Improving Performance Bounds for Weighted Round-Robin Schedulers under Constrained Cross-Traffic

Vlad-Cristian Constantin, Paul Nikolaus, Jens Schmitt
Distributed Computer Systems (DISCO) Lab
TU Kaiserslautern, Germany
{constantin,nikolaus,jschmitt}@cs.uni-kl.de

***Abstract*—Weighted round robin (WRR) is a simple, efficient packet scheduler providing low latency and fairness by assigning flow weights that define the number of possible packets to be sent consecutively. A variant of WRR that mitigates its tendency to increase burstiness, called interleaved weighted round robin (IWRR), has seen analytical treatment recently [1]; a network calculus approach was used to obtain the best-possible strict service curve. From a different perspective, WRR can also be interpreted as an emulation of an idealized fair scheduler known as generalized processor sharing (GPS). Inspired by profound literature results on the performance analysis of GPS, we show that both, WRR and IWRR, belong to a larger class of fair schedulers called bandwidth-sharing policies. We use this insight to derive new strict service curves for both schedulers that, under the additional assumption of constrained cross-traffic flows, can significantly improve the state-of-the-art results and lead to smaller delay bounds.**

***Index Terms*—Network calculus, weighted round robin, interleaved weighted round robin, generalized processor sharing, bandwidth-sharing policies**

I. Introduction

For a long time, round-robin schedulers have been found appealing for their simplicity and corresponding efficient implementation as well as their inherent fairness [2] (see [3] for an early reference). Weighted round robin (WRR) is a frequently used scheduling algorithm in packet-switched networks as well as in real-time processing systems to provide a different resource allocation among flows (or tasks). In its basic version, sometimes called plain WRR, we (conceptually) have a queue for each flow at a server and service is provided in rounds; in each round, a flow f_i receives the opportunity to send w_i packets consecutively. The term weighted round robin was coined in [4] in the context of ATM (i.e., a network with constant packet sizes), where also some modifications, such as interleaving, were suggested. WRR has been considered intensively in the literature on communication networks: e.g., investigating variants such as multi-class or multi-server WRR [5], [6]; or, in applications such as in the IEEE Standard 802.1Q [7], load balancing of cloud infrastructures [8], [9], or networks on chip (NoC) [10], [11]; and found usage in real-world equipment, e.g., in Ethernet switches [12]. In contrast to another popular round-robin scheduler, deficit round robin (DRR) [13], WRR does not assume that the size of the head-of-the-line packet is known at each queue. Therefore, it is also used in distributed queueing scenarios, for instance, as the uplink scheduler in an IEEE Standard 802.16 network [14].

ISBN 978-3-903176-48-5 © 2022 IFIP

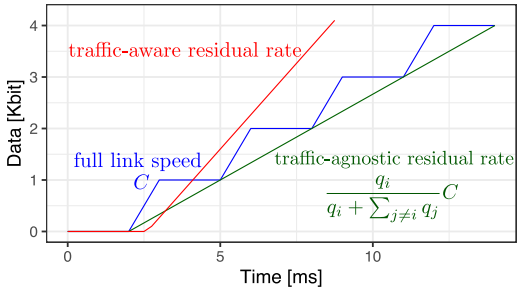


Fig. 1: State-of-the-art service curves for WRR and our service curve under constrained cross-traffic.

WRR has a tendency to increase the burstiness of flows due to several packets being sent back-to-back by the same flow per round. To mitigate this, interleaved weighted round robin (IWRR) introduces *cycles* within a round to disperse the packet transmissions from the same flow. This change is simple and the algorithm's complexity remains favorable (at $\mathcal{O}(1)$ work per packet [13]), though, it makes the mathematical performance analysis more challenging. The performance analysis of IWRR has attracted some recent attention [1], using network calculus.

Network calculus [15]–[19] is a versatile deterministic framework to derive worst-case per-flow performance guarantees. To that end, deterministic constraints on arrivals and service are assumed. These constraints are abstracted by so-called arrival and service curves, respectively. While the former allows us to forego any distribution assumptions on inter-arrival times, the latter comes with the power of scheduling abstraction [20]. The so-called leftover service curve can then directly be used to calculate per-flow performance bounds. Network calculus's level of modularization reduces the problem of improving performance bounds to finding a larger leftover service curve.

Several different leftover service curves for WRR have been provided in the network calculus literature. In [19, p.

200], three service curves are derived for WRR. The first one exploits knowledge on possible packet sequences by introducing so-called *packet curves* [21]. Since we do not make this assumption, we only focus on the latter two. Let us assume for the moment that the server provides a constant rate C. The least data per round for the flow of interest f_i is $q_i \coloneqq w_i l_i^{\min}$ and the most data for a cross-flow $f_{j\neq i}$ is $q_j \coloneqq w_j l_j^{\max}$; here $l_i^{\min}$ and $l_j^{\max}$ denote the minimum and maximum packet size of the respective flow. One leftover service curve is a stair function that closely models the packet scheduler's behavior of alternating between *full link speed* (rate C) and plateaus (rate 0) within a single round [22]; the other leftover service curve has a shifted linear shape (with some initial latency) with *traffic-agnostic residual rate* $\frac{q_i}{q_i+\sum_{j\neq i} q_j} C$ that is just below the stair (illustrated in Figure 1). For IWRR, a leftover service curve for IWRR is derived that dominates all service curves obtained by the WRR analysis [1].

The aforementioned leftover service curves do not take into account constraints on the cross-traffic, even though it is a common case to have such constraints in application scenarios where worst-case performance guarantees are desired. A key observation is that, given such constraints, all cross-flows will not remain backlogged for longer time intervals. In fact, this observation has already been used to improve performance bounds for generalized process sharing (GPS), an idealized fair scheduler that achieves nearly perfect isolation and fairness [23]. It appeared in the seminal work on GPS by Parekh and Gallagher [24] exploiting the *feasible ordering* of flows. Later, this concept was generalized to *feasible partitions* [25], larger classes of arrival curves and service curves [17, pp. 68], [26], [19, pp. 172], and in a recent publication to a larger, practically more relevant class of fair schedulers, called *bandwidth-sharing policies* [27]. Weighted round robin, from a different perspective, can be interpreted as a GPS emulation. As a consequence, we could benefit from the profound GPS results also in the WRR analysis.

In this work, under the assumption of constrained cross-flows, we provide new strict leftover service curves for WRR and IWRR using the network calculus framework. Essentially, these are based on mathematical proofs that WRR and IWRR are bandwidth-sharing policies. The new service curves can lead to significantly better delay bounds compared to the state of the art. The reason for the improvement is that the new service curves' *traffic-aware residual rate* is larger than their traffic-agnostic counterpart (see again Figure 1; of course, details follow below).

The rest of the paper is structured as follows: We provide the necessary background on network calculus in Section II. In Section III, we present the state of the art on WRR and IWRR as well as a class of fair schedulers called bandwidth-sharing policies. In Section IV, we show that WRR and IWRR, respectively, are bandwidth-sharing policies and derive new leftover service curves. We provide numerical evaluations in Section V. Section VI concludes the paper.

II. Network Calculus Background

In this section, we present the necessary network calculus definitions and theorems as we use them throughout this paper.

An *arrival process* (or input function) $A(t)$ of a flow f cumulatively counts the number of work units that arrive at a server in the interval $[0, t)$. We define it as an element of $\mathcal{F}$, the set of all wide-sense increasing functions:

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

Moreover, we use the shorthand notation $A(s,t) \coloneqq A(t) - A(s)$. Similarly, we denote its according *departure process* by $D(t) \in \mathcal{F}$. We assume a system to be *causal*, i.e., no data are created at the system: $A(t) \geq D(t)$ for all $t \geq 0$ and write $D(s,t)$ for $D(t) - D(s)$. Furthermore, we assume all systems to be lossless.

Definition 1 (Virtual Delay)**.** The *virtual delay* of data arriving at a server at time t is the time until this data would be served, assuming FIFO order of service,

$$d(t) \coloneqq \inf \{\tau \geq 0 : A(t) \leq D(t+\tau)\}. \tag{1}$$

In order to provide worst-case performance guarantees, we need upper bounds on arrivals and lower bounds on the service, respectively. We start off by defining arrival curves.

Definition 2 (Arrival Curve)**.** Given an increasing function $\alpha \in \mathcal{F}$. We say that α is an *arrival curve* for an arrival process A if for all $0 \leq s \leq t$

$$A(t) - A(s) \leq \alpha(t-s).$$

The most important example is the *token bucket* arrival curve $\gamma_{r,b}(t) = b + r \cdot t$ for $t \geq 0$. The parameter r denotes the rate and b the burst tolerance [18, p. 7].

For the service we need to introduce different notions. First, we define service curves as they are needed to provide performance bounds. Then, we introduce a stronger notion of so-called strict service curves, that we use for a per-flow analysis in a multi-flow system. Afterwards, we define variable capacity nodes which are mainly used as a technical assumption.

Definition 3 (Service Curve)**.** Consider an arrival process A traversing a server and its according departure process D. The server offers a *(minimum) service curve* β to A if $\beta \in \mathcal{F}$ and for all $t \geq 0$

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

We define a *backlogged period* such that $D(\tau) < A(\tau)$ for all $\tau \in (s,t]$.

Definition 4 (Strict Service Curve)**.** A server is said to offer a *strict service curve* $\beta \in \mathcal{F}$ to a flow if, during any backlogged period $(s,t]$,

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

The most important example of service curves we employ is the *rate-latency* service curve $\beta_{R,T}(t) \coloneqq R \cdot [t-T]^+$, where $[x]^+ \coloneqq \max\{x, 0\}$ denotes the positive part.

Algorithm 1 Weighted Round Robin [19, p. 200]

Input Integer weights $w_1, \ldots, w_n$

while True **do** ▷ A round starts.
  for $i = 1$ **to** n **do**
    $k \leftarrow 1$;
    while not empty(i) **and** $k \leq w_i$ **do**
      send(head(i));
      removeHead(i);
      $k \leftarrow k + 1$;
    end while
  end for
end while ▷ A round finishes.

Algorithm 2 Interleaved Weighted Round Robin [1]

Input Integer weights $w_1, \ldots, w_n$

$w_{\max} \leftarrow \max\{w_1, \ldots, w_n\}$
while True **do** ▷ A round starts.
  for $C = 1$ **to** $w_{\max}$ **do** ▷ A cycle starts.
    for $i = 1$ **to** n **do**
      if not empty(i) **and** $C \leq w_i$ **then**
        send(head(i));
        removeHead(i);
      end if
    end for
  end for ▷ A cycle finishes.
end while ▷ A round finishes.

We define the function $C(s,t) = C(t) - C(s)$ as the cumulative service process of a server for $0 \leq s \leq t$.

Definition 5 (Variable Capacity Node)**.** A server is a *variable capacity node* (VCN) if it offers $\beta \in \mathcal{F}$ such that for *any* $0 \leq s \leq t$

$$C(s,t) \geq \beta(t-s).$$

Typical examples such as constant-rate servers belong to this class. The assumption of a VCN is very mild, as it has been proven that it is equivalent to a strict service curve if the asymptotic growth rate of β is finite [19, pp. 222].

Under these basic concepts, we are able to derive *tight* performance bounds on the delay:

Theorem 6 (Delay Bound)**.** *Assume an arrival process A traversing a server. Further, let the arrivals be constrained by arrival curve α and let the system offer a service curve β. The virtual delay $d(t)$ satisfies for all t*

$$d(t) \leq \sup_{t \geq 0} \{\inf \{d \geq 0 \mid \alpha(t) \leq \beta(t+d)\}\} =: h(\alpha, \beta),$$

where $h(\alpha, \beta)$ is the horizontal deviation *between α and β.*

Tight means that we can create a sample path such that the delay is equal to its delay bound.

Using network calculus, we derive per-flow performance bounds. Throughout the rest of this paper, if not stated otherwise, our flow of interest has the index $i \in \{1, \ldots, n\} =: \mathcal{N}$ and we call the remaining $n-1$ flows cross-traffic.

III. State-of-the-Art on (I)WRR and Opening a Door for Improvement

In this section, we first explain the basic mechanics behind weighted round robin (WRR) and its interleaving variant, IWRR. Next, we present the state-of-the-art for the network calculus analysis of both, WRR and IWRR. At the end of this section, we introduce a general class of fair schedulers called bandwidth-sharing policies and explain how we can leverage from this abstraction.

A. Basics on (I)WRR

We start off with plain WRR: Conceptually, the packets of a flow f_i are queued in its own dedicated queue. A flow receives one service opportunity in each *round* (see Algorithm 1). Each flow is assigned a weight w_i that sets the maximum number of packets that can be served in a single round. Note that, since flow f_i sends all w_i packets consecutively, this may result in a considerable output burstiness for flows with higher weights. The packet sizes are not fixed making the performance analysis of networks with variable packet sizes significantly more challenging.

Mathematically, we assume a service guarantee for the aggregate of all flows in form of a strict service curve or a work-conserving server which is typical in the literature [1], [19], [24]. Obtaining a per-flow service guarantee enables us to provide respective per-flow performance guarantees.

The second WRR variant we consider, interleaved weighted round robin (IWRR) (see Algorithm 2), mitigates plain WRR's typical burstiness by supplementing the rounds with *cycles*, while maintaining the same complexity. Instead of receiving service for a burst of w_i packets in a round, flow f_i can only send one packet per cycle like any other flow which has not yet exhausted its weight allocation in the current round. After w_i rounds flow f_i has to wait until other flows f_j with $w_j > w_i$ finish off the round.

B. Network calculus analysis of (I)WRR

It was shown in the literature that if β is assumed to be a constant-rate server using WRR, then rate-latency leftover service curves $\beta_{R,T}$ can be derived. Several publications determined the rate term R and latency term T under WRR, such as [10], [28], [29] (for a detailed discussion, see [1]). Yet, rate-latency service curves do not yield tight results for WRR since each packet is only served with the traffic-agnostic residual rate [1]. Taking into account that a packet is transmitted at full link speed leads to a more precise model and consequently better service guarantees. Such a leftover service curve, which is obviously not a rate-latency function anymore, is elegantly derived in [19] for WRR (the improvement is depicted in Figure 1). We only state the two leftover service curves that do not require specific knowledge about possible packet sequences (so-called packet curves).

Theorem 7 (Strict Leftover Service Curves for WRR)**.** *Assume n flows arriving at a server performing weighted round robin*

(WRR) with weights $w_1, \dots, w_n$. Let this server offer a strict service curve β to these n flows. We define $q_i \coloneqq w_i l_i^{\min}$ and $Q_i \coloneqq \sum_{j \neq i} w_j l_j^{\max}$.

1. *Then,*

$$\beta^i(t) \coloneqq \gamma_i' \left([\beta(t) - Q_i]^+ \right),$$

is a strict service curve for flow f_i, where we define

$$\gamma_i'(t) \coloneqq \beta_{1,0} \otimes \nu_{q_i, q_i + Q_i}(t),$$

the stair function

$$\nu_{h,P}(t) \coloneqq h \left\lceil \frac{t}{P} \right\rceil \quad \text{for } t \geq 0, \tag{2}$$

and $\beta_{1,0}$ is a constant-rate function with slope 1.

2. *Moreover,*

$$\begin{aligned} \beta^i(t) &\coloneqq \frac{w_i l_i^{\min}}{w_i l_i^{\min} + \sum_{j \neq i} w_j l_j^{\max}} \left[\beta(t) - \sum_{j \neq i} w_j l_j^{\max} \right]^+ \\ &= \frac{q_i}{q_i + Q_i} \left[\beta(t) - Q_i \right]^+ \end{aligned}$$

is a strict service curve for flow f_i. If β is a constant-rate server, then the residual service curve β^i is a rate-latency service curve.

Note that for packets of constant size the delay bound of Theorem 7 has been shown to be tight [1], i.e., a sample path attaining the delay bound was constructed. To be precise, only the first part of the theorem, where each packet receives full link speed during transmission, is actually tight since the second part only provides the traffic-agnostic residual rate. For a constant-rate server with rate C, this rate is equal to $\frac{q_i}{q_i + Q_i} C$.

For IWRR, a leftover service curve has been derived in [1]. Not only do the authors show that exploiting the interleaving in the analysis can significantly improve delay bounds, they also prove that their bound is tight. We only state the leftover service curve for IWRR.

Theorem 8 (Strict Leftover Service Curves for IWRR)**.** *Assume n flows arriving at a server performing interleaved weighted round robin (IWRR) with weights $w_1, \dots, w_n$. Let this server offer a superadditive strict service curve β to these n flows. Then,*

$$\beta^i(t) \coloneqq \gamma_i'' \left(\beta(t) \right),$$

is a strict service curve for flow f_i, where

$$\begin{aligned} \gamma_i''(t) &\coloneqq \beta_{1,0} \otimes U_i(t), \\ U_i(t) &\coloneqq \sum_{k=0}^{w_i - 1} \nu_{l_i^{\min}, L_{\text{tot}}} \left(\left[t - \psi_i \left(k l_i^{\min} \right) \right]^+ \right) \\ L_{\text{tot}} &\coloneqq q_i + Q_i \\ \psi_i(x) &\coloneqq x + \sum_{j \neq i} \Psi_{ij} \left(\left\lfloor \frac{x}{l_i^{\min}} \right\rfloor \right) l_j^{\max} \\ \Psi_{ij}(p) &\coloneqq \left\lfloor \frac{p}{w_i} \right\rfloor w_j + [w_j - w_i]^+ \\ &\quad + \min \left\{ (p \mod w_i) + 1, w_j \right\} \end{aligned} \tag{3}$$

and the stair function $\nu_{h,P}(t)$ is defined in Equation (2) *as well as $q_i = w_i l_i^{\min}$ and $Q_i = \sum_{j \neq i} w_j l_j^{\max}$ again.*

C. The case of constrained cross-traffic

Both theorems, Theorem 7 and 8, do not make any assumptions on the cross-traffic. While this can be seen as a strength, in our work, we argue that by assuming cross-flows to be constrained, we open the door for improvement. This can lead to significantly reduced delay bounds (see our numerical evaluation in Section V). This reduction comes despite the fact that we do not closely model packet transmission at full link speed. However, as we see in the next section, since we consider the maximum of shifted linear functions, we can often still obtain more than just the traffic-agnostic residual rate. The central notion we employ to take knowledge on cross-flows into account is the bandwidth-sharing policy. We start off with its definition.

Definition 9 (Bandwidth-Sharing Policy [27])**.** A server has a *bandwidth-sharing policy* if there exist positive weights $\phi_i > 0, i = 1, \dots, n$ and nonnegative number $H_{ij} \geq 0$, $1 \leq i, j \leq n$ such that for a backlogged period $(s, t]$ of flow f_i it holds that

$$\frac{D_i(s,t)}{\phi_i} \geq \frac{[D_j(s,t) - H_{ij}]^+}{\phi_j}, \quad \text{for all } j \neq i. \tag{4}$$

Note that the bandwidth-sharing policy can be seen as a generalization of the resource allocation under GPS, where the H_{ij} would be 0. We therefore interpret it as a penalty term that is a consequence of emulating the ideal fluid fair sharing of GPS by a real packet scheduler implementation. For example, deficit round robin (DRR) has been show to be a bandwidth-sharing policy [27]. For this class of schedulers, a profound result is given in the literature. It was initially derived for GPS and then extended to schedulers which realize a bandwidth-sharing policy.

Theorem 10 (Strict Leftover Service Curves for Bandwidth-Sharing Policies)**.** *Assume n flows arriving at a server with a bandwidth-sharing policy with positive weights $\phi_i > 0, i = 1, \dots, n$ and a nonnegative penalty term $H_{ij} \geq 0$, $1 \leq i, j \leq n$. Let this server be a VCN that offers a convex β to these n flows. Let $\mathcal{N} = \{1, \dots, n\}$ and assume that each flow f_i is constrained by a concave arrival curve $\alpha_i, i = 1, \dots, n$. Then,*

$$\beta^i(t) = \max_{i \in M \subset \mathcal{N}} \frac{\phi_i}{\sum_{k \in M} \phi_k} \left[\beta(t) - \sum_{k \notin M} \alpha_k(t) - \sum_{k \in M} H_{ik} \right]^+ \tag{5}$$

is a strict service curve for flow f_i.

Proof. See Theorem 1 in [27]. □

The number of possible sets M to optimize over, of course, potentially becomes very large for a high number of flows: we have $2^{|\mathcal{N}|-1}$ subsets of $\mathcal{N}$ containing i. Yet, we point out that any selection of M provides a strict leftover service curve. That means, we can come up with heuristics that avoid a combinatorial explosion by trading efficiency for the accuracy of the calculated bounds.

IV. New Strict Service Curves for (Interleaved) Weighted Round Robin

In this section, we show that WRR as well as IWRR are bandwidth-sharing policies. This insight has the direct consequence of providing new leftover service curves which take into account arrival constraints on the cross-flows.

Theorem 11 (WRR is a Bandwidth-Sharing Policy)**.** *Assume n flows arriving at a server performing weighted round robin (WRR) with weights $w_1, \dots, w_n$. Then, WRR is a bandwidth-sharing policy for flow f_i with*

$$\begin{aligned} \phi_i' &= w_i l_i^{\max} =: q_i, \\ \phi_j' &= w_j l_j^{\max} =: q_j, \quad j \neq i \end{aligned} \tag{6}$$

and

$$H_{ij}' = w_j l_j^{\max} \mathbf{1}_{\{i \neq j\}} = q_j \mathbf{1}_{\{i \neq j\}},$$

where $\mathbf{1}_{\{i\neq j\}} = 1$ for $i \neq j$ and 0, else. In other words, we have

$$\frac{D_i(s,t)}{\phi_i'} \geq \frac{\left[D_j(s,t) - w_j l_j^{\max} \mathbf{1}_{\{i\neq j\}}\right]^+}{\phi_j'} \tag{7}$$

for any $(s,t]$ such that flow f_i is backlogged.

Proof. Following along the lines of [19, pp. 201], we consider a backlogged period of $(s,t]$ of flow f_i and let $p \in \mathbb{N}$ denote the number of completed services of flow f_i in the interval $(s,t]$. Constructing the worst case, we have

$$D_i(s,t) \geq p w_i l_i^{\min}. \tag{8}$$

Moreover, this yields directly an upper bound for $p \in \mathbb{N}$:

$$p \leq \left\lfloor \frac{D_i(s,t)}{w_i l_i^{\min}} \right\rfloor. \tag{9}$$

On the other hand, in this interval,

$$D_j(s,t) \leq (p+1) w_j l_j^{\max}, \quad \forall j \neq i. \tag{10}$$

Summing the inequalities in (8) and (10) yields

$$\begin{aligned} \frac{D_i(s,t)}{w_i} &\overset{(8),(10)}{\geq} \frac{D_j(s,t)}{w_j} + p l_i^{\min} - (p+1) l_j^{\max} \\ &= \frac{D_j(s,t)}{w_j} - p\left(l_j^{\max} - l_i^{\min}\right) - l_j^{\max} \\ &\overset{(9)}{\geq} \frac{D_j(s,t)}{w_j} - \left\lfloor \frac{D_i(s,t)}{w_i l_i^{\min}} \right\rfloor \left(l_j^{\max} - l_i^{\min}\right) - l_j^{\max} \\ &\geq \frac{D_j(s,t)}{w_j} - \frac{D_i(s,t)}{w_i} \cdot \frac{l_j^{\max} - l_i^{\min}}{l_i^{\min}} - l_j^{\max}. \end{aligned}$$

This is equivalent to

$$\frac{D_i(s,t)}{w_i l_i^{\min}} \geq \frac{1}{l_j^{\max}} \left(\frac{D_j(s,t)}{w_j} - l_j^{\max} \right) = \frac{D_j(s,t) - w_j l_j^{\max}}{w_j l_j^{\max}}.$$

Inserting ϕ_i', ϕ_j' as in Equation (6) and using that $D_i(s,t) \geq 0$ for all $0 \leq s \leq t$ finishes the proof. □

We interpret the assignment for $\phi_i', i = 1, \dots, n$ in Equation (6) as the weight corrected to the amount of data per round in the worst case from the perspective of the flow of interest.

Corollary 12 (Strict Leftover Service Curve for WRR)**.** *Assume n flows arriving at a server performing weighted round robin (WRR) with weights $w_1, \dots, w_n$. Let this server be a VCN that offers a convex β to these n flows. Let $\mathcal{N} \coloneqq \{1, \dots, n\}$ and assume that each flow f_i is constrained by a concave arrival curve $\alpha_i, i = 1, \dots, n$. We define*

$$Q_M' \coloneqq \sum_{k \in M \setminus \{i\}} w_k l_k^{\max}.$$

Then,

$$\beta^i(t) = \max_{i \in M \subset \mathcal{N}} \frac{q_i}{q_i + Q_M'} \left[\beta(t) - \sum_{k \notin M} \alpha_k(t) - Q_M' \right]^+ \tag{11}$$

is a strict service curve for flow f_i. In the following, we call the leftover service curve β^i WRR M.

Proof. The corollary is a consequence of WRR being a bandwidth sharing policy (Theorem 11) ($\phi_i', i = 1, \dots, n$ and H_{ik} constructed above) together with Theorem 10 which gives a strict service curve for these policies. □

One can easily show that if β is a constant-rate server, then β^i is a rate-latency service curve for any $i \in M \subset \mathcal{N}$.

Furthermore, note that Corollary 12 can be relaxed in the sense that we actually do not need all cross-flows to be constrained by arrival curves. For these flows f_k we can simply define $\alpha_k(t) = \infty$ and they are going to be an element of M, i.e., they do not increase the sum $\sum_{k \notin M} \alpha_k(t)$ when maximizing over M in Equation (11). The same applies when the long-term arrival rate of f_k, $\lim_{t\to\infty} \frac{\alpha_k(t)}{t}$, is actually larger than the long-term server rate of β. In other words, similar to the state of the art, we are not limited to stable systems.

Let us explain the actual gain compared to the state of the art under token-bucket constrained arrivals and a constant-rate server with rate C. Compared to Theorem 7.2, we have now the possibility to change the latency for $M \subsetneq \mathcal{N}$ ($\sum_{k \notin M} \gamma_{r_k, b_k}(t)$ is decreasing in M, while Q_M' is increasing) and at the same time change the traffic-agnostic residual rate to $\frac{q_i}{q_i + Q_M'} \left(C - \sum_{k \notin M} r_k\right)$. As an obvious consequence, the improvement impact is increased the more the cross-traffic can be constrained. Additionally, Corollary 12 directly recovers Theorem 7.2. However, the relation to the stair function in Theorem 7.1. is not as clear. In our numerical evaluation (Section V), we show that our leftover service curve can outperform the one in Theorem 7.1 and lead to better delay bounds.

Next, we continue with the analysis of interleaved weighted round robin. It mitigates burstiness by employing a simple trick, namely by introducing cycles that prevents flows from sending data consecutively. Yet, these cycles make the mathematical analysis more difficult. In this section, we show, based on some insights presented in [1], that IWRR is also

a bandwidth-sharing policy. To be precise, we show it for two different penalty terms. While one is obtained by exploiting peculiarities of IWRR, the other is the same as for WRR. Therefore, our new strict service curve for IWRR can only be equal or better than the one we obtained for WRR.

Theorem 13 (IWRR is a Bandwidth-Sharing Policy)**.** *Assume n flows arriving at a server performing interleaved weighted round robin (IWRR) with weights $w_1, \dots, w_n$. Then,*

1. IWRR is a bandwidth-sharing policy for flow f_i, where
$$\begin{aligned} \phi_i'' &= w_i l_i^{\min} = q_i, \\ \phi_j'' &= (w_j + w_i)\, l_j^{\max} = q_j + w_i l_j^{\max} \end{aligned} \tag{12}$$
and
$$H_{ij}'' = \left([w_j - w_i]^+ + 1\right) l_j^{\max} \mathbf{1}_{\{i \neq j\}}.$$
In other words, we have
$$\frac{D_i(s,t)}{\phi_i''} \geq \frac{\left[D_j(s,t) - \left([w_j - w_i]^+ + 1\right) l_j^{\max} \mathbf{1}_{\{i \neq j\}}\right]^+}{\phi_j''} \tag{13}$$
for any $(s,t]$ such that flow f_i is backlogged.
2. IWRR is bandwidth-sharing policy with the $\phi_i', \phi_j', H_{ij}'$ from WRR.

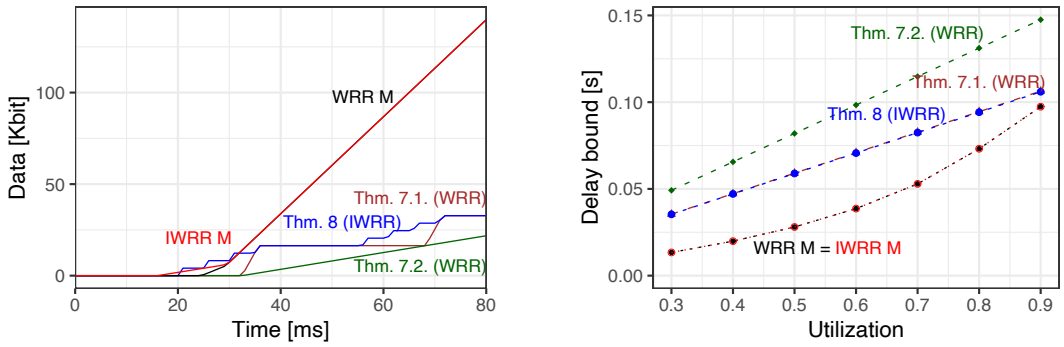
For the sake of conciseness, we provide a full proof of the first part and only a sketch of the second.

Proof of 1. Let $(s,t]$ be a backlogged period of flow f_i. Again, let $p \in \mathbb{N}$ denote the number of completed services of flow f_i in the interval $(s,t]$. By construction, it holds that
$$D_i(s,t) \geq p l_i^{\min}. \tag{14}$$
Note that, in comparison to WRR, we need to adjust the inequality since we have cycles in between rounds. As a direct consequence of Equation (14), we can upper bound $p \in \mathbb{N}$ by
$$p \leq \left\lfloor \frac{D_i(s,t)}{l_i^{\min}} \right\rfloor. \tag{15}$$
Moreover, by combining Lemma 4 and Lemma 6 in [1], we receive
$$D_j(s,t) \leq \Psi_{ij}(p) l_j^{\max}, \quad \forall j \neq i, \tag{16}$$
where is $\Psi_{ij}(p)$ is defined in Equation (3). Summing the inequalities in (14) and (16) yields
$$\begin{aligned} &\frac{D_i(s,t)}{w_i} \\ \geq &\frac{D_j(s,t)}{w_j} - \frac{p}{w_i}\left(l_j^{\max} - l_i^{\min}\right) \\ &- \frac{\left([w_j - w_i]^+ + \min\left\{(p \mod w_i) + 1, w_j\right\}\right) l_j^{\max}}{w_j} \\ \overset{(15)}{\geq} &\frac{D_j(s,t)}{w_j} - \frac{\left\lfloor \frac{D_i(s,t)}{l_i^{\min}} \right\rfloor}{w_i}\left(l_j^{\max} - l_i^{\min}\right) \\ &- \frac{\left([w_j - w_i]^+ + \min\left\{\left\lfloor \frac{D_i(s,t)}{l_i^{\min}} \right\rfloor \mod w_i + 1, w_j\right\}\right) l_j^{\max}}{w_j} \end{aligned} \tag{17}$$
$$\begin{aligned} \geq &\frac{D_j(s,t)}{w_j} - \frac{\left\lfloor \frac{D_i(s,t)}{l_i^{\min}} \right\rfloor}{w_i}\left(l_j^{\max} - l_i^{\min}\right) \\ &- \frac{\left([w_j - w_i]^+ + \left\lfloor \frac{D_i(s,t)}{l_i^{\min}} \right\rfloor + 1\right) l_j^{\max}}{w_j} \\ \geq &\frac{D_j(s,t)}{w_j} - \frac{D_i(s,t)}{w_i} \cdot \frac{1}{l_i^{\min}}\left(l_j^{\max} - l_i^{\min}\right) \\ &- \frac{D_i(s,t)}{w_j} \cdot \frac{l_j^{\max}}{l_i^{\min}} - \frac{\left([w_j - w_i]^+ + 1\right) l_j^{\max}}{w_j}. \end{aligned} \tag{18}$$
Note that in Equation (18), we upper bounded $\min\left\{\left(\left\lfloor \frac{D_i(s,t)}{l_i^{\min}} \right\rfloor \mod w_i\right) + 1, w_j\right\}$ by $\left\lfloor \frac{D_i(s,t)}{l_i^{\min}} \right\rfloor$. Above inequality is equivalent to
$$\begin{aligned} &\frac{D_i(s,t)}{w_i l_i^{\min}} \\ \geq &\frac{w_j}{w_j + w_i} \cdot \frac{D_j(s,t)}{w_j l_j^{\max}} - \frac{w_j}{w_j + w_i} \cdot \frac{\left([w_j - w_i]^+ + 1\right) l_j^{\max}}{w_j l_j^{\max}}. \end{aligned}$$
Replacing w_i and w_j with ϕ_i'' and ϕ_j'', respectively, from Equation (12) yields
$$\frac{D_i(s,t)}{\phi_i''} \geq \frac{D_j(s,t) - \left([w_j - w_i]^+ + 1\right) l_j^{\max}}{\phi_j''}.$$
In the final step, we use again that $D_i(s,t) \geq 0$. □

Sketch of 2. Starting in Equation (17), we could also continue to upper bound $\min\left\{\left(\left\lfloor \frac{D_i(s,t)}{l_i^{\min}} \right\rfloor \mod w_i\right) + 1, w_j\right\}$ by $\min\{w_i, w_j\}$. Applying again Equation (15) and using that
$$[w_j - w_i]^+ + \min\{w_i, w_j\} = w_j$$
gives the result. □

We can now formulate the leftover service curve.

Corollary 14 (Strict Leftover Service Curve for IWRR)**.** *Assume n flows arriving at a server performing interleaved weighted round robin (IWRR) with weights $w_1, \dots, w_n$. Let this server be a VCN that offers a convex β to these n flows. Let $\mathcal{N} := \{1, \dots, n\}$ and assume that each flow f_i is constrained by a concave arrival curve $\alpha_i, i = 1, \dots, n$. We define*
$$Q_M'' := \sum_{k \in M \setminus \{i\}} w_k \left(\left(1 + \frac{w_i}{w_k}\right) \cdot l_k^{\max}\right).$$
and
$$H_{ik}'' = \left([w_k - w_i]^+ + 1\right) l_k^{\max} \mathbf{1}_{\{i \neq k\}}.$$
Then,
$$\beta^i(t) = \max_{i \in M \subset \mathcal{N}} \left\{\max \beta_{1,M}^i, \beta_{2,M}^i\right\} \tag{19}$$
is a strict service curve for flow f_i, where
$$\begin{aligned} &\beta_{1,M}^i(t) \\ = &\frac{q_i}{q_i + Q_M''}\left[\beta(t) - \sum_{k \notin M} \alpha_k(t) - \sum_{k \in M \setminus \{i\}} H_{ik}''\right]^+ \end{aligned} \tag{20}$$



(a) Service curves (utilization $= 0.6$)

(b) Delay bounds

Fig. 2: Parameter set: weights $= \{4, 6, 7, 10\}$, $l^{\min} = \{4096, 3072, 4608, 3072\}$ bits, $l^{\max} = \{8704, 5632, 6656, 8192\}$ bits, burst sizes $= \{30208, 19968, 24576, 27648\}$ bits, arrival rates $= \{0.65, 0.85, 0.95, 0.55\}$ Mbit/s.

and $\beta^i_{2,M}$ is the strict leftover service curve for WRR (Equation (11)*). In the following, we call the leftover service curve β^i* IWRR M.

Proof. Similar to the WRR case, Theorem 13 proves that IWRR is a bandwidth sharing policy for two different penalty terms. Theorem 10 then gives us two different strict leftover service curve. At last, we use that the maximum of strict service curves is again a strict service curve [19, p. 109]. Note that this property is actually already used in the derivation of Theorem 10. □

Again, if β is a constant-rate server, then $\beta_{1,M}$ and $\beta_{2,M}$ are rate-latency service curves for any choice of $i \in M \subset \mathcal{N}$.

V. Numerical Evaluation

In this section, we numerically compare our newly obtained leftover service curves under constrained cross-traffic, WRR M (Corollary 12) and IWRR M (Corollary 14), to the state of the art in Theorem 7 (WRR) and Theorem 8 (IWRR).

We assume all flows f_i to be constrained by a token bucket arrival curve γ_{r_i,b_i}, $i = 1, \ldots, n$. For the service, we always assume a constant server rate $C > 0$ for the aggregate of flows.

In our numerical experiments, we first compare our results against the state of the art in a literature example and then evaluate the impact of factors such as the burst sizes of cross-flows, the (maximum and minimum) packet sizes, and the number of cross-flows. Last, we also take a look at scenarios with larger numbers of flows in which we need to search for the optimal set M in our leftover services curves heuristically.

A. Comparison to state of the art

First, let us consider the example presented in [1]. We start off with a direct comparison of the service curves. The results are given in Figure 2a.

As expected, the stair function from Theorem 7.1 always provides a larger service curve than the rate-latency variant in Theorem 7.2. Moreover, we can also see the positive effect of the cycles within rounds for the IWRR service curve (Theorem 8). The first new leftover service curve, WRR M, is significantly larger due to the larger traffic-aware residual rate, except for the very start due to a larger latency period (recalling the illustrative Figure 1 and the corresponding discussion in the introduction). IWRR M, on the other hand, has a smaller latency period, yet the curve is then dominated by the larger rate $\frac{q_i}{q_i+Q'_M}\left(C - \sum_{k \notin M} r_k\right)$ of WRR M. This observation is consistent with our expectation, since we invoked more inequalities in the proof of IWRR being a bandwidth-sharing policy. Therefore, it is more difficult for the IWRR M service curve to benefit from the interleaving.

Next, we continue by comparing the impact of the service curves on the delay bounds (calculated using Theorem 6). Note that, for Theorem 7.2. and our leftover service curves in Corollary 12 and 14, we receive rate-latency functions and we can therefore apply closed-form solutions for the delay bound [18, p. 24]. For the stair functions, on the other hand, calculation is more complex [30]. The results are depicted in Figure 2b.

As expected, the stair function for WRR leads to better delay bounds than the result with the traffic-agnostic residual rate, while the interleaving reduces delay bounds even further. However, most importantly, our new service curves, taking cross-traffic into account, lead to significantly smaller delay bounds compared to all other curves. The decreasing gain over the state of the art for higher utilizations is expected, since for high utilizations all flows tend to be backlogged most of the time which forces the traffic-aware residual rate to get ever closer to the traffic-agnostic one.

B. Impact of burst sizes

It is clear that our new leftover service curves depend on the burst sizes of the cross-flows, while the state of the art is oblivious to it and is only affected by the maximum packet sizes as well as the weights of the cross-traffic. Therefore, in this numerical experiment we investigate this impact.

We consider three different classes of cross-traffic: low-, mid-, and high-burstiness flows in addition to the flow of interest (foi). We distribute 9 cross-flows over these 3 classes such that we start off with a scenario of cross-traffic with low burstiness and then gradually turn it into a scenario with high burstiness. Specifically, we denote by $(n_{\text{low}}, n_{\text{mid}}, n_{\text{high}})$ the

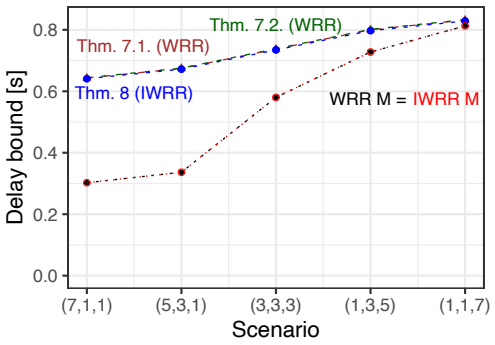


Fig. 3: For the parameter description of traffic classes, we use the notation $\{\text{low}, \text{mid}, \text{high}, \text{foi}\}$: weights $= \{4, 5, 6, 5\}$, burst sizes $= \{70, 700, 7000, 3000\}$ Kbit. We use the packet lengths $l_i^{\min} = 576$ bytes, $l_i^{\max} = 1500$ bytes and arrival rates $= 7$ Mb/s for each flow and a server utilization $= 0.7$.

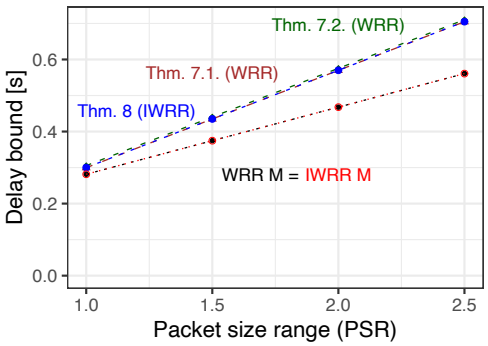


Fig. 4: Impact of the packet size range on the delay bounds. We choose the same parameters setting as in the previous experiment, except for the minimum packet size $l_i^{\min}, i = 1, \ldots, n$.

number of flows of each "burst class". The results are given in Figure 3.

As expected, we observe that for $(7, 1, 1)$, the case with the smallest cross-flow burstiness, the gain of using this information is the largest because the latency increase in our leftover service curves is small. In this case, the delay bounds are reduced by more than half compared to the state of the art. Increasing the burstiness, of course, reduces this advantage. However, it stays below even in the scenario with the highest burstiness.

C. Impact of packet sizes

The accuracy of all performance bounds depends on the variability of the packet sizes. In this numerical experiment, we examine how the ratio between maximum and minimum packet size of the flows impacts the delay bounds. To that end, we define the packet size range (PSR) as

$$\text{PSR} := \frac{\max_{i=1,\ldots,n} l_i^{\max}}{\min_{i=1,\ldots,n} l_i^{\min}}.$$

In the experiment, we start off with packets of equal size, that is, a PSR of 1, and then, by decreasing minimum packet sizes, increase the PSR. The results are displayed in Figure 4. Clearly, the delay bounds increase when the PSR increases. However, we observe that our analysis is affected less and the gain of our results over the state of the art increases. The likely reason is that our leftover service curves can better compensate for the higher packet size variability using the additional degree of freedom from the selection of flows that are assigned to the set M.

D. Impact of number of cross-flows

In the next experiment, we investigate the impact of the number of cross-flows. Specifically, we consider again the three "burst classes" with the same number of flows in each class, yet now increasing the number of flows per class. If k is the number of flows per class, clearly, we have $3k$ cross-flows in total. The results are depicted in Figure 5. We observe that

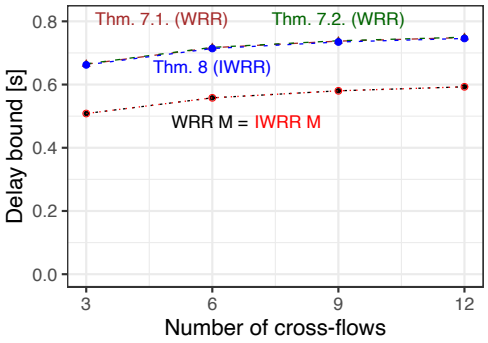


Fig. 5: The parameter setting is the same as in Figure 3, except the number of cross-flows per class.

our analysis improves on the state of the art by more than 20% across the different numbers of cross-flows.

E. Dealing with larger number of flows

In the previous experiment, we investigated the impact of the number of cross-flows on the delay bounds. The total number of flows was kept relatively moderate. In fact, as briefly discussed in Section III-C, if we want to deal with larger number of flows, we need to take into account that our leftover service curves rely on maximizing all possibly subsets M such that $i \in M \subset \mathcal{N}$, i.e., $2^{|\mathcal{N}|-1}$ combinations. Therefore, for a large number of flows, finding the optimal M to minimize the delay bound becomes computationally prohibitive. However, we do not have to evaluate all possible subsets but can actually apply a search heuristic to this combinatorial problem. Here, we briefly propose a very efficient and simple one:

- Let d_M denote the delay bound of WRR M for $i \in M \subset \mathcal{N}$.
- Set $d_{\text{opt}} := \infty$ and $M_{\text{opt}} := \{i\}$.
- Sort all cross-flows in a descending order by burst size, with j being the sorted flow index.
- For j from 1 to $|\mathcal{N}| - 1$
  calculate $d_{\text{new}} = \min\left\{d_{\text{opt}}, d_{M_{\text{opt}} \cup \{j\}}\right\}$
  if $d_{\text{new}} < d_{\text{opt}}$
  then set $d_{\text{opt}} := d_{\text{new}}$ and $M_{\text{opt}} := M_{\text{opt}} \cup \{j\}$.

Total number of flows	Delay bound heuristic	Delay bound Theorem 8
13	0.59	0.75
49	0.62	0.78
100	0.63	0.78
499	0.63	0.79
1000	0.64	0.79

TABLE I: Delay bound comparison between the heuristic and the state of the art under the parameters of Figure 3.

We compare this heuristic WRR M with the state of the art with the smallest delay bounds, Theorem 8 (IWRR), for large numbers of flows, again from the three classes as in previous experiments.

The results are given in Table I. We see that our heuristic yields significantly smaller delay bounds across the different numbers of flows. We measured a runtime of 39.3s to find the optimal M for 13 flows, while the heuristic took only $4 \cdot 10^{-4}$s. Even for the largest scenario with 1000 flows, the heuristic completed the search in less than 29.9s.

VI. Conclusion

In this paper, we have improved performance bounds of (interleaved) weighted round robin under the assumption of constrained cross-traffic. To that end, we showed that both discussed WRR variants are bandwidth-sharing policies. For WRR, we gained more insights by refining the flow weights with the respective worst-case packet lengths. Consequently, we exploited this property to derive new strict leftover service curves for (I)WRR. In a numerical evaluation, we observed that the improvement is not only substantial, but persistent across different experiments investigating the impact of several factors.

For future work, the extension to other round-robin schedulers and proving the bandwidth-sharing property for stair functions are interesting research challenges. Furthermore, motivated by the promising results, it will be very interesting to investigate systematically the heuristic optimization of the new leftover service curves for round-robin schedulers.

References

[1] S. M. Tabatabaee, J.-Y. Le Boudec, and M. Boyer, "Interleaved weighted round-robin: A network calculus analysis," *IEICE Transactions on Communications*, vol. 104, no. 12, pp. 1479–1493, 2021.

[2] E. L. Hahne, "Round-robin scheduling for max-min fairness in data networks," *IEEE Journal on Selected Areas in communications*, vol. 9, no. 7, pp. 1024–1039, 1991.

[3] L. Kleinrock, "Analysis of a time-shared processor," *Naval research logistics quarterly*, vol. 11, no. 1, pp. 59–73, 1964.

[4] M. Katevenis, S. Sidiropoulos, and C. Courcoubetis, "Weighted round-robin cell multiplexing in a general-purpose atm switch chip," *IEEE Journal on selected Areas in Communications*, vol. 9, no. 8, pp. 1265–1279, 1991.

[5] H. M. Chaskar and U. Madhow, "Fair scheduling with tunable latency: A round robin approach," in *Seamless Interconnection for Universal Services. Global Telecommunications Conference. GLOBECOM'99*, vol. 2, 1999, pp. 1328–1333.

[6] H. Xiao and Y. Jiang, "Analysis of multi-server round robin scheduling disciplines," *IEICE transactions on communications*, vol. 87, no. 12, pp. 3593–3602, 2004.

[7] "IEEE standard for local and metropolitan area network–bridges and bridged networks," *IEEE Std 802.1Q-2018 (Revision of IEEE Std 802.1Q-2014)*, pp. 1–1993, 2018.

[8] D. B. LD and P. V. Krishna, "Honey bee behavior inspired load balancing of tasks in cloud computing environments," *Applied soft computing*, vol. 13, no. 5, pp. 2292–2303, 2013.

[9] W. Wang and G. Casale, "Evaluating weighted round robin load balancing for cloud web services," in *2014 16th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing*, 2014, pp. 393–400.

[10] Y. Qian, Z. Lu, and W. Dou, "Analysis of worst-case delay bounds for best-effort communication in wormhole networks on chip," in *2009 3rd ACM/IEEE International Symposium on Networks-on-Chip*, 2009, pp. 44–53.

[11] J. Heißwolf, R. König, and J. Becker, "A scalable NoC router design providing QoS support using weighted round robin scheduling," in *2012 IEEE 10th International Symposium on Parallel and Distributed Processing with Applications*, 2012, pp. 625–632.

[12] *HPE FlexNetwork 5130 EI Switch Series*, Hewlett Packard Enterprise, 2017.

[13] M. Shreedhar and G. Varghese, "Efficient fair queueing using deficit round robin," in *Proc. of the conference on Applications, technologies, architectures, and protocols for computer communication*, 1995, pp. 231–242.

[14] C. Cicconetti, L. Lenzini, E. Mingozzi, and C. Eklund, "Quality of service support in IEEE 802.16 networks," *IEEE network*, vol. 20, no. 2, pp. 50–55, 2006.

[15] R. L. Cruz, "A calculus for network delay, part I: Network elements in isolation," *IEEE Transactions on information theory*, vol. 37, no. 1, pp. 114–131, 1991.

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

[17] C.-S. Chang, *Performance guarantees in communication networks*. London: Springer-Verlag, 2000.

[18] J.-Y. Le Boudec and P. Thiran, *Network calculus: a theory of deterministic queuing systems for the internet*. New York: Springer-Verlag, 2001.

[19] A. Bouillard, M. Boyer, and E. L. Corronc, *Deterministic Network Calculus: From Theory to Practical Implementation*. John Wiley & Sons, 2018.

[20] F. Ciucu and J. Schmitt, "Perspectives on network calculus – no free lunch, but still good value," in *Proc. ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM'12)*, New York, NY, USA, Aug. 2012, pp. 311–322.

[21] A. Bouillard, N. Farhi, and B. Gaujal, "Packetization and packet curves in network calculus," in *6th International ICST Conference on Performance Evaluation Methodologies and Tools*, 2012, pp. 136–137.

[22] M. Boyer and H. Daigmorte, "Improved service curve for element with known transmission rate," *IEEE Networking Letters*, 2022.

[23] A. G. Greenberg and N. Madras, "How fair is fair queuing," *Performance 90'*, vol. 39, no. 3, pp. 568–598, 1990.

[24] A. K. Parekh and R. G. Gallager, "A generalized processor sharing approach to flow control in integrated services networks: the single-node case," *IEEE /ACM Transactions on Networking*, vol. 1, no. 3, pp. 344–357, 1993.

[25] Z.-L. Zhang, D. Towsley, and J. Kurose, "Statistical analysis of the generalized processor sharing scheduling discipline," *IEEE Journal on Selected Areas in Communications*, vol. 13, no. 6, pp. 1071–1080, 1995.

[26] A. Burchard and J. Liebeherr, "A general per-flow service curve for gps," in *2018 30th International Teletraffic Congress (ITC 30)*, vol. 2, 2018, pp. 31–36.

[27] A. Bouillard, "Individual service curves for bandwidth-sharing policies using network calculus," *IEEE Networking Letters*, vol. 3, no. 2, pp. 80–83, 2021.

[28] J.-P. Georges, T. Divoux, and É. Rondeau, "Network calculus: application to switched real-time networking," in *5th International ICST Conference on Performance Evaluation Methodologies and Tools, VALUETOOLS'11*, 2011.

[29] A. Soni, X. Li, J.-L. Scharbarg, and C. Fraboul, "Wctt analysis of avionics switched ethernet network with wrr scheduling," in *Proc. of the 26th International Conference on Real-Time Networks and Systems (RTNS)*, 2018, pp. 213–222.

[30] A. Bouillard and É. Thierry, "An algorithmic toolbox for network calculus," *Discrete Event Dynamic Systems*, vol. 18, no. 1, pp. 3–49, 2008.