A new direction for quality of service: Flow-aware networking

S. Oueslati and J. Roberts
France Telecom R&D

Abstract— **In this paper we present the architectural implications of Flow-aware networking, or FAN, a new approach for realizing QoS guarantees. FAN performs traffic control based on user-defined flows using implicit admission control and per-flow scheduling. This allows adequate performance guarantees for streaming and elastic flows without requiring class of service distinction or relying on signalled traffic specifications. We argue that FAN is a new direction for Internet quality of service that is arguably necessary as an alternative to flawed classical architectures and sufficient to meet user performance requirements in a cost-effective way.**

I. Introduction

The way user quality of service requirements are satisfied has far-reaching implications on network architecture, notably through the need to introduce a number of specific mechanisms such as reservation protocols, packet schedulers, policy servers and bandwidth brokers. It is important, therefore, to conduct a careful cost-benefit analysis before proceeding to any major change from the current best effort Internet.

Analysis of the traffic-performance relation, linking capacity, demand and performance, for a range of streaming and elastic traffic types leads us to believe adequate performance can be assured much more simply than in the classical QoS architectures and more reliably than in an over-provisioned best effort network. The basis for this belief is our proposition for an architecture called *Flow-aware networking*, or FAN for short.

FAN performs traffic management based on user-defined flows using two per-flow mechanisms to control sharing of link bandwidth, namely implicit admission control and priority fair queueing. The scheduler realizes max-min fair sharing while ensuring negligible latency for packets on flows emitting at a rate less than the current fair rate. Admission control is used to keep the fair rate sufficiently high to enable a minimal level of performance for a chosen range of streaming and elastic flows. Admission control also has the fundamental role of maintaining performance in overload.

Overload occurs when expected demand, equal to the product of flow arrival rate and average flow size, exceeds link capacity. In this case, in the absence of admission control, the number of flows in progress increases leading to progressive performance degradation. This phenomenon is often ignored in analyses of QoS architectures where the number of flows in progress is assumed to be constant.

In FAN, admission control and service differentiation are implicit in that there is no class of service distinction and no need for *a priori* traffic specification. The user-network interface remains that of the best effort Internet and flexibility for end-to-end service creation is enhanced. Improved traffic control also enables more cost-effective capacity planning.

The present paper examines the architectural implications of the FAN mechanisms. After a rapid presentation of its main components in the next section, we argue in Section III that FAN conforms to the original Internet design philosophy, notably with respect to its goals of survivability and multiservice support. In Section IV we explain the misgivings about current proposals for a QoS capable architecture that lead us to believe a new direction is necessary. We discuss prospects for realizing FAN in Section V before presenting our concluding remarks.

II. What is Flow-aware networking

We first introduced a certain notion of flow-aware networking in [27]. Since then we have further elaborated on this proposal, notably by developing and evaluating mechanisms for implicit per-flow admission control [2], [4]. The most recent development consists in associating admission control with per-flow scheduling in a so-called *Cross-protect* router [21]. Parallel developments demonstrate the advancing maturity of other flow-based techniques [12], [23]. However, in this paper we reserve the term flow-aware networking, or FAN, for the architecture described below.

A. User defined flows

By flow we mean a flight of datagrams, localized in time and space and having the same unique identifier. It is localized in time by the fact that packets are spaced by no more than a certain interval (a few seconds). It is localized in space in that the packets in question are observed at a particular interface. A typical flow thus has multiple instances, one at each interface on its path.

The identifier is deduced from header fields including IP addresses and a user specified field like the IPv6 flow label. The expectation is that users define flows to correspond to a particular instance of some application such as a video stream or a document transfer. The flow label may also be deduced from IPv4 header fields such as the usual 5-tuple of IP addresses, protocol and port numbers. However this limits flexibility and runs into difficulties when flow identification data are hidden within tunnels.

One way of using the IPv6 flow label would be to reserve 2 bits to specify whether the label should be associated with the origin IP address, the destination IP address, both addresses or none, offering a range of possibilities appropriate for different applications [8], [24]. The intention is to allow the user as much flexibility as possible in defining what the network should consider as a flow.

B. *Per-flow implicit admission control*

Admission control is necessary to preserve application performance and network efficiency in situations of overload. The user-defined flow discussed above appears as the most appropriate entity on which to perform admission control. It is closely related to the notion of application instance while having a definition that lends itself to an efficient implicit implementation. By 'implicit' we mean an admission control that does not rely on explicit user-network signalling. Admission control is local to a particular network link.

Flows that have been admitted on a link and are currently in progress are registered in a *protected flow list*. If the flow identity of a newly arriving packet is already in this list, the packet is forwarded. If not, the flow is subject to admission control. If the link is congested, as determined by real-time measurements supplied by the scheduler described next, the packet is discarded. The discard of the first packet, or packets, of a flow is the *implicit* signal to the user that the link is congested. In the absence of congestion, the packet is forwarded and the flow added to the protected flow list.

The protected flow list is soft state with entries updated on the arrival of a packet from the corresponding flow. The state expires when no packet is observed in the interval used to define the flow in terms of time locality. Flows are 'protected' in that minimal performance requirements are satisfied (see Section II-D below).

Admission control is intended as a kind of insurance against the worst effects of exceptional events like failures: the performance of admitted flows is satisfactory even in overload. In the absence of admission control, sustained overload leads to unacceptable performance degradation for both elastic and streaming flows [7].

Note that all flows are subject to rejection in congestion without regard to their particular traffic characteristics. This avoids the (troublesome) requirement to signal the latter and ensures similar blocking probabilities for all types of flow.

C. *Per-flow scheduling*

Per–flow scheduling allows controlled performance independently of possible user misbehaviour. The scheduler used in FAN implements Priority Fair Queueing (PFQ) [21]. PFQ performs per-flow fair queueing using an efficient algorithm like Start-time Fair Queueing [15] with an original added twist: packets of flows whose arrival rate is less than the current fair rate are given head of line priority in the scheduler queue.

This enhancement ensures packets of streaming flows of low enough peak rate receive priority treatment. The delay jitter they acquire is therefore *negligible* in the precise sense defined in [9]. Of course, if packets arrive in bursts due to previously acquired jitter, their original spacing will tend to be restored by the scheduler.

The PFQ scheduler protects flows against user misbehaviour by enforcing fairness and provides *implicit* service differentiation. Service differentiation is between 'under' and 'over' flows (i.e., under and over the current fair rate) rather than between streaming and elastic flows. Admission control thresholds need to be set to ensure that flows of a targeted range of streaming applications are always in the 'under' category.

An obvious concern with per-flow scheduling is that of scalability. This is avoided in FAN by the concurrent use of admission control. The only flows that need be known to the scheduler are those that currently have a packet in the queue. Admission control ensures this number is bounded with high probability independently of the link rate (see Section V-A).

The scheduler algorithm naturally provides the measurements, of fair rate and load due to priority traffic, needed for admission control. It is the synergy of admission control and PFQ scheduling that leads us to call their association *Cross-protect*. This term is also an appropriate allusion to the service protection afforded to individual streaming and elastic flows.

D. *'Good enough' performance*

The performance guarantees of FAN are probabilistic. A streaming flow of sufficiently low peak rate is assured low packet loss and delay. Loss probabilities and delay quantiles, as well as the maximum peak rate for which streaming quality is assured, can be controlled by appropriately defining admission control criteria. Elastic flows whose rate is not limited elsewhere are assured throughput no less than a certain threshold. This threshold is again controllable by the choice of admission criteria.

For the same admission control criteria[1], the peak rate up to which streaming quality is assured is somewhat less than the limiting elastic flow throughput. This is due to unavoidable imprecision in scheduling packets of flows whose input rate is just less than the current fair rate.

Note that the limits on performance only apply in conditions of overload when admission control is necessary. In most cases, the fair rate is much higher than the limiting throughput threshold. Streaming flows with rate greater than the limiting peak rate then have negligible packet loss and delay. However, such flows may need to adapt their rate to preserve application quality in case of congestion.

The precise choice of performance guarantees, determined by the applied admission control criteria, is dictated by economic considerations. For given average demand and some low target blocking probability, it requires more capacity to support higher streaming flow peak rates or to assure higher elastic flow throughput. Reasonable efficiency is attained when the peak rate and elastic throughput threshold are around 1% of link capacity [2].

III. FAN AND THE INTERNET DESIGN PHILOSOPHY

We now discuss what FAN brings to the Internet architecture basing the discussion on aspects of the original design philosophy of the Internet outlined by Clark in 1988 [13]. We first discuss how the introduction of the *Cross-protect* mechanisms conforms to the end-to-end principle [28].

A. *The end-to-end principle*

The objective in FAN is to maintain the simple user network interface of the current best effort Internet while enabling a range of services that depend on performance guarantees. This requires placing admission control and scheduling mechanisms within the network. We argue below that the functions they realize cannot reasonably be placed outside the network and that, consequently, FAN respects the end-to-end principle.

It has been suggested by several authors that admission control can be performed at the edge (see [11] for a survey). However, an unresolved difficulty with such schemes is their reliance on user cooperation in reacting to admission refusal. The present proposition avoids this difficulty since the network elements manage their own lists of protected flows. FAN nevertheless relies on end user participation. Applications must emit probe packets to test resource availability. The specification of the probing phase and the reaction to failure are left to the designers of each application.

Per-flow scheduling imposes bandwidth sharing fairness. Fairness is currently realized end-to-end by TCP congestion avoidance. Although this generally works satisfactorily, it is a recurring concern that reliance on user cooperation renders service quality vulnerable to misbehaviour. Associating a price with ECN marks is an alternative means of ensuring appropriate user behaviour [19]. However, the network mechanisms required to control such a pricing scheme are arguably at least as complex as the current proposition.

Note that proper bandwidth sharing in FAN still relies on end-to-end protocols like TCP that are capable of finding the correct transmission rate. The association of admission control and scheduling may be viewed as a sophisticated active queue management scheme. The *Cross-protect* mechanisms could, for example, incorporate the AQM advocated by Suter et al. [29] where, if necessary, packets are dropped from the head of the longest per-flow queue.

B. *Survivability in the face of failure*

The first goal (after that of interconnecting networks) identified in [13] is that communications continue despite loss of networks or gateways. FAN inherits and enhances the survivability characteristics of the current Internet.

Flow-level admission control improves survivability by ensuring efficient use of reduced connectivity bandwidth in the event of failure. Even in a well-planned network, offered traffic may exceed available link capacity in case of failure. While such overloads last, per-flow throughput decreases as the arrival rate of new flows is greater than the rate at which flows complete. Admission control proactively rejects some new flows in order to preserve the performance of flows in progress. It is important to realize that, in sustained overload, this is preferable to allowing throughput to deteriorate. Ultimately, the same amount of traffic is transmitted since the link is always saturated. Admission control

[1] Recall that we block fbws in the same congestion conditions irrespective of their particular traffic characteristics.

simply ensures that this throughput is associated with satisfactory per-flow performance and thus preserves ‘goodput’.

Admission control mechanisms can be used to perform *selective blocking*. If some kinds of flow are rejected at the early onset of congestion, capacity is reserved for premium traffic. Discrimination might depend on the type of flow (e.g., an ‘emergency call’), the user identity (e.g., users paying for high availability) or the path used by the flow (e.g., prefer flows on shorter paths to preserve routing efficiency).

Flow rejection may be tolerated more easily if the user may retry using an alternative path. A simple realization of per-flow adaptive routing would consist in performing load balancing over multiple possible routes by using a hash function of the flow identifier to choose the route to be tested. On suffering a packet discard, the application could test an alternative path just by changing the flow label and making a reattempt.

The current resilience of the Internet is maintained since the *Cross-protect* mechanisms are essentially local to the equipment on which they are implemented and rely on soft-state. In the event of link failure, rerouted packets will appear as the first packets of new flows on their new route. Some may therefore be interrupted if the new link load results in congestion.

C. Types of service

Two main categories of service are evoked in [13]: services requiring reliable data transfer without undue concern for packet delay and throughput, and services like real-time delivery of digitized speech where delay is the essence and reliability can be sacrificed if necessary. *Cross-protect* allows an implicit distinction between these types of service as explained in Section II.

Within each category, one could list a large number of possible service classes distinguished by their specific performance requirements. However, the results of research on the performance of statistical multiplexing (e.g., [26], [9]) and statistical bandwidth sharing (e.g., [3], [6], [10], [25]) lead us to believe the guarantees provided by Cross-protect are sufficient. They may also constitute the best compromise that can be achieved between cost of implementation and realized quality.

Cross-protect effectively realizes bufferless statistical multiplexing for flows whose rate is less than the fair rate. Adequate performance is thus always assured for services emitting flows at a peak rate less than the lower limit on fair rate maintained by admission control. This applies for audio and most video services as well as control flows and a large range of gaming applications. Notice that there is no unnecessary limitation on flow rates and streaming quality is possible for very high peak rate flows in periods of light traffic.

FAN approximately realizes max-min fair bandwidth sharing [22], [17] . Arguably, this is sufficient for elastic flows. There is little or no advantage in introducing discriminatory sharing or size-dependent scheduling [25]. This is mainly because admission control can be calibrated so that the fair rate is always higher than the maximum possible rate of the vast majority of flows (this being limited by the speed of their access link, for instance).

FAN is compatible with the multitude of services and applications that currently use the best effort Internet and opens the possibility to develop new applications. In particular, given the guaranteed timely delivery of packets of low rate flows, it would become possible to develop (commoditized) telephone services at the edge without requiring specific network support.

FAN does not allow deterministic QoS guarantees. It is not possible either to create virtual leased lines. These services would require additional mechanisms.

D. Vulnerability to attack

Any innovation requires careful analysis of the potential for new denial of service attacks. We believe FAN does not introduce significant new DoS opportunities. Two types of behaviour can be envisaged:

- a user changes the identifier with every packet: this could rapidly saturate the protected flow list leading to the non-protection of certain flows; however, the latter would only suffer if the link in question was suffering congestion at the same time;
- a user maintains the same identifier for several flows: successive flows will not be subject to admission control as long as the inter-flow interval is less than the time out used to determine the end of a flow; flows emitted in parallel will be considered as one by the PFQ scheduler and will not attain a rate greater than the current fair rate; in both cases, the annoyance to other users appears very slight or nonexistent.

A user could, as now, establish several flows to transport the packets of a single application and thus gain higher throughput. This advantage would however only accrue to the small fraction of users having an access rate greater than the current fair rate which is necessarily greater than a chosen threshold. Note that this behaviour is possible also with the current network and with proposed QoS architectures like Diffserv.

The fact that FAN deals with flows and not just datagrams could open new possibilities for dealing with attacks. The same mechanisms used to identify flows for admission control and scheduling could be used to identify anomalous behaviour and act to mitigate the impact of an attack.

E. TCP

The fact that FAN realizes max-min fair sharing independently of user behaviour allows considerable flexibility in the choice of transport protocol for elastic flows. In particular, it is perfectly possible for different versions of TCP to co-exist. New versions adapted to high-speed transfers can be introduced with no adverse effect on other users.

The slow-start algorithm is currently a source of inefficiency for many flows. A more aggressive starting behaviour would be perfectly acceptable with FAN since flows are protected by fair queueing. One possibility would be for users to deduce the current path rate using a pair of back-to-back packets, as proposed by Keshav [20]. This is feasible when the bottleneck router implements fair queueing.

F. Other goals

Two lower priority goals mentioned in [13] are cost effectiveness and accountability. FAN would enable greater cost effectiveness by allowing more efficient capacity planning. This derives from the insurance provided by admission control: quality and efficiency are assured even in the event of overload. It is no longer necessary to practice extreme degrees of over-provisioning since the impact of failure is contained. For example, selective blocking makes it possible to limit the degree of over-provisioning to that required to guarantee the availability requirements of just the most exigent users and services.

Accountability, notably to attribute resource usage for billing, is simpler with FAN than with alternative QoS architectures. There are no complicated SLAs. All packets are equal with respect to accounting since they are classless. All packets (save exceptional discards) contribute to providing a useful service and are liable to charging.

IV. Why we need a new architecture

FAN is the result of long reflection on how to ensure adequate performance for a variety of services in an integrated (or converged) network. We have been led to define this architecture on identifying a number of serious deficiencies in alternative propositions. In the present section, we recall our main criticisms of alternative QoS architectures but refer to previous publications for more ample justification [8], [24].

A. The over-provisioned best effort network

Over-provisioning generally does ensure adequate performance, especially in the network core. It is not necessarily uneconomical but the common argument that over-provisioning is necessary anyway for reasons of reliability and therefore comes for free is largely spurious. It is clearly not necessary to safeguard all traffic in all situations of failure. In fact, the required *degree* of over-provisioning is usually left unspecified by advocates of this approach.

In an over-provisioned network there is, by definition, no requirement for admission control and, on high speed links, no need for per-flow scheduling. FAN also requires a sufficient degree of over-provisioning to ensure a negligible probability of blocking. Non-FIFO scheduling is then unnecessary for the large majority of flows, at least on backbone links. The main reason for introducing FAN is to provide insurance that performance is controlled even in situations of failure and overload. This insurance allows the degree of over-provisioning to be more closely tailored to requirements.

It is possible to evaluate blocking probabilities and therefore perform capacity planning to ensure optimal performance, accounting as necessary for envisaged failure scenarios. For example, the use of selective blocking can provide ‘five-nines’ availability for priority traffic with just a slight degree of over-provisioning.

B. QoS and the traffic contract

Most QoS architectures, including Intserv, are based on the notion of ‘traffic contract’: users specify traffic and performance parameters for their flow; the network performs admission control on the basis of these parameters; if accepted, the flow is policed or scheduled using the specified traffic parameters. The flow in question might be for a single application instance or a more broadly defined traffic aggregate.

A major difficulty is the choice of an adequate traffic specification. Note first that to require users to make such a specification is a significant obligation that is both difficult to fulfil and arguably unnecessary. Reliance on rule-based parameters, chosen more for ease of policing than relevance to resource allocation, is a significant source of imprecision. FAN dispenses completely with *a priori* traffic specification since all flows are treated equitably.

Admission control for variable bit rate flows is a widely researched subject. In our view, the only practical solutions to emerge are measurement-based and assume bufferless multiplexing. This is the approach adopted in FAN.

The notion of deterministic QoS, as envisaged in Intserv, does lead to rigorous admission control rules based on the network calculus. Reliance on a rule-based traffic specification is compounded here by extremely pessimistic worst case traffic assumptions. The end result is generally huge over-provisioning with realized performance orders of magnitude better than the guaranteed bounds.

The issue of scalability is well-known when state must be signalled and maintained within network elements. In FAN scalability is less of an issue since required state is simpler and has only local significance.

C. Dynamic pricing

A theoretically attractive scheme avoiding the need to introduce supplementary network mechanisms is to control resource sharing through price signals, as envisaged in [19] for example. We believe, however, that reliance on congestion pricing cannot constitute a viable solution for a commercial network provider.

The main role of pricing for a provider is return on investment. The entire cost of running a network must be shared between users by means of an appropriate tariff structure. Given RoI, there is no real reason for resources to be scarce in a well-managed network. Congestion could then be interpreted rather as a sign of bad management (inaccurate forecasting, unreliable equipment,...). It appears quite unreasonable to require users to pay extra, on top of tariffs designed for RoI, to compensate for these errors.

In FAN, congestion leads to blocking but admitted flows are guaranteed adequate quality. A pricing scheme based on usage may then be more acceptable: exceptionally supply may be insufficient but only users whose demand is satisfied bear the cost.

D. Diffserv and traffic engineering

The most recent QoS proposals emerging from IETF are based on a combination of MPLS traffic engineering and Diffserv mechanisms. Rather than building on previous proposals, this development seems to ignore some of the basic lessons of research on statistical multiplexing and bandwidth sharing.

The following is a quote from RFC 2702 [1]:

> For the purpose of bandwidth allocation, a single canonical value of bandwidth requirements can be computed from a traffic trunk's traffic parameters. Techniques for performing these computations are well known. One example of this is the theory of effective bandwidth.

This assumption underlies the development of traffic engineering methods that effectively assimilate all demands to constant rate pipes. In fact, the effective bandwidth of a flow is link dependent, varying with bit rate and buffer size [18]. There are no techniques to derive a 'single canonical value'.

Rule-based traffic parameters used to compute resource requirements are typically quite different to the actual traffic characteristics of the traffic trunk. Users generally largely overestimate their requirements. This discrepancy leads to an economic requirement to overbook. Of course, there are no analytical techniques to determine the necessary overbooking factor leading to considerable imprecision in the very meaning of QoS guarantee.

Diffserv-aware traffic engineering would, to the authors' understanding, realize a range of service classes by using different under- or over-provisioning ratios per class [14]. The least one can say is that there is no analytical or experimental evidence to support the validity of this approach. In fact, results from all mathematical models of statistical multiplexing, including the theory of effective bandwidth, lead us to conclude that this approach is flawed.

It is the inherent difficulty of characterizing flows and aggregates of flows that lead us to propose FAN. This architecture avoids the need to specify traffic parameters and uses measurement-based admission control to account precisely for real traffic characteristics.

V. Realizing FAN

The presented flow-aware networking architecture remains a vision. In this section we consider the prospects for realizing the *Cross-protect* mechanisms and discuss introduction strategies.

A. Cross-protect mechanisms

One possible realization of *Cross-protect* is presented in [21]. The critical points are the management of the list of protected flows and the realization of priority fair queueing.

The feasibility of maintaining the list of flows at link speeds up to OC 48 has been demonstrated by Caspian Networks [12]. Given that FAN has simpler flow state, the realization of protected flow lists can be realized at least as efficiently.

The PFQ algorithm proposed in [21] is derived from Start-time fair queueing [15]. The resulting complexity appears slight though simpler alternative realizations may be possible.

We claim that the number of flows to be considered by the scheduler is bounded independently of link rate. The required processing speed thus increases only in proportion to this rate. The reason is that the scheduler only deals with flows that currently have (or have very recently had) at least one packet in the queue and this number is bounded with high probability.

Flows whose rate is less than the fair rate have at most one packet in the queue. Assuming flows are independent, the arrival process of such packets is locally Poisson. Since admission control maintains the local load less than a certain level (the principle of bufferless multiplexing), the number of packets in queue behaves locally like an M/G/1 queue and can be bounded with any given high probability by appropriately choosing this level.

Flows that are backlogged actually realize a share of link bandwidth equal to the fair rate. Since admission control maintains the fair rate above a certain threshold, the number of such backlogged flows is necessarily bounded (less than the link rate divided by the imposed threshold). The threshold is chosen such that the probability of blocking is very small except in overload.

The maximum number of flows to be scheduled occurs when there is a mix of a few backlogged flows and a lot of low rate, small packet flows. The exact value depends on the required degree of precision realized by admission control and scheduling. It is measured in hundreds rather than millions, however.

B. Introducing FAN

The described concept of FAN can only be introduced when a vendor has implemented the *Cross-protect* mechanisms in its router architecture. A limited form of flow-aware networking, consisting in the use of implicit admission control, could be introduced before that. This would take the form of a box added to a link interface that is capable of identifying flows and discarding packets of new flows when necessary. This kind of partial solution could be used to preserve the performance of overloaded network links or to protect priority traffic on user access lines, for example.

Cross-protect routers could be introduced progressively, gradually improving the efficiency of a network by allowing controlled performance on possibly heavily loaded links. A completely equipped core network would appear transparent with respect to quality degradation for the majority of flows. Each link is controlled independently. This implies flows may be accepted on some links of a path but rejected on another. This simply means state in the protected flow lists may exist without purpose until erased on time-out.

There is no need for inter-network agreements or reliance on trust to interpret class of service code points. A significant advantage of FAN is that there is virtually no requirement for standardization. One exception would be an agreed convention for defining the flow identifier by combining the flow label with IP addresses, as suggested earlier. A large variety of proprietary implementations of admission control and PFQ can coexist.

C. Coexistence with other architectures

As discussed previously, we have limited faith in the effectiveness of existing proposals for a QoS architecture. However, these are based on mechanisms that are implemented in routers and are already used for certain premium services like VoIP trunks, virtual leased lines and virtual private networks. Fortunately, FAN can coexist with such services by controlling just the residual traffic handled in the default best effort class. It is sufficient to ensure the premium class services enjoy priority access to their reserved bandwidth. The *Cross-protect* mechanisms can then make optimal use of the residual capacity allowing the parallel development of user controlled streaming and elastic applications for which a minimum level of quality is assured.

VI. Concluding remarks

Flow-aware networking, based on the *Cross-protect* mechanisms, is a recent proposition and has considerable scope for improvement and further development. It is, however, built on a long reflection on the best way to meet quality of service requirements. We see it as a minimalist solution, enhancing the service building potential of the current Internet which remains resolutely located at the edge. The intention is also to make the network more cost effective by allowing closer control of provisioning and more precise capacity planning.

We hope this paper will stimulate further research. Our own work is currently in several directions. We are continuing the performance evaluation reported in [21] paying special attention to the calibration of measurement-based admission control. We are also seeking more efficient algorithms for realizing the admission control and PFQ mechanisms to facilitate implementation. Our aim is to work with a router vendor on a full scale implementation taking account of practical constraints including the coexistence of other scheduling and queue management mechanisms.

References

[1] D. Awduche, J. Malcolm, J. Agogbua, M. O'Dell, J. McManus, Requirements for Traffic Engineering Over MPLS, IETF RFC 2702, 1999.

[2] N. Benameur, S. Ben Fredj, S. Oueslati-Boulahia, J. Roberts. Quality of service and flow-aware admission control in the Internet, In Computer Networks, Vol 40, pages 57-71, 2002. (http://perso.rd.francetelecom.fr/roberts/Pub/BBOR02.pdf)

[3] S. Ben Fredj, T. Bonald, A. Proutière, G. Régnié and J.W. Roberts. Statistical bandwidth sharing: A study of congestion at flow level, In Proc. of ACM SIGCOMM 2001.

[4] N. Benameur, A. Kortebi, S. Oueslati, J. Roberts, Selective service protection in overload: differentiated services or per-flow admission control?, Networks 04, Munich, June 2004.

[5] S. Ben Fredj, S. Oueslati-Boulahia, J.W. Roberts. Measurement-based Admission Control for Elastic Traffic. in J. Moreira de Souza, N. Fonseca and E.A. de Souza e Silva, Teletraffic Engineering in the Internet Era, ITC 17, Elsevier, December 2001.
(http://perso.rd.francetelecom.fr/roberts/Pub/BOR01.pdf)

[6] T. Bonald and L. Massoulié, Impact of Fairness on Internet Performance, In Proc. of SIGMETRICS 2001, Cambridge, MA, USA, June 2001.

[7] T. Bonald, J. Roberts, Congestion at flow level and the impact of user behaviour, Computer Networks Vol. 42, 2003.

[8] T. Bonald, S. Oueslati, J. Roberts. IP traffic and QoS control: Towards a flow-aware architecture. In Proc. of World Telecom. Conf., Paris 2002.
(http://perso.rd.francetelecom.fr/roberts/Pub/BOR02a.pdf)

[9] T. Bonald, A. Proutière, J. Roberts. Statistical performance guarantees for streaming flows using expedited forwarding, In Proc. of IEEE INFOCOM 2001.

[10] T. Bonald, A. Proutire, Insensitive bandwidth sharing in data networks, Queueing Systems and Applications, Vol 44, pp 69-100, 2003.

[11] L. Breslau, E. W. Knightly, S. Shenker, I. Stoica, and H. Zhang. Endpoint Admission Control: Architectural Issues and Performance, In Proc. of ACM SIGCOMM 2000, pages 57-69, Stockholm, Sweden, October 2000.

[12] Caspian Networks http://www.caspiannetworks.com/

[13] D. Clark, The design philosophy of the DARPA Internet protocols, Proceedings of Sigcomm '88, August 1988.

[14] F. Le Faucheur, W. Lai, Requirements for Support of Differentiated Services-aware MPLS Traffic Engineering, IETF RFC 3564, 2003.

[15] P. Goyal, H. Vin, H. Cheng. Start-time fair queueing: A scheduling algorithm for integrated services packet switching networks. IEEE/ACM ToN, Vol 5, No 5, Oct 1997.

[16] M. Grossglauser and D. Tse, A Time-Scale Decomposition Approach to Measurement-Based Admission Control, IEEE/ACM Transactions on Networking, Vol 11, No 4, August, 2003.

[17] E. L. Hahne, "Round robin scheduling for fair flow control in data communications networks," IEEE JSAC, vol. 9, no. 7, pp. 1024-1039, Sept. 1991.

[18] F. Kelly, Notes on effective bandwidths, in Stochastic Networks: Theory and applications (ed. F. Kelly, S. Zachary, I. Ziedins), Royal Society Lecture Notes Series, Vol 4, OUP, pp 141-168, 1996.

[19] F. P. Kelly. Models for a self-managed Internet, In Philosophical Transactions of the Royal Society, Vol A358, pages 2335-2348, 2000.

[20] S. Keshav, A Control-theoretic Approach to Flow Control, Proc. ACM Sigcomm 1991, September 1991.

[21] A. Kortebi, S. Oueslati, J. Roberts, Cross-protect: implicit service differentiation and admission control, Proceedings of HPSR'04, Phoenix, 2004.
http://perso.rd.francetelecom.fr/roberts/Pub/KOR03.pdf

[22] L. Massoulié, J. Roberts, Bandwidth sharing: Objectives and algorithms, Proceedings of IEEE Infocom, 1999.

[23] Cisco Systems, Netflow Version 9, 2003

[24] J. Roberts, Internet Traffic, QoS and Pricing, Proceedings of IEEE, to appear, 2004.
http://perso.rd.francetelecom.fr/roberts/Pub/Rob03.pdf

[25] J. Roberts, A survey on statistical bandwidth sharing, Computer Networks, to appear, 2004.
http://perso.rd.francetelecom.fr/roberts/Pub/Rob04.pdf

[26] J. Roberts, U. Mocci, J. Virtamo (Eds), Broadband network teletraffic, LNCS Vol 1155, Springer, 1996.

[27] J. Roberts, S. Oueslati-Boulahia. Quality of service by flow-aware networking. In Philosophical Transactions of the Royal Society, Vol A358, pages 2197-2207, 2000.
(http://perso.rd.francetelecom.fr/roberts/Pub/RO00.pdf)

[28] J. Saltzer, D. Reed, D. Clark, End-to-end arguments in system design, ACM Transactions on Computer Systems, Vol 2, No 4, pp 277-288, 1984.

[29] 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 Journal in Selected Areas in Communications, August 1999.