

1 Engineering for Quality of Service

J. W. ROBERTS

France Télécom - CNET

Issy les Moulineaux, France

Keywords: quality of service, stream traffic, elastic traffic, pricing, fairness, statistical multiplexing, open loop control, closed loop control, admission control, leaky bucket.

1.1 INTRODUCTION

The traditional role of traffic engineering is to ensure that a telecommunications network has just enough capacity to meet expected demand with adequate quality of service. A critical requirement is to understand the three-way relationship between demand, capacity and performance, each of these being quantified in appropriate units. The degree to which this is possible in a future multiservice network remains uncertain, due notably to the inherent self-similarity of traffic and the modelling difficulty that this implies. The

purpose of the present chapter is to argue that sound traffic engineering remains *the* crucial element in providing quality of service and that the network must be designed to circumvent the self-similarity problem by applying traffic controls at an appropriate level.

Quality of service in a multiservice network depends essentially on two factors: the service model which identifies different service classes and specifies how network resources are shared, and the traffic engineering procedures used to determine the capacity of those resources. While the service model alone can provide differential levels of service ensuring that *some* users (generally those who pay most) have good quality, to provide that quality for a predefined *population* of users relies on previously providing sufficient capacity to handle their demand.

It is important in defining the service model to correctly identify the entity to which traffic controls apply. In a connectionless network where this entity is the datagram, there is little scope for offering more than “best effort” quality of service commitments to higher levels. At the other end of the scale, networks dealing mainly with self-similar traffic aggregates, such as all packets transitting from one LAN to another, can hardly make performance guarantees, unless that traffic is previously shaped into some kind of rigidly defined envelope. The service model discussed in this chapter is based on an intermediate traffic entity which we refer to as a “flow” defined for present purposes as the succession of packets pertaining to a single instance of some application, such as a videoconference or a document transfer.

By allocating resources at flow level, or more exactly, by rejecting newly arriving flows when available capacity is exhausted, quality of service provision is decomposed into two parts: service mechanisms and control protocols ensure that the quality of service of accepted flows is satisfactory; traffic engineering is applied to dimension network elements so that the probability of rejection remains tolerably small. The present chapter aims to demonstrate that this approach is feasible, sacrificing detail and depth somewhat in favour of a broad view of the range of issues which need to be addressed conjointly.

Other chapters in this book are particularly relevant to the present discussion. In Chapter ??, Adas and Mukherjee propose a framing scheme to ensure guaranteed quality for services like video transmission while Tuan and Park in Chapter ?? study congestion control algorithms for “elastic” data communications. Naturally, the schemes in both chapters take account of the self-similar nature of the considered traffic flows. They constitute alternatives to our own proposals. Chapter ?? by Feldmann gives a very precise description of Internet traffic characteristics at flow level which to some extent invalidates our too optimistic Poisson arrivals assumption. The latter assumption remains useful, however, notably in showing how heavy-tailed distributions do not lead to severe performance problems if closed loop control is used to dynamically share resources as in a processor sharing queue. The same Poisson approximation is exploited by Boxma and Cohen in Chapter ?? which contrasts the performance of FIFO (open loop control) and processor sharing (closed loop control) queues with heavy-tailed job sizes.

In the next section we discuss the nature of traffic in a multiservice network, identifying broad categories of flows with distinct quality of service requirements. Open loop and closed loop control options are discussed in Sections 1.3 and 1.4 where it is demonstrated notably that self-similar traffic does not necessarily lead to poor network performance if adapted flow level controls are implemented. A tentative service model drawing on the lessons of the preceding discussion is proposed in Section 1.5. Finally, in Section 1.6, we suggest how traditional approaches might be generalized to enable traffic engineering for a network based on this service model.

1.2 THE NATURE OF MULTISERVICE TRAFFIC

It is possible to identify an indefinite number of categories of telecommunications services, each having its own particular traffic characteristics and performance requirements. Often, however, these services are adaptable and there is no need for a network to offer multiple service classes each tailored to a specific application. In this section we seek a broad classification enabling the identification of distinct traffic handling requirements. We begin with a discussion on the nature of these requirements.

1.2.1 Quality of service requirements

It is useful to distinguish three kinds of quality of service measures which we refer to here as “transparency”, “accessibility” and “throughput”.

Transparency refers to the time and semantic integrity of transferred data. For real-time traffic delay should be negligible while a certain degree of data loss is tolerable. For data transfer, semantic integrity is generally required but (per packet) delay is not important.

Accessibility refers to the probability of admission refusal and the delay for set up in case of blocking. Blocking probability is the key parameter used in dimensioning the telephone network. In the Internet, there is currently no admission control and all new requests are accommodated by reducing the amount of bandwidth allocated to ongoing transfers. Accessibility becomes an issue, however, if it is considered necessary that transfers should be realized with a minimum acceptable throughput.

Realized throughput, for the transfer of documents such as files or Web pages, constitutes the main quality of service measure for data networks.. A throughput of 100 Kbit/s would ensure the transfer of most Web pages quasi-instantaneously (less than one second).

To meet transparency requirements the network must implement an appropriately designed service model. The accessibility requirements must then be satisfied by network sizing taking into account the random nature of user demand. Realized throughput is determined both by how much capacity is provided and how the service model shares this capacity between different flows. With respect to the above requirements, it proves useful to distinguish two broad classes of traffic which we term “stream” and “elastic”.

1.2.2 Stream traffic

Stream traffic entities are flows having an intrinsic duration and rate (which is generally variable) whose time integrity must be (more or less) preserved by the network. Such traffic is generated by applications like the telephone and interactive video services such as videoconferencing where significant delay would constitute an unacceptable degradation. A network service providing time integrity for video signals would also be useful for the transfer of pre-recorded video sequences and, although negligible network delay is not generally a requirement here, we consider this kind of application to be also a generator of stream traffic.

The way the rate of stream flows varies is important for the design of traffic controls. Speech signals are typically of on/off type with talkspurts interspersed by silences. Video signals generally exhibit more complex rate variations at multiple time scales. Importantly for traffic engineering, the bit rate of long video sequences exhibits long-range dependence [GW94], a plausible explanation for this phenomenon being that the duration of scenes in the sequence has a heavy-tailed probability distribution [Fra97].

The number of stream flows in progress on some link, say, is a random process varying as communications begin and end. The arrival intensity generally varies according to the time of day. In a multiservice network it may be natural to extend current practice for the telephone network by identifying a busy period (e.g., the one hour period with the greatest traffic demand)

and modelling arrivals in that period as a stationary stochastic process (e.g., a Poisson process). Traffic demand may then be expressed as the expected combined rate of all active flows: the product of the arrival rate, the mean duration and the mean rate of one flow. The duration of telephone calls is known to have a heavy-tailed distribution [Bol94] and this is likely to be true of other stream flows suggesting that the number of flows in progress and their combined rate are self-similar processes.

1.2.3 Elastic traffic

The second type of traffic we consider consists of digital objects or “documents” which must be transferred from one place to another. These documents might be data files, texts, pictures or video sequences transferred for local storage before viewing. This traffic is elastic in that the flow rate can vary due to external causes (e.g., bandwidth availability) without detrimental effect on quality of service.

Users may or may not have quality of service requirements with respect to throughput. They do for real-time information retrieval sessions where it is important for documents to appear rapidly on the user’s screen. They do not for e-mail or file transfers where deferred delivery, within a loose time limit, is perfectly acceptable.

The essential characteristics of elastic traffic are the arrival process of transfer requests and the distribution of object sizes. Observations on Web traffic provide useful pointers to the nature of these characteristics [AW96, CB96].

The average arrival intensity of transfer requests varies depending on underlying user activity patterns. As for stream traffic, it should be possible to identify representative busy periods where the arrival process can be considered to be stationary.

Measurements on Web sites reported by Arlitt and Williamson [AW96] suggest the possibility of modelling the arrivals as a Poisson process. A Poisson process indeed results naturally when members of a very large population of users independently make relatively widely spaced demands. Note, however, that more recent and thorough measurements suggest that the Poisson assumption may be too optimistic (see Chapter ??). Statistics on the size of Web documents reveal that they are extremely variable exhibiting a heavy-tailed probability distribution. Most objects are very small: measurements on Web document sizes reported by Arlitt and Williamson reveal that some 70% are less than 1 Kbyte and only around 5% exceed 10 Kbytes. The presence of a few extremely long documents has a significant impact on the overall traffic volume, however.

It is possible to define a notion of traffic demand for elastic flows, in analogy with the definition given above for stream traffic, as the product of an average arrival rate in a representative busy period and the average object size.

1.2.4 Traffic aggregations

Another category of traffic arises when individual flows and transactions are grouped together in an aggregate traffic stream. This occurs currently, for

example, when the flow between remotely located LANs must be treated as a traffic entity by a wide area network. Proposed evolutions to the Internet service model such as differentiated services and MPLS (multi-protocol label switching) also rely heavily on the notion of traffic aggregation.

Through aggregation, quality of service requirements are satisfied in a two step process: the network guarantees that an aggregate has access to a given bandwidth between designated end points; this bandwidth is then shared by flows within the aggregate according to mechanisms like those described in the rest of this chapter. Typically, the network provider has the simple traffic management task of reserving the guaranteed bandwidth while the responsibility for sharing this bandwidth between individual stream and elastic flows devolves to the customer. This division of responsibilities alleviates the so-called scalability problem where the capacity of network elements to maintain state on individual flows cannot keep up with the growth in traffic.

The situation would be clear if the guarantee provided by the network to the customer were for a fixed constant bandwidth throughout a given time interval. In practice, because traffic in an aggregation is generally extremely variable (and even self-similar), a constant rate is not usually a good match to user requirements. Some burstiness can be accounted for through a leaky bucket based traffic descriptor, although this is not a very satisfactory solution, especially for self-similar traffic (see Section 1.3.2).

In existing frame relay and ATM networks, current practice is to considerably overbook capacity (the sum of guaranteed rates may be several times

greater than available capacity), counting on the fact that users do not all require their guaranteed bandwidth at the same time. This allows a proportionate decrease in the bandwidth charge but, of course, there is no longer any real guarantee. In addition, in these networks users are generally allowed to emit traffic at a rate over and above their guaranteed bandwidth. This excess traffic, “tagged” to designate it as expendable in case of congestion, is handled on a best effort basis using momentarily available capacity.

Undeniably, the combination of overbooking and tagging leads to a commercial offer which is attractive to many customers. It does, however, lead to an imprecision in the nature of the offered service and in the basis of charging which may prove unacceptable as the multiservice networking market gains maturity. In the present chapter, we have sought to establish a more rigorous basis for network engineering where quality of service guarantees are real and verifiable.

This leads us to ignore the advantages of considering an aggregation as a single traffic entity and to require that individual stream and elastic flows be recognized for the purposes of admission control and routing. In other words, transparency, throughput and accessibility are guaranteed on an individual flow basis, not for the aggregate. Of course, it remains useful to aggregate traffic within the network and flows of like characteristics can share buffers and links without the need to maintain detailed state information.

1.3 OPEN LOOP CONTROL

In this and the next section we discuss traffic control options and their potential for realizing quality of service guarantees. Here we consider open loop, or preventive, traffic control based on the notion of “traffic contract”: a user requests a communication described in terms of a set of traffic parameters and the network performs admission control, accepting the communication only if quality of service requirements can be satisfied. Either ingress policing or service rate enforcement by scheduling in network nodes is then necessary to avoid performance degradation due to flows which do not conform to their declared traffic descriptor.

1.3.1 Multiplexing performance

The effectiveness of open loop control depends on how accurately it is possible to predict performance given the characteristics of variable rate flows. To discuss multiplexing options we make the simplifying assumption that flows have unambiguously defined rates like fluids, assimilating links to pipes and buffers to reservoirs. We also assume rate processes are stationary. It is useful to distinguish two forms of statistical multiplexing: bufferless multiplexing and buffered multiplexing.

In the fluid model, statistical multiplexing is possible without buffering if the combined input rate is maintained below link capacity. As all excess traffic is lost, the overall loss rate is simply $E[(\Lambda_t - c)^+]/E[\Lambda_t]$ where Λ_t is the input

rate process and c is the link capacity. It is important to notice that this loss rate only depends on the stationary distribution of Λ_t and not on its time dependent properties, including self-similarity. The latter do have an impact on other aspects of performance, such as the duration of overloads, but this can often be neglected if the loss rate is small enough.

The level of link utilization compatible with a given loss rate can be increased by providing a buffer to absorb some of the input rate excess. However, the loss rate realized with a given buffer size and link capacity then depends in a complicated way on the nature of the offered traffic. In particular, loss and delay performance are very difficult to predict when the input process is long-range dependent. The models developed in this book are, for instance, generally only capable of predicting asymptotic queue behaviour for particular classes of long-range dependent traffic.

An alternative to statistical multiplexing is to provide *deterministic* performance guarantees. Deterministic guarantees are possible, in particular, if the amount of data $A(t)$ generated by a flow in an interval of length t satisfies a constraint of the form: $A(t) \leq \rho t + \sigma$. If the link serves this flow at a rate at least equal to ρ then the maximum buffer content from this flow is σ . Loss can therefore be completely avoided and delay bounded by providing a buffer of size σ and implementing a scheduling discipline which ensures the service rate ρ [Cru91]. The constraint on the input rate can be enforced by means of a leaky bucket, as discussed below.

1.3.2 The leaky bucket traffic descriptor

Open loop control in both ATM and Internet service models relies on the leaky bucket to describe traffic flows. Despite this apparent convergence, there remain serious doubts about the efficacy of this choice.

For present purposes, we consider a leaky bucket as a reservoir of capacity σ emptying at rate ρ and filling due to the controlled input flow. Traffic conforms to the leaky bucket descriptor if the reservoir does not overflow and then satisfies the inequality $A(t) \leq \rho t + \sigma$ introduced above. The leaky bucket has been chosen mainly because it simplifies the problem of controlling input conformity. Its efficacy depends additionally on being able to choose appropriate parameter values for a given flow and then being able to efficiently guarantee quality of service by means of admission control.

The leaky bucket may be viewed either as a statistical descriptor approximating (or more exactly, providing usefully tight upper bounds on) the actual mean rate and burstiness of a given flow or as the definition of an envelope into which the traffic must be made to fit by shaping. Broadly speaking, the first viewpoint is appropriate for stream traffic, for which excessive shaping delay would be unacceptable, while the second would apply in the case of (aggregates of) elastic traffic.

Stream traffic should pass transparently through the policer without shaping by choosing large enough bucket rate and capacity parameters. Experience with video traces shows that it is very difficult to define a happy medium solu-

tion between a leak rate ρ close to the mean with an excessively large capacity σ , and a leak rate close to the peak with a moderate capacity [RB95]. In the former case, although the overall mean rate is accurately predicted, it is hardly a useful traffic characteristic since the rate averaged over periods of several seconds can be significantly different. In the latter, the rate information is insufficient to allow significant statistical multiplexing gains.

For elastic flows it is, by definition, possible to shape traffic to conform to the parameters of a leaky bucket. However, it remains difficult to choose appropriate leaky bucket parameters. If the traffic is long-range dependent, as in the case of an aggregation of flows, the performance models studied in this book indicate that queuing behaviour is particularly severe. For any choice of leak rate ρ less than the peak rate and a bucket capacity σ which is not impractically large, the majority of traffic will be smoothed and admitted to the network at rate ρ . The added value of a non-zero bucket capacity is thus extremely limited for such traffic.

We conclude that, for both stream and elastic traffic, the leaky bucket constitutes an extremely inadequate descriptor of traffic variability.

1.3.3 Admission control

To perform admission control based solely on the parameters of a leaky bucket implies unrealistic worst case traffic assumptions and leads to considerable resource allocation inefficiency. For statistical multiplexing, flows are typically assumed to independently emit periodic maximally sized peak rate bursts sep-

arated by minimal silence intervals compatible with the leaky bucket parameters [EMW95]. Deterministic delay bounds are attained only if flows emit the maximally sized peak rate bursts *simultaneously*. As discussed above, these worst case assumptions bear little relation to real traffic characteristics and can lead to extremely inefficient use of network resources.

An alternative is to rely on historical data to predict the statistical characteristics of known flow types. This is possible for applications like the telephone where an estimate of the average activity ratio is sufficient to predict performance when a set of conversations share a link using bufferless multiplexing. It is less obvious in the case of multiservice traffic where there is generally no means to identify the nature of the application underlying a given flow.

The most promising admission control approach is to use measurements to estimate currently available capacity and to admit a new flow only if quality of service would remain satisfactory assuming that flow were to generate worst case traffic compatible with its traffic descriptor. This is certainly feasible in the case of bufferless multiplexing. The only required flow traffic descriptor would be the peak rate with measurements performed in real-time to estimate the rate required by existing flows [GKK95, JSD97]. Without entering into details, a sufficiently high level of utilization is compatible with negligible overload probability, on condition that the peak rate of individual flows is a small fraction of the link rate. The latter condition ensures that variations in the combined input rate are of relatively low amplitude, limiting the risk of estimation errors and requiring only a small safety margin to account for the

most likely unfavourable coincidences in flow activities.

For buffered multiplexing, given the dependence of delay and loss performance on complex flow traffic characteristics, design of efficient admission control remains an open problem. It is probably preferable to avoid this type of multiplexing and to instead use reactive control for elastic traffic.

1.4 CLOSED LOOP CONTROL FOR ELASTIC TRAFFIC

Closed loop, or reactive, traffic control is suitable for elastic flows which can adjust their rate according to current traffic levels. This is the principle of TCP in the Internet and ABR in the case of ATM. Both protocols aim to fully exploit available network bandwidth while achieving fair shares between contending flows. In the following sections we discuss the objectives of closed loop control, first assuming a fixed set of flows routed over the network, and then taking account of the fact that this set of flows is a random process.

1.4.1 Bandwidth sharing objectives

It is customary to consider bandwidth sharing under the assumption that the number of contending flows remains fixed (or changes incrementally, when it is a question of studying convergence properties). The sharing objective is then essentially one of fairness: a single isolated link shared by n flows should allocate $1/n^{\text{th}}$ of its bandwidth to each. This fairness objective can be generalized to account for a weight φ_i attributed to each flow i , the bandwidth

allocated to flow i then being proportional to $\varphi_i / \sum_{\text{all flows}} \varphi_j$. The φ_i might typically relate to different tariff options.

In a network the generalization of the simple notion of fairness is max-min fairness [BG87]: allocated rates are as equal as possible subject only to constraints imposed by the capacity of network links and the flow's own peak rate limitation. The max-min fair allocation is unique and such that no flow rate λ , say, can be increased without having to decrease that of another flow whose allocation is already less than or equal to λ .

Max-min fairness can be achieved exactly by centralized or distributed algorithms which calculate the explicit rate of each flow. However, most practical algorithms sacrifice the ideal objective in favour of simplicity of implementation [AC96]. The simplest rate sharing algorithms are based on individual flows reacting to binary congestion signals. Fair sharing of a single link can be achieved by allowing rates to increase linearly in the absence of congestion and decrease exponentially as soon as congestion occurs [CJ89].

It has recently been pointed out that max-min fairness is not necessarily a desirable rate sharing objective and that one should rather aim to maximize overall utility where the utility of each flow is a certain non-decreasing function of its allocated rate [Kel97, KMT98]. General bandwidth sharing objectives and algorithms are further discussed in [MR99a]

Distributed bandwidth sharing algorithms and associated mechanisms need to be robust to non-cooperative user behaviour. A particularly promising solution is to perform bandwidth sharing by implementing per flow, fair queueing.

The feasibility of this approach is discussed by Suter et al. in [SLS98] where it is demonstrated that an appropriate choice of packets to be rejected in case of congestion (namely, packets at the front of the longest queues) considerably improves both fairness and efficiency.

1.4.2 Randomly varying traffic

Fairness is not a satisfactory substitute for quality of service, if only because users have no means of verifying that they do indeed receive a “fair share”. Perceived throughput depends as much on the number of flows currently in progress as on the way bandwidth is shared between them. This number is not fixed but varies randomly as new transfers begin and current transfers end.

A reasonable starting point to evaluating the impact of random traffic is to consider an isolated link and to assume new flows arrive according to a Poisson process. On further assuming the closed loop control achieves exact fair shares immediately as the number of flows changes, this system constitutes an M/G/1 processor sharing queue for which a number of interesting results are known [Kle75]. A related traffic model where a finite number of users retrieve a succession of documents is discussed by Heyman et al. in [HLN97].

Let the link capacity be c and its load (arrival rate \times mean size / c) be ρ . If $\rho < 1$, the number of transfers in progress N_t is geometrically distributed, $\Pr\{N_t = n\} = \rho^n(1 - \rho)$, and the average throughput of any flow is equal to $c(1 - \rho)$. These results are insensitive to the document size distribution. Notice

that the expected response time is finite for $\rho < 1$, even if the document size distribution is heavy tailed. This is in marked contrast with the case of a first come, first served M/G/1 queue where a heavy-tailed service time distribution with infinite variance leads to infinite expected delay for any positive load. In other words, for the assumed self-similar traffic model, closed loop control avoids the severe congestion problems associated with open loop control. We conjecture that this observation also applies for a more realistic flow arrival process.

If flows have weights φ_i as discussed above, the corresponding generalization of the above model is discriminatory processor sharing as considered, for example, by Fayolle et al. [FMI80]. The performance of this queueing model is not insensitive to the document size distribution and the results in [FMI80] apply only to distributions having finite variance. Let $R(p)$ denote the expected time to transfer a document of size p . Figure 1.1 shows the normalized response time $R(p)/p$, as a function of p for a two class discriminatory processor sharing system with the following parameters: unit link capacity, $c = 1$; both classes have a unit mean, exponential size distribution and an arrival rate of $1/3$; flows of class i have sharing parameter φ_i where $\{\varphi_1, \varphi_2\} = \{1, 2\}$.

From the figure we note that the sharing parameters ensure effective discrimination for the transfer time of short documents but that throughput for both classes tends to the limit $c(1 - \rho)$ as document size increases. The limiting large object throughput is explained by the fact that, whatever its sharing parameter φ_i , a very long transfer utilizes all the bandwidth except

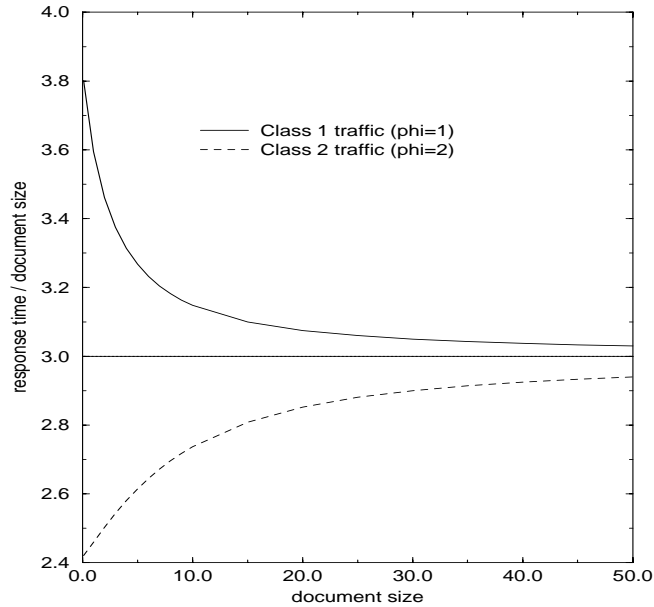


Fig. 1.1 Normalized response time $R(p)/p$ for discriminatory processor sharing.

that required by other users, equal on average to $c\rho$.

Results for hyperexponential distributions (not reported here) show that discrimination is more effective as the document size distribution variability increases. It is likely therefore that for a heavy-tailed distribution most document transfers will see an improvement in throughput with an increasing weight, although the improvement is less than proportional and still tends to disappear for exceptionally long documents.

Notice that throughput of large objects is not affected by the rate assigned to the transfer of short objects which start and finish within the transfer time of the former. Overall throughput can therefore be improved by giving priority

to short objects. Indeed, it is known that the response time performance of a shared resource is optimized on using the shortest remaining processing time first (SRPT) scheduling discipline: a controller is assumed to know the remaining volume of data of all documents to be transferred and devotes link capacity exclusively to the smallest; if a new arrival concerns a document whose size is less than that of the document in service, the latter is pre-empted; any pre-empted transfer resumes service where it left off, as soon as its remaining volume is again smaller than that of any other pending request.

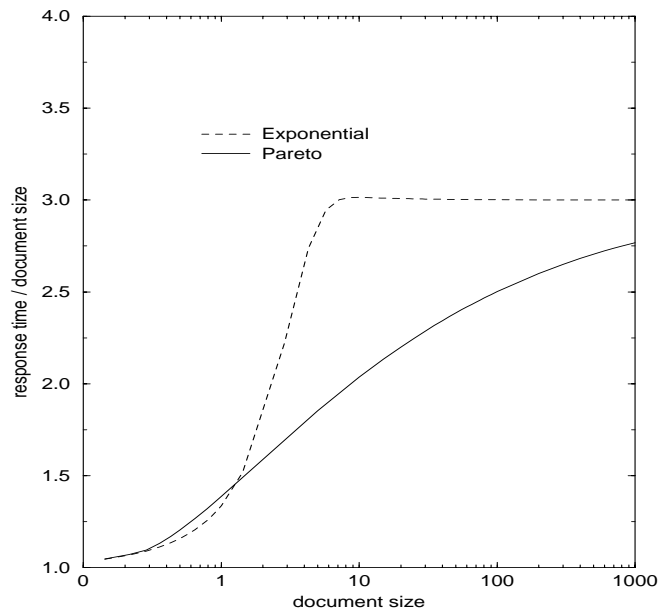


Fig. 1.2 Normalized response time $R(p)/p$ time for SRPT scheduling

The performance of SRPT was studied by Schrage and Miller [SM66]. They derive expressions for the response time $R(p)$ of a document of size p under an

assumption of Poisson arrivals and general service time distribution. Figure 1.2 shows a numerical evaluation of their formulas for exponential and infinite variance Pareto distributed document sizes, respectively. Link load is $2/3$, as in the example of Figure 1.1. The p axis, in units of the mean document size, is on a log scale to capture the heavy tail particularity of the Pareto distribution. The normalized response time $R(p)/p$ is considerably less than that of perfectly fair sharing (i.e., the processor sharing model), equal here to 3 for all values of p . It is interesting to note that, in this system, the response time for medium to large documents improves on passing from short range to long-range dependent processes.

Implementation of SRPT in the case of a single link would, of course, be very complex and the appropriate extension of this principle to a network remains unclear. However, it does provide a clear illustration that fairness, or weighted fairness, is not necessarily a useful objective in bandwidth sharing. In particular, both users and network provider stand to gain by employing a flow control protocol which discriminates in favour of short documents.

The processor sharing model illustrates how performance can deteriorate suddenly as offered load ρ increases through 1: if link bandwidth c is high, throughput performance is good even when ρ is close to 1. For heavier loads, throughput is zero and the number of transfers in progress increases indefinitely. Of course, the model then ceases to be accurate, since many real users will abandon transfers as soon as they begin to notice the effects of such congestion. Since an abandoned or otherwise incomplete transfer serves no useful

purpose and only adds to congestion, goodput can be improved by employing admission control.

1.4.3 Admission control for elastic traffic

Admission control, by limiting the number of flows using any given link, ensures that throughput never decreases below some minimum acceptable level for flows which are admitted. Exactly what would constitute a minimum acceptable throughput is not clear. The choice depends on a trade-off between the extra utility of accepting a new flow and the risk that existing transfers would be prematurely interrupted if their rate were decreased. It does seem clear that such a minimum exists (though it may be different for different users) since otherwise a saturated network would be unstable [MR99b].

Admission control does not necessarily imply a complex flow set up stage with explicit signalling exchanges between user and network nodes. This would be quite unacceptable for most elastic flows which are of very short duration. We envisage a network rather similar to the present Internet where users simply send their data as and when they wish. However, nodes implementing admission control would keep a record of the identities of existing flows currently traversing each link in order to be able to recognize the arrival of a packet from a new flow. Such a packet would be accepted and its identifier added to the list of active flows if the number of flows currently in progress were less than a threshold, and would otherwise be rejected. A flow would be erased from the list if it sent no packets during a certain time out interval.

Although many additional practical considerations would need to be addressed, such a control procedure does seem feasible technically given recent developments in router technology [KS98, KLS98]. Note finally, that knowledge of the state of network links in terms of the number of flows currently in progress would also allow intelligent routing strategies where flows are not sent blindly to saturated links when other paths are available.

1.5 TOWARDS A SIMPLE SERVICE MODEL

Given the above discussion on possible control options, it is tempting to speculate on the simplest service model capable of meeting identified requirements.

1.5.1 Service classes

We envisage a service model with just two service classes, one based on open loop control for stream traffic and the other using closed loop control for elastic traffic. In this service model, flows destined for the first class declare just a peak rate which is actively policed by packet spacing at the network ingress. Measurement-based admission control would be used to ensure negligible data loss assuming bufferless multiplexing. Although, in practice, a small buffer is necessary to account for the non-fluid nature of traffic, delay and delay variation remain very small. Loss and delay performance are independent of any long-range dependence in the rate process of flows. A low loss rate (10^{-9} , say) is compatible with a reasonable average link utilization (50%, say) if the

peak rate of flows is not more than a small fraction of the link bandwidth (1/100, say) [RMV96, chapter 16].

The necessary characteristics of the closed loop control are less well understood. We can rely on users reacting intelligently to congestion signals, as in TCP, if the network additionally implements queue management mechanisms preventing uncooperative flows from adversely affecting the quality of service of other users. A promising solution is to perform per flow queueing with flow identification performed “on the fly”, as suggested in [SLS98]. The identification of the set of flows currently using a link allows the implementation of a simple admission control procedure whereby any packets from new flows are rejected when the number of flows in progress exceeds a link capacity dependent threshold.

Sharing link capacity dynamically between stream and elastic flows is advantageous for both types of traffic: a very low loss rate for stream traffic is not incompatible with reasonable utilization if elastic traffic constitutes a significant proportion of the total load; elastic flows gain greater throughput by being able to exploit the residual capacity necessarily left over by stream traffic to meet data loss rate and blocking probability targets. Admission control for both stream and elastic flows would take account of the measured stream load and the current count of the number of active elastic flows.

Simple head of line priority is sufficient to meet the delay requirements of stream traffic while per flow queueing is the preferred solution for elastic traffic. Fair queueing among elastic flows leads to fair bandwidth sharing.

However, performance could be improved by implementing packet scheduling schemes giving priority to short documents. The performance of rate sharing schemes like fair queueing and SRPT does not appear to be adversely affected by the heavy-tailed nature of the document size distribution.

For any given application, a user might choose to set up a stream or an elastic flow. The choice depends on quality of service and cost. We have argued that open loop control can meet the strict delay requirements of stream traffic while closed loop control provides higher throughput for the transfer of elastic documents. The issue of providing price incentives to influence user choices is discussed in the next section (see also [Rob98, Odl98]).

1.5.2 The impact of charging

For largely historical reasons, most users of the Internet today are charged on a flat rate basis. They pay a fixed monthly charge which is independent of the volume of traffic they produce, although the charge does depend on the capacity of their network access line. The major advantage of *flat rate pricing* is its simplicity leading to lower network operating costs. A weakness is its inherent unfairness, a light user having to pay as much as a heavy user. A more immediate problem is the absence of restraint inherent in this charging scheme which may be said to contribute to the present state of congestion of the Internet.

Network usage can be controlled by the introduction of usage sensitive charging with rates determined by the level of congestion. This is the principle

of *congestion pricing*. Congestion pricing ideally leads to an economic optimum where available resources are used to produce maximum utility. While theoretically optimal schemes like the “smart market” [MV95] are unlikely to be implemented for reasons of practicality, it has been argued that the congestion control objective can be achieved simply by offering a number of differentially priced service classes with charges increasing with the expected level of quality of service [SCEH95]. Users determine the amount they are charged by their choice of service class. They have an incentive to choose more expensive classes in times of congestion. Such schemes suffer from a lack of transparency: how can users tell if the network provider isn’t deliberately causing congestion? why should they pay more to an inefficient provider? are they currently paying more than they need to, given current traffic levels? Note that congestion pricing is not generally employed in other service industries subject to demand overloads such as electricity supply, public transportation or the telephone network.

An alternative is to charge for use depending on the amount of resources used per transaction, accounting possibly for distance (number of hops) as well as volume. We refer to such a charging scheme as *transaction pricing*. Transaction pricing is widely used in the telephone network (with the notable exception of local networks in North America) where switches and links are sized to ensure that congestion occurs only exceptionally. The price must be set at a value allowing the network operator to recover the cost of investment. Differential pricing according to the time of day is used to smooth out the

demand profile to some extent but this is not generally viewed as a congestion control mechanism.

Choice between flat rate pricing, congestion pricing and transaction pricing depends among other things on their ability to assure the economic viability of the network provider. Congestion pricing is intended to optimize the use of a network, not to recover the cost of installed infrastructure which is regarded as a “sunk cost” in the economic optimization. If the network is well provisioned and always offers good quality of service, for example, costs must be entirely recovered by flat rate access charges. Transaction pricing has proved successful for telephone network operators, but then so has flat rate pricing in the case of North American local networks. Transaction pricing has the advantage of distributing the cost of shared network resources in relation to usage. In addition to being appealing from a fairness point of view, this is in line with the trend in telecommunications for “unbundling” and cost related pricing.

A second major issue is the complexity of implementing the different schemes. Any move from flat rate pricing appears as a major change for the Internet, requiring accounting and billing systems at least as complex as those of the telephone network. The cost of such systems must be weighed against any expected improvement in efficiency.

In proposing a simple two-class model, we have in mind a mixture of flat rate pricing and transaction pricing where the role of the latter would be to allow users to be charged in relation to their use of shared resources. We argue in [Rob98] that, in a large network sized to offer good quality of service,

resource provision is largely independent of whether the traffic is stream or elastic. This suggests a simple tariff based just on the number of bytes crossing an interface.

A likely evolutionary step is that cost related charging be introduced for large users, including ISPs connected to a backbone, with individual small users continuing to pay only a flat rate charge.

The simple service model makes no distinction between elastic documents like Web pages intended for immediate display and documents like mail whose delivery is deferrable. Users do not require minimal throughput for the latter and would arguably expect to pay less for their transport. A possible solution is that deferrable documents transit via servers, operated by a “postal service”, external to the transport network of routers and links. Users deliver a document directly to a local server which then takes charge of forwarding it to its destination(s), generally via intermediate servers. The users pay the “postal service” which in turn pays the transport network. The service is cheaper for end users because the servers can send data in off-peak hours and negotiate special tariff arrangements with the network provider.

1.6 NETWORK SIZING

Traffic engineering for a multiservice network handling both stream and elastic traffic is still a largely unexplored field. In this section we suggest how it may be possible to generalize the methods and tools developed over the years for

dimensioning the telephone network.

1.6.1 Provisioning for stream traffic

To determine the network capacity required to meet a target blocking probability for stream flows, it is necessary to make assumptions about the arrival process of new demands, their rate and their duration. For illustration purposes, we consider a simple traffic model consisting of one link receiving traffic from a very large population of users. Details and more general models may be found in [RMV96], for example.

First assume that it is possible to identify m distinct homogeneous classes, flows of each class having a common rate distribution. Flows from class i arrive according to a Poisson process of intensity λ_i (requests per second) and have an expected duration of $1/\mu_i$ seconds. Their peak rate is p_i . For a fixed (fairly large) link capacity c , the impact of a flow of class i on the probability of data loss can be summarized in a single figure, the effective bandwidth: the effective bandwidth e_i is such that the probability of data loss is negligible (less than a target value) as long as $\sum n_i e_i \leq c$, where n_i is the number of class i flows in progress.

Although measurement based admission control does not rely on the identification of the different classes (a new flow is denied access if its peak rate is greater than a real-time estimate of available bandwidth), for dimensioning purposes we can assume a flow of class j will be blocked if $\sum n_i e_i > c - e_j$. With this blocking condition and the assumption of Poisson arrivals, the dis-

tribution of the n_i has a well known product form enabling computation of the blocking probability. Note that blocking probabilities and data loss rates are insensitive to the distribution of flow duration.

A reasonable approximation for the blocking probability of a flow with peak rate p_i when c is large with respect to the e_i is given by:

$$B_i \approx \frac{p_i}{\delta} E(a/\delta, c/\delta) \quad (1.1)$$

where $a = \sum e_i \frac{\lambda_i}{\mu_i}$, $\delta = \sum e_i^2 \frac{\lambda_i}{\mu_i} / a$ and $E(a, n) = \frac{a^n}{n!} / \sum_{i \leq n} \frac{a^i}{i!}$ is Erlang's formula.

Formula (1.1) is a simplification of the formulas given by Lindberger [Lin94]. It is less accurate but more clearly demonstrates the structural relationship between performance and traffic characteristics. Instead of identifying traffic classes with common traffic characteristics, it may prove more practical to estimate the essential parameters a and δ directly.

It is well known that application of Erlang's formula leads to scale economies: to achieve a low blocking probability and high utilization (a/c), it is necessary to have a large capacity c . For multirate traffic with blocking probabilities given by (1.1), the same requirement implies a high value of c/δ . The line labelled "stream" in Figure 1.3 shows how achievable utilization a/c in a simple Erlang loss system varies with c for a target blocking probability of 0.01.

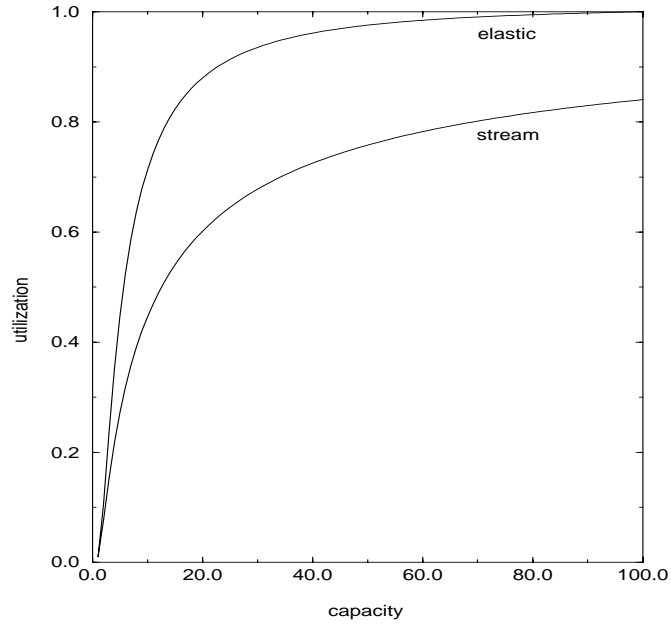


Fig. 1.3 Achievable utilization for stream and elastic traffic

1.6.2 Provisioning for elastic traffic

Following the simple service model introduced in Section 1.5, we assume throughput quality of service is satisfied by limiting the number of elastic flows on a link and seek to dimension link capacity such that the blocking probability is less than some low target value ϵ .

Consider first an isolated link handling only elastic flows. Assuming Poisson arrivals, a minimum throughput requirement θ , exact fair shares (i.e., processor sharing service) and a link bandwidth of $c = n\theta$, the probability of blocking is equal to the saturation probability in an M/G/1 processor sharing

queue of capacity n :

$$B_e = \rho^n(1 - \rho)/(1 - \rho^{n+1}) \quad (1.2)$$

where ρ is the link load.

Since elastic flows use bandwidth more efficiently, blocking probability (1.2) can be considerably less than the corresponding probability for stream traffic requiring constant rate θ , as given by Erlang's formula $E(n\rho, n)$. The line labelled "elastic" in Figure 1.3 shows achievable utilization ρ for elastic traffic such that B_e , given by (1.2), is equal to 0.01. These results clearly illustrate the scale economies effect and the greater efficiency of elastic sharing.

The advantage of elastic sharing with respect to rigid rate allocations is somewhat mitigated in a network where flows cannot always attain a full share of available link bandwidth because of congestion on other links of their path and their own limited peak rate. If, however, the flows can at least attain rate θ and this rate is guaranteed by admission control on every network link, the utilization predicted by the Erlang formula constitutes a lower bound. In other words, the Erlang formula can be used as a conservative dimensioning tool to determine the traffic capacity of a link dedicated to elastic traffic: a link of capacity c can handle a volume of elastic traffic A_e (flow arrival rate \times average size) with minimum throughput θ and blocking probability less than ϵ if $E(A_e, c/\theta) \leq \epsilon$. Given the scale economies achieved with the Erlang formula, this simple dimensioning approach is efficient if c/θ is large (e.g., $A_e/c > 0.8$ if $c/\theta > 100$ for a 1% blocking probability).

An advantage of the above approach is that the integration of stream and elastic traffic is taken into account simply by including the latter as an additional traffic class in the multirate dimensioning methods alluded to in the previous section.

1.7 CONCLUSION

The realization of quality of service guarantees in a multiservice network depends more on sound traffic engineering than on the definition of a service model allowing priority access for an undefined number of privileged users.

We have argued that the service model should facilitate traffic engineering by distinguishing two broad categories of traffic: stream and elastic. For each category, the appropriate entity for traffic management is an individual flow (e.g., one videoconference, one file transfer, ...) and not either an isolated packet or some aggregation of flows. A tentative simple service model is based on just two traffic classes.

One class destined for stream traffic is based on open loop control and uses “bufferless multiplexing” with measurement-based admission control. This choice enables delay and loss rate performance guarantees, even for self-similar flows. The leaky bucket is not useful as a traffic descriptor and the only traffic parameter required here is the flow peak rate.

The second service class uses closed loop control to share bandwidth between elastic flows. We advocate a lightweight form of admission control for

elastic traffic, requiring that each link identify the flows it is currently transporting. Per flow queueing would be useful to enforce fairness, or to share bandwidth more efficiently by giving priority to short transfers, for example. In the simple bandwidth sharing models considered here, the heavy-tailed distribution of the size of transferred documents does not adversely affect the response time performance of closed loop control.

We consider charging as a means to recover the network provider's costs rather than as a tool for congestion control. Prices would ideally be set to just ensure profitability when the network is dimensioned to handle all the offered traffic with good quality of service. There appears no essential reason to price stream and elastic traffic differently per byte transported. Users would naturally choose the service class best suited to their quality of service requirements: low delay for stream flows, high throughput for elastic flows.

We have given some indications of how traditional traffic engineering practice might be extended to a multiservice network based on the proposed simple service model. The basic principle is that transparency and throughput quality of service are assured by means of admission control acting at flow level, while the network is sized to produce a sufficiently low blocking probability.

REFERENCES

- AC96. A. Arulambalam and X.Q. Chen, Allocating fair rates for available bit rate service in ATM networks. *IEEE Communications Magazine* 92-100,

- 1996.
- AW96. M.F. Arlitt, C. Williamson. Web server workload characterization: The search for invariants. *ACM Sigmetrics 96*, 1996.
- BG87. D. Bertsekas and R. Gallager, *Data Networks*. Prentice Hall, 1987.
- Bol94. V. Bolotin. Telephone circuit holding time distribution, Proceedings of ITC 14 (J. Labetoulle, J. Roberts (eds). *The fundamental role of teletraffic in the evolution of telecommunications networks*) Elsevier, 1994.
- CB96. M. Crovella, A. Bestavros. Self-similarity in World Wide Web traffic: Evidence and possible causes. *ACM Sigmetrics 96*, 1996.
- CJ89. D.M. Chiu and R. Jain, Analysis of the increase and decrease algorithms for congestion avoidance in computer networks. *Computer Networks and ISDN Systems*, Vol 17, pp 1-14, 1989.
- Cru91. R. L. Cruz. A calculus of network delay. Part I: Network elements in isolation. *IEEE Trans. on Information Theory*, Vol 37, pp 114-31, 1991.
- EMW95. A. Elwalid, D. Mitra, R. H. Wentworth. A new approach to allocating buffers and bandwidth to heterogeneous regulated traffic in an ATM node. *IEEE JSAC*, Vol 13, No 6, August, 1995.
- FMI80. G. Fayolle, I. Mitrani, R. Iasnogorodski. Sharing a processor among many jobs. *Journal of the ACM*, Vol 27, No 3, pp519-532, 1980.
- Fra97. M. Frater. Origins of long range dependence in variable bit rate video

traffic. Proceedings of ITC15 (V. Ramaswami, P.E. Wirth (Eds). Tele-traffic contributions for the information age), Elsevier, 1997.

GKK95. R. Gibbens, F. Kelly, P. Key, A decision theoretic approach to call admission control in ATM networks, IEEE JSAC, Vol 13, No 6, August, 1995.

GW94. M. Garrett, W. Willinger. Analysis, modeling and generation of self-similar VBR video traffic. Proceedings of SIGCOMM'94, 1994,

HLN97. D. Heyman, T. Lakshman, A. Neidhart. A new method for analysing feedback-based protocols with application to engineering Web traffic over the Internet. ACM Sigmetrics '97, 1997.

JSD97. S. Jamin, S. J. Shenker, P. B. Danzig. Comparison of measurement-based admission control algorithms for controlled load service. Proc. INFOCOM'97, April 1997.

Kel97. F. Kelly. Charging and rate control for elastic traffic. Europ. Trans. Telecom. Vol 8, pp33-37, 1997.

Kle75. L. Kleinrock. *Queueing Systems, Volume 2*, J. Wiley & Sons. 1975.

KLS98. V. P. Kumar, T. V. Lakshman, D. Stiliadis. Beyond best effort: Router architectures for differentiated services of tomorrow's Internet. IEEE Communications Magazine, Vol 36, No 5, May 1998.

KMT98. F. Kelly, A. Maulloo and D. Tan. Rate control for communication

- networks: shadow prices, proportional fairness and stability. *Journal of the Operational Research Society* 49, 1998.
- KS98. S. Keshav, R. Sharma. Issues and trends in router design. *IEEE Communications Magazine*, Vol 36, No 5, May 1998.
- Lin94. K. Lindberger. Dimensioning and design methods for integrated ATM networks. Proceedings of ITC 14 (J. Labetoulle, J. Roberts (eds). *The fundamental role of teletraffic in the evolution of telecommunications networks*) Elsevier, 1994.
- MR99a. L. Massoulié, J. Roberts. Bandwidth sharing: objectives and algorithms. Proceedings of INFOCOM '99, March 1999.
- MR99b. L. Massoulié, J. Roberts. Arguments in favour of admission control for TCP flows. Proceedings of ITC 16, to be published by Elsevier, 1999.
- MV95. J. MacKie-Mason, H. Varian. Pricing the Internet, in B. Kahin, J. Keller (eds), *Public Access to the Internet*, Prentice Hall, 1995.
- Odl98. A. M. Odlyzko. The economics of the Internet: Utility, utilization, pricing and quality of service. Preprint, 1998.
- RB95. A. R. Reibman, A. W. Berger. Traffic descriptors for VBR videoconferencing over ATM networks. *IEEE/ACM Trans. on Networking*, Vol 3, No 3, June 1995.
- RMV96. J. Roberts, U. Mocchi, J. Virtamo (Eds). *Broadband Network Teletraffic (Final Report of COST 242)*, LNCS 1155, Springer Verlag, 1996.

Rob98. J. Roberts. Quality of service guarantees and charging in multiservice networks. IEICE Trans Commun. Special issue on ATM traffic control and performance evaluation, Vol E81-B, No 5, 1998.

SCEH95. S. Shenker, D. Clark, D. Estrin, S. Herzog, "Pricing in computer networks: Reshaping the research agenda", Telecommunications Policy, Vol 26, pp 183-201, 1996.

SLS98. B. Suter, T. V. Lakshman, D. Stiliadis. Design considerations for supporting TCP with per flow queueing. Proceedings of INFOCOM'98, San Francisco, 1998.

SM66. L. Schrage, L. Miller. The M/G/1 queue with the shortest remaining processing time first discipline. Operations Research, pp 670-684, 1966.