Tightening end to end delay upper bound for AFDX network calculus with rate latency FIFO servers using network calculus

Marc Boyer, Christian Fraboul
IRIT / ENSEEIHT – University of Toulouse
France

Abstract

This paper presents some new results in network calculus designed to improve the computation of end to end bounds for an AFDX network, using the FIFO assumption. The formal contribution is to allow to handle shaped leaky buckets flows traversing simple rate-latency network elements. Two new results are presented when a sum of some shaped flows crosses one single network element: the first one considers the global aggregated flow, and the second one considers each individual flow, assuming FIFO policy. Some configurations are tested, and the first results obtained are in almost all cases better than already known methods.

1. Context

The AFDX (Avionic Full Duplex Switched Ethernet) is the new embedded communication backbone of the civil avionic industry.

The AFDX network is based on the widely used Ethernet technology. To avoid the medium access control indeterminacy, it uses full duplex links, connected to (statically configured) switches. Then, the only indeterminacy comes from the delay in switch queues[1]. Moreover, the data flows are structured with the notion of *Virtual Link* (VL). A VL is a mono-sender multicast flow, with a throughput constraint, expressed by a minimal and maximal frame size, and a minimal inter frame delay, called "bag". A VL goes from switch to switch, up to the final receivers. One major goal to achieve certification of the system it to guarantee the end-to-end delay through the network for each data. Network calculus have been chosen to compute theses guaranteed bounds [1, 2, 3].

Network calculus is a theory designed to compute guaranteed upper bounds for delays and buffers size in networks. After the pioneer works of [4, 5], it had be widely studied [6, 7].

A simple modelling an AFDX network in network calculus is straightforward. The VL specification allow to describe the flow as constrained by a leaky bucket, where the maximal frame size is the bucket size, and the rate is this maximal frame size divided by the bag. The switches can be modelled by some rate-latency server. This modelling is also known as (σ, ρ)-calculus.

In [2, 3], such a simple model has been used, a tool has been written, and some delay bounds have been computed on the Airbus A380 configuration. But some of the computed bounds were not satisfying the application needs, opposed to the performances measured on test-beds [3].

Then, a tighter modelling have been done, taking into account the shaping introduced by the medium (on a link with bandwidth D there can not be any burst greater than the link bandwidth). But this modelling goes out of the well known (σ, ρ)-calculus, and some algorithms have been developed to handle this new aspect[2]. Then, taking into account the shaping allows to compute bounds up to 40% smaller than without modelling the shaping (for the A380 configuration) and the applicative delays have been certified.

Nevertheless, this method only computes local delays, *ie* the delays observed in each switch, and can not take into account the "pay burst only once" principle [6, § 1.4.3].

This article present some new analytical results, providing enhancement on the end to end computed bounds. Some configurations are tested, and the new results are in almost all cases better than the other methods.

This article is organised as follow: Section 2 briefly presents network calculus; Section 3 presents related works; Section 4 introduces a normal form for sum of shaped leaky buckets flows (Section 4.1), and gives some results on such flows traversing a rate-latency node (Section 4.2); Section 5 presents the main results of this works, the application of the result on aggregate shaped flows sharing a FIFO server; Section 6 takes some numerical configurations and compares the results of our new method and the previous ones. Section 7 concludes.

2. Network Calculus

Here is a (very short) introduction to network calculus. The reader should refer to [7, 6] for a complete presentation.

[1] And also from the negligible commutation time.

[2] The details of the methods are presented in Section 6.2.1.

978-1-4244-2350-7/08/$25.00 ©2008 IEEE.

Notations If X is a subset of $\mathbb{R}$, f, g two functions, $f =_X g$ means that $x \in X \Rightarrow f(x) = g(x)$. $\mathbb{R}_{\geq 0} \stackrel{\text{def}}{=} \{x \in \mathbb{R} \,|\, x \geq 0\}$ denotes the set of non negative reals. The function $1_{\{t>m\}}$ value is 1 if $t > m$ and 0 otherwise. $[x]^+ = max(x, 0)$. $x \wedge y = \min\{x, y\}$, $\bigwedge_{i=1}^{n} f_i = f_1 \wedge f_2 \ldots \wedge f_n$, and $\bigwedge_i f_i$ will sometimes be used as short-hand for $\bigwedge_{i=1}^{n} f_i$.

Network calculus is a theory to get deterministic upper bounds in networks. It is mathematically based on the $(\wedge, +)$ dioid. Three basic operators on functions are of interest, convolution $\otimes$, deconvolution $\oslash$ and the sub-additive closure $\overline{f}$.

$$(f \otimes g)(t) = \inf_{0 \leq u \leq t} (f(t-u) + g(u))$$
$$(f \oslash g)(t) = \sup_{0 \leq u} (f(t+u) - g(u))$$
$$\overline{f} = \delta_0 \wedge f \wedge (f \otimes f) \wedge (f \otimes f \otimes f) \wedge \cdots$$

A flow is represented by its cumulative function R, where $R(t)$ is the total number a bits send by this flow up to time t. A flow R is said to have a function α as *arrival curve* iff $R \leq R \otimes \alpha$. If α is an arrival curve for R, also is $\overline{\alpha}$. A server has a *service curve* β iff for all arrival flow, the relation between the input flow R and the output flow R', we have $R' \geq R \otimes \beta$. In this case, $\alpha' = \alpha \oslash \beta$ is an arrival curve for R'. The delay experimented by the flow R can be bounded by the maximal horizontal difference between curves α and β, formally defined by $h(\alpha, \beta)$ (a graphical interpretation of h can be found in Figure 7).

$$h(\alpha, \beta) = \sup_{s \geq 0} \left(\inf \{\tau \geq 0 \,|\, \alpha(s) \leq \beta(s + \tau)\} \right)$$

They are some common arrival and service curves, δ_d, λ_R, $\beta_{R,T}$, $\gamma_{r,b}$, defined by: $\delta_d(t) = 0$ if $t \leq d$, ∞ otherwise, $\lambda_R(t) = Rt$, $\beta_{R,T}(t) = R[t-T]^+$, and $\gamma_{r,b}(t) = rt + b$ if $t > 0$, 0 otherwise.

One of the most popular result of network calculus is the ability to handle the "pay burst only once" phenomenon [6, § 1.4.3]. In a network, when a flow goes through two network elements in sequence, it is well known that the maximal end to end delay is less than the sum of the delays in each component. In network calculus, the sequence of two network elements S, S' of service curves β, β' (Fig 1) is equivalent to a single network element with service curve $\beta \otimes \beta'$. And it can be shown that, for a flow R with arrival curve α, and with $\alpha' = \overline{\alpha \oslash \beta}$, equation (1) holds, which mean that the end to end delay $h(\alpha, \beta \otimes \beta')$ is less than the sum of the two local delays.

$$h(\alpha, \beta \otimes \beta') \leq h(\alpha, \beta) + h(\alpha', \beta') \tag{1}$$

3. Related works

The leaky bucket is a very common model for specifying traffic constraint. In network calculus, such a constraint is modelled by a $\gamma_{r,b}$ arrival curve. Since this class

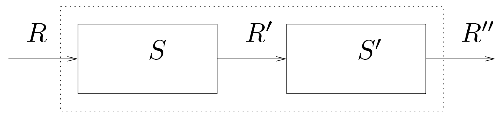

Figure 1. A flow going through two network elements in sequence

of curve has a lot of good properties[3] it had been deeply studied, and some well known results can be applied.

The core idea of [3, 1, 2] is to take into account the shaping introduced by the medium (on a link with bandwidth D there can not be any burst greater than the link bandwidth). But this modelling goes out of the well known $\gamma_{r,b}$ class of curve. A VL constrained (*ie* shaped) by a link is modelled by a $\lambda_D \wedge \gamma_{r,b}$ curve, and the sum of such shaped VL traversing a switch is under the general form $\bigwedge_{i=1}^{n} \gamma_{r_i,b_i}$.

This kind of curve have been used in [8] as an approximation of video traffic arrival curve for admission control, and in [9], where it is called "concave piecewise linear (PCL)" curve, to compute some schedulability condition. But they were not able to compute the deformation induced by traversing a server.

[10] is also interested in an end to end delay computation taking into account the FIFO policy of some rate-latency servers, but is only deals with $\gamma_{r,b}$ arrival curves.

In [11], a very general framework of piecewise linear function is defined, and the results in Section 4.2 are subsumed by they results. They are presented only because our proof of this results helps to understand the most interesting one of Section 5.

Our main contribution is to handle general $\bigwedge_i \gamma_i$ curves and to use the FIFO assumption: the results are presented in Section 5.

4. Sum of shaped leaky buckets flows thought rate-latency node

First of all, to simplify calculations, we need to define a *normal form*, where there is no useless term, and the terms are sorted in a convenient way.

4.1. Normal form of sum of shaped leaky buckets

Definition 1 (Normal form of $\bigwedge_{i=1}^{n} \gamma_{r_i,b_i}$) Let $\gamma_{r_1,b_1}, \ldots, \gamma_{r_n,b_n}$ be a set of γ functions. Let γ_i denotes γ_{r_i,b_i}. The term $\bigwedge_{i=1}^{n} \gamma_i$ is said to be *under normal form of minimum of γ functions*, iff there is no useless constraint (2) and the γ_i are sorted by decreasing rate (3).

$$\forall i, \exists t_i > 0, \forall j \neq i : \gamma_i(t_i) < \gamma_j(t_i) \tag{2}$$
$$i < j \Rightarrow r_i > r_j \tag{3}$$

If $\bigwedge_{i \in [1,n]} \gamma_{r_i,b_i}$ is under normal form, the sequence $x_1, \cdots, x_{n+1}$ of intersection points, and $y_1, \cdots, y_{n+1}$ the

[3] It is closed under addition and deconvolution by a rate-latency server (up to sub-additive closure).

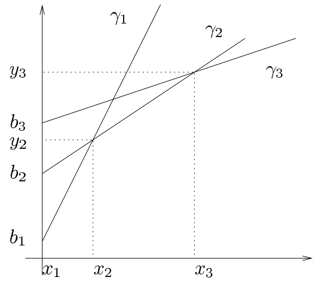

Figure 2. Set of γ functions under normal form

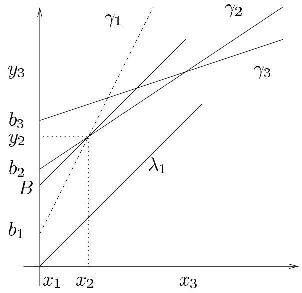

Figure 3. $\bigwedge_{i=1}^{3} \gamma_i \oslash \lambda_1$

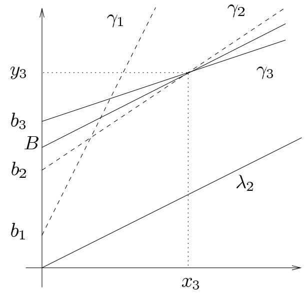

Figure 4. $\bigwedge_{i=1}^{3} \gamma_i \oslash \lambda_{\frac{1}{2}}$

intersection values are formally defined by:

$$\begin{cases} x_1 = 0, y_1 = b_1 \\ \gamma_i(x_i) = \gamma_{i+1}(x_i) = y_k \quad \text{for } 1 \leq i \leq n \\ x_{n+1} = y_{n+1} = \infty \end{cases} \tag{4}$$

The condition (2) is there to avoid to handle useless term. An equivalent definition could have been that, if any i is removed, the function is not the same[4].

The condition (3) is just a useful permutation.

The definition of the x_i is given to express the semantics of theses points. The value can of course be computed: $x_i = \frac{b_{i+1}-b_i}{r_i - r_{i+1}}$.

An example of such a set and related definitions is shown in Figure 2.

Property 1 (Properties of normal form of $\bigwedge_{i=1}^{n} \gamma_{r_i,b_i}$) *If the expression $\bigwedge_{i=1}^{n} \gamma_{r_i,b_i}$ is under normal form, as defined in Definition 1, we have some properties: the sequence x_i is increasing ($x_i < x_{i+1}$), the sequence b_i is increasing ($b_i < b_{i+1}$), the sequence r_i is decreasing ($r_i > r_{i+1}$), and the function is piecewise linear*

$$\forall i \in [1, n] : \bigwedge_{j=1}^{n} \gamma_j =_{[x_i, x_{i+1}]} \gamma_i$$

It is quite obvious to see that this normal form exists and is unique, but it has to be computed. An algorithm in given in [12].

4.2. Deconvolution of normal form $\bigwedge \gamma_i$ by $\beta_{R,T}$

Here comes the first results on handling $\bigwedge \gamma_i$ flows: the deconvolution of such a flow by a $\beta_{R,T}$ function. The proof is done in two steps: firstly (Lemma 1), the deconvolution by λ_R is done, secondly (Lemma 2), the results is extended to $\beta_{R,T}$ by noticing that $\beta_{R,T} = \lambda_R \otimes \delta_T$ and $f \oslash (g \otimes h) = (f \oslash g) \oslash h$. Then, the main result comes: Theorem 1 compute the sub-additive closure of the previous result.

These results were developed independently of the ones of [11]. Here, a direct proof is used, based on the definition of the deconvolution. In [11] a more general algorithm is given, based on the left and right distributivity of the deconvolution. Using their algorithm should produce a smaller proof[5], but our main result (Theorem 2) is not covered by their results, and uses the same kind of proof as this direct one: the exhibition of the maximum $(x_k - t)$. This is why we have included the proof of Lemma 1, as an introduction of the one of Theorem 2.

Lemma 1 (Deconvolution of normal form of $\bigwedge_i \gamma_i$ by λ_R) *Let $\bigwedge_{i=1}^{n} \gamma_i$ be a function under normal form, as defined in Definition 1, and $\beta_{R,T}$ a rate-latency function such that $R \geq r_n$. Then, $(\bigwedge_{i=1}^{n} \gamma_i) \oslash \lambda_R$ can be computed on $\mathbb{R}_{\geq 0}$.*

$$(\textstyle\bigwedge_{i=1}^{n} \gamma_i) \oslash \lambda_R =_{\mathbb{R}_+} \gamma_{R,B} \wedge \textstyle\bigwedge_{i=k}^{n} \gamma_i$$
$$((\textstyle\bigwedge_{i=1}^{n} \gamma_i) \oslash \lambda_R)(0) = B$$

with $k = \min \{ i \,|\, r_i \leq R \}$, and $B = y_k - R x_k$.

The graphical interpretation could be the following: the λ_R can be shifted up, as long as it intersects only with γ_i with higher slope. Figure 2 shows such a $\bigwedge_{i=1}^{n} \gamma_i$ set, and its deconvolution by λ_1 in Figure 3 and by $\lambda_{\frac{1}{2}}$ in Figure 4. Another point should be noted: at 0, the deconvolution value is not null.

PROOF At first step, we can compute the deconvolution in one point $t > 0$ (the case $t = 0$ will be studied after).

$$\left(\left(\bigwedge_{i=1}^{n} \gamma_i \right) \oslash \lambda_R \right)(t) = \sup_{0 \leq s} \left\{ \bigwedge_i \gamma_i(t+s) - Rs \right\}$$

Let be $f_t(s) = \bigwedge_i \gamma_i(t+s) - Rs$. Notice that $\gamma_{r_i,b_i}(t+s) - Rs = r_i(t+s) + b_i - Rs = (r_i - R)s + b_i + r_i t = r_i - R)(t+s) + b_i + Rt$. Then, $f_t(s) = \bigwedge_{i=1}^{n} \gamma_{r_i - R, b_i}(t+s) + Rt$. It should be clear that, for all j, $f_t =_{[x_j - t, x_{j+1} - t]} \gamma_{r_j - R, b_j} - Rt$, and $f'_t = =_{[x_j - t, x_{j+1} - t]} r_j - R$.

s		$x_k - t$	
f'_t	< 0	0	≥ 0
f_t	$\nearrow$		$\searrow$

(5)

The maximum of f_t can easily can be found: it is reached when $t + s = x_k$, and $\sup_{s \geq 0} f_t(s)$ depends of the ordering of x_k and t.

[4] $\forall i \in [1, n] : \bigwedge_{j \in [1,n], i \neq j} \gamma_{r_j,b_j} \neq \bigwedge_{j \in [1,n]} \gamma_{r_j,b_j}$

[5] It produces a term $\max_{i \in [1,n]} f_i$, where f_i is a two segments term [11, Fig. 7], and some segments could be deleted, giving the same result.

- if $t \leq x_k$, $\sup_{s\geq 0} f_t(s)$ is reached for $s = x_k - t$ and $\sup_{s\geq 0} f_t(s) = Rt + \bigwedge_{i=1}^{n} \gamma_{r_i - R, b_i}(x_k) = Rt - Rx_k + \gamma_k(x_k)$
- if $t \geq x_k$, $\sup_{s\geq 0} f_t(s)$ is reached for $s = 0$ and $\sup_{s\geq 0} f_t(s) = \bigwedge_i \gamma_i(t)$.

If $t = 0$, we get the same result but the $\sup_{s\geq 0} f_t(s)$ is the limit to the left at 0, not a maximum. In this special case, we can have $k = 1$ *ie* $x_k = 0$. In this case, the limit to the left is not $Rt - Rx_k + \gamma_k(x_k)$ because γ_1 is not continuous at 0, neither $\bigwedge_i \gamma_i(0)$; but the value is b_1. This is why we have defined $y_1 = b_1$ and not $y_1 = \bigwedge_i \gamma_i(x_1) = 0$...

To sum up

$$\begin{cases} (\bigwedge \gamma_i) \oslash \lambda_R =_{[0,x_k]} Rt + B \\ (\bigwedge \gamma_i) \oslash \lambda_R =_{[x_k,\infty]} \bigwedge_i \gamma_i =_{[x_k,\infty]} \bigwedge_{i=k}^{n} \gamma_i \end{cases}$$
$$\Rightarrow \begin{cases} (\bigwedge \gamma_i) \oslash \lambda_R =_{\mathbb{R}_+} \gamma_{R,B} \wedge \bigwedge_{i=k}^{n} \gamma_i \\ ((\bigwedge \gamma_i) \oslash \lambda_R)(0) = B \end{cases}$$

□

Lemma 2 (Deconvolution of normal form of $\bigwedge_i \gamma_i$ by $\beta_{R,T}$) *Let $\bigwedge_{i=1}^{n} \gamma_i$ be a function under normal form, as defined in Definition 1, and $\beta_{R,T}$ a rate-latency function such that $R \geq r_n$.*

Then, $(\bigwedge_{i=1}^{n} \gamma_i) \oslash \beta_{R,T}$ can be computed on $\mathbb{R}_{\geq 0}$.

$$(\bigwedge_{i=1}^{n} \gamma_i) \oslash \beta_{R,T} =_{\mathbb{R}_+} \gamma_{R,B} \wedge \bigwedge_{i=k}^{n} (\gamma_i \oslash \beta_{R,T})$$

$$((\bigwedge_{i=1}^{n} \gamma_i) \oslash \beta_{R,T})(0) = B \wedge \bigwedge_{i=k}^{n} (b_i + r_i T)$$

with $B = RT + y_k - Rx_k$.

PROOF The proof is quite simple: just note that $\beta_{R,T} = \delta_T \otimes \lambda_R$ and remind that $f \oslash (g \otimes h) = (f \oslash g) \oslash h$. Then, $(\bigwedge_{i=1}^{n} \gamma_i) \oslash \beta_{R,T} = ((\bigwedge_{i=1}^{n} \gamma_i) \oslash \lambda_R) \oslash \delta_T$. And, for continuous function, $(f \oslash \delta_T)(t) = f(t - T)$.

From previous Lemma, $(\bigwedge_{i=1}^{n} \gamma_i) \oslash \beta_{R,T}(t) =_{\mathbb{R}_+} \gamma_{R,B} \wedge \bigwedge_{i=k}^{n} \gamma_i$. And [6, Figure 3.8] shows $(\gamma_i \oslash \beta_{R,T})(t) =_{\mathbb{R}_{\geq 0}} \gamma_i(t - T)$. □

Note that, if all r_i are smaller or equal to R ($k = 1$) then the term $\gamma_{R,B}$ is useless, because $\gamma_{R,B} = \gamma_{R,RT+b1} \geq \gamma_{r_1,b_1}$.

Moreover, if $T \geq x_k$, the term $\gamma_{R,B}$ is also useless: is comes from the fact that $\gamma_{R,B} \geq \gamma_{r_k, b_k + r_k T}$ (see Corollary 1 for details).

Corollary 1 (Properties of the $\gamma_{R,B}$ term) *Here are a few properties of the terms of Lemma 2.*

(i) A big T removes the $\gamma_{R,B}$ term: if $T \geq x_k$, the term $\gamma_{R,B}$ is useless.

$$T \geq x_k \Rightarrow \gamma_{R,B} \wedge \bigwedge_{i=k}^{n} (\gamma_i \oslash \beta_{R,T}) =_{\mathbb{R}_+} \bigwedge_{i=k}^{n} (\gamma_i \oslash \beta_{R,T})$$

(ii) If $T \leq x_k$, the functions $\gamma_{R,B}$ and $\bigwedge_{i=k}^{n} (\gamma_i \oslash \beta_{R,T})$ intersect at point $x_k - T$.

$$\gamma_{R,B}(x_k - T) = y_k = \bigwedge_{i=k}^{n} (\gamma_i \oslash \beta_{R,T})(x_k - T)$$

PROOF (I) To prove that $\gamma_{R,B} \geq \gamma_{r_k, b'_k}$, we have to prove that $R \geq r_k$ (obvious, by definition of k) and $B \geq b'_k = b_k + r_k T$. The definition of B is $B = y_k + R(T - x_k)$. So, under the assumption that $T \geq x_k$, we just have to prove that $y_k \geq b_k + r_k T$. It comes from $y_k = r_k x_k + b_k$. □

PROOF (II) $\gamma_{R,B}(x_k - T) = R(x_k - T) + B = Rx_k - RT + RT + y_k - Rx_k = y_k = \gamma_k(x_k) = \gamma_k(x_k + T - T) = (\gamma_k \oslash \beta_{R,T})(x_k - T)$ □

Theorem 1 (Sub-additive closure of $\bigwedge_i \gamma_i \oslash \beta_{R,T}$) . *Let $\{\gamma_i\}$ be a finite set of γ_{r_i,b_i} functions under the normal form of Definition 1.*

$$\overline{\left((\bigwedge_i \gamma_i) \oslash \beta_{R,T} \right)} = \gamma_{R,B} \wedge \bigwedge_{r_i \leq R} \left(\overline{\gamma_{r_i,b_i} \oslash \beta_{R,T}} \right)$$
$$= \gamma_{R,B} \wedge \bigwedge_{r_i \leq R} \gamma_{r_i, b_i + r_i T}$$

with the same definition of B than in Lemma 2.

PROOF Going from the equality on $\mathbb{R}_+$ to equality of sub-additive closure is done the same way that for Theorem 3 and the fact that the convolution of star-shaped function null at origin is simply the minimum of the functions [6, Theorem 6.3.1].

We have:

$$(\bigwedge_i \gamma_i) \oslash \beta_{R,T} \geq \gamma_{R,B} \wedge \bigwedge_{r_i \leq R} (\gamma_i \oslash \beta_{R,T})$$
$$\Rightarrow \overline{(\bigwedge_i \gamma_i) \oslash \beta_{R,T}} \geq \overline{\gamma_{R,B} \wedge \bigwedge_{r_i \leq R} (\gamma_i \oslash \beta_{R,T})}$$

and the right term can be simplified

$$\overline{\gamma_{R,B} \wedge \bigwedge_{r_i \leq R} (\gamma_i \oslash \beta_{R,T})}$$
$$= \overline{\gamma_{R,B}} \otimes \bigotimes_{r_i \leq R} \overline{\gamma_i \oslash \beta_{R,T}} \quad \text{by [6, Theorem 3.1.11]}$$
$$= \gamma_{R,B} \otimes \bigotimes_{r_i \leq R} \gamma_{r_i, b_i + r_i T} \quad \text{by Theorem 3}$$
$$= \gamma_{R,B} \wedge \bigwedge_{r_i \leq R} \gamma_{r_i, b_i + r_i T} \quad \text{by [6, Theorem 6.3.1]}$$

That is to say: $\overline{(\bigwedge_i \gamma_i) \oslash \beta_{R,T}} \geq \gamma_{R,B} \wedge \bigwedge_{r_i \leq R} \gamma_{r_i, b_i + r_i T}$

Moreover, by definition of sub-additive closure ($\overline{f} = \delta_0 \wedge f \wedge \ldots$), we always have $\overline{f} \leq \delta_0 \wedge f$. And, for $f = (\bigwedge_i \gamma_i) \oslash \beta_{R,T}$ we have $\delta_0 \wedge f = \gamma_{R,B} \wedge \bigwedge_{r_i \leq R} \gamma_{r_i, b_i + r_i T}$.

Both inequalities have been shown, then, equality follows. □

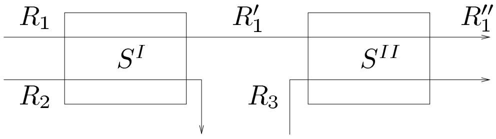

Figure 5. Three flows sharing two servers

Corollary 2 ($\overline{\bigwedge \gamma_i \oslash \delta_T}$) *Let $\{\gamma_i\}$ a finite set of γ_{r_i,b_i} functions under the normal form of Lemma 2. Then*

$$\overline{\left((\bigwedge_i \gamma_i) \oslash \delta_T\right)} = \bigwedge_i \left(\overline{\gamma_{r_i,b_i} \oslash \delta_T}\right)$$

PROOF The proof directory come from the observation that $\delta_T = \beta_{\infty,T}$. □

5. Aggregation of $\bigwedge \gamma_i$ flows sharing a FIFO β server

5.1. What is aggregation in network calculus ?

The basic paradigm of network calculus is one flow R, characterised by an arrival curve α, traversing one server S characterised by one service curve β, which results in an output flows R' with arrival curve $\alpha' = \alpha \oslash \beta$. But it also appends than several flows shares a single server, that we are interested by each flow and not only their sum, (this is called "aggregate scheduling" [6, Chapter 6]), as presented in Figure 5.

In this case, two results are of interest: the first is to know which is the individual service received by each flow, to be able to compute the delay of each flow in the server (for R_2, we would like to be able to know the service offered by the virtual server S_2^I, which can be defined as the "part" of S^I devoted to R_2), and the second is the effect on the flow (the arrival curve of R'_1) to be able to propagate the computation. It should be noted that, in the basic paradigm, the second is a direct application of the first ($\alpha' = \alpha \oslash \beta$), but not for aggregate scheduling.

Moreover, speaking about *the* service receive by each flow is an imprecision. We are in general not able to compute the exact service, but an under approximation. And the sum of the individual arrival curves is less precise than the computation based on the global flow ($\alpha'_1 + \alpha'_2 \geq (\alpha_1 + \alpha_2) \oslash \beta$).

To sum up some results, if two flows R_1 and R_2 with arrival curves α_1 and α_2 share a server S with β service curve, we have some results depending on the server policy.

- **Blind:** the server offers to R_1 the service curve β_1 of (6), [6, Theorem 6.2.1], under the conditions that β is a *strict* service curve for S, and that β_1 is a service curve
$$\beta_1 = [\beta - \alpha_2]^+ \tag{6}$$
- **Non-preemptive priority:** if R_1 has higher priority, each R_i get the service curve β_i of (7) [6].
$$\beta_1 = [\beta - l_M^2]^+ \qquad \beta_2 = [\beta - \alpha_1]^+ \tag{7}$$
- **FIFO:** for each $\theta > 0$, the server offers to R_1 the service curve β_1^θ of (8), [6, Proposition 6.2.1], under the natural condition that β_1^θ is a service curve
$$\beta_1^\theta(t) = [\beta(t) - \alpha_2(t-\theta)]^+ 1_{\{t>\theta\}} \tag{8}$$
However, this does not gives *one* service curve, but a family. And it may not exist a θ better than all others. Nevertheless, we knows that the output flow R'_1 is constrained by α'_1 [6, Proposition 6.2.2].
$$\alpha'_1 = \inf_{\theta \geq 0} \left(\alpha_1 \oslash \beta_1^\theta\right) \tag{9}$$

The aim of this part is to study these results of aggregation of shaped leaky bucket.

5.2. Our contribution: handling shaped leaky buckets

First of all, it must be precised that we do not handle the problem of two flows with arrival curves $\bigwedge_{i=1}^n \gamma_{r_i,b_i}$ and $\bigwedge_{i=1}^m \gamma_{r'_i,b'_i}$ sharing a server, but the sub-case $m = 1$, *i.e.* one $\bigwedge_{i=1}^n \gamma_{r_i,b_i}$ shares a server with a flow of arrival curve $\gamma_{r,b}$. This approximation is made because we are not able, up to now, to compute the general case, and because if a flow has $\bigwedge_{i=1}^m \gamma_{r'_i,b'_i}$ as arrival curve, it also have $\gamma_{r'_m,b'_m}$ as arrival curve. Then, our results can also be applied in the general case, but it gives a pessimistic upper approximation.

That is to says, we only consider $\alpha_1 = \bigwedge_{i=1}^n \gamma_{r_i,b_i}$, $\alpha_2 = \gamma_{r,b}$ and $\beta = \beta_{R,T}$, with $R \geq r + r_n$.

Under the previous simplification, both blind and priority are very simple to compute, because the β_1 functions are of the form $\beta_{R,T}$ (11),(10).

$$[\beta_{R,T} - \gamma_{r,b}]^+ = \beta_{R-r,\frac{RT+b}{R-r}} \tag{10}$$
$$[\beta_{R,T} - l_M]^+ = \beta_{R,T+\frac{l_M}{R}} \tag{11}$$

The FIFO case is more complex.

If $\theta \leq T + \frac{b}{R}$, then, the form of β_1^θ, as defined in (8), is still a $\beta_{R,T}$ function, and the arrival curves of the output flow can be computed with Theorem 1.

$$\beta_1^\theta(t) = \beta_{R-r,T_\theta}$$
$$\overline{\bigwedge \gamma_i \oslash \beta_1^\theta} = \gamma_{R',B'_\theta} \wedge \bigwedge_{r_i \leq R'} \gamma_i \oslash \delta_{T_\theta}$$

with $R' = R - r$, $T_\theta = \frac{RT+b-r\theta}{R-r}$, $B'_\theta = R'(T_\theta - x_k) + y_k$, and always the same definition of $k = \min\{i \,|\, r_i \leq R'\}$. It is clear that, on $[0, T + \frac{b}{R}]$, increasing θ improves the

[6] with l_M^2 the maximal size of a R_2 message and under the natural conditions that the β_i are service curves [6, Corollary 6.2.1]

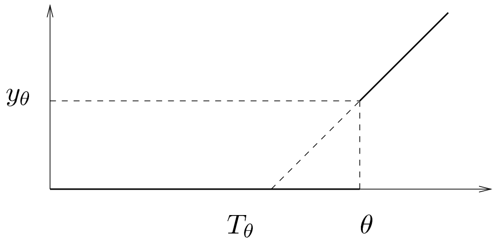

Figure 6. Function β_1^θ for $\theta > T + \frac{b}{R}$

service [7] and decreases the service curve[8]. Then, choosing $\theta = T + \frac{b}{R}$ seems a good strategy to have a tight service. This is (not surprisingly) the same value as when two γ flows share a server (see [6, Corollary 6.4.1]).

If $\theta > T + \frac{b}{R}$, then, the form of β_1^θ, is more complex. In this case, $\beta_1^\theta(t) = \beta_{R-r,T_\theta} 1_{t>\theta}$ ([9]) and looks like the curve on Figure 6. There is no more any relationships between the β_1^θ curves[10] and the output flow are also more complex[11].

$$\overline{\bigwedge \gamma_i \oslash \beta_1^\theta} = \overline{\bigwedge \gamma_i \oslash \delta_\theta} \vee \gamma_{R',B'_\theta} 1_{t \le x_k - \theta}$$

The study of β_1^θ shows that $T + \frac{b}{R}$ seems to be a good choice for θ value. But we also have a second equation to study, (9), to get a tight output flow. This is done in the following theorem.

Theorem 2 (Sum of $\bigwedge \gamma_i$ and $\gamma_{r,b}$ through a FIFO $\beta_{R,T}$) *Let $\bigwedge_i \gamma_i$ be a finite set of γ_{r_i,b_i} functions under the normal form of Lemma 2, let be $\alpha_2 = \gamma_{r,b}$ and $\beta_{R,T}$ such that $R \ge r + r_n$. Then, we have*

$$\overline{\inf_{\theta \ge 0} \left(\bigwedge \gamma_i \oslash \beta_1^\theta \right)} \le \gamma_{R',B'} \wedge \bigwedge_{r_i \le R_i} \gamma_{r_i, b_i + r_i(T + \frac{b}{R})} \quad (12)$$

with $R' = R - r$, $B' = y_k + R'(T + \frac{b}{R} - x_k)$, $k = \min\{i \,|\, r_i \le R'\}$.

PROOF The proof is given in appendix B.

[7] We have $0 \le \theta \le \theta' \le T + \frac{b}{R} \Rightarrow \beta_1^\theta \le \beta_1^{\theta'}$

[8]
$$T_\theta - T + \frac{b}{R} = \frac{r}{R-r}\left(T + \frac{b}{R} - \theta\right)$$
$$B'_\theta - B' = R'\left(T_\theta - T + \frac{b}{R}\right) = r\left(T + \frac{b}{R} - \theta\right)$$

[9] See Section B.1.1 for details.

[10] $0 \le \theta \le T + \frac{b}{R} \le \theta' \Rightarrow \beta_1^\theta(t) \ge \beta_1^{\theta'}(t)$ for $t \in [0, \theta']$ and $\beta_1^\theta(t) \le \beta_1^{\theta'}(t)$ for $t \in [\theta', \infty]$.

[11] This computation is like the one of Lemma 1. We only gives a sketch of proof. $\sup_{0 \le u} \{\bigwedge \gamma_i(t+u) - \beta_1^\theta\} = \sup_{0 \le u \le \theta} \{\bigwedge \gamma_i(t+u)\} \vee \sup_{u \ge \theta} \{\bigwedge \gamma_i(t+u) - R'(u - T_\theta)\}$. The first term can easily be reduced $\sup_{0 \le u \le \theta} \{\bigwedge \gamma_i(t+u)\} = \bigwedge \gamma_i(t + \theta)$. The second term is solved using the same proof than for Lemma 1, a function with sup at $t + u = x_k$. Then $\sup_{u \ge \theta} \{\bigwedge \gamma_i(t+u) - R'(u - T_\theta)\} = \begin{cases} \bigwedge \gamma_i(x_k) - R'(x_t - t - T_\theta) & \text{if } t \le x_k - \theta \\ \bigwedge \gamma_i(t+\theta) - R'(\theta - T_\theta) & \text{otherwise} \end{cases}$.

The result is not very surprising: it is the output computed with $\theta = T + \frac{b}{R}$. It confirms that this is the "best" value choice. All we get is an upper bound, but looking at the proof shows that we are not very far away. There is an under bound where term $\gamma_{R',B'}$ is replaced by a term $\gamma_{R,B''}$, with B'' such that both curves join at point $x_k - T - \frac{b}{R}$.

That is to say, a $\beta_{R,T}$ FIFO server is like a server with a smaller service rate $R' = R - r$, and a delay $T + \frac{b}{R}$, that is to say, the own delay of the server, T, plus eventually the time necessary to handle a burst from the other flow, $\frac{b}{R}$.

6. Example and comparison

To get an idea of the benefit of this method, we are going to study the example of Figure 5 with several methods.

Let assume that each flow R_i (resp. R'_i) has α_i (resp. α'_i) as an arrival curve, with $i \in \{1, 2, 3\}$ and each server S^k offers service curve $\beta^k = \beta_{R^k,T^k}$, with $k \in \{I, II\}$.

In Section 6.1, the α_i are single γ functions; in Section 6.2, the shaping introduced by the link is modelled, and the arrival curves are of the $\bigwedge_j \gamma_j$ family: in 6.2.1, the "distribution of aggregate delay" method is used, and 6.2.2 presents our new method.

6.1. Curves are leaky bucket

In the first case, we are modelling the system with the well known leaky bucket constraint ($\alpha_i = \gamma_{r_i,b_i}$).

In this case, it is well known ([6, Corollary 6.2.3]) that the optimal choice of θ for $\beta_1^\theta(t)$ (cf (8)) is $\theta = T^I + \frac{b_2}{R^I}$, and leads to $\alpha'_1 = \gamma_{r_1,b'_1}$ with $b'_1 = b_1 + r_1(T^I + \frac{b_2}{R^I})$.

The delays d_2^I and d_3^{II} experimented by the flows R_2 and R_3 can be computed.

Lemma 3 (Delay of two γ sharing a FIFO $\beta_{R,T}$ node) *When two flows R_1, R_2 of respective arrival curve $\gamma_{r_1,b_1}, \gamma_{r_2,b_2}$ share a server with FIFO policy and service curve $\beta_{R,T}$ ($R \ge r_1 + r_2$), they both experiment the same maximum delay d defined by:*

$$d = T + \frac{b_1 + b_2}{R}$$

This result is easy to get. It is a bit surprising, because it means that the optimal choice of θ from delay point of view in equation (8) for each flow is $\theta = T + \frac{b_1+b_2}{R}$, as for tightening the output, the best choice is $\theta_1 = T + \frac{b_2}{R}$ for the flow R_1 and $\theta_2 = T + \frac{b_1}{R}$ for the flow R_2. But, it is not surprising from another point of view, since $T + \frac{b_1+b_2}{R}$ is the global delay for the aggregate flow $R_1 + R_2$, and in FIFO mode, the delay experimented by each flow can not be greater than the one of the global flow.

PROOF The proof of Lemma 3 is in [12, Appendix B].

From the previous Lemma 3, we have:

$$d_2^I = T^I + \frac{b_1 + b_2}{R^I}$$
$$d_3^{II} = T^{II} + \frac{b_1 + b_3}{R^{II}} + \frac{r_1}{R^{II}}\left(T^I + \frac{b_2}{R^I}\right)$$

To compute the end to end delay observed by the R_1, with the help of the "pay burst only once" principle, we have to compute the service offered by the concatenation of both servers S^I and S^{II}. Using the FIFO assumption and equation (8), for all θ^I and θ^{II}, $\left(\left[\beta^I(t) - \alpha_2(t-\theta^I)\right]^+ 1_{\{t>\theta^I\}}\right) \otimes \left(\left[\beta^{II}(t) - \alpha_2(t-\theta^{II})\right]^+ 1_{\{t>\theta^{II}\}}\right)$. Using [6, Corollary 6.2.3] once more, the best possible values are

$$\theta^I = T^I + \frac{b_2}{R^I} \qquad \theta^{II} = T^{II} + \frac{b_3}{R^{II}}$$

With these values

$$\beta_1^{I,\theta^I} \otimes \beta^{II,\theta^{II}} = \beta_{R^I - r_2, T^I + \frac{b_2}{R^I}} \otimes \beta_{R^{II} - r_3, T^{II}\frac{b_3}{R^{II}}}$$
$$= \beta_{(R^I - r_2)\wedge(R^{II}-r_3), T^I + T^{II} + \frac{b_1}{R^I} + \frac{b_3}{R^{II}}}$$

and the end-to-end delay d_1 can be computed:

$$d_1 = T^I + T^{II} + \frac{b_2}{R^I} + \frac{b_3}{R^{II}} + \frac{b_1}{(R^I - r_2)\wedge(R^{II} - r_3)}$$

It could be compared with the sum of the local delays d_1^I and d_1^{II} ([12]).

$$d_1^I + d_1^{II} = T^I + T^{II} + \frac{b_2}{R^I} + \frac{b_3}{R^{II}} + \frac{b_1}{R^I} + \frac{b_1}{R^{II}} + \frac{r_1}{R^{II}}\left(T^I + \frac{b_2}{R^I}\right)$$

6.2. Curves are shaped leaky buckets

In this section, the modelling of the arrival curves α_i is different. It is based on the observation that the arrival rate is constrained by the throughput D of the link. Then, the flows can be modelled by arrival curves of the form $\alpha_i = \lambda_D \wedge \gamma_{r_i,b_i}$. In network calculus terminology, the link can be seen as a shaper with curve λ_D. The idea of adding the link constraint to the flow comes from [3, 2].

The Figure 7 illustrates the benefits of the modelling of the shaper in our example: instead of computing the delay d, its gives d^g.

6.2.1 Computation with "distribution of aggregate delay"

But, since they were not able to compute convolution and deconvolution on such arrival curves, a conservative approximation was done, which can be called "distribution of aggregate delay". The idea is the following: even with this kind of curve, it is easy to compute the delay d^g experimented by the aggregate flow (cf. Figure 7). And, in a FIFO server, the delay observed by each individual flow is at most the one of the aggregate flow. Then, the service offered to each individual flow can be approximated by a single delay δ_{d^g}.

[12] In general, computing the individual end-to-end delay gives a better result than the sum of the global local delays. In fact, computing individual service received by one flow is, in general, a pessimistic approximation, and using the "pay burst only once" principle can gain on what was lost, but not always. The end-to-end makes no gain when one term $R^I - r_2$ or $R^{II} - r_3$ is very small, that is to say, when the considered flow has a very small part of the bandwidth.

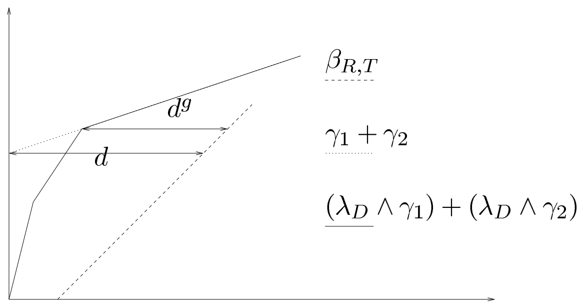

Figure 7. Benefit from the shaper modelling

Then, if d^I denotes the delay observed by the aggregate flow $R_1 + R_2$ in server S^I, this modelling give another expression of α_1'.

$$\alpha_1' = \overline{(\lambda_D \wedge \gamma_{r_1,b_1}) \oslash \delta_{d^I}} = \gamma_{D,Dd^I} \wedge \gamma_{r_1,b_1+Dd^I}$$

We do not give an analytical expression of d^I, since it depends on the relations between r_1,b_1,r_2,b_2,D and R^I, but it can easily be computed by a simple tool (cf Figure 7).

Then, the delay d^{II} observed by the aggregate flow $R_1' + R_3$ can be computed.

In this modelling, there is no way to compute the end-to-end delay, and d_1 is the sum of the two local delays.

$$d_2 = d^I \qquad d_3 = d^{II} \qquad d_1 = d^I + d^{II}$$

6.2.2 Computation with our results

With our new results, we can compute the service S_1^I and S_2^I offered by server S^I to the flow R_1 and R_2, using $T + \frac{b_j}{R}$ as a good choice for θ.

$$\beta_1^I = \beta_{R^I - r_2, T^I + \frac{b_2}{R^I}} \qquad \beta_2^I = \beta_{R^I - r_1, T^I + \frac{b_1}{R^I}}$$

With this curves and the tool presented in previous section, we are able to compute d_2 and d_1^I, the delays respectively observed by R_2 and R_1 in server S^I.

The same way, we can compute β_1^{II} and β_3^{II}. The expression of β_1^{II} is simple, but the one of β_3^{II} depend on the computation of $\alpha_1' = \overline{\inf_{\theta\geq 0}\left((\lambda_D \wedge \gamma_{r_1,b_1}) \oslash \beta_1^\theta\right)}$, and as a complex analytical expression.

$$\beta_1^{II} = \beta_{R^{II} - r_3, T^{II} + \frac{b_3}{R^{II}}}$$

6.3. Numerical examples

To compare the different strategies, we have defined some numerical configuration and computed the delays d_1, d_2 and d_3 of the flows R_1, R_2 and R_3. The results are presented in Table 1. In all the example, we have chosen $R^I = R^{II} = D$, and $T^I = T^{II}$. Notation d_i denotes the delay d of flow i computed with the three methods, "o" is the original method, without shaping presented in Section 6.1, "g" is the method of [3, 2], and "n" is our new method.

In experiments E1 and E2, the network elements are charged at 83%, and the size of the different flow are comparable. In experiments E3, E4 and E5, the network load is smaller (43%) and some big and small flows are mixed. Experiments E6 and E7 are the sames than E1 and E2, except that the network elements are 10 times more powerful, which lead to a small load (8%). At least, E8 and E9 are the sames than E1 and E2, but with smaller delays from the network elements.

The global conclusions of these experiments are: the method g is always better than o for local delays (d_2 and d_3), and in most cases better even for the end-to-end delay d_1 (all except E1 and E9). Our new method is also better than o for local delays (except for E1 and E9), and always better than o for end-to-end delay.

The comparison between g and n is not so clear: g is always better than n for local delays, and n is in general better than g for the end-to-end delay (always except E5 and E7).

The particularity of the E5 and E7 configurations is that the end-to-end flow is very small compared to the other[13].

The fact that g is better than n for local delay is perhaps also related to the choice of θ for β_1^θ. For γ arrival curves, the optimal choice from the delay point of view is $\frac{b_1+b_2}{R}$ (Lemma 3), but from the output curve point of view, the best choice is $\frac{b_2}{R}$. The same effect certainly arises for shaped curves, and after having computed the optimal θ from output flow, we should also try to get the best from local delay point of view. In other words, we have studied the equation (9) and we have to study $\inf_{\theta>0} h(\bigwedge \gamma_i, \beta_1^\theta)$.

7. Conclusion

It had been shown in [3] that taking into account the shaping of aggregate leaky bucket really decreases the bounds computed in network calculus, and it is known than the "pay burst only once" principle also decreases the end-to-end delays. Nevertheless, up to now, it was impossible to uses both techniques together, since aggregate shaped flows have arrival curves of the form $\bigwedge \gamma_i$. So, we have calculated how to handle this kind of curve with $\beta_{R,T}$ service curves, globally and individually in case of FIFO policy.

[13] In this cases, the "pay burst only once" principle, that avoid to pay its own burst in each network element, gives a little gain compared to the cost of multiplexing with the other flows.

Our results have been compared on some configurations with the previous works. They are always better than without shaping for the end to end delay, and often on local delays. Compared with the global shaping, our results are in general worst for local delays and better for the end to end delay.

To sum up, there are three phenomena: modelling shaping mechanisms improves the bounds computation, applying the "pay burst only once" principle also does, but, to apply it in case of aggregation, the service offered to each individual flow must be approximated, and this approximation is pessimistic. Our contribution allows to handle the three, but, depending on the configuration, the gain obtained by modelling one phenomenon could be spoilt by the modelling of another: in this cases, looking for the service offered to an individual flow gives very pessimistic approximation when this flow has a "very small" part of the bandwidth compared to the other ones.

There are still several problems to solve in order to improve the calculated bounds. First is that we do not solve equation (9), we only have an upper bound. Perhaps could we find an analytical solution, or an algorithm to compute it. Second, equation (9) is about tightening the output flows in FIFO mode, but as presented at the end of Section 6.3, having a better local delay could also be possible. Third, when two $\bigwedge \gamma_i$ flows share a FIFO server, our modelling make the hypothesis to consider one only as a γ flow. Removing this simplification could improve the result.

In parallel with these theoretical study, further works will evaluate the gains on a real industrial network, like the embedded AFDX.

References

[1] F. Frances, C. Fraboul, and J. Grieu, "Using Network Calculus to optimize AFDX network", in *Proceeding of the 3thd European congress on Embedded Real Time Software (ERTS06)*, January 2006, Toulouse.

[2] J. Grieu, F. Frances, and C. Fraboul, "Preuve de déterminisme d'un réseau embarqué avionique", in *Actes du Colloque Francophone sur l'Ingenierie des Protocoles (CFIP 2003)*, Octobre 2003, Paris.

[3] J. Grieu, *Analyse et évaluation de techniques de commutation Ethernet pour l'interconnexion des systèmes avioniques*, PhD thesis, Institut National Polytechnique de Toulouse (INPT), Toulouse, Juin 2004.

[4] R. L. Cruz, "A calculus for network delay, Part I: Network elements in isolation", *IEEE Transactions on information theory*, vol. 37, no. 1, pp. 114–131, January 1991.

[5] R. L. Cruz, "A calculus for network delay, Part II: Network Analysis", *IEEE Transactions on information theory*, vol. 37, no. 1, pp. 132–141, January 1991.

[6] J.-Y. Le Boudec and P. Thiran, *Network Calculus*, volume 2050 of *LNCS*, Springer Verlag, 2001, http://lrcwww.epfl.ch/PS_files/NetCal.htm.

[7] C.-S. Chang, *Perfomance Guarantees in communication networks*, Telecommunication Networks and Computer Systems. Springer, 2000.

									d_1			d_2			d_3		
Exp	R	T	r_1	b_1	r_2	b_2	r_3	b_3	o	g	n	o	g	n	o	g	n
E1	1	1	$\frac{1}{3}$	4	$\frac{1}{2}$	2	$\frac{1}{2}$	2	14	13.5	12	7	6	7	8	7.5	8
E2	1	1	$\frac{1}{2}$	2	$\frac{1}{3}$	4	$\frac{1}{3}$	4	13	14.33	12	7	6	9	9.5	8.33	11.5
E3	1	1	$\frac{1}{3}$	4	$\frac{1}{10}$	$\frac{1}{2}$	$\frac{1}{10}$	$\frac{1}{2}$	7.44	4.30	3.66	5.5	2.10	5.27	6.0	2.20	5.77
E4	1	1	$\frac{1}{3}$	4	$\frac{1}{10}$	$\frac{1}{2}$	$\frac{1}{10}$	$\frac{1}{2}$	7.44	4.30	3.66	5.5	2.10	5.27	6.0	2.20	5.77
E5	1	1	$\frac{1}{10}$	$\frac{1}{2}$	$\frac{1}{3}$	4	$\frac{1}{3}$	4	10.75	4.41	10.27	5.5	2.1	2.16	6.0	2.31	2.16
E6	10	1	$\frac{1}{3}$	4	$\frac{1}{2}$	2	$\frac{1}{2}$	2	2.82	2.44	2.42	1.60	1.22	1.41	1.64	1.22	1.45
E7	10	1	$\frac{1}{2}$	2	$\frac{1}{3}$	4	$\frac{1}{3}$	4	3.01	2.50	2.81	1.60	1.22	1.22	1.67	1.28	1.29
E8	1	$\frac{1}{4}$	$\frac{1}{3}$	4	$\frac{1}{2}$	2	$\frac{1}{2}$	2	12.5	11.81	10.5	6.25	5.25	6.25	7	6.56	7
E9	1	$\frac{1}{4}$	$\frac{1}{2}$	2	$\frac{1}{3}$	4	$\frac{1}{3}$	4	11.5	12.58	10.5	6.25	5.25	8.25	8.35	7.33	10.37

Table 1. Numerical comparison of the different strategies for the net of Figure 5

[8] D. E. Wrege, E. W. Knightly, H. Zhang, and J. Liebeherr, "Deterministic delay bounds for VBR video in packet-switching networks: fundamental limits and practical tradeoffs", *IEEE/ACM Transactions on Networking*, vol. 4, no. 3, pp. 352–362, June 1996.

[9] H. Sariowan, R. L. Cruz, and G. C. Polyzos, "SCED: A generalized scheduling policy for guaranteeing Quality-of-Service", *IEEE/ACM transactions on networking*, vol. 7, no. 5, pp. 669–684, October 1999.

[10] M. Fidler, "Extending the Network Calculus Pay Bursts Only Once Principle to Aggregate Scheduling.", in *Proc. of the Second International Workshop on Quality of Service in Multiservice IP Network (QoS-IP 2003)*, volume 2601 of *LNCS*, February 2003, pp. 19–34, Milano, Italy.

[11] A. Bouillard and E. Thierry, "An Algorithmic Toolbox for Network Calculus", *Discrete Event Dynamic Systems*, vol. 17, no. 4, october 2007, http://www.springerlink.com/content/876x51r6647r8g68/.

[12] M. Boyer and A. Chagou, "Managing aggregation of shaped leaky buckets flows through GPS node in network calculus", Technical report, 2007, http://irt.enseeiht.fr/boyer/Publications.html#IRIT-2007-21.

A. Sub-additive closure $\gamma_{r,b} \oslash \beta_{R,T}$

Theorem 3 (Sub-additive closure of $\gamma_{r,b} \oslash \beta_{R,T}$) *Assuming $R \geq r$:*

$$\overline{\gamma_{r,b} \oslash \beta_{R,T}} = \gamma_{r,b+rT}$$

PROOF From [6, §3.1.9], we have $\gamma_{r,b} \oslash \beta_{R,T} =_{\mathbb{R}_+} \gamma_{r,b+rT}$ and $\gamma_{r,b} \oslash \beta_{R,T} \geq \gamma_{r,b+rT}$.

First step: $\overline{\gamma_{r,b} \oslash \beta_{R,T}} \geq \gamma_{r,b+rT}$

By isotonicity of sub-additive closure [6, Theorem 3.1.11], $\overline{\gamma_{r,b} \oslash \beta_{R,T}} \geq \overline{\gamma_{r,b+rT}}$ and $\overline{\gamma_{r,b+rT}} = \gamma_{r,b+rT}$ because γ functions are sub-additive.

Second step: $\overline{\gamma_{r,b} \oslash \beta_{R,T}} \leq \gamma_{r,b+rT}$

By definition $\overline{\gamma_{r,b} \oslash \beta_{R,T}} = \delta_0 \wedge (\gamma_{r,b} \oslash \beta_{R,T}) \wedge ((\gamma_{r,b} \oslash \beta_{R,T}) \otimes (\gamma_{r,b} \oslash \beta_{R,T})) \wedge \ldots$. It obviously follows, $\overline{\gamma_{r,b} \oslash \beta_{R,T}} \leq \delta_0 \wedge (\gamma_{r,b} \oslash \beta_{R,T})$.

Because $\forall t \leq 0 : \delta_0(t) = 0$ we have $\forall t \leq 0 : \overline{\gamma_{r,b} \oslash \beta_{R,T}} = 0$. Combining this with equality of $\gamma_{r,b} \oslash \beta_{R,T}$ and $\gamma_{r,b+rT}$ on $\mathbb{R}_+$, we have $\delta_0 \wedge (\gamma_{r,b} \oslash \beta_{R,T}) = \gamma_{r,b+rT}$. □

B. Sketch of proof of Theorem 2

By lack of space, only a sketch of proof is given. A full version can be found on-line [12].

B.1. Solving $S_{\theta,t} = (\bigwedge \gamma_i \oslash \beta^1_\theta)(t)$

The definition of the deconvolution is:

$$\left(\bigwedge \gamma_i \oslash \beta^1_\theta \right)(t) = \sup_{u \geq 0} \left\{ \bigwedge \gamma_i(t+u) - \beta^1_\theta(u) \right\}$$

The first step consists in having an explicit expression of $\beta^1_\theta(u) = [\beta_{R,T}(u) - \gamma_{r,b}(u-\theta)]^+ 1_{u>\theta}$.

B.1.1 Simplification of the term $\beta^1_\theta(u)$

Let us define $R' = R - r$ and $T_\theta = \frac{RT+b-r\theta}{R-r}$.

$$\beta^1_\theta(u) = \begin{cases} 0 & \text{if } u \leq (\theta \vee T_\theta) \\ R'(u - T_\theta) & \text{if } u > (\theta \vee T_\theta) \end{cases} \tag{13}$$

Note that $\theta \geq T_\theta \iff \theta \geq T + \frac{b}{R}$. Then, we get the same results as in Section 5.2.

B.1.2 Inserting the expression of $\beta^\theta_1(u)$ into $S_{\theta,t}$

Because $\beta^\theta_1(u)$ expression changes depending on $\theta \vee T_\theta$, the inf research can be decomposed into two sub-parts: on $[0, \theta \vee T_\theta]$ and on $]\theta \vee T_\theta, \infty[$.

$$S_{\theta,t} = S^1_{\theta,t} \vee S^2_{\theta,t} \tag{14}$$

$$S^1_{\theta,t} = \sup_{0 \leq u \leq \theta \vee T_\theta} \left(\bigwedge \gamma_i(t+u) \right) = \bigwedge \gamma_i(t + \theta \vee T_\theta)$$

$$S^2_{\theta,t} = \sup_{u > \theta \vee T_\theta} \left(\bigwedge \gamma_i(t+u) - R'(u - T_\theta) \right)$$

B.1.3 Study of the term $S^2_{\theta,t}$

In order to solve the term $S^2_{\theta,t}$, we first study the function $f_{t,\theta}$ from $\mathbb{R}_+$ to $\mathbb{R}$ defined by: $f(u) = \bigwedge \gamma_i(t+u) - R'(u - T_\theta)$ Of course, we have $S^2_{\theta,t} = \sup_{u > \theta \vee T_\theta} \{f_{t,\theta}(u)\}$

Variations of the function $f_{t,\theta}$ The study of the variations of the functions $f_{t,\theta}$ is a simple job: it has already been done in proof of Lemma 1. $f_{t,\theta}$ reaches its upper bound at $x_k - t$.

Expression of $S^2_{t,\theta}$

With the variations of $f_{t,\theta}$ in mind, the value of $S^2_{t,\theta}$ is easy to get: either $\theta \vee T_\theta$ is less than $x_k - t$, and the upper bound is reached, otherwise, $f_{t,\theta}$ is (weakly) decreasing and its upper bound is the limit to the right at $\theta \vee T_\theta$.

$$S^2_{t,\theta} = \begin{cases} f_{t,\theta}(x_k - t) & \text{if } x_k - t \geq \theta \vee T_\theta \\ f_{t,\theta}(\theta \vee T_\theta) & \text{if } x_k - t \leq \theta \vee T_\theta \end{cases} \tag{15}$$

B.1.4 Expression of $S_{t,\theta}$

The insertion of (15) into (14) leads to 3 expressions.

$$S^a_{t,\theta} \overset{\text{def}}{=} \left(\bigwedge \gamma_i(t+\theta)\right) \vee \left(\bigwedge \gamma_i(x_k) + R'(t + T_\theta - x_k)\right)$$

$$S^b_{t,\theta} \overset{\text{def}}{=} \left(\bigwedge \gamma_i(t+T_\theta)\right) \vee \left(\bigwedge \gamma_i(x_k) + R'(t + T_\theta - x_k)\right)$$

$$S^c_{t,\theta} \overset{\text{def}}{=} \left(\bigwedge \gamma_i(t+\theta \vee T_\theta)\right)$$

$$S_{\theta,t} = \begin{cases} S^a_{t,\theta} & \text{if } x_k - t \geq \theta \geq T_\theta \\ S^b_{t,\theta} & \text{if } x_k - t \geq T_\theta \geq \theta \\ S^c_{t,\theta} & \text{if } x_k - t \leq \theta \vee T_\theta \end{cases} \tag{16}$$

B.2. Computation of $\inf_{\theta \geq 0} S_{t,\theta}$

One part of the complexity of this part is that the expression of $S_{t,\theta}$ changes with the relationships between $x_k - t$ (constant in the $\inf_{\theta \geq 0}$), θ and T_θ, which is function of θ. So, we are going to find some intervals I where the expression does not change[14], make a partition of $[0, \infty[$, compute the inf on each interval, and get the minimum of all theses inf, which is the inf on $[0, \infty[$.

To find these intervals, let us show how the relationships between $x_k - t$, θ and T_θ can be deducted from relations between $x_k - t$ and two others terms, independent from θ (Section B.2.1).

B.2.1 Relations between $x_k - t$, θ and T_θ

The relationships between $x_k - t$, θ and T_θ can be derived from the relationships between θ and two other expressions, without θ, denoted x' and x''. Moreover, as we are going to make an intensive use of $x_k - t$, let us define x^t as a shorthand.

$$x^t \overset{\text{def}}{=} x_k - t \tag{17}$$

$$\theta \geq T_\theta \iff \theta \geq x' \qquad x' \overset{\text{def}}{=} T + \frac{b}{R} \tag{18}$$

$$x^t \geq T_\theta \iff \theta \geq x'' \qquad x'' \overset{\text{def}}{=} \frac{RT + b + R'(t - x_k)}{r} \tag{19}$$

Building a partition of $[0, \infty[$ where the expression of $S_{t,\theta}$ is stable depends of the ordering between x^t, x' and x''. Did the 6 interleaving exists? It does not, indeed, because $x'' \leq x' \iff x' \leq x^t$ (cf (20) and (21)).

$$x'' \leq x' \iff t \leq x_k - T - \frac{b}{R} \tag{20}$$

$$x' \leq x^t \iff t \leq x_k - T - \frac{b}{R} \tag{21}$$

The reader should also keep in mind that T_θ is a decreasing function of θ.

[14]It is either $S^a_{t,\theta}$, $S^b_{t,\theta}$ or $S^c_{t,\theta}$.

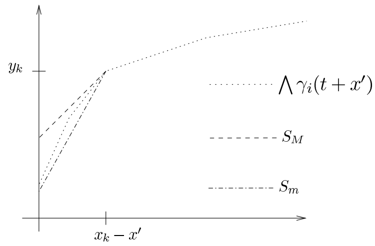

Figure 8. S_m, S_M **and** $\bigwedge \gamma_i(t+x')$

B.2.2 Computing $\inf_{\theta \leq 0} S_{t,\theta}$

The expression of $S_{t,\theta}$ depends on the relations between θ, T_θ and x^t, and the relations between these three expressions can be deduced from the relation between x^t, x' and x'' (see (16), (17), (18), (19)). Moreover, there are only two possible ordering of theses variables, depending on the relation between t and $x_k - T - \frac{b}{R}$ (see (20) and (21)). This section is about computing $\inf_{\theta \leq 0} S_{t,\theta}$ is these two cases.

By lack of space, only the results are presented here.

If $x'' \leq x' \leq x^t$ (ie $t \leq x_k - T - \frac{b}{R}$ then the best approximation we are able to get is:

$$S_M \geq \inf_{x' \leq \theta \leq x_k} S^a_{t,\theta} \geq S_m$$

with $S_M = \bigwedge \gamma_i(x_k) + R'(t + x' - x_k)$ and $S_m = (\bigwedge \gamma_i(t + x')) \vee (\bigwedge \gamma_i(x_k) + R(t + x' - x_k))$.

If $x^t \leq x' \leq x''$ (ie $t \geq x_k - T - \frac{b}{R}$) then

$$\inf_{\theta \geq 0} S_{t,\theta} = \bigwedge \gamma_i(t + x')$$

B.3. From conditional expressions to a piecewise linear function

The previous section has computed the expression $\inf_{\theta \geq 0} S_{t,\theta}$ when $t \geq x_k - T - \frac{b}{R}$ ($\bigwedge \gamma_i(t + x')$) and two lower and upper bounds when $t \leq x_k - T - \frac{b}{R}$ (S_m and S_M).

At point $t = x_k - T - \frac{b}{R}$ the three curves intersect. Then, the situation is the one presented in the Figure 8. Because S_M is a linear function with slope $R' > r_k$ (by definition), we can have a simple expression of the global upper bound:

$$\left(R'(t + x' - x_k) + y_k\right) \wedge \bigwedge_{r_i \leq R'} \gamma_i(t + x') \tag{22}$$

Some work can also be done on the lower bound S_m, but it is useless, since only the upper bound is used for arrival curve.

Last step of the proof is to get the sub-additive closure. This is done like in the proof of Theorem 3. Then, we get the result.

$$\overline{\inf_{\theta \geq 0} \left(\bigwedge \gamma_i \oslash \beta^\theta_1\right)} \leq \gamma_{R',B'} \wedge \bigwedge_{r_i \leq R_i} \gamma_{r_i, b_i + r_i(T + \frac{b}{R})} \tag{23}$$

with $R' = R - r$, $B' = y_k + R'(T + \frac{b}{R} - x_k)$, $k = \min\{i \,|\, r_i \leq R'\}$ and $\beta^\theta_1(t) = [\beta(t) - \alpha_2(t - \theta)]^+ 1_{\{t > \theta\}}$.