



**HAL**  
open science

# A first Network Calculus model for Cyclic Queuing and Forwarding

Damien Guidolin–Pina, Marc Boyer

► **To cite this version:**

Damien Guidolin–Pina, Marc Boyer. A first Network Calculus model for Cyclic Queuing and Forwarding. Junior Researcher Workshop on Real-Time Computing, Jun 2023, DORTMUND, Germany. hal-04171262v2

**HAL Id: hal-04171262**

**<https://hal.science/hal-04171262v2>**

Submitted on 20 Dec 2024

**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.

# A first Network Calculus model for Cyclic Queuing and Forwarding

Damien GUIDOLIN--PINA, RealTime-at-Work, Nancy, France  
Marc BOYER, ONERA / DTIS, Université de Toulouse – F-31055 Toulouse – France

**Abstract**—Cyclic Queuing and Forwarding (CQF) is a mechanism of the IEEE TSN Group for providing guaranteed latency and jitter in a network. CQF deals with a common cycle time in the network and during one cycle, one queue is queuing the arriving frames when a second one is forwarding the previously queued frames. The roles are inverted at each cycle. To analyze this mechanism, we use the Network Calculus theory. It makes the computation of bounds on the delay possible and it is now largely used in the network analysis. In this paper, we show that the burstiness does not increase along the path and we provide a residual capacity for the low priority flows.

**Keywords**—TSN, CQF, Real-time network, Network Calculus

## I. INTRODUCTION

Nowadays, the real-time networks are complex and one of the challenges is to be able to bound end-to-end delay for all the frames in these networks. The IEEE TSN Group defines a set of mechanisms that makes it possible to configure a real-time network based on Ethernet. However, it is complicated to have a bound on the delay of each flow when we mix different mechanisms.

Among all of the TSN mechanisms, Cyclic Queuing and Forwarding (CQF) is the subject of our study. Briefly, CQF is a protocol which guarantees an end-to-end delay of a flow between  $(h-1)T$  and  $(h+1)T$  where  $h$  is the number of hops of the flow in the network and  $T$  the common cycle time of CQF. Further explanations are given in Section V. Even if the delay of CQF flows is easy to compute, a problem is that it is complicated to compute the end-to-end delay of the non-CQF frames sharing the same path of the CQF ones.

Several theories exist to bound the end-to-end delay of flows in a network. We choose the Network Calculus theory because it has proven itself in the analysis of real-time network, especially in the AFDX networks. However, the basic curves for analyzing the CQF have not yet been proven. Then, in this paper, we suggest a way to model the CQF server and an arrival curve of the output of this CQF server. Also, we provide the residual service curve offered to the low priority flows. The aim is to mix CQF with other mechanisms of TSN while being able to compute the end-to-end delay of each flow of each mechanism.

To do that, we first introduce notations (Section III), remind some useful definitions of the Network Calculus (Section IV) and explain the operating of CQF (Section V). Then, we give the main contribution: the arrival curve of the output of the CQF server and the residual service curve offered to the low priority flows in Section VI.

## II. RELATED WORK

The IEEE 802.1Q standard [1] specifies the CQF behavior. With a well configured network, it guarantees an end-to-end delay for the CQF flows between  $(h-1)T$  and  $(h+1)T$  (where  $h$  is the number of hop of the flow and  $T$  the cycle time of CQF).

However, TSN offer different mechanisms for ensuring real-time, and one of the main challenges is to analyze a network with different mechanisms (not only CQF for example). In the Network Calculus theory, we can bound the delay of each flow, but we need to have a model of the CQF flows to be able to do it. To the best of our knowledge, there doesn't seem to be a CQF model defined in network calculus.

However, CQF seems to generate interest in the community, as several extensions have been proposed in the literature, such as CQF 3-queues [2], Paternoster [3], Large-Scale Deterministic Network (LDN, [4], [5]), Cycle Specified Queuing and Forwarding (CSQF, [6]) and Tagged Cyclic Queuing and Forwarding (TCQF, [7]). A global survey, up to 2019, can be found in [8]. The future works can be able to analyze these extensions with other protocols, if the CQF protocol turns out to need these extensions.

Some works are done regarding the CQF operating like [9] where simulations are done with CQF in an automotive context. Also, [10] deals with CQF and try to make it practical in a network.

Regarding the analysis of a network implementing a mix of the TSN mechanisms, [11] [12] [13] are some examples. They work on networks with at least two different mechanisms and make an analysis of them, computing network performances as the end-to-end delay for instance. More generally, [14] presents a state of the art of the Network Calculus results related to TSN.

In this paper, we focus the model on the 'classic' CQF defined by the IEEE standard and try to express the curves needed to analyze a whole network using the Network Calculus theory.

## III. NOTATIONS

First, let  $\mathbb{R}$  and  $\mathbb{R}^+$  denote the sets of reals and non-negative reals respectively.  $\mathbb{N}$  denotes the set of integers. Also,  $\mathbb{N}^*$  denotes the set of non-zero integers.

Let  $\lfloor \cdot \rfloor$  and  $\lceil \cdot \rceil$  denote the floor and ceiling functions respectively such that  $\forall x \in \mathbb{R}$ ,  $\lfloor x \rfloor \in \mathbb{N}$ ,  $\lfloor x \rfloor \leq x < \lfloor x \rfloor + 1$  and  $\lceil x \rceil \in \mathbb{N}$ ,  $\lceil x \rceil - 1 < x \leq \lceil x \rceil$ . Also,  $\lceil \cdot \rceil^+$  represents the non-negative closure:  $\forall f$  a function,  $\lceil f \rceil^+(t) \stackrel{def}{=} \max(f(t), 0)$ . And  $\lceil \cdot \rceil_{\uparrow}$  represents the non-decreasing closure:  $\forall f$  a function,

$[f]_{\uparrow}(t) \stackrel{\text{def}}{=} \sup(f(t), 0)$ . Finally, to ease the expression, we denote the linear functions  $: R \in \mathbb{R}, \lambda_R : t \mapsto Rt$ .

#### IV. NETWORK CALCULUS REMINDER

The Network Calculus theory is a theory based on the min-plus dioid. It deals with functions which represent the 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 piece-wise continuous.

Then, to have a bound on the end-to-end delay, the Network Calculus uses envelopes of the cumulative curves called *arrival curves*. They are functions from  $\mathbb{R}^+$  to  $\mathbb{R}^+$ , non decreasing and piece-wise continuous and are such that, for  $A$  a cumulative curve, the maximal arrival curves  $\alpha_A$  is defined as

$$\forall d, t \in \mathbb{R}^+ : A(t+d) - A(t) \leq \alpha_A(d) \quad (1)$$

Also, the elements of the network are modeled by *servers*. A server is a left-total<sup>1</sup> relation, associating to each arrival at least one departure. We note  $(A, D) \in S$  any couple input flow:  $A$ , output flow:  $D$  of the server  $S$ .

A server  $S$  offers a min-plus minimal service of curve  $\beta$  if for all  $A$  and for all  $D$  such that  $D$  is the output of  $A$  through  $S$  then  $D \geq A * \beta$  where  $*$  is the min-plus convolution defined as  $\forall f, g$  two functions from  $\mathbb{R}^+$  to  $\mathbb{R}^+$ ,

$$f * g(t) \stackrel{\text{def}}{=} \inf_{0 \leq s \leq t} (f(t-s) + g(s)).$$

This curve  $\beta$  depends on the server global capacity policy and the arbitration policy between flows (CQF, Round Robin, etc.) and allows the computation of an upper bound on the delay of each flow.

Another kind of service defined in the Network Calculus theory is the strict minimal service. It ensures that the server is working if there is some backlog. An interval  $I$  is a backlogged period for  $(A, D) \in S$  if  $\forall t \in I, A(t) - D(t) > 0$ . Then, a server  $S$  offers a *strict* minimal service of curve  $\beta$  if  $\forall (A, D) \in S, \forall (s, t]$  backlogged period,  $D(t) - D(s) \geq \beta(t-s)$ .

In order to compute a end-to-end delay for a specific flow, the Network Calculus owns the residual service curve. It corresponds to the specific service offered to this flow (regarding the higher and lower priority, the policy, etc...).

More details and the entire theory can be found, for instance, in [15] or more recent in [16].

#### V. CQF OPERATING

The standard [1] specifies the operating of CQF. Among the eight queues of a port of a TSN switch, two are reserved for the operating of CQF. One is called 'even' and the other one is called 'odd'. A cycle time noted  $T$  divides the time into an alternating of intervals also called 'even' and 'odd'. We assume here that the cycle time is common to the whole network. During each even interval, all the arriving frames are stored in the odd queue meanwhile the frames previously stored in the even queue compete with other queues for transmission. During the odd intervals, the symmetric behavior

<sup>1</sup>A relation  $S$  is left-total when  $\forall A, \exists D$  such that  $(A, D) \in S$

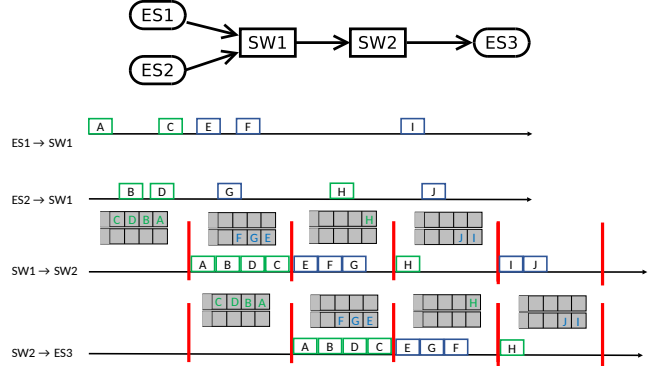


Fig. 1. Operation of CQF on an illustrative topology.

occurs. The standard also introduces a guard time, at the beginning and the end of each interval to avoid any emission. However, we simplify here the model of CQF and do not consider this guard band.

To illustrate the operating of CQF, Figure 1 shows a small simple network composed with two nodes ( $ES1$  and  $ES2$ ) forwarding frames to a third one ( $ES3$ ) through two switches ( $SW1$  and  $SW2$ ) using CQF.

The interval of duration  $T$  is delimited by the red vertical lines and the two queues of CQF are illustrated between these lines. During the first interval, the frames called  $A$  and  $C$  are sent by  $ES1$  and the frames  $B$  and  $D$  are sent by  $ES2$ . They are received by  $SW1$  during an odd interval and then stored in the even queue (the one on the top). During the following interval, an even one, the previously stored frames ( $A, B, C$  and  $D$ ) are forwarded to  $SW2$  meanwhile the frames  $E, F$  and  $G$  are stored in the odd queue (the one on the bottom). The operating is inverted at each cycle and is the same on the following CQF nodes.

Then, to model the CQF operating in the Network Calculus theory, we introduce a new definition: a CQF server. We consider here that the CQF flows are the higher priority flows. This hypothesis is justified by the literature where most of the papers uses CQF alone or as the higher priority flows. However, in our model we consider the lower priority flows and then a blocking factor due to them. This can happen if a lower priority flow is forwarded just before the opening of the CQF cycle.

*Definition 1 (CQF server):* We denote by  $S_{CQF}(T, R)$  a server offering a CQF policy of cycle time  $T$  and output port speed  $R$  where  $\forall (A, D) \in S_{CQF}, A$  a CQF flow:

- 1) The output during the first cycle is zero:

$$\forall u \in \mathbb{R}^+, u < T, D(u) = 0.$$

- 2) All queued frames in a cycle are emitted in the following one:

$$\forall k \in \mathbb{N}^*, D((k+1)T) - D(kT) = A(kT) - A((k-1)T).$$

- 3) The frames are emitted from the beginning of the cycle, until the queue is empty and at the speed of the output port  $R, \forall u \in \mathbb{R}^+$ :

$$D(u) = D(k_u T) +$$

$$\min(R[u - k_u T - b(k_u)]^+; A(k_u T) - A((k_u - 1)T)).$$

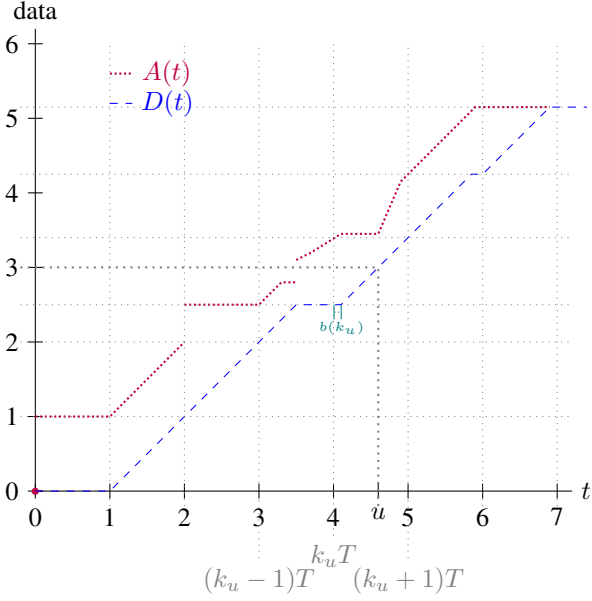


Fig. 2. Illustration of Definition 1 with a CQF server  $S_{CQF}(1,1)$ .

where  $k_u = \lfloor \frac{u}{T} \rfloor$  and  $b(k_u) \in [0, L_{\max}/R]$  is the blocking factor due to the lower priority flows of maximum size  $L_{\max}$  at the beginning of the cycle of  $u$ .

*Remark 1:* Note that under these conditions,  $D$  is continuous and is an alternation of constant segments and segments with a rate of  $R$ .

Figure 2 illustrates the definition of a CQF server. During the first cycle, the output of the server is zero. Then, the grid allows us to see the second rule. The quantity of data leaving the server during a cycle is equal to the quantity of data which entered in the server during the previous cycle (for instance  $D(2) - D(1) = 1 = A(1) - A(0)$ ). Finally, let us take  $u = 4.6$  for instance. The third rule is illustrated between  $k_u = 4$  and 5.  $D(4.6)$  is then equal to the quantity of data at the beginning of the cycle ( $D(4) = 2.5$ ) added by the data leaving the server between 4 and 4.6. However, a blocking factor due to the low priority flows can occur as it is illustrated by  $b(4) = 0.1$ . Then,  $D(4.6) = 2.5 + (4.6 - 4 - 0.1) \times 1 = 3$ .

## VI. CQF RESULTS

In this section, we give the main results: the arrival curve of the output of the CQF server and the residual service curve offered to the low priority flows.

First, we need a property showing that the output cumulative curve of the CQF server at a beginning of a cycle is equal to the input cumulative curve at the beginning the previous cycle.

*Property 1 (Equality at the start of cycles):* Let  $(A, D) \in S_{CQF}(R, T)$ . Then,  $\forall k \in \mathbb{N}, k > 0$ ,

$$D(kT) = A((k-1)T) \quad (2)$$

*Proof:* We will prove this property by induction. Let  $k$  be an integer,  $k > 0$ .

- $k = 1$ : According to the first rule of the CQF server and knowing that  $D$  is continuous,  $D(T) = 0 = A(0)$ .
- Supposing that  $D(kT) = A((k-1)T)$ : According to the second rule of the CQF server,  $D((k+1)T) - D(kT) =$

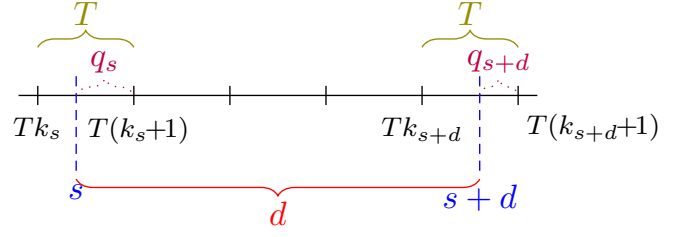


Fig. 3. Illustration of the notation introduced in Theorem 1. Plain curly brackets represent time durations; Dotted curly brackets are related to data amount.

$A(kT) - A((k-1)T)$ . Applying the induction hypothesis,  $D((k+1)T) = A(kT)$ .

- By induction,  $\forall k \in \mathbb{N}, k > 0, D(kT) = A((k-1)T)$ . ■

Then, we can express the arrival curve of the output flow of the CQF server according to the input arrival curve.

*Theorem 1 (Output arrival curve of CQF flows):* Let  $S$  be a server offering a CQF policy and  $\bar{\alpha}$  an arrival curve of a flow entering CQF server. Then, the curve defined as  $\bar{\alpha}' : d \mapsto \bar{\alpha}(\lceil \frac{d}{T} \rceil T) + L_{\max}$  is an arrival curve of the flow at server output.

It can be a bit surprising to see the  $L_{\max}$  term, related to lower priority flow, accounting in the arrival curve of the CQF flows. The reason is that, when looking only at intervals  $[t, t+d]$  such that  $t$  is an interval bound (i.e.,  $t = kT$  for some  $k \in \mathbb{N}$ ), the amount of CQF data can be bounded by  $\bar{\alpha}(\lceil \frac{d}{T} \rceil T)$ , but if any value of  $t$  is possible (which is the definition of an arrival curve), the CQF burst can be shifted by lower priority flow, which somehow increases the global burstiness when considering a value of  $t$  just after the transmission of the lower priority frame.

*Proof:* Let  $s, d$  be two non-negative reals. The aim of the proof is to upper bound the quantity  $D(s+d) - D(s)$ . We reuse the notations:  $k_s = \lfloor \frac{s}{T} \rfloor$  and  $k_{s+d} = \lfloor \frac{s+d}{T} \rfloor$  to simplify the expressions of the proof.

To illustrate this notation, let us see Figure 3 where  $s, s+d, k_s$  and  $k_{s+d}$  are illustrated. We can also see  $d$ , the distance between  $s$  and  $s+d$ .

Consequently, we have  $k_s T \leq s < (k_s + 1)T$  and  $k_{s+d} T \leq s+d < (k_{s+d} + 1)T$  and

$$\begin{aligned} k_{s+d} T - (k_s + 1)T &< d < (k_{s+d} + 1)T - k_s T \\ \implies k_{s+d} - k_s - 1 &< \left\lceil \frac{d}{T} \right\rceil \leq k_{s+d} - k_s + 1 \end{aligned}$$

As  $k_{s+d} - k_s - 1 \in \mathbb{N}$  and also  $k_{s+d} - k_s + 1 \in \mathbb{N}$ , there are two cases. Let us distinguish the two cases:

- 1)  $\lceil \frac{d}{T} \rceil = k_{s+d} - k_s + 1$ :

$$\begin{aligned}
D(s+d) - D(s) &\leq D((k_{s+d}+1)T) - D(k_s T) \\
&\stackrel{(2)}{=} A(k_{s+d}T) - A((k_s-1)T) \\
&\stackrel{(1)}{\leq} \alpha_A((k_{s+d} - k_s + 1)T) \\
&= \alpha_A \left( \left\lceil \frac{d}{T} \right\rceil T \right) \\
&\leq \alpha_A \left( \left\lceil \frac{d}{T} \right\rceil T \right) + L_{\max}.
\end{aligned}$$

2)  $\lceil \frac{d}{T} \rceil = k_{s+d} - k_s$ : Let us compare the positive quantities:

$$\begin{aligned}
q_s &= D((k_s+1)T) - D(s) \\
q_{s+d} &= D((k_{s+d}+1)T) - D(s+d).
\end{aligned}$$

These quantities are illustrated Figure 3. There are two possibilities:

a)  $q_s \leq q_{s+d}$ : We can decompose the initial quantity by

$$\begin{aligned}
D(s+d) - D(s) &= D(s+d) - D((k_{s+d}+1)T) \\
&\quad + D((k_{s+d}+1)T) - D((k_s+1)T) \\
&\quad + D((k_s+1)T) - D(s) \\
&= -q_{s+d} + D((k_{s+d}+1)T) - D((k_s+1)T) + q_s \\
&\leq D((k_{s+d}+1)T) - D((k_s+1)T) \\
&\stackrel{(2)}{=} A(k_{s+d}T) - A(k_s T) \\
&\stackrel{(1)}{\leq} \alpha_A((k_{s+d} - k_s)T) \\
&= \alpha_A \left( \left\lceil \frac{d}{T} \right\rceil T \right) \\
&\leq \alpha_A \left( \left\lceil \frac{d}{T} \right\rceil T \right) + L_{\max}.
\end{aligned}$$

b)  $q_s > q_{s+d}$ : In this case,  $q_s > 0$ . This imply that the CQF queue is still emitting after time  $s$ . Since CQF always starts emission at start of period, it means that we emit, a fortiori, some data between  $k_s T$  and  $s$ , i.e.  $D(s) - D(k_s T) = R(s - k_s T) + b(k_s)$ . Then  $(s - k_s T)R - L_{\max} \leq D(s) - D(k_s T) \leq (s - k_s T)R$ . Also,  $D(s+d) - D(k_{s+d}T) \leq ((s+d) - k_{s+d}T)R$ . Then,

$$\begin{aligned}
D(s+d) - D(s) &= D(s+d) - D(k_{s+d}T) \\
&\quad + D(k_{s+d}T) - D(k_s T) \\
&\quad + D(k_s T) - D(s) \\
&\leq ((s+d) - k_{s+d}T)R \\
&\quad + D(k_{s+d}T) - D(k_s T) \\
&\quad + (k_s T - s)R + L_{\max} \\
&= (d + (k_s - k_{s+d})T)R \\
&\quad + D(k_{s+d}T) - D(k_s T) \\
&\quad + L_{\max}
\end{aligned}$$

$$\begin{aligned}
&\text{as } \frac{d}{T} \leq \left\lceil \frac{d}{T} \right\rceil \\
&\leq \left( \left\lceil \frac{d}{T} \right\rceil + k_s - k_{s+d} \right) T R \\
&\quad + D(k_{s+d}T) - D(k_s T) + L_{\max} \\
&\leq D(k_{s+d}T) - D(k_s T) + L_{\max} \\
&\stackrel{(2)}{=} A((k_{s+d}-1)T) - A((k_s-1)T) + L_{\max} \\
&\stackrel{(1)}{\leq} \alpha_A((k_{s+d} - k_s)T) + L_{\max} \\
&= \alpha_A \left( \left\lceil \frac{d}{T} \right\rceil T \right) + L_{\max}
\end{aligned}$$

Consequently,  $\forall s, d \in \mathbb{R}^+$ ,

$$D(s+d) - D(s) \leq \alpha_A \left( \left\lceil \frac{d}{T} \right\rceil T \right) + L_{\max}.$$

Then,  $\alpha_A \left( \left\lceil \frac{t}{T} \right\rceil T \right) + L_{\max}$  is an arrival curve for the output of the CQF server.  $\blacksquare$

Finally, we want to know what is the residual service offered to the low priority flows.

*Theorem 2 (Residual service of lower priority flows):* Let  $S_{CQF}(R, T)$  be a server offering a CQF policy. The server offers a strict residual service to the lower priority frames of curve  $\beta^{res} = \left[ \lambda_R - \overline{\alpha'_{CQF}} \right]_+^+$  where  $\overline{\alpha'_{CQF}}$  is an arrival curve of the output CQF flows.

The proof is very similar to the static priority one, [16, Theorem 7.6] for instance. However, we have to do this proof because CQF is not work-conserving i.e., the server does not always emit frame if there is backlog (due to the CQF rules).

*Proof:* Let us introduce some notations regarding the output flows passing through the server:

- $CQF$ : the cumulative curve of the CQF flows,
- $L$ : the cumulative curve of the lower priority flows (than CQF flows).

Let  $s, t$  be two non-negative reals such  $]s; t]$  is a backlogged period of  $L$ .  $\forall u \in ]s; t]$ , we have

$$(CQF(u) + L(u)) - (CQF(s) + L(s)) \geq R \cdot (u - s).$$

Then,  $L(u) - L(s) \geq R \cdot (u - s) + CQF(s) - CQF(u)$ . As  $u \geq s$ , and the cumulative curves are non-decreasing, we have  $L(u) - L(s) \geq 0$ . Thus, we can add the non-negative closure to the right side of the previous equation and  $L(u) - L(s) \geq [R \cdot (u - s) + CQF(s) - CQF(u)]_+^+$ . By definition of the arrival curve,  $CQF(u) - CQF(s) \leq \alpha_{CQF}(u - s)$ . Then,  $L(u) - L(s) \geq [R \cdot (u - s) - \alpha_{CQF}(u - s)]_+^+$ . Finally,  $t \geq u$  and  $L$  is non-decreasing, so

$$\begin{aligned}
L(t) - L(s) &\geq L(u) - L(s) \\
&\geq [R \cdot (u - s) - \alpha_{CQF}(u - s)]_+^+ \\
&\geq \max_{u \in ]s; t]} [R \cdot (u - s) - \alpha_{CQF}(u - s)]_+^+ \\
&\geq [R \cdot (u - s) - \alpha_{CQF}(u - s)]_+^+ \\
&\geq [\lambda_R - \alpha_{CQF}]_+^+(t - s).
\end{aligned}$$

Consequently,  $\forall s, t \in \mathbb{R}^+, ]s; t]$  a backlogged period of  $L$ ,  $L(t) - L(s) \geq [\lambda_R - \alpha_{CQF}]_+^+(t - s)$  and  $[\lambda_R - \alpha_{CQF}]_+^+$  is a strict residual service curve for the lower priority flows.  $\blacksquare$

## VII. CONCLUSION

We first give a way to model CQF in the Network Calculus theory, with a definition of the CQF server. Then, we provide some curves according to CQF policy: we give an expression of the arrival output curve according to the input one and we give the residual service curve offered to the low priority flows. These curves make the end-to-end delay computation of all the frames from different policy mixed with CQF in the same path of the network possible.

This result deserves further work. First, we would like to investigate on the blocking factor appearing in the arrival curve bound. Even if it can not be neglected, a better bound may certainly exist. A second work consist in considering that CQF may not have the higher priority, and take into account the interference of higher priority flows. A third improvement consists in taking into account the guard band in the model.

## REFERENCES

- [1] IEEE standard for local and metropolitan area networks – bridges and bridged networks. IEEE Standard 802.1Q, IEEE, 2018.
- [2] N Finn, E Mohammadpour, J Zhang, and B Varga. Deterministic Networking (Detnet) Bounded Latency. RFC 9320, 2022.
- [3] Mick Seaman. Paternoster policing and scheduling. Technical report, IEEE, May 2019. Revision 2.1.
- [4] Li Qiang, Xuesong Geng, Bingyang Liu, Toerless Eckert, Liang Geng, and Guangpeng Li. Large-Scale Deterministic IP Network. Internet-Draft draft-qiang-detnet-large-scale-detnet-05, IETF, 2019.
- [5] Bingyang Liu, Shoushou Ren, Chuang Wang, Vincent Angilella, Paolo Medagliani, Sebastien Martin, and Jeremie Leguay. Towards large-scale deterministic ip networks. In *2021 IFIP Networking Conference (IFIP Networking)*, pages 1–9, 2021.
- [6] Mach Guoyi Chen, Gengn Xuesong, and Zhenqiang Li. Segment Routing (SR) Based Bounded Latency. Internet-Draft draft-chen-detnet-sr-based-bounded-latency-01, IETF, May 2019.
- [7] Toerless Eckert, Stewart Bryant, Andrew G. Malis, and Guangpeng Li. Deterministic Networking (DetNet) Data Plane - Tagged Cyclic Queuing and Forwarding (TCQF) for bounded latency with low jitter in large scale DetNets. Internet-Draft draft-eckert-detnet-tcqf-01, Internet Engineering Task Force, November 2022. Work in Progress.
- [8] Ahmed Nasrallah, Venkatraman Balasubramanian, Akhilesh S Thyagaturu, Martin Reisslein, and Hesham ElBakoury. Cyclic queuing and forwarding for large scale deterministic networks: A survey. *ArXiv, vol. abs/1905.08478*, 2019.
- [9] Luca Leonardi, Lucia Lo Bello, and Gaetano Patti. Performance assessment of the ieee 802.1qch in an automotive scenario. In *2020 AEIT International Conference of Electrical and Electronic Technologies for Automotive (AEIT AUTOMOTIVE)*, pages 1–6, 2020.
- [10] Jinli Yan, Wei Quan, Xuyan Jiang, and Zhigang Sun. Injection time planning: Making cqf practical in time-sensitive networking. In *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications*, pages 616–625, 2020.
- [11] Luxi Zhao, Paul Pop, Zhong Zheng, Hugo Daigmorte, and Marc Boyer. Latency analysis of multiple classes of avb traffic in tsn with standard credit behavior using network calculus. *IEEE Transactions on Industrial Electronics*, 68(10):10291–10302, 2021.
- [12] Jörn Migge, Jostexo Villanueva, Nicolas Navet, and Marc Boyer. Insights on the performance and configuration of avb and tsn in automotive ethernet networks. In *Proc. of the 9th European Congress on Embedded Real Time Software and Systems (ERTS<sup>2</sup> 2018)*, Toulouse, France, January 2018.
- [13] Luxi Zhao, Paul Pop, and Sebastian Steinhorst. Quantitative performance comparison of various traffic shapers in time-sensitive networking. *IEEE Transactions on Network and Service Management*, 19(3):2899–2928, 2022.
- [14] Lisa Maile, Kai-Steffen Hielscher, and Reinhard German. Network calculus results for tsn: An introduction. In *Proc. of the Information Communication Technologies Conference (ICTC 2020)*, pages 131–140. IEEE, 2020.
- [15] Jean-Yves Le Boudec and Patrick Thiran. *Network calculus: a theory of deterministic queuing systems for the internet*. Springer, 2001.
- [16] Marc Boyer, Euriell Le Corronc, and Anne Bouillard. *Deterministic network calculus: From theory to practical implementation*. John Wiley & Sons, 2018.