



Per-Flow Performance Guarantees in Networked Systems with Complex Feedback Structures

Anja Hamscher ✉ 

Distributed Computer Systems Lab (DISCO), RPTU Kaiserslautern-Landau, Germany

Lukas Wildberger ✉ 

Distributed Computer Systems Lab (DISCO), RPTU Kaiserslautern-Landau, Germany

Jens Schmitt ✉ 

Distributed Computer Systems Lab (DISCO), RPTU Kaiserslautern-Landau, Germany

Abstract

Many modern networked real-time systems encompass complex feedback structures *and* require stringent timing guarantees, especially bounds on the network delay. Network Calculus (NC) is a versatile methodology to compute such performance guarantees per individual flow; in particular, some fundamental results on how to deal with feedback exist. Yet, these are restricted to simple feedback structures and are mostly constrained to an analysis at the aggregate level (not per flow). In our work, we analyze more complex feedback structures than previously investigated by reducing them to canonical structures. We transform these closed-loop systems (with feedback) into open-loop systems (without feedback) and, subsequently, perform a per-flow analysis exploiting very recent NC results on per-flow performance guarantees. In a numerical experiment, we compare our new method to the current state-of-the-art which only allows for an aggregate FIFO analysis. We also compute how feedback constraints need to be allocated to ensure that a feedback system provides the same service as the system without feedback, in a sense providing for an optimal control. Furthermore, we compare different allocation strategies under a fixed budget for the feedback constraints.

2012 ACM Subject Classification Networks → Network architectures; Networks → Network performance evaluation; Computer systems organization → Real-time systems

Keywords and phrases Real-Time Networks, Network Calculus, Feedback Control

Digital Object Identifier 10.4230/LIPIcs.ECRTS.2025.18

1 Introduction

Traffic control is an important issue for networked systems with stringent QoS requirements, such as packet loss and deadlines. This control can be realized through various feedback mechanisms signaling between components in a system. The signaling can take different shapes. It can be used to model some type of acknowledgement for a sender to adjust its sending rate (window flow control). Finite memory buffers are another interesting concern, as generally in real-time systems lossless packet transfer is assumed. In this scenario, exceeding a buffer's capacity would inevitably lead to packet loss in the system. Consequently, some form of feedback is required, such that a receiving node can signal to its predecessor in the system that its output has to be paced down to not cause packet loss at the receiving node. Such feedback mechanisms are found in different application areas, for instance, in Network-on-Chip (NoC) [24], networked control systems [3] and manufacturing systems [11, 21, 23].

There are different ways to conduct the timing analysis of feedback systems. For NoCs, deterministic [27, 34, 35] as well as stochastic [17, 18, 22] modeling can be applied. We can also optimize buffer allocations to reduce the impact of an existing feedback control [13, 20]. For manufacturing systems, the predominant modeling and analysis is done with stochastic models [10, 28, 38]. A comprehensive review of existing analysis methods for manufacturing systems is found in [29].



© Anja Hamscher, Lukas Wildberger, and Jens Schmitt;
licensed under Creative Commons License CC-BY 4.0

37th Euromicro Conference on Real-Time Systems (ECRTS 2025).

Editor: Renato Mancuso; Article No. 18; pp. 18:1–18:25

Leibniz International Proceedings in Informatics



Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Another interesting option is to use Network Calculus (NC) to conduct a *worst-case* performance analysis of systems with feedback structures. NC is a versatile analytical methodology which is directly related to the real-time calculus [36] and has a formal link to response time analysis [31]. NC has been utilized to analyze a variety of systems, for instance, again NoCs [4, 9, 26, 30]. The basic framework resulted from the seminal work of Cruz [14, 15] and has been further developed by Chang [12] and Le Boudec [6]. An up-to-date overview of the framework is found in [7]. At its core, traffic is upper-bounded by *maximal arrival curves* and lower-bounded by *minimal arrival curves*, while server capacities are lower-bounded by *service curves*. Using these curves, performance bounds can be derived. These bounds are either computed *per-flow*, i.e., only a single flow of interest is considered, or on the aggregate, where all traffic along a given network path is considered.

Initial work on NC analysis of systems with feedback constraints has been done in [1, 2, 16]. Here, hop-by-hop and end-to-end window flow control was considered, where only a finite amount of unacknowledged packets can be sent. The initial work on feedback structures was extended for systems with bounded buffers in [12] for a system with multiple nodes; and in [37], where methods to model and analyze systems with finite buffers that cause system-wide back-pressure are explored. Based on [12], the technique was mapped to manufacturing blocking systems [5] and further analyzed to generalize the formulas for stream processing systems that deal with complex state dependencies in [8]. In [40], the authors compared the computation times of different end-to-end service curves for such systems. In [1, 12], the minimal window size for a single feedback loop to have no impact on a service curve is determined.

Despite these fundamental results, the current state-of-the-art NC analysis for systems with feedback control is still limited. While we can analyze systems on an aggregate level, per-flow performance bounds can only be calculated when assuming network-wide FIFO scheduling [30]. For non-FIFO scheduling (e.g. static priority), however, we encounter the inherent need of *strict service curves*. This special type of service curve enables the scheduling-agnostic subtraction of arrival curves. However, the concatenation of multiple of such curves generally destroys their strictness property [7, Prop. 6.2]. Yet, the concatenation of servers is crucial for an accurate end-to-end analysis.

In [19], an alternative to obtaining a per-flow service curve is presented, where the existing NC framework was extended to not require a strict service curve to calculate a residual service curve even under non-FIFO scheduling disciplines. In return, the existence of a minimal arrival curve is assumed. Exploiting this result, we are able to derive a general end-to-end service curve for a system with feedback structures while maintaining the ability to calculate per-flow performance bounds under non-FIFO scheduling. To this end, we make the following contributions in this paper:

- We generalize existing NC results for the transformation of an arbitrarily large closed-loop system, having a single feedback constraint, into an open-loop system.
- We study all possible types of canonical interdependence types between two feedback constraints and show how the interdependency is resolved. We derive a transformed service curve for each type, as well as an end-to-end service curve.
- We show how to resolve the transformation of a closed-loop system with arbitrarily many feedback constraints using the previously derived results. We present a method to calculate per-flow end-to-end performance bounds for arbitrary cross flows and feedback structures in tandem networks.
- We show how the optimal buffer dimensioning for arbitrary tandem networks and feedback structures without any adverse effects on the performance bounds can be calculated.

- We study different resource allocation schemes for the feedback constraints in a network. We evaluate their effects on per-flow delay bounds in comparison to optimal feedback resource allocations.

The remaining paper is structured as follows. We introduce the NC framework and important results in Section 2. Afterwards, we present the state-of-the-art NC results on systems with feedback structures in Section 3.1. In Section 3.2, we show that strict service curves are not suitable in the modelling of feedback systems. We conclude in Section 3.3 with a discussion on what is missing to perform a NC analysis for a general tandem network with arbitrary, potentially nested, feedback structures. In Section 4, we first derive an end-to-end service curve for a single feedback constraint in a general tandem. Then, we turn to general feedback tandems with many potentially interdependent feedback constraints. Here, we investigate different interdependence types between feedback constraints and show how to derive transformed service curves for their canonical versions. Finally, we discuss how to deal with cross flows and obtain per-flow performance bounds. In Section 5, we evaluate our new results against the existing NC analysis under FIFO scheduling. We demonstrate how the existing result for calculating optimal window sizes can be generalized for feedback constraints with interdependencies. At the end, we compare various resource allocation strategies for the feedback constraints to minimize delay bounds.

2 Network Calculus Background

We proceed with a brief overview of the relevant NC results that are used throughout the paper. Let \mathbb{R}^+ be the set of non-negative real numbers. We define the set of (min, plus) functions as $\mathcal{F} := \{f : \mathbb{R}^+ \rightarrow \mathbb{R} \cup \{+\infty\}\}$. Based on \mathcal{F} , we let \mathcal{F}^\uparrow be the set of non-decreasing functions $f \in \mathcal{F}$, and \mathcal{F}_0^\uparrow be the set of functions in \mathcal{F}^\uparrow with $f(0) = 0$. We define some basic operators that are the basis for other NC results.

► **Definition 1** (Basic Operators [7, Chapter 2]). *Let $f, g \in \mathcal{F}$. The min-plus convolution of f and g is defined as $f \otimes g(t) := \inf_{0 \leq s \leq t} \{f(t-s) + g(s)\}$, the min-plus deconvolution is defined as $f \oslash g(t) := \sup_{s \geq 0} \{f(t+s) - g(s)\}$. The max-plus deconvolution is defined as $f \overline{\otimes} g(t) := \inf_{s \geq 0} \{f(t+s) - g(s)\}$. We define the \wedge operator as $f \wedge g(t) := \min\{f(t), g(t)\}$.*

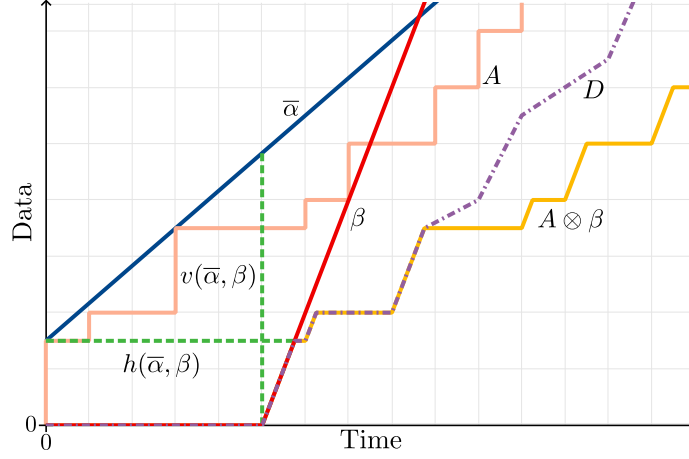
The *neutral element* of \otimes is defined as $\delta_0(0) = 0$ and $+\infty$ else. We define the *lower non-decreasing closure* as $\beta_\downarrow := \beta \overline{\otimes} 0$. We further make note of several properties of the \wedge and \otimes operators:

► **Remark 2** (Distributivity, Commutativity, Associativity [12, Chapter 2.1.1]). Let $f, g, h \in \mathcal{F}$. Then, it holds that

$$\begin{aligned} (f \otimes g) \wedge (f \otimes h) &= f \otimes (g \wedge h), \\ f \otimes g &= g \otimes f, \\ (f \otimes g) \otimes h &= f \otimes (g \otimes h). \end{aligned}$$

► **Remark 3** (Isotonicity of \otimes [6, Thm. 3.1.7]). Let $f, g, f', g' \in \mathcal{F}$. If $f \leq g$ and $f' \leq g'$, then $f \otimes f' \leq g \otimes g'$.

Next, we define various notions that are used to model a network and derive its performance bounds. Let $A, D \in \mathcal{F}_0^\uparrow$ be the *cumulative arrival* and *departure process* of a flow in the network, assuming causality $A \geq D$. Furthermore, we assume all systems to be lossless. We define the most important performance measures for such a system:



■ **Figure 1** Illustration of different processes, curves and deviations.

► **Definition 4** (Backlog at Time t [6, Def. 1.1.1]). *The backlog of system \mathcal{S} at time t is the vertical distance between arrival process A and departure process D at time t ,*

$$q(t) := A(t) - D(t).$$

► **Definition 5** (Virtual Delay at Time t [6, Def. 1.1.1]). *The virtual delay of data arriving at system \mathcal{S} at time t is the time until this data would be served, assuming FIFO-per-flow order of service,*

$$d(t) := \inf \{ \tau \geq 0 : A(t) \leq D(t + \tau) \}.$$

Arrival and service curves are essential elements of the performance analysis using NC.

► **Definition 6** (Maximal and Minimal Arrival Curve [7, Def. 5.1,5.2]). *Let $\bar{\alpha}, \underline{\alpha} \in \mathcal{F}_0^+$. We say that $\bar{\alpha}$ is a maximal arrival curve for arrival process A , and $\underline{\alpha}$ is a minimal arrival curve for A , if it holds for all $0 \leq s \leq t$ that*

$$\underline{\alpha}(t-s) \leq A(t) - A(s) \leq \bar{\alpha}(t-s).$$

► **Definition 7** (Min-Plus Service Curve [7, Def 5.3]). *Let a flow with arrival process A and departure process D traverse a system \mathcal{S} . The system offers a min-plus service curve β to the flow if $\beta \in \mathcal{F}$ and it holds for all $t \geq 0$ that*

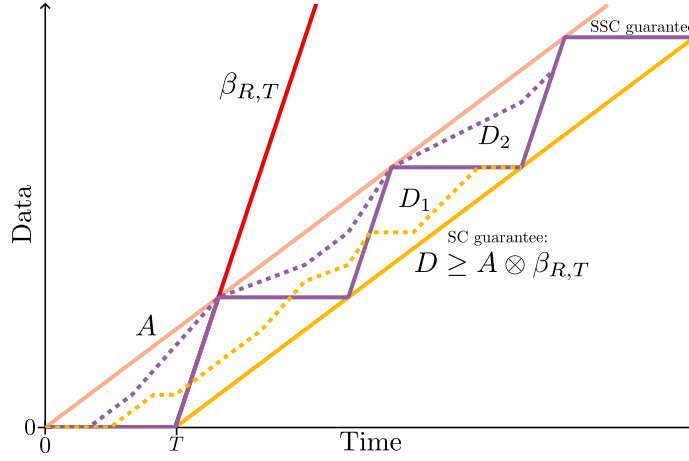
$$D(t) \geq A \otimes \beta(t) = \inf_{0 \leq s \leq t} \{ A(t-s) + \beta(s) \}.$$

An example of a maximal arrival curve is the *token-bucket* arrival curve, given by $\gamma_{r,b}(t) = b + rt$ for $t > 0$, and $\gamma_{r,b}(0) = 0$. A frequently employed function for both minimal arrival curves and service curves is the *rate-latency* curve $\beta_{R,T}(t) := R \cdot [t - T]^+$. An illustration of a token-bucket arrival, a rate-latency service curve, as well as the corresponding arrival and departure process, is given in Figure 1.

The following definition provides a stricter notion of a service curve.

► **Definition 8** (Strict Service Curve [7, Def. 5.5]). *A system offers a strict service curve $\beta \in \mathcal{F}$ to a flow if, during any backlogged period $(s, t]$ (i.e. $\forall t' \in (s, t], q(t') > 0$), it holds that*

$$D(t) - D(s) \geq \beta(t-s).$$



■ **Figure 2** Difference between the provided guarantees of strict service curve (SSC) and service curve (SC). D_1 is an example departure process adhering only to the service curve guarantee, D_2 adheres to both guarantees.

The difference between a min-plus and a strict service curve is depicted in Figure 2. It can be observed that whenever the system is backlogged, the strict service curve offers rate R to the backlogged arrivals until they have been served. On the contrary, the min-plus service curve offers a minimal guarantee of $A \otimes \beta$, independent of the amount of backlog in the system.

Now consider a system that arbitrarily multiplexes two flows constrained by arrival curves α_1, α_2 and a server guaranteeing a strict service β to the aggregate of the flows. A so-called *residual service curve* for a specific flow, i.e., the fraction of the full service a server offers that is guaranteed to this specific flow can be calculated by [6]

$$\beta_{\text{ARB}}^1(t) := [\beta - \alpha_2]^+.$$

A larger residual service curve, under FIFO multiplexing, is given and applied in Section 5.1 (see Equation (15)).

We define two characteristic distances between two given functions.

► **Definition 9.** Let $f, g \in \mathcal{F}$. The vertical deviation between f and g is defined as

$$v(f, g) := \sup_{t \geq 0} \{f(t) - g(t)\},$$

and the horizontal deviation between f and g is defined as

$$h(f, g) := \inf\{d \geq 0 \mid f \oslash g(-d) \leq 0\}.$$

Using these concepts, one can derive performance bounds for the measures defined previously [7, p. 115], [6, p. 118].

► **Theorem 10 (Performance Bounds).** Assume an arrival process A , constrained by maximal arrival curve $\bar{\alpha} \in \mathcal{F}_0^\uparrow$, traverses a system \mathcal{S} . Let the system \mathcal{S} offer a service curve $\beta \in \mathcal{F}_0^\uparrow$. The virtual delay $d(t)$ satisfies for all t

$$d(t) \leq h(\bar{\alpha}, \beta).$$

The backlog $q(t)$ satisfies for all t

$$q(t) \leq v(\bar{\alpha}, \beta).$$

In Figure 1, an example on where the performance bounds are measured for the token-bucket arrival and rate-latency service curve is given.

A central result of NC is the concatenation theorem. It is crucial in order to calculate accurate end-to-end performance bounds.

► **Theorem 11** (Concatenation Theorem [12, Thm. 4.3.2]). *Let a flow with arrival process A traverse systems \mathcal{S}_1 and \mathcal{S}_2 , offering service curves $\beta_1, \beta_2 \in \mathcal{F}$, in sequence. Then, the concatenation of the two systems, $S_{1,2} = \langle \mathcal{S}_1, \mathcal{S}_2 \rangle$, offers an end-to-end service curve $\beta_{1,2} = \beta_1 \otimes \beta_2$ to the arrival process.*

In the context of the modeling of feedback mechanisms, an important concept is the so-called sub-additive closure. It transforms a curve such that the sub-additive property holds.

► **Definition 12** (Sub-additive Functions [6, Def. 3.1.11]). *Let $f \in \mathcal{F}$. Then f is sub-additive if for all $s, t \geq 0$*

$$f(t + s) \leq f(t) + f(s).$$

► **Definition 13** (Sub-additive Closure [6, Def. 3.1.12]). *Let $f \in \mathcal{F}$. The sub-additive closure of f is defined by*

$$f^* := \inf_{n \geq 0} \left\{ f^{(n)} \right\},$$

where $f^{(n)}$ is the n -fold self-convolution of f , i.e., $f^{(0)} = \delta_0$, $f^{(1)} = f$ and $f^{(n)} = \bigotimes_{i=0}^{n-1} f^{(i)}$ for $n \geq 2$.

An example of the result obtained through the sub-additive closure is illustrated in Figure 5. We state some useful properties of the sub-additive closure.

► **Lemma 14** (Properties of Sub-Additive Closure [12, Lem. 2.1.5]). *Let $f, g \in \mathcal{F}^\uparrow$, it holds that*

$$(f \otimes g)^* \geq f^* \otimes g^*, \tag{1}$$

$$(f^*)^* = f^*, \tag{2}$$

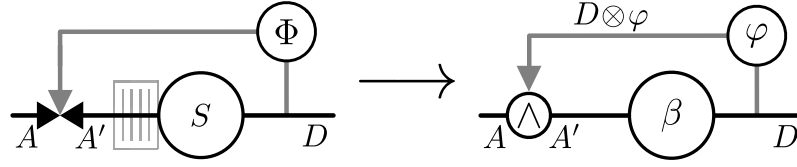
$$f^* \otimes f^* = f^*. \tag{3}$$

3 Simple Feedback Systems

We divide this section into three parts. We first cover the state of the art on the feedback system analysis using NC in Section 3.1. In Section 3.3, we discuss how we extend the state of the art to deal with general feedback tandem networks with arbitrary length and number of feedback constraints in the system, including interdependencies between feedback structures. In Section 3.2, we show that, assuming strict service curves, existing feedback constraints would have no effect on the system, and, thus, strict service curves are not suitable for modelling feedback systems.

3.1 State of the Art

We first consider a basic system with a feedback structure as in Figure 3 on the left. We assume that the server S has a finite buffer that holds incoming packets of the arrivals A . After processing, some departures D leave the server. To ensure that there are no more



■ **Figure 3** General structure of a feedback constraint under consideration.

arrivals to the node than the buffer can hold at a time, the node signals to the sender (or previous node) whether space is still left in the buffer. This feedback is modeled by some function Φ that depends on the departures. If there is no more space in the buffer, the sender throttles its sending rate accordingly and buffers the data itself until it is allowed to send again, essentially performing a blocking write. This mechanism is represented by the gray backwards loop in Figure 3. The effect of the throttling adjusts A to A' , which we call the *effective arrivals* to the server.

The analysis of the system is then done in several steps. We call the model in Figure 3 on the left a *closed-loop system*, i.e., the system has (feedback) edges that form a closed loop. The closed-loop system is *min-plus non-linear* [6, p. xix], and consequently cannot be directly analyzed using NC, as the framework is tailored to *min-plus linear* systems. It is hence the first step to resolve this feedback loop by transforming the closed-loop into a min-plus linear *open-loop* system. The general idea here is that we “encode” the flow control into the node at which the control signal is fed back into the data flow. The feedback is then effectively exerted by the server itself by adjusting the service it can offer to incoming arrivals. The resulting open-loop system is min-plus linear, and can be analyzed using NC. Since the transformation is a safe operation, any performance bounds we obtain for the open-loop system are also valid bounds for the closed-loop system. There is, however, generally no equivalence between the two systems, i.e., the open-loop system is an upper bound on the closed-loop system.

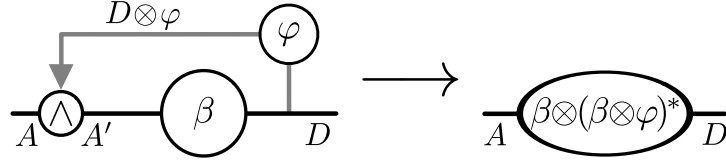
For us to be able to transform the closed-loop system for a NC analysis, we perform the transformation of the example system in Figure 3 from the left to the right. To this end, we introduce some preliminaries. We let $F = (i, j, \Phi)$ be a feedback constraint or loop, where i is the server in front of which the feedback is entering the data path, j is the server behind which the feedback loop starts. $\Phi(D(t))$ represents the cumulative feedback at time t , and represents the type of feedback that is employed, as discussed previously. We call $\varphi \in \mathcal{F}^\uparrow$ with $\varphi(0) > 0$ a *feedback curve* when $\Phi \circ D \geq D \otimes \varphi$. By abuse of notation, we also write $F = (i, j, \varphi)$ interchangeably with $F = (i, j, \Phi)$. In most existing NC work on flow control, a fixed window size u is assumed that cannot be exceeded by incoming traffic, i.e., $\Phi(D(t)) = D(t) + u$, $u > 0$. The feedback curve for this fixed window size is part of the general model presented in [7, Chapter 6.2.2] and is defined as

$$I_u(t) := \begin{cases} \infty, & \text{if } t > 0, \\ u, & \text{otherwise,} \end{cases} \quad (4)$$

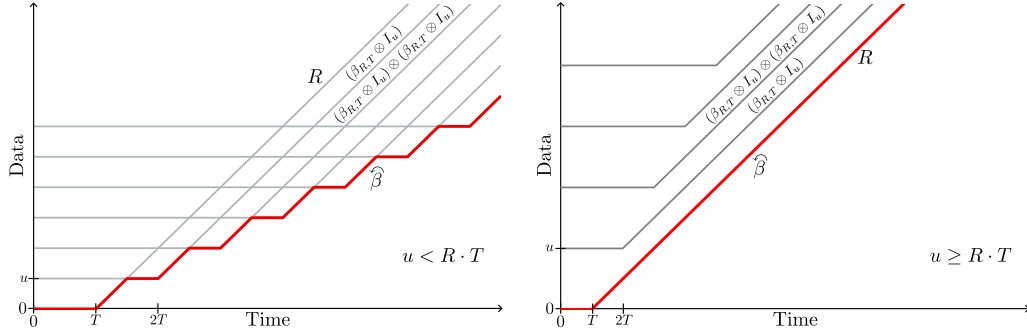
since $D \otimes I_u(t) = D(t) + u$ (see also [12, p. 79]). Further, we specify the effective arrivals as

$$A' := A \wedge (D \otimes \varphi). \quad (5)$$

Through the usage of the minimum (\wedge) as a throttle, we ensure that the arrivals can never exceed the allowed maximum size $D \otimes \varphi$ to not violate the feedback control.



■ **Figure 4** Transformation of a closed- into an open-loop system.



(a) Transformed service curve $\hat{\beta}$ for $u < R \cdot T$. (b) Transformed service curve $\hat{\beta}$ for $u \geq R \cdot T$.

■ **Figure 5** Effect of the window size u on the transformed service curve $\hat{\beta}$.

Besides causality in the system, i.e., $D \leq A'$, it is further assumed that the server S offers a min-plus service curve $\beta \in \mathcal{F}^\uparrow$ to A'

$$D \geq A' \otimes \beta = (A \wedge (D \otimes \varphi)) \otimes \beta, \quad (6)$$

where we also used Equation (5). Based on Equation (6), the next step is to transform the closed-loop into an open-loop system (see also Figure 4). We restate the central result from [12, Theorem 2.1.6]:

► **Theorem 15.** *Let an arrival process A traverse a system S offering service curve $\beta \in \mathcal{F}$. The system is subject to a feedback curve φ . If*

$$D \geq (A \wedge (D \otimes \varphi)) \otimes \beta,$$

then

$$D \geq A \otimes \beta \otimes (\beta \otimes \varphi)^*.$$

Using this result, we effectively encode the feedback constraint into β by calculating a *transformed* service curve $\hat{\beta}$ (also called equivalent service curve in [40]). Here, the transformed service curve of β is $\hat{\beta} = \beta \otimes (\beta \otimes \varphi)^*$. The transformation of the example closed-loop into open-loop system is illustrated in Figure 4. Here, the feedback loop is replaced by the transformed service curve. Recalling the definition of the sub-additive closure in Definition 13, we illustrate its effects in the transformed service curve. Assuming that β is a rate-latency service curve and $\varphi = I_u$, the transformed service curve is depicted as the red curve in Figure 5. The grey curves illustrate the sub-additive closure operation $(\beta_{R,T} \otimes I_u)^*$. Here, each curve represents the result of another self-convolution of $\beta_{R,T} \otimes I_u$. The illustrated two different scenarios depend on the relation between the service curve parameters R and T and the size of the feedback constraint u . If we have for the so-called bandwidth-delay

product [1] that $R \cdot T \leq u$, the feedback constraint has no effect on the system, and $\widehat{\beta} = \beta$ (see Figure 5b). This stems from the fact that $\beta_{R,T}$ is always smaller than $(\beta_{R,T} \otimes I_u)^*$. If $R \cdot T > u$, calculating the sub-additive closure leads to a reduction of the transformed service curve, and consequently $\widehat{\beta} < \beta$. This can be observed in Figure 5a. This reduction stems from the fact that $\beta_{R,T}$ quickly becomes larger than $(\beta_{R,T} \otimes I_u)^*$. Hence, $\widehat{\beta}$ follows the sub-additive closure $(\beta_{R,T} \otimes I_u)^*$.

To recapitulate, we are given a closed-loop system, similar to the system we have examined in the introductory example. We use Theorem 15 to encode the feedback constraint existing in the closed-loop system into the service curve directly, resulting in an open-loop system that has no feedback any more, but a transformed service curve $\widehat{\beta}$. This new system constitutes a min-plus linear upper bound to the original system and can be analyzed with respect to performance bounds using NC.

We introduce another aspect of feedback constraints that will be subject to analysis in the paper. Assuming a static window flow control using $\varphi = I_u$ for some $u > 0$, we can obtain the optimal size of u such that $\beta = \widehat{\beta}$ holds by

$$u = (\beta \otimes \beta) \otimes \beta(0).$$

This is studied in detail in [1, 7, 12]. Looking again at Figure 5, the optimal u would be attained if $u = R \cdot T$, since all the grey curves would then lie on the red curve, or, more formally, $\beta_{R,T} = (\beta_{R,T} \otimes I_u)^*$. We will expand on this result in Section 5.2.

We note that Theorem 15 makes no assumption about the type of service curve that is required to perform the transformation to an open-loop system, i.e., it only requires a min-plus service curve. In fact, we can show that this is the only relevant type of service curve we have to consider. We will discuss this in more detail in the next section.

3.2 The Issue of Strict Service Curves

Here, we show that, assuming strict service curves, feedback constraints would have *no* effect on closed-loop systems. We prove this observation for the special case of fixed window flow control, i.e. $\varphi = I_u$. To this end, we first show that a backlogged period of a system without feedback is contained in the backlogged period of a closed-loop system.

► **Lemma 16.** *Let an arrival process A traverse a server \mathcal{S} offering a strict service curve β . The feedback constraint in the system has a fixed size u . Let $A' := A \wedge D \otimes I_u$. It holds that*

$$A(t) > D(t) \Rightarrow A'(t) > D(t). \quad (7)$$

Proof. We prove by contradiction. Assume that $A(t) > D(t)$, but $A'(t) = D(t)$ (due to causality $A'(t) < D(t)$ is impossible). Then,

$$D(t) = A'(t) = A(t) \wedge D \otimes I_u. \quad (8)$$

But, as per assumption it holds that $A(t) > D(t)$ and also $D \otimes I_u > D$ with $u > 0$. Hence, Equation (8) does not hold and thus Equation (7) must hold. ◀

This shows that, under strict service, any backlogged period for the original arrivals A is also a backlogged period for the effective arrivals A' and therefore we have:

► **Proposition 17.** *Let an arrival process A traverse a server \mathcal{S} offering a strict service curve β . The system has a feedback constraint of size u . Let $A' := A \wedge D \otimes I_u$. If β is a strict service curve for the closed-loop system, it is also a strict service curve for the open-loop system, i.e., for any backlogged period $[s, t]$ for the arrivals A , we have that*

$$D(t) - D(s) \geq \beta(t - s).$$

Proof. Let $[s, t]$ be a backlogged period for the arrivals A . From Lemma 16, it holds that $[s, t]$ is also a backlogged period for A' . It follows that for both the closed- and open-loop system

$$D(t) - D(s) \geq \beta(t - s). \quad \blacktriangleleft$$

The reason behind this somewhat surprising observation is that the latency of a strict service curve does not model a physical propagation delay (as the service curve). As such, we are never facing the case that the bandwidth delay product [1] is larger than the feedback constraint u , and thus the feedback constraint has no impact on a system with a strict service curve. In other words, this renders strict service curves meaningless in the context of feedback system modelling.

3.3 Extending the State of the Art

So far, we only treated the basic case of a single feedback constraint. However, what we try to achieve is to be able to transform a closed-loop system with *arbitrary* many flow control mechanisms into an open-loop system. This has not been done in literature, previously. We have results for a feedback constraint spanning up to two nodes [5, 7, 8, 12, 19], but there are no existing results for feedback constraints spanning an arbitrary number of nodes. This is desirable, however, as we can have systems that employ different kinds of feedback mechanisms, such as an end-to-end window flow control or more complex signaling mechanisms for finite buffers that span over more than two nodes. This inevitably leads to feedback constraints that have interdependencies with other feedback constraints, where the transformed service curve of one feedback constraint is part of another feedback constraint. In Section 4, we identify *all* of these interdependence types, and how to transform them into corresponding open-loop systems. This, in return, enables the NC analysis for any kind of feedback structure in a tandem network, regardless of the existing interdependencies between feedback constraints in the network.

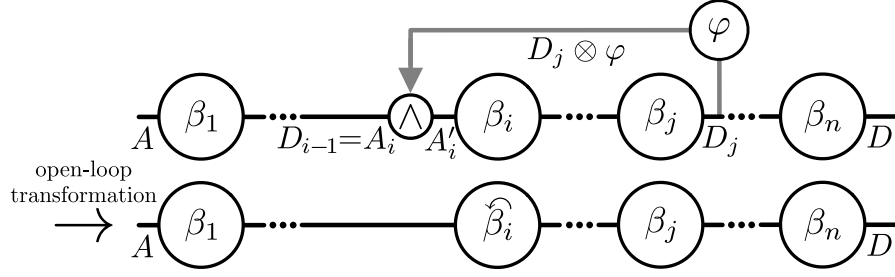
4 Complex Feedback Systems

In the following, we extend existing results for analyzing more complex feedback systems using NC. We first provide a straightforward generalization of Theorem 15, allowing us to analyze a single feedback constraint spanning an arbitrary number of nodes. Then, we study the different canonical interdependence structures systems with multiple feedback constraints can exhibit, and how to transform them into open-loop systems. Here, we show how to obtain transformed service curves for all relevant service curves in the closed-loop system, as well as an end-to-end service curve for the open-loop system. Finally, we discuss how we can obtain per-flow performance bounds using the respective open-loop systems despite the open-loop system consisting of min-plus servers.

4.1 Single Feedback Systems of Arbitrary Length

We assume a network as in Figure 6, where we have n nodes as well as one feedback constraint $F = (i, j, \varphi)$ in the system. To that end, we first generalize some of the equations introduced in the previous section. We define the effective arrivals A'_i at server i as

$$A'_i := A_i \wedge (D_j \otimes \varphi).$$



■ **Figure 6** Single feedback spanning over $j - i$ nodes.

The departures D_i at server i are characterized by

$$D_i \geq A'_i \otimes \beta_i.$$

Theorem 15 can then be generalized as follows:

► **Proposition 18.** *Let a flow f traverse a tandem of n nodes, each offering a service curve $\beta_k, k \in \{1, \dots, n\}$, with a single feedback constraint $F = (i, j, \varphi)$. The tandem offers an end-to-end service curve to the flow f*

$$D \geq A \otimes \beta_{e2e}, \text{ with } \beta_{e2e} := \bigotimes_{k=1}^{i-1} \beta_k \otimes \widehat{\beta}_i \otimes \bigotimes_{l=i+1}^n \beta_l, \quad (9)$$

and a transformed service curve at node i

$$\widehat{\beta}_i := \beta_i \otimes \left(\bigotimes_{k=i}^j \beta_k \otimes \varphi \right)^*. \quad (10)$$

Proof. Let $A_i, D_i, i \in \{1, \dots, n\}$, be the arrival and departure process at each node i . First, using Theorem 11, we calculate respective service curves for the subsystems from node 1 to node $i - 1$, node $i + 1$ to node j^1 , and node $j + 1$ to node n :

$$\beta_{1,i-1} = \bigotimes_{k=1}^{i-1} \beta_k, \quad \beta_{i+1,j} = \bigotimes_{k=i+1}^j \beta_k, \quad \beta_{j+1,n} = \bigotimes_{k=j+1}^n \beta_k.$$

The arrivals at node i can then be characterized by

$$A_i = D_{i-1} \geq A \otimes \beta_{1,i-1}.$$

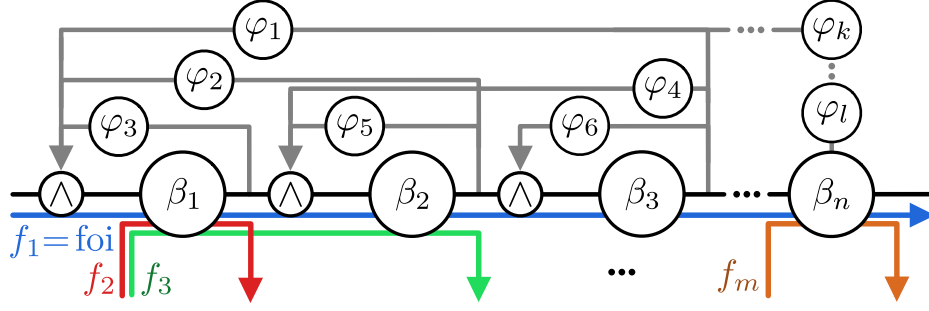
For the departures at node j we have that

$$D_j \geq (A_i \otimes \beta_i \otimes \beta_{i+1,j}) \wedge (D_j \otimes (\beta_i \otimes \beta_{i+1,j} \otimes \varphi)).$$

Consequently, the feedback loop at node i can be transformed into an open-loop system using Theorem 15:

$$\widehat{\beta}_i = \beta_i \otimes (\beta_i \otimes \beta_{i+1,j} \otimes \varphi)^* = \beta_i \otimes (\beta_{i,j} \otimes \varphi)^*,$$

¹ If $i = j$, we assume as usual that the empty convolution becomes the neutral element δ_0 .



■ **Figure 7** General feedback tandem for n nodes and m flows.

proving Equation (10). Then, using Theorem 11 again, the final output of the tandem is bounded by

$$\begin{aligned} D &\geq A \otimes \beta_{1,i-1} \otimes \hat{\beta}_i \otimes \beta_{i+1,j} \otimes \beta_{j+1,n} \\ &= A \otimes \beta_{1,i-1} \otimes \hat{\beta}_i \otimes \beta_{i+1,n}, \end{aligned}$$

where $\beta_{i+1,n} = \beta_{i+1,j} \otimes \beta_{j+1,n}$. This proves Equation (9). ◀

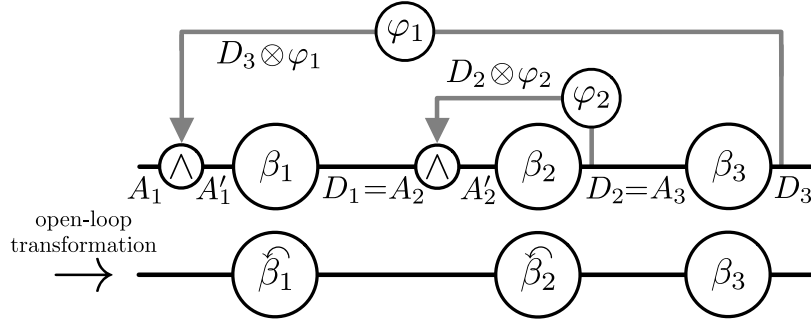
Equipped with Proposition 18, we can deal with single feedback loops of arbitrary length. Next, we proceed with an analysis of closed-loop systems that have multiple feedback loops.

4.2 Feedback Interdependence Types

Figure 7 illustrates the general feedback tandem network that we aim to be able to analyze. Here, we consider n nodes in a network, as well as an arbitrary number of feedback constraints l . In Figure 7, *all* possible interdependence types between feedback constraints that may occur are shown. In addition, a general flow aggregate consisting of multiple flows that may interfere with the flow of interest in any possible way across the nodes is illustrated (in colors). We point out again that the transformation of a closed-loop into an open-loop system is done *before* we consider any existing flows in the system.

In the following, we assume an ordered set of feedback constraints $F_l = (i_l, j_l, \varphi_l)$, $l = 1, \dots, k$. The feedback loops are lexicographically ordered by the data path entering node i_l as first criterion in ascending order, and the starting node of the control signal j_l as second criterion in descending order. Let F_l and F_m be two feedback constraints with $l \leq m$. If $j_l < i_m$, then the two feedback loops are not interdependent with each other and can be evaluated independently from each other using Proposition 18. If, however, $j_l \geq i_m$, then they are interdependent and their respective evaluation needs to take that into account.

There are different types of interdependence. First, let us assume that $i_l < i_m$, i.e., the second feedback enters the data path at a strictly later point in the tandem than the first one. If also $j_l \geq j_m$, we have a so-called *contained* interdependency between the constraints F_l and F_m (see also Figure 8 for an example). If, on the other hand, $j_l < j_m$, then we call it an *overlapping* interdependency between F_l and F_m (see also Figure 9 for an example). Now let us assume $i_l = i_m$, i.e., both feedback loops enter the data path at the same point in the tandem (see also Figure 10a for an example). Due to the lexicographical ordering of the feedback constraints we have that $j_l \geq j_m$. While this configuration is similar to the contained interdependence type, the treatment of this special case justifies a separate type called *compounded* interdependency.



■ **Figure 8** Canonical contained feedback system.

Figures 8-10 illustrate canonical versions of these three interdependence types, i.e., minimal tandem networks that exhibit each of the previously classified interdependence types. We note that we can adjust these structures to a larger size by making use of Proposition 18. In the following, we derive transformed service curves as well as an end-to-end service curve for each canonical interdependence type.

4.2.1 Contained Feedback

As illustrated in Figure 8, we have an arrival process A , and each node offers a service curve β_i . We also have feedback constraints $F_1 = (1, 3, \varphi_1)$ and $F_2 = (2, 2, \varphi_2)$. We first transform F_2 into an open-loop system, as none of the servers it spans are subject to the feedback constraint F_1 . We have for F_2 that

$$A'_2 = A_2 \wedge (D_2 \otimes \varphi_2), \quad D_2 \geq A'_2 \otimes \beta_2.$$

Applying Theorem 15 to this, we obtain

$$D_2 \geq A_2 \otimes \beta_2 \otimes (\beta_2 \otimes \varphi_2)^*, \quad (11)$$

and hence $\hat{\beta}_2 = \beta_2 \otimes (\beta_2 \otimes \varphi_2)^*$.

Turning to the feedback constraint F_1 , with Equation (11) and starting from the final departures D_3 , we derive

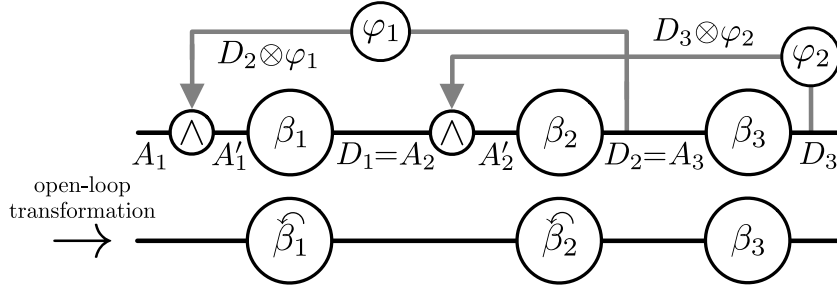
$$\begin{aligned} D_3 &\geq D_2 \otimes \beta_3 \\ &\geq A_2 \otimes \beta_2 \otimes (\beta_2 \otimes \varphi_2)^* \otimes \beta_3 \\ &\geq A'_1 \otimes \beta_1 \otimes \beta_2 \otimes (\beta_2 \otimes \varphi_2)^* \otimes \beta_3. \end{aligned}$$

In the third line, the service curve property of β_1 is used and that $A_2 = D_1$. We have the effective arrivals $A'_1 = A_1 \wedge D_3 \otimes \varphi_1$, and can apply Proposition 18 to obtain

$$\begin{aligned} D_3 &\geq A_1 \otimes \beta_1 \otimes (\beta_1 \otimes \beta_2 \otimes (\beta_2 \otimes \varphi_2)^* \otimes \beta_3 \otimes \varphi_1)^* \otimes \beta_2 \otimes (\beta_2 \otimes \varphi_2)^* \otimes \beta_3 \\ &\geq A_1 \otimes \beta_1 \otimes (\beta_1 \otimes \beta_2 \otimes \beta_3 \otimes \varphi_1)^* \otimes ((\beta_2 \otimes \varphi_2)^*)^* \otimes \beta_2 \otimes (\beta_2 \otimes \varphi_2)^* \otimes \beta_3 \\ &= A_1 \otimes \beta_1 \otimes (\beta_1 \otimes \beta_2 \otimes \beta_3 \otimes \varphi_1)^* \otimes \beta_2 \otimes (\beta_2 \otimes \varphi_2)^* \otimes \beta_3, \end{aligned}$$

where we used Lemma 14, Equation (1) in line 2 and Equation (2) in line 3, as well as the commutativity of the min-plus convolution (see Remark 2). From this, we directly obtain the end-to-end service curve for the flow aggregate as

$$\beta_{e2e} = \beta_1 \otimes (\beta_1 \otimes \beta_2 \otimes \beta_3 \otimes \varphi_1)^* \otimes \beta_2 \otimes (\beta_2 \otimes \varphi_2)^* \otimes \beta_3,$$



■ **Figure 9** Canonical overlapping feedback system.

and from Proposition 18, we have that

$$\hat{\beta}_1 = \beta_1 \otimes (\beta_2 \otimes \varphi_2)^* \otimes (\beta_1 \otimes \beta_2 \otimes \beta_3 \otimes \varphi_1)^*.$$

It can be observed that we can effectively “open the loop” for F_2 *before* using it to open the loop for F_1 . For the analysis of larger feedback structures this ordering relation is important to take into account.

4.2.2 Overlapping Feedback

Let $F_1 = (1, 2, \varphi_1)$ and $F_2 = (2, 3, \varphi_2)$ be the two feedback constraints as illustrated in Figure 9. We first focus on F_2 only and transform the corresponding subsystem (node 2 and 3) into an open-loop system using Proposition 18 to obtain

$$D_3 \geq A_2 \otimes \beta_2 \otimes (\beta_2 \otimes \beta_3 \otimes \varphi_2)^* \otimes \beta_3,$$

and hence $\hat{\beta}_2 = \beta_2 \otimes (\beta_2 \otimes \beta_3 \otimes \varphi_2)^*$. Turning to F_1 now and using $\hat{\beta}_2$, we have a single loop feedback system for which we can apply Proposition 18 again to obtain

$$\begin{aligned} D_3 &\geq A_1 \otimes \beta_1 \otimes (\beta_1 \otimes \hat{\beta}_2 \otimes \varphi_1)^* \otimes \hat{\beta}_2 \otimes \beta_3 \\ &= A_1 \otimes \beta_1 \otimes (\beta_1 \otimes \beta_2 \otimes \varphi_1 \otimes (\beta_2 \otimes \beta_3 \otimes \varphi_2)^*)^* \otimes \beta_2 \otimes (\beta_2 \otimes \beta_3 \otimes \varphi_2)^* \otimes \beta_3 \\ &\geq A_1 \otimes \beta_1 \otimes \beta_2 \otimes (\beta_1 \otimes \beta_2 \otimes \varphi_1)^* \otimes ((\beta_2 \otimes \beta_3 \otimes \varphi_2)^*)^* \otimes (\beta_2 \otimes \beta_3 \otimes \varphi_2)^* \otimes \beta_3 \\ &= A_1 \otimes \beta_1 \otimes (\beta_1 \otimes \beta_2 \otimes \varphi_1)^* \otimes \beta_2 \otimes (\beta_2 \otimes \beta_3 \otimes \varphi_2)^* \otimes \beta_3, \end{aligned}$$

where we used Lemma 14, Equation (1) in line 3, and Equations (2) and (3) in line 4, as well as the commutativity of \otimes . So, the end-to-end service curve is given by

$$\beta_{\text{e2e}} = \beta_1 \otimes (\beta_1 \otimes \beta_2 \otimes \varphi_1)^* \otimes \beta_2 \otimes (\beta_2 \otimes \beta_3 \otimes \varphi_2)^* \otimes \beta_3.$$

From Proposition 18, we also have that

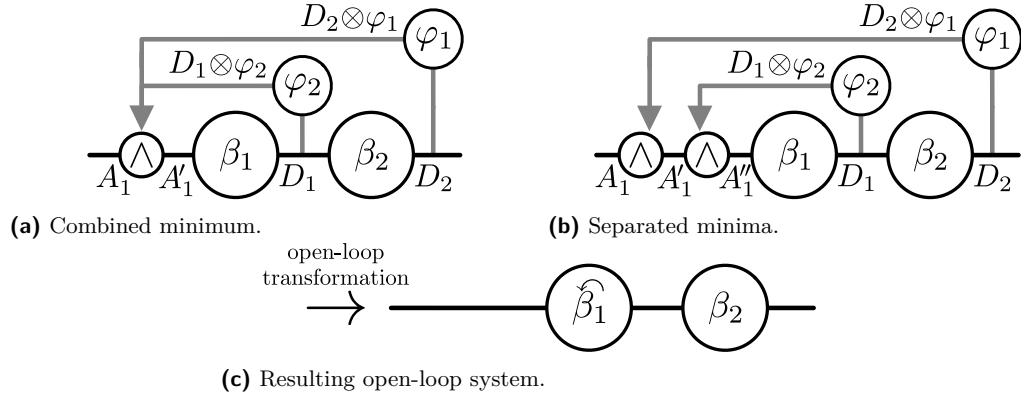
$$\hat{\beta}_1 = \beta_1 \otimes (\beta_1 \otimes \beta_2 \otimes \varphi_1)^* \otimes (\beta_2 \otimes \beta_3 \otimes \varphi_2)^*.$$

These results have been previously stated and proven in [5, Theorem 3.18], [12, p. 82-83], but they are included in this paper for completeness.

Again, it can be observed that we effectively open the loop for F_2 *before* F_1 .

4.2.3 Compounded Feedback

The canonical compounded feedback pattern illustrated in Figure 10a is structurally different from the others as the feedback loops enter the data path at the same point in the tandem. In order to ease deriving an end-to-end service curve for compounded feedback, we first observe



■ **Figure 10** Equivalent canonical compounded feedback structures.

that both feedback structures in Figures 10a and 10b are equivalent. Let $F_1 = (1, 2, \varphi_1)$ and $F_2 = (1, 1, \varphi_2)$ be the two feedback constraints. For the system in Figure 10a, the effective arrivals are given as

$$A'_1 = A_1 \wedge (D_2 \otimes \varphi_1) \wedge (D_1 \otimes \varphi_2). \quad (12)$$

For the system in Figure 10b, they are

$$\begin{aligned} A'_1 &= A_1 \wedge (D_2 \otimes \varphi_1), \\ A''_1 &= A'_1 \wedge (D_1 \otimes \varphi_2) = A_1 \wedge (D_2 \otimes \varphi_1) \wedge (D_1 \otimes \varphi_2). \end{aligned} \quad (13)$$

As Equations (12) and (13) are equivalent and each characterizes the respective feedback structure, we can also use the pattern in Figure 10b to evaluate the canonical compounded interdependence pattern. To that end, we transform the inner feedback F_2 first. The partial open-loop system immediately follows from Theorem 15:

$$D_1 \geq A'_1 \otimes \beta_1 \otimes (\beta_1 \otimes \varphi_2)^*.$$

For the outer feedback loop F_1 , we can use Proposition 18 again to obtain

$$\begin{aligned} D_2 &\geq A_1 \otimes \beta_1 \otimes (\beta_1 \otimes \varphi_2)^* \otimes (\beta_1 \otimes (\beta_1 \otimes \varphi_2)^* \otimes \beta_2 \otimes \varphi_1)^* \otimes \beta_2 \\ &\geq A_1 \otimes \beta_1 \otimes (\beta_1 \otimes \varphi_2)^* \otimes (\beta_1 \otimes \beta_2 \otimes \varphi_1)^* \otimes \beta_2, \end{aligned}$$

where we used Lemma 14 again in line 2.

Therefore, $\beta_{e2e} = \beta_1 \otimes (\beta_1 \otimes \varphi_2)^* \otimes (\beta_1 \otimes \beta_2 \otimes \varphi_1)^* \otimes \beta_2$ is the end-to-end service curve for the flow aggregate and the transformed service curve for β_1 is obtained as

$$\hat{\beta}_1 = \beta_1 \otimes (\beta_1 \otimes \varphi_2)^* \otimes (\beta_1 \otimes \beta_2 \otimes \varphi_1)^*.$$

Again, we observe that F_2 is evaluated *before* F_1 .

Note that in the case of $i_k = i_l$ and $j_k = j_l$, i.e., the extreme case of compounded feedback spanning the same set of nodes, we only need to evaluate a single feedback loop, however, with feedback constraint $\varphi_k \wedge \varphi_l$. For instance, assuming in Figure 10a that φ_2 started behind the second server (like φ_1), we would have that (using the distributivity from Remark 2)

$$A'_1 = A_1 \wedge (D_2 \otimes \varphi_1) \wedge (D_2 \otimes \varphi_2) = A_1 \wedge D_2 \otimes (\varphi_1 \wedge \varphi_2).$$

4.3 Order of Evaluation in the Feedback Tandem

It is left to discuss in which order the closed-loop systems have to be transformed into open-loop systems to deal with the complex feedback structure of a general tandem as in Figure 7. In fact, the lexicographic order defined in Section 4.2 is instrumental for that purpose. As we have observed for all interdependence types, F_2 had to be evaluated before F_1 , i.e., we need to proceed in reverse lexicographic order to transform the canonical cases into open-loop systems.

This directly generalizes for the feedback tandem. Given a network with k feedback constraints in lexicographic order, the last feedback constraint F_k does not depend on any other constraint F_l , $l < k$. More specifically, since $i_k \geq i_l$, F_l cannot overlap F_k ; also, in case that $i_k = i_l$, then $j_k \leq j_l$ and thus F_l is not contained in F_k . At most we could have the special case of a compounded interdependency where $i_k = i_l$ and $j_k = j_l$. Yet, as discussed at the end of the previous subsection, we can combine them into a single feedback constraint. Hence, we first open the loop for F_k , then apply the same reasoning for the remaining feedback constraints until we have reached a completely open-loop tandem network.

We point out that even more general feedback networks than a tandem could be analyzed, based on the evaluation precedence rules as we obtain them from the different interdependence types. We can build a (directed) precedence graph with the feedback constraints as vertices; determining the interdependence type between two constraints F_i and F_j would then govern whether an edge between vertex i and j exists and its direction. If this precedence graph had no cycles, i.e. it is a directed acyclic graph, then we could perform a topological sort to find a feasible evaluation order of the feedback loops. The cyclic case we leave for future work as it is out of scope for this paper.

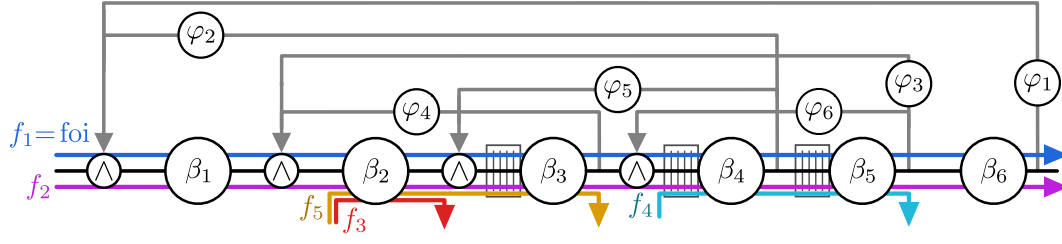
4.4 Network Analysis

In principle, after we have dealt with all feedback loops and have arrived at an open-loop system, we *seem* to face a classical NC problem: We consider f_1 to be the flow of interest (foi), and f_2, \dots, f_m to be cross flows interfering with the flow of interest at different sub-paths of the tandem (see again Figure 7, especially the lower part); each flow f_j is constrained by a maximal arrival curve $\bar{\alpha}_j$. Yet, for the servers S_i , classically we would need to assume strict service curves β_i in order to make use of existing network analyses like, e.g. PMOO [32,33] (the only exception being FIFO scheduling). However, the transformed service curves in the open-loop tandem are not strict but just min-plus service curves (see Section 3.2 on why strict service curves are generally not useful in the modelling of feedback systems).

Instead, we make use of recent results in [19] where a residual service curve β_i^{res} is calculated from a min-plus service curve β_i as

$$\beta_i^{\text{res}} = \beta_i - \bar{\alpha}_i^c, \quad (14)$$

where $\bar{\alpha}_i^c$ is the aggregate maximal arrival curve of all cross flows at node i . This is still a min-plus service curve (see also [7, Theorem 7.3]) and can be used to calculate a valid residual service curve for *any* scheduling policy. Yet, in general, it is partially negative and decreasing and thus not in \mathcal{F}_0^\uparrow . Therefore, we cannot apply Theorem 10 to compute performance bounds anymore. However, in [19] we find a replacement for Theorem 10 when a service curve can be partially negative, enabled by the assumption of a minimal arrival curve $\underline{\alpha}$ for the foi:



■ **Figure 11** Tandem network with several interdependent feedback structures and interfering cross flows.

► **Theorem 19.** *Let an arrival process A traverse a system \mathcal{S} . Further, let the arrivals be constrained by maximal arrival curve $\bar{\alpha} \in \mathcal{F}_0^\uparrow$, and minimal arrival curve $\underline{\alpha} \in \mathcal{F}_0^\uparrow$, and let the system offer a min-plus service curve $\beta \in \mathcal{F}$, with $\xi := \beta_\downarrow$ its lower non-decreasing closure. The backlog $q(t)$ satisfies for all t*

$$q(t) \leq v(\bar{\alpha}, \xi) \wedge \sup_{s \geq 0} \{\bar{\alpha}(s)\}.$$

The virtual delay $d(t)$ satisfies for all $t \geq 0$

$$d(t) \leq \max\{z(\underline{\alpha}, \xi), h(\bar{\alpha}, \xi)\}.$$

Here, the z -distance is defined as $z(\underline{\alpha}, \xi) := \inf\{\tau \geq 0 \mid \underline{\alpha} \otimes \xi(\tau) \geq 0\}$. Consequently, we can make use of network analysis algorithms like PMOO again, despite not having a strict service curve, also when scheduling is not just FIFO. This makes feedback systems in general an interesting application for the new results. In fact, in [19] feedback systems are an application example, yet only a very simple instance with just a single server and two competing flows is treated. The following numerical experiments use our new insights in much more complex feedback systems.

5 Numerical Experiments

By means of numerical experiments we illustrate the application of the proposed method to deal with complex feedback systems as well as provide some quantitative comparison against state-of-the-art techniques. We consider the sample feedback tandem network in Figure 11. Here, we have two application types of feedback constraints: F_4 - F_6 are bounded buffers requiring a feedback signalling to prevent packet loss; F_2 and F_3 are due to local window flow control mechanisms, and F_1 is due to an end-to-end flow control in order not to overwhelm the final receiver. With these feedback applications we cover all types of interdependence. All feedback curves are modelled by $\varphi_l = I_{u_l}$, where I_{u_l} is defined as in Equation (4). As such they all signal a fixed upper bound to preceding nodes that must not be exceeded by incoming traffic. This fixed upper bound is either the size of a finite buffer or the amount of unacknowledged packets that can be in transit at a time.

First, we calculate delay bounds using a state-of-the-art FIFO analysis for feedback systems and compare it to the analysis of a Static Priority (SP) scheduling for sub-flows of our flow of interest. The latter is only possible with our new method. Afterwards, we determine the optimal feedback size u_l for each F_l to avoid any adverse effects of the feedback constraints on the delay analysis. Finally, under the assumption of a limited total budget for all feedback constraints, we compare different strategies to allocate the budget to each of the feedback constraints.



■ **Figure 12** Transformed open-loop system of the tandem network in Figure 11.

5.1 System Modeling and Delay Analysis for FIFO and SP Scheduling

In the following, we have five flows f_1, \dots, f_5 , where f_1 is the flow of interest. We also assume that f_1 consists of four sub-flows f_1^1, \dots, f_1^4 with different priorities, where f_1^4 has the highest priority and f_1^1 the lowest. Application-wise, we interpret the highest priority as absolutely time-critical, for instance, sensor data, while f_1^1 represents Best-Effort (BE) traffic.

In a first step, we derive transformed service curves $\hat{\beta}_1, \hat{\beta}_2, \hat{\beta}_3, \hat{\beta}_4$ that turn the closed-loop system from Figure 11 into an open-loop system. To this end, the feedback constraints are evaluated in the reverse lexicographic order $F_6 > F_5 > F_4 > F_3 > F_2 > F_1$. Thus,

$$\hat{\beta}_4 = \beta_4 \otimes (\beta_4 \otimes \beta_5 \otimes \varphi_6)^* = \beta_4 \otimes \varphi'_6,$$

with $\varphi'_6 = (\beta_4 \otimes \beta_5 \otimes \varphi_6)^*$,

$$\hat{\beta}_3 = \beta_3 \otimes (\beta_3 \otimes \beta_4 \otimes \varphi_5)^* \otimes \varphi'_6 = \beta_3 \otimes \varphi'_5 \otimes \varphi'_6,$$

with $\varphi'_5 = (\beta_3 \otimes \beta_4 \otimes \varphi_5)^*$,

$$\hat{\beta}_2 = \beta_2 \otimes (\beta_2 \otimes \beta_3 \otimes \beta_4 \otimes \beta_5 \otimes \varphi_3)^* \otimes (\beta_2 \otimes \beta_3 \otimes \varphi_4)^* \otimes \varphi'_5 \otimes \varphi'_6 = \beta_2 \otimes \varphi'_3 \otimes \varphi'_4 \otimes \varphi'_5 \otimes \varphi'_6,$$

with $\varphi'_3 = (\beta_2 \otimes \beta_3 \otimes \beta_4 \otimes \beta_5 \otimes \varphi_3)^*$ and $\varphi'_4 = (\beta_2 \otimes \beta_3 \otimes \varphi_4)^*$,

$$\begin{aligned} \hat{\beta}_1 &= \beta_1 \otimes \left(\bigotimes_{j=1}^6 \beta_j \otimes \varphi_1 \right)^* \otimes (\beta_1 \otimes \beta_2 \otimes \beta_3 \otimes \beta_4 \otimes \varphi_2)^* \otimes \varphi'_3 \otimes \varphi'_4 \otimes \varphi'_5 \otimes \varphi'_6 \\ &= \beta_1 \otimes \varphi'_1 \otimes \varphi'_2 \otimes \varphi'_3 \otimes \varphi'_4 \otimes \varphi'_5 \otimes \varphi'_6, \end{aligned}$$

with $\varphi'_1 = \left(\bigotimes_{j=1}^6 \beta_j \otimes \varphi_1 \right)^*$ and $\varphi'_2 = (\beta_1 \otimes \beta_2 \otimes \beta_3 \otimes \beta_4 \otimes \varphi_2)^*$.

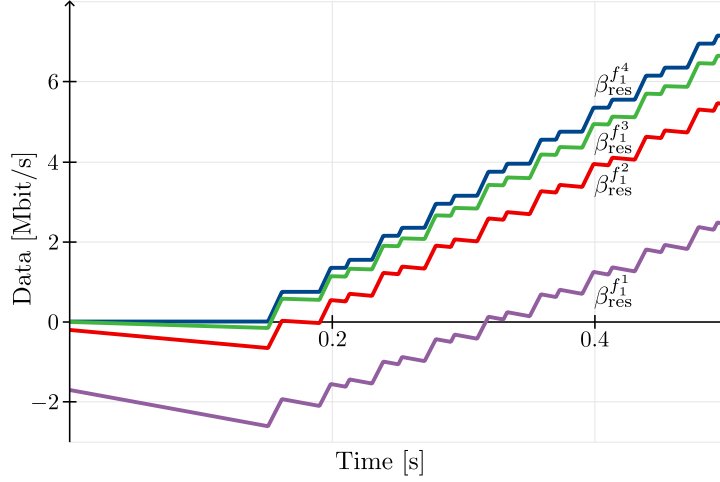
This takes us to the open-loop network in Figure 12, where β_1, \dots, β_4 are replaced by their transformed service curves $\hat{\beta}_1, \dots, \hat{\beta}_4$. We observe that there are *no* feedback constraints left in the open-loop system, but the respective feedback properties are effectively contained within the transformed service curves.

Next, we calculate the FIFO residual end-to-end service curve for our flow of interest in the open-loop system. This is done using the PMOO analysis following the principle to always concatenate as many nodes as possible before computing residual service curves [32, 33]. For instance, in our case we first compute the concatenation of node 4 and 5 by convoluting the respective service curves $\hat{\beta}_4$ and β_5 , and then calculate the residual service curve with respect to cross flow f_4 . For the calculation of the residual service curves we apply the well-known FIFO result [7, Theorem 7.5]

$$\beta_{\text{FIFO}}^\theta(t) := \begin{cases} [\beta(t) - \bar{\alpha}^c(t - \theta)]^+, & \text{if } t > \theta, \\ 0, & \text{otherwise,} \end{cases} \quad (15)$$

where we choose $\theta = \sum_R b_2^i$. This choice for θ minimizes the length of the interval $I = [0, s]$ over which $\beta_{\text{FIFO}}^\theta(t) = 0, \forall t \in I$.

This choice for θ minimizes the latency (period for which it is 0) of the residual service curve under token-bucket arrival and rate-latency service curves [6].



■ **Figure 13** Residual service curves for the SP scheduling analysis.

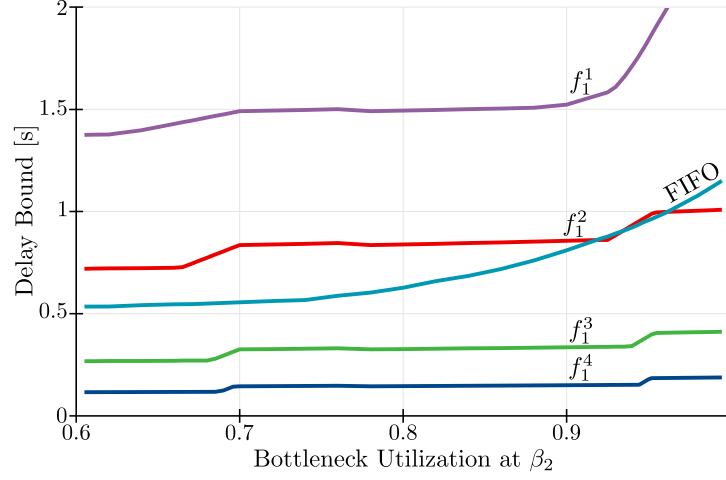
■ **Table 1** Default parameter set for the numerical experiments.

Parameter		Value
Service curve:	R_i	100 Mbit/s
	T_i	0.01 s
Feedback:	φ_l	I_2
Cross flows:	r_2^i	10 Mbit/s
	b_2^i	1 Mbit
Flow of interest:	\bar{r}_1, r_1	10 Mbit/s, 5 Mbit/s
	$\bar{b}_1, \underline{T}_1$	9.71 Mbit(= $\sum b_1^i$), 0.1 s
Sub-flows:	\bar{r}_1^i, r_1^i	$\bar{r}_1 \cdot \frac{4-i+1}{10}, r_1 \cdot \frac{4-i+1}{10}$
	\underline{T}_1^i	$\underline{T}_1 \cdot \frac{4-i+1}{10}$
	$\{b_1^4, b_1^3, b_1^2, b_1^1\}$	$\{0.01, 0.2, 1.5, 8\}$ Mbit

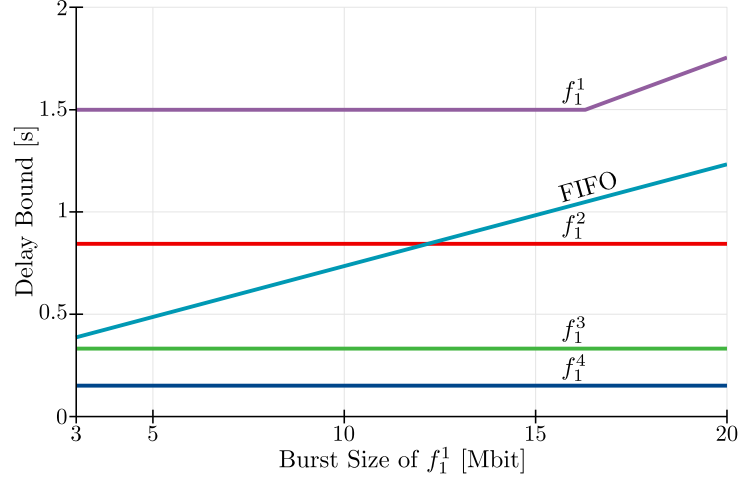
For the FIFO analysis of f_1 , we can now directly calculate performance bounds based on Theorem 10. For the SP analysis of the sub-flows of f_1 , we use Equation (14) to calculate their respective residual service curves, taking into account sub-flows with higher priorities as crossflows. We note that, since we assume β_i to be min-plus service curves, we cannot make use of the residual service curve calculation in [7, Theorem 7.6], as this requires β_i to be a strict service curve. Rather, we use Theorem 19 to calculate the sub-flow specific delay bounds.

We point out that the residual service curves calculated using Equation (15) are always positive and non-decreasing, while the residual curves calculated with Equation (14) are partially negative and decreasing. This is also shown in Figure 13.

The parameters used in our experiments are given in Table 1. Here, $\varphi_l = I_2$ means that each feedback constraint has a fixed size $u = 2$ Mbit. Also note that the burst of the maximal arrival curve of the foi \bar{b}_1 is the sum of the bursts of its sub-flows b_1^4, b_1^3, b_1^2 and b_1^1 , as it holds that the sum of token-bucket arrival curves is again a token-bucket arrival curve for the aggregate. All experiments are conducted using the network calculus library Nancy [39]. We examine two different factors: bottleneck utilization and burstiness of BE traffic f_1^1 . First,



■ **Figure 14** Delay bounds under varying bottleneck utilizations for FIFO and SP analyses.



■ **Figure 15** Delay bounds under varying burst sizes of f_1^1 for FIFO and SP analyses.

we compare the delay bounds for the FIFO and SP analyses against the utilization of the bottleneck node S_2 . The results are illustrated in Figure 14. We see that the delay bounds for the FIFO analysis lie between the SP bounds of f_1^2 and f_1^3 . This exhibits the inherent issue of FIFO scheduling, not being able to protect time-critical traffic from BE. We remark that, except for f_1^4 , all SP delay bounds were calculated using the z -value of Theorem 19, explaining the constant behavior of the delay bounds for some periods as the z -distance does not react directly to a decreasing bottleneck capacity. On a similar note, due to the residual service curve shapes as shown in Figure 13, we see several utilizations where the bounds are increasing sharply. This stems from the fact that changing the utilization also affects the creation and periods of these functions, resulting in fluctuations of the bounds.

Next, we explore the effects of increasing the burst size of the BE traffic f_1^1 . We use the default parameter set from Table 1, but let b_1^1 run from 3 Mbit to 20 Mbit, instead. The results are illustrated in Figure 15. This scenario reinforces what we discussed in the previous paragraph. As the FIFO analysis is an aggregate analysis, increasing the burst size of a

lower priority class also increases the delay bounds for more time-critical traffic. For the SP analysis, we do not observe this behavior other than for flow f_1^1 , as SP isolates the other flows from the “bad behavior” of f_1^1 and we can still calculate good per-flow delay bounds.

5.2 Optimal Feedback to Avoid Adverse Feedback Effects

We have seen how to deal with existing feedback mechanisms in a complex feedback system. However, depending on the application, it may not be desirable to have adverse effects from feedback control on the performance bounds of the system, at all. From a system design perspective, we can try to achieve this by providing enough resources, such as buffer space, for instance. But how do we determine the required resources that are needed in the system? We briefly restate the previously presented result from Section 3.1. For a feedback curve $\varphi = I_u$, and for a single feedback constraint, we obtain the optimal u such that $\hat{\beta} = \beta$ as

$$u^{\text{opt}} = (\beta \circ \beta) \circ \beta(0). \quad (16)$$

Indeed, even with more complex feedback structures, Equation (16) can still be applied. To that end, we make use of the same (lexicographic) order of feedback resolutions as discussed in Section 4.3 and proceed as follows: Assuming we have k feedback loops in the network, we calculate the optimal window size u_k^{opt} for feedback loop $F_k = (i_k, j_k, \varphi_k)$ and transform this closed loop into an open loop system with $\hat{\beta}_{i_k} = \beta_{i_k}$. Next, we can apply Equation (16) again to obtain u_{k-1}^{opt} and transform F_{k-1} . We proceed in this manner for all remaining feedback loops until we have obtained a transformed service curve for all existing closed-loop feedback constraints. Essentially, since we only consider a single feedback constraint at a time, we can iteratively apply Equation (16). This procedure results in a network that has no adverse effects on performance bounds from feedback constraints. Indeed, we can interpret this as the minimal cost of the system (in terms of resource requirements) at which we have no adverse effects of the feedback, and thus view this as *optimal* feedback. We can further calculate performance bounds for the system and confirm whether the system still meets existing timing requirements.

Applying this procedure to the example network, we obtain the optimal u_l^{opt} as $u_1^{\text{opt}} = 6, u_2^{\text{opt}} = u_3^{\text{opt}} = 4, u_4^{\text{opt}} = u_5^{\text{opt}} = u_6^{\text{opt}} = 2$. Comparing the optimal u_l values to the ones we (arbitrarily) picked in Section 5.1, we realize that the feedback constraints spanning more nodes (F_1, F_2, F_3) are under-provisioned, while F_4 to F_6 were already given their optimal values.

5.3 Quantitative Analysis of Fixed Budget Allocation

While calculating the optimal feedback is possible, network design usually has inherent restrictions on the budget that is available to dimension buffers. As a consequence, a required resource budget $\eta_{\text{opt}} = \sum u_l^{\text{opt}}$ is usually higher than an available resource budget η_a . In the following, we study different strategies for resource allocations for the feedback constraints and compare them to each other and the optimal allocation from the previous subsection with respect to delay bounds.

5.3.1 Uniform Allocation

We divide the available budget η_a by the number of feedback constraints in the network and assign the result to each φ_l . This is exactly the allocation that was used in Section 5.1. This may be a sensible allocation strategy when there is not enough information about the network to use one of the following strategies, as they require knowledge of η_{opt} and the optimal values for the feedback constraints.

■ **Table 2** Delay bounds for FIFO and SP analyses under varying available budgets and allocation schemes.

Available Budget		90%			70%		
	Opt.	Max-Min	Prop.	Unif.	Max-Min	Prop.	Unif.
FIFO	242 ms	277 ms	270 ms	358 ms	409 ms	366 ms	484 ms
SP f_1^4	104 ms	104 ms	107 ms	116 ms	121 ms	124 ms	125 ms
SP f_1^3	238 ms	238 ms	245 ms	263 ms	273 ms	279 ms	279 ms
SP f_1^2	655 ms	655 ms	669 ms	705 ms	725 ms	737 ms	738 ms
BE	1310 ms	1310 ms	1324 ms	1360 ms	1380 ms	1392 ms	1438 ms

50%			
	Max-Min	Prop.	Uniform
FIFO	963 ms	644 ms	963 ms
SP f_1^4	190 ms	162 ms	190 ms
SP f_1^3	411 ms	354 ms	411 ms
SP f_1^2	1001 ms	888 ms	1001 ms
BE	1857 ms	1543 ms	1857 ms

5.3.2 Proportional Allocation

Under the assumption that we know η_{opt} and the optimal u_l^{opt} , and have an available budget η_a , we can make an allocation that is proportional to the optimal allocation by setting $u_l = u_l^{\text{opt}} \cdot \frac{\eta_a}{\eta_{\text{opt}}}$.

5.3.3 Max-Min Fair Allocation

We again assume that we have knowledge of η_{opt} and the optimal u_l^{opt} . We make use of the water-filling algorithm from [25], which results in a max-min fair allocation. Here, we fill all constraints equally until we reach the optimal value for some F_l . We remove all such F_l from the set, then repeat the previous step until the available budget η_a is exhausted (or all optimal values u_l^{opt} are reached).

5.3.4 Evaluation

We use the same parameter set as given in Table 1, as well as $\eta_{\text{opt}} = 20$ with u_l^{opt} as $u_1^{\text{opt}} = 6, u_2^{\text{opt}} = u_3^{\text{opt}} = 4, u_4^{\text{opt}} = u_5^{\text{opt}} = u_6^{\text{opt}} = 2$ as calculated in Section 5.2. We set η_a to be either 90%, 70% or 50% of η_{opt} and determine the resulting delay bounds for the three allocation schemes for the FIFO and SP analyses. The results are given in Table 2, where the most accurate delay bound for each analysis and proportion is marked in gray. We make some interesting observations regarding the different allocations. We find that the max-min allocation provides the most accurate bounds for the SP analysis for 70 and 90% of η_{opt} . If η_a is only 50% of η_{opt} , the allocation degenerates into the uniform allocation, as we have already used the whole budget during the first iteration of the water-filling algorithm. For the FIFO analysis, the proportional allocation provides the most accurate delay bounds across all three budgets. Expectedly, the uniform allocation provides the most inaccurate delay bounds for both analyses, as it does not take into account any knowledge about the system under study.

6 Conclusion

In this paper, we have tackled the issue of analyzing per-flow timing characteristics in networked systems with complex feedback structures for scheduling policies other than FIFO. We have generalized existing results for single feedback constraints and shown how to deal with different interdependence types between feedback constraints. We have presented a method to determine the order in which closed-loop systems can be transformed into open-loop systems for arbitrary tandem networks, as well as discussed how more general network topologies can be analyzed. In a numerical experiment, we have shown how the new approach performs in comparison to a FIFO analysis and showed that it is now possible to obtain per-flow delay bounds for non-FIFO scheduling policies. We showed how the optimal window size for a feedback constraint can be iteratively applied in the case of multiple interdependent feedback constraints. We also explored different resource allocation schemes for the feedback constraints to minimize delay bounds. Here, we have seen that the optimal scheme depends on the scheduling policy.

References

- 1 Rajeev Agrawal, Rene L. Cruz, Clayton Okino, and Rajendran Rajan. Performance bounds for flow control protocols. *IEEE/ACM Trans. Netw.*, 7(3):310–323, 1999. doi:10.1109/90.779197.
- 2 Rajeev Agrawal and Rajendran Rajan. *Performance bounds for guaranteed and adaptive services*, volume 20649. Citeseer, 1996.
- 3 John Baillieul and Panos J. Antsaklis. Control and communication challenges in networked real-time systems. *Proc. IEEE*, 95(1):9–28, 2007. doi:10.1109/JPROC.2006.887290.
- 4 Mohamed Bakhouya, Suboh A. Suboh, Jaafar Gaber, and Tarek A. El-Ghazawi. Analytical modeling and evaluation of on-chip interconnects using network calculus. In *Third International Symposium on Networks-on-Chips, NOCS 2009, May 10-13 2009, La Jolla, CA, USA. Proceedings*, pages 74–79. IEEE, IEEE Computer Society, 2009. doi:10.1109/NOCS.2009.5071447.
- 5 Amit Bose, Xiaoyue Jiang, Bin Liu, and Gang Li. Analysis of manufacturing blocking systems with network calculus. *Perform. Evaluation*, 63(12):1216–1234, 2006. doi:10.1016/j.peva.2005.12.001.
- 6 Jean-Yves Le Boudec and Patrick Thiran. *Network Calculus: A Theory of Deterministic Queuing Systems for the Internet*, volume 2050 of *Lecture Notes in Computer Science*. Springer, 2001. doi:10.1007/3-540-45318-0.
- 7 Anne Bouillard, Marc Boyer, and Euriell Le Corronc. *Deterministic Network Calculus: From Theory to Practical Implementation*. John Wiley & Sons, 2018.
- 8 Anne Bouillard, Linh T. X. Phan, and Samarjit Chakraborty. Lightweight modeling of complex state dependencies in stream processing systems. In *15th IEEE Real-Time and Embedded Technology and Applications Symposium, RTAS 2009, San Francisco, CA, USA, 13-16 April 2009*, pages 195–204. IEEE, IEEE Computer Society, 2009. doi:10.1109/RTAS.2009.27.
- 9 Marc Boyer, Amaury Graillat, Benoît Dupont de Dinechin, and Jörn Migge. Bounding the delays of the MPPA network-on-chip with network calculus: Models and benchmarks. *Perform. Evaluation*, 143:102–124, 2020. doi:10.1016/j.peva.2020.102124.
- 10 Alexandre Brandwajn and Yung-Li Lily Jow. An approximation method for tandem queues with blocking. *Oper. Res.*, 36(1):73–83, 1988. doi:10.1287/opre.36.1.73.
- 11 John A. Buzacott and J. George Shanthikumar. *Stochastic models of manufacturing systems*. Prentice Hall Englewood Cliffs, 1993.
- 12 Cheng-Shang Chang. *Performance Guarantees in Communication Networks*. Telecommunication Networks and Computer Systems. Springer, 2000. doi:10.1007/978-1-4471-0459-9.

- 13 Martijn Coenen, Srinivasan Murali, Andrei Radulescu, Kees Goossens, and Giovanni De Micheli. A buffer-sizing algorithm for networks on chip using TDMA and credit-based end-to-end flow control. In Reinaldo A. Bergamaschi and Kiyoun Choi, editors, *Proceedings of the 4th International Conference on Hardware/Software Codesign and System Synthesis, CODES+ISSS 2006, Seoul, Korea, October 22-25, 2006*, pages 130–135. ACM, 2006. doi:10.1145/1176254.1176287.
- 14 Rene L. Cruz. A calculus for network delay, part I: network elements in isolation. *IEEE Trans. Inf. Theory*, 37(1):114–131, 1991. doi:10.1109/18.61109.
- 15 Rene L. Cruz. A calculus for network delay, part II: network analysis. *IEEE Trans. Inf. Theory*, 37(1):132–141, 1991. doi:10.1109/18.61110.
- 16 Rene L Cruz and CM Okino. Service guarantees for window flow control. In *Proceedings of the Annual Allerton Conference on Communication Control and Computing*, volume 34, pages 10–21. Citeseer, 1996.
- 17 Sahar Foroutan, Yvain Thonnart, Richard Hersemeule, and Ahmed Jerraya. Analytical computation of packet latency in a 2d-mesh noc. In *Joint IEEE North-East Workshop on Circuits and Systems and TAISA Conference*, pages 1–4. IEEE, 2009.
- 18 Wei-Jing Guan, Wei Kang Tsai, and Douglas M. Blough. An analytical model for wormhole routing in multicomputer interconnection networks. In *The Seventh International Parallel Processing Symposium, Proceedings, Newport Beach, California, USA, April 13-16, 1993*, pages 650–654. IEEE, IEEE Computer Society, 1993. doi:10.1109/IPPS.1993.262804.
- 19 Anja Hamscher, Vlad-Cristian Constantin, and Jens B. Schmitt. Extending network calculus to deal with min-plus service curves in multiple flow scenarios. In *30th IEEE Real-Time and Embedded Technology and Applications Symposium, RTAS 2024, Hong Kong, May 13-16, 2024*, pages 95–107. IEEE, IEEE, 2024. doi:10.1109/RTAS61025.2024.00016.
- 20 Jingcao Hu, Ümit Y. Ogras, and Radu Marculescu. System-level buffer allocation for application-specific networks-on-chip router design. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.*, 25(12):2919–2933, 2006. doi:10.1109/TCAD.2006.882474.
- 21 Manjunath Kamath and N. Viswanadham. Applications of petri net based models in the modelling and analysis of flexible manufacturing systems. In *Proceedings of the 1986 IEEE International Conference on Robotics and Automation, San Francisco, California, USA, April 7-10, 1986*, volume 3, pages 312–317. IEEE, IEEE, 1986. doi:10.1109/ROBOT.1986.1087700.
- 22 Abbas Eslami Kiasari, Zhonghai Lu, and Axel Jantsch. An analytical latency model for networks-on-chip. *IEEE Trans. Very Large Scale Integr. Syst.*, 21(1):113–123, 2013. doi:10.1109/TVLSI.2011.2178620.
- 23 Y. Koren, U. Heisel, F. Jovane, T. Moriwaki, G. Pritschow, G. Ulsoy, and H. Van Brussel. Reconfigurable manufacturing systems. *CIRP Annals*, 48(2):527–540, 1999.
- 24 Shashi Kumar, Axel Jantsch, Mikael Millberg, Johnny Öberg, Juha-Pekka Soininen, Martti Forsell, Kari Tiensyrjä, and Ahmed Hemani. A network on chip architecture and design methodology. In *2002 IEEE Computer Society Annual Symposium on VLSI (ISVLSI 2002), 25-26 April 2002, Pittsburgh, PA, USA*, pages 117–124. IEEE, IEEE Computer Society, 2002. doi:10.1109/ISVLSI.2002.1016885.
- 25 Jean-Yves Le Boudec. Rate adaptation, congestion control and fairness: A tutorial. *Web page*, November, 4, 2005.
- 26 Zhonghai Lu, Mikael Millberg, Axel Jantsch, Alistair C. Bruce, Pieter van der Wolf, and Tomas Henriksson. Flow regulation for on-chip communication. In Luca Benini, Giovanni De Micheli, Bashir M. Al-Hashimi, and Wolfgang Müller, editors, *Design, Automation and Test in Europe, DATE 2009, Nice, France, April 20-24, 2009*, pages 578–581. IEEE, IEEE, 2009. doi:10.1109/DATE.2009.5090731.
- 27 Milos Panic, Carles Hernández, Eduardo Quiñones, Jaume Abella, and Francisco J. Cazorla. Modeling high-performance wormhole nocs for critical real-time embedded systems. In *2016 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS), Vienna, Austria, April 11-14, 2016*, pages 267–278. IEEE, IEEE Computer Society, 2016. doi:10.1109/RTAS.2016.7461342.

- 28 HT Papadopolous, Cathal Heavey, and Jimmie Browne. *Queueing theory in manufacturing systems analysis and design*. Chapman and Hall, 1993.
- 29 Chrissoleon T. Papadopoulos, Jingshan Li, and Michael E. J. O’Kelly. A classification and review of timed markov models of manufacturing systems. *Computers & Industrial Engineering*, 128:219–244, 2019. doi:10.1016/j.cie.2018.12.019.
- 30 Yue Qian, Zhonghai Lu, and Wenhua Dou. Analysis of worst-case delay bounds for best-effort communication in wormhole networks on chip. In *Third International Symposium on Networks-on-Chips, NOCS 2009, May 10-13 2009, La Jolla, CA, USA. Proceedings*, pages 44–53. IEEE, IEEE Computer Society, 2009. doi:10.1109/NOCS.2009.5071444.
- 31 Pierre Roux, Sophie Quinton, and Marc Boyer. A formal link between response time analysis and network calculus. In Martina Maggio, editor, *34th Euromicro Conference on Real-Time Systems, ECRTS 2022, July 5-8, 2022, Modena, Italy*, volume 231 of *LIPICs*, pages 5:1–5:22, Modene, Italy, July 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPICs.ECRTS.2022.5.
- 32 Jens B. Schmitt, Frank A. Zdarsky, and Markus Fidler. Delay bounds under arbitrary multiplexing: When network calculus leaves you in the lurch. In *INFOCOM 2008. 27th IEEE International Conference on Computer Communications, Joint Conference of the IEEE Computer and Communications Societies, 13-18 April 2008, Phoenix, AZ, USA*, pages 1669–1677. IEEE, IEEE, 2008. doi:10.1109/INFOCOM.2008.228.
- 33 Jens B. Schmitt, Frank A. Zdarsky, and Ivan Martinovic. Improving performance bounds in feed-forward networks by paying multiplexing only once. In Falko Bause and Peter Buchholz, editors, *Proceedings 14th GI/ITG Conference on Measurement, Modelling and Evaluation of Computer and Communication Systems (MMB 2008), March 31 - April 2, 2008, Dortmund, Germany*, pages 13–28. VDE, VDE Verlag, 2008.
- 34 Zheng Shi and Alan Burns. Real-time communication analysis for on-chip networks with wormhole switching. In *Second International Symposium on Networks-on-Chips, NOCS 2008, 5-6 April 2008, Newcastle University, UK. Proceedings*, pages 161–170. IEEE, IEEE Computer Society, 2008. doi:10.1109/NOCS.2008.11.
- 35 Sundararajan Sriram and Shuvra S Bhattacharyya. *Embedded Multiprocessors: Scheduling and Synchronization, Second Edition*. CRC press, 2009.
- 36 Lothar Thiele, Samarjit Chakraborty, and Martin Naedele. Real-time calculus for scheduling hard real-time systems. In *IEEE International Symposium on Circuits and Systems, ISCAS 2000, Emerging Technologies for the 21st Century, Geneva, Switzerland, 28-31 May 2000, Proceedings*, volume 4, pages 101–104. IEEE, IEEE, 2000. doi:10.1109/ISCAS.2000.858698.
- 37 Lothar Thiele and Nikolay Stoimenov. Modular performance analysis of cyclic dataflow graphs. In Samarjit Chakraborty and Nicolas Halbwachs, editors, *Proceedings of the 9th ACM & IEEE International conference on Embedded software, EMSOFT 2009, Grenoble, France, October 12-16, 2009*, pages 127–136. ACM, 2009. doi:10.1145/1629335.1629353.
- 38 Fatih Tüysüz and Cengiz Kahraman. Modeling a flexible manufacturing cell using stochastic petri nets with fuzzy parameters. *Expert Syst. Appl.*, 37(5):3910–3920, 2010. doi:10.1016/j.eswa.2009.11.026.
- 39 Raffaele Zippo and Giovanni Stea. Nancy: an efficient parallel network calculus library. *SoftwareX*, 19:101178, 2022. doi:10.1016/j.softx.2022.101178.
- 40 Raffaele Zippo and Giovanni Stea. Computationally efficient worst-case analysis of flow-controlled networks with network calculus. *IEEE Transactions on Information Theory*, 69(4):2664–2690, 2023. doi:10.1109/TIT.2023.3244276.