

IP traffic and QoS control: the need for a flow-aware architecture

T. Bonald, S. Oueslati-Bouahia, J. Roberts

France Telecom R&D

38-40, rue du Général Leclerc

92794 Issy-Moulineaux Cedex 9

France

Phone: +33 1 45 29 58 82, Fax: +33 1 45 29 60 69

{Thomas.Bonald,Sara.Bouahia,James.Roberts}@francetelecom.com

Abstract

In this paper we challenge the accepted wisdom that over-provisioning and the various mechanisms of the Intserv and Diffserv models can be combined to constitute a viable and effective QoS paradigm for the commercial Internet. The basis of our claim is an analysis of the statistical nature of demand and of the scope for QoS control. We conclude that the key to QoS is effective control of demand overload realized by flow-aware admission control. The lightweight measurement-based implementation envisaged does not pose insurmountable problems of scalability.

Keywords QoS, Traffic theory, Admission control, Service differentiation

1 Introduction

In this paper we argue that neither over-provisioning, Intserv nor Diffserv, nor a judicious combination of all three constitutes a cost-effective quality of service solution for the commercial Internet. This opinion is based on an analysis of how perceived quality depends on the statistical nature of demand and the extent to which it can be controlled by provisioning and resource management. We find the above proposed solutions wanting with respect to the results of this analysis and deduce the need for a new QoS paradigm.

The considered networking context is that of a commercial Internet whose viability depends solely on the sale of transport services. Conclusions might be different if the network were a public service paid for out of taxes or if transport were provided free of charge by an entrepreneur using it to sell content. However, the considered context corresponds to the current business model and most of the identified constraints and requirements would be common to private networks. A commercial network is cost effective when it federates the demands of a large customer base. We generally assume, therefore, that the network realizes a high degree

of sharing and that resource requirements for individual demands are small compared to network capacity. We also suppose a competitive environment where the provider has the incentive to minimize cost.

The position we defend is unlikely to be popular and is at odds with much accepted wisdom. In the interests of clarity and to facilitate discussion we therefore present our arguments in the form of a series of claims. The first claim is as follows.

Claim 1 *Over-provisioning solves almost all QoS problems but is not a viable solution*

The spectacular success of the Internet and the best effort paradigm is one of the defining events of the present period. This success has not yet been compromised by the absence of QoS mechanisms. Most service providers offer excellent quality of service simply by keeping link utilization low (less than 50%, say). Packet delays are then very small and loss almost negligible. The network is virtually *transparent* with respect to quality degradation for any application.

Transparency is one aspect of quality of service. A second is *accessibility*. If the network cannot assure transparency for a given service, that service is inaccessible. This occurs in a best effort network when traffic demand exceeds available capacity. Accessibility is denied in overload because the arrival rate of new demands exceeds the completion rate leading to a state of quasi-saturation. Demand overloads occur even in nominally over-provisioned networks due to a variety of reasons including inaccurate traffic forecasts and equipment failures. The cost of over-provisioning depends on the kinds of risk taken into account and the announced QoS guarantees. It can be considerable.

The best effort network does not distinguish transparency and accessibility. Quality of service control tends to be “all or nothing”. In overload, the network is “opaque” (or, at best, translucent) and service is inaccessible for all. There is no means to ensure accessibility just for the customers who would be prepared to pay the implied cost.

QoS service models like Intserv and Diffserv define mechanisms allowing service differentiation, either with respect to transparency or accessibility. However, these mechanisms are inadapted to the statistical nature of traffic and are either ineffective or inefficient. The ideal QoS paradigm would retain the ubiquity and simplicity of the best effort architecture while implementing accessibility controls to preserve efficiency in overload. The QoS paradigm proposed in this paper would achieve this by separating transparency and accessibility guarantees and controlling traffic at flow level.

In the next section, we discuss the statistical nature of Internet traffic and elaborate on user requirements for transparency and accessibility. Section 3 summarizes known results on the way transparency depends on flow traffic characteristics and the assumed bandwidth sharing mechanisms. The claims in this section highlight the advantages of bufferless multiplexing and fair sharing applied to streaming and elastic traffic, respectively. In Section 4 we critically appraise the effectiveness of the accessibility control mechanisms of “classical” QoS architectures. Finding these mechanisms deficient, we proceed in Section 5 to an outline of the proposed flow-aware QoS paradigm.

2 Characterizing demand

To meet transparency and accessibility QoS requirements necessitates an understanding of the nature of traffic variations. We distinguish systematic long-term traffic variations due to regular patterns of user activity and more rapid short-term variations reflecting the uncoordinated and independent actions of a large user population. We then discuss the nature of QoS requirements.

Claim 2 *Long-term traffic variations reflect user activity and are recurring*

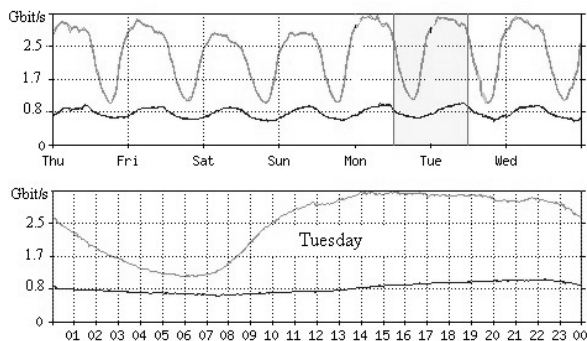


Figure 1: Weekly and daily demand profiles on an OC192 link

Systematic long-term variations are typified by those depicted in Fig. 1. This shows the evolution of traffic

carried over an over-provisioned high capacity link over one week and one day. Traffic in bits/s is derived from byte counts sampled at 5 minute intervals. The variations in rate reflect systematic changes in the activity of the user population throughout the day. There is a clearly recurring busy period occurring in the afternoon. Traffic in this period attains roughly the same value on successive working days.

The daily average busy period load is subject to statistical variations as well as an underlying growth trend. These factors must be taken into account in determining a set of representative traffic demands to be used for provisioning.

A typical provisioning criterion is to require that link utilization, given representative demand, is less than a certain threshold. The choice of this threshold or, more generally, the definition of a provisioning rule ensuring required transparency and accessibility, depends on the impact of the short-term stochastic component of traffic variations.

Claim 3 *Short-term traffic variations can be modeled simply and accurately at flow level*

The impact of short-term stochastic variations can be evaluated on assuming they are produced by an appropriate stationary process. The intensity of this process is equal to the assumed representative demand.

Characteristics of IP traffic at packet level are notoriously complex [1]. Arguably, however, this complexity derives from much simpler flow level characteristics. It proves most convenient to study broad behavior using a flow-based traffic model and to deduce packet level performance, as necessary, in a second phase.

By “flow” we mean here the set of packets related to an instance of some application observed at a given point in the network with an inter-packet interval less than a few seconds. This is a rather vague definition which is not necessarily useful for identifying a flow in practice. The flow may, for example, include several TCP connections used for transferring a single document such as a Web page.

Flows generally occur within “sessions”. A session observed at a given point in the network consists of a sequence of flows separated by silent periods. We call the inter-flow intervals think-times. It is not generally possible to identify a session by simply observing packets in the network.

The session relates to some extended activity undertaken by a user or group of users. This might, for instance, relate to Web browsing, e-commerce, consulting e-mail or playing a networked game. An essential defining characteristic is that, for all practical purposes, sessions are mutually independent. When the user population is large, and each user contributes a small proportion of the overall traffic, independence naturally leads to a Poisson session arrival process. Empirical evidence derived from situations where sessions and flows

can be easily identified suggests this property is one of the rare Internet traffic invariants [2].

The statistical characteristics of a session, including distributions of flow size, think-time length and number of flows, as well as any correlation between successive flows, clearly depend on the applications. The resulting flow arrival process is highly complex and depends significantly on the underlying mix of applications.

Claim 4 *There are just two types of transparency requirement: “signal conservation” and “throughput conservation”*

The broad requirement is for the network to be virtually invisible with respect to the quality degradation experienced by end-to-end applications. We distinguish two kinds of transparency: signal conservation and throughput conservation.

Signal conservation is required by real time and streaming audio and video applications. The signal is contained in the rate of the transmitted data stream so that signal conservation implies limits on acceptable packet loss and delay. We refer to traffic produced by such applications as streaming traffic. Individual streaming flows generally have variable bit rate, due to the use of compression coding.

The second kind of transparency applies to applications involving the transfer over the network of some form of digital document. The digital document in question might, for instance, be an e-mail, a Web page or an MP3 track. The resulting traffic is called elastic since the data rate can be varied without significant detriment to quality of service which depends on the overall transfer time. By throughput conservation we mean that the network does not significantly lengthen the transfer time compared to what it would be if the links had unlimited capacity. This transfer time is then determined by rate limitations independent of the network, including the user’s access line rate, modem speed and server capacity.

It is possible to distinguish different classes of streaming or elastic applications according to their required *degree* of transparency (packet loss rate, expected throughput,...). However, requirements are rarely absolute and applications can generally adapt to the quality offered by the network. The latter in practice depends more on what is technologically and economically feasible than on precise user requirements.

Experiments on subjective appreciation of transparency performed by Bouch *et al.* make the important observation that, while requirements depend critically on the underlying task, users generally expect quality to be predictable [3].

The distinction between streaming and elastic traffic is robust with respect to evolutions in network usage. The introduction of new applications just changes their relative proportions.

The traffic produced by adaptive streaming applications may be considered as a third, hybrid category. The transparency requirements for such applications are imprecise, however, and significant rate changes would hardly respect the predictability requirement identified in [3].

Claim 5 *Transparency requirements and accessibility requirements are largely orthogonal*

It is not possible for a network to be transparent all of the time. Equipment failures, inaccurate forecasts and unpredictable traffic surges can all result in available capacity being insufficient to meet demand. We qualify the network’s capacity to provide transparency as accessibility.

It is important to clearly distinguish the two QoS notions of transparency and accessibility. It appears to be a frequent misunderstanding that the rate adaptivity of elastic traffic preserves accessibility, albeit to a reduced degree of transparency. For example, removing half the capacity of a link at worst halves the throughput of ongoing elastic flows. In fact, the number of flows in progress depends on how much capacity is available and tends to increase indefinitely whenever this is less than demand (equal to the product flow arrival rate \times average flow size). Throughput is then not just halved but tends rapidly to zero. The separation of transparency and accessibility is key to the position defended in this paper.

Users have different accessibility requirements depending on the importance of the underlying task. These requirements are largely orthogonal to transparency requirements. For instance, it may be equally important for a user to make an urgent phone call as to consult vital data for an important transaction. A recognized deficiency of the best effort architecture is its incapacity to ensure accessibility for essential traffic such as that of the emergency services. Commercial considerations might further dictate that business customers experience a higher degree of accessibility than leisure users.

Understanding traffic characteristics is essential in defining provisioning procedures capable of meeting the transparency requirements of applications. In the next section, we examine the consequences of known results on the relationship between demand, capacity and performance.

3 Provisioning for transparency

The network is transparent when sufficient, or adequate, capacity is available. What is meant by adequate provisioning depends on traffic characteristics. In this section we suppose traffic can be described as a stationary process (cf. Claim 3) and evaluate the limits of achievable utilization depending on the characteristics of this process and on the way bandwidth is shared.

We suppose the network distinguishes streaming and elastic traffic and implements a scheduling mechanism (e.g., priority queuing) to effectively isolate streaming flows from the impact of bursty elastic traffic.

Claim 6 *The best way to ensure transparency for streaming flows is “bufferless” statistical multiplexing*

This claim derives from known results on the performance of statistical multiplexing of variable rate flows. It is convenient first to adopt the simplifying assumption that streaming flows emit data continuously like a fluid. In this fluid approximation there is a clear distinction between buffered and bufferless multiplexing. Buffered multiplexing relies on a queue (or “reservoir”) to absorb momentary input rate excess. With bufferless multiplexing, data is lost whenever the input rate exceeds the service rate. Buffered multiplexing allows higher utilization for the same loss rate but implies more complex traffic management, as discussed below.

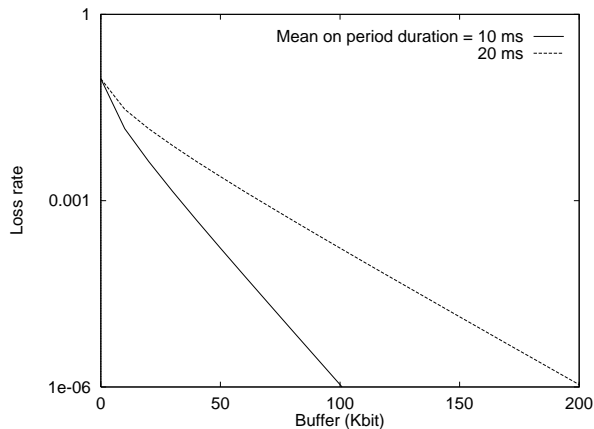


Figure 2: Overflow probability against buffer size for on-off flows

The main disadvantage of using a buffer is that the overflow probability depends significantly on flow characteristics. To illustrate this, Fig. 2 depicts the probability of overflow against buffer size for two sets of streaming flows. The lower line corresponds to a superposition of 20 independent on-off flows with exponentially distributed on and off periods and an activity rate of 0.33. For the upper line the only difference is that the mean on and off periods are doubled. The link capacity is ten times the flow peak rate.

These results demonstrate that the buffer requirement for a given overflow probability objective is proportional to the length of an on period or burst (it doubles in this example). Roughly the same conclusion would apply had we kept the same mean burst length but doubled the standard deviation (e.g., suppose the burst length distribution were hyper-exponential with exponentially distributed bursts either of mean duration 20 ms or a tiny value ϵ with equal probability).

An even greater buffer would be required for a set of flows with the same average rate but with a self-similar rate process. The significant dependence of the overflow probability on the structure of rate variations means that achievable utilization, given buffer size and link rate, can only be tightly controlled if this structure is known. Provisioning also needs to account for statistical variations in the traffic mix as new flows arrive and others complete. These complications are largely avoided with bufferless multiplexing.

The performance of bufferless multiplexing is extremely robust with respect to flow traffic characteristics. To illustrate this, assume again that flows have on-off rate variations with a common peak rate. In this case we adopt the Poisson session traffic model introduced in Claim 3. It matters little how flows are generated within sessions or how bursts are generated within flows: the distribution of the number of active bursts at an arbitrary instant is Poisson¹.

Let a be the offered demand in bit/sec. To dimension link capacity C to ensure a sufficiently small probability of rate overload, the only additional parameter required is the common flow peak rate p . The probability the data rate due to active sessions is greater than link capacity C is:

$$\Pr[\text{rate} > C] = \sum_{np > C} \frac{(a/p)^n \exp -a/p}{n!}. \quad (1)$$

While flows in reality have a variety of different peak rates, conservative provisioning can be performed on assuming they in fact all have a peak rate equal to some upper bound. Suppose this upper bound is p . We can consider a flow emitting a burst of 10 packets, say, at nominal rate $p' < p$ to be emitting 10 successive bursts at rate p with an appropriate inter-burst gap. The above fluid model of bufferless multiplexing still applies. The right hand side of (1) is just an upper bound in this case since we have ignored the information that the “bursts” at rate p are evenly spaced. It is possible to evaluate an equivalent exact expression for the probability of overload but this would be complicated and require knowledge of the precise mix of flow rates.

The maximum utilization compatible with a given overflow probability depends on the relative values of the peak rate and the link rate. In Fig. 3 we show how the utilization varies with link capacity for an overflow probability of 10^{-6} and three peak rates: 1, 5 and 10 Mbit/s. The results illustrate that a utilization of more than 60% can be attained when the peak rate to link rate ratio is less than 1/100. These are worst case results if the assumed peak rate is just an upper bound. They are valid for any traffic mix.

¹The system behaves like a stochastic network with two infinite server queues, one for active flows and one for think times [4, 5].

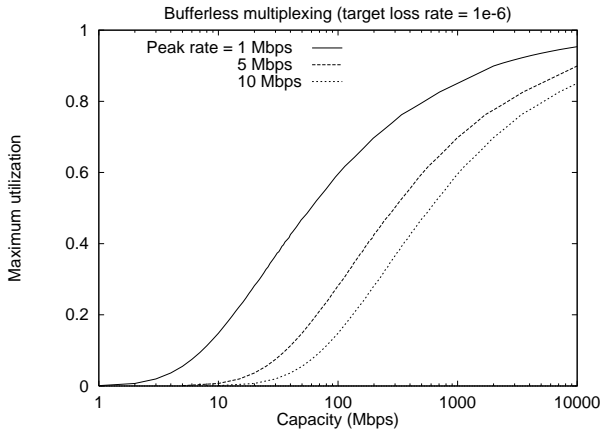


Figure 3: Maximum utilization with bufferless multiplexing

The bufferless multiplexer is, of course, only an abstraction. The fluid approximation does not allow an evaluation of packet delay. Results presented in [6] suggest that delay remains controlled in successive multiplexer queues as long as the flows are shaped to their nominal peak rate at the network ingress. It is consistent with the above performance models to assume flows are shaped to a common maximum peak rate, materializing the upper bound used to determine achievable utilization.

Claim 7 *Signal conservation and throughput conservation are compatible with high utilization in an integrated network*

We first consider the case of elastic traffic only. It was shown in [4] that under the assumptions of Poisson session traffic and fair sharing between ongoing flows, average throughput performance is independent of the detailed traffic characteristics of the different kinds of session included in the traffic mix. Moreover, throughput conservation can easily be ensured by limiting link utilization.

Suppose the transparency objective is that users experience an average throughput not less than 95% of their access rate (whenever this and the network are the only potential bottlenecks). Fig. 4 shows achievable utilization as a function of link capacity derived from the results in [4] when the access rate is 1, 5 and 10 Mbit/s. As for the corresponding results of Fig. 3, the relevant parameter for determining achievable utilization is the access (or “peak”) rate to link rate ratio. When this is less than 1/100, achievable utilization is more than 90%.

In a network with both streaming and elastic traffic, elastic flows are assumed to fairly share the bandwidth not used by streaming flows. The performance of such an integrated system is difficult to model analytically. However, simulation results presented in [7]

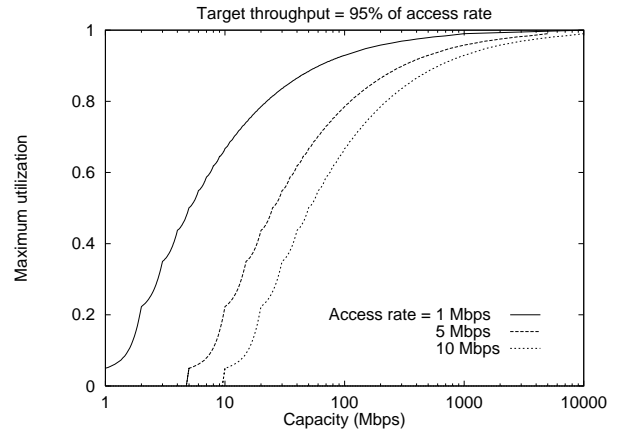


Figure 4: Maximum utilization of a fairly shared bottleneck with access rate limited flows

confirm that provisioning for transparency can be performed simply and is virtually independent of detailed traffic characteristics. It is necessary however to assume the peak rate of streaming flows is relatively small (1/100 th of the link rate, say) and the majority of traffic is elastic.

It is frequently assumed that streaming flows in an integrated system should be rate adaptive. In practice, in an adequately provisioned network, there is no need for adaptation as long as the flow rate is not greater than the assumed maximum peak rate. Signal conservation and throughput conservation are assured by adequate provisioning, by definition. Since streaming flows are assumed to be shaped to a maximum rate (cf. Claim 6), however, any application requiring a greater peak rate would indeed have to behave like an elastic flow.

The cited performance results rely on the assumption of perfectly fair sharing when the link is actually a bottleneck. Further results reported in [8] suggest, however, that conclusions are unaltered if sharing is somewhat unfair, due to the dependence of TCP throughput on round trip times, for example. Scope for unfairness is very limited when user access rates are much smaller than link capacities. Fig. 5 from [8] shows how a bias in sharing weights of 10 to 1 only has a significant impact at high utilization. Unfairness is only significant when utilization exceeds the limit for throughput conservation (given by Fig. 4 as 60% for a link rate ten times the access rate, as assumed in Fig. 5).

Claim 8 *There is little scope for service differentiation with respect to the guaranteed degree of transparency*

It is frequently supposed that a multiservice network can offer a range of transparency guarantees specifically adapted to a set of applications. A further assumption

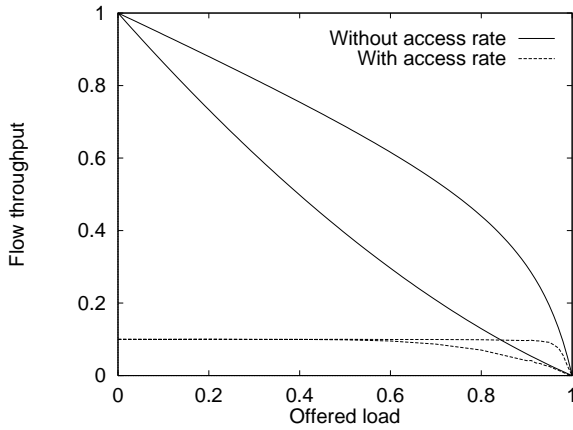


Figure 5: Flow throughput against load when flows of type 1 (top) receive 10 times more bandwidth than concurrent flows of type 2 (bottom)

is that to provide services with more relaxed guarantees is significantly less costly. These assumptions are misguided; they do not take proper account of the statistical nature of network traffic.

The discussion under Claim 6 shows that it is not possible to practically control queuing delays. The only feasible approach is to make delays as small as possible using bufferless multiplexing. With bufferless multiplexing, the packet loss rate is controllable and it is conceivable that transparency could be differentiated by applying a class dependent discard mechanism (such as WRED). However, for any reasonably large link, the advantage in increasing the objective loss rate for certain service classes is marginal. For example, the utilization of 60% derived from Fig. 3 for a peak to link rate ratio of 100 and an overflow probability of 10^{-6} only increases to around 70% for an overflow probability of 10^{-3} . In practice, when streaming flows constitute a minority of traffic on an integrated link, their packet loss rate is negligible for all classes and there is no scope at all for differentiation.

Fig. 5 illustrates the difficulty in realizing throughput differentiation for elastic flows by modulating their sharing weights (this could be realized approximately by differential packet discard or delay). As previously noted, adequate provisioning renders the network transparent for all service classes. There is no scope for service differentiation in normal load.

4 Controlling accessibility

We now consider how mechanisms in proposed QoS architectures might protect accessibility for different categories of users in the event of overload. We distinguish three broad approaches: specification of a traffic contract with nominal resource reservation for specific

flows or traffic aggregates, class of service differentiation and dynamic pricing.

Claim 9 *Without accessibility control, performance in overload is unacceptable*

The study of statistical bandwidth sharing by Ben Fredj *et al.* [4] includes a model of a link handling elastic traffic under overload i.e., flow arrival rate \times average flow size $>$ service rate. It is shown that the number of flows in progress keeps growing while realized throughput tends to zero. Performance degradation is eventually stabilized by the fact that certain flows are prematurely interrupted. However, link goodput is then low since resources are wasted on partially completed flows. Moreover, the throughput of flows which eventually do complete is barely acceptable.

In an integrated system, streaming flows are protected by their queue priority and quality degradation is suffered by elastic flows alone. It may be argued that adaptive audio and video coding would alleviate the situation. A reduction in the rate of a streaming flow indeed reduces overall demand (unlike a reduction in the rate of an elastic flow). However, the impact on performance is likely to be marginal unless adaptive streaming flows constitute a large proportion of overall traffic. We would also question the assumption that unpredictable degradation of signal quality would be acceptable for users of streaming applications.

Claim 10 *The notion of traffic contract is ill adapted to the statistical nature of traffic*

Most QoS architectures (including Frame Relay, ATM, Intserv, Diffserv, and the traffic engineering solutions of MPLS) incorporate the notion of traffic contract:

- the user declares a traffic descriptor, or *TSpec*, for a flow or traffic aggregate;
- the network performs admission control (to preserve transparency) and allocates resources;
- to prevent abuse, the network conditions traffic at the ingress or implements scheduling to isolate individual flows.

We successively consider how the traffic contract applies to an individual (long-lived) flow and to a broad traffic aggregate.

Application to individual flows The root of the problem in this case is that it is impossible to choose a *TSpec* for a variable rate flow which is, at the same time, useful for resource allocation, meaningful to the user and verifiable by the network. A frequently proposed *TSpec* consists of the peak rate and the parameters of a token bucket. This is verifiable (by design)

but is hardly useful for efficient resource allocation, as discussed below.

The $TSpec$ can be used by admission control to ensure flows experience a strict packet delay bound (e.g., delay $< T$, always) by applying the tools of network calculus [9, 10, 11]. Use of probabilistic bounds (e.g., delay $< T$ with high probability) leads to more efficient resource usage. These bounds can be derived as in [12, 13], for example, on assuming flows emit adversarial traffic (producing the longest delays possible given the $TSpec$) but are statistically independent.

Yet more efficient resource utilization could be attained if the network had more knowledge about the flow characteristics than just the $TSpec$. In particular, bufferless multiplexing, as discussed under Claim 6, can be performed if flow peak rate and mean rate are known.

Fig. 6 compares achievable utilization against link bandwidth for the above access controls in the case of a particular type of flow. We assume flows have on-off rate variations with exponentially distributed on and off periods. They have the following characteristics: peak rate, $p = 1.5$ Mbit/s, mean rate, $m = 50$ Kbit/s, mean activity period, $b = 3$ ms. The $TSpec$ specifies p and a token bucket with parameters $\sigma = 95$ Kbits and $\rho = 150$ Kbit/s. The chosen mean rate and burst length correspond to a non-conformance probability of 10^{-6} . The delay bound for worst case assumptions is $T = 50$ ms.

We have used standard network calculus to determine maximum utilization for the strict delay bound [10], and the local effective envelope method of Boorstyn *et al.* [12] for the adversarial traffic statistical bound assuming an overflow probability of 10^{-6} . The bufferless multiplexing utilization is calculated for a rate excess probability of 10^{-6} . These data are chosen to correspond to the example considered in Fig. 9 in [12].

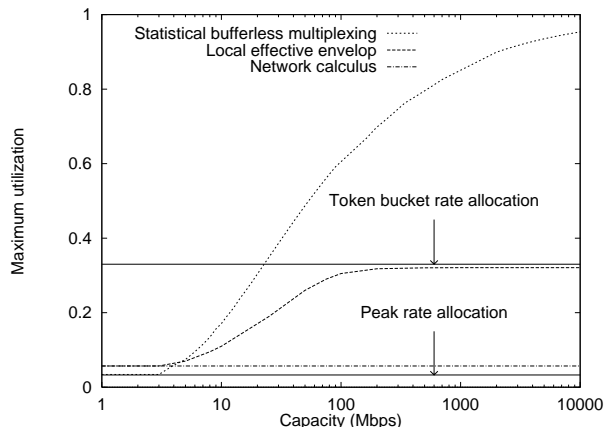


Figure 6: Achievable utilization depending on assumed multiplexing criterion

Fig. 6 demonstrates that admission control based on

adversarial traffic assumptions leads to considerable inefficiency compared to the utilization realizable with bufferless multiplexing (up to 3 times greater in this example). The degree of inefficiency clearly depends on the assumed traffic characteristics. It would have been greater had we chosen flows with more variable traffic (e.g., with heavy-tailed on or off periods).

Of course, the network cannot know the flow mean rate in advance. The superiority of bufferless multiplexing relies on the assumption that we can efficiently perform measurement-based admission control (MBAC) [14, 15]. This assumption is reasonable given the demonstrated insensitivity of bufferless multiplexing (i.e., performance depends only on overall load and the maximum peak rate). The decision theoretic MBAC proposed by Gibbens *et al.* [15] is particularly simple, requiring knowledge of just the flow peak rate and an estimation of current load.

A further disadvantage of basing accessibility control on the $TSpec$ is that we must account for the evolution of worst case traffic characteristics in successive multiplexing stages. To maintain a reasonably tight control generally imposes the use of per-flow scheduling. This constraint is avoided with bufferless multiplexing.

The requirement that users previously declare a $TSpec$ is a significant constraint to impose on customers and a needless one since bufferless multiplexing with MBAC is more efficient. One might also ask why a network provider would advertise a delay bound of 50 ms say, when the actual delays are very much smaller (zero in the fluid approximation), as in the example of Fig. 6.

Application to aggregates In most existing data networks and proposed QoS architectures, traffic contracts apply not to individual flows but to aggregates. The aggregate in question might, for instance, be all the traffic between two sites in a virtual private network (VPN). In this case, the $TSpec$ is not so much a traffic descriptor as a bound on the data rate the customer is prepared to pay for. Traffic contracts for aggregates generally do not impose strict traffic conditioning but just mark excess packets as liable for discard in case of congestion.

Experience shows that users frequently overestimate their actual bandwidth requirement by a large margin. The net result is that reserved bandwidth for a group of such traffic contracts sharing a given link is generally many times greater than the actual amount of traffic to be handled. To improve utilization, the network operator has two options:

- reserve resources in accordance with the $TSpec$ but allow lower priority best effort traffic to occupy the unused capacity,
- overbook link capacity by admitting several times more aggregates than can strictly be accommodated.

The first solution is valid only if there is a sufficient amount of BE traffic. It constitutes a form of service differentiation and is discussed under the next claim. The second appears to contradict the notion of controlled accessibility since there is no protection against possible overload. It is significant, however, that overbooking is currently employed, notably by operators of Frame Relay networks.

Overbooking, by a factor of 5 or 10, say², is generally possible without detrimental impact. This is possible both because contracts are “over-provisioned” and because traffic is mainly elastic and adjusts to the available rate. This should not be construed as proof of success for the traffic contract, however. It is evidence rather that there is little correlation between the *TSpec* and the resources actually consumed. It is, of course, always possible that the overbooking booking will fail if the applied factor is too high.

Accessibility control is yet more uncertain when it is not possible to precisely locate the network paths to be used by aggregates. This occurs, for instance, with the “hose model” for VPNs [16] or with Diffserv. The *TSpec* has to be interpreted here as an “expected capacity profile” providing a rough guide for admission control and provisioning [17]. It remains to be demonstrated, however, that this information can actually be used to improve on the usual provisioning methods of a best effort network.

Claim 11 *Class of service differentiation provides crude, inefficient accessibility control*

We argued under Claim 8 that there is limited scope for service differentiation in normal load conditions. In overload, QoS architectures usually protect the accessibility of certain classes of service but do nothing to prevent the phenomenon of resource wastage identified under Claim 9.

Per-flow reservation, within the terms of a traffic contract and implemented using an appropriate scheduling mechanism, does preserve the QoS of the flows in question. However, the flows composing the best effort traffic sharing the residual capacity are denied accessibility. This residual capacity is then largely wasted. Similarly, an architecture using class-based WFQ to share bandwidth between a small number of classes can be configured to protect the higher priority flows. It is sufficient to ensure that their guaranteed service rate is adequately provisioned. Inevitably, however, the lower priority classes do not have access to sufficient resources and suffer from the effects of overload described previously.

Some service differentiation proposals would discriminate between classes via the packet loss rate (e.g., using WRED) or the mean packet delay [18]. The consequence of this discrimination in overload is unclear.

²The actual factor is usually a closely guarded secret!

At best, some classes will avoid saturation with the impact of overload being concentrated on the lower priority classes. At worst, all classes suffer from the overload but to varying degrees: premium class is bad but regular is even worse.

Quality of service differentiation necessarily implies price differentiation since users would otherwise always choose the class with highest quality. Conversely, a wish to exploit the economic efficiency of price differentiation may be identified as a major motivation for the development of the Diffserv architecture. In the next claim we defend the contrary point of view, that congestion pricing is not a practical accessibility control for the commercial Internet.

Claim 12 *Congestion pricing is not a practical accessibility control.*

The primary function of pricing in a commercial network is return on investment. Infrastructure and operating costs must be shared between different users and the pricing structure determines the contribution of each. Prices may depend on a large number of criteria including connection speed, accessibility guarantees and actual usage, and the optimal structure is a complex trade-off between sometimes conflicting requirements for economic efficiency and transparency. Experience in the commercial Internet and similar service industries shows that customers have a very strong preference for simplicity and risk avoidance [19]. It is unlikely on these grounds alone that they would ever accept the unpredictability of congestion pricing.

Congestion pricing consists in managing use of a scarce resource by allowing potential users to declare the utility they would gain and then limiting access in order to maximize overall utility. Schemes like the “smart market” of MacKie-Mason and Varian [20], the pragmatic class-based scheme of Shenker *et al.* [21], or Kelly’s self-managed network [22] are provably optimal in maximizing overall welfare. However, they ignore return on investment, considering the network infrastructure and operation as a sunk cost. In the absence of congestion in an adequately provisioned network, no revenue is generated.

In practice, network resources are not scarce. The provider can easily upgrade capacity and will do so before congestion occurs if return on investment is assured. Congestion may then be interpreted by users as a sign of bad management. To expect them to then pay a supplement (given that they are already paying for the network) is like adding insult to injury.

5 A flow-aware QoS paradigm

The ideal QoS architecture would perform overload control to eliminate excess traffic while maintaining high

utilization and ensuring that all admitted traffic has adequate transparency. The network would also provide differentiated accessibility to allow preferential treatment for certain users or service classes. In the present section we suggest that this architecture must be flow-aware, performing admission control and routing traffic flow by flow.

Claim 13 *Flow level measurement-based admission control is the key to QoS control*

The flow, as defined in Claim 3, constitutes the appropriate level of granularity for traffic control. It is the closest identifiable object which can be assimilated to a communication service provided by the network. Admission control allows the network to protect the quality of service of ongoing flows by refusing new demands when necessary.

In Section 2, we gave rather loose definitions of flow and session. These were sufficient for traffic modeling but to implement flow-level admission control it is necessary to be more precise. One possibility with considerable flexibility would be for the user to set a flow ID field in the IP header (as envisaged in IPv6) with the flow being identified by the association of this field with either the source address, the destination address or both. Two bits of the flow ID could be used to specify which of the IP addresses are relevant, as appropriate for a given application. We refer to flows so identified as “appliflows”.

For example, a Web page containing elements retrieved from more than one server, could constitute a single appliflow if packets in all contributing microflows contained a common user-supplied flow ID to be associated with the destination IP address. Alternatively, a game server simultaneously sending packets to multiple players might define a common flow by associating flow ID and originating IP address. The objective is to avoid admission control decisions resulting in the corruption of entities which are significant for a considered application.

Consider a link handling streaming and elastic traffic using priority queuing as envisaged in Section 3 and assume users identify their appliflows as either streaming or elastic. Admissibility conditions must be such that transparency would be preserved if an appliflow of either type were admitted. A flow would be admitted only if the current load were less than one threshold, determined as in [15] to ensure signal conservation, and the available bandwidth³ were greater than another threshold, determined as in [23] to ensure throughput conservation.

Models of statistical bandwidth sharing show that the choice of available bandwidth threshold is not highly critical for performance. It is relatively easy to set a

³The bandwidth a new flow would attain assuming fair sharing.

value such that virtually no flows are rejected in normal load while throughput conservation is assured in overload [23]. It is also straightforward to realize differentiated admissibility by applying different thresholds: regular flows are blocked at lower levels of occupation than premium flows, effectively preserving the accessibility of the latter⁴. For a wide range of possible thresholds, blocking is negligible in normal load while nearly 100% utilization is maintained in overload (see [24]).

Traffic conditioning and/or scheduling are necessary to prevent abuse. We previously noted the need to shape streaming flows to a maximum acceptable peak rate. To prevent unfair sharing for elastic flows (by users having an access rate greater than the available bandwidth threshold), it may be considered necessary to implement per-flow fair queuing.

Claim 14 *Flow-aware networking is feasible*

To propose flow-aware networking immediately raises concerns of complexity and scalability. The implementation we have in mind limits such problems by avoiding the need for signaling and requiring minimal per-flow state [23].

A newly arriving flow at a given router interface can be recognized as such “on the fly” without explicit signaling. The appliflow ID of every packet would be compared to a list of flows in progress. If the flow exists the packet is forwarded; if not, the admission test is applied. If the flow can be admitted, its ID is added to the list; if not, the packet is simply discarded. The loss of this first packet would be interpreted by the user’s application as flow rejection, rather like the loss of a probe in an endpoint MBAC [25] or the loss of the SYN or SYN-ACK packet of a TCP connection [26]. Retrials are at the user’s discretion.

Maintenance of the list of flows in progress appears as the most complex task. Consultation of the table is necessary for every packet and must be performed as rapidly as a route look-up. However, this is not an unsurmountable problem and our research to date suggests practical solutions can be found even for Gbit/s interfaces. Flows must be erased after a suitably chosen time-out following the last packet arrival.

Per-flow fair queuing is feasible even on very high speed interfaces [27]. It is important to note that the scheduling algorithm operates only on flows currently having a backlog. The number of such flows is orders of magnitude smaller than the number of flows in progress as determined by the list used for admission control. Moreover, the application of admission control ensures that this number remains bounded even in situations of overload.

⁴This is the principle of trunk reservation used in circuit switched networks.

Implementation would require a small standardization effort, necessary essentially just to specify the appliflow ID convention. Implementation could also be incremental with admission control equipped links contributing to a progressive improvement in network performance.

Pricing considerations are much simpler than for current QoS architectures. Usage based charging can be based on simple byte counting since all packets except discarded probes correspond to flows with assured transparency. Pricing differences between users would more naturally reflect the different admissibility guarantees provided by applying selective admission control thresholds.

Claim 15 *Flow aware networking allows efficient traffic dependent routing*

When a link is saturated, rather than rejecting a newly arriving flow, it would be better to seek a more lightly loaded alternative path. This kind of adaptive routing leads to a more efficient use of installed capacity and improves accessibility from the user perspective. Adaptive routing on a per-flow basis can be envisaged using the admission control mechanisms described above. The routing decision associated with a flow needs to be memorized in the list of active flows to ensure all packets are forwarded over the same path.

The choice of route would typically depend on its length in hops and the current load of its component links. A number of possible per-flow routing algorithms using both metrics have been evaluated for elastic flows in [28]. The algorithms select a feasible path from a set of predefined options, materialized as LSPs in an MPLS capable domain, for example. Admission thresholds would typically be more severe for longer paths in order to preserve efficiency in overload: to use a longer route increases congestion, possibly obliging subsequent flows to use longer routes which further increases congestion...

Adaptive routing at flow level would not suffer from the problems of instability observed when routing is based on shifting traffic aggregates depending on the value of periodically updated accessibility metrics. Unlike a traffic aggregate, to add an additional flow has a marginal and predictable impact on QoS. Link load and available bandwidth are measured continuously avoiding the phenomenon of route thrashing arising when routing is based on outdated information.

6 Conclusions

QoS is mainly a question of dealing effectively with overloads arising either from planning oversight or capacity reduction due to equipment failure. Bufferless multiplexing and fair sharing for streaming and elastic traffic, respectively, form the basis for robust traffic

management with the notion of adequate provisioning being independent of precise traffic characteristics. In overload, however, in the absence of accessibility control, network transparency is not assured.

To rely on over-provisioning to avoid overloads appears unreasonably costly. The mechanisms of Intserv and Diffserv can protect the transparency of privileged high priority service classes but do nothing to avoid the general quality degradation of “best effort” traffic.

To preserve network efficiency we must deal separately with transparency and accessibility requirements implying the use of admission control. The appropriate entity to which admission control should be applied is a user defined flow (or “appliflow”). Implementation is possible using lightweight procedures which avoid the familiar problems of scalability. Admission control also allows effective accessibility differentiation by applying progressive admission thresholds to different service classes. The same admission control mechanisms can be used to perform stable and effective traffic dependent adaptive routing.

The above conclusions derive from an analysis of the relationship between realized performance, available capacity and the characteristics of handled traffic. Understanding this relationship is vital to evaluating the scope for traffic management and for defining what is actually meant by adequate provisioning.

Flow-aware admission control appears as an essential function for the commercial Internet. The implementation we have sketched would retain much of the ubiquity and simplicity of the best effort architecture while ensuring cost-effective service protection in overload. It remains for vendors to appreciate the importance of this requirement and to meet the challenge of implementing the flow-aware QoS paradigm in their routers.

References

- [1] W.E. Leland, M.S. Taqqu, W. Willinger, and D.V. Wilson. On the self-similar nature of ethernet traffic. *IEEE/ACM Transactions on Networking*, 2(1):1–15, 1994.
- [2] S. Floyd and V. Paxson. Difficulties in Simulating the Internet. *IEEE/ACM Transactions on Networking*, 9(4):392–403, August 2001.
- [3] A. Bouch, M. A. Sasse, and H. DeMeer. Of packets and people: A user-centered approach to quality of service. In *IWQoS'00*, June 2000.
- [4] S. Ben Fredj, T. Bonald, A. Proutière, G. Régnié, and J.W. Roberts. Statistical bandwidth sharing: A study of congestion at flow level. In *ACM SIGCOMM*, pages 111–122, 2001.

- [5] F. P. Kelly. *Reversibility in Stochastic Networks*. Wiley, 1979.
- [6] T. Bonald, A. Proutière, and J.W. Roberts. Statistical Performance Guarantees for Streaming Flows using Expedited Forwarding. In *IEEE INFOCOM*, pages 1104–1112, 2001.
- [7] F. Delcoigne, A. Proutière, and G. Régnié. Modelling integration of elastic and streaming data. Preprint, 2002.
- [8] T. Bonald and L. Massoulié. Impact of Fairness on Internet Performance. In *SIGMETRICS Performance Evaluation Review*, pages 82–91, June 2001.
- [9] R. L. Cruz. A calculus for network delay. *IEEE Transactions on Information Theory*, January 1991.
- [10] J. Y. Le Boudec and P. Thiran. *Network Calculus*. Springer Verlag LNCS 2050, June 2001.
- [11] C-S. Chang. *Performance guarantees in communication networks*. Springer-Verlag, New York, 2000.
- [12] R. R. Boorstyn, A. Burchard, J. Liebeherr, and C. Ottamakorn. Statistical Service Assurance for Traffic Scheduling Algorithms. *JSAC*, 18(12):2651–2664, December 2000.
- [13] C-S Chang, W Song, and Y. Chiu. On the performance of multiplexing independent regulated inputs. In *ACM SIGMETRICS*, June 2001.
- [14] Lee Breslau, Sugih Jamin, and Scott Shenker. Comments on the Performance of Measurement-Based Admission Control Algorithms. In *IEEE INFOCOM 2000*, pages 1233–1242, March 2000. Tel Aviv, Israel.
- [15] R.J. Gibbens, F.P. Kelly, and P.B. Key. A Decision-Theoretic Approach to Call Admission Control in ATM Networks. *IEEE Journal on Selected Areas in Communications*, 13(6):1101–1114, August 1995.
- [16] N.G. Duffield, P. Goyal, A. Greenberg, P. Mishra, K.K. Ramakrishnan, and J. E. van der Merwe. A flexible model for resource management in virtual private networks. In *ACM SIGCOMM Computer Communication Review*, October 1999.
- [17] D. Clark and W. Fang. Explicit Allocation of Best-Effort Service. *IEEE/ACM Transactions on Networking*, 6(4), August 1998.
- [18] C. Dovrolis, D. Stiliadis, and P. Ramanathan. Proportional differentiated services: Delay differentiation and packet scheduling. In *ACM SIGCOMM*, October 1999.
- [19] A. M. Odlyzko. Internet pricing and the history of communications. *Computer Networks*, 36:493–517, 2001.
- [20] J. K. MacKie-Mason and H. Varian. *Pricing the Internet*, chapter Public Access to the Internet. Prentice-Hall, Englewood Cliffs, New Jersey, 1995.
- [21] S. Shenker, D. D. Clark, D. Estrin, and S. Herzog. Pricing in Computer Networks: Reshaping the Research Agenda. *ACM Computer Communication Review*, 26:19–43, April 1996.
- [22] F. P. Kelly. Models for a self-managed internet. *Philosophical Transactions of the Royal Society*, A358:2335–2348, 2000.
- [23] N. Benameur, S. Ben Fredj, F. Delcoigne, S. Oueslati-Boulahia, and J.W. Roberts. Integrated admission control for streaming and elastic traffic. In *QofIS*, pages 69–81. Second COST 263 International Workshop, Coimbra, Portugal, Springer, September 2001.
- [24] S. Ben Fredj, S. Oueslati-Boulahia, and J.W. Roberts. Measurement-based Admission Control for Elastic Traffic. In J. Moreira de Souza, N. L.S. da Fonseca, and E.A. de Souza e Silva, editors, *Teletraffic Engineering in the Internet Era*, pages 161–172. ITC 17, Elsevier, December 2001.
- [25] L. Breslau, E. W. Knightly, S. Shenker, I. Stoica, and H. Zhang. Endpoint Admission Control: Architectural Issues and Performance. In *ACM SIGCOMM*, pages 57–69, October 2000. Stockholm, Sweden.
- [26] R. Mortier, I. Pratt, C. Clark, and S. Crosby. Implicit Admission Control. *IEEE Journal on Selected Areas in Communications*, December 2000.
- [27] B. Suter, T.V. Lakshman, D. Stiliadis, and A.K. Choudhury. Buffer management schemes for supporting tcp in gigabit routers with per-flow queueing. *IEEE Journals in Selected Areas in Communications*, 17(6):1159–1170, August 1999.
- [28] S. Oueslati-Boulahia and J.W. Roberts. Impact of “trunk reservation” on elastic flow routing. In G. Pujolle, editor, *Lecture Notes in Computer Science*, volume 1815, pages 823–834. Networking 2000, Paris, France, Springer, May 2000.

Glossary

ATM	Asynchronous Transfer Mode
LSP	Label switched path
MBAC	Measurement-based admission control
MPLS	Multi-protocol label switching
VPN	Virtual private network
WRED	Weighted random early discard