Per-flow Guarantees under Class-Based Priority Queueing

Jens Schmitt, Paul Hurley, Matthias Hollick, Ralf Steinmetz
Multimedia Communications Lab (KOM), Department of Electrical Engineering & Information Technology
Darmstadt University of Technology, Merckstrasse 25, 64283 Darmstadt, Germany
{Jens.Schmitt, Paul.Hurley, Matthias.Hollick, Ralf.Steinmetz}@KOM.tu-darmstadt.de

Abstract -- **In this paper, we present an admission control scheme which provides per-flow delay and bandwidth guarantees based solely upon simple class-based strict priority queueing. We derive basic properties of the worst-case behaviour in strict priority queueing systems using network calculus. Building upon these properties the flow admission control scheme is devised. The rationale behind this work is the appealing simplicity as well as the almost ubiquitous availability of strict priority queueing in today's routers and the thus promising applicability of our results for practical purposes in providing quality of service (QoS) in the Internet.**

Keywords -- **QoS, admission control, strict priority queueing.**

I. Introduction

A. Motivation

The provision of multiple differentiated services over the Internet, often coined by the term QoS, is a notoriously difficult problem. Many different schemes have been devised and supported in standard efforts [1, 2]. Yet, success looks different. There is of course many reasons and often they are much more contrived than just technical issues like scalability. We argue that one of the big problems of existing approaches is a lack of simplicity and availability. Therefore, we want to make a first step towards a very simple solution available today, based on strict priority queueing (many router products have been offering strict priority queueing for some time, see e.g. [3]). Furthermore, we wish to keep the interface towards the differentiated services simple by providing worst-case properties like the maximum delay that may be experienced by a flow. From our perspective it is important that the service interface allows for per-flow guarantees while the service implementation is only based on class differentiation. This allows for both, simplicity in the implementation and in the semantics of the "service contract".

In particular, we derive the basis for providing per-flow delay and bandwidth guarantees under class-based strict priority queueing by developing a suitable admission control.

B. Related Work

There is of course an almost intimidating body of work in providing QoS in the Internet (see [4] for a recent and excellent overview). Very directly related to our work on strict priority queueing there are interesting results from queueing theory which are however restricted mainly to the equilibrium behaviour of the system and furthermore make usually fairly strong assumptions on the statistical characteristics of the arrival processes (see [5] for a good overview). We focus on worst-case guarantees because they can be derived without knowledge of statistical properties of arrival processes as long as these can be bounded. Furthermore, statistical assurances bring along a number of difficulties at the service interface since violations of the service contract cannot be interpreted unambiguously.

Closely related to our work are [6, 7]. They derive a similar result as our Theorem 4 (see Section III.B) for the special case of two service classes in the context of DiffServ's Expedited Forwarding Per-Hop Behaviour (EF PHB) [8]. However, they focus on a different aspect, namely what they call aggregate scheduling. This is the problem where flows from multiple entry points to the network accumulate inside the network and how this affects the worst-case bounds. They assume no knowledge of the network topology and thus arrive at very restrictive bounds. We focus on the case where the topology and the paths taken by flows are known by the admission control scheme, e.g., by using a (logically) centralized bandwidth broker or by using multi-protocol label switching (MPLS) [9] to prevent flows from accumulating inside the network.

In [10], statistical guarantees for two class priority queueing are derived based on the so-called negligible jitter conjecture. We focus on worst-case respectively deterministic guarantees.

To some extent similar in spirit is the work in [11], because it also aims at providing deterministic per-flow guarantees without per-flow state in routers. However, the approach is very different and based on a concept called dynamic packet state which essentially means that packets carry with them state that allows the enrichment of non-per-flow state in routers towards per-flow state and thus to do per-flow traffic management again. This is an interesting theoretical approach which however is a far cry from current router technology and involves some non-trivial implementation issues.

C. Outline

After giving some background information on network calculus and its notation in Section II, we derive basic properties of strict priority queueing based on network calculus methods in Section III. In Section IV, we then use these basic properties to develop a fairly simple admission control scheme which allows to offer per-flow delay and bandwidth guarantees despite purely class-based traffic control mechanisms on the data path of the system. Section V concludes the paper.

II. Network Calculus Background

Network Calculus is a tool to analyse flow control problems in networks with particular focus on determination of bounds on worst-case performance. In particular, it abstracts traffic regulation and scheduling schemes from which one may derive general results. It is a framework to derive deterministic guarantees on throughput, delay, and to ensure no losses in packet-switched networks. The focus of this paper is on routers which internally operate with strict priority queueing and which input is constrained by the use of traffic regulation schemes as for example token buckets.

We shall now provide some basic definitions and notation before summarizing some basic results from network calculus. In depth results are given in the text [12].

DEFINITION: The *input function* $R(t)$ of an arrival process is the number of bits that arrive in the interval [0,*t*]. In particular, $R(0) = 0$ and *R* is wide-sense increasing, i.e. $R(t_1) \le R(t_2)$ for all $t_1 \le t_2$.

DEFINITION: The *output function* $R^o(t)$ of a system *S* is the number of bits that have left *S* in the interval [0,*t*]. In particular, $R^o(0) = 0$ and R^o is wide-sense increasing.

DEFINITION: Min-Plus Convolution. Let *f* and *g* be wide-sense increasing and $f(0) = g(0) = 0$. Then their convolution under min-plus algebra is defined as $(f \otimes g)(t) = inf_{0 \le s \le t}\{f(t-s) + g(s)\}$.

We now define, by means of the min-plus convolution, the arrival and service curve. A wide-sense increasing function α with $\alpha(t) = 0$ for $t < 0$ is called an arrival curve for an input function *R* if $R \le R \otimes \alpha$. We also say *R* is α-smooth or *R* is constrained by α.

DEFINITION: Arrival Curve. Let α be a wide-sense increasing function α such that $\alpha(t) = 0$ for $t < 0$. α is an arrival curve for an input function *R* if $R \le R \otimes \alpha$.

DEFINITION: Service Curve. Consider a system *S* and a flow through *S* with *R* and R^o. *S* offers a service curve β to the flow if β is wide-sense increasing and $R^o \ge R \otimes \beta$.

From these, it is now possible to capture the major worst-case properties for data flows: maximum delay and maximum backlog. These are stated in the following theorems.

THEOREM 1: Backlog Bound
Let a flow *R(t)*, constrained by an arrival curve α, traverse a system *S* that offers a service curve β. The backlog *x(t)* for all *t* satisfies:

$$x(t) \le sup_{s \ge 0}\{\alpha(s) - \beta(s)\} = v(\alpha, \beta). \qquad (1)$$

$v(\alpha, \beta)$ is also often called the vertical deviation between α and β.

THEOREM 2: Delay Bound
Assume a flow *R(t)* constrained by arrival curve α traverses a system *S* that offers a service curve β. At any time *t,* the virtual delay *d(t)* satisfies:

$$d(t) \le sup_{s \ge 0}\{inf_{\tau \ge 0}\{\alpha(s) \le \beta(s+\tau)\}\} = h(\alpha, \beta) \qquad (2)$$

$h(\alpha, \beta)$ is also often called the horizontal deviation between α and β.

A typical example of an arrival curve is given by

$$\gamma_{r,b}(t) = rt + b \qquad (3)$$

which results from using the prominent token bucket algorithm as traffic regulation mechanism.

A typical example of a service curve is given by

$$\beta_{R,T}(t) = R(t-T)^+ \qquad (4)$$

where the notation $(x)^+$ denotes *x* if $x \ge 0$ and 0 otherwise. This is often also called a rate-latency service curve.

III. STRICT PRIORITY QUEUEING: ANALYSIS OF WORST-CASE

A. Strict Priority Queueing under General Arrival Curves

We now use network calculus to analyse strict priority queueing for a given number of classes *n* and under the assumption that the input of each class *i* is constrained by α_i for $i = 1, \ldots, n$. There are often no guarantees associated with the lowest priority and thus it is often not necessary to have a constraint on its input functions. However, we follow the more general case where guarantees are required also from the lowest priority.

1) Service Curve for Strict Priority Queueing: First we derive the respective service curves for each class under strict priority queueing. The following theorem states the interesting result that service curves of lower priority classes are dependent on the arrival curves of higher priority classes

THEOREM 3: Let *C* be the overall capacity of the system. Let α_i be the arrival curve for input to class *i*. The service curve β_i^P for class *i* is given by

$$\beta_i^P(t) = \left(Ct - \sum_{j=1}^{i-1} \alpha_j(t) - max_{i+1 \le j \le n}\{l_j^{max}\}\right)^+ \qquad (5)$$

for $i = 1, \ldots, n$.

Here l_j^{max} is the maximum size of a packet in class *j*.

PROOF:

Let $R_i(t)$, $R_i^o(t)$ be the input and output function for traffic from class *i* for $i = 1, \ldots, n$. Now, let *s* be the start of the last busy period due to traffic from classes 1 to *i* before a fixed time *t*. Then the amount of service given traffic from class i is lower bounded by the server output minus the service given to higher traffic classes and the maximum packet size for lower traffic classes for which a single packet might just have started service before *s*. The server output in interval [*s*,*t*] is given by $C(t-s)$ due to the definition of a busy period. Thus we have

$$\begin{aligned} R_i^o(t) - R_i^o(s) &\ge C(t-s) \\ &\quad - \sum_{j=1}^{i-1}(R_j^o(t) - R_j^o(s)) - max_{i+1 \le j \le n}\{l_j^{max}\} \end{aligned} \qquad (6)$$

Due to s being the start of a busy period for traffic from classes $j = 1, \ldots, i$ we also have $R_j^o(s) = R_j(s)$. Thus

$$\begin{aligned} R_j^o(t) - R_j^o(s) &= R_j^o(t) - R_j(s) \\ &\le R_j(t) - R_j(s) \le \alpha_j(t-s) \end{aligned} \qquad (7)$$

That means we can bring the arrival curve constraints into (6).

Note that the bound in (7) is tight because at time *t* input and output function for traffic from class *i* could well be equal and of course traffic could be greedy. Introducing (7) in (6) we obtain

$$R_i^o(t) - R_i^o(s) \geq C(t-s) - \sum_{j=1}^{i-1} \alpha_j(t-s) - max_{i+1 \leq j \leq n}\{l_j^{max}\} \tag{8}$$

Since R_j^o is wide-sense increasing we obtain

$$\begin{aligned} R_i^o(t) &\geq R_i^o(s) + \left(C(t-s) - \sum_{j=1}^{i-1} \alpha_j(t-s) - max_{i+1 \leq j \leq n}\{l_j^{max}\} \right)^+ \\ &= R_i^o(s) + \beta_i^P(t-s) \\ &\geq inf_{0 \leq s \leq t}\{R_i^o(s) + \beta_i^P(t-s)\} \\ &= (R_i^o \otimes \beta_i^P)(t) \end{aligned} \tag{9}$$

Thus, indeed, strict priority queueing offers β_i^P as a service curve towards traffic from class *i*.

■

The theorem is the basis for all subsequent findings of the paper. Moreover, it contains a very constructive result:

> *There is a quantifiable dependency of lower priorities' service curves on arrival curves of higher priority classes.*

There are several ways to use this in practical networking problems as for example in flow or packet admission control for class-based networks. In particular for flow admission control, it allows to dimension aggregate arrival curves for each class such that certain delay targets for each class are achieved. New flow requests for a class can then be checked by the admission control against whether the sum of arrival curves of already admitted flows and the new flow is still below the aggregate arrival curve which is necessary to achieve the delay target.

B. Strict Priority Queueing under Token Buckets

In this section, we now assume a particular arrival curve, the popular token bucket [13]. Under this assumption we can concretize the service curve for general arrival curves and can then derive bounds on maximum backlog and delay per class.

1) Service Curve: First we apply Theorem 3 to the special case of token buckets as arrival curves for the different classes in order to derive the service curve for strict priority queueing. Theorem 4 states the result.

THEOREM 4: (Service Curve under Token Buckets)
Let $\alpha_j = \gamma_{r_j, b_j}$ be the arrival curves for all traffic classes $j = 1, \ldots, n$, i.e. each traffic class is constrained by a token bucket (each with its own parameters). The service curve for class *i* under strict priority queueing is then given by

$$\beta_i^P = \beta_{R^P, T^P} \tag{10}$$

with

$$R_i^P = C - \sum_{j=1}^{i-1} r_j \text{ and } T_i^P = \frac{\sum_{j=1}^{i-1} b_j + max_{i+1 \leq j \leq n}\{l_j^{max}\}}{C - \sum_{j=1}^{i-1} r_j}$$

That means the service curve is of the rate-latency type.

PROOF:
The theorem is a consequence of Theorem 3 and the definition of the rate-latency service curve in (4):

$$\begin{aligned} \beta_i^P(t) &= \left(Ct - \sum_{j=1}^{i-1} \alpha_j(t) - max_{i+1 \leq j \leq n}\{l_j^{max}\} \right)^+ \\ &= \left(Ct - \sum_{j=1}^{i-1} (r_j t + b_j) - max_{i+1 \leq j \leq n}\{l_j^{max}\} \right)^+ \\ &= \left(C - \sum_{j=1}^{i-1} r_j \right) \left(t - \frac{\sum_{j=1}^{i-1} b_j + max_{i+1 \leq j \leq n}\{l_j^{max}\}}{C - \sum_{j=1}^{i-1} r_j} \right)^+ \\ &= \beta_{R_i^P, T_i^P}(t) \end{aligned}$$

■

2) Delay and Backlog: Using the service curve for strict priority queueing we can now derive the worst-case delay bound as well as the maximum backlog bound for each traffic class. The backlog bound is given by the following theorem.

THEOREM 5: (Per-Class Backlog Bound under Token Buckets)
Let $\alpha_j = \gamma_{r_j, b_j}$ be the arrival curves for all traffic classes $j = 1, \ldots, n$, i.e. each traffic class is constrained by a token bucket (each with its own parameters). For stability we further assume that $C \geq \sum_{i=1}^{n} r_i$.

The maximum backlog per traffic class *i* is bounded by the vertical deviation between the arrival curve to class *i*, γ_{r_i, b_i}, and its service curve, β_i^P

$$v(\gamma_{r_i, b_i}, \beta_i^P) = r_i \times \frac{\sum_{j=1}^{i-1} b_j + max_{i+1 \leq j \leq n}\{l_j^{max}\}}{C - \sum_{j=1}^{i-1} r_j} + b_i \tag{11}$$

PROOF:

Due to the stability condition we have $C - \sum_{j=1}^{i-1} r_j \ge r_i$, i.e. the slope of the service curve is higher than that of the arrival curve. That means the maximum vertical deviation is taken on at the latency of the service curve, because the service curve comes ever closer once the service is “started”, i.e.

$$\begin{aligned} v(\gamma_{r_i, b_i}, \beta_i^P) &= sup_{s \ge 0}\{\gamma_{r_i, b_i}(s) - \beta_i^P(s)\} \\ &= \gamma_{r_i, b_i}(T_i^P) - \beta_i^P(T_i^P) \\ &= \gamma_{r_i, b_i}(T_i^P) \\ &= r_i \times \frac{\sum_{i=1}^{i-1} b_j + max_{i+1 \le j \le n}\{l_j^{max}\}}{C - \sum_{j=1}^{i-1} r_j} + b_i \end{aligned}$$

■

Next, we derive the per-class maximum delay bound under the same assumptions in Theorem 6.

THEOREM 6: (Per-Class Delay Bound under Token Buckets) Let $\alpha_j = \gamma_{r_j, b_j}$ be the arrival curves for all traffic classes $j = 1, \ldots, n$, i.e. each traffic class is constrained by a token bucket (each with its own parameters). For stability we further assume that $C \ge \sum_{i=1}^{n} r_i$.

The maximum delay per traffic class *i* is bounded by the horizontal deviation between the arrival curve to class *i*, γ_{r_i, b_i}, and its service curve, β_i^P

$$h(\gamma_{r_i, b_i}, \beta_i^P) = \frac{\sum_{i=1}^{i} b_j + max_{i+1 \le j \le n}\{l_j^{max}\}}{C - \sum_{j=1}^{i-1} r_j} \qquad (12)$$

PROOF:

Following the same arguments as in the proof of Theorem 5, it is clear that the maximum horizontal deviation is taken on at the origin, i.e.

$$\begin{aligned} h(\gamma_{r_i, b_i}, \beta_i^P) &= sup_{s \ge 0}\{inf_{\tau \ge 0}\{\gamma_{r_i, b_i}(s) \le \beta_i^P(s + \tau)\}\} \\ &= inf_{\tau \ge 0}\{\gamma_{r_i, b_i}(0) \le \beta_i^P(\tau)\} \\ &= inf_{\tau \ge 0}\left\{ \begin{array}{c} b_i \le -\sum_{j=1}^{i-1} b_j - max_{i+1 \le j \le n}\{l_j^{max}\} \\ + \left(C - \sum_{j=1}^{i-1} r_j\right)\tau \end{array} \right\} \\ &= inf_{\tau \ge 0}\left\{ \frac{\sum_{i=1}^{i} b_j + max_{i+1 \le j \le n}\{l_j^{max}\}}{C - \sum_{j=1}^{i-1} r_j} \le \tau \right\} \\ &= \frac{\sum_{i=1}^{i} b_j + max_{i+1 \le j \le n}\{l_j^{max}\}}{C - \sum_{j=1}^{i-1} r_j} \end{aligned}$$

■

So, we can now compute the worst-case properties for strict priority queueing if we assume each traffic class conforms to a token bucket (respectively make it conform to it by either using admission control at ingress to the network or drop packets according to the token bucket).

IV. ADMISSION CONTROL FOR STRICT PRIORITY QUEUEING

In this section, we now use the basic results on worst-case bounds for strict priority queueing from the preceding section to design an admission control scheme which allows to give per-flow delay and rate guarantees despite the purely class-based priority queueing. We can distinguish two cases here:

- static bandwidth shares for different priority classes,
- dynamic bandwidth shares depending on current traffic.

Both cases can be useful from the perspective of network providers. The static case corresponds to a situation where a network provider wants fairly strict control on how resources are allocated to classes. The dynamic case is inherently more efficient but also involves more complexity due to on-line reconfigurations for the class token buckets. Therefore, we decide to only treat the static case here and postpone the dynamic case to future work.

A. Admission Control under Static Bandwidth Shares

Here, we assume that we are given maximum bandwidth shares per class, $\phi_i > 0$, $i = 1, \ldots, n$, and want to ensure certain class delay targets D_i, $i = 1, \ldots, n$, with $D_i < D_j$ if $i < j$. The following theorem provides how the class token buckets have to be dimensioned:

THEOREM 7: Dimensioning of Class Token Buckets

To achieve the class delay targets D_i, $i = 1, \ldots, n$ and to ensure that each class obtains its bandwidth share ϕ_i, the class token buckets have to be chosen as

$$r_i = \phi_i C \qquad (13)$$

$$b_i = D_i\left(C - \sum_{j=1}^{i-1} r_j\right) - \sum_{j=1}^{i-1} b_j - max_{i+1 \le j \le n}\{l_j^{max}\}$$
$$\Leftrightarrow \sum_{j=1}^{i} b_j = D_i\left(C - \sum_{j=1}^{i-1} r_j\right) - max_{i+1 \le j \le n}\{l_j^{max}\} \qquad (14)$$

for $i = 1, \ldots, n$.

PROOF:

It is obvious that in order to achieve the bandwidth assurance per class the token bucket rates need to be set proportional to their shares as in (13). The class bucket depths can then be calculated by setting the horizontal deviation of the classes' service curves equal to the class delay target:

$$h(\gamma_{r_i, b_i}, \beta_i^P) = D_i \qquad (15)$$

$$\Rightarrow \frac{\sum_{j=1}^{i} b_j + max_{i+1 \le j \le n}\{l_j^{max}\}}{C - \sum_{j=1}^{i-1} r_j} = D_i \qquad (16)$$

$$\Rightarrow b_i = D_i\left(C - \sum_{j=1}^{i-1} r_j\right) - \sum_{j=1}^{i-1} b_j - max_{i+1 \le j \le n}\{l_j^{max}\} \qquad (17)$$

■

Note that (14) constitutes a system of n linearly independent equations for n unknowns, i.e. it always has a unique solution. However, if some of the b_i are negative this indicates that for the given bandwidth shares and class delay targets there is *no* allocation of token buckets which can achieve these. Furthermore, as (14) results in a lower triangular matrix it is very simple to solve.

This result can now be used for a simple admission control scheme: if a new flow with bandwidth requirements (r_{new}, b_{new}) and a maximum delay requirement d_{new} arrives it can be assigned to the lowest priority class i for which

$$d_{new} \ge D_i, \sum_{j=1}^{k} r_i^j + r_{new} \le r_i \text{ and } \sum_{j=1}^{k} b_i^j + b_{new} \le b_i$$

where r_i^j and b_i^j are the bandwidth requirements of already accepted flows in class i, $j = 1, \ldots, k$.

Such a class i may not exist which means the new flow has to be rejected. Since assigning flows to classes, which provide a much lower maximum delay than expected by the flow, might be undesirable under certain circumstances (e.g., because it may economically be more promising to serve more delay-critical future flows) the admission control scheme could also reject flows already if they do not "fit" in that class any more which provides the highest class delay that is just below the flow's target delay.

B. Numerical Example

We assume that 8 priority classes are used, a link speed C of 100 Mb/s (= 12500 KB/s), and a maximum packet size of 1500 bytes for all classes. The bandwidth shares ϕ_i, $i = 1, \ldots, 8$ for each priority class are given in Table I as well as the classes delay targets D_i. From this input the token bucket rates and depths, r_i and b_i, for each class can be calculated, the results are also given in Table I.

TABLE I

EXAMPLE WITH 8 PRIORITY CLASSES.

Class i	ϕ_i (%)	D_i (ms)	r_i (KB/s)	b_i (KB)
1	5	5	625	61
2	5	20	625	175
3	5	40	625	213
4	5	60	625	188
5	10	80	1250	163
6	10	100	1250	75
7	10	150	1250	250
8	50	200	6250	125

With these numbers the admission control decision can now be made simply by assigning a flow to the class which provides an appropriate delay and which can still accommodate the flow with respect to the class token bucket.

The selection of delay targets governs the efficiency of the admission control scheme because if the distribution of delay requirements does not fit well with the distribution of class delay targets then the assignment of flows to classes can result in some wastage of resources due to providing "too good" service to flows. Here, a provider needs some experience with its customers' demands to select delay targets advantageously. Note, however, that it is not always possible to find a set of class token buckets for a given set of class delay targets and bandwidth shares. E.g., if in our example from Table I class 6 should be assigned a delay target of 90 ms (instead of 100 ms) this would result in negative token bucket depths which means this assignment of delay targets is infeasible.

Similarly, the selection of bandwidth shares needs some experience by a provider. The choice of the bandwidth shares determines how many flows can be admitted for certain delay requirement regions. Again, this should be aligned as well as possible with actual traffic demand.

V. Conclusions

We propose simple admission control rules under strict class-based priority queueing which facilitate deterministic guarantees on delay and bandwidth per flow. To do so we derived basic worst-case properties for the case where strict priority queueing is used as packet scheduling mechanism in routers and arrivals to classes are regulated by simple token buckets. One advantage of our scheme, particularly in comparison to other approaches, that we perceive is the availability of all required base mechanisms in today's routers as well as the simplicity of the scheme, both in terms of implementation and at the service interface. The admission control scheme makes use of the fact that for priority queueing there is a quantifiable dependency between the service curves of lower priority classes and the arrival curves of higher priority classes. This allows the dimensioning of the token buckets (as particular choice of arrival curves) such that a suitable range of maximum delay requirements can be matched well against the classes' service menu.

VI. References

[1] R. Braden, D. Clark, and S. Shenker. Integrated Services in the Internet Architecture: an Overview. Informational RFC 1633, June 1994.

[2] K. Nichols, S. Blake, F. Baker, and D. Black. Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers. Proposed Standard RFC 2474, December 1998.

[3] Cisco Systems: Configuring Priority Queueing, 2000. Available under http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/qos_% c/qcprt2/qcdpq.pdf.

[4] V. Firoiu, J.-Y. Le Boudec, D. Towsley, and Z.-L. Zhang. Theories and Models for Internet Quality of Service. *Proceedings of the IEEE*, 90(9):1565–1591, September 2002.

[5] T. G. Robertazzi. *Computer Networks and Systems*. Springer, 3rd Edition, 2000.

[6] F. Farkas and J.-Y. Le Boudec. A Delay Bound for a Network with Aggregate Scheduling. In *Proceedings of Sixteenth UK Teletraffic Symposium on Management of Quality of Service, Harlow, UK*, May 2000.

[7] A. Charny and J. Y. L. Boudec. Delay Bounds in a Network with Aggregate Scheduling. In *Proceedings of Quality of future Internet Services Workshop (QofIS 2000), Berlin, Germany*, pages 105–116. Springer LNCS, September 2000. ISBN 3-540-41076-7.

[8] B. Davie, A. Charny, J. Bennett, K. Benson, J.-Y. L. W. Courtney, S. Davari, V. Firoiu, and D. Stiliadis. An Expedited Forwarding PHB (Per-Hop Behavior). Proposed Standard RFC 3246, March 2002.

[9] E. Rosen, A. Viswanathan, and R. Callon. Multiprotocol Label Switching Architecture. Proposed Standard RFC 3031, January 2001.

[10] T. Bonald, A. Proutiere, and J. Roberts. Statistical performance guarantees for streaming flows using expedited forwarding. In *Proceedings of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'2001)*, pages 1104–1112. IEEE, April 2001.

[11] I. Stoica and H. Zhang. Providing Guaranteed Services Without Per Flow Management. *ACM Computer Communication Review*, 29(4):81–94, September 1999. Proceedings of SIGCOMM'99 Conference.

[12] J.-Y. Le Boudec and P. Thiran. *Network Calculus - A Theory of Deterministic Queueing Systems for the Internet*. Springer, Lecture Notes in Computer Science, LNCS 2050, 2001.

[13] J. S. Turner. New Directions in Communications. *IEEE Communications Magazine*, 24(10):8–15, October 1986.