# Pay-Burst-Only-Once in Real-Time Calculus

Yue Tang[1], Yuming Jiang[2], Xu Jiang[1], Nan Guan[1]

[1]The Hong Kong Polytechnic University, Hong Kong [2]Norwegian University of Science and Technology, Norway

***Abstract*—Real-Time Calculus (RTC) is a powerful framework for modeling and analyzing complex networked real-time systems. RTC builds up on and shares many similarities with Network Calculus (NC), but some concepts are not completely the same in RTC and NC. One of the most important properties in NC is pay-burst-only-once, which can improve the precision of end-to-end performance analysis. Naturally, people would expect the pay-burst-only-once property to also hold in RTC. In fact, some existing work has used it in some performance analysis problems. Unfortunately, the pay-burst-only-once property has never been proved in RTC. There are even some results seeming to be against the pay-burst-only-once property in RTC. In this paper, we prove that the pay-burst-only-once property indeed holds in RTC.**

***Index Terms*—Real-Time Calculus, Network Calculus, Concatenation, Pay-burst-only-once**

## I. Introduction

Network Calculus (NC) is a theory analyzing queuing systems found in computer networks, whose focus is to provide performance guarantees [1], [2]. NC models traffic and service with arrival and service curves respectively, and performance bounds are derived with min-plus and max-plus algebra. NC has been widely and successfully used for modeling and analysis in networked systems such as in aeronautics industry for AFDX [3].

Real-Time Calculus (RTC) originates from NC and is also a queuing theory for performance analysis of real-time networked embedded systems [4]. Similar to NC, RTC uses arrival and service curves to model workload and resource (the comparison will be conducted in Section IV), and the performance analysis is also mainly conducted with min-plus and max-plus algebra. Compared with the traditional real-time scheduling theory, RTC is more expressive and can model a much wider range of realistic systems due to usage of much more general workload and resource models. At the same time, the models and analysis techniques of RTC generate closed-form analytical results, thus having higher analysis efficiency, compared to state-based modeling and analysis techniques such as model checking [5], [6]. All these advantages make RTC draw much attention from both real-time community and industry, and much work has been done to extend RTC for wider application [7], [8].

Pay-burst-only-once is a fundamental and powerful property in NC, which states that the delay bound obtained by considering the overall service curve of nodes a flow traverses is better (tighter) than that by summing up the delay bound of each individual traversed node. The pay-burst-only-once property relies on the concatenation property in NC, which proves that concatenating the service curve of a sequence of nodes generates the overall service curve for the whole system composed of these nodes. The pay-burst-only-once property is proved based on the concatenation property.

Since RTC inherits many concepts and properties from NC, people naturally expect the pay-burst-only-once property to also hold in RTC. Much work has already assumed it to be true and applies it for performance analysis. However, the pay-burst-only-once property has never been proved in RTC. Even worse, some existing results seem to be against the concatenation property in RTC [9] (recall that in NC the pay-burst-only-once property is proved based on the concatenation property). Therefore, it leaves an important open problem to find out whether the pay-burst-only-once property holds in RTC.

Our work gives the answer to the problem mentioned above. We prove that the pay-burst-only-once property indeed holds in RTC. The basic idea is that we first find some numerical relations between curves in RTC and NC and then prove the property in RTC indirectly by using the pay-burst-only-once property in NC.

Moreover, since the concatenation property does not hold in RTC, the service curve for the whole system can not be derived, and then the output arrival curves of the whole system seemingly can only be derived by calculating the output arrival curves one node after another. However, this process is so tedious, especially with large-scale systems. As a consequence, it raises a question whether we can find a once-for-all derivation for output arrival curves without knowing the exact overall service curve.

In our work, we solve the output arrival curve problem: we figure out how to compute the output arrival curves of a concatenated system with a once-for-all method. The basic idea is similar to the proof of the pay-burst-only-once property, and the difficulty lies in the calculation of lower output arrival curves in NC, which as far as we know have never been defined and calculated before.

The comparison between our contribution and existing work is summarized in Table I, where our work is marked in blue.

## II. Related work

In the development of NC, two branches have appeared: algebraic NC [11], which derives delay bound based on the composition of operators such as min-plus and max-plus

978-1-7281-3197-9/19/$31.00 © 2019 IEEE

| | Output arrival curve (a single node) | | Concatenation | Pay-burst-only-once | Output arrival curve with concatenation |
|---|---|---|---|---|---|
| RTC | upper output arrival curve | lower output arrival curve | about lower service curve | | |
| Status | derived in [10] | | proved to ***not*** hold in [9] | used but not proved | no results |
| | | | | proved to hold in our work | derived in our work |
| NC | output arrival curve | lower output arrival curve | about maximum/minimum service curve | | |
| Status | derived in [1] | derived in our work | proved to hold in [1] | proved to hold in [1] | same as for a single node |

TABLE I
THE COMPARISON OF OUR CONTRIBUTION AND EXISTING WORK

algebra [12], and optimization-based NC, which derives a linear program formulation whose solution bounds the end-to-end delay [13]. In our work, we focus on algebraic NC (called NC for short).

The service guarantee in NC is described with three types of service curve, the comparison of which is introduced in [9], [14], [15]. Although the comparison with curves in RTC is considered in these work, numerical relations are not given and maximum service curve is not considered.

In NC, the concatenation property plays an important role in deriving tighter delay bound. The first well-known application of the concatenation property is the pay-burst-only-once property, which avoids considering the analyzed flow's burst at each node it traverses. Another application is the pay-multiplexing-only-once property [16], [17], which extends the pay-burst-only-once property to cross-flows and convolves as much service curves as possible before subtracting the cross traffic. The influence of the concatenation property is further explored in [18], which stated that concatenation brings pessimism to delay bound under arbitrary multiplexing. The reason is that the convolution operation is not sensitive to the operation order so that the topological information is ignored, making the burst paid at nodes that the flow does not even traverse. All above work focuses on how to improve the application of concatenation for deriving tighter bounds and does not explore the results for lower bound of arrival curves.

In RTC, the pay-burst-only-once property has been adopted to derive tighter delay bound, which is first proposed in [10]. Then it is used in [19] for the delay bound calculation under operator-finitary RTC, and also in [20] to minimize the energy consumption for pipelined multiprocessor embedded systems. However, none of these includes the proof of correctness.

## III. NETWORK CALCULUS AND REAL-TIME CALCULUS BASICS

In this section, we introduce some basic definitions and properties in NC and RTC respectively.

### A. Basics in Network Calculus

In NC, the accumulated number of input bits on a flow from time 0 to $t$ is denoted by input function $R(t)$, which by definition is wide-sense increasing. Correspondingly, the number of output bits during the same interval is denoted by output function $R^*(t)$. The limitation of bits is denoted by arrival curve, which describes an upper constraint of the flow.

**Definition 1** (Arrival Curve in NC)**.** *Assume the input function of a flow is $R(t)$, then the flow has an arrival curve $\alpha^U$ iff*

$$\forall s \leq t, R(t) - R(s) \leq \alpha^U(t-s)$$

*. which is equivalent to*

$$R \leq R \otimes \alpha^U$$

*, where* $(f \otimes g)(\Delta) = \inf_{0 \leq \lambda \leq \Delta} \{f(\Delta - \lambda) + g(\lambda)\}$.

While arrival curve constrains the input flows, service curve describes the service guarantees provided to these flows by network nodes. There are three types of service curve in NC: minimum service curve, strict service curve, and maximum service curve.

**Definition 2** (Minimum Service Curve)**.** *The network node offers a minimum service curve $\beta$ to a flow with input function $R(t)$ and output function $R^*(t)$ iff*

$$R^* \geq R \otimes \beta$$

**Definition 3** (Strict Service Curve)**.** *The network node offers a strict service curve $\beta_{strict}$ to a flow with output function $R^*(t)$ iff during any backlogged period from $s$ to $t$, it holds*

$$R^*(t) - R^*(s) \geq \beta_{strict}(t-s),$$

**Definition 4** (Maximum Service Curve)**.** *The network node offers a maximum service curve $\gamma$ to a flow with input function $R(t)$ and output function $R^*(t)$ iff it holds*

$$R^* \leq R \otimes \gamma$$

Next we introduce how to obtain performance bounds in NC.

Assume that a flow with input function $R(t)$, output function $R^*(t)$, arrival curve $\alpha^U$, traverses a node that offers a minimum service curve $\beta$ and a maximum service curve $\gamma$. The virtual delay $d(t)$ for all $t$ satisfies:

$$d(t) = inf\{\tau \geq 0 : R(t) \leq R^*(t+\tau)\} \leq h(\alpha^U, \beta)$$

where

$$h(f,g) = \sup_{s \geq 0}\{\inf\{\tau \geq 0 : f(s) \leq g(s+\tau)\}\}$$

The output flow is constrained by output arrival curve

$$\alpha^{*U} = (\alpha^U \otimes \gamma) \oslash \beta$$

where $(f \oslash g)(\Delta) = \sup_{\lambda \geq 0}\{f(\Delta+\lambda) - g(\lambda)\}$,
or

$$\alpha^{*U} = \alpha^U \oslash \beta$$

when $\gamma$ is unknown.

After introducing the properties for a single node, we consider networked nodes. One of the important properties is concatenation, which concerns the calculation of service curve when a flow traverses several nodes.

**Property 1** (Concatenation)**.** *Assume that a flow traverses $m$ nodes $S_1, S_2, ..., S_m$ in sequence, each offering a minimum service curve $\beta_i$, a maximum service curve $\gamma_i$, $i = 1, 2, ...m$ to the flow respectively. Then the concatenation of the $m$ nodes offers a minimum service curve $\beta_1 \otimes \beta_2 ... \otimes \beta_m$ and a maximum service curve $\gamma_1 \otimes \gamma_2 ... \otimes \gamma_m$ to the flow.*

According to the concatenation property, the overall minimum service curve of networked nodes equals the convolution of minimum service curve offered by each node in the network. As a result, the output arrival curves and delay bound can be calculated in the same way as that of a single node, except replacing the minimum service curve with the overall service curve. Specially, the delay bound derived in this way is tighter than that by simply adding up delay bound at each traversed node, which is known as the pay-burst-only-once property.

**Pay-burst-only-once in NC!** Assume that a flow with arrival curve $\alpha^U$ traverses $m$ nodes $S_1, S_2, ..., S_m$ in sequence, each offering a minimum service curve $\beta_i, i = 1, 2, ...m$ to the flow respectively. Then

$$\begin{aligned} & h(\alpha^U, \beta_1 \otimes \beta_2 ... \otimes \beta_m) \\ \leq & h(\alpha_0^{*U}, \beta_1) + h(\alpha_1^{*U}, \beta_2) + ... + h(\alpha_{m-1}^{*U}, \beta_m) \end{aligned}$$

where $\alpha_i^{*U} = \alpha_{i-1}^{*U} \oslash \beta_i$ is the output arrival curve of the $i-th$ node and the input arrival curve of the $(i+1)-th$ node, and $\alpha_0^{*U} = \alpha^U$. The proof of the pay-burst-only-once property can be found in [1].

### B. Basics in Real-Time Calculus

RTC uses arrival curves and service curves to describe timing properties of event streams and available resource[1].

**Definition 5** (Arrival Curve in RTC)**.** *Let $R(s, t]$ denote the total number of arrived events in time interval $(s, t]$ [2], where $s$ and $t$ are two arbitrary non-negative real numbers. Then, the corresponding upper and lower arrival curves are denoted as $\alpha^u$ and $\alpha^l$, respectively, and satisfy:*

$$\forall s < t, \quad \alpha^l(t - s) \leq R(s, t] \leq \alpha^u(t - s),$$

*where $\alpha^u(0) = \alpha^l(0) = 0$.*

**Definition 6** (Service Curve)**.** *Let $C(s, t]$ denote the total available resource in time interval $(s, t]$. Then, the corresponding upper and lower service curves are denoted as $\beta^u$ and $\beta^l$, respectively, and satisfy:*

$$\forall s < t, \quad \beta^l(t - s) \leq C(s, t] \leq \beta^u(t - s),$$

*where $\beta^u(0) = \beta^l(0) = 0$.*

[1] We assume that the concept of event, event stream and component (to be introduced) are equivalent to that of bit, flow, and node respectively.

[2] When $s = 0$, $R(s, t] = R(0, t] = R(t)$

In this paper, we focus on the most widely used abstract component in RTC called Greedy Processing Component (GPC). A GPC processes events from the input event stream in a greedy fashion, as long as it complies with the availability of resources. The output event stream produced by GPC is described by arrival curve $\alpha^{*u}, \alpha^{*l}$

$$\begin{aligned} \alpha^{*u} &= \min((\alpha^u \otimes \beta^u) \oslash \beta^l, \beta^u) \\ \alpha^{*l} &= \min((\alpha^l \oslash \beta^u) \otimes \beta^l, \beta^l) \end{aligned}$$

The maximum delay $d_{max}$ experienced by any event on the event stream is bounded by

$$d_{max} \leq h(\alpha^u, \beta^l) = \sup_{\lambda \geq 0} \left\{ \inf\{\delta \geq 0 : \alpha^u(\lambda) \leq \beta^l(\lambda + \delta)\} \right\}$$

where $\otimes$, $\oslash$, $h(f, g)$ is introduced in last subsection.

**Pay-burst-only-once in RTC?** Assume that an event stream with upper arrival curve $\alpha^u$ traverses $m$ GPC components with lower service curve $\beta_i^l, i = 1, 2, ...m$, then

$$\begin{aligned} & h(\alpha^u, \beta_1^l \otimes \beta_2^l ... \otimes \beta_m^l) \\ \overset{?}{\leq} & h(\alpha_0^{*u}, \beta_1^l) + h(\alpha_1^{*u}, \beta_2^l) + ... + h(\alpha_{m-1}^{*u}, \beta_m^l) \end{aligned}$$

where $\alpha_i^{*u} = \alpha_{i-1}^{*u} \oslash \beta_i^l$ is the output arrival curve of the $i-th$ component and the input arrival curve of the $(i+1)-th$ component, and $\alpha_0^{*u} = \alpha^u$.

Note that although the pay-burst-only-once property has already been widely used in RTC for tighter delay bound, it has not been formally proved yet. Even worse, the concatenation property, which is the basis of the pay-burst-only-once property in NC, has been proved not to hold in RTC. As a consequence, the correctness of applying the pay-burst-only-once property in RTC is not clear yet.

## IV. Comparing curves in NC and RTC

Before stepping into the analysis of the pay-burst-only-once property, we first explore the relations among curves in NC and RTC, which are the basis of further analysis and summarized in Table II.

| Curve A | Curve B | Relation |
|---|---|---|
| arrival curve in NC | upper arrival curve in RTC | $A \iff B$ |
| lower arrival curve in NC | lower arrival curve in RTC | $A \iff B$ |
| strict service curve in NC | minimum service curve in NC | $A \implies B$ |
| lower service curve in RTC | strict service curve in NC | $A \implies B$ |
| upper service curve in RTC | maximum service curve in NC | $A \implies B$ |

TABLE II
THE SUMMARY OF RELATIONS AMONG CURVES IN RTC AND NC

### A. Comparison of arrival curves

Based on the definitions in Section III, the upper arrival curve in RTC is equivalent to the arrival curve in NC. As a counterpart, we adopt the definition of lower arrival curve in NC from [21], which lower bounds the number of arrived bits in NC and is equivalent to lower arrival curve in RTC by definition.

**Definition 7** (Lower Arrival Curve in NC)**.** *A lower arrival curve $\alpha^L$ describes the minimum number of arrived bits on a flow with input function $R(t)$ in any time interval iff for all $0 \le s \le t$, there holds*

$$R(t) - R(s) \ge \alpha^L(t-s),$$

*which is equivalent to say:*

$$R(t) \ge R\bar{\otimes}\alpha^L,$$

*where*$(f\bar{\otimes}g)(\Delta) = \sup_{0\le\lambda\le\Delta}\{f(\Delta-\lambda)+g(\lambda)\}$.

### *B. Comparison of service curves*

In both NC and RTC, the amount of available resource is described with (different types of) service curves. However, what the definitions in these two frameworks emphasize is different. In NC, service curve expresses more about how much resource is used by some input, that is, how much service is provided, while in RTC, it addresses the capability of the system, no matter whether there is an input or how much traffic the input brings. That is why the definitions of service curves in NC always involve an input. In this section we first explore the relations between strict service curve and minimum service curve in NC, and then the relations between lower/upper service curve in RTC and strict/maximum service curve in NC[3].

**Theorem 1.** *If a node offers $\beta_{strict}$ as a strict service curve to a flow with input function $R(t)$ and output function $R^*(t)$, and then it also offers $\beta_{strict}$ as a minimum service curve to the flow.*

*Proof.* For any time $t \ge 0$, consider the output $R^*(t)$. There are two cases: $t$ is either in a backlogged period or not.

If $t$ is in a backlogged period, let $s_0$ denote the start time of the backlogged period, implying all arrivals that have arrived before time $s_0$ have been served by $s_0$ and hence $R^*(s_0) = R(s_0)$. We then have:

$$\begin{aligned} R^*(t) &= R^*(s_0) + R^*(t) - R^*(s_0) \\ &= R(s_0) + R^*(t) - R^*(s_0) \\ &\ge R(s_0) + \beta_{strict}(t-s_0) \ge (R \otimes \beta_{strict})(t) \end{aligned}$$

If $t$ is not in any backlogged period[4], let $t_0$ denote the finish time of the latest backlogged period before $t$. Then $R(t_0) = R(t)$ and $R^*(t_0) = R^*(t)$, since otherwise we would not have had $t_0$ being the finish time of that backlogged period. In addition, the definition of $t_0$ also means that all arrivals that have arrived before time $t_0$ have been served by $t_0$, and hence $R^*(t_0) = R(t_0)$. Combining these, we consequently have $R^*(t) = R(t)$. Finally in this case, together with $\beta_{strict} = 0$, we obtain

$$R\otimes\beta_{strict} = \inf_{0\le s\le t}\{R(t)-R(s)+\beta_{strict}(s)\} \le R(t) = R^*(t)$$

[3]Note that in the following analysis, we assume that the start time of the system is 0, which is consistent with the definitions of arrival and service curves in both RTC and NC.

[4]This case is not considered in [9].

Then the theorem is proved.

□

**Corollary 1.** *Let $\beta^{sup}$ denote the supremum among all minimum service curves offered by a node in NC, then $\beta_{strict} \le \beta^{sup}$.*

Next we show the relation between lower service curve in RTC and strict service curve in NC.

**Theorem 2.** *If a node offers a lower service curve $\beta^l$ to a flow with input function $R(t)$ and output function $R^*(t)$, then it offers strict service curve $\beta^l$, also minimum service curve $\beta^l$ to the flow.*

*Proof.* Consider any backlogged period from $s$ to $t$. In this period, the node will be busy serving, implying no capacity remaining in the period after serving the input. Hence, $R^*(0,t] - R^*(0,s] = C(0,t] - C(0,s]$. Since the node offers lower service curve, i.e. $C(0,t] - C(0,s] \ge \beta^l(t-s)$, we hence have

$$R^*(t) - R^*(s) = R^*(0,t] - R^*(0,s] \ge \beta^l$$

This proves that the node offers a strict service curve $\beta^l$. In addition, since strict service curve implies minimum service curve in NC, $\beta^l$ is a minimum service curve to the flow. □

**Corollary 2.** *Let $\beta^{sup}_{strict}$ denote the supremum among all strict service curves offered by a node in NC, then $\beta^l \le \beta^{sup}_{strict}$.*

**Theorem 3.** *If a node offers an upper service curve $\beta^u$ to a flow with input function $R(t)$ and output function $R^*(t)$, then it offers a maximum service curve $\beta^u$ to the flow.*

*Proof.* For any time $t \ge 0$, consider the output $R^*(0,t]$. For any $0 \le s \le t$, by the definition of upper service curve together with the fact $R^*(0,s] \le R(0,s]$, there holds: $\forall 0 \le s \le t$,

$$\begin{aligned} R^*(t) = R^*(0,t] &= R^*(0,s] + R^*(0,t] - R^*(0,s] \\ &\le R(0,s] + R^*(0,t] - R^*(0,s] \\ &\le R(s) + \beta^u(t-s) \end{aligned}$$

Hence

$$R^*(t) \le \inf_{0\le s\le t}\{R(s) + \beta^u(t-s)\} = R \otimes \beta^u(t)$$

□

**Corollary 3.** *Let $\gamma^{inf}$ denote the infimum among all maximum service curves offered by a node in NC, then $\beta^u \ge \gamma^{inf}$.*

## V. Pay-Burst-Only-Once in RTC

The pay-burst-only-once property has been assumed to be true in RTC and much work has already applied it for deriving tighter delay bound. However, none of these work directly proves the correctness of applying the pay-burst-only-once property in RTC, but simply refers to the proof in NC. This direct mapping seems not so convincing since the concatenation property, based on which the pay-burst-only-once property is proved in NC, does not hold in RTC.

**Theorem 4.** *If $\beta_1^l$, $\beta_2^l$ are lower service curves offered by two nodes $S_1, S_2$ respectively, $\beta_1^l \otimes \beta_2^l$ is not necessarily a lower service curve offered by the concatenation of these two nodes.*

*Proof.* Based on Theorem 2, if a node $S_i$ offers $\beta_i^l$ as a lower service curve, then it offers $\beta_i^l$ as a strict service curve. Since it has been proved in [9] that the concatenation of strict service curves does not necessarily generate a strict service curve, the concatenation of lower service curve is not necessarily a lower service curve, so the concatenation property does not hold for lower service curve in RTC. Then the theorem is proved. □

However, although the concatenation property does not hold in RTC, the pay-burst-only-once property can be proved in RTC. Next we first explain the basic idea of the proof and then give the formal deduction.

According to Section III, the delay bound a flow experiences when it traverses a concatenated system in RTC is calculated with a similar method as in NC and the difference lies in the service curve involved: in RTC lower service curve is adopted while in NC it is minimum service curve. To prove the pay-burst-only-once property in RTC, we first calculate the delay bound in RTC and NC respectively. Then by utilizing the numerical relations among curves explored in Section IV, we compare the derived delay bounds and then prove the property.

**Theorem 5.** *The pay-burst-only-once property holds in RTC.*

*Proof.* The proof of the theorem involves two aspects:

(1) Safety. The delay bound derived with convolved lower service curve is no smaller than maximum delay experienced by the flow.

(2) Tightness. The delay bound derived with convolved lower service curve is no larger than summing up the delay bound at each node.

Assume that a flow with upper arrival curve $\alpha^u$ in RTC, arrival curve $\alpha^U$ in NC traverses a sequence of $m$ nodes, each offering a lower service curve $\beta_i^l, i = 1, 2, ...m$, the supremum of strict service curves of node $S_i$ is denoted by $\beta_{strict,i}^{sup}$, and supremum of minimum service curves of node $S_i$ is denoted by $\beta_i^{sup}$.

We first prove (1). According to Section III, the maximum delay experienced $d_{max}$ satisfies

$$d_{max} \leq h(\alpha^U, \beta_1^{sup} \otimes \beta_2^{sup} ... \otimes \beta_m^{sup})$$

Since arrival curve in NC is equivalent to upper arrival curve in RTC, we have $\alpha^u = \alpha^U$.

According to Section IV, $\beta_i^l \leq \beta_{strict,i}^{sup} \leq \beta_i^{sup}$, then we have

$$\begin{aligned}&\beta_1^l \otimes \beta_2^l ... \otimes \beta_m^l \\ \leq &\beta_{strict,1}^{sup} \otimes \beta_{strict,2}^{sup} ... \otimes \beta_{strict,m}^{sup} \leq \beta_1^{sup} \otimes \beta_2^{sup} ... \otimes \beta_m^{sup}\end{aligned}$$

Then

$$h(\alpha^u, \beta_1^l \otimes \beta_2^l ... \otimes \beta_m^l) \geq h(\alpha^U, \beta_1^{sup} \otimes \beta_2^{sup} ... \otimes \beta_m^{sup})$$

So

$$d_{max} \leq h(\alpha^u, \beta_1^l \otimes \beta_2^l ... \otimes \beta_m^l)$$

showing that the delay bound calculated with concatenated lower service curve in RTC is safe.

Next we prove (2). Since $\beta_i^l \leq \beta_{strict,i}^{sup} \leq \beta_i^{sup}$, there must exist a minimum service curve $\beta_i' = \beta_i^l$ for each node $S_i$. According to the pay-burst-only-once property in NC, we have

$$\begin{aligned}&h(\alpha^U, \beta_1' \otimes \beta_2' ... \otimes \beta_m') \\ \leq &h(\alpha_0^{*U}, \beta_1') + h(\alpha_1^{*U}, \beta_2') + ... + h(\alpha_{m-1}^{*U}, \beta_m')\end{aligned}$$

Then by replacing $\beta_i'$ with $\beta_i^l$, and replacing $\alpha^U$ with $\alpha^u$, we have

$$\begin{aligned}&h(\alpha^u, \beta_1^l \otimes \beta_2^l ... \otimes \beta_m^l) \\ \leq &h(\alpha_0^{*u}, \beta_1^l) + h(\alpha_1^{*u}, \beta_2^l) + ... + h(\alpha_{m-1}^{*u}, \beta_m^l)\end{aligned}$$

Then (2) is proved.

□

## VI. Computation of Output Arrival Curves

After verifying the correctness of the pay-burst-only-once property in RTC, we move on to compute the output arrival curves when concatenating the nodes a flow traverses. Similar as the proof of Theorem 5, we first explore how to calculate output arrival curves in NC, then prove that the corresponding results are applicable in RTC.

Since the derivation of (upper) output arrival curve is known in NC, we focus on how to calculate the lower output arrival curve in NC.

**Theorem 6.** *Consider a node offers a minimum service curve $\beta$ and maximum service curve $\gamma$ to the input flow $R(t)$ with arrival curve $\alpha^U$ and lower arrival curve $\alpha^L$ in NC. Then output flow has a lower output arrival curve $\alpha^{*L}$ with*

$$\alpha^{*L} = [(\alpha^L \otimes \beta) \wedge (\beta \bar{\oslash} \alpha^U)]^+$$

*, where $f \bar{\oslash} g(\Delta) = \inf_{\lambda \geq 0}\{f(\Delta + \lambda) - g(\lambda)\}$.*

*Proof.* Since the output in any time interval can not be larger than the input, we have $R^*(t) \leq R(t)$. Based on the definitions of minimum service curve and maximum service curve, for any time $t$ we have $(R \otimes \beta)(t) \leq R^*(t) \leq (R \otimes \gamma)(t)$. Based on the definitions of upper and lower arrival curve, we have $(R \bar{\otimes} \alpha^L)(t) \leq R(t) \leq (R \otimes \alpha^U)(t)$.

Then

$$\begin{aligned}&R^*(t-s) \\ &= R^*(t) - R^*(s) \\ &\geq (R \otimes \beta)(t) - (R \otimes \gamma)(s) \\ &= \inf_{0 \leq u \leq t}[R(t-u) + \beta(u)] - \inf_{0 \leq v \leq s}[R(s-v) + \gamma(v)] \\ &= \inf_{0 \leq u \leq t} \sup_{0 \leq v \leq s} [R(t-u) - R(s-v) + \beta(u) - \gamma(v)] \\ &= \sup_{0 \leq v \leq s} \inf_{0 \leq u \leq t} [R(t-u) - R(s-v) + \beta(u) - \gamma(v)] \\ &= \sup_{0 \leq v \leq s} [\inf_{0 \leq u \leq t-s+v} [R(t-u) - R(s-v) + \beta(u) - \gamma(v)] \\ &\wedge \inf_{t-s+v \leq u \leq t} [R(t-u) - R(s-v) + \beta(u) - \gamma(v)]]\end{aligned}$$

$$= \sup_{0\le v\le s} \inf_{0\le u\le t-s+v}[R(t-u)-R(s-v)+\beta(u)-\gamma(v)]$$
$$\wedge \sup_{0\le v\le s} \inf_{t-s+v\le u\le t}[R(t-u)-R(s-v)+\beta(u)-\gamma(v)] \quad (1)$$
$$\ge \sup_{0\le v\le s} \inf_{0\le u\le t-s+v}[\alpha^L(t-u-s+v)+\beta(u)-\gamma(v)]$$
$$\wedge \inf_{t-s\le u\le t}[R(t-u)-R(s)+\beta(u)] \quad (2)$$
$$= \sup_{0\le v\le s} \inf_{0\le u\le t-s+v}[\alpha^L(t-u-s+v)+\beta(u)-\gamma(v)]$$
$$\wedge \inf_{t-s\le u\le t}[\beta(u)-(R(s)-R(t-u))]$$
$$= \sup_{0\le v\le s}[(\alpha^L\otimes\beta)(t-s+v)-\gamma(v)]$$
$$\wedge \inf_{0\le u'\le s}[R(u')-R(s)+\beta(t-u')] \quad (3)$$
$$\ge (\alpha^L\otimes\beta)(t-s)\wedge \inf_{0\le u'\le s}[\beta(t-u')-\alpha^U(s-u')] \quad (4)$$
$$= (\alpha^L\otimes\beta)(t-s)\wedge \inf_{0\le u'\le s}[\beta(t-s+s-u')-\alpha^U(s-u')]$$
$$= (\alpha^L\otimes\beta)(t-s)\wedge \inf_{0\le u''\le s}[\beta(t-s+u'')-\alpha^U(u'')]$$
$$\ge (\alpha^L\otimes\beta)(t-s)\wedge \inf_{0\le u''}[\beta(t-s+u'')-\alpha^U(u'')] \quad (5)$$
$$= (\alpha^L\otimes\beta)\wedge(\beta\bar{\oslash}\alpha^U) \quad (6)$$

where $\wedge$ gets the minimum of the two parts it connects; in obtaining (2) we take $v=0$ of the second part of (1); (4) is to take $v=0$ in the first item of (3) and apply upper arrival curve to $R(s)-R(u')$ in the second half of (3); (5) is because of taking larger range for infimum operation.

Based on the definition of lower arrival curve, for any interval $\Delta$, $\alpha^{*L}(\Delta)\ge 0$, then we have

$$\alpha^{*L} = [(\alpha^L\otimes\beta)\wedge(\beta\bar{\oslash}\alpha^U)]^+$$

Then we know how to calculate the output arrival curves after concatenation in RTC.

**Theorem 7.** *Assume an event stream with arrival curve* $\alpha^u, \alpha^l$ *traverses* $m$ *GPC components with service curves* $(\beta_1^u, \beta_1^l), (\beta_2^u, \beta_2^l), ..., (\beta_m^u, \beta_m^l)$*, then the output arrival curve at* $m-th$ *component is upper and lower bounded by:*

$$\alpha^{*u} = (\alpha^u\otimes\beta^u)\oslash\beta^l$$
$$\alpha^{*l} = [(\alpha^l\otimes\beta^l)\wedge(\beta^l\bar{\oslash}\alpha^u)]^+$$

*Proof.* The proof is similar to that of Theorem 5 with the combination of Theorem 1, 2 and 3. □

□

## VII. CONCLUSION

In this work, we for the first time prove the pay-burst-only-once property holds in RTC, which benefits the calculation of delay bound of networked systems. We also explore how to calculate the output arrival curves of a concatenated system in a once-for-all way, rather than calculate it one node by another.

For future work, we will explore the application of other properties in RTC and try to enrich the framework of RTC.

## VIII. ACKNOWLEDGMENT

This work is supported by the Research Grants Council of Hong Kong (GRF 15204917 and 15213818), National Natural Science Foundation of China (No.61532007 and 61672140) and the Ministry of Education Joint Foundation for Equipment Pre-Research under grant 6141A020333.

## REFERENCES

[1] J. L. Boudec and P. Thiran, "Network calculus - a theory of deterministic queuing systems for the internet." Springer Verlag, 2012.
[2] Y. Liu and Y. Jiang, "Stochastic network calculus." Springer Verlag, 2008.
[3] F. Geyer and G. Carle, "Network engineering for real-time networks: comparison of automotive and aeronautic industries approaches," *IEEE Communications Magazine*, vol. 54, no. 2, pp. 106–112, 2016.
[4] L. Thiele, S. Chakraborty, and M. Naedele, "Real-time calculus for scheduling hard real-time systems." in *ISCAS*, 2000.
[5] S. Chakraborty, L. T. X. Phan, and P. S. Thiagarajan, "Event count automata: a state-based model for stream processing systems," in *RTSS*, 2005.
[6] E. Fersman, L. Mokrushin, P. Pettersson, and W. Yi, "Schedulability analysis of fixed-priority systems using timed automata," *Theoretical Computer Science - Tools and algorithms for the construction and analysis of systems*, vol. 354, no. 2, pp. 301–317, 2006.
[7] L. T. X. Phan, S. Chakraborty, P. S. Thiagarajan, and L. Thiele, "Composing functional and state-based performance models for analyzing heterogeneous real-time systems," in *RTSS*, 2007.
[8] K. Lampka, S. Perathoner, and L. Thiele, "Analytic real-time analysis and timed automata: a hybrid methodology for the performance analysis of embedded real-time systems," *Design Automation for Embedded Systems*, vol. 14, no. 3, pp. 193–227, 2010.
[9] A. Bouillard, L. Jouhet, and E. Thierry, "Service curves in network calculus: dos and donts." Research Report, 2009.
[10] E. Wandeler, "Modular performance analysis and interface-based design for embedded real-time systems," in *Ph.D. Thesis, Swiss federal institute of technology Zurich*, 2006.
[11] K. Lampka, S. Bondorf, and J. B. Schmitt, "Achieving efficiency without sacrificing model accuracy: Network calculus on compact domains," in *MASCOTS*, 2016.
[12] A. Bouillard, L. Jouhet, and E. Thierry, "Tight performance bounds in the worst-case analysis of feed-forward networks," in *INFOCOM*, 2010.
[13] S. Bondorf, P. Nikolaus, and J. B. Schmitt, "Quality and cost of deterministic network calculus: design and evaluation of an accurate and fast analysis," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 1, no. 1, pp. 15:1–15:34, 2017.
[14] A. Bouillard, L. Jouhet, and . Thierry, "Comparison of different classes of service curves in network calculus," *IFAC Proceedings Volume*, vol. 43, no. 12, pp. 306–311, 2010.
[15] A. Bouillard, "Composition of service curves in network calculus," *Proceedings of the 1st International Workshop on Worst-Case Traversal Time*, pp. 35–42, 2011.
[16] J. B. Schmitt, F. A. Zdarsky, and I. Martinovic, "Improving performance bounds in feed-forward networks by paying multiplexing only once," in *MMB*, 2008.
[17] S. Bondorf and J. B. Schmitt, "Boosting sensor network calculus by thoroughly bounding cross-traffic," in *INFOCOM*, 2015.
[18] J. B. Schmitt, F. A. Zdarsky, and M. Fidler, "Delay bounds under arbitrary multiplexing: When network calculus leaves you in the lurch..." in *INFOCOM*, 2008.
[19] K. Lampka, S. Bondorf, J. B. Schmitt, N. Guan, and W. Yi, "Generalized finitary real-time calculus," in *INFOCOM*, 2017.
[20] G. Chen, K. Huang, C. Buckl, and A. Knoll, "Applying pay-burst-only-once principle for periodic power management in hard real-time pipelined multiprocessor systems," *ACM Trans. Design Autom. Electr. Syst.*, vol. 20, no. 2, pp. 26:1–26:27, 2015.
[21] H. Schioler, J. J. Jessen, J. D. Nielsen, and K. G. Larsen, "Network calculus for real time analysis of embedded systems with cyclic task dependencies," in *CATA*, 2005.