INSTITUTO DE COMPUTAÇÃO
UNIVERSIDADE ESTADUAL DE CAMPINAS

# On the performance of Generalized Processor Sharing Servers under Long-Range Dependent Traffic

*Flávio M. Pereira, Nelson L. S. Fonseca and Dalton S. Arantes*

Technical Report - IC-01-11 - Relatório Técnico

October - 2001 - Outubro

# On the performance of Generalized Processor Sharing Servers under Long-Range Dependent Traffic *

Flavio de M. Pereira,† Nelson L. S. Fonseca‡ and Dalton S. Arantes§

**Abstract**

This paper introduces the computation of delay and backlog bounds for a Generalized Processor Sharing (GPS) server under self-similar traffic. The traffic is supposed to be regulated by the Fractal Leaky Bucket policing mechanism, which is an appropriate regulator for self-similar traffic. Results are extended to a network of GPS servers with arbitrary topology, and the stability of those networks is analysed.

## 1 Introduction

In order to support the diverse QoS requirements of multimedia applications, a network has to be empowered with proper traffic control mechanisms. Scheduling mechanisms determine the share of link capacity distributed among different sessions (flows or connections). Generalized Processor Sharing (GPS) scheduling is a work-conserving, idealized fluid-model discipline which considers traffic infinitely divisible and serves each session at a rate proportional to a pre-assigned weight $\phi$ (GPS assignment). For any session $i$ that is continuously backlogged, and any other session $j$,

$$\frac{S_i(\tau;t)}{S_j(\tau;t)} \geq \frac{\phi_i}{\phi_j}\,,\ j = 1, 2, \ldots, N, \tag{1}$$

where $S(\tau;t)$ denotes the amount of service offered to a session in the interval $[\tau;t]$. It can thus be proved that the GPS scheduling guarantees a minimum service rate for each session, given by

$$g_i = \frac{\phi_i}{\sum_{i=1}^{N} \phi_j} r\,,\ i = 1, 2, \ldots, N, \tag{2}$$

where $r$ is the server rate.

Packetized versions of GPS scheduling, such as Weighted Fair Queueing, have been implemented in various commercial switches. The exact analysis of a GPS server is not trivial, since the service offered to each session depends not only on its traffic, but also on the traffic of all other sessions. The existence and uniqueness of the mapping between arrival process and backlog process for a

*This work was sponsored by FAPESP (process no. 00/00069-0), by CNPq (process no. 300064/95-0) and by Ericsson Research (contract UNI-35/19-2000).

†F. M. Pereira and D. S. Arantes are with the Department of Communications, School of Electrical and Computer Engineering, State University of Campinas, P.O. Box 6101, 13083-970 Campinas/SP – Brazil.

‡N. L. S. Fonseca is with the Institute of Computer Science, State University of Campinas, P.O. Box 6176, 13084-971 Campinas/SP – Brazil.

§*E-mails* : pereira@decom.fee.unicamp.br; nfonseca@ic.unicamp.br; dalton@decom.fee.unicamp.br

GPS server have been proved by Dupuis and Ramanan [1]. Since, however, obtaining an analytical representation of this mapping is not an easy task, loose bounds are often used to characterize the performance of a GPS server. In general, such bounds are obtained by assuming specific arrival processes or network topologies. In their seminal papers, by assuming that flows are Leaky Bucket constrained, Parekh and Gallager [2, 3] obtained deterministic upper bounds on individual session backlog and delay, both for an isolated GPS server and for a network of GPS servers.

Recently, it has been shown that certain types of traffic, ranging from local to wide area network traffic, are long-range dependent (LRD) [4]. In fact, there is clear evidence that LRD traffic can potentially cause massive packet losses [5]. A new traffic model, the fractional Brownian motion (fBm) envelope process, has been proposed for the characterization of LRD sources, and extensively validated against video and LAN traffic [6, 7]. The Fractal Leaky Bucket (FLB) policing mechanism, capable of monitoring LRD sources was also introduced [6, 7]. This FLB mechanism has been proved to be much more effective for monitoring LRD traffic than is the classic Leaky Bucket mechanism.

Although several investigations on the identification of LRD traffic are available, less attention has been paid to the analysis of the impact of LRD on traffic control mechanisms. In this paper, the behavior of both an isolated GPS server and a network of GPS servers with arbitrary topology and fed by LRD traffic are analysed, with this traffic assumed to be regulated by the FLB mechanism.

This paper is organized as follows. In Section 2, some related work is discussed. In Section 3, the fBm envelope process and the FLB policing mechanism are presented. In Section 4, an isolated GPS server is analysed, and an algorithm for obtaining bounds on individual session backlog and delay is proposed. In Section 5, a GPS network with arbitrary topology is analysed, focusing on issues of stability and on the provision of bounds for individual session end-to-end delay and backlog. Finally, in Section 6, conclusions are drawn.

## 2 Related Work

Since the seminal papers of Parekh and Gallager [2, 3], several investigations providing a performance analysis of GPS servers have been conducted. Zhang, Towsley and Kurose [8] decomposed a GPS server with $N$ sessions into an equivalent system consisting of $N$ queues, each one fed by a separate flow. They assumed that the input traffic flows were modeled by Exponentially Bounded Burstiness (EBB) processes [9, 10] and derived statistical bounds based on a sample path analysis. An interesting property of such GPS servers fed by EBB process is that the output process is also an EBB process, which simplifies the derivation of stochastic bounds for a network of queues. Nevertheless, this approach leads to conservative bounds.

Only recently has the performance of a GPS server under LRD traffic been considered. Borst, Boxma and Jelenković [11, 12] analysed the asymptotic behavior of a GPS server fed by heavy-tailed ON/OFF sources, as well as by subexponential instantaneous bursts (i.e. renewal processes where the interarrival times are generally distributed and burst sizes are subexponential). These authors pointed out that the service rate of a connection depends on other connections only in relation to mean arrival rate, which is also valid for networks of queues. Van Uitert and Borst [13] extended those results to feed-forward networks of GPS servers, proving that, if the bottleneck node for a session is the last one on a path, the tail behaviour of the corresponding backlog process is equivalent to that of a tandem network with two nodes, in which service is provided at a constant rate which depends only on the mean arrival rate of the other flows.

Jelenković and Momčilović [14] analysed a GPS server with a single buffer of a fixed size, fed by a set of heavy-tailed ON/OFF sources to which a clearing rate greater than the mean arrival rate is guaranteed. Assuming the discard of the necessary amount of the most demanding sessions backlog to accommodate the incoming traffic whenever the buffer is full, they showed that the loss rate of a particular flow is asymptotically close to the loss rate that it would experience if it were stored in a buffer of the same fixed size, and served at a constant rate equal to the server capacity minus the mean arrival rate of the other flows. Each flow thus sees the whole common buffer as if it were dedicated to this flow, with the whole system operating as if the queueing capacity were multiplied. Such a result offers an interesting guideline for buffer dimensioning in GPS servers.

# 3 Traffic model and regulation

## 3.1 Fractional Brownian traffic

The use of fBm in traffic modeling has been intense after the introduction of the following process [5]:

$$\mathbf{A}(t) = \rho t + \sigma \mathbf{Z}(t)\,,\; t \geq 0, \tag{3}$$

where $\mathbf{Z}(t)$ is a normalized fBm process with a mean of zero, a variance of $t^{2H}$ and a Hurst parameter of $H$. This process, called *fractional Brownian traffic,* is considered to be a parsimonious model, since it is based on only three parameters: the mean rate $\rho > 0$ ; the Hurst parameter of the process $\mathbf{Z}(t)$**,** $0.5 \leq H < 1$ and a coefficient $\sigma > 0$ associated with the incremental variance of $\mathbf{A}(t)$, as given by the following:

$$\mathrm{Var}\left[\mathbf{A}(t+s) - \mathbf{A}(t)\right] = \sigma^2 s^{2H}. \tag{4}$$

## 3.2 Fractional Brownian motion envelope process

Despite its parsimony, the use of fractional Brownian traffic in performance analysis is rather complex, due to its hard mathematical tractability; an alternative is the fractional Brownian motion (fBm) envelope process [7, 6]. This fBm envelope process provides most of the advantageous properties of fractional Brownian traffic without a high mathematical overhead. It can be defined as follows: for a fractional Brownian traffic process $\mathbf{A}(t)$, with mean rate $\rho$, Hurst parameter $H$ and incremental variance given by (4), the corresponding envelope process is

$$\widehat{A}(t) = \rho t + k\sigma t^H. \tag{5}$$

The additional parameter $k$ determines the probability that $\mathbf{A}(t)$ exceeds $\widehat{A}(t)$ at time $t$, i.e.

$$\mathbb{P}\left[\mathbf{A}(t) > \widehat{A}(t)\right] = \overline{\Phi}(k), \tag{6}$$

where $\overline{\Phi}(y)$ is the Gaussian residual distribution function. For relatively large values of $k$, an approximation is possible:

$$\mathbb{P}\left[\mathbf{A}(t) > \widehat{A}(t)\right] \sim \exp\left(-\frac{k^2}{2}\right). \tag{7}$$

Some peculiarities of the fBm envelope process are presented in [6]. Moreover, it can also be defined an envelope process for the size of the queue in a server that operates at constant rate and which is fed by fBm traffic. If $\mathbf{q}(t)$ denotes the size of the queue, this envelope process is given by:

$$\widehat{q}(t) = (\rho - g)\,t + k\sigma t^H. \tag{8}$$

Furthermore, it can be shown that:

$$\mathbb{P}\left[\mathbf{q}(t) > \widehat{q}(t)\right] = \mathbb{P}\left[\mathbf{A}(t) > \widehat{A}(t)\right] = \overline{\Phi}(k). \tag{9}$$

Mayor and Silvester [7] defined a time scale of interest for queues fed by the fBm envelope process, called the Maximum Timescale (MaxTs). This represents the instant of time at which the unfinished work in a queueing system achieves its maximum value in a probabilistic sense. After that, the instantaneous arrival rate is lower than the service rate, and queue size starts decreasing.

Mathematically, MaxTS is defined as

$$t^* = \arg\max_{t\geq 0} \widehat{q}(t) = \left[\frac{k\sigma H}{g-\rho}\right]^{\frac{1}{1-H}}.$$

It is, then, easy to verify that:

$$q^* = \max_{t\geq 0} \widehat{q}(t) = (g-\rho)^{\frac{H}{H-1}} (k\sigma)^{\frac{1}{1-H}} H^{\frac{H}{1-H}} (1-H), \tag{10}$$

and,

$$\mathbb{P}\left[\mathbf{q}(t) > q^*\right] \sim \exp\left(-\frac{k^2}{2}\right). \tag{11}$$

*Remark.* Equation (11) is implicitly based on the queueing analysis conducted by Norros [5] and by Duffield and O'Connell [15]. Using the theory of large deviations, these authors obtained a lower bound on the probability distribution of the maximum queue size, $\mathbf{q}^*$, which approximates the actual distribution only in asymptotic sense.

### 3.3 Fractal Leaky Bucket policing mechanism

Once a new session is admitted into the network, policing is employed to certify that the traffic generated by the session is in agreement with the description provided at the time of admission. The most frequently used mechanism for this purpose is the Leaky Bucket algorithm, which constrains traffic to the following envelope

$$\widehat{A}_{LB}(t) = \rho_{LB} t + \sigma_{LB}, \tag{12}$$

where $\rho_{LB}$ is the leaky rate and $\sigma_{LB}$ is the bucket size. Nevertheless, it has already been shown that policing bursty sources with a Leaky Bucket mechanism can be a difficult task, since its two parameters are not sufficient to completely characterize the target traffic [16]. Actually, the inadequacy of the Leaky Bucket mechanism is due to an implicit premise that aggregated traffic may be bounded by a linear function of time, which is not true for bursty sources. For instance, consider a source which traffic is characterized by the fBm traffic process $\mathbf{A}(t)$ given in (3). The probability that it violates the Leaky Bucket constraint $\widehat{A}_{LB}(t)$ at time $t$ is given by

$$\begin{aligned} \mathcal{P}_{LB}(t) &= \mathbb{P}\left\{\mathbf{A}(t) > \widehat{A}_{LB}(t)\right\} \\ &= \mathbb{P}\left\{\rho t + \sigma \mathbf{Z}(t) > \rho_{LB} t + \sigma_{LB}\right\}. \end{aligned} \tag{13}$$

Using the self-similar property $\mathbf{Z}(t) = t^H \mathbf{Z}(1)$ the following is obtained:

$$\begin{aligned} \mathcal{P}_{LB}(t) &= \mathbb{P}\left\{\rho t + \sigma t^H \mathbf{Z}(1) > \rho_{LB} t + \sigma_{LB}\right\} \\ &= \mathbb{P}\left\{\mathbf{Z}(1) > \frac{\rho_{LB} - \rho}{\sigma} t^{1-H} + \frac{\sigma_{LB}}{\sigma} t^{-H}\right\}, \end{aligned} \tag{14}$$

where the distribution of $\mathbb{P}\{\mathbf{Z}(1) > y\}$ is given by $\exp\left(-y^2/2\right)$ for a large $y$. The inadequacy of the Leaky Bucket mechanism for policing fBm traffic can be verified in an asymptotic sense by analysing (14) as $t \to \infty$, i.e.

$$\begin{aligned} \mathcal{P}_{LB} &= \lim_{t\to\infty} \mathbb{P}\left\{\mathbf{Z}(1) > \frac{\rho_{LB} - \rho}{\sigma} t^{1-H} + \frac{\sigma_{LB}}{\sigma} t^{-H}\right\} \\ &= \mathbb{P}\left\{\mathbf{Z}(1) > \lim_{t\to\infty} \left[\frac{\rho_{LB} - \rho}{\sigma} t^{1-H} + \frac{\sigma_{LB}}{\sigma} t^{-H}\right]\right\}. \end{aligned}$$

Clearly, if the leaky rate $\rho_{LB}$ is lower than the source mean rate $\rho$, $\mathcal{P}_{LB} = 1$, which means that the policing mechanism will always discard some of the traffic. However, low violation probabilities cannot be achieved even when $\rho_{LB} = \rho$. In that case, $\mathcal{P}_{LB}$ is

$$\begin{aligned} \mathcal{P}_{LB} &= \mathbb{P}\left\{\mathbf{Z}(1) > \lim_{t\to\infty} \left[\frac{\sigma_{LB}}{\sigma} t^{-H}\right]\right\} \\ &\simeq \mathbb{P}\{\mathbf{Z}(1) > 0\} = 0.5. \end{aligned} \tag{15}$$

It is then easy to prove that low violation probabilities are attainable only if $\rho_{LB} > \rho$. In this case,

$$\begin{aligned} \mathcal{P}_{LB} &= \mathbb{P}\left\{\mathbf{Z}(1) > \lim_{t\to\infty} \left[\frac{\rho_{LB} - \rho}{\sigma} t^{1-H} + \frac{\sigma_{LB}}{\sigma} t^{-H}\right]\right\} \\ &= \mathbb{P}\left\{\mathbf{Z}(1) > \lim_{t\to\infty} \left[\frac{\rho_{LB} - \rho}{\sigma} t^{1-H}\right]\right\} \\ &= \mathbb{P}\{\mathbf{Z}(1) > \infty\} = 0. \end{aligned} \tag{16}$$

In practice, (15) and (16) indicate that, if the leaky rate is set close to the source mean rate, policing may consider well-behaving packets as violators and mark them for discard. On the other hand, if the leaky rate is greater than the mean rate (if, for example, it is set to the source peak rate), policing efficiency and gains in statistical multiplexing are greatly reduced.

It is worth noticing that neither (15) nor (16) depends on the bucket size $\sigma_{LB}$, which has no asymptotic effect on the operation of the Leaky Bucket mechanism under fBm traffic. On a more limited timescale, however, larger buckets permit the acceptance of large bursts into the network, thus reducing the efficacy of the policing mechanism.

The Fractal Leaky Bucket (FLB) mechanism has been the proposal designed to overcome the inefficacy of the Leaky Bucket mechanism for policing LRD traffic [6, 7]. This mechanism constrains traffic to the fBm envelope process, as follows:

$$\widehat{A}_{FLB}(t) = \rho t + \psi t^H, \tag{17}$$

where $\rho$ corresponds to the mean arrival rate of the source and $H$ is the Hurst parameter. The parameter $\psi$ is given by $k\sigma$, where $\sigma$ and $k$ are constants associated, respectively, to the standard deviation of the arrival process and to the probability of the violation of $\widehat{A}_{FLB}(t)$ by the arrival process. If all parameters of the FLB mechanism are in agreement with those of the incoming traffic, the probability of violating $\widehat{A}_{FLB}(t)$ is

$$\mathbb{P}\left[\mathbf{A}(t) > \widehat{A}_{FLB}(t)\right] = \overline{\Phi}(k),$$

which can be evaluated by using (7). Notice that low values for the probability of violation can be attained just by choosing an adequate value for the parameter $k$, which proves the efficacy of the FLB mechanism for policing LRD sources, especially those modeled by the fBm traffic process.

#### 3.3.1 Operation of the Fractal Leaky Bucket mechanism

The operation of the FLB mechanism can be described as follows. Consider a time window with a length of $\tau$ time units. If the arrival process exceeds the declared mean value inside that window (given by $\rho\tau$), then all packets exceeding the envelope process $\widehat{A}_{FLB}(t)$ in that period are discarded, and the time window is enlarged by $\tau$ time units. This new window goes into effect at the time that the arrival process violates the declared mean rate. This process is then repeated as long as the incoming traffic violates the declared mean rate inside the window. However, since certain violating packets have already been discarded in the previous window, the number of packets to be discarded is now equal to the number of violating packets in the current window minus the number of packets that have already been discarded in the previous window. When the mean number of arrivals drops below the declared value, the window is shrunk to $\tau$ time units, and the policing process is reinitiated.

The FLB mechanism is presented in Algorithm 1.

**Algorithm 1** The Fractal Leaky Bucket policing mechanism.

$t \leftarrow 0$; $n \leftarrow 1$;
**loop**
  **if** $\mathbf{A}(t, t+n\tau) > \rho n\tau$ **then**
    **if** $\mathbf{A}(t, t+n\tau) > \widehat{A}_{FLB}(t, t+n\tau)$ **then**
      Discard $\mathbf{A}(t, t+n\tau) - \widehat{A}_{FLB}(t, t+n\tau) - \mathbf{A}(t, t+(n-1)\tau) + \widehat{A}_{FLB}(t, t+(n-1)\tau)$ packets;
    **end if**
    $n \leftarrow n+1$;
  **else**
    $t \leftarrow t+n\tau$; $n \leftarrow 1$;
  **end if**
**end loop**

#### 3.3.2 Use of buffers with Fractal Leaky Bucket mechanism

There is, however, another option for dealing with violating packets. Instead of being discarded, they can be stored in a buffer until they become eligible for transmission. In such case, a certain amount of traffic would be retained inside the policing mechanisms, thus introducing a certain backlog and delay in the traffic flow. If the buffer has a fixed size, $B$, the total amount of traffic accepted by the FLB mechanism would be given by

$$\widehat{A}_{FLB-B}(t) = \rho t + \psi t^H + B. \tag{18}$$

It can be shown that this buffer has no asymptotic effect on the probability that an arrival process violates $\widehat{A}_{FLB}(t)$, no matter how large it is. For instance, consider $\widehat{\rho}$ as the declared mean rate. The probability that the incoming traffic violates the buffered FLB constraint is

$$\begin{aligned}
\mathcal{P}_{FLB-B}(t) &= \mathbb{P}\left\{\mathbf{A}(t) > \widehat{A}_{FLB-B}(t)\right\} \\
&= \mathbb{P}\left\{\rho t + \sigma \mathbf{Z}(t) > \widehat{\rho} t + \psi t^H + B\right\} \\
&= \mathbb{P}\left\{\sigma t^H \mathbf{Z}(1) > (\widehat{\rho} - \rho) t + \psi t^H + B\right\} \\
&= \mathbb{P}\left\{\mathbf{Z}(1) > \frac{(\widehat{\rho} - \rho)}{\sigma} t^{1-H} + \frac{\psi}{\sigma} + \frac{B}{\sigma} t^{-H}\right\}. \qquad (19)
\end{aligned}$$

Hence, the asymptotic value of $\mathcal{P}_{FLB-B}(t)$ as $t \to \infty$ is

$$\begin{aligned}
\mathcal{P}_{FLB-B} = \exp\Big\{ &-\frac{1}{2\sigma^2} \lim_{t\to\infty} \left[(\widehat{\rho} - \rho)^2 t^{2-2H} + 2\psi(\widehat{\rho} - \rho) t^{1-H} + \psi^2 \right. \\
&\left. + 2(\widehat{\rho} - \rho) B t^{1-2H} + 2\psi B t^{-H} + B^2 t^{-2H}\right] \Big\}. \qquad (20)
\end{aligned}$$

Notice that all terms multiplying $B$ inside the brackets in (20) undergo an asymptotic decrease so they have little influence in $\mathcal{P}_{FLB-B}(t)$ as $t \to \infty$. This result may be regarded as a consequence of the phenomenon of buffer inefficiency, which is generally associated with LRD traffic: since the tail of the buffer occupancy probability distribution decreases slowly, the buffer will probably be full very often, and the violating traffic will be discarded anyway. Thus, queueing of violating LRD traffic for latter transmission is not an effective way to reduce the amount of data discarded by policing mechanisms. Moreover, the buffer introduces additional backlog and delay.

# 4 Performance analysis of GPS servers fed by LRD traffic

In this section, an algorithm for computing bounds for individual session backlog and delay in a GPS server fed by LRD traffic is introduced. These sources are assumed to be regulated by the FLB policing mechanism. The following notation is used throughout this section. For each session $i$, the amount of traffic that enters the network during the interval $[\tau; t]$ is denoted by $\mathbf{A}_i(\tau, t)$, which has a sample-path realization of $A_i(\tau, t)$. Let $\mathbf{S}_i(\tau, t)$ be the amount of session $i$ traffic served by the server during the interval $[\tau; t]$, and $S_i(\tau, t)$ the corresponding sample-path realization. To simplify notation, $\tau$ is omitted whenever it is equal to zero. The amount of session $i$ traffic that is backlogged at time $t$ is denoted by $\mathbf{Q}_i(t) = \sup_{\tau \leq t} \{\mathbf{A}_i(\tau, t) - \mathbf{S}_i(\tau, t)\}$, with a sample-path realization of $Q_i(t)$. $\mathbf{D}_i(t)$ denotes the delay experienced by the session $i$ traffic that arrives at the server at time $t$, and its sample-path realization is $D_i(t)$. Since no loss of generality will result, the server rate $r = 1$ will be assumed.

The algorithm proposed here uses the definition of a deterministic, piecewise-linear function that provides lower bounds for the service offered by a GPS server to a session [2], denoted as $\widehat{S}_i(t)$. Such a session is considered to be greedy from time $\tau$ if its traffic exactly follows the envelope imposed by the policing mechanism from $t = \tau$. In the case of the fBm envelope process, this means that the session $i$ is continuously backlogged for $t < e_i$; once the backlog has been completely consumed, the queue is expected to remain empty. In other words, session $i$ is busy during the interval $[0; e_i]$, which is called the busy period of session $i$, although, for unstable sources, it is possible that $e_i \to \infty$.

Parekh and Gallager have proved that, if all sources are Leaky Bucket constrained, the maximum backlog and delay are achieved when all sessions are greedy from time zero [2, Theorem 3]. This result can be extended to the case of FLB constrained traffic, as is proved below.

**Lemma 4.1.** *Assume that the session $i$ is in a busy period during the interval $[\tau;t]$. Then, for any subset $\mathcal{M}$ of $n$ sessions, $1 \leq n \leq N$ and for any time $t \geq \tau$*

$$S_i(\tau;t) \geq \frac{\phi_i}{\sum_{j\in\mathcal{M}} \phi_j}\left[t-\tau-\sum_{j\notin\mathcal{M}} \widehat{A}_j(\tau;t)\right]. \tag{21}$$

This lemma was established by Parekh and Gallager for Leaky Bucket constrained traffic [2, Lemma 6]. It is easy to extend their reasoning to show that (21) does not depend on any particular traffic model. By definition, the following inequality holds for all $j$:

$$S_j(\tau;t) \leq \widehat{A}_j(\tau;t).$$

Also, since the session $i$ is busy during the interval $[\tau;t]$, (1) states that $S_j(\tau;t) \leq \frac{\phi_j}{\phi_i} S_i(\tau;t)$. Thus,

$$S_j(\tau;t) \leq \min\left\{\widehat{A}_j(\tau;t); \frac{\phi_j}{\phi_i} S_i(\tau;t)\right\}.$$

The total amount of service offered by the server to all sessions during the interval $[\tau;t]$ can be represented by the following:

$$S(\tau;t) = \sum_{j=1}^{N} S_j(\tau;t). \tag{22}$$

Since GPS scheduling is work-conserving and there is at least one busy session during the interval $[\tau;t]$ (Session $i$ itself), $S(\tau;t) = t-\tau$. Therefore, (22) can be rewritten as

$$\begin{aligned}
t-\tau &= \sum_{j=1}^{N} S_j(\tau;t) \\
&\leq \sum_{j=1}^{N} \min\left\{\widehat{A}_j(\tau;t); \frac{\phi_j}{\phi_i} S_i(\tau;t)\right\} \\
&= \sum_{j\in\mathcal{M}} \min\left\{\widehat{A}_j(\tau;t); \frac{\phi_j}{\phi_i} S_i(\tau;t)\right\} + \sum_{j\notin\mathcal{M}} \min\left\{\widehat{A}_j(\tau;t); \frac{\phi_j}{\phi_i} S_i(\tau;t)\right\} \\
&\leq \sum_{j\in\mathcal{M}} \widehat{A}_j(\tau;t) + \sum_{j\notin\mathcal{M}} \frac{\phi_j}{\phi_i} S_i(\tau;t),
\end{aligned} \tag{23}$$

for any subset of sessions $\mathcal{M}$. After rearranging the terms in (23), (21) is obtained.

**Lemma 4.2.** *Let $\widehat{A}_1, \ldots, \widehat{A}_N$ be the set of arrival functions in which all sessions are greedy from $\tau = 0$. For every session $i$, let $\widehat{S}_i(\tau,t)$, $\widehat{Q}_i(t)$ and $\widehat{D}_i(t)$ be the session $i$ service, backlog and delay functions under $\widehat{A}_1, \ldots, \widehat{A}_N$. Suppose that the time $t$ is within a session $i$ busy period that begins at the time $\tau$. Then,*

$$\widehat{S}_i(t-\tau) \leq S_i(\tau;t). \tag{24}$$

This lemma was also established by Parekh and Gallager for Leaky Bucket constrained traffic [2, Lemma 10]. Here, it is extended to FLB constrained traffic. If $\mathcal{B}$ denotes the set of sessions under $\widehat{A}_1, \ldots, \widehat{A}_N$ that are busy at the time $t-\tau$, Lemma 4.1 establishes that:

$$\begin{aligned} S_i(\tau;t) &\geq \frac{\phi_i}{\sum_{j\in\mathcal{B}}\phi_j}\left[t-\tau-\sum_{j\notin\mathcal{B}}\widehat{A}_j(\tau;t)\right] \\ &= \frac{\phi_i}{\sum_{j\in\mathcal{B}}\phi_j}\left[t-\tau-\sum_{j\notin\mathcal{B}}\rho_j(t-\tau)+\psi_j\left(t^{H_j}-\tau^{H_j}\right)\right]. \end{aligned} \tag{25}$$

Notice that $t^{H_j}-\tau^{H_j}\leq(t-\tau)^{H_j}$, since $H_j<1\ \forall j$. Substituting this inequality into (25) yields

$$\begin{aligned} S_i(\tau;t) &\geq \frac{\phi_i}{\sum_{j\in\mathcal{B}}\phi_j}\left[t-\tau-\sum_{j\notin\mathcal{B}}\rho_j(t-\tau)+\psi_j\left(t-\tau\right)^{H_j}\right] \\ &= \frac{\phi_i}{\sum_{j\in\mathcal{B}}\phi_j}\left[t-\tau-\sum_{j\notin\mathcal{B}}\widehat{A}_j(t-\tau)\right]. \end{aligned} \tag{26}$$

Since the right-hand side of (26) is equal to $\widehat{S}_i(t-\tau)$, (24) is proved.

**Lemma 4.3.** *Let $\widehat{A}_1, \ldots, \widehat{A}_N$ be the set of arrival functions in which all sessions are greedy from $\tau=0$. For every session $i$, let $\widehat{S}_i(\tau,t)$, $\widehat{Q}_i(t)$ and $\widehat{D}_i(t)$ be the session $i$ service, backlog and delay functions under $\widehat{A}_1, \ldots, \widehat{A}_N$. Suppose that $t$ is within a session $i$ busy period which begins at $\tau$. Then,*

$$\widehat{A}_i(t-\tau)\geq A_i(\tau;t).$$

This lemma was also established by Parekh and Gallager for Leaky Bucket constrained traffic [2, Lemma 11]. For FLB constrained traffic, it can be generalized to the following:

$$\begin{aligned} A_i(\tau;t) &\leq \rho_i(t-\tau)+\psi_i\left(t^{H_i}-\tau^{H_i}\right) \\ &\leq \rho_i(t-\tau)+\psi_i\left(t-\tau\right)^{H_i} \\ &= \widehat{A}_i(t-\tau). \end{aligned}$$

Lemmas 4.2 and 4.3 can be used to prove the following theorem:

**Theorem 4.4.** *For every session $i$, the maximum backlog and delay, namely $Q_i^*$ and $D_i^*$, are achieved (not necessarily at the same time) when every session is greedy from $\tau=0$.*

*Proof.* The proof for this theorem is an extension for FLB constrained traffic of Lemma 11 presented in [2]. For a busy period of the session $i$ that begins at the time $\tau$, suppose that

$$Q_i^* = Q_i(t^*) = \max_{t\geq\tau} Q_i(t).$$

From Lemmas 4.2 and 4.3 it is easy to verify that,

$$
\begin{aligned}
\widehat{A}_i(t^* - \tau) - \widehat{S}_i(t^* - \tau) &\geq A_i(\tau; t^*) - S_i(\tau; t^*) \\
\widehat{Q}_i(t^* - \tau) &\geq Q_i(t^*) - Q_i(\tau).
\end{aligned}
$$

Since it is assumed that the busy period of the session $i$ begins at the time $\tau$, $Q_i(\tau) = 0$ and $\widehat{Q}_i(t^* - \tau) \geq Q_i(t^*)$, which indicates that the backlog is maximized under $\widehat{A}_1, \ldots, \widehat{A}_N$. A similar proof shows that this set of arrivals also maximizes the delay. □

Theorem 4.4 is quite intuitive. If any session does not send as much traffic as possible from $\tau = 0$, it leaves some unused bandwidth to be distributed among the other sessions, thus reducing their backlog and delay. On the other hand, if a session sends as much traffic as possible, it prevents the anticipated serving of packets of other sessions. The worst-case scenario is, therefore, attained when all sessions are greedy from $\tau = 0$.

Suppose now that the backlog of one session, say the first one, is completely consumed at $t = e_1$. The value $e_1$ can be directly obtained from the definition of $Q_1(t)$, that is, $Q_1(e_1) = \widehat{A}_1(e_1) - c_1^1(e_1) = 0$. Since all sessions are greedy, the service provided for each session inside the interval $[0; e_1]$ is

$$
c_1^i(t) = \frac{\phi_i}{\sum_{j=1}^{N} \phi_j} t.
$$

At $t = e_2$ the backlog of another session, say $i = 2$, is completely consumed. During the interval $[e_1; e_2]$ the service offered to each session is thus

$$
c_2^i(t) = \begin{cases} \widehat{A}_1(t) & , i = 1 \\ \frac{\phi_i}{\sum_{j=2}^{N} \phi_j} \left[ t - e_1 - \widehat{A}_1(t) \right] + c_1^i(e_1) & , i > 1. \end{cases}
$$

The value of $e_2$ can be obtained considering that $Q_2(e_2) = \widehat{A}_2(e_2) - c_2^2(e_2) = 0$. Suppose that sessions are ordered by the time when their backlog is completely consumed by the server. This ordering can be considered to be a generalization of the feasible ordering proposed in [2]. The service offered for any session $i$ during the interval $[e_{j-1}; e_j]$ is

$$
c_j^i(t) = \begin{cases} \widehat{A}_i(t) & , 1 \leq i < j \\ \frac{\phi_i}{\sum_{l=j}^{N} \phi_l} [t - \xi_j] + c_{j-1}^i(e_{j-1}) & , i \geq j. \end{cases} \tag{27}
$$

where $\xi_j = e_{j-1} + \sum_{l=1}^{j-1} \widehat{A}_l(t)$. The value of $e_j$ may be obtained considering that $Q_j(e_j) = \widehat{A}_j(e_j) - c_l^j(e_l) = 0$. Segments $s_j^i$ are then given by

$$
s_j^i = \left. \frac{dc_j^i(t)}{dt} \right|_{t = e_{j-1}}. \tag{28}
$$

Fig. 1 shows how these segments are juxtaposed to construct $\widehat{S}_i(t)$. Obviously, the number of segments is limited to the number of sessions with backlogs which can be completely consumed by the server, which is conditioned by

$$\rho_i < \frac{\phi_i}{\sum_{j=i}^{N} \phi_j} \left[ 1 - \sum_{j=1}^{i-1} \frac{d\widehat{A}_j}{dt} (e_{i-1}) \right]. \tag{29}$$

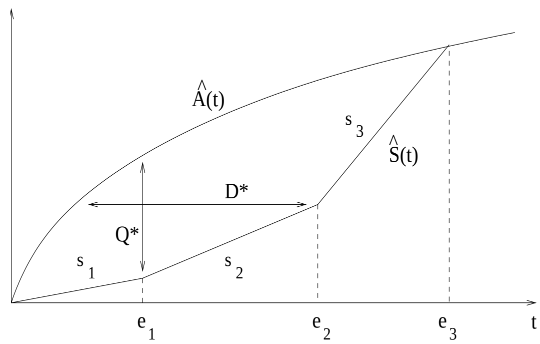


Figure 1: Construction of $\widehat{S}_i(t)$ and definition of $Q_i^*$ and $D_i^*$.

The maximum backlog and delay of the session $i$ , namely $Q_i^*$ and $D_i^*$, can thus be obtained by solving the following optimization problems:

$$Q_i^* = \max_t \left\{ \widehat{A}_i(t) - \widehat{S}_i(t) \right\}, \tag{30}$$

$$D_i^* = \arg\max_{d(t)} \left\{ d(t) : \widehat{A}_i(t) - \widehat{S}_i(t + d(t)) = 0 \right\}. \tag{31}$$

Fig. 1 also shows how $Q_i^*$ and $D_i^*$ are obtained. Solving the optimization problem (31) requires the inverse of $\widehat{A}_i(t)$ for $t \geq 0$. Although for $\widehat{A}_i(t)$ the inverse inside the interval of interest exists, this function cannot be expressed analytically. For this reason it is hard to solve (31) . One alternative is to use the approximation:

$$D_i^* \leq \frac{Q_i^*}{g_i} \tag{32}$$

Due to its mathematical simplicity, (32) will be used instead of (31) to evaluate $D_i^*$.

## 4.1 Existence and uniqueness of $Q_i^*$ and $D_i^*$

In order to develop an algorithm to evaluate $Q_i^*$ and $D_i^*$, it is necessary to prove that (30) can be solved when (29) is satisfied. In this case, the uniqueness of the solution can also be proved.

**Theorem 4.5.** *Consider a session $i$ greedy from time zero, with accumulated traffic described by the envelope $\widehat{A}_i(t) = \rho_i t + k_i \sigma_i t^{H_i}$, $t \geq 0$. The equation (30) can be solved whenever (29) holds.*

*Proof.* By definition, $\widehat{S}_i(t) < \widehat{A}_i(t)$ for $t \in (0; e_i)$. The equality condition $\widehat{S}_i(t) = \widehat{A}_i(t)$ is attained at $t = 0$ and $t = e_i$. Therefore, $Q_i(0) = Q_i(e_i) = 0$ and $Q_i(t) > 0$ for $t \in (0; e_i)$. To prove that there is at least one solution for (30), it is sufficient to show that $e_i$ is finite. Since $\frac{d\widehat{S}_i}{dt}(t)$ is strictly crescent in the interval $[0; e_i]$, $e_i$ is finite if, for any $j < i$,

$$\rho_i < s_j^i = \frac{d\widehat{S}_i}{dt}(t)\,,\ t \in [e_{j-1}; e_j)\,,$$

since

$$\lim_{t\to\infty} Q_i(t)|_{t<e_i} = \lim_{t\to\infty} \rho_i t + k_i\sigma_i t^{H_i} - s_j^i t \to \begin{cases} -\infty & s_j^i > \rho_i \\ +\infty & s_j^i \le \rho_i. \end{cases}$$

Hence, if the service rate offered to session $i$ is never greater than its mean rate, $\widehat{S}_i(t)$ never reaches $\widehat{A}_i(t)$ and the backlog grows indefinitely; however, if at any moment, that session is serviced at a rate greater than its mean rate, this situation will remain until $\widehat{S}_i(t)$ reaches $\widehat{A}_i(t)$, thus defining a time $e_i$ when the backlog is completely cleared. It can thus be concluded that the sufficient condition for the existence of an $s_j^i$ that is greater than $\rho$ is given by (29); if this condition is satisfied, at least one solution for (30) can be obtained. □

**Theorem 4.6.** *Consider a session $i$ greedy from time zero, with accumulated traffic described by the envelope $\widehat{A}_i(t) = \rho_i t + k_i\sigma_i t^{H_i}$, $t \ge 0$. If the optimization problem represented by (30) is solvable, then its solution is unique.*

*Proof.* Consider that $\xi_j^i = s_{j-1}^i - s_j^i$, where $s_j^i = 0$ for $j < 0$. Then the service curve $\widehat{S}_i(t)$ can be written as

$$\widehat{S}_i(t) = t \cdot \sum_{j=0}^{N} \xi_j^i u(t - e_j)\ , t \ge 0,$$

where $u(.)$ is the step function. Focusing on the interval $0 < t < e_i$ in which $Q_i(t) > 0$ and all terms $\xi_j^i$ are positive,

$$\widehat{S}_i(t) = t \cdot \sum_{j=0}^{i-1} \xi_j^i u(t - e_j)\ , 0 < t < e_i.$$

The backlog $Q_i(t)$ is then given by

$$Q_i(t) = \widehat{A}_i(t) - t \cdot \sum_{j=0}^{i-1} \xi_j^i u(t - e_j)\ , 0 < t < e_i.$$

And the first and the second derivatives of $Q_i(t)$ in that interval are

$$\frac{dQ_i}{dt}(t) = \frac{d\widehat{A}_i}{dt}(t) - \sum_{j=0}^{i-1} \xi_j^i \left[e_j \delta(t - e_j) + u(t - e_j)\right]\ , 0 < t < e_i,$$

and

$$\frac{d^2Q_i}{dt^2}(t) = \frac{d^2\widehat{A}_i}{dt^2}(t) - \sum_{j=0}^{i-1}\xi_j^i\left[e_j\delta^2(t-e_j)+\delta(t-e_j)\right]\ ,0<t<e_i,$$

where $\delta(.)$ and $\delta^2(.)$ are the Dirac delta function and its derivative, respectively. The second derivative of $\widehat{A}_i(t)$ is equal to $\psi_i H_i(H_i-1)t^{H_i-2}$, which is always negative for $t>0$. Since all $\xi_j^i$ are positive for $j<i$, the second derivative of $Q_i(t)$ is always negative for $0<t<e_i$. This condition is sufficient to guarantee that, if there is a solution to (30) inside that interval, it is unique. □

Similar results can be obtained to prove the existence and uniqueness of $D_i^*$ when it is evaluated by (31). However, a consideration of (32) shows that it is sufficient to prove the existence and uniqueness of $Q_i^*$ to guarantee the same properties for $D_i^*$.

## 4.2 Characterization of the output process

The analysis of the networks of GPS servers requires an adequate characterization of $S_i(t)$ which describes the traffic from a session $i$ that has already left the GPS server up to the time $t$. In [2], Parekh and Gallager use a definition of output burstiness suggested by Cruz [17, 18] to represent $S_i(t)$. This characterization is not adequate here, however, since it assumes that $S_i(t)$ can be upper bounded by a linear function of time, which is not true if the arrival process is FLB constrained.

Hence, the characterization of $S_i(t)$ in terms of FLB parameters is proposed here. The best possible characterization for this case was found to be the following:

$$S_i(t) \sim \widehat{A}_i(t), \tag{33}$$

i.e., if the envelope process $\widehat{A}_i(t)$ constrains the arrival process of session $i$, it also constrains the corresponding departure process, $S_i(t)$. It is easy to see why this happens: if a session $i$ satisfies (29), there will be a time $e_i$, such that $S_i(t) < \widehat{A}_i(t)$ for $t<e_i$ and $S_i(t) = \widehat{A}_i(t)$ for $t \geq e_i$. Therefore, the tightest upper bound that can be obtained for all $t>0$ is $S_i(t) = \widehat{A}_i(t)$. On the other hand, if (29) is not satisfied, then $S_i(t) < \widehat{A}_i(t)$ for all $t>0$, which means that $\widehat{A}_i(t)$ is also a valid upper bound, although it is not the tightest one.

## 4.3 Computational algorithm to evaluate $Q_i^*$ and $D_i^*$

The computational algorithm for the obtention of the upper bounds for the backlog and for delay in an isolated GPS server is presented in Algorithm 2. Although the method proposed by Zhang *et al.* [8] is based on closed-form expressions, it requires sessions to be previously sorted according to the feasible ordering proposed in [2]. Algorithm 2 does not require pre-sorting of sources, which constitutes an advantage in relation to Zhang's approach. Moreover, this approach is able to obtain bounds only when the mean arrival rate is lower than the minimum guaranteed service rate. On the other hand, the algorithm proposed here also obtains bounds for all sessions that satisfies (29), i.e., for sessions that are initially unstable, although they become stable as the backlog from other sessions is cleared up, and the service is redistributed.

The main loop of Algorithm 2 operates as follows. First, the slope of the $j^{th}$segment of $\widehat{S}_i(t)$, which is denoted by $s_j^i$, is computed for all session $i$. The slope of this segment depends on whether the busy period of the session $i$ is already finished, as can be seen in (27) and (28). It defines a set of sessions $i$ which are stable (i.e. $s_j^i > \rho_i$ ) and for which $Q_i^*$ and $D_i^*$ have not yet been evaluated. The algorithm will then determine which session finishes its busy period first. This information is used to determine the time at which the $j^{th}$ segment will end ($e_j$). Finally, for all sessions in which the maximum backlog has been achieved during the $j^{th}$ segment (at the time $t^*$), the backlog and the delay bounds are computed using (10) and (32), respectively. Since Theorem 4.6 assures the uniqueness of the solution, bounds are not computed for those sessions where the maximum backlog has already been achieved in a previous segment. The algorithm finishes when $Q_i^*$ and $D_i^*$ are computed for all sessions which satisfy (29).

**Algorithm 2** Algorithm to evaluate $Q_i^*$ and $D_i^*$.

```
e_0 ← 0;
for ∀i, γ_i ← 0;
j ← 1;
E ←Set of all sessions;
{Evaluate s_j^i for all session i.}
loop
   for ∀i ∉ E, s_j^i ← ρ_i + H_i ψ_i e_{j-1}^{H_i - 1};
   for ∀i ∈ E, s_j^i ← φ_i / (Σ_{k∈E} φ_k) (1 − Σ_{k∉E} s_k^i);
   C ← { i ∈ E | s_j^i > ρ_i }
   if C = ∅ then
      break; {No more flows can be bounded.}
   end if
   {Determine which session finishes its busy period first.}
   for ∀i ∈ C do
      Find t_i s.t. (ρ_i − s_j^i) t_i + s_j^i e_{j-1} − γ_i + ψ_i t_i^{H_i} = 0;
   end for
   e_j ← min_i t_i;
   E ← E − arg min_i t_i
   {For all i ∈ C, evaluate Q_i^* and D_i^* if possible. }
   for ∀i ∈ C do
      t^* ← ((s_j^i − ρ_i) / (ψ_i H_i))^{1/(H_i − 1)};
      t^* ← max (t^*; e_{j-1});
      {Evaluate Q_i^* and D_i^*, if t^* ∈ [e_{j-1}; e_j). }
      if t^* < e_j then
         Q_i^* ← (ρ_i − s_j^i) t^* − γ_i + s_j^i e_{j-1} + ψ_i t^{*H_i};
         D_i^* ← φ_i / (Σ_{k=1}^N φ_k) Q_i^*;
      end if
      γ_i ← γ_i + s_j^i (e_j − e_{j-1});
   end for
   j ← j + 1;
end loop
```

## 4.4 Numerical example

In this section, the computation of an individual session delay via Algorithm 2 is illustrated. The accuracy of the estimated delay is compared to the delay obtained via simulation.

Fig. 2, shows the delay of three sessions as a function of server utilization, which can be varied by changing the service rate. The traffic parameters and GPS scheduling weights are shown in Table 1. Simulation results were obtained using traces which are $10^6$ samples long and greedy from time zero.

The bound resulting from a constant rate service (Fig. 2) is obtained when the $i^{th}$ flow is served at the minimum service rate $g_i$ (2). Such a bound is a loose one obtained by assuming that there is no statistical multiplexing.

Given the traffic parameters and GPS scheduling weights shown in Table 1, the session which finishes its busy period first is Session 3. Since all sessions are greedy from $t = 0$, they are all busy during the busy period of Session 3. Thus, Session 3 is effectively serviced at a constant rate, equal to the minimum rate, until the end of this busy period. For this reason, the bounds from constant rate service and the service curve analysis (given by Algorithm 2) coincide.

The next session to finish its busy period is Session 2. For this session, the bound obtained by applying the service curve analysis can be seen to be tighter than that resulting from a constant rate service. This becomes even more obvious for the bounds for Session 1, which is the last to finish its busy period. It is thus possible to conclude that the Service Curve bounds given by Algorithm 2 are valid and that they can actually be much tighter than the constant rate bounds, since they take the work-conserving nature of GPS scheduling into consideration.

Table 1: Traffic parameters used in the example.

| Session | $\rho_i$ | $\sigma_i$ | $k_i$ | $H_i$ | $\phi_i$ |
|---|---|---|---|---|---|
| 1 | 0.20 | 0.15 | 5.25 | 0.85 | 0.215 |
| 2 | 0.40 | 0.30 | 5.25 | 0.75 | 0.430 |
| 3 | 0.30 | 0.20 | 5.25 | 0.70 | 0.355 |

# 5 Performance analysis of a network of GPS servers with LRD traffic

## 5.1 Network model

In this section, an analysis of networks of GPS servers with arbitrary topology is carried out. This network is assumed to be represented as a directed graph, with nodes representing switches and arcs representing links. Each switch $m$ is composed of a GPS server with a rate of $r^m$, and a set of independent queues, one for each session served by the server $m$. Let $N$ be the number of sessions that arrive at the network and $n_m$ be the number of sessions served by the server $m$. The path taken by a session $i$ is denoted by $P(i)$, and $K_i$ is the total number of nodes along it.

For each session $i$, the amount of traffic that enters the network during the interval $[\tau; t]$ is denoted by $\mathbf{A}_i(\tau, t)$, with a sample-path realization of $A_i(\tau, t)$. The amount of session $i$ traffic served by the $k^{th}$ node in $P(i)$ during the interval $[\tau; t]$ is represented by $\mathbf{S}_i^{(k)}(\tau, t)$, $k = 1, \ldots, K_i$ , and $S_i^{(k)}(\tau, t)$ is the corresponding sample-path realization. To simplify the notation, $\tau$ will be omitted whenever it is equal to zero.

The set of sessions served by the server $m$ is represented by $I(m)$. For every session $i \in I(m)$, $\mathbf{A}_i^m(\tau, t)$ represents the amount of session $i$ traffic that arrives at the server $m$ during the interval $[\tau; t]$, and $\mathbf{S}_i^m(\tau, t)$ is the amount that leaves it during the same interval. The corresponding sample-path realizations are $A_i^m(\tau, t)$ and $S_i^m(\tau, t)$, respectively.

In the rest of this paper, the following conditions are assumed to hold for every session $i$:

1. The traffic is constrained to $P(i)$ .
2. The traffic is regulated by a FLB mechanism before entering the network.

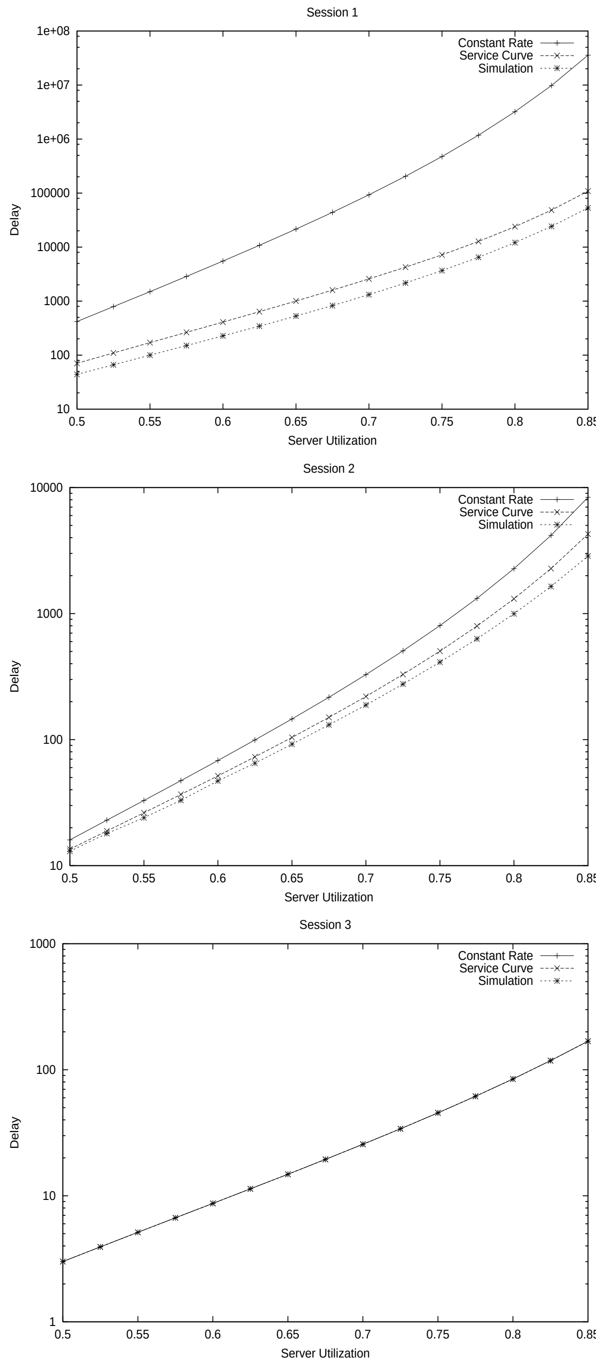


Figure 2: Examples of bounds on individual session delays.

Since the traffic entering the network is regulated by the FLB mechanism, it can be represented by the fBm envelope process (5). This fact will be denoted by $A_i(t) \sim \widehat{A}_i(t)$. It is also necessary to characterize $A_i^m(t)$ as a function of FLB parameters. Section 4.2 showed that $S_i^m(t)$ can be represented by the envelope process associated with $A_i^m(t)$. If this reasoning is recursively applied to all nodes in $P(i)$, the following lemma can be established:

**Lemma 5.1.** *If the envelope process $\widehat{A}_i(t) = \rho_i t + k_i \sigma_i t^{H_i}$ represents the session i traffic that arrives in the network, denoted by $A_i(t)$, then it also represents the amount of session i traffic that arrives at the node m for all $m \in P(i)$, i.e. $A_i^m(t) \sim \widehat{A}_i(t)$.*

Finally, it is necessary to define the bounds for individual session delays and backlogs for a network of GPS servers. Let $Q_i^m(t) = A_i^m(t) - S_i^m(t)$ be the sample-path realization of the session $i$ backlog process at the node $m \in P(i)$. By definition, $Q_i^m(t) = 0, \forall t \leq 0$. The total session $i$ backlog is given by $Q_i(t) = \sum_{m \in P(i)} Q_i^m(t)$, and $D_i(t) = \{d(t) : A_i(t) - S_i^{(K_i)}(t + d(t)) = 0\}$ is the delay introduced for an infinitesimal amount of session $i$ traffic which arrives in the network at the time $t$. The bounds on end-to-end delay and backlog can then be stated as

$$Q_i^* = \max_t \sum_{m \in P(i)} A_i^m(t) - S_i^m(t), \tag{34}$$

$$D_i^* = \arg\max_{d(t)} \left\{ d(t) : A_i(t) - S_i^{(K_i)}(t + d(t)) = 0 \right\}. \tag{35}$$

## 5.2 Stability issues

### 5.2.1 Locally stable sessions

Both (34) and (35) are hard to solve for networks of GPS servers with arbitrary topology, because the service offered for any session $i$ does not depend exclusively on its traffic parameters, but also on the traffic of all other sessions that are served at the nodes along $P(i)$. In some cases, however, it is possible to derive loose bounds. At server $m$, a minimum rate for each session $i \in I(m)$ is guaranteed:

$$g_i^m = \frac{\phi_i^m}{\sum_{j \in I(m)} \phi_j^m} r^m,$$

where $\phi_i^m$ is the GPS weight associated with the session $i$ at the node $m$. Thus, the minimum service rate guaranteed along the path of the session $i$ is the following:

$$g_i = \min_{m \in P(i)} g_i^m.$$

In [3] it was shown that, if the session $i$ is Leaky Bucket constrained and $g_i$ is greater than the mean declared rate, the bounds for the solutions of (34) and (35) can be obtained. Sessions for which these conditions hold are defined as locally stable, because their stability is guaranteed, even if the network as a whole is unstable. Similar results can be obtained when the session $i$ is FLB regulated, as is proved by the following theorem:

**Theorem 5.2.** *If $g_i > \rho_i$ for any session i that is regulated by the FLB mechanism,*

$$Q_i^* = (g_i - \rho_i)^{\frac{H_i}{H_i - 1}} (k_i \sigma_i)^{\frac{1}{1-H_i}} H_i^{\frac{H_i}{1-H_i}} (1 - H_i), \tag{36}$$

$$D_i^* = \frac{1}{g_i} \left[ (g_i - \rho_i)^{\frac{H_i}{H_i - 1}} (k_i \sigma_i)^{\frac{1}{1-H_i}} H_i^{\frac{H_i}{1-H_i}} (1 - H_i) \right]. \tag{37}$$

*Proof.* Equations (36) and (37) are obtained directly from (10) and (32), respectively. □

### 5.2.2 Network stability for locally non-stable sessions

Whenever all sessions in the network are locally stable, the stability of the whole network is assured. In some situations, however, it may be desirable for only a subset of those sessions to be locally stable. For instance, one may decide to allow $g_i$ to be greater than $\rho_i$ only for sessions that are admitted under guaranteed performance service and let $g_i$ be less than $\rho_i$ for sessions that are admitted under best-effort service. Although the latter sessions are not locally stable, under certain conditions, they will become stable due to the work-conserving nature of GPS scheduling.

Although admitting sessions that are not locally stable into a network may lead to feedback effects that can drive the whole network towards instability, such effects may be due to inconsistent treatment of sessions inside the network [3]. In order to provide end-to-end delay and backlog bounds for all sessions in a GPS network, it is necessary to establish the conditions under which such inconsistent treatment will not occur so that network stability can be assured.

Stability conditions for networks of GPS servers can be established using the class of GPS assignments known as *Consistent Relative Session Treatment* (CRST), if the sessions are Leaky Bucket regulated. Moreover, it is possible to generalize these results when the traffic is FLB regulated. The following definitions are provided in [3] :

**Definition 5.3.** Session $j$ is said to *impede* a session $i$ at a node $m$ if

$$\frac{\phi_i^m}{\phi_j^m} < \frac{\rho_i}{\rho_j}.$$

**Definition 5.4.** The *Consistent Relative Session Treatment* (CRST) is a GPS assignment for which there exists a strict ordering of the sessions such that, for any two sessions $i$, $j$, if the session $i$ is prior to the session $j$ in order, then the session $i$ does not impede the session $j$ at any node of the network.

It is possible to partition all sessions in a network into a set of nonempty classes $H_1, \ldots, H_L$ such that the sessions in $H_k$ are impeded only by those in $H_l$, $l < k$. This partition process is useful to characterize the traffic inside the network and prove its stability [3]. In the present scenario, since all sessions are FLB constrained, the traffic inside the network can be characterized by using Lemma 5.1, and the partitioning process is not necessary to prove the stability of networks under CRST.

**Theorem 5.5.** *Suppose that the sessions $i$ and $j$ contend for a link at the node $m$, and that the session $j$ is stable. If this session $j$ does not impede the session $i$ at that node, the session $j$ traffic cannot drive the session $i$ toward instability at that node.*

*Proof.* Let $\tau_i^m$ be the time at which the backlog of the session $i$ achieves its maximum value and $e_i^m$ be the time at which the backlog is completely consumed, as defined in Section 4. Obviously, $e_i^m \geq \tau_i^m$. If $e_j^m$ denotes the time at which the backlog of the session $j$ is completely consumed, there are two possibilities:

1. $e_i^m \leq e_j^m$.

   In this case, the session $j$ finishes its busy period after the session $i$. Since the total amount of service offered to the session $i$ under the greedy regime depends only on sessions that finish their busy periods before this session, its stability does not depend on the session $j$.

2. $e_i^m > e_j^m$.

   In this case, the session $j$ finishes its busy period before the session $i$. Since the total amount of service offered to the session $i$ under the greedy regime depends on all sessions that finish their busy periods before it does, it is possible that this session $i$ is unstable at the node $m$, which certainly depends on the session $j$ traffic. However, since it is assumed that the session $j$ does not impede the session $i$ at the node $m$, it is possible to prove that the session $i$ can be stable, regardless of traffic of the session $j$. If the session $j$ is stable, then

   $$S_j^m\left(0; e_j^m\right) > \rho_j e_j^m \geq \frac{\rho_i \phi_j^m}{\phi_i^m} e_j^m. \tag{38}$$

   The preceding inequality is due to the fact that the session $j$ does not impede the session $i$. Substituting (38) for $S_j(.)$ in (1) gives

   $$S_i^m\left(0; e_j^m\right) = \frac{\phi_i^m}{\phi_j^m} S_j^m\left(0; e_j^m\right) > \rho_i e_j^m.$$

   Thus, the stability of the session $i$ does not depend on the traffic of the session $j$ , i.e. , if the session $i$ is in fact unstable, this instability is not due to the traffic of the session $j$.

□

From Theorem 5.5, it is possible to conclude that only sessions that impede the session $i$ at any node $m \in P(i)$ can drive the session $i$ toward instability. Clearly, the session $i$ would experience a feedback effect if it were impeded by a session $j$ at a node $m_1$ and it impeded the session $j$ at another node $m_2$. In this case, the amount of service offered to the session $i$ would depend on the service offered to the session $j$, which, in turn, would depend on the service offered to the session $i$ at another node. This effect can be completely eliminated if a CRST assignment is employed — in this case, the stability of a session depends only on its own traffic, as well as on the stability of the sessions that impede it at any node along its path.

Of course, there is no point in having sessions in the network for which the backlog and the delay cannot be bounded. It is thus important to establish a general condition under which the stability of the network is assured. In order to do this, the following definition and lemmas are introduced:

**Definition 5.6.** The utilization of the the server $m$ is given by:

$$u^m = \frac{1}{r^m} \sum_{j \in I(m)} \rho_j.$$

**Lemma 5.7.** *If $u^m < 1$ and if all sources are FLB regulated, all sessions that contend for a link at the node $m$ are stable at that node.*

Due to the work-conserving nature of GPS scheduling, the total backlog at the node $m$ is given by

$$Q^m(t) = \sum_{j \in I(m)} Q_j^m(t) = \left( \sum_{j \in I(m)} \rho_j t + k_j \sigma_j t^{H_j} \right) - r^m t.$$

The total backlog asymptotically approaches the following:

$$\begin{aligned} \lim_{t \to \infty} Q^m(t) &= \lim_{t \to \infty} \left( \sum_{j \in I(m)} \rho_j t + k_j \sigma_j t^{H_j} \right) - r^m t \\ &= \lim_{t \to \infty} \left( -r^m + \sum_{j \in I(m)} \rho_j + k_j \sigma_j t^{H_j - 1} \right) t \\ &\sim \lim_{t \to \infty} \left( -r^m + \sum_{j \in I(m)} \rho_j \right) t, \end{aligned}$$

since $H_j < 1$, for $\forall j$. If $u^m < 1$, $\lim_{t \to \infty} Q^m(t) \to -\infty$, which indicates that all sessions are stable.

**Lemma 5.8.** *If a network of GPS servers operates under CRST assignment, a session $i$ will be stable if $u^m < 1$, $\forall m \in P(i)$.*

**Corollary 5.9.** *A network of GPS servers that operates under CRST assignment will be stable if $u^m < 1$, $\forall m$.*

## 5.3 Computing delay and backlog for stable networks with LRD traffic

Consider a stable network of GPS servers that operates under CRST assignment, with sessions that are FLB constrained. In Section 5.2.1, bounds on individual session backlog and delay were obtained for a particular class of sessions known as locally stable sessions. These bounds incorporate the dependency among queues at the nodes in a session path. Thus, it can be shown that they are much tighter than those that would be obtained by simply summing up the bounds on backlog and delay at each node of the network.

Nevertheless, these bounds do not take the dependence among sessions inside the network into consideration, nor are they valid for sessions that are not locally stable. In [3], an algorithm was introduced to obtain bounds on individual session delays and backlogs for all sessions inside a stable GPS-CRST network. Although this algorithm was obtained considering all sources to be Leaky Bucket constrained, the results are also valid when the traffic is FLB regulated.

For the sake of simplicity, a target session $i$ is considered. The path of traffic flow for this session is $P(i) = \{1, 2, \ldots, K\}$, and $\widehat{A}_i(t)$ is the corresponding envelope process. Lemma 5.1 states that $A_i^m(t) \sim \widehat{A}_i(t)$ for all $m \in P(i)$. Additionally, it is assumed that the following assumptions hold:

1. Every session $j \in I(m) - \{i\}$ for all $m \in P(i)$ is free to send traffic in any manner as long as $A_j^m(t) \sim \widehat{A}_j(t)$.
2. Session $i$ traffic is constrained to $P(i)$.

These assumptions constitute the independent session relaxation [3]. The values of $D_i^*$ and $Q_i^*$ obtained under these assumptions are upper bounds on the solutions for (30) and (31); they can be evaluated precisely by considering that all sessions are greedy, i.e., they send as much traffic as possible so that $A_j(t) = \widehat{A}_j(t)$. Although such reasoning was valid for the single-node analysis presented in Section 4, where all sources were assumed to be greedy from time zero, this assumption does not hold for multiple-node cases. Parekh and Gallager [3] showed that what is necessary for this multiple-node cases is that the traffic follow a staggered greedy regime, in which sessions at the node $m$ simultaneously become greedy, but only after sessions at the node $m-1$ do. Additionally, they showed that $D_i^*$ and $Q_i^*$ may be obtained under different staggered greedy regimes. This condition was illustrated for Leaky Bucket constrained sessions in [3, Fig. 4]. In order to avoid complications due to the multiplicity of greedy regimes, Parekh and Gallager [3] propose the use of a single function, the Universal Service Curve, from which both $D_i^*$ and $Q_i^*$ can be precisely evaluated using the independent session relaxation. In fact, this approach is an extension of the one adopted for the single-node analysis in Section 4, where this curve is $\widehat{S}_i(t)$.

The same reasoning can also be used for FLB constrained sessions, where the fact that $D_i^*$ and $Q_i^*$ can be achieved under different staggered greedy regimes is not as important as it is in the case of Leaky Bucket constrained sessions. In this case, the lack of a mathematical expression for the inverse function of $\widehat{A}_i(t)$, which poses difficulties for the evaluation of $D_i^*$ for the single-node case, is an advantage: it is only necessary to establish the staggered greedy regime to evaluate $Q_i^*$, from which the bounds for $D_i^*$ can be established using (32). Since $Q_i^*$ can also be evaluated precisely from the Universal Service Curve, this will be used instead of the staggered greedy regime corresponding to $Q_i^*$, thus maintaining an analysis which is consistent with that proposed in [3].

### 5.3.1 The Universal Service Curve

The Universal Service Curve for the session $i$ is obtained as follows. Let $\widehat{S}_i^1, \ldots, \widehat{S}_i^K$ be session $i$ service curves which are obtained for each node $1, \ldots, K$ when session $i$ traffic entering that node is represented by $\widehat{A}_i(t)$ (from Lemma 5.1 , $A_i^m(t) \sim \widehat{A}_i(t)$), and the single-node analysis given in Section 4 is applied. For each node $m \in P(i)$, $\widehat{S}_i^m$ is a continuous, piecewise linear and strictly crescent function in the interval $[0; e_i^m]$, where $e_i^m$ is the duration of the busy period of the session $i$ at the node $m$ when all sources at that node are simultaneously greedy from time zero. Denoting the slope of the $j^{th}$ segment of $\widehat{S}_i^m$ as $s_j^{i,m}$, and its duration as $d_j^{i,m}$, a set $E_i^m$ can be defined as

$$E_i^m = \left\{ \left( s_1^{i,m}; d_1^{i,m} \right) \left( s_2^{i,m}; d_2^{i,m} \right) \ldots \left( s_{n_m}^{i,m}; d_{n_m}^{i,m} \right) \right\},$$

where $n_m$ is the number of segments within the interval $[0; e_i^m]$. Let $E_i$ be the union of the sets $E_i^m$ for all $m \in P(i)$:

$$E_i = \bigcup_{m \in P(i)} \bigcup_{j=1}^{n_m} \left\{ \left( s_j^{i,m}; d_j^{i;m} \right) \right\}. \tag{39}$$

The curve $G_i(t)$ is thus the piecewise linear curve obtained by juxtaposing the elements from $E_i$ after they have been sorted in order of increasing slope. The Universal Service Curve for the session $i$ is then defined as

$$U_i(t) = \min \left\{ G_i(t); \widehat{A}_i(t) \right\}.$$

Fig. 3 illustrates the construction of $U_i(t)$ for the case of FLB constrained sessions. It is easy to verify that the curve $G_i(t)$ always intersects $\widehat{A}_i(t)$ if the network is stable. In fact, Parekh and Gallager [3] have proved that $G_i(t)$ intersects $\widehat{A}_i(t)$ at or before $\sum_{m\in P(i)} e_i^m$, if sources are Leaky Bucket constrained. Since their proof is also valid for the case of FLB constrained traffic, the following lemma can be established:

**Lemma 5.10.**

$$G_i\left(\sum_{m\in P(i)} e_i^m\right) \geq \widehat{A}_i\left(\sum_{m\in P(i)} e_i^m\right).$$

Given this lemma, at least one solution for $Q_i^*$ (and $D_i^*$) can be found. Moreover, it is possible that not all segments from $E_i$ need to be used to construct $U_i(t)$, since $G_i(t)$ can intersect $\widehat{A}_i(t)$ before $t = \sum_{m\in P(i)} e_i^m$. The following lemma can also be established:

**Lemma 5.11.** *For every session $i$,*

$$Q_i^* \leq \max_{t\geq 0}\left\{\widehat{A}_i(t) - U_i(t)\right\}, \tag{40}$$

*and $D_i^*$ can be evaluated using the approximation given by (32).*

Since $U_i(t) = \min\left\{G_i(t); \widehat{A}_i(t)\right\}$, (40) can be rewritten as $Q_i^* \leq \max_{t\geq 0}\left\{\widehat{A}_i(t) - G_i(t)\right\}$. This inequality is valid for Leaky Bucket constrained sessions [3, Theorem 3]. The corresponding proof can be applied to show that the inequality is also valid for FLB constrained sessions.

Lemmas 5.10 and 5.11 lead to the following theorem:

**Theorem 5.12.** *If the optimization problem represented by (40) is solvable, then its solution is unique.*

*Proof.* Consider that the elements in the set $E_i$ are sorted in order of increasing slope, so that $s_j^i$ is the slope of the $j^{th}$ element in the sorted set and $e_j^i = \sum_{k=1}^{j} d_k^i$, where $d_j^i$ is the duration of the $j^{th}$ segment in the sorted set. In the interval for which the curve $U_i(t)$ is lower than $\widehat{A}_i(t)$, it can be written as

$$U_i(t) = t\cdot\sum_{j=0}^{N}\xi_j^i u(t-e_j),$$

where $\xi_j^i = s_{j-1}^i - s_j^i$ and $u(.)$ is the step function. In the interval $\left[0; t_i^0\right]$ all terms $\xi_j^i$ are positive; therefore,

$$U_i(t) = t\cdot\sum_{j=0}^{i-1}\xi_j^i u(t-e_j).$$

Then, the backlog $Q_i(t)$ is

$$Q_i(t) = \widehat{A}_i(t) - t\cdot\sum_{j=0}^{i-1}\xi_j^i u(t-e_j).$$

The first and the second derivatives of $Q_i(t)$ in this interval are the following:

$$\frac{dQ_i}{dt}(t) = \frac{d\widehat{A}_i}{dt}(t) - \sum_{j=0}^{i-1} \xi_j^i \left[e_j \delta(t-e_j) + u(t-e_j)\right],$$

and

$$\frac{d^2Q_i}{dt^2}(t) = \frac{d^2\widehat{A}_i}{dt^2}(t) - \sum_{j=0}^{i-1} \xi_j^i \left[e_j \delta^2(t-e_j) + \delta(t-e_j)\right],$$

where $\delta(.)$ and $\delta^2(.)$ are the Dirac delta function and its derivative, respectively.The second derivative of $\widehat{A}_i(t)$ is equal to $\psi_i H_i(H_i-1)t^{H_i-2}$, which is always negative for $t > 0$. Since all $\xi_j^i$ are positive for $j < i$, the second derivative of $Q_i(t)$ is always negative for $0 < t < e_i$. This condition is sufficient to guarantee that, if there is a solution to (40) in that interval, it is unique. □

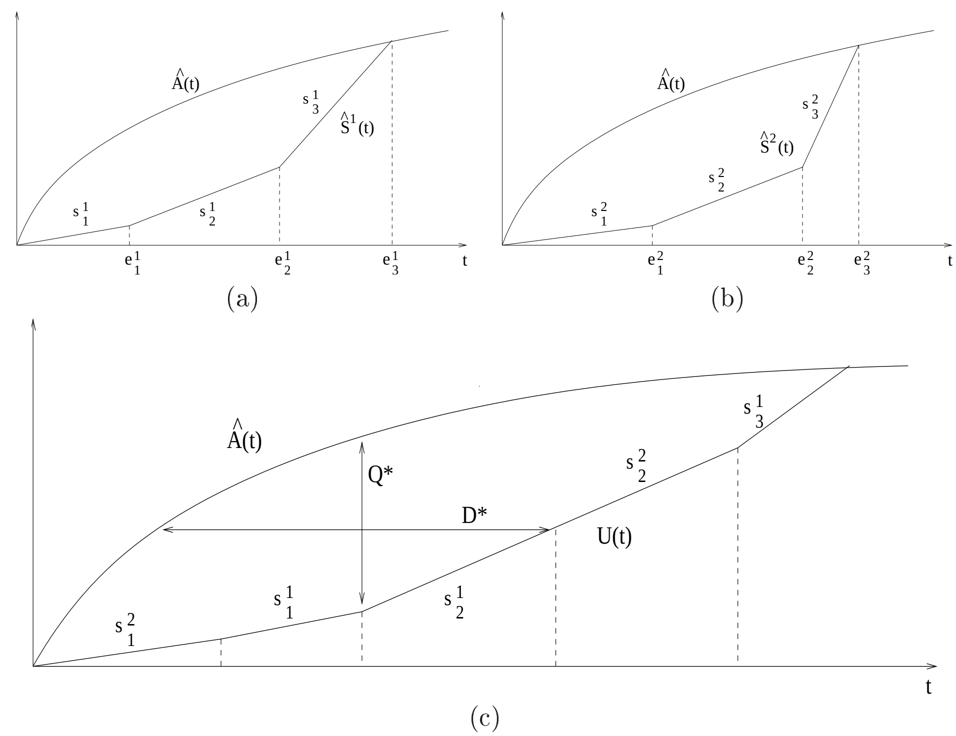


Figure 3: Example of construction of $U_i(t)$ supposing that $P(i)$ has two nodes. (a) Analysis of the first node. (b) Analysis of the second node. (c) Construction of $U_i(t)$ and definition of $Q_i^*$ and $D_i^*$.

## 5.4 An algorithm for computing $Q_i^*$ and $D_i^*$

The computational algorithm to obtain upper bounds for the end-to-end backlog and delay in a network of GPS servers under CRST assignments is presented in Algorithm 3. This algorithm is able to obtain bounds for all sessions in the network, an advantage in relation to the results of the approach proposed by Zhang *et al.* in [8], which assumes that the network operates under a subclass of CRST assignment for which all sessions are locally stable.

This algorithm is divided into two steps; it operates as follows. In the first step, all slopes $s_j^{i,m}$ and durations $d_j^{i,m}$ are computed for all nodes $m$ and for all sessions $i \in I(m)$. For each node $m$, the computation of the slopes is similar to that of the single-node case, i.e., the slope of the session $i$ depends on the situation at that node in relation to whether the busy period of session $i$ has already finished. The algorithm then determines which busy session is the first to finish its busy period at the node $m$. This information is used to determine the time at which the $j^{th}$ segment of $\widehat{S}_i(t)$ at the node $m$ finishes ($e_j^m$), which, in turn, is used to compute $d_j^{i,m}$.

In the second step of the algorithm, the end-to-end backlog and delay bounds are computed for all sessions in the network. For each session $i$, a set of segments $E_i$, in agreement with (39), is defined. Segments are taken from this set in order of increasing slope and, for each segment, the maximum backlog is identified. If this backlog is achieved in the segment, $Q_i^*$ and $D_i^*$ are computed, and the procedure is repeated for another session. The algorithm finishes when $Q_i^*$ and $D_i^*$ have been computed for all sessions in the network.

## 5.5 Numerical example

A numerical example of the application of Algorithm 3 is presented in this section. In this example, the end-to-end delay bound is compared to the actual delay obtained via simulation, and the validity of this bound is verified.

Fig. 4 presents a tandem network consisting of four nodes. Three different flows (F1, F2 and F3) enter the network at Node 1, pass through Nodes 2 and 3, and leave the network after Node 4. At Node 2, two interfering flows (F4 and F5) enter the network. Flow F4 leaves the network after Node 3, and Flow F5 leaves it after Node 2. At Node 3, another interfering flow (F6) enters the network, and leaves it after the same node. The traffic parameters and GPS scheduling weights for all the flows are shown in Table 2.

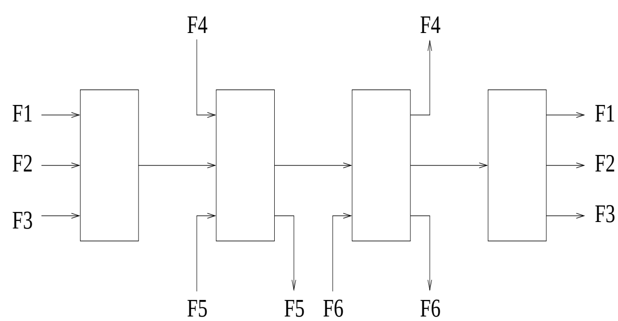


Figure 4: Example network.

Fig. 5 shows the end-to-end delay of flows F1, F2 and F3 as a function of the utilization of the links of the network, which varies according to changes in the service rate at each node. Simulation results were obtained using traces which are $10^6$ samples long and greedy from time zero. The bounds obtained using Algorithm 3 are found to be valid and close to the delay obtained via simulation.

**Algorithm 3** Algorithm to evaluate $Q_i^*$ and $D_i^*$ using the Universal Service Curve.

**for all** node $m$ **do**
  $e_0^m \leftarrow 0$;
  **for** $\forall i \in I(m),\ \gamma_i^m \leftarrow 0$;
  $j \leftarrow 1$;
  $\mathbf{V} \leftarrow$Set of sessions in $m$;
  {Evaluate $s_j^{i,m}$ for all session $i \in I(m)$.}
  **loop**
    **for** $\forall i \notin \mathbf{V},\ s_j^{i,m} \leftarrow \rho_i + H_i \psi_i e_{j-1}^{H_i - 1}$;
    **for** $\forall i \in \mathbf{V},\ s_j^{i,m} \leftarrow \frac{\phi_i^m}{\sum_{k \in \mathbf{V}} \phi_k^m} \left( r^m - \sum_{k \notin \mathbf{V}} s_k^{i,m} \right)$;
    $\mathbf{C} \leftarrow \left\{ i \in \mathbf{V} | \, s_j^{i,m} > \rho_i \right\}$
    **if** $\mathbf{C} = \emptyset$ **then**
      break;
    **end if**
    {Determine which session finishes service first.}
    **for** $\forall i \in \mathbf{C}$ **do**
      Find $t_i$ s.t. $\left( \rho_i - s_i^{i,m} \right) t_i + s_j^{i,m} e_{j-1}^m - \gamma_i^m + \psi_i t_i^{H_i} = 0$;
    **end for**
    $e_j^m \leftarrow \min_i t_i$;
    **for** $\forall i \in \mathbf{C}$ **do**
      $d_j^{i,m} \leftarrow e_j^m - e_{j-1}^m$;
      $\gamma_i \leftarrow \gamma_i + s_j^{i,m} d_j^{i,m}$;
    **end for**
    $\mathbf{V} \leftarrow \mathbf{V} - \{\arg\min_i t_i\}$;
    $j \leftarrow j + 1$;
  **end loop**
**end for**
{Evaluate bounds on $Q_i^*$ and $D_i^*$ for every }
{session $i$ in the network.}
**for all** session $i$ **do**
  $E_i = \bigcup_{m \in P(i)} \bigcup_j \left\{ \left( s_j^{i,m}; d_j^{i;m} \right) \right\}$
  $\gamma_i \leftarrow 0$; $\tau \leftarrow 0$;
  **loop**
    $j \leftarrow$Element whose slope is the lowest in $E_i$;
    $(s_j; d_j) \leftarrow$Slope and duration of the element $j$;
    Remove element $j$ from $E_i$;
    $t^* \leftarrow \left( \frac{s_j - \rho_i}{\psi_i H_i} \right)^{\frac{1}{H_i - 1}}$
    $t^* \leftarrow \max(\tau, t^*)$;
    **if** $t^* < \tau + d_j$ **then**
      $Q_i^* \leftarrow (\rho_i - s_j) t^* - \gamma_i + s_j \tau + \psi_i t^{*H_i}$;
      $D_i^* \leftarrow Q_i^* \min_{m \in P(i)} \frac{\phi_i^m}{\sum_{k=1}^N \phi_k^m}$;
      break; {Execute for next session.}
    **else**
      $\gamma_i \leftarrow \gamma_i + s_j d_j$;
      $\tau \leftarrow \tau + d_j$;
    **end if**
  **end loop**
**end for**

Table 2: Traffic parameters used in the example.

| Flow | $\rho_i$ | $\sigma_i$ | $k_i$ | $H_i$ | $\phi_i$ |
|---|---|---|---|---|---|
| F1 | 0.20 | 0.15 | 5.25 | 0.85 | 0.20 |
| F2 | 0.40 | 0.30 | 5.25 | 0.75 | 0.40 |
| F3 | 0.30 | 0.20 | 5.25 | 0.70 | 0.30 |
| F4 | 0.35 | 0.20 | 5.25 | 0.80 | 0.35 |
| F5 | 0.15 | 0.10 | 5.25 | 0.75 | 0.15 |
| F6 | 0.20 | 0.15 | 5.25 | 0.65 | 0.20 |

## 6 Conclusions

Parekh and Gallager [2, 3] proved that GPS scheduling is able to guarantee delay bounds when it is fed by Leaky Bucket constrained streams. The Leaky Bucket mechanism, however, is not appropriate for the regulation of streams which exhibit long-range dependencies [6, 7].

This paper presents an algorithm for the computation of the delay and backlog bounds in a GPS server for traffic regulated by the FLB mechanism. Results obtained for an isolated GPS server were then applied for the analysis of a network of GPS servers with arbitrary topology fed by FLB constrained streams. Parekh and Gallager [3] showed that networks with Leaky Bucket constrained sessions, and which operate under a broad class of assignments known as Consistent Relative Session Treatment (CRST), are stable. Similar results have been proved to be valid for FLB constrained sessions. Moreover, an algorithm for the computation of delay and backlog bounds for networks of GPS servers operating under CRST was introduced for such traffic. Such computation is of paramount importance for providing delay bounds in networks of GPS servers, especially for a QoS-oriented Internet.

## References


[1] P. Dupuis and K. Ramanan, "A skorokhod problem formulation and large deviation analysis of a processor sharing model," tech. rep., Division of Applied Mathematics, Brown University, 1997.

[2] A. K. Parekh and R. G. Gallager, "A Generalized Processor Sharing approach to flow control in integrated services networks: the single-node case," *IEEE/ACM Trans. Networking*, vol. 1, pp. 344–357, June 1993.

[3] A. K. Parekh and R. G. Gallager, "A Generalized Processor Sharing approach to flow control in integrated services networks: the multiple-node case," *IEEE/ACM Trans. Networking*, vol. 2, pp. 137–150, Apr. 1994.

[4] K. Park and W. Willinger, eds., *Self-similar network traffic and performance evaluation*. New York: Wiley, 2000.

[5] I. Norros, "On the use of fractional brownian motion in the theory of connectionless networks," *IEEE J. Select. Areas Commun.*, vol. 13, pp. 953–962, Aug. 1995.

[6] N. L. S. Fonseca, G. S. Mayor, and C. A. V. Neto, "On the equivalent bandwidth of self-similar sources," *ACM Trans. on Modeling and Computer Simulation*, vol. 10, pp. 104–124, Apr. 2000.

[7] G. S. Mayor and J. A. Silvester, "Time scale analysis of an ATM queueing system with long-range dependent traffic," in *Proc. IEEE/INFOCOM*, pp. 205–212, 1997.

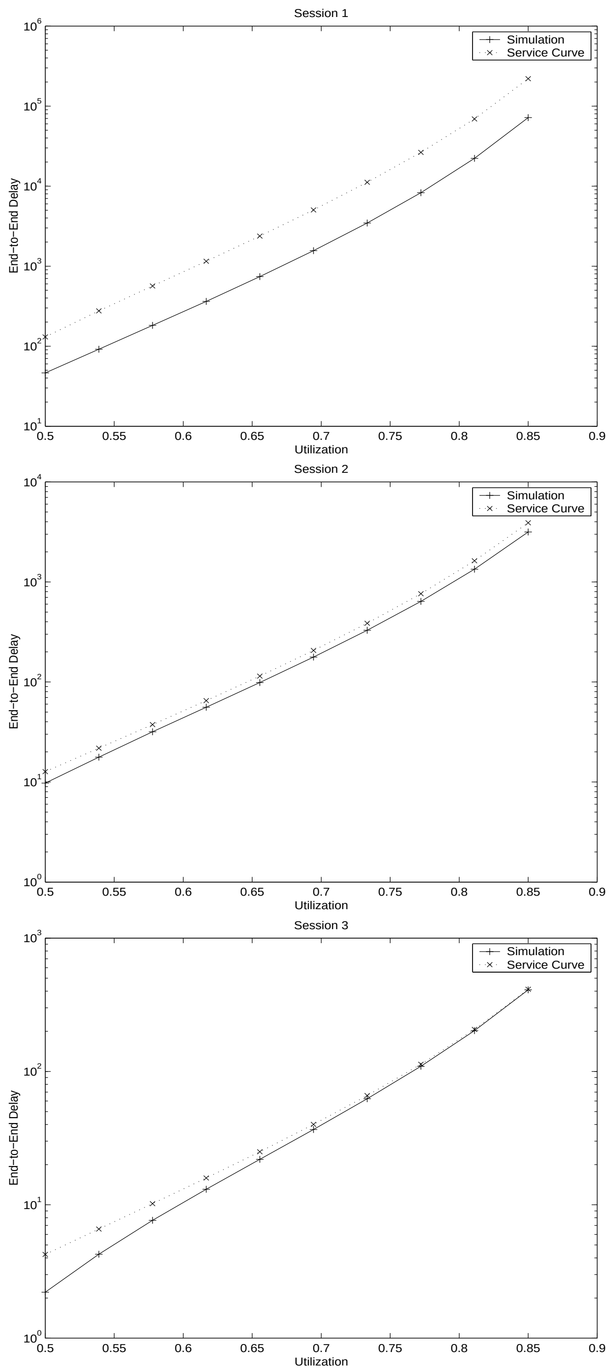


Figure 5: Examples of bounds on end-to-end delay of individual sessions.

[8] Z.-L. Zhang, D. Towsley, and J. Kurose, "Statistical analysis of the Generalized Processor Sharing scheduling discipline," *IEEE J. Select. Areas Commun.*, vol. 13, pp. 1071–1080, Aug. 1995.

[9] O. Yaron and M. Sidi, "Performance and stability of communication networks via robust exponential bounds," *IEEE/ACM Trans. Networking*, vol. 1, pp. 372–385, 1993.

[10] O. Yaron and M. Sidi, "Generalized Processor Sharing networks with exponentially bounded burstiness arrivals," in *Proc. IEEE/ INFOCOM*, pp. 628–634, 1994.

[11] S. Borst, O. Boxma, and P. Jelenković, "Generalized Processor Sharing with long-tailed traffic sources," in *Telletrafic engineering in a competitive world, Proc. ITC-16* (P. K. e D. Smith, ed.), pp. 345–354, 2000.

[12] S. Borst, O. Boxma, and P. Jelenković, "Asymptotic behavior of Generalized Processor Sharing with long-tailed traffic sources," in *Proc. IEEE/ INFOCOM*, pp. 912–921, 2000.

[13] M. van Uitert and S. Borst, "Generalised Processor Sharing networks fed by heavy tailed traffic flows," in *Proc. IEEE/INFOCOM*, 2001.

[14] P. Jelenković and P. Momčilović, "Network multiplexer with Generalized Processor Sharing and heavy-tailed on-off flows," in *Teletraffic Engineering in the Internet era, Proc. of ITC-17* (J. M. de Souza, N. L. Fonseca, and E. A. S. e Silva, eds.), Elsevier, 2001.

[15] N. G. Duffield and N. O'Connell, "Large deviation and overflow probabilities for the general single-server queue, with applications," *Proc. Cambridge Philosophical Society*, no. 118, pp. 363–374, 1995.

[16] R. Cohen, N. L. S. Fonseca, and M. Zukerman, "Traffic management and control," in *Multimedia Communications Networks* (M. Talipamula and B. Kasnabesh, eds.), Artech House, 1998.

[17] R. Cruz, "A calculus for network delay, part i : Network elements in isolation," *IEEE Trans. Inform. Theory*, vol. 37, pp. 114–131, Jan. 1991.

[18] R. Cruz, "A calculus for network delay, part ii : Network analysis," *IEEE Trans. Inform. Theory*, vol. 37, no. 1, pp. 132–141, 1991.