Achieving Efficiency Without Sacrificing Model Accuracy: Network Calculus on Compact Domains

Kai Lampka
Uppsala University
Email: kai.lampka@it.uu.se

Steffen Bondorf
TU Kaiserslautern, Germany
Email: bondorf@cs.uni-kl.de

Jens Schmitt
TU Kaiserslautern, Germany
Email: jschmitt@cs.uni-kl.de

***Abstract*—Messages traversing a network commonly experience waiting times due to sharing the forwarding resources. During those times, the crossed systems must provide sufficient buffer space for queueing messages. Network Calculus (NC) is a mathematical methodology for bounding flow delays and system buffer requirements. The accuracy of these performance bounds depends mainly on two factors: the principles manifesting in the NC flow equation and the functions describing the system. We focus on the latter aspect. Common implementations of NC overapproximate these functions in order to keep the analysis computationally feasible. However, overapproximation often results in a loss of accuracy of the performance bounds. In this paper, we make such compromising tradeoffs between model accuracy and computational effort obsolete. We limit the accurate system description to functions of a compact domain, such that the accuracy of the NC analysis is preserved. Tying the domain bound to the algebraic operators of NC instead of the operational semantics of components, allows us to directly apply our solution to algebraic NC analyses that implement principles such as pay burst only once and pay multiplexing only once.**

I. Introduction

A. Motivation

Verifying timing correctness is a key issue for distributed real-time systems. Ensuring impeccable runtime behaviour is attained with precise analysis in the design phase of such a real-time system. For safety-critical systems, formal verification of delay requirements is even a certification prerequisite.

In addition to mathematical correctness, real-time analysis methods need to be efficient and accurate, because:

1) The analysis is often part of an automated design space exploration where a large set of system configurations is evaluated. Analysing alternatives must often be done in at most a few seconds in order to keep the computational effort of design space exploration feasible.
2) The computed performance measures should be sufficiently accurate to prevent an expensive overprovisioning of hardware. Furthermore, inaccuracies may also lead to wrong assessment of different design alternatives as these may be affected non-uniformly.

Throughout the past decades different analysis methodologies have been developed. These methodologies vary in expressiveness, objective and in their computational costs.

Analytic methodologies like Network Calculus (NC) are based on sound mathematical theories and have the potential to scale well to the size and complexity of modern distributed real-time systems. Correctness and accuracy of the NC performance bounds depend (a) on the sound and accurate derivation of the flow equations of NC and (b) on the precision of the bounding functions provided as input to the analysis.

Soundness and accuracy of flow equations. To achieve soundness, flow equations of NC have to subsume the operational behaviour of the modelled network entities. E. g., a complex server element can be modelled by flow equations which do not make any assumptions about the processing order of packets. We call this arbitrary multiplexing. It constitutes a conservative model that guarantees for invariably valid bounds for delay and backlog for any scheduling behaviour.

The closer the derivations of NC capture the worst-case behaviour occurring during system operation, the more accurate NC performance bounds are. Principles that enable accurately capturing system behaviour are Pay Burst Only Once (PBOO) and Pay Multiplexing Only Once (PMOO). PBOO eliminates repeatedly considering the analysed flow's burst term and PMOO extends this feature to cross-flows. These principles allow to outperform a component-wise analysis where the local worst-case delays at each server are summed up.

Precision of input functions. As long as input functions provide upper bounds on the assumed workload and lower bounds on the available service, conservativeness of delay and backlog bounds is guaranteed. In consequence, functions with complex shapes can be replaced by piecewise linear functions with a moderate number of segments. This comes with the benefit of simplifying the mathematical operations of NC, but it increases delay and backlog bounds, i. e., it is compromising on the accuracy of NC-based system analysis.

This paper addresses both of the above issues by proposing to use NC with so-called *compact domains*, i. e., restricting functions to a bounded domain $[0, K]$ instead of $[0, +\infty)$.

- The restriction to compact domains significantly improves the computational efficiency of implementations of NC. It enables to use complex functions to accurately describe system behaviour in the time scale relevant to the analysis. The improved precision of input functions on their relevant parts eventually yields more accurate performance bounds while the suppression of the tail of the domain results in reduced computational effort.
- We relate the use of compact domains directly to the operators of the (min,+)- and (max,+)-algebra. Thereby, compact domains can be directly exploited in known

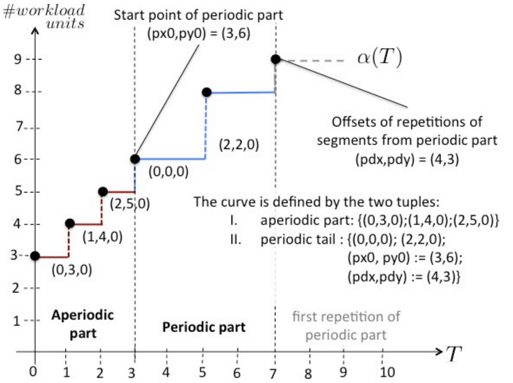


Fig. 1. Curve representation (within the MPA-toolbox)

flow equations of NC: the prevalent PBOO and PMOO principles as well as very recent improvements [1], [2].

These advantages are key for universal improvements in the computational efficiency of NC without sacrificing the accuracy of the system model or flow equations.

B. Technical Problem

NC is based on the evaluation of flow equations. These are formulae of (min,+)- and (max,+)-algebra applied to functions bounding data arrivals and forwarding service. Piecewise linear, pseudo-periodic functions are commonly used for this purpose. An example is illustrated in Fig. 1. In the following, we call the arrival and service functions of NC *curves* to emphasize their restriction to this shape.

Applying basic operations of NC on pseudo-periodic input curves yields output curves whose periods equal the hyperperiod of the input curves. The exact shape of the result must be computed on the least common multiple of the input curves' periods [3]. A network analysis spreads over multiple nodes, each requiring a sequence of (min,+)- and (max,+)-based curve transformations. Solving the resulting flow equation can quickly become computationally expensive. It is possible to avoid the above scaling problem by simplifying the tail descriptions of the curves, i. e., restricting the precise description of curves to a compact domain. However, this requires to determine the parts of the functions which are relevant for computing the delay and backlog bounds beforehand, such that their inclusion within chosen compact domains is guaranteed. I.e., on the one hand, compact domains need to be sufficiently large such that performance bounds are not negatively affected. On the other hand, computational effort decreases with reduction of the compact domains of curves. We achieve balance of both aspects by analysing the NC flow equation of a system with the following scheme: We derive a bound on the domain length ensuring that delay and backlog bounds are enclosed. This initial domain bound is then used to derive the required domain lengths of all curves.

The remainder of the paper is structured as follows: Section II presents the related work and Section III provides the required NC background. The detailed mathematical treatment of sufficiently large compact domains is given in Section IV before we conclude the paper in Section Section V.

II. Related Work

Network calculus has found application in different areas; not only in networking [4], [5], but also in the domain of embedded real-time systems [6], [7]. Most NC tools implement bounding curves as piecewise linear, pseudo-periodic functions [3], [7], [8]. Closure of this class is shown in [3], and it is shown that NC operators suffer from the hyperperiod problem.

Reducing the domain of curves to finite intervals as part of NC-based system analysis has been introduced and empirically evaluated in the work of Perathoner et al. [9], [10]. It is, however, preliminary, as it lacks proofs for the preservation of accuracy of performance bounds. Guan and Wang [11] derived conditions for the lengths of these intervals and proved that these lengths preserve accuracy of the performance bounds. Yet, these proofs only hold for the input/output relation (operational semantics) of the greedy-processing component (GPC) commonly used in Real-Time Calculus (RTC). Although NC and RTC are based on the same dioid algebra, the component-level of abstraction in the proofs imposes crucial limitations for application in NC. As we depict below (Fig. 2(A) to Fig. 2(C)), NC analyses implementing the PBOO and PMOO principles do not preserve the component-structure of the network. The only analysis that does so and therefore can apply the work of [11] is the TFA (Total Flow Analysis), but TFA is known to derive inferior delay bounds (e. g., see Fig. 2(D)(ii)).

In this paper, we explicitly depart from the component-wise view. Instead, we focus on the (min,+)- and (max,+)-operators of the flow equations. This idea allows compact domains in analyses exploiting prevailing principles PBOO, PMOO, or [12], and future principles based on the standard operators.

III. Theoretical Background

A. The Network Calculus System Description

Network calculus is built around the following system description [13]. Flows are characterised by curves cumulatively counting their data. They belong to the set $\mathcal{F}_0$ of non-negative, wide-sense increasing functions:

$$\mathcal{F}_0 = \left\{ f : \mathbb{R}_\infty^+ \to \mathbb{R}_\infty^+ \mid f(0) = 0,\ \forall s \le t : f(s) \le f(t) \right\},$$
$$\mathbb{R}_\infty^+ := [0, +\infty) \cup \{+\infty\}.$$

We are particularly interested in $A(t)$ and $A'(t)$ that cumulatively count a flow's data put into a server s and put out from s, both up until time t.

Definition 1: (Backlog and Delay) Assume a flow with input A which crosses a server s, resulting in the output A'. The *backlog* of the flow at time t is defined as $B(t) = A(t) - A'(t)$. The *(virtual) delay* at time t is defined as $D(t) = \inf \{ \tau \ge 0 \mid A(t) \le A'(t + \tau) \}$.

NC bounds these arrivals in the interval time domain:

Definition 2: (Arrival Curve) Given a flow with input function A, a function $\alpha \in \mathcal{F}_0$ is an arrival curve for A iff

$$\forall\, 0 \le d \le t : A(t) - A(t - d) \le \alpha(d).$$

Flow transformations leading to the output function $A'(t)$ depend on a server's forwarding capabilities:

Definition 3: (Service Curve) If the service provided by a server s for a given input function A results in an output function A', we say that s offers a service curve $\beta \in \mathcal{F}_0$ iff

$$A'(t) \geq \inf_{0 \leq d \leq t} \{A(t-d) + \beta(d)\}.$$

A number of servers fulfill a stricter definition of service curves that guarantees a higher output during periods of queued data, the so-called backlogged periods of a server.

Definition 4: (Strict Service Curve) Let $\beta \in \mathcal{F}_0$. Server s offers a *strict* service curve β to a flow iff, during any backlogged period of duration d, the output of the flow is at least equal to $\beta(d)$.

B. Algebraic Network Calculus Operations

NC was cast in a (min,+)/(max,+)-algebraic framework in [4], [14]. The following operations allow to manipulate arrival and service curves in flow equations.

Definition 5: ((min,+)- and (max,+)-Operations) The algebraic NC operations to derive an output curve from two given input curves $f, g \in \mathcal{F}_0$ are:

$$\begin{aligned}
\text{aggregation:}\ (f+g)(t) &= f(t) + g(t),\\
\text{separation:}\ (f-g)(t) &= f(t) - g(t),\\
\text{(min,+)-convolution:}\ (f \otimes g)(t) &= \inf_{0 \leq s \leq t} \{f(t-s) + g(s)\},\\
\text{(max,+)-convolution:}\ (f \,\overline{\otimes}\, g)(t) &= \sup_{0 \leq s \leq t} \{f(t-s) + g(s)\},\\
\text{(min,+)-deconvolution:}\ (f \oslash g)(t) &= \sup_{u \geq 0} \{f(t+u) - g(u)\}.
\end{aligned}$$

Applied to pseudo-periodic functions $f, g \in \mathcal{F}_0$ with periods p_f and p_g where $\frac{p_f}{p_g}$ is rational, the operations $+$, $-$, $\otimes$, $\overline{\otimes}$ as well as $\min$ and $\max$ result in a curve with p equal to their hyperperiod $\mathrm{lcm}(p_f, p_g)$.

Theorem 1: (Performance Bounds) Consider a server s that offers a service curve β and a flow with arrival curve α traversing the server. Then we obtain the following bounds:

$$\begin{aligned}
\text{delay:}\ \forall t \in \mathbb{R}^+_\infty &: D(t) \leq \sup_{u \geq 0}\{\inf\{s \geq 0 \colon \alpha(u) \leq \beta(u+s)\}\},\\
\text{backlog:}\ \forall t \in \mathbb{R}^+_\infty &: B(t) \leq \sup_{r \geq 0}\{\alpha(r) - \beta(r)\},\\
\text{output:}\ \forall d \in \mathbb{R}^+_\infty &: \alpha'(d) = (\alpha \oslash \beta)(d),
\end{aligned}$$

where the delay and backlog bounds are abbreviated by D and B, respectively, as they hold independent of t and α' is an arrival curve for A'.

Analysing an entire flow and considering cross-traffic on its path warrants the following theorems.

Theorem 2: (Concatenation of Servers) Consider a flow that traverses a tandem of servers s_i, $i = 1, \ldots, n$. Each s_i offers a service curve β_{s_i} to the flow. Then the concatenation of the n systems offers a service curve $\bigotimes_{i=1}^{n} \beta_{s_i}$ to the flow.

Theorem 3: (Left-Over Service Curve) Consider a server s that offers a strict service curve β_s. Let s be crossed by two flows f_0 and f_1 with arrival curves α^{f_0} and α^{f_1}, respectively. f_1's left-over service curve under arbitrary multiplexing is

$$\beta_s^{\text{l.o.}f_1} := \beta_s \ominus \alpha^{f_0} = \left(\beta_s - \alpha^{f_0}\right) \overline{\otimes}\, \lambda_0, \;\; \lambda_0(t) = 0.$$

C. NC Analysis: Accuracy of Flow Equations

The main task of a NC analysis is to derive a flow equation from the system description. A flow equation is compiled to either derive the delay bound of a flow (flow of interest, foi) or the backlog bound of a component, i. e., its buffer requirement. Current NC alternatives to derive this flow equations are:

Total Flow Analysis (TFA) [13]: This analysis derives delay and backlog bounds that are valid for the totality of flows at a component. A foi's end-to-end delay is derived by summing up the component delay bounds on its path. TFA thus assumes the foi experiences the worst case at every component.

Overcoming this assumption has seen much treatment in the literature. The component-wise analysis was superseded by *tandem analyses* to bound the foi's end-to-end delay. This approach improves the ability to capture scheduling and multiplexing effects and derives more accurate delay bounds. Tandem analyses derive an end-to-end left-over service curve for the foi. Two basic principles have been established:

PBOO principle (Separate Flow Analysis, SFA) [4]: SFA applies Theorem 3 and 2: At every component, subtract cross-traffic arrivals, then concatenate the resulting left-over service curves. Computing the delay bound with this single tandem left-over service curve considers the foi's burst only once.

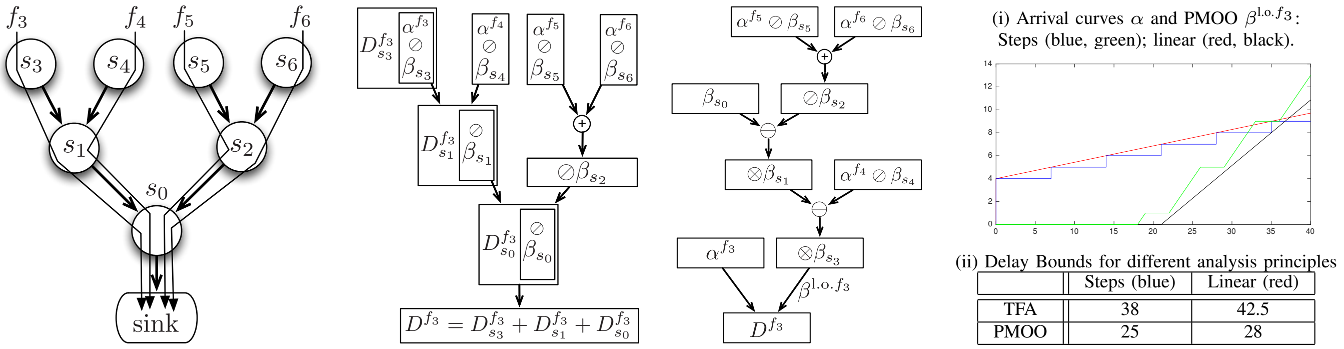
PMOO principle [15]: While the SFA does not suffer from the foi's burst multiple times, it does so w. r. t. to cross-traffic. Computing the left-over service curve component-by-component inhibits an improvement. The key for not paying multiplexing with cross-traffic more than once is to reverse the SFA's operations – a tandem of servers is convolved before subtracting cross-traffic [5], [16]–[18]. In this paper, we use the sink-tree PMOO analysis presented in [16] as it only uses the operators presented above.

Fig. 2(B) and Fig. 2(C) illustrate the flow equations for TFA and sink-tree PMOO analysis. Note that the latter does not preserve the network's component structure in its derivation.

D. Precision of Input Curves

Modelling should feature non-determinism in the arrival and service processes, since the exact points in time when a specific packet arrives or is processed are often unknown. Addressing this issue, different models have been proposed in the real-time systems literature. These range from strictly periodic arrivals, over sporadic arrivals with minimum inter-arrival distance up to periodic arrivals with jitter and minimal release distance (pjd-model), where the latter allows bursts in the packet arrivals. In particular, the pjd-model is often found in the literature [10], [19]. Such models can be directly reflected within NC, curve α provides an upper bound on the assumed data arrivals and curve β provides a lower bound on the available forwarding service. E. g., how to translate a pjd arrival model into a curve of the NC is shown in [20].

The precision in bounding data arrival and forwarding service constitutes the first step towards accurate performance bounds. Striving for precision often results in complex curves, such as staircase functions with an aperiodic prefix and a periodic tail as depicted in Fig. 1. Although these curves


f_3
f_4
f_5
f_6
s_3
s_4
s_5
s_6
s_1
s_2
s_0
sink
$D^{f_3}_{s_3}$
$\alpha^{f_3} \oslash \beta_{s_3}$
$\alpha^{f_4} \oslash \beta_{s_4}$
$\alpha^{f_5} \oslash \beta_{s_5}$
$\alpha^{f_6} \oslash \beta_{s_6}$
$D^{f_3}_{s_1}$
$\oslash \beta_{s_1}$
$+$
$\oslash \beta_{s_2}$
$D^{f_3}_{s_0}$
$\oslash \beta_{s_0}$
$D^{f_3} = D^{f_3}_{s_3} + D^{f_3}_{s_1} + D^{f_3}_{s_0}$
$\alpha^{f_5} \oslash \beta_{s_5}$
$\alpha^{f_6} \oslash \beta_{s_6}$
$+$
β_{s_0}
$\oslash \beta_{s_2}$
$\ominus$
$\otimes \beta_{s_1}$
$\alpha^{f_4} \oslash \beta_{s_4}$
$\ominus$
α^{f_3}
$\otimes \beta_{s_3}$
$\beta^{\text{l.o.}f_3}$
D^{f_3}
(i) Arrival curves α and PMOO $\beta^{\text{l.o.}f_3}$:
Steps (blue, green); linear (red, black).
0
2
4
6
8
10
12
14
0
5
10
15
20
25
30
35
40
(ii) Delay Bounds for different analysis principles

	Steps (blue)	Linear (red)
TFA	38	42.5
PMOO	25	28

(A) Sample Network (B) TFA Delay Bounding (C) PMOO Delay Bounding (D) Input curves and computed delay bounds

Fig. 2. Illustrative Example of a network converted into alternative flow equations and the impact of curve shapes on the resulting delay bounds.

have already been discussed in [4], analyses implementing the PMOO principle oftentimes cannot be applied to those curves. For instance, [5], [17], [18] rely on
(a) service curves that can be decomposed into a maximum over rate latencies $\beta_{R,T}(t) = \max\{0, R \cdot (t-T)\}$ and
(b) arrival curves that can be decomposed into a minimum over token buckets $\alpha_{r,b}(t) = r \cdot t + b$ if $t > 0$, $\alpha_{r,b}(t) = 0$ otherwise,
(c) each decomposition results in finitely many curves.
Fulfilling this analysis prerequisite requires to compromise on the input curve precision early on.

The sink-tree PMOO [16] does not share these constraints on the curves. It can be applied to any curves from $\mathcal{F}_0$. We use the sink-tree PMOO to illustrate the loss in performance bound accuracy caused by imprecise modelling. Fig. 2(A) depicts a sample network with homogeneous service $\beta(t) = t$. The bound on flow arrivals is assumed to be precisely modelled with $pjd = (7, 12, 0)$, where we approximate this arrival curve with a single linear segment, i.e., $\alpha_{\frac{1}{7},4}(t) = \frac{1}{7} \cdot t + 4$. This is the tightest possible linear overapproximation of the sample staircase function due to cutting each step's starting point. Already in this simple example, a $\sim$10% difference in end-to-end delay bounds between the second and third column in Fig. 2(D)(ii) is solely caused by the modelling inaccuracy of the arrivals that, in turn, carries over to an inaccurate $\beta^{\text{l.o.}f_3}$.

IV. NC on Compact Domains

A. Preliminaries

As has been discussed, our approach is based on the idea to reduce the domain of each curve to a closed interval.

Definition 6: Let $\mathbb{CD} := [0, K]$ be a closed interval. If a curve γ provides a mapping of the kind $\gamma : [0, K] \to \mathbb{R}^+_\infty$ we denote $\mathbb{CD}$ as its compact domain.

In this setting, we define the following notation.

- A primed Greek character, e.g., γ' refers to some curve employed in the NC analysis which provides an *unrestricted* mapping $\mathbb{R}^+_\infty \to \mathbb{R}^+_\infty$.
- A corresponding (*restricted*) curve refers to a curve which we obtain by restricting to a compact domain, i.e., $\gamma : [0, K_\gamma] \to \mathbb{R}^+_\infty$.

For unrestricted curves α' and β', we define the following approximations:

- A curve $\downarrow\!\alpha'$ is an overapproximation of an unrestricted arrival curve α' if $\forall t \in \mathbb{R}^+_\infty : \alpha'(t) \leq \downarrow\!\alpha'(t)$.
- A curve $\uparrow\!\beta'$ is an overapproximation of an unrestricted service curve β' if $\forall t \in \mathbb{R}^+_\infty : \beta'(t) \geq \uparrow\!\beta'(t)$.

With arrival curve $\downarrow\!\alpha'$, more workload than originally specified with α' is allowed. In case of a service curve $\uparrow\!\beta'$, the assumed service available to a flow is decreased when compared to the original service curve β'. Underapproximations $\uparrow\!\alpha'$ and $\downarrow\!\beta'$ for more optimistic behaviour are defined analogously.

In the following, the concrete form of approximations is irrelevant as long as the above conditions are met, e.g., one could use specific traffic patterns as proposed in [19], [20], approximations based on single linear segments [10], or minimum-composed sets of linear segments [21].

As we are emphasizing abstract network modelling, we do not rely on any assumptions about the processing order of packets, e.g., first-in-first-out. We solely demand that there is enough service available to process all inputs, i.e., we require that $\exists t.\, \forall t' \geq t.\ \alpha(t') \leq \beta(t')$ and $\frac{\alpha(t')}{t'} < \frac{\beta(t')}{t'}$. This way we restrict ourselves to resource consuming components where, in the longterm, input buffers become and remain empty.

B. Enclosure of Backlog and Delay Bounds

Deriving a performance bound, backlog or delay, constitutes the eventual step of any NC analysis; see Fig. 2(B) and Fig. 2(C). Working on compact domains, we need to guarantee that these are large enough to enclose the bounds. Otherwise, the bounds derived on the compact domain are not valid. The following corollary is a direct consequence of Theorem 1, ensuring its enclosure of the delay and backlog bound.

Corollary 1: Backlog and Delay Bound Enclosure Assume the delay and backlog bound for an unrestricted arrival curve α' and an unrestricted service curve β' to be known. I.e., we already solved these derivations and know the values of u, s and r for which the suprema in Theorem 1 are attained. Then we derive the domains of the restricted curves α and β by

$$K_\alpha = \max(u,\ r), \quad K_\beta = \max(u{+}s,\ r).$$

Corollary 1's prerequisite to know the performance bounds for α' and β' a priori is crucial. In an analysis, both are only known a posteriori, i.e., after we solved the flow equation with unrestricted curves. Yet, the analysis with unrestricted arrival and service curves imposes a high analysis effort, when being performed with accurate modelled curves, e. g., piecewise linear, pseudo-periodic ones (imposing the hyperperiod problem). Hence, this requirement is thus contradicting our aim to execute the analysis with less computational effort.

We break this circular dependency by deriving delay and backlog bounds with linearly underapproximated and overapproximated arrival and service curves. Fig. 2(D)(i) provides an example for such an overapproximation (for the underapproximation, one simply needs to shift the curves horizontally). From this step, which can be executed very efficiently if the approximations are simple curves (e. g., curves only consisting of a single linear segment), we obtain upper and lower bounds on the delay and backlog. Equipped with these bounds, we can compute valid compact domains for the α and β.

In conclusion of this idea, we propose to proceed as follows:

1) With the pair $\uparrow\alpha'$ and $\downarrow\beta'$, we compute estimated $\uparrow D$ and $\uparrow B$ for bounding the delay and backlog. As the involved curves $\uparrow\alpha'$ and $\downarrow\beta'$ are underapproximations of the unrestricted curves α' and β', the obtained estimates are not safe, but they truly provide valid lower bounds for the delay and backlog.
2) For the lower bounds $\uparrow D$ and $\uparrow B$, we compute the largest value of t, such that for the function values beyond t solely smaller delay and backlog bounds can be derived.

We formalize this compact domain for delay and backlog bounds as follows.

Theorem 4: We define the following variables:

- $U := \sup\{t \geq 0 \ : \downarrow\alpha'(t) \geq \uparrow\beta'(t + \uparrow D)\}$ and
- $R := \sup\{t \geq 0 \ : \downarrow\alpha'(t) - \uparrow\beta'(t) \geq \uparrow B\}$.

For $K_\alpha \geq \max(U, R)$ and $K_\beta \geq \max(U + \downarrow D, R)$ the domain bounds K_α and K_β are sufficiently large such that accurate delay and backlog bounds can be computed with α and β.

Proof 1: The above theorem is correct, if K_α and K_β provide bounds on the input values for which delay and backlog bounds can be found for α' and β'.

We defined the (largest) pseudo-inverse of the lower delay and backlog bound as follows:

$$\begin{aligned} u^* &:= \sup\{\mathbf{u} \geq 0 : \uparrow D \leq \\ &\qquad\qquad \sup\nolimits_{u \geq \mathbf{u}} \{\inf\{s \geq 0 \ : \ \alpha'(u) \leq \beta'(u+s)\}\}\}, \\ r^* &:= \sup\left\{\mathbf{r} \geq 0 : \uparrow B \leq \sup\nolimits_{r \geq \mathbf{r}} \{\alpha'(r) - \beta'(r)\}\right\}. \end{aligned}$$

Sub- and superadditivity of curves α and β, together with $\uparrow D \leq \downarrow D$ implies that $u^* \leq U$. Thus U is truly a safe upper bound for guaranteeing delay bound enclosure in curve α.

Exploiting $u^* \leq U$ together with the inequality $D \leq \downarrow D$ yields $u^* + D \leq U + D \leq U + \downarrow D$. Note that D is a bound on s in Theorem 1.

This implies that $U + \downarrow D$ is truly an upper bound for guaranteeing delay bound enclosure w. r. t. curve β.

With r^* being bounded by R we obtain that $K_\alpha = \max(U, R)$ and $K_\beta = \max(U + \downarrow D, R)$ guarantee that also the backlog bound can be found with α' and β' being restricted to $[0, K_\alpha]$, resp. $[0, K_\beta]$. ■

An alternative bound on the domain size is given in [11]. There, the authors propose to use the largest T such that $\forall t \geq T : \downarrow\alpha'(t) \leq \uparrow\beta'(t)$ holds. However, this leads to a significantly larger domain bound, especially when the utilization is high.

C. Ensuring Accuracy of Computed Curves

We now know how to efficiently derive a bound on the compact domains for arrival and service involved in bounding delay and backlog. Next, we want to use this insight to increase the computational efficiency of the entire analysis. We aim to derive the compact domain for each arrival and service curve. The smaller these domains, the faster the flow equation can be solved in comparison to the unrestricted curves. This is due to the fact that output curves only need to be precisely computed on the compact domain, not on the length of the hyperperiod. On the other hand, compact domains need to be sufficiently large in order to allow deriving valid performance bounds.

Computing each curve's domain requires to traverse the flow equation backwards. This means we start with Theorem 1 whose input curve domains we derived in Theorem 4. Each of this curve is either

- the output of a preceding (min,+)- or (max,+)-operation presented in Theorem 5, or
- an original arrival or service curve that models system behaviour of data arrivals entering it.

The former requires us to recursively continue backtracking domain requirements through the flow equation; the latter represents the termination condition for this backtracking. In the following, we derive the compact domains of input curves given a target domain required for the result of a NC operation.

1) Notation: Let $\odot$ be a binary NC-operator as defined in Definition 5, i. e., $\odot \in \{\otimes, \overline{\otimes}, \oslash, \min, +, -\}$, and let $\gamma(t)$ be the output curve of an operation.

When backtracking we know the bound for the compact domain of output curve γ, K_γ. Moreover, from the flow equation we know $(\alpha' \odot \beta')\,(t) = \gamma(t)$.

In the following, we derive the conditions for $(\alpha \odot \beta)(t) = \gamma(t)$, i. e., the domain length K_α and K_β such that γ remains defined on its domain K_γ. This directly implies that $\forall t \in [0, K_\gamma] \ : \ (\alpha \odot \beta)(t) = (\alpha' \odot \beta')(t)$ and that for a K_γ of sufficient size the obtained performance bounds are derived inside the compact domain, i.e., the accurate part of the curves.

2) Compact domains with operators of the NC: For $\odot \in \{\otimes, \overline{\otimes}, \min, +, -\}$, the domain sizes K_α and K_β for input curves to operation $(\alpha \odot \beta)(t) = \gamma(t)$ are given by $K_\alpha = K_\beta = K_\gamma$. This is a direct consequence from their definition: to compute $\gamma(t) = \alpha(t) \odot \beta(t)$, α and β only need to be be evaluated for values less or equal to t.

Handling of the (min,+)-deconvolution is more involved.

Theorem 5: Domain bounds for (min,+)-deconvolution $\oslash$ As before, let K_γ be the domain size of the output curve. We define U to be the largest input value for which $\downarrow\alpha(U + K_\gamma) \geq \uparrow\beta(U)$, i. e.,

$$U := \min\left(0, \sup\{t \in \mathbb{R} : \downarrow\alpha(t + K_\gamma) - \uparrow\beta(t) \geq 0\}\right)$$

This yields that for

$$K_\alpha = U + K_\gamma,\ \ K_\beta = U \text{ and}$$
$$\forall t \in [0, K_\gamma] :\ (\alpha \oslash \beta)(t) = (\alpha' \oslash \beta')(t)$$

Proof 2: For the output curve, we are only interested in function values for input values from the compact domain $[0, K_\gamma]$. By exploiting U as defined above, we obtain that

$$\forall t \leq K_\gamma : \sup_{u>U}\{\alpha'(t+u) - \beta'(u)\} < 0.$$

This shows that function values from α' and β' beyond $U + K_\gamma$, resp. U turn irrelevant as $\alpha'(t+u) - \beta'(u)$ becomes negative for these input values. Hence, the positive supremum of $\alpha(t+u) - \beta(u)$ can only be found for input values to α' and β' from the domain $[0, K_\alpha := U + K_\gamma]$, resp. $[0, K_\beta := U]$. As α coincides with α' for $t \in [0, K_\alpha]$ and β coincides with β' for $t \in [0, K_\beta]$, the above construction yields that

$$\text{for } t \leq K_\gamma : (\alpha \oslash \beta)(t) = (\alpha' \oslash \beta')(t). \qquad \blacksquare$$

As an example, we show how the domain bounding parameter U can be found when using linear overapproximations with a single segment. Accordingly, we define an overapproximation $\downarrow\!\alpha'(t)$ to an input curve α' as:

$$\alpha'(t) \leq \downarrow\!\alpha'(t) = \begin{cases} \max(0,\ N_\alpha + \rho \cdot t) & \text{if } N_\alpha + \rho \cdot t > 0 \\ 0 & \text{else} \end{cases}$$

and let $\uparrow\!\beta'(t)$ be defined analogously such that $\beta'(t) \geq \uparrow\!\beta'(t)$ holds, which gives that $N_\beta \leq 0$. The slope of $\uparrow\!\beta'(t)$ is denoted by σ in the following.

In this setting, we bound U for a constant K_γ as follows:

$$\begin{align}
&\alpha'(u+t) - \beta'(u) \geq 0 \tag{1}\\
&\text{with } \alpha'(t) \leq \downarrow\!\alpha'(t) \text{ and } \beta'(t) \geq \uparrow\!\beta'(t) \text{ we have} \tag{2}\\
&\downarrow\!\alpha'(u+t) - \uparrow\!\beta'(u) \geq \alpha'(u+t) - \beta'(u) \geq 0 \tag{3}\\
&\text{With } 0 \leq t \leq K_\gamma \text{ we get} \tag{4}\\
&\downarrow\!\alpha'(u+K_\gamma) - \uparrow\!\beta'(u) \geq \downarrow\!\alpha'(u+t) - \uparrow\!\beta'(t) \tag{5}
\end{align}$$

The definitions of the linear bounding functions yield:

$$\begin{align}
&N_\alpha + \rho(u + K_\gamma) - (N_\beta + \sigma u) \geq 0 \tag{7}\\
&\text{such that } u \leq \tfrac{N_\beta - N_\alpha + K_\gamma \rho}{\rho - \sigma} \tag{8}\\
&\text{and for } N_\beta < N_\alpha \text{ and } \rho < \sigma \text{ this is equal to} \tag{9}\\
&u \leq \tfrac{N_\alpha - N_\beta + K_\gamma \rho}{\sigma - \rho} \text{and for } U = \tfrac{N_\alpha - N_\beta + K_\gamma \rho}{\sigma - \rho} \tag{10}\\
&\text{we get } u \leq U \text{ as condition} \tag{11}
\end{align}$$

For $t \leq K_\gamma$ and $u > U : \alpha'(u+t) - \beta'(u) < 0$ and the requested bounds are $U + K_\gamma \leq K_\alpha$ and $U \leq K_\beta$. Beyond the domain bounds one only gets negative function values with the (min,+)-deconvolution. Hence, the function value of $\gamma(t)$ for any $t \in [0, K_\gamma]$ clearly takes its source in the compact domains of the input curves $[0, K_\alpha]$ and $[0, K_\beta]$.

D. Network Calculus Analysis on Compact Domains

After the backtracking through the flow equation terminates, a finite domain bound for each arrival and service curve is known. In the final step, we restrict the curves to their respective compact domains and solve the NC flow equation again. I.e., we analyze the network with arrival and service curves that accurately model the system on a domain sufficient for performance bounding. Fig. 2(D) already depicted the improved PMOO left-over service curve and the more accurate bounds it derives.

V. Conclusion

We presented the ingredients for implementing NC with compact domains. On the one hand, this idea fits well into implementations of NC analyses that are based on piecewise linear, pseudo-periodic curves. On the other hand, the idea is known to pay off [11]: NC as well as RTC analysis on compact domains increases the computational efficiency considerably. In this paper, we brought this advantage to advanced NC analysis principles such as PBOO and PMOO without sacrificing the accuracy of the system model.

References

[1] S. Bondorf and J. B. Schmitt, "Calculating Accurate End-to-End Delay Bounds – You Better Know Your Cross-Traffic," in *Proc. of ValueTools*, 2015.

[2] ——, "Improving Cross-Traffic Bounds in Feed-Forward Networks – There is a Job for Everyone," in *Proc. GI/ITG MMB & DFT*, 2016.

[3] A. Bouillard and E. Thierry, "An Algorithmic Toolbox for Network Calculus," *Journal of Discrete Event Dynamic Systems*, 2008.

[4] J.-Y. Le Boudec and P. Thiran, *Network Calculus: A Theory of Deterministic Queuing Systems for the Internet*. Springer, 2001.

[5] J. B. Schmitt, F. A. Zdarsky, and M. Fidler, "Delay Bounds under Arbitrary Multiplexing: When Network Calculus Leaves You in the Lurch ..." in *Proc. of IEEE INFOCOM*, 2008.

[6] L. Thiele, S. Chakraborty, and M. Naedele, "Real-Time Calculus for Scheduling Hard Real-Time Systems," in *Proc. of IEEE ISCAS*, 2000.

[7] R. Henia, A. Hamann, M. Jersak, R. Racu, K. Richter, and R. Ernst, *System Level Performance Analysis - the SymTA/S Approach*. The Institution of Electrical Engineers, London, United Kingdom, 2006.

[8] "Modular Performance Analysis Framework," www.mpa.ethz.ch.

[9] U. Suppiger, S. Perathoner, K. Lampka, and L. Thiele, "Modular performance analysis of large-scale distributed embedded systems: An industrial case study," ETH Zurich, Tech. Rep. 330, Nov 2010.

[10] ——, "A simple approximation method for reducing the complexity of modular performance analysis," ETH Zurich, Tech. Rep. 329, Aug 2010.

[11] N. Guan and Y. Wang, "Finitary real-time calculus: Efficient performance analysis of distributed embedded systems," in *Proc. of IEEE RTSS*, 2013.

[12] S. Perathoner, T. Rein, L. Thiele, K. Lampka, and J. Rox, "Modeling structured event streams in system level performance analysis," in *Proc. ACM SIGPLAN/SIGBED LCTES*, 2010.

[13] R. L. Cruz, "A Calculus for Network Delay. Part I: Network Elements in Isolation / Part II: Network Analysis," *IEEE Trans. Inf. Theory*, 1991.

[14] C.-S. Chang, *Performance Guarantees in Communication Networks*. Springer, 2000.

[15] M. Fidler, "Extending the Network Calculus Pay Bursts Only Once Principle to Aggregate Scheduling," in *Proc. of QoS-IP*, 2003.

[16] J. Schmitt, F. Zdarsky, and L. Thiele, "A Comprehensive Worst-Case Calculus for Wireless Sensor Networks with In-Network Processing," in *Proc. of IEEE RTSS*, 2007.

[17] J. B. Schmitt, F. A. Zdarsky, and I. Martinovic, "Improving Performance Bounds in Feed-Forward Networks by Paying Multiplexing Only Once," in *Proc. of GI/ITG MMB*, 2008.

[18] A. Bouillard, L. Jouhet, and E. Thierry, "Tight Performance Bounds in the Worst-Case Analysis of Feed-Forward Networks," in *Proc. of IEEE INFOCOM*, 2010.

[19] S. Bondorf and J. B. Schmitt, "Boosting Sensor Network Calculus by Thoroughly Bounding Cross-Traffic," in *Proc. IEEE INFOCOM*, 2015.

[20] S. Perathoner, K. Lampka, and L. Thiele, "Composing heterogeneous components for system-wide performance analysis," in *Proc. of DATE*, 2011.

[21] K. Lampka, S. Perathoner, and L. Thiele, "Component-based system design: analytic real-time interfaces for state-based component implementations," *Software Tools for Technology Transfer (STTT)*, 2012.