12

Enforcing quality of service for adaptive multimedia applications via fair queuing

F. Toutain

ENST Bretagne, Département Réseaux et Services Multimédia
2, rue de la Châtaigneraie, BP 78
35512 Cesson-Sévigné Cedex, FRANCE

tel (+33) 2 99 12 70 41
fax (+33) 2 99 12 70 30
ftoutain@rennes.enst-bretagne.fr

Adaptive applications are more and more perceived as a cornerstone of future integrated services packet networks. However, for these applications to be widely deployed, we advocate that they should be guaranteed some quality of service, in terms of floor rate, delay, and resource sharing equity. In this paper we propose an integrated approach that deals with these issues by means of a novel Generalized Processor Sharing -like scheduling scheme. Moreover we show that our proposal has the same implementation complexity as the Self Clocked Fair Queuing one, and that it provides the optimal delay bound for packet GPS-like schedulers.

Keywords: Adaptive applications, QoS, Fair queuing

1 INTRODUCTION

Adaptive multimedia applications have been more and more considered for the past few years. These are applications being able to modify their throughput in response to changing network conditions. They adapt to congestion by reducing the amount of traffic they put in the network, and conversely react to available bandwidth by increasing it, much in the same way as TCP does in the Internet. These variations obviously impact the quality of service delivered to the user, in terms of audio or video resolution, frame rate, frame size, color depth, or end-to-end delay, depending on the adaptation scheme that is used [10, 14]. Adaptive applications traditionally target so-called best-effort networks such as the Internet, where they are considered

A. Tantawy (ed.), *High Performance Networking VII*

as an alternative to deploying QoS scheduling and resource reservation schemes (e.g., in [5]). They will also be useful in future integrated services networks, not because these networks are envisioned to provide a best-effort service class, but as a means to deal with resource scarcity. Indeed, there is anecdotal evidence from computer science history that resources always end up to be scarce (RAM, peak CPU power, hard disks, ...). And despite the successful dimensioning of telephone networks, there is no evidence that integrated services networks will have so high a bandwidth and so predictable a traffic pattern that they will never be congested [3, 8, 24, 28]. This is why, for instance, service classes dedicated to adaptive applications are envisioned by the IETF for the future integrated services Internet [31]. Furthermore, it is worth noticing, as in [13], that adaptive applications may be used in a guaranteed QoS context, when it is available, and as such outperform non-adaptive ones, in terms of flexibility and usability. Hence adaptive multimedia applications design and support tend to become major trends in the multimedia networking research community. However we identify three major issues that must be tackled for adaptive applications to be successful: floor bandwidth, delay, and equity. These issues are detailed below.

The first issue is linked to common-sense deductions, and has been already described into several publications (e.g., [3, 6, 7, 12]). For short, it is: continuous media streams cannot be degraded below some threshold, or floor rate. Indeed, it is well-known that audio compression below a few hundreds bits per second produces barely understandable results. For video, different bounds, depending on the compression method as well as the application itself, could be roughly given. For instance consider that a multi-layered video stream can be easily degraded by dropping some layers, but that its baseband must not be dropped. The point here is that each media stream has a minimum bandwidth requirement, which must be fulfilled for the content to be useful. This is in contrast with traditional data communications using TCP, which may reduce to several bits per second while still reaching adequate quality of service (that is, data integrity). Furthermore, different users will have different sensitivities to the noise introduced by degrading the media content. Consider for instance people listening to a foreign language audio stream, as compared to people hearing their native language. Similarly, depending on the user itself and also on the communication purpose, the same generic video tool could place different requirements on the floor bandwidth to be guaranteed: a private user saying hello to family would probably tolerate more degradations than a professional user wishing to examine some complex diagrams. Consequently, it turns out that almost any adaptive application should be given rate guarantees for its minimum floor bandwidth needs, which should be freely selectable.

The second issue is that of end-to-end delay. A yet unknown amount of multimedia applications are targeted towards interpersonal communications (phone, videophone, teleconferencing, teleteaching, ...). As such they require *interactive* communications, that is, end-to-end delay on the order of a hundred milliseconds

[15]. Media synchronization is even more stringent. In addition, several future applications, among which are distributed simulations, distributed games, cooperative teleworking, distributed whiteboards, ..., will be real-time applications, with very tight delay constraints [13]. Even playback applications benefit from delay bounds: (1) in order to limit their playout buffer size so as to save memory, hence production costs (set top boxes); (2) to have reasonable response times (stop, fast-forward,...). All these applications can be adaptive but they will not adapt much in the delay domain. Hence all these applications require specific guaranteed delay bounds [24].

The third issue does not address applications themselves, but rather application competition. Several studies have shown that different adaptive applications, when competing for network (or end-system) resources, do not converge to a fair equilibrium (e.g., [9, 16]). Rather, the equilibrium tends to be appealing only for the most aggressive application, since it gets the whole bandwidth while the others are doomed to service interruption. This is a major issue because the simplest way for an application to be aggressive is to not adapt to a congestion (i.e. to be greedy). Hence, unless fairness is enforced, adaptive applications are unable (and not expected to be able) to behave as "good citizens" and reach a fair equilibrium in the network sharing process, both with each other, and in the presence of non-adaptive applications. Notice that the same is true for traditional computer communications, and imposes to rely on a single congestion control algorithm (e.g. TCP in the Internet). However imposing a unique adaptation scheme for all adaptive applications is unrealistic, as it would severely affect the deployment of such applications [29]. Furthermore, even with fairness enforcement mechanisms, the issue that remains is to define fairness. The equity we will consider throughout this paper has a common-sense definition: we require every application to be affected in the same proportion by network conditions. Hence, if for instance the aggregated rates of all sessions sharing a link exceed the link capacity by 10%, then every session will have to lower its rate by 10%. This definition is indeed proportional fairness, as described in [23], where it is shown to reach optimal network efficiency provided that (1) users declare their true needs, and (2) the network allocates bandwidth in proportion to these needs. Also notice that GPS schedulers enforce proportional fairness (due to lack of space, we will not describe much the GPS approach; see [2, 11, 19, 26]).

The three issues described above have been previously treated in isolation. We will show in section 2 that these solutions cannot be combined to solve the three problems, and we will propose an integrated approach that provides floor bandwidth and delay guarantees, and enforces a fair competition between adaptive (and also non-adaptive) multimedia applications. Section 3 and 4 respectively relate to the fluid-flow (i.e. ideal) and packetized (i.e. realistic) scheduling models that we propose. Implementation is addressed by section 5, along with some delay bound results, and concluding remarks are given in section 6.

2 PREVIOUS APPROACHES

In this section, we review different proposals that were aimed at solving the floor rate, guaranteed delay, and equity issues. Since no integrated approach has been proposed yet, to our knowledge, we examine the three issues one after the other in the following subsections, and we show that they cannot be mixed into an integrated approach.

2.1 Guaranteed Delay

It is well known that a single FIFO server, shared by several sessions, provides a guaranteed bound on the queuing delay. However in this paper we envision applications having different delay constraints, and as such, requiring individual delay bounds. Furthermore, FIFO scheduling does not enforce fairness, nor does it provide rate guarantees. Individual delay bounds can be obtained by using fair queuing strategies. Depending on the buffer sizes and reserved rates, specific delay bounds can be achieved. But we will show in subsection 2.3 that using standard GPS-like queuing service raises the fairness issue.

2.2 Floor Bandwidth

Once adaptive application needs for floor bandwidth guarantees are recognized, it becomes obvious that pure best-effort networks cannot actually support these applications. This is extensively described by Lefelhocz, Lyles, Shenker and Zhang in [24], and also exemplified in [3, 28], who independently state that admission control mechanisms must be used in order to exercise congestion avoidance. Indeed, the use of guaranteed QoS schemes is advocated in these studies, along with reservation protocols which allow admission control to be done inside each node along the session path. The idea is then to make use of guaranteed service classes for the application floor rate, and to send all traffic above this minimum inside a best-effort class, where resources can be dynamically shared by several applications. Then, as the node becomes loaded, the amount of resources dedicated to the best-effort class decreases, which causes adaptive applications to lower their rates. While allowing adaptive applications to get their floor bandwidth needs, this approach has a number of drawbacks. First, it forces the applications to open two sessions inside the network: one for the guaranteed traffic, the other for the best-effort traffic. This involves additional processing to demultiplex / multiplex the original stream into end-systems. This potentially doubles the number of session contexts to be handled by the network nodes. This also requires that the two sessions be routed as a whole. Next, fairness needs to be enforced inside the best-effort service class. This imposes additional mechanisms, such as Random Early Detection (see [17]). But the main limitation is that, whereas the guaranteed part of the application traffic will have bounded delay (any value is possible, depending on the resource reservation), the best-effort part will not (unless it is a bufferless class, which is rather unrealistic). As a consequence, it may well be the case that, for some applications, the best-effort

traffic reaches the end-system too late to be used. In other words, despite fairness enforcement, the application would not get its fair share of the network.

2.3 Equity

The fairness enforced by GPS-like queuing policies is a proportional one. Here we consider using such a scheduling policy in the context of adaptive applications. A work-conserving GPS server is defined as follows. Let N sessions be characterized by positive real numbers $\phi_1, \phi_2, \ldots, \phi_N$. Let $W_i(s, t)$ be the amount of session i traffic served in the time interval $[s, t]$. The inequality

$$\frac{W_i(s,t)}{W_j(s,t)} \geq \frac{\phi_i}{\phi_j} \qquad j = 1, 2, \ldots, N, \tag{1}$$

holding for any session i that is backlogged throughout $[s, t]$, defines the GPS server. A session is said to be backlogged if it has at least one data unit awaiting service or being serviced. It follows from (1) that, for any two sessions i, j that are backlogged throughout $[s, t]$,

$$\frac{W_i(s,t)}{\phi_i} = \frac{W_j(s,t)}{\phi_j}, \tag{2}$$

that is, the server distributes bandwidth to all backlogged sessions in proportion to their ϕ_i, which are thereby called service shares or weights.

Adaptive applications are characterized by their floor rate, delay bound requirement, and also by their maximum rate, which corresponds to the highest quality that can be reached by the application. The maximum rate is meant to depict the user's needs. Either floor rate or maximum rate can be given as the weight that is used by the GPS-like scheduler (i.e. the ϕ_i values). If the floor rate value is used, then, assuming that an admission control procedure is exercised, the floor rate requirements can be met, and, consequently, the delay bounds can also be. But then, the proportional fairness is defined over the different floor values. Based on a very simple example, we show that this kind of fairness is problematic: Consider that a link capacity $C = 10$ is shared by two adaptive applications A_1 and A_2 with floor bandwidth requirements $f_1 = f_2 = 1$, and maximum rates $m_1 = 5$ and $m_2 = 10$. Based on the floor rate proportions (i.e. 1:1), the capacity is split in two equal parts $c_1 = c_2 = 5$. Hence the first application gets its maximum rate whereas the second one must adapt to a 50% degradation. It is clear from this example that applications are not affected by congestion in the same proportion. If however we decide to give the maximum rates for the scheduling values (and hence define fairness over these values), then the allocated capacities become $c_1 = 10 / 3$ and $c_2 = 20 / 3$. It turns out that each application must adapt to a 33.33% degradation, which conforms to our idea of fairness, as detailed in the introduction. Such a fairness ensures that each application takes its part of a congestion, so that no one is forced to zero before the

others when available bandwidth is reduced. Unfortunately, the guaranteed rate for any session i is then given by:

$$x_i \geq \frac{m_i}{\sum_N m_j} \cdot C, \tag{3}$$

that is, it relates to the maximum rates of all sessions sharing the node. Hence it is not possible for an application to freely choose its floor rate, and consequently its delay bound. The trouble is that minimum requirement (that is, floor rate) and maximum requirement (maximum rate) cannot be independently stated using GPS. Another way to present this is the following: split the total bandwidth into a guaranteed part (for floor rates) and a spare part. The guaranteed part allocation is to give each session exactly its floor rate. The spare part allocation must be in proportion to users' needs. With GPS:

- Either use the floor rate to depict users' needs. In such a case, one cannot both request a low floor rate and a high need for available bandwidth. In the limiting case, a session requesting a null floor rate will get zero additional bandwidth. It turns out that such a scheme has a bias against highly adaptive sessions.
- Or use the maximum rate to depict the floor rate. Then the floor rate depends on the other sessions' maximum rates, so it cannot be individually stated. Also consider that adding admission control based on the maximum rates to gain control on the floor rates turns us back to the first case.

In the remaining of this paper, we propose a new scheduling strategy, which relies heavily on GPS, but makes it possible to independently reserve a floor rate (hence to get any desired delay bound) and still to achieve the maximum rate fairness.

3 FLUID-FLOW MODEL

Our purpose is to ensure that any session is served above its floor rate, provided that adequate admission control is exercised, while simultaneously allowing all sessions to share available capacity with respect to their maximum rates. Once this is achieved, independent delay bounds can be obtained by requiring a specific amount of buffer. In this section we focus on the ideal model by making the fluid-flow assumption. We start by introducing some notations. For all sessions k, $k = 1, 2,..., N$, we denote by f_k the floor rates that are to be guaranteed and by m_k the maximum rates over which fairness is defined. An issue that arises is to choose whether fairness must be defined over m_k or over $m_k - f_k$, given that f_k is already guaranteed. From a practical standpoint, we consider defining fairness over m_k, and notice that the other option can be achieved merely by replacing m_k with $m_k - f_k$.. $W_k(s, t)$

denotes the amount of traffic from session k served in the time interval $[s, t]$. The desired fairness is thus depicted by:

$$\frac{W_i(s,t)-(t-s)f_i}{W_j(s,t)-(t-s)f_j} \geq \frac{m_i}{m_j}, \tag{4}$$

holding for any session i that is backlogged throughout $[s, t]$. Summing (4) over all sessions j, we get

$$W_i(s,t)-(t-s)f_i \geq \frac{m_i}{\sum_{j=1}^{N} m_j}(t-s)[C-\sum_{j=1}^{N} f_j]. \tag{5}$$

Hence the rate allocated to session i at time t is:

$$x_i(t) = f_i + \frac{m_i}{\sum_{j \in \mathcal{B}(\tau)} m_j}\left[C - \sum_{j \in \mathcal{B}(\tau)} f_j\right] \geq x_i = f_i + \frac{m_i}{\sum_{j=1}^{N} m_j}\left[C - \sum_{j=1}^{N} f_j\right]. \tag{6}$$

The right hand part of (6), namely x_i, depicts the minimum service rate, which is guaranteed regardless of the behavior of other sessions. That is, the isolation property remains valid for this fluid-flow model. More, it is easily shown that using

$$\sum_{j=1}^{N} f_j \leq C \tag{7}$$

for admission control leads to an absolute guarantee of the floor rate. Such a guarantee allows for a guaranteed delay bound, provided that an adequate amount of buffer is dedicated to the session. Because of the isolation property, it turns out that this scheme can serve a variety of applications having different needs. For instance, a non-adaptive application requiring real-time service needs only set its floor rate to the bandwidth it needs, and request a null maximum rate, thereby reserving bandwidth for its whole stream. In contrast, an application being able to adapt to extreme congestion with no time constraint may indicate a null floor rate, hence only use whatever bandwidth is available. Between these two extremes, many combinations of guaranteed and shared bandwidth can be requested by applications requiring both. The isolation property ensures that any session gets its fair share, be it adaptive or not. As for the delay guarantee, it may be remarked that, similarly to GPS, sessions are guaranteed a worst-case delay provided their traffics conform to the allocated rates. This implicitly assumes that losses are not tolerated. But since adaptive applications, by nature, tend to shape their traffics to the available rate, so as to avoid losses, they actually tend to conform to their dynamically allocated rates, thereby getting delay guarantees. Moreover, by using segregated buffers, adaptive as well as non-adaptive applications incur losses depending only on their own behavior (such losses can be further corrected using *Forward Error Correction* techniques as applied to continuous data streams [1, 4]). Thus, following [15], we consider that apart from the lost packets these applications actually have guaranteed

delay bounds. We next focus on the actual fairness embedded in the model. Define $\rho(\tau)$ to be the *congestion ratio* (or theoretical loss rate) for the dynamically shared part of the bandwidth at time τ:

$$\rho(\tau) = 1-(C-\sum_{j \in \mathcal{B}(\tau)} f_j)(\sum_{j \in \mathcal{B}(\tau)} m_j)^{-1}. \tag{8}$$

Let the incoming traffic at time τ be:

$$\sum_{j \in \mathcal{B}(\tau)} m_j = K(C-\sum_{j \in \mathcal{B}(\tau)} f_j) \qquad K \geq 0. \tag{9}$$

A K greater than one depicts a congestion, and a K less than one indicates underutilization. From (6), we have, for any session i

$$x_i(\tau) = f_i + \frac{m_i}{K}, \tag{10}$$

hence

$$x_i(\tau) = f_i + m_i(1-\rho(\tau)). \tag{11}$$

Equation (11) shows that any session is required to adapt its incoming traffic in the same proportion $\rho(\tau)$. This conforms to the envisioned fairness. Hence it turns out that such a scheduling model simultaneously answers the three issues that are raised by adaptive applications.

4 PACKET-BY-PACKET SCHEDULER

In this section, we consider actual packet networks, the nodes of which serve one packet at a time in a non-preemptable way. The beginning of our approach is almost identical to the packetized GPS scheme, except for the definition of the normalized service $w_k(s, t)$ received by a session k constantly backlogged during $[s, t]$, which comes from equation (6) above:

$$w_k(s, t) = \frac{W_k(s, t)}{x_k}, \tag{12}$$

where

$$x_k = f_k + \frac{m_k}{\sum_{j \in B(s,t)} m_j}\left[C - \sum_{j \in B(s,t)} f_j\right]. \tag{13}$$

We may define a virtual time function $v(t)$ representing the progress of work in the server, by:

$$v(t)-v(s) = w_k(s, t) \qquad \forall k \in B(s, t). \tag{14}$$

Multiplying each side of (14) by x_k and summing up over backlogged sessions k:

$$v(t)-v(s)\sum_{k \in B(s,t)} x_k = \sum_{k \in B(s,t)} x_k w_k(s,t) , \tag{15}$$

hence

$$v(t)-v(s) = \frac{C(t-s)}{\sum_{k \in B(s,t)} x_k} . \tag{16}$$

By computing that

$$\sum_{k \in B(s,t)} x_k = C , \tag{17}$$

we conclude that our virtual time function is indeed the real time.

Now, focusing on the i'th packet from session k, i.e. p_k^i, we notice that the beginning of its service is

$$b_k^i = \max(a_k^i, d_k^{i-1}) . \tag{18}$$

The finishing time of p_k^i is thus

$$d_k^i = \frac{L_k^i}{x_k} + \max(a_k^i, d_k^{i-1}) . \tag{19}$$

Equation (19), which depicts the iterative algorithm for computing finishing times (i.e. tags) of successive packets from session k, is indeed very close to Zhang's *Virtual Clock* scheme [32]. However in our case, the x_k values are not constant over time, but depend on the set $\mathcal{B}(t)$ of backlogged sessions at time t. Hence, the issue that remains is that of computing the set $\mathcal{B}(t)$, so as to compute the values for $\sum_{k \in \mathcal{B}(t)} f_k$ and $\sum_{k \in \mathcal{B}(t)} m_k$ at the time of tagging a new packet. We could just argue that the virtual time function of the *Packetized Generalized Processor Sharing* (PGPS, [26]) scheme also involves such a computation, hence our proposal requires very few additional work, and has the same complexity. However, some of the most important contributions to the fair queuing area were aimed at removing this computation, because it is considered too complex to be actually implemented (see for instance [20]). For this reason, we will discuss in the next section, which deals with implementation, one possible algorithm that reduces this complexity.

5 IMPLEMENTATION

In the following, we focus on the multi-queue implementation of GPS-like schedulers, i.e. each session traffic is stored inside a FIFO queue, and tags are computed only for the first packet of each backlogged session. As a side note, consider that the multi-queue version is advocated in several papers dealing with hardware implementation of the GPS scheme (e.g. [18, 27]). As exemplified in [19], determining the set of backlogged sessions at any time may be computationally intensive,

because, by the time a single packet is served, all sessions may potentially become backlogged or cease to be backlogged. For this reason, we restrict to approximate knowledge which allows for a greedy approach.

Let F and M be approximated values for $\sum_{k \in \mathcal{B}(t)} f_k$ and $\sum_{k \in \mathcal{B}(t)} m_k$, respectively. Each time a packet from session k is picked up for service, let

$$\begin{aligned} F &\leftarrow F - f_k \\ M &\leftarrow M - m_k \end{aligned}.$$

Similarly, each time a packet becomes first inside session k's FIFO queue (either because the previous packet has been picked up for service, or because the session is newly backlogged), let

$$\begin{aligned} F &\leftarrow F + f_k \\ M &\leftarrow M + m_k \end{aligned}.$$

before computing its tag. It immediately follows that each time a session ceases to be backlogged its floor rate and maximum rate values are removed from F and M, respectively. Conversely, these values remain taken into account for a backlogged session. Nevertheless, a trouble may arise as a consequence of such a greedy approach. Consider the packet p_k^i from session k, being tagged at time s and picked up for service at time t. During $[s, t]$, several sessions may become backlogged or cease to be backlogged. Such changes will not be reflected by the tag of p_k^i. Hence the order in which packets depart the server may be different to the order in which they would have been served by the fluid-flow server (in the following, we call this phenomenon an *inversion problem*). Consequently we may wonder if the packet receives proper service. We answer this question below by assessing the worst-case delay bound of our scheme, with respect to the theoretical (i.e. fluid-flow) model. We first focus on a similar, albeit simpler, tag computation algorithm, in which tags are iteratively given by:

$$d_k^i = \frac{L_k^i}{\phi_k \left(\sum_{k \in \mathcal{B}(t)} \phi_k\right)^{-1} C} + \max(a_k^i, d_k^{i-1}), \tag{20}$$

where ϕ_k is the weight of session k in the GPS definition, and where the summation over $\mathcal{B}(t)$ is approximated by the greedy method above. This is actually an approximation of GPS. We now conduct a worst-case study for the delay bound of this scheme. Consider $N + 1$ sessions, numbered 1, 2,..., N+1, so that

$$\sum_{k=1}^{N+1} \phi_k = C. \tag{21}$$

holds. Since we are looking for a worst-case delay bound, we must and will hereaf-

ter consider that all sessions are backlogged. Initially, only session 1 is backlogged and has a packet in service. During the service of this single packet, we make the worst-case assumption that all other sessions become backlogged, and we focus on the delay experienced by session N+1, when its packet is picked up last, contrary to the fluid-flow model in which it would be served first (i.e. an inversion problem happens). Indeed, we actually look for the maximum difference between the actual delay and the theoretical delay, as given by the fluid-flow model. Without loss of generality, we make the following simplifications: All sessions i belonging to 1,..., N have the same weight ϕ_i. All packet sizes are equal to the maximum packet size L_{max}. We also take $C = L_{max}$ to clarify the computations. We start at time $t = 0$, and all sessions in 2,..., N+1 become backlogged one after the other, in order. The service of the packet from session 1 has a finishing time equal to 1. Hence the tag for session 1 (i.e. for the packet in its FIFO queue) is $T_1 = 1$ (for its service time) + max(1, 0) = 2. When session 2 becomes backlogged, its tag is also 2, but now because the weight summation (as approximated by the greedy approach) equals 2. The same applies to any session i up to N, receiving a tag $T_i = i$. Now, considering session N+1, we look for its weight such that an inversion problem arises. For this to happen, its tag must be greater or equal to the highest tag already computed in the system, that is:

$$T_{N+1} \geq N\,, \tag{22}$$

hence (from 20)

$$\frac{L_{max}}{\phi_{N+1}(N\phi_i + \phi_{N+1})^{-1}C} + \max(0, 0) \geq N\,. \tag{23}$$

From (21), since all sessions are backlogged

$$\phi_{N+1} \leq \frac{C}{N}\,. \tag{24}$$

We select the highest possible value for ϕ_{N+1} so as to minimize the fluid-flow delay (thereby maximizing its difference with the actual delay). Then, the fluid-flow delay it experiences is:

$$D_{GPS} = \frac{L_{max}}{\phi_{N+1}} = N\frac{L_{max}}{C}\,. \tag{25}$$

The actual delay it experiences is the time it takes to serve one packet from all sessions, since session N+1 is served last because of the inversion problem. Hence

$$D_{actual} = (N+1)\frac{L_{max}}{C}\,. \tag{26}$$

It turns out that the difference between fluid-flow delay and actual delay is the time it takes to serve one packet of maximum size at the server rate. Since this is a worst-case value, in the general case our implementation ensures the following delay bound, for the i'th packet of any session k:

$$d_k^i - a_k^i \leq \frac{L_k^i}{\phi_k} + \frac{L_{max}}{C} . \tag{27}$$

Equation (27) is well known to depict the optimal delay bound for packet-by-packet GPS-like schedulers [26]. For instance it is achieved by the PGPS scheme introduced in [26], which has, however, an overall complexity of O(N). Conversely, the *Self Clocked Fair Queuing* proposal ([19, 20]) has a low complexity of O(log N), which makes it appealing for broadband implementations, but suffers from a sub-optimal delay bound, where the discrepancy between theoretical and actual delay can be as large as N times the optimal value [21]. We actually proposed in (20) above a new implementation of GPS, which combines reduced complexity of O(log N) with the optimal delay bound. A third criterion relates to the maximum discrepancy between the service received by any two sessions, as studied in [2, 19]. We state without proof that the scheme depicted by (20) is within a factor two of the minimal discrepancy. The proof will be released in a forthcoming paper. Turning back to our previous scheduler integrating guaranteed floor rates and maximum rate fairness, we immediately get its delay bound by merely reminding that, for the worst-case to happen, all sessions must be backlogged and the guaranteed rate summation must equal C. While the guaranteed rates were depicted by ϕ_k values in (20), they are now given by the different floor rates f_k. In such a case, it follows from (6) that $x_i = f_i$. As a consequence, the worst-case study for this scheme can be conducted merely by replacing ϕ_k with f_k in the above demonstration. Since two summations over $\mathcal{B}(t)$ are to be approximated instead of one, it turns out that our proposal also has both a low complexity and the optimal delay bound.

6 CONCLUSION

We have proposed in this paper a new scheduling policy of the Generalized Processor Sharing family, which provides an integrated solution to the enforcement of quality of service guarantees for adaptive multimedia applications. These QoS guarantees relate to three major issues: floor rate guarantee, bounded delay for all interactive applications, and need to enforce adequate fairness. We have shown that our proposal, by providing isolation, makes it possible for different adaptive (and also non-adaptive) applications to harmoniously share network resources, based on a fairness definition which is both a common-sense one, and a well accepted one (i.e. proportional fairness). We also proposed a low complexity implementation (i.e. amenable to broadband speeds) that provides the optimal delay bound for a packet GPS-like scheduler. The simplified example we studied is actually an implementation of GPS, with the same complexity as the *Self-Clocked Fair Queuing* scheme

proposed by Golestani in [19], and a better delay bound. In addition, we point out that QoS routing (which is still largely an open issue) could indeed use the F and M values as indicators of the actual node occupancy (although no effective congestion takes place, due to applications adapting to available bandwidth), so as to select the least loaded paths. These indicators could also be of great value for answering network dimensioning issues. But the remaining issue we believe is the most important relates to the maximum rate values, that must be provided to the network for proper fairness enforcement. It is user cooperation, as it is required that users tell the truth in stating their maximum rate for fairness to be consistent. Walking into the steps of Bolot in [3], Kelly in [23], and Shenker in [30], we believe that a workable way to enforce this is to make use of pricing as an incentive not to overestimate one's needs. Indeed, guaranteed floor rates could be charged a high price (depending on bandwidth reservation), in order to discourage overprovisioning, and, conversely, shared bandwidth could be charged a lower price related to maximum rates, that is, independent of the actual rates achieved by applications. The latter price would reflect relative bandwidth purchases rather than absolute ones. We believe that such an idea is close to the one advocated by MacKie-Mason and Varian in [25], where an auction process for bandwidth allocation is presented. In addition, consider that such a pricing scheme ultimately gives an economic meaning to the fairness that the network enforces.

The author is fully indebted to Maher Hamdi, Laurent Toutain, Pierre Rolin and Alain Léger for their valuable support. Also thanks to the anonymous reviewers for their insightful comments. This work was done within the CNET-DEST collaboration number 93 PE 7301.

7 REFERENCES

[1] A. Albanese, J. Blömer, J. Edmonds, M. Luby. *Priority Encoding Transmission*. ICSI Technical Report TR-94-039, Berkeley, 1994.

[2] J.C.R. Benett, H. Zhang. *WF^2Q: Worst-Case Fair Weighted Fair Queueing*. In Proceedings of IEEE Infocom, pp 120-128, San Francisco, CA, march 1996.

[3] J-C. Bolot. *Cost-Quality Tradeoffs in the Internet*. To appear in Computer Networks and ISDN Systems.

[4] J-C. Bolot, A. Vega-García. *Control Mechanisms for Packet Audio in the Internet*. Procs. IEEE INFOCOM'96, pp 232-239. San Francisco, march 1996.

[5] J.C. Bolot, T. Turletti. A Rate Control Mechanism for Packet Video in the Internet. Procs. IEEE Infocom'94, pp. 1216-1223, Toronto, Canada, June 1994.

[6] R. Braden, D. Clark, S. Shenker, *Integrated Services in the Internet Architec-*

ture: an Overview. Request for Comments 1633, june 1994.

[7] A. Campbell, G. Coulson. *Experiences with an adaptive multimedia transport system in a QoS Architecture.* Technical Report, Columbia University, available at URL
http://www.ctr.columbia.edu/~campbell/papers/adapt.ps.gz

[8] D.D. Clark, S. Shenker, L. Zhang. *Supporting Real-Time Applications in an Integrated Services Packet Network: Architecture and Mechanism.* Proc. ACM SIGCOMM'92, august 1992.

[9] C. Compton, D. Tennenhouse. *Collaborative Load Shedding for Multimedia Based Applications.* Int. Conference on Multimedia Computing Systems, Boston. may 1994.

[10] L. Delgrossi *et al. Media Scaling for Audiovisual Communications with the Heidelberg Transport System.* Procs. ACM Multimedia'93, Anaheim, California, august 1993.

[11] A. Demers, S. Keshav, S. Shenker. *Analysis and simulation of a fair queueing algorithm.* In Journal of Internetworking Research and Experiment, pp 3-26, october 1990. Also in Procs. ACM SIGCOMM'89, pp 3-12.

[12] C. Diot, A. Seneviratne. *Quality of Service in Heterogeneous Distributed Systems.* Submitted to HICSS-30. Hawai. January 1997.

[13] C. Diot, C. Huitema, T. Turletti. *Multimedia Applications should be Adaptive.* HPCS Workshop. Mystic (CN), August 23-25, 1995.

[14] C. Diot. *Adaptive Applications and QoS Guaranties.* IEEE Multimedia Networking Conference. Aizu (Japan). September 27-29, 1995.

[15] D. Ferrari. *Client Requirements for Real-Time Communication Services.* RFC 1193, also in IEEE Communications Magazine vol 28(11), november 1990.

[16] A. Fladenmuller, A. Seneviratne, E. Horlait. *A Hybrid QoS Management Scheme for distributed multimedia applications.* Procs. PROMS'95 workshop, Slazburg, Austria, october 1995.

[17] S. Floyd, V. Jacobson. *Random Early Detection Gateways for Congestion Avoidance.* IEEE/ACM Transactions on Networking, vol 1, no 4, pp 397-413, august 1993.

[18] M.W. Garrett. *A Service Architecture for ATM: From Applications to Scheduling.* IEEE Network, May /June 1996, pp 6-14.

[19] S. J. Golestani. *A Self-Clocked Fair Queueing Scheme for Broadband Applications.* In Procs IEEE INFOCOM'94, pp 636-646, Toronto, CA, june 1994.

[20] S. J. Golestani. *Fair Queueing Algorithms for Packet Scheduling in BISDN.* Procs. IZS'96, Zürich, 1996.

[21] P. Goyal, S. Lam, H.M. Vin. *Determining End-to-end Delay Bounds in Heterogeneous Networks.* in Procs. NOSSDAV'95, Durham, april 1995.

[22] P. Goyal, H.M. Vin, H. Cheng. *Start-time Fair Queueing: A Scheduling Algorithm for Integrated Services Packet Switching Networks.* in Procs. ACM SIGCOMM'96, Stanford, CA, august 1996.

[23] F. Kelly. *Charging and rate control for elastic traffic.* to appear in European Transactions on Telecommunications, January 1997.

[24] C. Lefelhocz, B. Lyles, S. Shenker, L. Zhang. *Congestion Control for Best-Effort Service: Why We Need a New Paradigm.* IEEE Network vol 10(1), january / february 1996.

[25] J. MacKie-Mason, H. Varian. *Some Economics of the INTERNET.* 10th Michigan Public Utility Conference, march 1993.

[26] A.K. Parekh and R.G. Gallager. *A Generalized Processor Sharing Approach to Flow Control In Integrated Services Networks-The Single Node Case.* In Proceedings of Infocom, pages 914-924. IEEE, 1992.

[27] J.L. Rexford, A.G. Greenberg, F.G. Bonomi. *Hardware-Efficient Fair Queueing Architectures for High-Speed Networks.* Procs. IEEE INFOCOM'96, San Francisco, 1996, pp 638-646.

[28] S. Shenker. *Fundamental Design Issues for the Future Internet.* IEEE Journal on Selected Areas in Communications, vol. 13 (no. 7): 1176-1188, Sep. 1995.

[29] S. Shenker. *Making Greed Work in Networks: A Game-theoretic Analysis of Switch Service Disciplines.* Procs. SIGCOMM'94 Conference, Oct. 1994, ACM, New York, pp. 47-57.

[30] S. Shenker. *Service Models and Pricing Policies for an Integrated Services Internet.* in Public Access to the Internet, B. Kahin and J. Keller eds., MIT Press.

[31] J. Wroclawski. *Specification of the Controlled-Load Network Element Service.* Internet Engineering Task Force, INTERNET-DRAFT draft-ietf-intserv-ctrl-load-svc-02.txt, june 1996, work in progress.

[32] L. Zhang. *Virtual Clock: A New Traffic Control Algorithm for Packet Switching Networks.* In Procs. ACM SIGCOMM'90, pp 19-29, Philadelphia, PA, september 1990.