The lost equation of Network Calculus

Damien Guidolin Pina, Marc Boyer

The lost equation of Network Calculus

Damien GUIDOLIN--PINA
RealTime-at-Work
Nancy, FRANCE
https://orcid.org/0000-0003-1149-0861

Marc BOYER
ONERA/DTIS, Université de Toulouse
F-31055 Toulouse, France
https://orcid.org/0000-0003-0344-6991

***Abstract*—Network Calculus is a theory designed to analyze real-time systems, especially real-time networks. It offers a unified, extendable mathematical framework handling flows crossing servers, and deriving upper bounds on delays and memory usage. The real-time research community globally agrees that network calculus has powerful mathematical roots. However, the criticism is that the semantics are hidden behind mathematical details, limiting its adoption by non-specialists. In particular, the semantics of a server behavior was unclear to derive from most used equations. This paper claims that both formal and intuitive definitions of the server were lost in the literature. This paper presents it in a comprehensive way and gives new proof, adding a few new minor results. It also presents the history of the main notions through the state of the art.**

I. Introduction

Network Calculus (NC) is a theory that has been first designed to compute bound on network delays [1], [2], making a specific use of the (min, plus) dioid [3], [4]. It has been extended to also compute the response time of task scheduling and then named Real-Time Calculus (RTC) [5]–[7]. A global overview can be found in [8], [9].

As will be formally defined in Section II, Network Calculus is based on the notion of cumulative arrival curve A, representing input data flow, and the notion of server, S, that transforms the arrival A into a departure D.

Performance bounds on delay, and memory usage, can be derived from contracts on A and S. The contract on A is called an *arrival curve* or *envelope*, denoted α. There are several kinds of service contracts, minimal strict service, often denoted β_S, minimal min-plus service, often denoted β_m, variable capacity node, often denoted β_{vnc}, maximal min-plus service, often denoted β^M, and shaping curves, denoted σ.

Such models set constraints on the possible departures associated to an arrival A and a server S, but do not (in general) explicitly associate one departure D to an arrival A.

Conversely, in real-time calculus [5], an explicit notion of capacity function C of a server is given, with also a contract on workload (named "request curves", also denoted α), and contract on capacity (named "delivery curves", also denoted β).

This notion of capacity is also the root of the definition of the *variable capacity node* [8, § 4.3.2].

Both branches have developed similar results, and it has been proved in [10] that RTC is equivalent to the notion of minimal strict service for contracts with finite values (cf. Section IV for details).

In this paper, we show clearly how the notion of capacity of a server is related to the notion of variable capacity nodes.

This paper claims that this capacity is the core of the service notion (as stated by RTC) and can be formally defined simply and intuitively (by rewriting a result on variable capacity nodes from [8]). Moreover, it provides an elegant expression of the departure function.

The paper aims to ease the understanding of network calculus.

This paper first presents a recap of Network Calculus' main definitions (Section II). Then Section III derives, from the notion of server *capacity*, the expression of a server output. Last, Section IV presents the history of the different notions.

II. Network Calculus reminder

Network Calculus theory is rooted in the seminal work of [1], [2]. From these works, a large area of research has been developed [8], [9], [11], [12], with some small variations. Here is a small recap of Network Calculus in its modern form.

The Network Calculus theory is a theory based on the min-plus dioid. It deals with functions from time to data amount, e.g. A, such that $A(t)$ represents the total amount of data observed up to t through an observation point of the network. These functions, called *cumulative curves*, are non-decreasing, from $\mathbb{R}_+$ to $\mathbb{R}_+$, and piecewise continuous. The set of cumulative curves is noted $\mathcal{C}$.

Since the exact behavior is in general unknown at design time, the Network Calculus theory uses envelopes of the cumulative curves called *arrival curves*. A function $\overline{\alpha}_A$ (resp. $\underline{\alpha}_A$) is a maximal (resp. minimal) arrival curve of cumulative curve A if $\forall d, t \in \mathbb{R}_+ : \underline{\alpha} \leq A(t+d) - A(t) \leq \overline{\alpha}(d)$.

Also, the elements of the network are modeled by *servers*. A server is a left-total[1] relation, associating to each arrival at least one departure D such that $\forall (A, D) \in S : D \leq A$, where $D \leq A$ represents the fact data are forwarded *after* being received. Note that a server is not a deterministic function[2], it is just a relation.

[1] A relation S is left-total when $\forall A, \exists D$ such that $(A, D) \in S$

[2] For example, sending back-to-back two frames of size L or a single frame of size $2L$ leads to the same amount of data, so the same cumulative function A whereas the server can react with different behaviors and different departure cumulative curves

An interval I is a backlogged period (in the equation BP) for $(A, D) \in S$ if $\forall t \in I, A(t) - D(t) > 0$ (since $A(t) - D(t)$ represents the backlog at time t).

Whereas there exist only two kinds of contracts on cumulative curves, there are five main contracts on server behavior. The server S can:

1) offer a min-plus minimal service curve β_m if
$$\forall (A, D) \in S : D \geq A * \beta. \tag{1}$$
2) offer a maximal service curve β^M if
$$\forall (A, D) \in S : D \leq A * \beta^M. \tag{2}$$
3) be a σ-shaper if
$$\forall (A, D) \in S \implies D \leq D * \sigma. \tag{3}$$
4) offer a strict minimal service curve β_S if
$$\forall (A, D) \in S, \forall (s, t] \text{ a } BP \text{ }, D(t) - D(s) \geq \beta_S(t - s) \tag{4}$$
5) be a variable capacity node of curve β_{vnc} if
$$\begin{aligned} &\forall (A, D) \in S, \exists C, \\ &\quad \forall t \geq 0 : D(t) = \inf_{0 \leq s \leq t} A(s) + C(t) - A(s), \\ &\qquad \forall 0 \leq s \leq t : C(t) - C(s) \geq \beta_{vns}(t - s) \end{aligned} \tag{5}$$

where $*$ is the min-plus convolution defined by $\forall f, g \in \mathcal{F}$, $(f * g)(t) = \inf_{0 \leq s \leq t} \{f(s) + g(t - s)\}$.

However, as it is said previously, they all are stationary functions: they are independent of the time and only consider duration.

III. A new formulation of capacity-based server

This section provides a (partially new) definition and characterization of a server. The links with past definitions will be provided in Section IV.

Definition 1 (Capacity functions)**.** *A* capacity function *is a function $\hat{C} : \mathbb{R}_+ \times \mathbb{R}_+ \to \mathbb{R}$ such that $s \leq t \implies \hat{C}(s, t) \geq 0$ and that respects the Chasle's relation: $\forall s, t, u \in \mathbb{R}_+ : \hat{C}(s, t) + \hat{C}(t, u) = \hat{C}(s, u)$.*

Definition 2 (Capacity-based server)**.** *Let A a cumulative curve, S a server, and $\hat{C}$ a capacity function. Then S partially offers to A the capacity $\hat{C}$ if*
$$\text{for all interval } [u, t] : D(t) - D(u) \leq \hat{C}(u, t). \tag{6}$$
Let $\mathcal{D} = \left\{ X \in \mathcal{C} \,\middle|\, D \leq A, \forall u \leq v : D(t) - D(u) \leq \hat{C}(u, t) \right\}$ the set of possible departures. Then S offers to A the full capacity $\hat{C}$ if $D = \sup \mathcal{D}$,

We claim that this definition of server captures both a clear and formal way of the notion of service: the departure can never be more than the capacity of the server, and the full capacity is the maximal possible departure respecting the server capacity.

Definition 3 (Capacity cumulative function)**.** *If $\hat{C}$ is a capacity function, the capacity cumulative function $C : \mathbb{R}_+ \to \mathbb{R}_+$ is defined by $\forall t : C(t) = \hat{C}(0, t)$. This function is non-decreasing.*

Conversely, from a non-decreasing function $C : \mathbb{R}_+ \to \mathbb{R}_+$, one can define $\hat{C}(s, t) = C(t) - C(s)$, so in the following, we will call "capacity function" both $\hat{C}$ and C.

Theorem 1 (Output of a capacity-based server)**.** *Let A a cumulative curve, S a server, and $\hat{C}$ a capacity function. If S offers to A the full capacity $\hat{C}$, then*
$$D = C + (A - C) * 0 \quad \textit{and} \quad D \in \mathcal{D}. \tag{7}$$
Moreover, if A and C are both left-continuous (resp. right continuous), D also is.

Note that eq. 7 is not new by itself: the output of an RTC component was already defined in [5] as
$$D(t) = \inf_{0 \leq u \leq t} \{A(u) + C(t) - C(u)\}, \tag{8}$$
and since $\min_{0 \leq u \leq t} \{R(u) + C(t) - C(u)\} = C(t) + \min_{0 \leq u \leq t} \{A(u) - C(u) + 0(t - u)\} = C(t) + ((A - C) * 0)(t)$, this is just a rewriting. Theorem 1 is neither new since it was already given in [8], as will be detailed in Section IV.

The result can also be written $D(t) = C(t) - \sup_{0 \leq s \leq t} C(s) - A(t)$ to highlight the fact that the output is the capacity of the server minus the part of the capacity that overtakes the input.

Continuity is often an annoying technical detail in network calculus, but Definition 2 here is independent of continuity, and Theorem 1 shows that continuity choice (left or right) is preserved.

Proof. Let A be a cumulative curve and $C(t)$ the amount of service that the server can at most offer up to t. Let us introduce $\widetilde{D} = C + (A - C) * 0$. Then, we first have to prove that $\widetilde{D}$ satisfies the conditions of a cumulative curve and of Definition 2 and secondly that it is the supremum.

1) $\widetilde{D}$ is a non-decreasing function: Suppose there exists $v, w \in \mathbb{R}_+, v \leq w$ such that $\widetilde{D}(w) - \widetilde{D}(v) < 0$. Then, it means
$$\begin{aligned} &C(w) - C(v) - \inf_{0 \leq s \leq v} \{A(s) - C(s)\} \\ &+ \inf_{0 \leq s \leq w} \{A(s) - C(s)\} < 0 \\ &\Leftrightarrow \; C(w) - C(v) - \inf_{0 \leq s \leq v} \{A(s) - C(s)\} \\ &+ \left(\inf_{0 \leq s \leq v} \{A(s) - C(s)\} \wedge \inf_{v \leq s \leq w} \{A(s) - C(s)\} \right) < 0 \end{aligned}$$
Note that $(x \wedge y) - x = 0 \wedge (y - x)$. Then,
$$\begin{aligned} &\Leftrightarrow \; C(w) - C(v) \\ &+ 0 \wedge \left(\inf_{v \leq s \leq w} \{A(s) - C(s)\} - \inf_{0 \leq s \leq v} \{A(s) - C(s)\} \right) < 0 \\ &\Leftrightarrow \; 0 \wedge \left(\inf_{v \leq s \leq w} \{A(s) - C(s)\} \right. \\ &\left. - \inf_{0 \leq s \leq v} \{A(s) - C(s)\} \right) < C(v) - C(w) \end{aligned}$$

But $v \leq w$ and C is nondecreasing. So,

$$\begin{aligned}
&\implies \inf_{v\leq s\leq w}\{A(s)-C(s)\}\\
&-\inf_{0\leq s\leq v}\{A(s)-C(s)\} < C(v)-C(w)\\
&\Leftrightarrow \inf_{v\leq s\leq w}\{A(s)-C(s)\}\\
&< \inf_{0\leq s\leq v}\{A(s)-C(s)+C(v)-C(w)\}
\end{aligned}$$

As it is infinium, it means that

$\exists t \in [v,w], \forall s \in [0,v]$,

$A(t)-C(t) < A(s)-C(s)+C(v)-C(w)$

In particular for $s=v$:

$\exists t \in [v,w], A(t)-C(t) < A(v)-C(w)$

$\implies \exists t \in [v,w], A(t)-A(v) < C(t)-C(w)$

But $t \in [v,w]$ and A and C are nondecreasing. So, $A(t)-A(v) \geq 0$ and $C(t)-C(w) \leq 0$. Consequently, $\widetilde{D}$ is nondecreasing.

2) $\widetilde{D} \leq A$: Let $t \in \mathbb{R}_+$, By definition of the infinium,

$$\begin{aligned}
&\inf_{0\leq s\leq t}\{A(s)-C(s)\} \leq A(t)-C(t)\\
\Leftrightarrow\ & C(t)+(A-C)*0(t) \leq A(t).
\end{aligned}$$

3) $\forall t,u \in \mathbb{R}_+, u \leq t : \widetilde{D}(t)-\widetilde{D}(u) \leq C(t)-C(u)$: Let $\forall t,u \in \mathbb{R}_+, u \leq t$, as $[0,u] \subseteq [0,t]$,

$$\inf_{0\leq s\leq t}\{A(s)-C(s)\} - \inf_{0\leq s\leq u}\{A(s)-C(s)\} \leq 0$$

by adding $C(t)-C(u)$ on both sides:

$\Leftrightarrow\ \widetilde{D}(t)-\widetilde{D}(u) \leq C(t)-C(u)$.

4) $\widetilde{D} \in \mathcal{D}$: direct from 6–3

5) $\widetilde{D} = \sup \mathcal{D}$: By contradiction, assume $\exists D$ which satisfies the conditions of Definition 2 and $\exists t \in \mathbb{R}_+$ such that $D(t) > \widetilde{D}(t)$. Let us introduce $2\varepsilon = D(t)-\widetilde{D}(t)$. Then,

$$D(t)-\varepsilon > \widetilde{D}(t) = C(t) + \inf_{0\leq s\leq t}\{A(s)-C(s)\}$$

But, $\forall \eta > 0, \exists s_\eta$ such that $\inf_{0\leq s\leq t}\{A(s)-C(s)\} > A(s_\eta)+C(s_\eta)-\eta$ and in particular for $\eta=\varepsilon$. Then,

$$D(t)-\varepsilon > C(t)+A(s_\varepsilon)-C(s_\varepsilon)-\varepsilon$$

But, D satisfies the first condition: $D \leq A$, then

$$\begin{aligned}
&\implies D(t) > C(t)+A(s_\varepsilon)-C(s_\varepsilon)\\
&\implies D(t) > C(t)+D(s_\varepsilon)-C(s_\varepsilon)\\
&\implies D(t)-D(s_\varepsilon) > C(t)-C(s_\varepsilon)
\end{aligned}$$

Consequently, D doesn't satisfy the second condition: $\forall t,u \in \mathbb{R}_+, u \leq t : D(t)-D(u) \leq C(t)-C(u)$.

6) $\widetilde{D}$ has the same continuity as A and C: First, the sum and the subtraction is stable regarding the continuity. Also, according to [13], $\forall f,g$ two functions, if they are both right-continuous, or both left-continuous, then the convolution has the same continuity. As the function $0=t \mapsto 0$ is left and right continuous, $(A-C)*0$ has the same continuity as A and C. Consequently, $\widetilde{D}$ has the same continuity as A and C.

□

Definition 4 (Capacity bounds)**.** *Let $\hat{C}$ a capacity function. A pair $\underline{\beta}, \overline{\beta}$ is a capacity min/max service pair if*

$$\forall s \leq t : \underline{\beta}(t-s) \leq \hat{C}(s,t) \leq \overline{\beta}(t-s). \tag{9}$$

Then, S is a variable capacity node of curve $\underline{\beta}$ and is a $\overline{\beta}$-shaper.

IV. State of the art

A. History of service notion

The first work on network calculus [1], [2] where considered only pure rate service functions (i.e. $\beta(t) = Rt$ with R the link rate), which is time-invariant (for all $s \leq t$, $s' \leq t'$, if $t-s = t'-s'$, $Rt-Rs = Rt'-Rs'$). This notion is generalized in [14, Eq. (13)] to any kind of function, but with a time-invariant contract. Using modern notations, it states that it should exist a function β such that for any instant t, it exists $s \leq t$ such that there is no backlog at s ($A(s)=D(s)$) and $D(t)-D(s) \geq \beta(t-s)$. This is a kind of time-invariant property since for any t,t', and corresponding s,s', if $t-s=t'-s'$, then the same contact is offered. The service is dependent on the duration of the backlogged interval, not on the starting instant of the intervals. Notice that this definition is currently called *weakly strict* service [9], [10], not presented in this paper.

In [3], the "no backlog on s" condition is removed, leading to the introduction of min-plus convolution and the min-plus minimal service. Linking the definition with the min-plus operators allow to use of the rich set of associated results, but it also makes it more difficult to have an intuition of the semantics of the notion of service. This is also a time-invariant notion.

A time dependant notion, the *capacity function* C is introduced in [5], when defining the *Real-Time Calculus* (RTC) with the following semantics "$C(t)$ represents the maximum amount of computation that could be delivered up to time t (if the processor runs under full load)." The fundamental relation $D = \inf_{0\leq s\leq t} A(s)-C(t)-C(s)$ is given, without any long justification. The *delivery curve* is a function β such that $\forall s \leq t : C(t)-C(s) \geq \beta(t-s)$. This notion of capacity is more formally introduced in [7]. In § 2.3.1, is introduced a "differential service function" $\hat{C}$ with two arguments such that $\hat{C}(s,t)$ represents denotes "the sum of available resource units, e.g. processor cycles or transmittable bits on a bus, in the time interval $[s,t)$." The cumulative function $C(t) = \hat{C}(0,t)$ is also defined. But the formal definitions of the service are given in [7, § A.4.1]: to a given arrival A (and the associated $\hat{A}$), the departure $\hat{D}$ is defined using the set of equations

$$B(s) = A(s) - D(s), \tag{10}$$

$$\hat{D}(s,t) = \hat{C}(s,t) - \hat{C}'(s,t), \tag{11}$$

$$\hat{C}'(s,t) = \sup_{s\leq u\leq t}\left\{\hat{C}(s,u)-R(s,u)-B(s),0\right\}. \tag{12}$$

Then, several results are formally defined using these equations.

The notion of *variable capacity* is introduced in [8, § 4.3.2] also as a function C such that "$C(t)$ is the total capacity available to the flow between times 0 and t". Then, a *variable capacity node* is defined as the maximal output D satisfying $A \leq D$ and $\forall 0 \leq s \leq t : D(t) - D(s) \leq C(t) - C(s)$ (which is exactly the Definition 2). Then, using some very general theorem on "upper semi-continuous operator" [8, Thm. 4.3.1], it shows that $D = \inf_{0 \leq s \leq t} A(s) - C(t) - C(s)$. The proof does not consider continuity issues.

The notion of *strict* service is introduced in [15].

Proof that a server with a strict service β is also a service with a min-plus service β can be found in [8].

In [10] and [16], it is shown that a variable capacity node of the curve, β is equivalent to an RTC capacity node with delivery curve β. And also that the variable capacity is a stronger property than the strict service property, but for a function β such that $\beta \oslash \beta$ has only finite values (with $(f \oslash g)(t) = \sup_{u \geq t} f(t+u) - g(u)$), both notions are equivalent (more will be discussed further).

Also, note that all notions collapse in the case of a constant rate capacity $C(t) = Rt$ for some R.

B. More on VNC and strict service

It has been shown in [10] that any server being a variable capacity node of curve β also offers a strict service of curve β, meaning that VNC is a stronger property than strict service and that both notions collapse when $\beta \oslash \beta$ has only finite values. We may wonder what this means in practice this restriction.

Consider d such that $(\beta \oslash \beta)(d) = \infty$, where can come from? Either there exists u such that $\beta(u+d) = \infty$ and $\beta(u)$ is finite. If β has only finite values, it means that $\forall M > 0, \exists u_M$ such $\beta(u_M + d) - \beta(u_M) \geq M$. If a server offers a variable capacity of curve β, then it exists C such that $C(u_M + d) - C(u_M) \geq M$. It means that the capacity offered by the server on an interval of duration d can be as large as possible. It somehow states that the "speed" of the server always increases. It does not imply that the server offers an infinite capacity but that this capacity tends to infinity on finite intervals.

Then, when considering systems with bounded capacity per interval (i.e., with some finite constants r, b such that $\forall t, d : C(t) - C(t) \leq rd + b$), both strict service and VCN nodes are equivalents.

V. Conclusion

Several mathematical definitions have been provided over the years to capture the service offered by a server. We claim that defining a server as the one offering its full capacity C to an arrival A is the one that best fits the engineer's intuition. This intuition was expressed by [5], [7] but the mathematical proof was provided by [8]. Unfortunately, this result did not received the full attention of the community, perhaps because it is presented as a corollary of a more generic and quite complex result.

This paper presents it as a more central definition and gives a direct proof.

We hope that this presentation will help in the understanding of what is network calculus. It shows that shaping and variable capacity nodes are the core properties and that other services are approximations, used for their mathematical properties.

Some illustrations on two example are provided in appendix: a TDMA system and an example from [17].

References

[1] 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.

[2] ——, "A calculus for network delay, part II: Network analysis," *IEEE Transactions on information theory*, vol. 37, no. 1, pp. 132–141, January 1991.

[3] J.-y. Le Boudec, "Network calculus made easy," Ecole Polytechnique Fédérale de Lausanne (EPFL), Technical Report EPFL-DI 96/218, December 1996.

[4] C.-S. Chang, "A filtering theory for deterministic traffic regulation," in *INFOCOM '97. Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Driving the Information Revolution., Proceedings IEEE*, vol. 2, Apr 1997, pp. 436–443 vol.2.

[5] L. Thiele, S. Chakraborty, and M. Naedele, "Real-time calculus for scheduling hard real-time systems." in *Proc. IEEE International Symposium on Circuits and Systems (ISCAS)*, 2000, pp. 101–104.

[6] E. Wandeler, A. Maxiaguine, and L. Thiele, "On the use of greedy shapers in real-time embedded systems," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 11, no. 1, p. 1, 2012.

[7] E. Wandeler, "Modular performance analysis and interface-based design for embedded real-time systems," Ph.D. dissertation, ETH Zurich, September 2006.

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

[9] A. Bouillard, M. Boyer, and E. Le Corronc, *Deterministic Network Calculus – From theory to practical implementation*. Wiley, 2018, no. ISBN: 978-1-119-56341-9.

[10] A. Bouillard, L. Jouhet, and E. Thierry, "Service curves in Network Calculus: dos and don'ts," INRIA, Research Report RR-7094, 2009. [Online]. Available: http://hal.inria.fr/inria-00431674/en/

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

[12] M. Fidler, "Survey of deterministic and stochastic service curve models in the network calculus," *IEEE Communications Surveys and Tutorials*, vol. 12, no. 1, pp. 59–86, First 2010.

[13] D. Guidolin-Pina and M. Boyer, "Looking for equivalences of the services between left and right continuity in the Network Calculus theory," Sep. 2022, working paper or preprint. [Online]. Available: https://hal.science/hal-03772867

[14] R. Cruz, "Quality of service guarantees in virtual circuit switched networks," *IEEE Journal on Selected Areas in Communications*, vol. 13, no. 6, pp. 1048–1056, Aug 1995.

[15] R. Cruz and C. Okino, "Service guarantees for window flow control," in *Proc. of the annual Allerton Conf. on communication control and computing*, vol. 34, 1996, pp. 10–21.

[16] A. Bouillard, L. Jouhet, and E. Thierry, "Comparison of different classes of service curves in network calculus," in *Proc. of the 10th International Workshop on Discrete Event Systems (WODES 2010)*, Technische Universität Berlin, August 30 - September 1 2010.

[17] S. Perathoner, E. Wandeler, L. Thiele, A. Hamann, S. Schliecker, R. Henia, R. Racu, R. Ernst, and M. G. Harbour, "Influence of different system abstractions on the performance analysis of distributed real-time systems," in *Proc. of the 7th ACM & IEEE Int. Conf. on Embedded Software (EMSOFT'07)*. ACM, 2007, pp. 193–202.

APPENDIX

In the case of a constant rate server (ie. a simple data link), all service notions collapse, so this VNC expression give the same result as the others.

Here are presented two examples that illustrate how this formula can capture the expected behaviour of a system in other contexts.

A. TDMA

Consider a Time-Division Multiple Access server with, for instance, an output speed rate of $1KB/s$ and access to the output only during intervals $[2n+1, 2(n+1)], n \in \mathbb{N}$. The capacity function is an alternation of constant slope and plateaus, and can be written as $C(t) =$

$$\begin{cases} 0 & \text{if } t = 0 \\ C(2n) & \text{if } t \in [2n, 2n+1], n \in \mathbb{N} \\ C(2n+1) + t - (2n+1) & \text{if } t \in [2n+1, 2(n+1)], n \in \mathbb{N} \end{cases} \tag{13}$$

The function C is depicted in Figure 1.

Consider now the arrival of four packets, the first of size 300B at time 0 (while the TDMA offer no access), the second of size 200B at time t=1.5 (during an active slot), the third of size 400B at time t=2.9 (during a no-service slot) and the fourth of size 100B at time t=3.2 (during an active slot). The expected behaviour is that the first and the third packets will start their transmission at start of next active slot, the second is received during an active slot after the transmission of the first and can start its transmission immediately, whereas the fourth is also received during an active slot, but during the transmission of the third packet and have to wait. This behaviour is depicted on upper part of Figure 1.

The cumulative curve A corresponding to this arrival pattern can be defined as

$$A(t) = \begin{cases} 0B & \text{if } t = 0 \\ 300B & \text{if } 1.5 > t > 0 \\ 500B & \text{if } 2.9 > t > 1.5 \\ 900B & \text{if } t > 2.9 \end{cases}$$

and is drawn in the lower part of Figure 1, with the cumulative departure $D = C + (A - C) \otimes 0$.

We can see that the shape of D perfectly fits the expected behaviour.

B. Static priority and task chain

In [17], several examples are provided to compare different methods. The first benchmark consists of four tasks, T1–T4, with constant execution time, running on two CPU, using a fully preemptive static priority, as illustrated in Figure 2. The parameters of the tasks are given in Table I.

Tasks T1, T2, T4 are purely periodic without offsets. The the end of execution of T2 releases T3. Such a system can be modelled using VCN. The constant CPU capacity can be modelled by $\beta(t) = t$. The cumulative functions of tasks T1, T2, T4 are simple staircase functions. And a fully preemptive

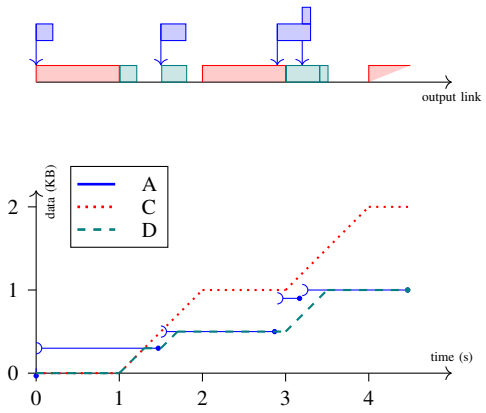

Fig. 1: TDMA system: on upper part, active and idle TDMA intervals (plain red boxes) plus packets arrivals (plain blue boxes with arrows) and transmissions (plain green boxes); on lower part, capacity function C, arrival A and departure D.

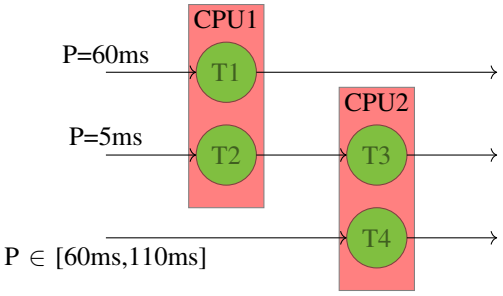

Fig. 2: Four tasks on two CPU

VNC node shared by two arrivals A_1, A_2 can be modelled as $D_1 = C - (A_1 - C) \otimes 0$ (the higher priority flow get the full capacity), and the global output $D_1 + D2$ also receive the full capacity, î.e. $D_1 + D2 = C + ((A_1 + A_2) - C) \otimes 0$, leading to $D_2 = C + (A_1 + A_2 - C) \otimes 0 - D_1$. And the same for the second CPU. The fact that T3 starts its execution when T2 is fully executed is modelled in network calculus as a packetizer, which is a flooring function in case of constant size. The change of size between T2 and T3 is done with a scaling function in network calculus. The curves are not drawn, but the set of expressions is given in listing 1, which can be run online [3].

The delay computed with this model is equal to 43ms, which is exactly the worst delay computed in [17] using model-checking of timed automata. In [17], no other method was able to get this value.

[3]https://www.realtimeatwork.com/minplus-playground

TABLE I: Table of the parameters of the four tasks on two CPU represented Figure 2

	T1	T2	T3	T4
Priority	high	low	low	high
Execution time	35ms	2ms	4ms	12 ms

```
T4 := 60
A1 := stair(0,60,35)
A2 := stair(0,5,2)
A4 := stair(0,T4,12)
C := affine(1,0)
D1 := C + (A1 - C)*zero
D2 := C + (A1 + A2 - C) * zero - D1
D4 := C + (A4 - C)*zero
floor := right-ext(stair(1,1,1))
A3 := (floor comp (D2/2))*4
D3 := C + (A3 + A4 - C) * zero - D4
hDev(A3, D3)
```

Listing 1: Set of equations for static priority example