DSS: A Deterministic and Scalable QoS Provisioning Scheme

Guillaume Urvoy-Keller and Ernst W. Biersack

Institut Eurecom
2229, route des Crêtes
06904 Sophia-Antipolis, France
{urvoy,erbi}@eurecom.fr

Abstract. The design of traffic management schemes for multimedia applications is a challenge for the future Internet. The main problem is the high burstiness of the multimedia traffic. Most of the traffic management schemes try to minimize the total bandwidth required to serve multimedia applications. However, these schemes typically suffer from a poor scalability. Rather than minimizing the burstiness of the multimedia sessions, we present an approach that tries to benefit from high variability of multimedia sessions to maximize the bandwidth offered to the best-effort traffic. The proposed scheme, called DSS, is based on shaping of the flows and on the use of the GPS scheduling policy in the routers. We show that DSS is highly scalable and that its bandwidth requirement for multimedia sessions remains reasonable, in particular smaller than the bandwidth requirement of RPPS.

1 Introduction

Provisioning of Quality of Service (QoS) for multimedia applications in the Internet has received a lot of attention during the last decade. A major difficulty is to accommodate sources with QoS requirements as well as best-effort traffic while maintaining a high level of network bandwidth utilization. The key issues in the design of such a service are: (i) the choice of a service policy (see [1] for an extensive review of the service policies) and (ii) the design of the corresponding admission control algorithm.
However, defining these two elements is not enough. Another important parameter is the scalability of the solution. We say: *a service is scalable if the effort to provide this service does not depend on the number of admitted sessions.* Note that the requirement for scalability is not restricted to the data-transfer phase of a communication, but also applies to the admission control phase.
In this paper, we propose a traffic management scheme that guarantees deterministic QoS to multimedia applications in a scalable way. Sources are assumed to be leaky-bucket constrained with a maximal end-to-end delay requirement. We assume a fluid model that closely approximates the behavior of a packet network with a small packet size compared to the service rate of the servers. The fluid model enables us to concentrate on the central issues. However, recent studies [2–4] have emphasized the complexity of deriving bounds in a packet network due to the discrete nature of the problem.

Since all our results are derived for the case of a fluid flow model, we will discuss this point.
Our service works as follows: (i) sources are shaped adequately prior to their entrance in the network and (ii) each server reserves an amount of bandwidth equal to the sum of the peak rate of the shaped sources. The simplicity of the scheme helps to assure its scalability. Shaping allows to limit the peak rate of the sources in the network. Guaranteeing the resource at each server is achieved with the GPS scheduling policy. Each server will serve two kinds of sources: sources that use the QoS provisioning service and best-effort sources. Since the sources with QoS constraints are variable bit rate sources, using GPS allows to redistribute dynamically the unused bandwidth to the best effort sources. There lies the key idea of our scheme: rather than to maximize the number of sources that can be served simultaneously by the QoS provisioning scheme, a method that often leads to the loss of scalability, we propose a simple and scalable QoS provisioning scheme that tries to maximize the bandwidth provided to the best-effort sources.
The remainder of this paper is organized as follows. In Section 2, we recall the main features of the GPS scheduling policy. In Section 3, we present our QoS provisioning scheme, called DSS for Deterministic Shaping Scheme. In Section 4, we review the relevant studies on the GPS policy concerning the delay bounds and the admission control procedures and show their lack of scalability. We also compare our approach with the one of Reisslein et al. [5], which uses the same deterministic shaping but relies on a statistical bufferless multiplexing to achieve a high bandwidth utilization. In Section 5, we evaluate DSS. We first compare DSS with RPPS and show that DSS requires less bandwidth than RPPS. We also evaluate the ability of DSS to redistribute the bandwidth unused by the sources requiring QoS to best-effort sources. In Section 6, we conclude and provide some insights for future work.

2 Properties of the GPS Scheduler

Generalized Processor Sharing (GPS) is the general model of a work-conserving scheduling policy where each session is assigned a weight and is served according to its relative weight among the backlogged sources. GPS assumes a fluid model of all flows, which have to be infinitely divisible. Let $S_i(t,\tau)$ denote the amount of work received by session i ($i \in \{1,\ldots,N\}$) during $[t,\tau]$ and Φ_i its weight. Then, if session i is continuously backlogged during $[t,\tau]$, we have:

$$\frac{S_i(t,\tau)}{S_j(t,\tau)} \geq \frac{\Phi_i}{\Phi_j} \forall j \in \{1,\ldots,N\} \tag{1}$$

Summing equation (1) over all active sessions j, we obtain:

$$S_i(t,\tau) \geq \frac{\Phi_i}{\sum_j \Phi_j} C = r_i^{\min} \tag{2}$$

Equation (2) means that each session i is guaranteed a minimum service rate $r_i^{\min}$ independently of the behavior of the other active sessions (Isolation property). Equation

(1) further indicates that GPS has the ability to distribute unused bandwidth between the active sessions consistently with their weight assignment. These features, namely isolation between sources and controlled distribution of the server bandwidth, explain why GPS is so attractive in the context of QoS provisioning.

GPS is also popular due to the delay bound it is able to guarantee. The first results concerning the delay bounds achievable with GPS were obtained by Parekh et al. [6]. Bounds are provided for leaky-bucket constrained sources in the particular case of Rate Proportional Processor Sharing (RPPS). RPPS is a special case of GPS where each source S_i is assigned a weight Φ_i equal to its mean rate R_i, i.e. $\Phi_i = R_i$. Assuming a fluid-flow model and neglecting constant propagation delays, the delay bound in [6] for source S_i with leaky bucket parameters R_i (mean rate) and M_i (maximum burst size), may be written as follows:

$$d_i^{\text{eff}} = \frac{M_i}{\min\limits_{j \in \{1,\ldots,K\}} r_i^{\min}(j)} \tag{3}$$

where $r_i^{\min}(j)$ is the minimum service rate guaranteed at node $j \in \{1,\ldots,K\}$ to source S_i. Note that the end-to-end delay in equation (3) is not simply the sum of local bounds.

GPS is only a model of procedure, which is not implementable in a packet network since it assumes that each flow is infinitely divisible (fluid-flow model). A scheme, called Packet Generalized Processor Sharing (PGPS) [7, 6], has been proposed to emulate GPS in a packet network. The emulation used by PGPS is cumbersome and lots of research has concentrated on reducing the computational complexity of GPS emulations [8–10].

3 DSS: Deterministic Shaping Service

3.1 Leaky bucket constrained sources

We consider sources that are leaky bucket constrained with an additional constraint on their peak rate. A traffic descriptor for a given source S comprises three parameters (p, R, M), which are respectively the peak rate, the mean rate and the maximum burst size of the source. Such a source is able to pass through the leaky bucket controller depicted in Figure 1 without experiencing any loss. The size of the token bucket is $M' = M\frac{p-R}{p} < M$ since the peak rate of the source is finite. Our traffic management scheme is based on deterministic QoS guarantees. A given source S has thus one QoS constraint, which is its maximum end-to-end delay requirement (the traffic management scheme ensures a zero loss rate).

3.2 Deterministic Effective Bandwidth

Central to our proposal is the concept of *deterministic effective bandwidth*. It has been first introduced by Le Boudec [11] in the context of the Network Calculus [12]. The

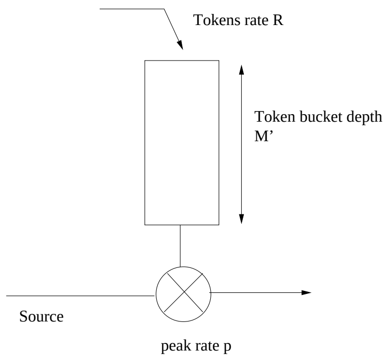

Fig. 1. Leaky bucket controller

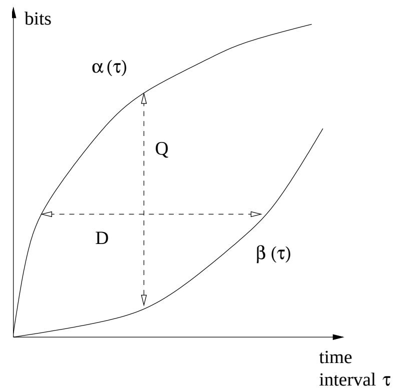

Fig. 2. Upper bound on backlog and virtual delay

Network Calculus provides deterministic bounds on end-to-end delays and backlogs. A source is modeled through an arrival curve that represents an upper bound on the volume of traffic it can send during any time interval τ. In the case of a leaky bucket constrained source S with parameters (p, R, M), an arrival curve α is (see [11]):

$$\alpha(\tau) = \min\left(p\tau, R\tau + M\frac{p-R}{p}\right),\ \tau \geq 0 \tag{4}$$

Servers are modeled through a service curve representing a lower bound on the service they are able to provide during some time intervals. For instance, a service curve β for a FIFO server (and, more generally, for any server implementing a work-conserving policy) with a service rate C is $\beta(\tau) = C \cdot \tau$ (the time intervals, here, can be the backlog periods). Consider now a system with an input flow characterized by an arrival curve α and a server with a service curve β. An upper bound D on virtual delay (resp. Q on backlog), which corresponds to a real delay if the system works in a FIFO manner, is given by the maximal horizontal (resp. vertical) distance (see Figure 2) between the arrival curve and the service curve of the system (Theorems 1 and 2 of [12]). The concept of deterministic effective bandwidth makes use of these theorems. The deterministic effective bandwidth $e_d(\alpha)$ associated to a given source S for a given arrival curve α and a delay constraint d is the minimum service rate that ensures to this source a delay smaller than d. The following result for $e_d(\alpha)$ is given in [11]:

$$e_d(\alpha) = \sup_{s \geq 0}\left(\frac{\alpha(s)}{s+d}\right) \tag{5}$$

In the special case of a leaky bucket constrained source S with an arrival curve given by equation (4) and a delay requirement d, equation (5) may be re-written as (see Figure 3):

$$e_d(\alpha) = \begin{cases} \frac{M}{d+\frac{M}{p}} & \text{if } 0 \leq d \leq M\left(\frac{1}{R} - \frac{1}{p}\right) \\ R & \text{if } d \geq M\left(\frac{1}{R} - \frac{1}{p}\right) \end{cases} \tag{6}$$

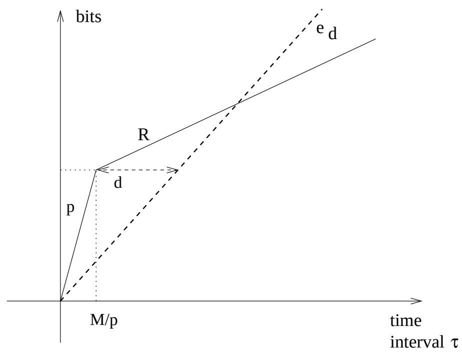

Fig. 3. Deterministic effective bandwidth of a leaky bucket constrained source

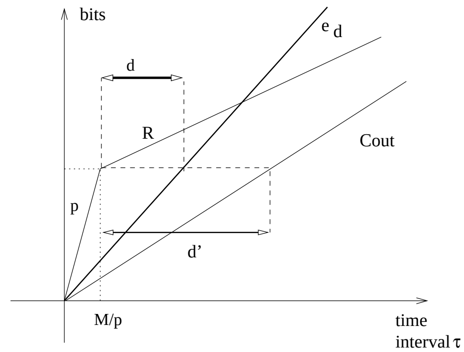

Fig. 4. Choice of the shaping rate

3.3 QoS provisioning in DSS

With our traffic management scheme, sources are shaped, prior to their entrance in the network, to a rate equal to their deterministic effective bandwidth. This operation may be done using a spacer [13].

Theorem 1. *The choice of the deterministic effective bandwidth as a shaping rate is optimal, in the sense that any smaller shaping rate could lead to a QoS violation.*

Proof: let us prove the theorem by contradiction. Suppose we choose a shaping rate $C_{out} < e_d(\alpha)$. Let us prove that there is at least one trajectory of the source such that one bit experiences a delay strictly greater than d. Consider the greedy trajectory of S, where S consumes its tokens as soon as they are available. Then, the cumulative arrival rate of this trajectory corresponds to α and the service rate offered by the spacer is $\beta(t) = C_{out} \cdot t$. As a consequence, the bit emitted at time $t = \frac{M}{p}$ experiments a delay d' strictly larger than d (see Figure 4). This proves the result. □

Let us now focus on the behavior of the servers. Our traffic management scheme, DSS, is designed to guarantee a delay equal to zero (within the context of the fluid model) once a source has been accepted in the network. To guarantee a deterministic QoS, we must allocate to each source a service rate equal to its peak rate, i.e. its deterministic effective bandwidth, on each server along its route. This is done by using the GPS policy. For a given server with service rate C, let $(S_i)_{i \in I}$ (resp. $(e_{d_i}(\alpha_i))_{i \in I}$) be the set of sources crossing this server (resp. their deterministic effective bandwidths). The server works as follows: it manages two classes, QoS and BE, with their respective weights Φ_{QoS} and Φ_{BE} defined as follows:

$$\Phi_{QoS} = \sum_{i \in I} e_{d_i}(\alpha_i) \text{ and } \Phi_{BE} = C - \sum_{i \in I} e_{d_i}(\alpha_i)$$

With such a weight assignment, the minimum guaranteed rate $r_{QoS}^{\min}$ for the QoS class is: $r_{QoS}^{\min} = \frac{\Phi_{QoS}}{\Phi_{QoS} + \Phi_{BE}} C = \sum_{i \in I} e_{d_i}(\alpha_i)$. As a consequence, each source using DSS

receives, at each server, a minimum service rate equal to its deterministic effective bandwidth. Thus, globally, the DSS network guarantees to each source a minimum service rate equal to its deterministic effective bandwidth. This ensures that the source experiences a delay equal to zero in the network. Moreover, since each server manages only two classes, QoS and BE, (independently of the number of sources actually using DSS) DSS remains scalable during the data-transfer phase. This is noteworthy since the solutions purely based on GPS that will be presented in Section 4 suffer from scalability problems at the data-transfer stage since each server must treat each session individually.
Up to now, we have mainly focused on delay. Another important problem is the buffer space requirement. Since the resource allocation is peak rate based, no buffer space needs to be allocated at the nodes inside the network. This is obviously true only in the context of a strict fluid flow model. Nevertheless, this is a major improvement as compared to other traffic management schemes, like the ones using GPS which, even in a fluid framework, must allocate for each source and at each server a buffer space equal to the maximum burst size of the source (see [1]).

3.4 Admission Control Algorithm

Admitting a New Session: for a given server k with a service rate C_k, let $C_k^{\mathrm{QoS}} = \sum_{i \in I} e_{d_i}(\alpha_i)$ be the sum of the deterministic effective bandwidths of all the sessions in the QoS class (at this server). Consider the admission process of a new source S with a deterministic effective bandwidth $e_d(\alpha)$. The admission algorithm consists in checking the following condition for every node k along the route of source S: $C_k^{\mathrm{QoS}} + e_d(\alpha) \leq C_k$. If the session request can be accepted, then every node k only has to increment C_k^{QoS} by $e_d(\alpha)$. Note also that the conditions to be checked involve only the *total* load of the servers and not the number of previously accepted sources. The admission control is therefore scalable.

Session Termination: servers store only the total service rate allocated to the QoS class. Therefore, the admission control algorithm must ensure that when a session ends, each server along the path of this session receives a message indicating the corresponding deterministic effective bandwidth $e_d(\alpha)$ of the terminating session. Each server k must then decrement C_k^{QoS} by $e_d(\alpha)$.

3.5 Fluid versus Packet Models

DSS has been presented so far in the context of a pure fluid flow model. Recent studies [2–4] have emphasized the complexity of deriving bounds in a packet network due to the discrete nature of the problem. Problems arise when some flows are mixed in the same class of service. This is notably the case with DiffServ [14, 15]. Specifically, it has been shown in [4] that a deterministic bound may be derived only in the case of a low utilization ρ, with ρ inversely proportional to the maximum hop counts of each flow. This result was obtained in the case of leaky bucket constrained sources and relates more generally to the issue of stability of FIFO networks. Chlamtac et al [2] have derived some necessary conditions for the existence of a finite delay bound in a FIFO network with a general topology in the case of peak rate constrained sources. The authors show that if the peak rate of a source satisfies a constraint related to the number of sources that the source meets on its route, then bounds on end-to-end delays and backlogs exist.

With DSS, each source of the QoS class is treated as a constant bit rate source in the network since resources are reserved based on the peak rate of each source (i.e. their deterministic effective bandwidth). GPS is used in the network to redistribute unused bandwidth to the BE class. From the QoS provisioning point of view, we are in the most favorable case: admission control of constant bit rate sources. However, some caution needs to be taken when deploying DSS in a real network. We may have to limit the utilization of each link, the number of hops in each path or maybe the number of connection that mutually interfere. This is clearly a complex problem left for future study.

4 Related Work

GPS based approach: DSS uses GPS to *redistribute the unused bandwidth* to the best-effort sources. This is not the way GPS is commonly used. Indeed, GPS is mostly used to offer end-to-end delay bounds. The first results concerning the delay bounds achievable with GPS were derived by Parekh et al. (see Section 2). However, these bounds are restricted to the special case of RPPS. In [17], the authors have extended the result on delay bounds to all CRST (Consistent Relative Session Treatment) networks. Broadly speaking, a CRST network is a GPS network where the relative order between sessions does not vary throughout the network. A session has priority over another one at a given node if its associated weight is higher. CRST includes the case of RPPS and DSS that use the same weight for a source throughout the network. In [17], the authors provide closed-form bounds for arbitrary weight assignments that are compliant with the CRST condition. The key idea is to partition the set of sources in F subsets to form what is called a feasible partition, where the effective service rate received by the sources of subsets $i \in \{1, \ldots, F\}$ depends only on the sources in subset $j < i$. As a consequence, the delay bound obtained for each source will depend on the parameters and weight of the sources belonging to subsets with higher priority. While the result is clearly interesting from an analytical point of view, using it in practice is very complicated: if we were to derive an admission procedure from the delay bounds of [17], it would comprise at least the following operations (assume n sources $S_1, \ldots, S_n$ are already accepted and we try to admit S_{n+1}):

1. Determine the subset j to which S_{n+1} may belong (or create a new subset, depending on its parameters and weight).
2. Re-compute the delay bound associated to all sources (using the closed-form bounds) that belong to subsets $j+1, \ldots, F$, since S_{n+1} affects only the sources belonging to these subsets.
3. Check the non-violation of the delay constraint for each of the $n+1$ sources.

Clearly, such an admission control algorithm is not scalable due to the dependence that exists between the new source and all the sources over which it has priority. Note also that even with the simple RPPS policy, the admission control algorithm is not scalable since the new source affects the minimum service rate of every source whose route has at least one server in common with the route of the new source.

Bufferless multiplexing approach: We saw that QoS provisioning schemes purely based on GPS suffer from scalability problems. Still, the isolation property exhibited by GPS is very attractive. Based on this idea, Reisslein et al. [5, 16] introduced a new approach to define a traffic management scheme. The key idea is to shape the traffic prior to its entrance in the network. The shaping rate is chosen subject to the following constraint: minimizing the rate of the source in the network, or, equivalently, ensuring that the maximum delay in the shaper is as close as possible to the required maximum delay. A statistical bufferless multiplexing method is then used to maximize the number of admitted flows. The statistical multiplexing is done via an accurate upper bound on the loss rate experienced by the flows. It allows to accept twice as many sources than RPPS. Reisslein et al. introduced this scheme first for the single node network case [5]. However, to extend the bound on loss to the multiple node network case, sessions must be independent at each node. This is clearly an important limitation since this assumption does not hold in practice. Our approach is similar to this one but with an important difference: rather than maximizing the number of admitted flows, we focus on the redistribution of the unused bandwidth to the best-effort sources.

5 Evaluation of DSS

We have presented DSS in Section 3, emphasizing its scalability during admission control phase as well as during the data-transfer phase. We now evaluate its efficiency. First, we present evidence that DSS requires less bandwidth than RPPS in a network context. In a second stage, we provide some elements to evaluate the ability of DSS to redistribute the bandwidth unused by the QoS class to the BE class. This is an important point since the design goal is to keep our traffic management scalable while trying to provide as much bandwidth as possible to the best-effort traffic.

5.1 Comparison with RPPS

Consider a network with K servers and n sources. We do not make any assumption about the network topology or the routes of the sources in the network, except that the routes are loop-free and that the basic stability conditions are met at each node. For our comparison, we make the following assumptions about the sources:

- all the sources $S_i,\ i \in \{1, \ldots, n\}$ have the same leaky bucket parameters (p, R, M).
- the delay requirement d_i of a given source i is such that the deterministic effective bandwidth of source i is greater than its mean rate $R_i = R$. Using equation (6), we obtain that $d_i \in [0, M(\frac{1}{R} - \frac{1}{p})]$.

For a given server j, let n_j be the number of sources transiting this server. Let us also define two sets associated to server j:

– $I_j = \{i_1, i_2, \ldots, i_{n_j}\}$: set of sources transiting via server j.
– $I_j^{\min}$: set of sources of I_j for which server j is the one providing the minimum guaranteed rate in the RPPS network.

Theorem 2. *For any network compliant with the above assumptions, DSS requires less bandwidth than RPPS.*

Proof: we prove that, at each server, RPPS requires an amount of bandwidth strictly greater than the sum of the deterministic effective bandwidth of each source (which is the service rate required by DSS).
Let us first derive the minimum bandwidth C_j^{DSS} required with DSS at each server j. Let e_i be the deterministic effective bandwidth of S_i, for $i \in \{1, \ldots, n\}$. Using equation (6), we obtain: $e_i = \frac{M}{\frac{M}{p}+d_i}, \forall i \in \{1, \ldots, n\}$. Thus,

$$C_j^{\mathrm{DSS}} = \sum_{i \in I_j} \left(\frac{M}{\frac{M}{p} + d_i} \right) \tag{7}$$

Let us now derive the minimum bandwidth C_j^{RPPS} required with RPPS at server j. Consider a given source $S_i, i \in \{1, \ldots, n\}$. Let $j \in \{1, \ldots, K\}$ be the server, with service rate C_j, providing S_i with a minimum guaranteed service rate $r_i^{\min}$ (see Section 2). Since the weight associated to each source is the same ($\Phi_i = R, \forall i \in \{1, \ldots, n\}$), we obtain:

$$r_i^{\min} \overset{(2)}{=} \frac{R}{n_j \cdot R} C_j = \frac{C_j}{n_j} \tag{8}$$

The effective delay d_i^{eff} provided by RPPS to S_i is :

$$d_i^{\mathrm{eff}} \overset{(3)}{=} \frac{M}{r_i^{\min}} \overset{(8)}{=} \frac{M \cdot n_j}{C_j} \tag{9}$$

S_i is to be accepted if its delay requirement is less than the delay provided by RPPS, i.e. if:

$$d_i^{\mathrm{eff}} \leq d_i \tag{10}$$

Obviously, the minimum service rate C_j^{RPPS} is obtained when equation (10) is an equality. The service rate for server j must thus ensure that $d_i^{\mathrm{eff}} = d_i, \forall i \in I_j^{min}$. However, server j must guarantee to all sources from I_j (even if they do not belong to I_j^{min}) a delay smaller than their required delay. Otherwise, the RPPS network can't guarantee to each source of I_j the maximum delay it requires. Thus, the service rate for server j must ensure that $d_i^{\mathrm{eff}} = d_i, \forall i \in I_j$ and we obtain:

$$C_j^{\mathrm{RPPS}} \overset{(9)}{=} \max_{i \in I_j} \left(M \cdot \frac{n_j}{d_i} \right) = M \frac{n_j}{\min_{i \in I_j}(d_i)} \tag{11}$$

We thus obtain:

$$C_j^{\mathrm{RPPS}} = \underbrace{\frac{M}{\min_{i \in I_j} d_i} + \ldots + \frac{M}{\min_{i \in I_j} d_i}}_{n_j times}$$

$$\geq \frac{M}{d_{i_1}} + \frac{M}{d_{i_2}} \cdots \frac{M}{d_{i_{n_j}}}$$
$$> \frac{M}{\frac{M}{p} + d_{i_1}} + \frac{M}{\frac{M}{p} + d_{i_2}} \cdots \frac{M}{\frac{M}{p} + d_{i_{n_j}}} \stackrel{(7)}{=} C_j^{\mathrm{DSS}}$$

This proves the theorem. □

To further quantify how much better DSS performs compared to RPPS in terms of bandwidth consumption (in the context of Theorem 2), we now provide a numerical example. We consider a network with $K = 3$ servers and n sources. We make the following assumptions:

– the mean number of sources crossing each server is the same and is equal to $\bar{n} = 10$. We do not make any assumptions about the routes of the sources,
– the leaky bucket parameters of all the sources are the same, namely (p, R, M),
– the delay requirement d_i of source i is drawn randomly in the interval $]0, M(\frac{1}{R} - \frac{1}{p})[$ using a uniform law. The corresponding deterministic effective bandwidth is e_i. Let also D be the maximum possible value of d_i, i.e. $D = M(\frac{1}{R} - \frac{1}{p})$.

As previously, let n_j be the number of sources at server j and I_j the set of indices of these n_j sources. Let also C^{RPPS} (resp. C^{DSS}) be the sum of the minimum service rates at each of the k servers when RPPS (resp. DSS) is in use. Using equation (11), we obtain:

$$C^{\mathrm{RPPS}} = \sum_{j \in \{1,2,3\}} \frac{M \cdot n_j}{\min_{i \in I_j} d_i} \text{ and } C^{\mathrm{DSS}} = \sum_{j \in \{1,2,3\}} \sum_{i \in I_j} e_i$$

We are interested in the expectations $\overline{C^{\mathrm{RPPS}}} \stackrel{\triangle}{=} E(C^{\mathrm{RPPS}})$ and $\overline{C^{\mathrm{DSS}}} \stackrel{\triangle}{=} E(C^{\mathrm{DSS}})$ of C^{RPPS} and C^{DSS}. Due to the independence between the delay requirements and the number of sources, we obtain:

$$\overline{C^{\mathrm{RPPS}}} = M \cdot k \frac{\bar{n}}{\overline{\min_{i \in I_j} d_i}} \text{ and } \overline{C^{\mathrm{DSS}}} = k \cdot \bar{n} \frac{M}{\overline{d_i} + \frac{M}{p}}$$

Since delay requirements are drawn using a uniform law, we obtain: $\overline{d_i} = \frac{D}{2}$, $\overline{\min_{i \in I_j} d_i} = \frac{D}{k+1}$ (the proof is straightforward using the probability distribution function of the minimum of a set of variables). We are now able to compute $\overline{C^{\mathrm{RPPS}}}$ and $\overline{C^{\mathrm{DSS}}}$ for some values of the leaky bucket parameters of the sources. In Table 1, we present such results for different configurations. These results clearly show that the bandwidth requirement of DSS can be significantly smaller than the one of RPPS.

5.2 Bandwidth Re-distribution

DSS performs a peak rate allocation for the QoS class. However, these sources are not, even after shaping, constant bit rate sources. Using GPS allows to perform a peak rate allocation while redistributing the bandwidth unused by the QoS class. Evaluating this redistribution mechanism is not easy since it is difficult to make hypotheses about the BE class. In the following, we carry out one experiment to illustrate how the bandwidth distribution is performed. We use as criteria:

(p,R,M)	$\overline{C}^{\text{RPPS}}$	$\overline{C}^{\text{DSS}}$
(100,10,10)	1333.33	545.45
(100,1,10)	121.21	59.40
(100,50,1000)	12000.00	2000.00

Table 1. Minimum service rates for RPPS and DSS

- $E(S)$: the mean service rate of the BE class,
- $Min(S)$: the minimum service rate of the BE class.

We consider a single server with a service rate C. We assume that DSS is used and that the QoS class comprises n sources $(S_i)_{i\in\{1,\ldots,n\}}$. The traffic descriptor of S_i is (p_i, R_i, M_i). Let also e_i be the deterministic effective bandwidth of S_i. The stability of the DSS system ensures that $E(S) = C - \sum_{i\in\{1,\ldots,n\}} R_i$. Moreover, since the BE class is granted a weight $\Phi_{BE} = C - \sum_{i\in\{1,\ldots,n\}} e_i$, it is guaranteed a minimum service rate $Min(S) = \frac{\Phi_{BE}}{\Phi_{BE}+\Phi_{QoS}} C = C - \sum_{i\in\{1,\ldots,n\}} e_i$. These two criteria are interesting since they capture two time scales: a short-term one with $Min(S)$ and a long-term one with $E(S)$. However, it would be interesting to characterize more accurately the dynamics of the system and to evaluate this dynamics for a "worst-case" situation. Such a situation can be the one where the worst-case delay is achieved with a GPS server. It is proven in [7] that the worst-case delays are achieved when all the sources are greedy and synchronous. Going back to our server with service rate C and n sources in the QoS class, we are going to evaluate the service rate that is effectively received by the best-effort class. The effective cumulative service provided by the server corresponds to its service curve offered $\beta(t) = C\cdot t$ since GPS is work-conserving (see Section 3.2) and also since the BE class is assumed to consume all the capacity unused by the QoS class. The cumulative bit rate of the traffic issued by the QoS class corresponds to the arrival curve α characterizing the n sources of the QoS class when they are synchronized. The latter is simply the sum of the individual arrival curves of each source given by equation (4). Thus, one obtains:

$$\alpha(t) = \sum_{i\in\{1,\ldots,n\}} \min\left(p_i \cdot t, R_i \cdot t + M_i \frac{p_i - R_i}{p_i}\right)$$

$\alpha(t)$ characterizes the maximum service that the QoS class can require during the worst-case situation. Since $\beta(t)$ characterizes the service provided to both classes (BE and QoS), we deduce that the cumulative service effectively received by the BE class is the $\beta(t) - \alpha(t)$. The latter is represented as the shaded area of Figure 5. A change in the slope of $\alpha(t)$ corresponds to a time instant when a bucket of one source gets empty. The corresponding source, which was previously emitting at its peak rate, now emits at its mean rate. This is why the slope of $\alpha(t)$ decreases with time. When the last bucket gets empty, the slope of $\alpha(t)$ is the sum of the mean rates of all sources. Since the basic stability conditions are met, the vertical distance between $\alpha(t)$ and $\beta(t)$, i.e. $\beta(t)-\alpha(t)$ goes to infinity when t increases. To further illustrate the dynamics of DSS, we now provide a numerical example with a single server with service rate C and $n = 40$ sources

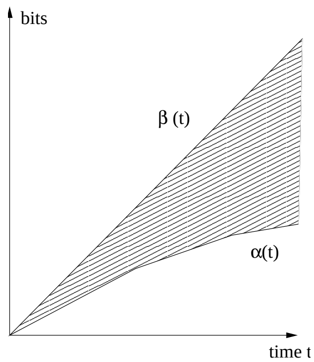

Fig. 5. Cumulative service rate received by the BE class during the “worst-case” period

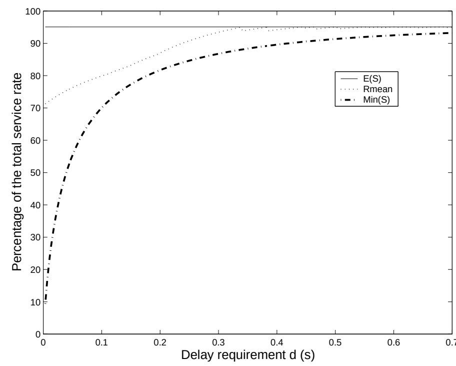

Fig. 6. Mean service rate received during the “worst-case” scenario

in the QoS class. For a given source with parameters (p_i, R_i, M_i), we draw the traffic parameters using a uniform law, as follows: $p_i \in [200, 1000]$ (kbits/s), $R_i \in [10, 50]$ (kbits/s) and $M_i \in [10, 50]$ (kbits). These parameters have been chosen so that the total mean rate of the QoS class represents a low fraction of the service rate C (about 5% of the service rate) while the total peak rate of the QoS class is close to C. The sources from the QoS class have thus a high variability in terms of bandwidth requirement and we expect that the use of GPS will enable the BE class to receive an important fraction of the unused bandwidth. We assume that the n sources require the same delay bound d and make d vary within an interval $[0, d_{\max}]$. For a given set of sources with parameters $(p_i, R_i, M_i)_{i\in\{1,\ldots,n\}}$, we set: $d_{max} = \max_{i\in\{1,\ldots,n\}} \left(M_i \big(\frac{1}{R_i} - \frac{1}{p_i}\big) \right)$. This choice of $d_{\max}$ ensures that at least one source has a deterministic effective bandwidth strictly greater than its mean rate. Otherwise, if one chooses a value greater than $d_{\max}$, all the sources are shaped at their mean rate and the results are of little interest. We consider the “worst-case” situation presented above and we evaluate the mean percentage R_{mean} of the service rate C received by the BE class during this period. We have to fix a length T for this period. Since all the sources are greedy and synchronous, we choose T as the minimum time it takes to empty the backlogs of all the sources.

The result obtained for a given experiment is presented on Figure 6. All the values are expressed as percentages of the total bandwidth C, which remains unchanged during all the experiments and is chosen to be greater than the sum of all the deterministic effective bandwidths (a requirement of DSS). The upper curve represents the maximum service that may be received by the BE class. This maximum service is equal to $C - \sum_{i\in\{1,\ldots,n\}} R_i$, i.e. $E(S)$, since during the “worst-case” situation, the sources of the QoS class emit at least at their mean rate. The lower curve represents the minimum service rate $Min(S)$ that may be received by the BE class, i.e. $C - \sum_{i\in\{1,\ldots,n\}} e_i$. The latter varies with the delay requirement d since the deterministic effective bandwidths are functions of d. The intermediate curve represents the mean service rate R_{mean} received by the BE class during T. It is interesting to note that even when the delay requirement is low, the mean service received is high, above 70%.

The previous scenario is a realistic one since the burstiness of the sources is high (considering the ratio $\frac{p}{R}$ to characterize the burstiness). Even if it applies only to a single trajectory, the experiment clearly shows that even during a typical “worst-case” scenario, the percentage of service received by the best-effort class may be high.

From this experiment, we can conclude that there is strong evidence that DSS, by using GPS, efficiently redistributes the bandwidth unused by the QoS class.

6 Conclusion

Recent traffic management schemes that aim at providing QoS to multimedia applications rely on complex mechanisms requiring an individual treatment of each source or based on a complex statistical multiplexing of the sources. The main objective of these traffic management schemes is to maintain the global resource requirement at a reasonable level. However, minimizing the total bandwidth used by the multimedia applications often results in a poor scalability. Things worsen as non-scalability may occur during the data-transfer phase, if each source must be treated individually, or during the admission control phase, if the admission of a new source affects many of the already established sessions.

In order to avoid these scalability problems, we propose a new approach. The key idea is not to minimize the bandwidth consumption of the multimedia sources but rather to redistribute efficiently the bandwidth unused by the multimedia applications. To limit the greediness of the multimedia applications, we first limit as much as possible their peak rate before they enter the network. This is achieved by an adequate shaping of the sources. We then use GPS inside the network, which allows to offer to the multimedia application the bandwidth they need during bursty periods and to redistribute as much bandwidth as possible to the best-effort traffic during periods when the multimedia applications are less active. The resulting traffic management scheme, called DSS, is fully scalable during the data-transfer phase and the admission control phase. We also provide evidence that this scalability is not obtained at the expense of a too important bandwidth consumption: (i) DSS requires less bandwidth than RPPS, the most popular version of GPS and (ii) with DSS, the best-effort traffic is always granted an important amount of bandwidth, even during periods of high activity of the multimedia sessions.

Future work should address a possible formulation of DSS in a statistical setting, which would help to admit more QoS sessions. However, the challenge is to keep the scheme scalable. Making use of a statistical multiplexing seems difficult (see [16]). The notion of statistical effective bandwidth, with a scheme remaining deterministic in the core of network, could be a first step in this direction.

References

1. Zhang, H.: Service discipline for guaranteed performance service in packet-switching networks. Proceedings of the IEEE **83** (1995) 1374 –1396
2. Chlamtac, I., Faragó, A., Zhang, H.: A deterministic approach to the end-to-end analysis of packet flows in connection-oriented networks. IEEE Transactions on Networking (1998) 422–431

3. Charny, A., Le Boudec, J.Y.: Delay bounds in a network with aggregate scheduling. In: Quality of Future Internet Services. Volume 1922 of Lecture Notes in Computer Science., Berlin, Germany, Springer (2000) 1–13
4. Bennett, J., Benson, K., Charny, A., Courtney, W.F., Le Boudec, J.Y.: Delay jitter and packet scale rate guarantee for expedited forwarding. In: Proceedings of IEEE Infocom, Anchorage, Alaska (2001)
5. Reisslein, M., Ross, K., Rajagopal, S.: Guaranteeing statistical QoS to regulated traffic: The single node case. In: Proceedings of IEEE Infocom, New York (1999)
6. Parekh, A.K., Gallager, R.G.: A generalized processor sharing approach to flow control in integrated services networks: The multiple node case. IEEE/ACM Transactions on Networking **2** (1994) 137–150
7. Parekh, A.K., Gallager, R.G.: A generalized processor sharing approach to flow control in integrated services networks: The single node case. IEEE/ACM Transactions on Networking **1** (1993) 344–357
8. Golestani, S.J.: A self-clocked fair queueing scheme for broadband applications. In: Proc. of IEEE INFOCOM'94, Toronto, Canada (1994) 636–646
9. Bennett, J.C.R., Zhang, H.: WF2Q: worst-case fair weighted fair queueing. In: Proceedings of IEEE Infocom, San Fransisco, California (1996) 120–128
10. Shreedhar, M., Varghese, G.: Efficient fair queueing using deficit round robin. ACM Computer Communication Review **25** (1995) 231–242
11. Le Boudec, J.Y: An application of network calculus to guaranteed service networks. IEEE Transactions on Information Theory **44** (1998)
12. Cruz, R.L.: Quality of service guarantees in virtual circuit switched networks. IEEE Journal on Selected Areas in Communications **13** (1995) 1048–1056
13. Boyer, P., Guillemin, F., Servel, M.: The spacer-controller : An efficient upc/npc for atm networks. In: ISS'92. (1992)
14. Blake, S., Black, D., Carlson, M., Davies, E., Z. Wang, Weiss, W.: An architecture for differentiated services. IETF RFC 2475 (1998)
15. Bennett, J., Benson, K., Charny, A., Courtney, W.F., Le Boudec, J.Y., Shiu, A., Kalmanek, C., Stiliadis, D., Fioriru, V., Ramakrishnam, K.K.: An expedited forwarding phb. (Technical Report draft-ietf-diffserv-rfc2598bis-00.txt)
16. Reisslein, M., Ross, K., Rajagopal, S.: Guaranteeing statistical QoS to regulated traffic: The multiple node case. In: 37th IEEE Conference on Decision and Control (CDC), Tampa, Florida (1998)
17. Zhang, Z.L., Liu, Z., Towsley, D.: Closed-form deterministic performance bounds for the generalized processor sharing scheduling discipline. Journal of Combinatorial Optimization, Special Issue on Scheduling **1** (1998)