Improved Service Curve for Element With Known Transmission Rate

Marc Boyer, Hugo Daigmorte

▶ **To cite this version:**

Marc Boyer, Hugo Daigmorte. Improved Service Curve for Element With Known Transmission Rate. IEEE Networking Letters, 2023, 5 (1), pp.46-49. 10.1109/LNET.2022.3150649 . hal-04151223

HAL Id: hal-04151223
https://hal.science/hal-04151223

Submitted on 4 Jul 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Improved service curve for element with known transmission rate

Marc Boyer – ONERA / DTIS, Université de Toulouse – F-31055 Toulouse – France
Hugo Daigmorte – RealTime-at-Work – F-54600 Villers-lès-Nancy – France

***Abstract*—Network calculus is a well established method for computing guaranteed delay bounds in networks. When a network element is shared between several flows, its aggregate capacity is shared with regards to a scheduling policy. In this case, network calculus can compute a residual service curve for each flow, based on scheduling information. In most systems, when a frame is selected for transmission, it is transmitted at full (constant) line rate up to completion or preemption. In this letter, we propose a generic approach that can enhance any residual service curve by taking this effect into account.**

I. Introduction

Networks are often a critical element of computing systems, and the latency introduced by the network can be a critical part of the response time of a system. Depending on the context, one may be interested in the average delay or by the worst case. Network calculus is a well established method for computing guaranteed bounds on such delay. In network calculus, the workload generated by the data flows is modelled using arrival curves, and the capacity of the network elements is modelled using service curves. When this capacity is shared by different flows, one can derive for each flow a *residual* service curve, using either a result matching the scheduling policy or a generic one. Nevertheless, in most systems, when a frame is selected for transmission, it is transmitted at full (constant) line rate up to completion or preemption, and this effect is rarely taken into account. In this letter, we propose a generic result that can enhance any residual service curve by taking this effect into account.

After a short recall on network calculus (Section II) and the presentation of a motivating example (Section III), the main theorems will be presented in Section IV. They will be compared qualitatively with state of the art results in Section V, and Section VI will evaluate the gain on some examples.

II. Network calculus

Here is a short introduction to network calculus, a more detailed presentation can be found in [1], [2]. Let $\mathbb{N}$ the set of integers, $\mathbb{R}^+$ the set of non-negative reals, $\lceil\cdot\rceil : \mathbb{R}^+ \to \mathbb{N}$ the ceiling function ($\lceil 1\rceil = 1, \lceil 1.9\rceil = 2$), $\mathcal{F}_\uparrow$ the set of non-decreasing functions from $\mathbb{R}^+$ to $\mathbb{R}^+$. The *convolution* is defined by $(f * g)(t) = \inf_{0\le s\le t} f(t-s) + g(s)$. It is associative, commutative and isotone ($\forall f, g, h \in \mathcal{F}_\uparrow : f \ge g \implies f*h \ge g*h$). For any $R, L, C \ge 0$, $\lambda_C, \beta_{R,L} \in \mathcal{F}_\uparrow$ are defined by $\lambda_C(t) = Ct$, $\beta_{R,L}(t) = R\max(0, t-L)$.

In network calculus, the workload of a flow at an observation point is represented by a function $A : \mathbb{R}^+ \to \mathbb{R}^+$, non-decreasing, left-continuous, and $A(t)$ represents the cumulative amount of data (in some data unit, like bit or words) observed up to time t. This exact behavior can be abstracted by an *arrival curve* α satisfying $\forall t, d \in \mathbb{R}^+ : A(t+d) - A(t) \le \alpha(d)$. This is a contract on the amount of data received on any interval of duration d. A server associates to a list of input flows $(A_1, \ldots, A_n)$ a list of departure flows $(D_1, \ldots, D_n)$, with $A_i \ge D_i$ (because the arrival happens before the departure). A *backlogged period* is an interval $(s, t]$ where $\forall u \in (s, t] : D(u) < A(u)$. For any instant t, let $s = Start(t) = \sup\{u \le t \mid D(u) = A(u)\}$, then $(s, t]$ is a backlogged period, and $A(s) = D(s)$. A minimal capacity of a server is represented either by a *simple minimal service* curve β when $D \ge A * \beta$ holds for any pair of arrival and departure, with $A = \sum_{i=1}^n A_i$ and $D = \sum_{i=1}^n D_i$, or by a *strict minimal service* curve β when $D(t+d) - D(t) \ge \beta(d)$ holds for any backlogged period $(t, t+d]$. When a flow A with arrival curve α crosses a server with simple or strict service curve β, its delay is bounded by the horizontal deviation $hdev(\alpha, \beta) = \sup_{t\ge 0} \inf\{u \mid \alpha(t) \le \beta(t+u)\}$.

III. Motivating example

To illustrate the different approaches, let us consider as a motivating example the famous CAN configuration from [3] with 3 flows, A, B, C, listed by order of priority, sending packets of 125 bits, A, B, C having respective periods of 2.5 ms, 3.5 ms and 3.5 ms. sharing a bus with 125Kb/s speed.

At top of Figure 1 are represented the arrivals of the frames, and the resulting schedule, resulting in a delay of 3ms for frame C_1 and 3.5ms for frame C_2. It has been shown in [3] that this is the worst case for the flow C. Just below are plotted α_C (the best arrival curve of C), α_C^{lin} (the best linear arrival curve of C), $\alpha_A + \alpha_B$ (the sum of the best arrival curves for A and B) and $\alpha_A^{\text{lin}} + \alpha_B^{\text{lin}}$ (the sum of their best linear arrival curves).

The existing result on static priority scheduling states that, if a server offers a strict service curve β, then C, the lowest priority flow receives a residual service $\beta_C = [\beta - (\alpha_A + \alpha_B)]^+_\uparrow$, plotted at bottom of Figure 1. From this service, network calculus ensures that the delay experienced by C is less than $hdev(\alpha_C, \beta_C) = 5$. This bound is correct, but quite large with regard to the real worst case. The reason comes from what happens at time 2.5ms. Whereas C_1 begins its transmission at 2ms, the high priority frame A_2 is released at 2.5ms. In a

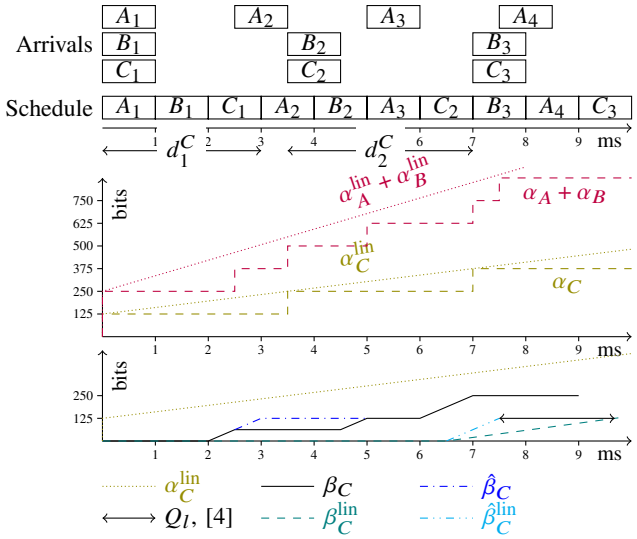

Fig. 1. A CAN bus shared by three flows: worst schedule and related arrival and service curves

preemptive scheduling, it would stop the transmission of C_1, leading to a delay of 5ms, but on a non-preemptive bus, C_1 is transmitted up to completion. From residual service point of view, it means that $\hat{\beta}_C$, drawn at bottom of Figure 1, is also a service curve for C.

IV. Enhancing service curves

Here are presented the theorems enhancing any service curve in the case of a server transmitting packets at a minimal constant rate up to completion[1].

A. Strict minimal service curve

Theorem 1 (Strict minimal service curve)**.** *Let S be a server and $(A, D) \in S$ with A made of packets of length in $[l_{\min}, l_{\max}]$, such as when a packet starts transmission, it is transmitted up to completion with at least the constant speed C. If S offers to A a strict minimal service curve β, then it also offers to A the strict minimal service curve*

$$\hat{\beta} = l_{min}\left\lceil \frac{\beta}{l_{max}} \right\rceil * \lambda_C. \tag{1}$$

Proof. Let $(t, t+d]$ a backlogged period and consider the amount of data $D(t+d) - D(t)$.

Define x as: If a packet is transmitting a t then x is defined as the end of transmission of this packet, (we know that each packet is transmitted until the end at constant speed C so this moment exists); otherwise, $x = t$.

Similarly define y as: If a packet is transmitting a $t+d$ then y is the start of transmission of this packet, otherwise $y = t+d$.

By definition, we know that $t \le x$ and $y \le t + d$.

- First consider the special case $y < x$:

This means that the same packet is transmitted at t and at $t + d$. This packet is transmitted at minimal constant speed C so:

$$D(t+d) - D(t) \ge \lambda_C(d) \tag{2}$$

Since $\beta(0) = \lambda_C(0) = 0$, $\lambda_C \ge l_{\min}\left\lceil \frac{\beta}{l_{\max}} \right\rceil * \lambda_C$ [1, Thm. 3.1.6], $D(t+d) - D(t) \ge \hat{\beta}(d)$.

- Now consider the case $x \le y$:
During $[t, x]$ and $[y, t+d]$, part of a packet is transmitted, at least at constant speed C, so:

$$\begin{aligned} & D(t+d) - D(t) && (3)\\ =&(D(t+d) - D(y)) + (D(y) - D(x)) + (D(x) - D(t))\\ \ge&(t + d - y)C + (D(y) - D(x)) + (x - t)C\\ \ge&C(x - y + d) + (D(y) - D(x)) && (4)\end{aligned}$$

But by construction, there are $n \in \mathbb{N}$ packets transmitted during $[x, y]$. For the proof, let us introduce $l_{mean}^{x,y} = \frac{D(y)-D(x)}{n}$, the average size of these n packets and notice that $l_{\min} \le l_{mean}^{x,y} \le l_{\max}$. By definition of strict service, $D(y) - D(x) \ge \beta(y - x)$, so

$$nl_{mean}^{x,y} \ge \beta(y - x) \tag{5}$$

$$\Rightarrow \lceil n \rceil = n \ge \left\lceil \frac{\beta(y-x)}{l_{mean}^{x,y}} \right\rceil \ge \left\lceil \frac{\beta(y-x)}{l_{max}} \right\rceil. \tag{6}$$

It also holds

$$D(y) - D(x) \ge nl_{\min}. \tag{7}$$

Then, by combining eq (4), (6) and (7), it comes

$$D(t+d) - D(t) \ge l_{min}\left\lceil \frac{\beta(y-x)}{l_{max}} \right\rceil + C(x - y + d). \tag{8}$$

Let $s = d + x - y$

$$D(t+d) - D(t) \ge l_{min}\left\lceil \frac{\beta(d-s)}{l_{max}} \right\rceil + C(s) \tag{9}$$

$$\ge \inf_{0 \le s \le d}\left(l_{min}\left\lceil \frac{\beta(d-s)}{l_{max}} \right\rceil + \lambda_C(s) \right) \tag{10}$$

$$\ge \left(l_{min}\left\lceil \frac{\beta}{l_{max}} \right\rceil * \lambda_C \right)(d) = \hat{\beta}(d) \tag{11}$$

□

Corollary 1. *With the same hypotheses and notations than in Theorem 1, S offers to A the strict minimal service curve*

$$\max(\beta, \hat{\beta}). \tag{12}$$

Proof. The maximum of two strict minimal service curves is a strict minimal service curve [2, Prop. 5.6], leading to the result. □

B. Simple service curve

Theorem 2 (Simple service curve)**.** *Let S be a server and $(A, D) \in S$ with A left-continuous where data is grouped by packets of fixed length l, such as when a packet starts transmitting, it is transmitted up to completion with at least the constant speed C. If S offers to A a simple minimal service curve β with β a left-continuous function and if A is packetized*

[1]The real transmission speed may be larger that C (e.g. due to clock drift)

i.e. $A = l\left\lceil \frac{A}{l} \right\rceil$*, then S also offers to A a simple minimal service curve*

$$\beta' = l\left\lceil \frac{\beta}{l} \right\rceil * \lambda_C. \tag{13}$$

Proof. Let $t \in \mathbb{R}^+$, and consider the amount of data $D(t)$.

Define s as: if a packet is transmitting a t then s is defined as the start of transmission of this packet; otherwise $s = t$.

During $[s, t]$, part of the packet is transmitted, at constant speed C, so:

$$D(t) \geq D(s) + C(t - s) \tag{14}$$

Since S offers to A a simple minimal service curve β, $D(s) \geq (A * \beta)(s)$. But by construction, there are $n \in \mathbb{N}$ packets of length l transmitted during $[0, s]$ *i.e.* $D(s) = nl$, so

$$nl \geq (A * \beta)(s) \Rightarrow \lceil n \rceil = n \geq \left\lceil \frac{(A * \beta)(s)}{l} \right\rceil \tag{15}$$

and

$$D(t) \geq l\left\lceil \frac{(A * \beta)(s)}{l} \right\rceil + C(t - s) \tag{16}$$

Moreover since A and β are left-continuous

$$\exists u \leq s \text{ such that } (A * \beta)(s) = A(u) + \beta(s - u) \tag{17}$$

Then it can be deduced

$$D(t) \geq l\left\lceil \frac{A(u) + \beta(s - u)}{l} \right\rceil + C(t - s) \tag{18}$$

$$\geq l\left\lceil \frac{A(u)}{l} + \frac{\beta(s - u)}{l} \right\rceil + C(t - s) \tag{19}$$

Since A is packetized $A = l\left\lceil \frac{A}{l} \right\rceil$, so:

$$D(t) \geq l\left\lceil \left\lceil \frac{A(u)}{l} \right\rceil + \frac{\beta(s - u)}{l} \right\rceil + C(t - s) \tag{20}$$

$$\geq l\left\lceil \frac{A(u)}{l} \right\rceil + l\left\lceil \frac{\beta(s - u)}{l} \right\rceil + C(t - s) \tag{21}$$

$$= A(u) + l\left\lceil \frac{\beta(s - u)}{l} \right\rceil + C(t - s) \tag{22}$$

$$\geq \left(A * l\left\lceil \frac{\beta}{l} \right\rceil\right)(s) + \lambda_C(t - s) \tag{23}$$

$$\geq \inf_{0 \leq s \leq t} \left(\left(A * l\left\lceil \frac{\beta}{l} \right\rceil\right)(s) + \lambda_C(t - s) \right) \tag{24}$$

$$= \left(A * \left(l\left\lceil \frac{\beta}{l} \right\rceil * \lambda_C\right)\right)(t) = (A * \hat{\beta})(t) \tag{25}$$

□

V. State of the art

It is well known that taking into account the packets sizes in network calculus gives significantly better bounds than a linear modeling (gain of 18% in [5], and between 20% and 25% in [6]), either in the arrival curves or the services curves.

On arrival curves, a generic result is presented in [6, Thm. 6] with hypotheses very close to the ones of this paper: if all packets in a flow have the same size l and if each packet is emitted up to completion at constant rate C, then any arrival curve α can be enhanced into $l\left\lfloor \frac{\alpha}{l} \right\rfloor \oslash \lambda_C$ where $\oslash$ denotes the deconvolution operator.

On service curves, one may divide the contributions into two kinds: the ones offering generic results, not related to a specific policy, and the ones devoted to one specific scheduling policy.

The oldest specific work starts with the same motivating example as in Section III. It addresses non-preemptive static priority scheduling, but lead to a very complex expression and seems not having being reused [7].

Another piece of work addresses two approximations of the GPS policy: Weighted Round Robin and Deficit Round Robin. There were linear residual curves [8], [9] based on the amount of bits transmitted. By counting the number of packets and using pseudo-inverses, one may generate a service curve as an alternation of plateaus and full speed service, for Weighted Round Robin [2, § 8.2.4], Interleaved Weighted Round Robin [10] and Deficit Round Robin [11].

The oldest generic work, in [12], is a framework to transform service curve based on packet sizes, devoted to a better modeling of packet-based scheduling policies, not an enhancement of generic service curve.

The result closer to the ones presented in this paper is in [4]. It also takes into account the effect of transmission at full line speed (C) up to the end of the packet, independent of the scheduling policy and applicable for packets of any size, but applicable only for rate-latency service curves. It proves that, if $\beta = \beta_{R,L}$, and if the horizontal deviation computes an upper bound on delay d, then $d - Q_l$ is also a valid bound with

$$Q_l = l\left(\frac{1}{R} - \frac{1}{C}\right). \tag{26}$$

VI. Illustration on some examples

Single node, constant packet size, static priority: A first comparison between Theorem 1 and [4] has been plotted in Figure 1. Applying Theorem 1 to β_C gives the bound $d^h = hdev(\alpha_C, \hat{\beta}_C) = 3.5$ms whereas [4] gives $d^Q = hdev(\alpha_C, \beta_C^{lin}) - Q_l = 9.5 - 2 = 7.5$ms. Nevertheless, in this case, it exists better rate-latency service curves for C, and applying [4] on such a curve will give a better delay, but as illustrated by [11, Thm. 2], computing a good rate-latency lower bound is not a simple problem.

Single node: Consider for a second comparison a server offering a service $\beta_{R,L}$ to a flow whose minimal and maximal frame sizes are $l_{\min}, l_{\max}$ and with arrival curve conforming to a token-bucket of rate r and burst b, i.e. $\alpha(t) = rt + b$.

The delay bound without considering frame sizes is $d^h = hdev(\alpha, \beta) = L + \frac{b}{R}$.

Assume now that each packet is served at line speed C, and to ease interpretation, set $C = 4R$.

At first glance, the use of [4] requires a frame size l, since the gain Q_l depends on the frame size. Consider a frame of size $l_{\min}$, the bound $L + \frac{b}{R}$ is valid, and the gain is $Q_l = \frac{3}{4}\frac{l_{\min}}{R}$, leading to a bound $d_{\min} = L + \frac{b - l_{\min}\frac{3}{4}}{R}$. The same for a frame of size $l_{\max}$ leads to a bound, $d_{\max} = L + \frac{b - l_{\max}\frac{3}{4}}{R} < d_{\min}$.

To solve the paradox $d_{\max} < d_{\min}$, one has to state that, inside a given flow, the delay of a given frame is always

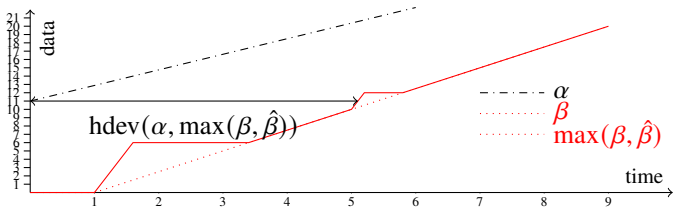

Fig. 2. Application of Theorem 1 and Corollary 1 with $\beta(t) = R\max(0, t - L)$, $\alpha(t) = rt + b$, $C = 10$, $R = 5/2$, $L = 1$, $l_{\min} = 6$, $l_{\max} = 9$, $b = 11$, $r = 15/8$

Parameters			Single server			Sequence of switches		
b	$l_{\min}$	$l_{\max}$	d^h	$d^{\mathcal{Q}}$	$\hat{d}$	e^h	$e^{\mathcal{Q}}$	$\hat{e}$
12	6	6	5.80	4	4	12.60	17.80	10.80
12	6	7	5.80	3.70	4.40	13.60	17.20	13.60
12	6	8	5.80	3.40	4.80	14.60	16.40	14.60
12	6	9	5.80	3.10	5.20	15.60	15.60	15.60
12	6	10	5.80	2.80	5.60	16.60	14.40	16.60
12	6	11	5.80	2.50	5.80	17.60	13.10	17.60
12	6	12	5.80	2.20	5.80	18.60	11.80	18.60
9	6	9	4.60	1.90	4.60	14.70	10.60	14.70
10	6	9	5	2.30	5	15	12.30	15
11	6	9	5.40	2.70	5.10	15.30	14	15.30
12	6	9	5.80	3.10	5.20	15.60	15.60	15.60
13	6	9	6.20	3.50	6.20	15.90	17.17	15.90
12	6	6	5.80	4	4	12.60	17.80	10.80
12	7	7	5.80	3.70	4.30	13.60	17.20	12.10
12	8	8	5.80	3.40	4.60	14.60	16.40	13.40
12	9	9	5.80	3.10	4.90	15.60	15.60	14.70
12	10	10	5.80	2.80	5.20	16.60	14.40	16
12	11	11	5.80	2.50	5.50	17.60	13.10	17.30
12	12	12	5.80	2.20	2.20	18.60	11.80	15

TABLE I
DELAY BOUNDS USING VALUES $C = 10$, $R = 5/2$, $L = 1$, $r = 15/8$.

smaller that the one of any larger frame. Then, $d_{\max}$ is a valid bound for any frame, and let $d^{\mathcal{Q}}$ denote this bound.

Conversely, the application of $\hat{\beta}$ from Theorem 1 is direct, and leads to $\hat{d} = hdev(\alpha, \max(\hat{\beta}, \beta))$ but there is no simple analytic solution. Then, the values of the bounds d^h, $d^{\mathcal{Q}}$ and d' are reported in Table I for several values of burst b, minimal and maximal packet sizes $l_{\min}$ and $l_{\max}$. In a first series, $b = 12$ and $l_{\min} = 6$ while $l_{\max}$ goes from 6 to 12, in a second series, $l_{\min} = 6$, $l_{\max} = 9$ while b goes from 9 to 13, and in a third, $b = 12$ and $l_{\min} = l_{\max}$, both going from 6 to 7.

It can be seen that the $d^{\mathcal{Q}}$ always gives the best bound, that the quality of $\hat{d}$ decreases whereas the difference between l_{min} and $l_{\max}$ increases. Also notice that the quality of $\hat{d}$ decreases with the rest of the Euclidean division of the burst by $l_{\min}$.

Multiple nodes: Consider now that the flow crosses a sequence of three store-and-forward switches, where each switch is made of one element that stores bits up to having one full packet (packetizer), and then forwards it to a queue with a per-bit minimal service $\beta_{R,L}$. A bound on the end-to-end delay of the flow can be computed either as the sum of the delay bound in each server, or using the "pay burst only once" principle (PBOO), stating that when a flow crosses a sequence of two servers of respective service β_1, β_2, the end-to-end delay is bounded by $hdev(\alpha, \beta_1 * \beta_2)$. To use it in a store-and-forward context, one may use the fact that the sequence of a per-bit service β and a packetizer offers a service $\beta - l_{\max}$. Using these two results permit us to compute a PMOO bound e^h using only the service curves, and a PMOO bound $\hat{e}$ that uses Corollary 1. The bound $e^{\mathcal{Q}}$ can be computed using the sum of local delays. The script used for this example can be found at https://doi.org/10.5281/zenodo.5226867 and interpreted using an online interpreter [13].

The results are presented in Table I. Like for the single node, the quality of $\hat{e}$ is highly dependent of the difference between $l_{\min}$ and $l_{\max}$, and the improvement w.r.t. e^h can be null (when $b = 12$, $l_{\min} = 6$, and $l_{\max} \geq 7$ or when $l_{\min} = 6$, $l_{\max} = 9$). The enhancement provided by [4] is dependent of the size of the packet size w.r.t. the burst size.

To sum up, both results in [4] and Corollary 1 try to save "one packet time" per server, but the efficiency of Corollary 1 depends on the ratio $l_{\min}/l_{\max}$. On the opposite, the result in [4] can not be used to pay the burst only once in case of sequence of servers, and its efficiency decreases when the size of burst is large w.r.t. the packet size (and when the path length increases).

VII. Conclusion

This paper presents a generic result that enhances the service curve of a server whose per packet transmission rate is lower bounded, whatever the scheduling policy or the service curve is. This result can be applied in more contexts than already existing results, but its gain decreases when the difference between the maximal an minimal packet size increases.

References

[1] J.-Y. Le Boudec and P. Thiran, *Network Calculus*, ser. LNCS. Springer Verlag, 2001, vol. 2050. [Online]. Available: https://leboudec.github.io/netcal/

[2] A. Bouillard, M. Boyer, and E. Le Corronc, *Deterministic Network Calculus – From theory to practical implementation*. Wiley, 2018.

[3] R. Davis, A. Burns, R. Bril, and J. Lukkien, "Controller area network (CAN) schedulability analysis: Refuted, revisited and revised," *Real-Time Systems*, vol. 35, pp. 239–272, 2007.

[4] E. Mohammadpour, E. Stai, and J.-Y. Le Boudec, "Improved delay bound for a service curve element with known transmission rate," *IEEE Networking Letters*, pp. 1–1, 2019.

[5] M. Boyer, N. Navet, and M. Fumey, "Experimental assessment of timing verification techniques for AFDX," in *Proc. of the 6th Int. Congress on Embedded Real Time Software and Systems*, Toulouse, France, February 2012. [Online]. Available: https://hal.archives-ouvertes.fr/hal-02189869

[6] M. Boyer, A. Graillat, B. Dupont De Dinechin, and J. Migge, "Bounding the delays of the mppa network-on-chip with network calculus: models and benchmarks," *Performance Evaluation*, July 2020.

[7] W. . Mangoua Sofack and M. Boyer, "Non preemptive static priority with network calculus: Enhancement," in *Proc. of the Workshop on Network Calculus (WoNeCa 2012)*, Kaiserslautern, Germany, March 2012.

[8] J.-P. Georges, E. Rondeau, and T. Divoux, "Evaluation of switched ethernet in an industrial context by using the network calculus," in *4th IEEE Int. Workshop on Factory Communication Systems (WFCS 2002)*. IEEE, 2002, pp. 19–26.

[9] M. Boyer, G. Stea, and W. Mangoua Sofack, "Deficit round robin with network calculus," in *Proc. of the 6th Int. Conf. on Performance Evaluation Methodologies and Tools (ValueTools)*, France, 2012.

[10] S. M. Tabatabaee, J.-Y. Le Boudec, and M. Boyer, "Interleaved weighted round-robin: A network calculus analysis," *IEICE Transactions on Communications*, 2021.

[11] S. M. Tabatabaee and J.-Y. Le Boudec, "Deficit round robin: a second network calculus analysis," in *Proc. of the 27th Real-Time and Embedded Technology and Application Symposium (RTAS 2021)*, 2021.

[12] A. Bouillard, N. Farhi, and B. Gaujal, "Packetization and packet curves in network calculus," in *Proc. of the 6th Int. Conf. on Performance Evaluation Methodologies and Tools (ValueTools 2012)*, France, 2012.

[13] "RealTime-at-Work online Min-Plus interpreter for Network Calculus," https://www.realtimeatwork.com/minplus-playground.