and assume that flows of type 1 receive φ times more bandwidth than flows of type 2 (except when there are no flows in progress of one or both types). In the absence of an access rate limitation, the associated model is then a so-called *discriminatory* processor sharing queue.

The mean response time of an arbitrary flow is no longer insensitive to the traffic characteristics. In the case of Poisson flow arrivals and an exponential document size distribution, it follows easily from [7] that the throughput for flows of type 1 and type 2 is given by

$$\gamma_1=\frac{1+\varphi(1-\rho_1)-\rho_2}{1+\varphi(1-\rho)}C(1-\rho)$$

and

$$\gamma_2=\frac{1+\varphi(1-\rho_1)-\rho_2}{1+\varphi-\rho}C(1-\rho),$$

respectively, where $\rho_1=\lambda_1 E[\sigma]/C$ and $\rho_2=\lambda_2 E[\sigma]/C$ are the associated traffic loads. In this system, performance depends on the flow size distribution and the above results for Poisson flow arrivals do not strictly apply under the relaxed assumption of Poisson session arrivals. However, it appears from a limited number of simulation experiments that sensitivity to detailed traffic characteristics is not great. In particular, the throughputs γ_1 and γ_2 are approximately valid for a range of distributions with higher variance and correspond to per-flow throughput for flows of any size up to the very largest. Throughput of the latter tends to $C(1-\rho)$, corresponding to the mean residual capacity, for any user type and any (work conserving) bandwidth sharing scheme.

Fig. 3 shows the results obtained in the case $\varphi = 10$ and $\rho_1 = \rho_2 = \rho/2$. Numerical results for hyperexponentially distributed flow size and simulations with the Pareto distribution show that these results are approximately insensitive to the flow size distribution. Note first that the relative difference in throughput depends on load and is

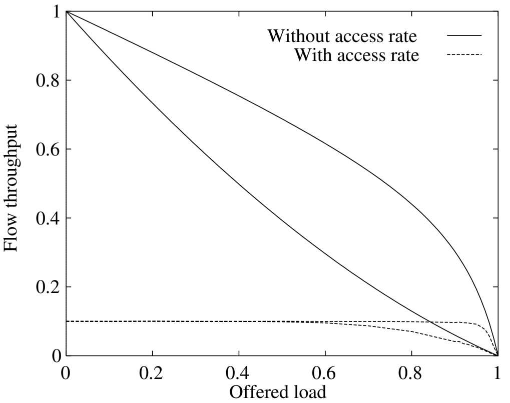

Fig. 3. Normalized flow throughput against offered load when flows of type 1 (top) receive $\varphi = 10$ times more bandwidth than flows of type 2 (bottom).

considerably less than 10 for light and moderate loads. The impact of an access rate limitation r has been evaluated by simulation. The results are plotted in Fig. 3 for the case where $r/C = 0.1$. We observe that, as in the case of fair sharing, the throughput for each type of flow is approximately equal to the minimum of that obtained in the absence of access rate limitation and the access rate r.

We conclude that, for small access rates, flow throughput is roughly insensitive to the discrimination parameter φ and to offered load, provided $\rho < 1$. In particular, the problem of unwanted discrimination (due to heterogeneous RTTs, different versions of TCP, or the presence of unresponsive flows) hardly exists in the absence of overload, except in a narrow region close to critical loading ($\rho \approx 1$). The same results suggest that there is little scope for deliberate discriminatory sharing. We return to this question in Section 5.

In practice, link bandwidth is not shared as precisely as assumed in the above fluid models. In particular, the slow-start algorithms of TCP can severely restrict the throughput of small flows. We maintain, however, that the fluid models provide very valuable insight into the impact on performance of traffic characteristics. The insensitivity of average performance to the detailed statistical properties of sessions is of great importance for network engineering (since performance depends only on overall load). This property is likely to be maintained approximately even when accounting for disparities due to packet level behaviour. In particular, network links are still virtually transparent as long as their offered load is somewhat less than one.

4. Instability and impatience

In the previous section, we have shown that an isolated link is virtually transparent to users in the absence of overload, in the sense that perceived QoS depends almost exclusively on external throughput constraints. In this section, we consider the overload situation, and identify user (or application) *impatience* as a key factor in evaluating network performance. For simplicity, we assume that the flow arrival process is independent of link congestion. This assumption is not strictly true due to the impact of session structure. However, to account for the latter seems very complicated and would obscur the insights derived from the simpler model.

4.1. Excess of traffic demand

Consider again the simple model of Section 3.1, i.e., the case of an isolated link whose capacity is shared perfectly fairly between the flows in progress. When traffic demand exceeds link capacity, i.e., when $\rho > 1$, the model is *unstable*, in the sense that the number of flows in progress increases indefinitely. Though in practice this number eventually stabilizes due to some users abandoning their transfers, it is interesting to first consider the *transient* behaviour of the system prior to users becoming impatient.

Denote by μ the asymptotic *flow completion rate* defined as the limit when t tends to infinity of the number of completed flows between times 0 and t divided by t. Jean-Marie and Robert [9] have shown that μ is a solution of the following equation:

$$\mu = \lambda E[\mathrm{e}^{-(\lambda-\mu)\sigma/C}]. \tag{5}$$

The solution $\mu = \lambda$ corresponds to the stable case $\rho < 1$ where the number of flows in progress remains finite. In overload, the flow completion rate is less than λ and depends significantly on the distribution of flow size σ.

Deterministic distribution. First assume that all documents have the same size $\sigma = E[\sigma]$. In this case, we have $\mu = \lambda e^{-(\lambda-\mu)\sigma/C}$ so that the flow completion rate tends to zero when λ tends to ∞, with

$$\mu \sim \lambda e^{-\lambda\sigma/C} \quad \text{when } \lambda \to \infty.$$

Exponential distribution. For an exponential document size distribution, we simply obtain $\mu = C/E[\sigma]$, i.e., the flow completion rate is equal to the flow service rate, whatever the flow arrival rate λ.

Hyperexponential distribution. Finally, we consider the hyperexponential distribution with parameter $a \geqslant 1$, defined by

$$\forall x \geqslant 0, \quad \Pr[\sigma > x] = \frac{a e^{-ax/E[\sigma]} + e^{-x/(aE[\sigma])}}{a+1}. \qquad (6)$$

For a large parameter a, this gives a high fraction of small documents of mean size $E[\sigma]/a$, representing a proportion $1/(a+1)$ of the load, and a low fraction of large documents of mean size $aE[\sigma]$, representing a proportion $a/(a+1)$ of the load.

From (5), we obtain after simple calculations:

$$\mu = \lambda + \frac{C}{2E[\sigma]}\left(a + \frac{1}{a} - \rho - \sqrt{\left(a + \frac{1}{a} - \rho\right)^2 + 4(\rho - 1)}\right).$$

Fig. 4 gives the flow completion rate μ against the flow arrival rate λ when the document size distribution is deterministic, exponential, and hyperexponential with parameter $a = 100$, respectively. We observe that the flow completion rate increases with the variability of the document size. For the hyperexponential distribution, the flow completion rate is close to the flow arrival rate even when $\rho = 2$. This can be explained by the fact that most of the documents are very short and complete their transfer rapidly; only large documents that contribute most of the offered load are significantly affected by the overload.

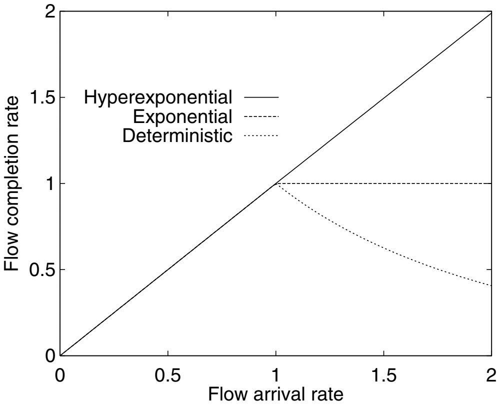

Fig. 4. Flow completion rate against flow arrival rate for different document size distributions.

4.2. Modelling impatience

In the previous model, the number of flows in progress increases indefinitely and the fraction of link capacity available to each flow decreases to zero as overload lasts. In practice, of course, some users become impatient as transfer times grow longer and interrupt the transfer, by clicking on the browser stop button, for instance. Alternatively, TCP or higher layer protocols may interpret the extremely low throughput as a broken connection and interrupt without user intervention. Connection aborts stabilize the number of flows in progress at a high but finite value thus maintaining the system at a stable operating point.

We are unaware of any objective study of user (or application) patience based on measurements. We would ideally like to know the probability distribution of the time a user is prepared to wait for flow completion as a function of the size of the document to be transferred. However, even this would be a simplification of the real phenomenon since user behaviour depends on many factors including past experience of response times, the type of activity (Web, e-commerce or e-mail, for example) and an estimation of current congestion status.

Impatience is also manifested more often by non-completion of a session than by the interruption of a flow. Such impatience is impossible to detect just by observing traffic traces since nothing distinguishes a complete short session from an interrupted longer session. In the following we introduce some simple models of impatience at flow level which provide a qualitative appraisal of its impact on QoS.

Patience duration. We assume that the duration of a flow cannot exceed a given "patience duration" τ associated with the flow. When flow arrivals are Poisson and σ, τ are independent exponential random variables, the stationary distribution of the number of flows in progress can be explicitely evaluated [11]. We have

$$\Pr[N = n] = \prod_{k=1}^{n} \frac{\theta}{k/\lambda E[\tau] + 1/\rho},$$

where θ denotes the normalization constant. An interesting remark with this formula is that, for patient users ($\lambda E[\tau] \gg 1$) and heavy traffic ($\rho \gg 1$), the number of flows in progress remains approximately constant. For link capacity $C = 10$ Mbits/s, mean document size $E[\sigma] = 100$ Kbits, mean patience duration $E[\tau] = 10$ s and offered load $\rho = 2$, for instance, there are between 900 and 1100 flows in progress 99% of the time. We conclude that the throughput of each flow is roughly constant and equal to 10 Kbits/s in this case.

The fact that flow throughput is approximately constant in steady state is actually independent of the document size and patience duration distributions. Provided the patience duration and the offered load are sufficiently large, the steady state number of flows in progress varies slightly about a high value n. The time necessary to transfer a document of size σ is then approximately equal to σ/γ, where $\gamma = C/n$ represents the link capacity allocated to each flow.

Maximum document size. Assume now that patience duration depends only on flow size, i.e., $\tau = \tau(\sigma)$. It seems natural to assume that τ is an increasing but concave function of σ since users have a response time expectation which increases with the flow size but need proportionally more throughput. A simple example is the linear function:

$$\forall\sigma, \quad \tau(\sigma) = \delta + \frac{\sigma}{\alpha}, \tag{7}$$

where δ represents the threshold below which users are not impatient, referred to as the *tolerance*, and α represents the minimum throughput required to transfer extremely large documents, referred to as the *sustainable throughput*. A flow is completed if and only if

$$\frac{\sigma}{\gamma} < \tau(\sigma).$$

The assumption of a concave patience function implies that there exists a *maximum document size* $\sigma^\star$, satisfying $\sigma = \gamma\tau(\sigma)$, beyond which all flows are interrupted.

This is illustrated in Fig. 5 which shows simulation results corresponding to a linear patience duration with tolerance $\delta = 10$ s and sustainable throughput $\alpha = 100$ Kbits/s. The figure plots the duration of flows (completed or interrupted) as a function of their size. Link capacity is $C = 10$ Mbits/s and offered load $\rho = 2$. Document sizes follow a Pareto distribution of mean $E[\sigma] = 100$ Kbits. The figure confirms the assumption that per-flow throughput is roughly constant being equal to the inverse of the slope of the steeper diagonal line, i.e., around 60 Kbits/s. The maximum document size $\sigma^\star$ occurs at the intersection of this line with the line representing the patience duration. Note that the mean number of flows in progress is here around 1700. Variation about the mean is relatively slight so that we obtain excellent agreement between the model (assuming a constant number of flows) and simulation results.

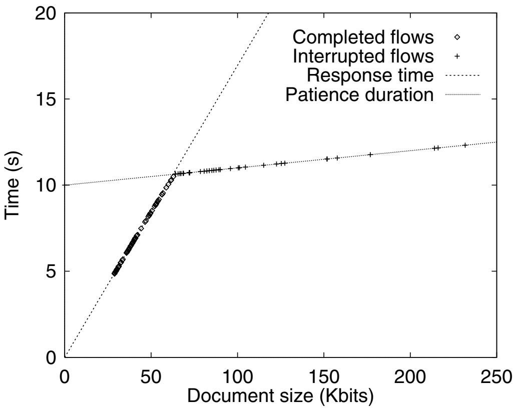

Fig. 5. Duration of the flows against their size for a Pareto document size distribution of mean $E[\sigma] = 100$ Kbits, a tolerance $\delta = 10$ s and a sustainable throughput $\alpha = 100$ Kbits/s.

In the following, we always assume that the flow throughput γ is constant and evaluate its value when the patience duration τ is fixed, random and dependent on document size σ, respectively.

4.3. Fixed patience duration

Denote by $\rho^\star$ the *effective* traffic load, namely the ratio between effective traffic and link capacity:

$$\rho^\star = \frac{\lambda E[\min(\sigma, \sigma^\star)]}{C}.$$

Since the number of flows in progress remains approximately constant and equal to C/γ, the effective traffic load $\rho^\star$ is necessarily equal to 1. We conclude that

$$\lambda E[\min(\sigma, \sigma^\star)] = C. \tag{8}$$

This equation characterizes the maximum document size $\sigma^\star$, whose value is thus independent of the patience duration τ. Greater patience leads simply to lower throughput since γ decreases like $\sigma^\star/\tau$. We now evaluate $\sigma^\star$ for different document size distributions, and derive the *useful work* of the link, that is the fraction of link capacity used to transfer flows that will eventually be completed:

$$U = \frac{\lambda E[\sigma \mathbb{1}_{\{\sigma<\sigma^\star\}}]}{C}.$$

Deterministic distribution. When all documents have the same size $\sigma = E[\sigma]$, all transfers are interrupted after a volume $\sigma^\star = \sigma/\rho$ has already been transferred. The useful work U is equal to zero in this case.

Exponential distribution. For the exponential distribution, we have

$$E[\min(\sigma, \sigma^\star)] = E[\sigma](1 - \mathrm{e}^{-\sigma^\star/E[\sigma]}),$$

so that from (8),

$$\sigma^\star = E[\sigma] \ln\left(\frac{\rho}{\rho - 1}\right).$$

The useful work is then given by

$$U = 1 - (\rho - 1) \ln\left(\frac{\rho}{\rho - 1}\right).$$

Note that U decreases like $1/\rho$ when the offered load ρ tends to infinity. The same asymptotic result was derived in the case of an independent exponentially distributed patience duration in [11].

Pareto distribution. For the Pareto distribution, we have:

$$E[\min(\sigma, \sigma^\star)] = E[\sigma]\left(1 - \frac{1}{a}\left(\frac{b}{\sigma^\star}\right)^{a-1}\right).$$

It follows then from (8) that

$$\sigma^\star = E[\sigma]\frac{a-1}{a}\left(\frac{\rho}{a(\rho - 1)}\right)^{1/(a-1)},$$

provided $\sigma^\star \geqslant b$. The useful work is given by

$$U = \max\left\{0, 1 - \rho\frac{a-1}{a}\left(\frac{b}{\sigma^\star}\right)^{a-1}\right\},$$

that is,

$$U = \max\{0, 1 - (a-1)(\rho - 1)\}.$$

Note that the useful work tends to zero when the offered load approaches $a/(a-1)$. This corresponds to the case where the maximum completed document size $\sigma^\star$ is close to the minimum document size $(a-1)E[\sigma]/a$, so that all transfers are interrupted.

Fig. 6 plots flow throughput $\gamma = \sigma^\star/\tau$ and useful work U against offered load ρ for the three document size distributions. Common parameters are: link capacity $C = 10$ Mbits/s, mean document size $E[\sigma] = 100$ Kbits and patience duration $\tau = 10$ s. We observe that flow throughput and useful work increase with the variability of the document size. This is simply because the maximum completed document size is larger, due to the fact that large documents represent a greater weight in the overall distribution. For an excess of traffic demand of 50%, for instance, only 20% of capacity is wasted with a Pareto document size distribution (with parameter $a = 1.4$), while more than 50% of capacity is wasted with an exponential document size distribution.

4.4. Random patience duration

We now consider the case of random patience duration τ, corresponding to a range of user behaviours. In the following, we restrict ourselves to the Pareto document size distribution.

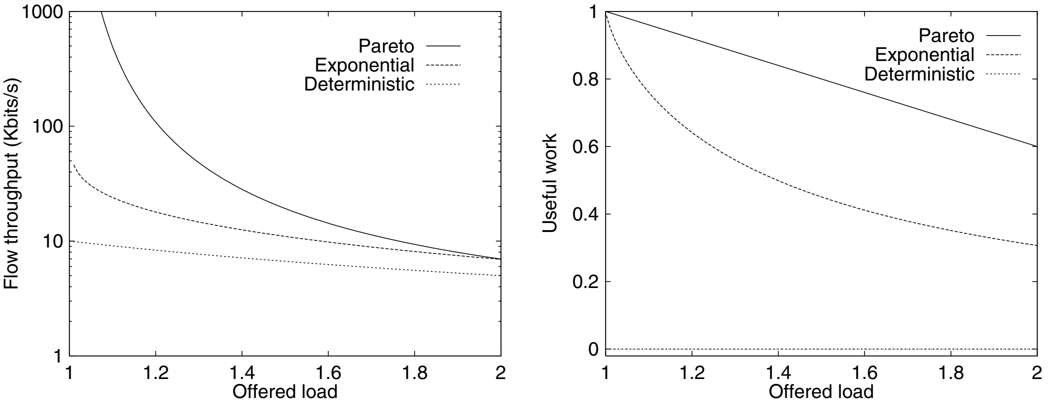

Fig. 6. Flow throughput and useful work against offered load for an average document size $E[\sigma] = 100$ Kbits, a patience duration $\tau = 10$ s and different document size distributions.

Flow throughput. The maximum document size $\sigma^\star = \gamma\tau$ is now a random variable independent of σ. In view of the results of Section 4.3, we have

$$E\left[\left(\frac{b}{\sigma^\star}\right)^{a-1}\right] = a\frac{\rho - 1}{\rho}. \tag{9}$$

By convexity of the function $x\mapsto(1/x)^{a-1}$, we conclude that the expected maximum document size $E[\sigma^\star]$ is necessarily larger than the maximum document size obtained with a fixed patience duration. In particular, the flow throughput $\gamma = \sigma^\star/\tau$ is *larger* with random patience duration τ than with fixed patience duration equal to $E[\tau]$.

Useful work. Concerning the useful work, we have

$$U = 1 - \rho\frac{a-1}{a}E\left[\left(\frac{b}{\sigma^\star}\right)^{a-1}\right].$$

Using (9), we obtain

$$U = 1 - (a-1)(\rho - 1).$$

We conclude that useful work is *insensitive* to the distribution of patience duration.

4.5. Variable patience duration

Finally, consider the case where the patience duration τ depends on the size σ of the transferred document. For the sake of simplicity, we restrict attention to a linear patience duration as in (7). The condition for a document of size σ to be transferred becomes:

$$\frac{\sigma}{\gamma} < \delta + \frac{\sigma}{\alpha}.$$

Note that flow throughput γ is necessarily smaller than the sustainable throughput α in steady state. The maximum document size is given by

$$\sigma^\star = \frac{\delta}{1/\gamma - 1/\alpha}. \tag{10}$$

The effective load is now:

$$\rho^\star = \frac{\lambda E[\min(\sigma, \gamma(\delta + \sigma/\alpha))]}{C}.$$

Since $\rho^\star = 1$, it follows from (10) that the maximum document size satisfies the equation:

$$\rho E\left[\min\left(\sigma, \sigma^\star\frac{\delta + \sigma/\alpha}{\delta + \sigma^\star/\alpha}\right)\right] = 1.$$

After simple calculations, for the Pareto distribution we deduce

$$\rho\left(1 - \frac{1}{a}\left(\frac{b}{\sigma^\star}\right)^{a-1}\frac{\delta}{\delta + \sigma^\star/\alpha}\right) = 1.$$

Fig. 7 gives the results obtained numerically for a link of capacity $C = 10$ Mbits/s with offered load $\rho = 2$, mean document size $E[\sigma] = 100$ Kbits, tolerance $\delta = 10$ s, and different values of the sustainable throughput α. The top line in both plots corresponds to results for fixed patience duration,

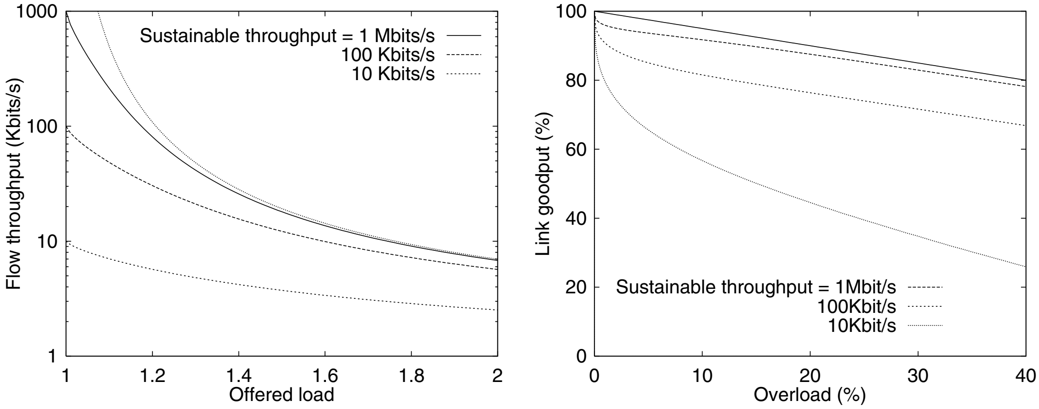

Fig. 7. Flow throughput and useful work against offered load, for average document size $E[\sigma] = 100$ Kbits, tolerance $\delta = 10$ s, and different values of sustainable throughput α. The top line in each figure corresponds to the limiting case $\alpha \to \infty$.

i.e., for $\alpha = \infty$. As expected, both throughput and useful work decrease when users become more patient.

4.6. Reattempts

Aborted flows are not generally abandoned immediately as users will frequently make a repeat attempt. The impact of this behavior is to exacerbate the waste of bandwidth due to impatience as it is likely that the reattempts will also be interrupted. Assume for instance that if a user aborts a reattempt is made with fixed probability p. We consider a size dependent patience duration τ as introduced above. Reasoning as above, the effective load is now:

$$\rho^\star = \frac{\lambda E[\sigma \mathbb{1}_{\{\sigma \leqslant \gamma\tau(\sigma)\}}]}{C} + \frac{\lambda E[\gamma\tau(\sigma)\mathbb{1}_{\{\sigma > \gamma\tau(\sigma)\}}]}{(1-p)C}.$$

Fig. 8 plots the corresponding useful U against p in the case of a 20% overload assuming fixed patience duration and a Pareto size distribution. The figure shows that the wasted bandwidth can be considerable. While the model is overly simple, it does illustrate the negative impact user behavior can have in case of overload.

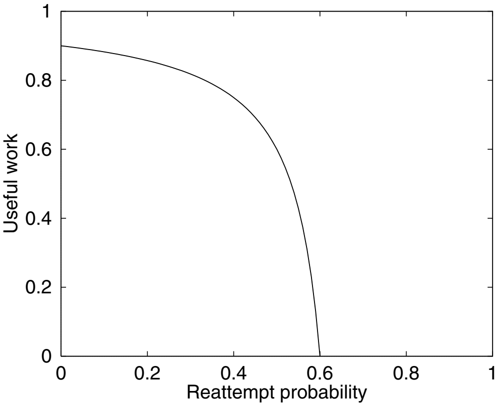

Fig. 8. Impact of reattempts on useful work.

4.7. Unequal sharing

We now consider the case of unequal bandwidth sharing. It turns out that, under overload and with user impatience, QoS discrimination is much more significant than in the stable case considered in Section 3.2. Again, we consider two types of flow with associated arrival rates λ_1 and λ_2 and assume that flows of type 1 receive φ times more bandwidth than flows of type 2, that is, $\gamma_1 = \varphi\gamma_2$. As illustrated by the simulation results of Fig. 9 with $\varphi = 2$, a tolerance $\delta = 10$ s and a sustainable throughput $\alpha = 100$ Kbits/s, in this case there are two maximum document sizes $\sigma_1^\star$ and $\sigma_2^\star$ with $\sigma_1^\star \geqslant \sigma_2^\star$.

Assume users of both classes have the same patience duration $\tau = \delta + \sigma/\alpha$ (we do not consider reattempts here). The effective offered load is then given by

$$\rho^{\star} = \frac{\lambda_1 E[\min(\sigma, \gamma_1(\delta + \sigma/\alpha))]}{C} + \frac{\lambda_2 E[\min(\sigma, \gamma_2(\delta + \sigma/\alpha))]}{C}.$$

Setting $\rho^{\star} = 1$, we deduce as above:

$$\rho - \frac{\rho_1}{a}\left(\frac{b}{\sigma_1^{\star}}\right)^{a-1} \frac{\delta}{\delta + \sigma_1^{\star}/\alpha} - \frac{\rho_2}{a}\left(\frac{b}{\sigma_2^{\star}}\right)^{a-1} \frac{\delta}{\delta + \sigma_2^{\star}/\alpha} = 1.$$

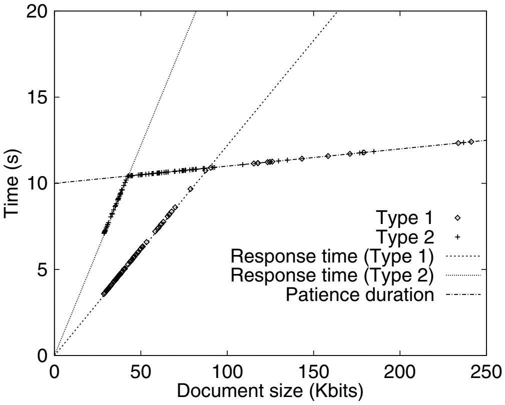

Fig. 9. Duration of completed flows against their size for a Pareto document size distribution of mean $E[\sigma] = 100$ Kbits in case of unequal bandwidth sharing.

Whereas $\sigma_2^{\star}$ is necessarily finite, $\sigma_1^{\star}$ may be finite or infinite depending on the traffic load ρ. If $\sigma_1^{\star}$ is finite, we have in addition the equation:

$$\frac{\sigma_1^{\star}}{\delta + \sigma_1^{\star}/\alpha} = \varphi \frac{\sigma_2^{\star}}{\delta + \sigma_2^{\star}/\alpha}.$$

Otherwise, the throughput of flows of type 1 is necessarily larger than the sustainable throughput. The condition for which all flows of type 1 are completed is thus given by

$$\varphi \frac{\sigma_2^{\star}}{\delta + \sigma_2^{\star}/\alpha} > \alpha,$$

yielding

$$\sigma_2^{\star} > \frac{\delta\alpha}{\varphi - 1}.$$

Fig. 10 shows the results obtained in the case $\varphi = 10$ and $\rho_1 = \rho_2 = \rho/2$, for a link of capacity $C = 10$ Mbits/s, mean document size $E[\sigma] = 100$ Kbits, tolerance $\delta = 10$ s and sustainable throughput $\alpha = 100$ Kbits/s. We observe in this case that all flows of type 1 are completed as long as excess traffic demand is less than 20%.

The results in this section illustrate that under overload, unfairness in bandwidth sharing performance is significant. Connections with a short RTT obtain significantly more throughput than connections over longer paths and tend to profit from the impatience of the latter.

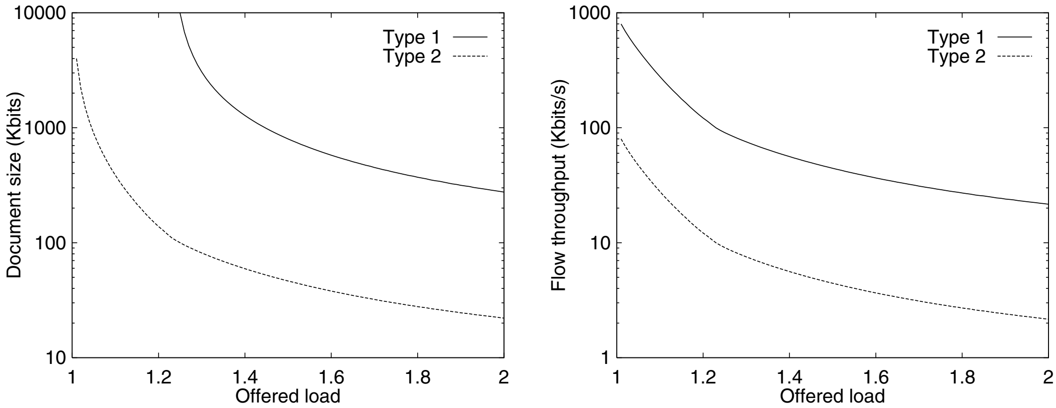

Fig. 10. Maximum size of completed flows and flow throughput for a Pareto document size distribution of mean $E[\sigma] = 100$ Kbits in case of unequal bandwidth sharing.

5. QoS differentiation

In view of the above results, the QoS of individual elastic flows looks like a threshold-type function of the offered load ρ. Roughly speaking, either $\rho < 1$, and the stability of the number of flows in progress results in excellent QoS for all flows, with realized throughput close to their access rate or $\rho > 1$, and the number of flows in progress accumulates and reaches a steady state such that QoS is very bad for all flows. This suggests that QoS differentiation can be achieved in case of overload only, by preserving higher priority classes from the adverse effects of congestion. This observation motivates the following stability analysis of some generic bandwidth sharing mechanisms that are likely to be used in a Diffserv architecture [3].

In the rest of the paper, we consider N service classes. The arrival rate of flows of class i is denoted by λ_i. Assuming that the distribution of document size σ is not class dependent, we obtain the traffic load of class i:

$$\rho_i = \frac{\lambda_i E[\sigma]}{C} \quad \text{for } i = 1, \dots, N.$$

5.1. Per-flow weighted fair queueing

Let $\varphi_1 \geqslant \varphi_2 \geqslant \cdots \geqslant \varphi_N$ be pre-assigned weights such that, when at least one flow of classes i and j is in progress, a class i flow receives φ_i/φ_j times more bandwidth than a class j flow. We have seen in Section 3.2 that, provided the total offered load ρ is less than 1, such unequal sharing of bandwidth has limited impact on the QoS of individual flows, expect in a narrow region close to critical loading. This is due to the fact that, in the absence of overload, link bandwidth is rarely fully utilized so that per-flow discrimination is typically ineffective. Flow throughput is limited by external rate limitations which are independent of user class.

On the other hand, discrimination does become significant in the presence of overload, as explained in Section 4.7. In particular, depending on the offered load, some transfers of class j may be interrupted whereas all transfers of class $i < j$ are completed. It is worth noting, however, that the QoS of *all* flows in progress depends on the level of user impatience and thus is likely to be adversely affected by the excess of traffic demand. In this sense, per-flow weighted fair queueing is unable to effectively protect higher priority service classes from overload.

5.2. Priority queueing

We now consider a scheduling scheme which gives (non-preemptive) priority to the packets of the flows of class i with respect to packets of flows of any class j, for $i < j$. We assume that this packet-level scheduling results in a preemptive priority at the flow level, as soon as higher priority flows are able to use all the available bandwidth (i.e., when they are not limited by their access rate). [1]

First note that flows of class 1 are then not affected by flows of any class $i > 1$. From (4), the average throughput of the flows of class 1 is given by

$$\gamma_1 = \frac{C(1-\rho_1)}{C(1-\rho_1)/r + f(\rho_1)}. \tag{11}$$

The throughput of flows of any other class depends on all traffic characteristics. In case of Poisson flow arrivals and an exponential size distribution, the total number of ongoing flows of classes $1, \dots, i$ is the same as that in a multi-server processor-sharing queue of load $\overline{\rho}_i = \rho_1 + \cdots + \rho_i$, so that

$$\forall i, \quad \sum_{j=1}^{i} E[N_j] = \overline{\rho}_i C/r + \frac{\overline{\rho}_i f(\overline{\rho}_i)}{1-\overline{\rho}_i},$$

provided $\overline{\rho}_i < 1$. Using this and (11), we can deduce the throughput of flows for any class $i > 1$. Again, the QoS of individual flows is essentially the same for all classes provided $\rho < 1$, as flows are typically limited by their access rate. This is illustrated in Fig. 11 for the case of two classes with the

[1] Packet level priority preserves higher priority classes from loss allowing their TCP connections to ramp up to a high rate. TCP connections of low priority classes, on the other hand, tend to suffer repeated loss and receive minimal throughput.

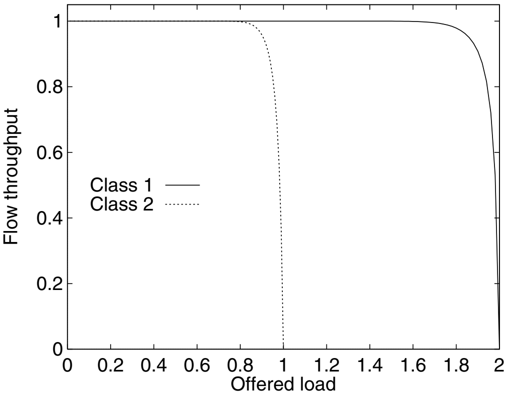

Fig. 11. Flow throughput as a fraction of the access rate against offered load with priority queueing, when $\rho_1 = \rho_2 = \rho/2$.

same traffic load $\rho_1 = \rho_2 = \rho/2$ when the access rate is equal to 1% of link capacity. As expected, QoS differentiation is really effective in case of overload only. In this particular case, priority queueing protects class 1 from overload as long as the excess of traffic demand is less than 100%. Results for this figure are derived without taking account of user impatience.

5.3. Class-based weighted fair queueing

Now assume that bandwidth is shared among classes that have at least one flow in progress in proportion to fixed weights $\phi_1, \ldots, \phi_N$. Assume without loss of generality that $\phi_1 + \cdots + \phi_N = 1$, and that classes are numbered in such a way that

$$\frac{\rho_1}{\phi_1} \leqslant \frac{\rho_2}{\phi_2} \leqslant \cdots \leqslant \frac{\rho_N}{\phi_N}.$$

The QoS offered to the flows is always better than that obtained if the link were divided into N virtual links of capacity $\phi_1 C, \ldots, \phi_N C$, the virtual link of capacity $\phi_i C$ being dedicated to the flows of class i. As for priority queueing, it turns out that for small access rates, QoS differentiation is not significant in the non-saturated case, $\rho < 1$. This is illustrated in Fig. 12 for the case of two classes with the same traffic load $\rho_1 = \rho_2 = \rho/2$ and $\phi_1 = 3\phi_2$, when the access rate is equal to 1% of the link capacity.

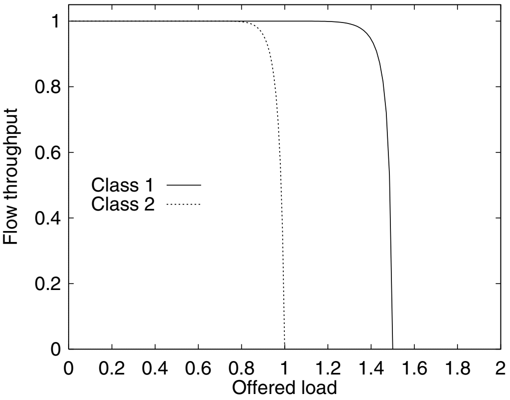

Fig. 12. Flow throughput as a fraction of the access rate against offered load with per-class weighted-fair queueing, when $\rho_1 = \rho_2 = \rho/2$ and $\phi_1 = 3\phi_2$.

When $\rho > 1$, let j be such that

$$\sum_{i=1}^{j-1} \rho_i < \sum_{i=1}^{j-1} \phi_i \quad \text{and} \quad \sum_{i=1}^{j} \rho_i > \sum_{i=1}^{j} \phi_i.$$

The fraction of link capacity allocated to flows of classes $1, \ldots, j-1$ is sufficient to satisfy their traffic demand. Thus their perceived QoS is what they would have obtained in a non-saturated link and is excellent. On the other hand, the traffic demand of flows of classes $j, \ldots, N$ exceeds their share of bandwidth. We conclude that their QoS is necessarily bad and largely determined by user impatience. In the particular case illustrated in Fig. 12, we observe that class-based weighted fair queuing protects class 1 from overload as long as the excess of traffic demand is less than 50%. The results are derived without accounting for user impatience.

5.4. Admission control

To alleviate the negative impact of demand overload, it may be considered preferable to deny access to certain users rather than to allow the number flows in progress to increase indefinitely with consequent quality of service degradation for all. This is the notion of admission control explored in [1], for example.

A significant advantage of performing admission control at flow level would be to allow service differentiation with respect to accessibility. High priority flows (distinguished by a class attribute as in Diffserv) would be allowed access to a congested link while lower priority flows are rejected. Since the rejection only occurs when the available bandwidth is low (i.e., when necessary to preserve performance), only a fraction of flows are rejected and the link continues to provide excellent throughput to all accepted flows.

The realization of flow level admission control appears to be within the capabilities of present technology. It remains, however, for router vendors to recognize the superiority of this means of overload control compared to the service differentiation techniques analysed above.

6. Conclusions

Traffic in the Internet is a random process resulting from the uncoordinated actions of a very large number of users. In this paper, we have applied traditional modelling approaches to study the traffic process resulting from so-called elastic applications. We have specifically assumed traffic in the busy period can be represented as a stationary process of flow arrivals where each flow corresponds to the transfer of a document whose size is drawn independently from a certain probability distribution.

The models are applied at flow level and use a fluid approximation to investigate the impact of different bandwidth sharing objectives on the performance of an isolated bottleneck link. Packet level protocols are not modelled directly but are simply assumed to realize the considered bandwidth sharing objectives.

The models illustrate that per-flow QoS depends critically on whether traffic demand (flow arrival rate × average flow size) is less than or greater than link capacity. If demand is less than capacity, the queuing system represented by the link and all the transfers it supports at any instant is stable. Except when the link is very close to saturation, it is effectively transparent to flows whose throughput is limited rather by external constraints (access rate, maximum TCP window, server performance, etc.). The impact of unfair discrimination due, for example, to different RTTs is then negligible. There is also very little scope for planned service differentiation since QoS for all categories is excellent.

On the other hand, discrimination is effective in overload. Substantial relative differences in throughput result from unequal bandwidth sharing whether this is planned or not. However, performance of all flows is rather poor except when priority or class-based scheduling mechanisms are implemented. These can preserve the QoS of certain classes by containing the impact of overload.

In overload, stability is in fact maintained by the impatience of users (and other causes of aborted transfers). Our models illustrate the significant impact of the flow size distribution on performance when user impatience is taken into account. Wasted bandwidth due to abandoned transfers turns out to be least in the practically interesting case of heavy tailed distributions. However, in all cases this source of inefficiency is significant and increases as users become more patient. The impact of greater patience is typically to reduce the average throughput of all flows without changing the proportion of flows which eventually complete. We conclude that the application of admission control, whereby certain flows are denied access from the start, would be a more effective overload control than simply relying on impatience.

A reasonable provisioning objective for the core of the Internet would be to render its links transparent with respect to the QoS experienced by users. We have shown that this can be achieved efficiently by ensuring the stability of the distributed queueing system represented by the links and their offered traffic. Adaptive flow routing and admission control appear as more efficient means for realizing this objective than reliance on impatience and the use of discriminatory scheduling mechanisms.

References

[1] N. Benameur, S. Ben Fredj, S. Oueslati, J. Roberts, Quality of service and flow level admission control in the Internet, Computer Networks 40 (2002) 57–71.

[2] S. BenFredj, T. Bonald, A. Proutière, G. Régnié, J.W. Roberts, Statistical bandwidth sharing: a study of congestion at flow level, in: Proc of ACM SIGCOMM 2001, ACM Computer Communications Review 31 (4) (2001) 111–122.

[3] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss, An Architecture for Differentiated Services, RFC 2475, December 1998.

[4] T. Bonald, A. Proutière, G. Régnié, J.W. Roberts, Insensitivity results in statistical bandwidth sharing, in: J. Moreira de Souza, N. da Fonseca, E.A. de Souza e Silva (Eds.), Proceedings of ITC17, Teletraffic Engineering in the Internet Age, Elsevier, Amsterdam, 2001.

[5] J.W. Cohen, The multiple phase service network with generalized processor sharing, Acta Informatica 12 (1979) 245–284.

[6] M. Crovella, A. Bestravos, Self-similarity in world wide web traffic: evidence and possible cause, in: Proceedings of ACM SIGMETRICS'96, 1996.

[7] G. Fayolle, I. Mitrani, R. Iasnogorodski, Sharing a processor among many classes, Journal of the ACM 27 (1980) 519–532.

[8] A. Feldmann, Characteristics of TCP connection arrivals, in: K. Park, W. Willinger (Eds.), Self-similar Network Traffic and Performance Evaluation, Wiley, New York, 2000.

[9] A. Jean-Marie, P. Robert, On the transient behavior of the processor sharing queue, Queueing Systems Theory and Applications 17 (1994) 129–136.

[10] L. Kleinrock, Queueing Systems, vol. 2, Wiley, New York, 1975.

[11] L. Massoulie, J.W. Roberts, Arguments in favour of admission control for TCP flows, in: P. Key, D. Smith (Eds.), Proceedings of ITC 16, Teletraffic Engineering in a Competitive World, Elsevier, Amsterdam, 1999, pp. 33–44.

[12] V. Paxson, S. Floyd, Wide-area traffic: the failure of poisson modeling, IEEE/ACM Transactions Networking 3 (3) (1995) 226–255.

Thomas Bonald graduated from Ecole Polytechnique (Paris) in 1994 and qualified as an engineer at Ecole Nationale Supérieure des Télécommunications (Paris) in 1996. He has a Ph.D. in applied mathematics from Ecole Polytechnique (Paris), his graduate research being performed at INRIA in the area of network flow control. He is currently in the traffic modelling group at France Telecom R&D, working on performance evaluation and design of overload controls for IP networks. His research interests include queueing theory and stochastic models of communication networks. He is a member of several conference program committees including Infocom 2003.

Jim Roberts has a B.Sc. in mathematics from the University of Surrey, UK and a Ph.D. from the University of Paris. He has been with the France Telecom Research Labs since 1978. His research has been mainly in the field of performance evaluation and design of traffic controls for multiservice networks. He was chairman of three successive European COST projects on the performance of multiservice networks, this activity culminating in the publication of the book "Broadband Network Teletraffic" (Springer 1996). He has published quite extensively and is or has been a member of a several journal editorial boards including Computer Networks and IEEE/ACM Transactions on Networking. He was a guest editor for the IEEE JSAC issue on Internet QoS in 2000. He is member of many conference programme committees in the networking field including Infocom, ITC and SIGCOMM. He is TPC co-chair for Infocom 2003.