

Deficit Round-Robin: A Second Network Calculus Analysis

Seyed Mohammadhossein Tabatabaee
EPFL
 Lausanne, Switzerland
 hossein.tabatabaee@epfl.ch

Jean-Yves Le Boudec
EPFL
 Lausanne, Switzerland
 jean-yves.leboudec@epfl.ch

Abstract—Deficit Round-Robin (DRR) is a widespread scheduling algorithm that provides fair queuing with variable-length packets. Bounds on worst-case delays obtained with DRR were found by Boyer et al. They used a rigorous network calculus approach and characterized the service obtained by one flow of interest by means of a strict service curve. These bounds do not make any assumptions on the interfering traffic flows hence are pessimistic when the interfering traffic is constrained by some arrival curves. For such cases, Soni et al. improved the worst-case delay bounds by a correction term that accounts for arrival curve constraints of interfering traffic, using a semi-rigorous approach. Unfortunately, these latter bounds are incorrect, as we show by exhibiting a counter-example. Then we derive new service curves for DRR, which are rigorously proven, and we account for arrival curve constraints of interfering traffic. Hence, the resulting delay bounds are guaranteed to be correct. Furthermore, we find numerically that they are smaller than the incorrect ones obtained with the method of Soni et al. These bounds also improve on the results by Boyer et al. when there is no constraint on interfering traffic. Therefore, as of today, they are the best known delay bounds for DRR. Our results are obtained by applying the method of the pseudo-inverse.

I. INTRODUCTION

Deficit Round-Robin [1] is a scheduling algorithm that is often used for scheduling tasks, or packets, in real-time systems or communication networks. It is a variation of Weighted Round-Robin (WRR) that enables flows with variable packet lengths to fairly share the capacity. The capacity is shared among several clients or queues by giving each of them a quantum value and by providing more service to those with larger quantum. DRR is widely used because it exhibits a low complexity, $O(1)$, provided that an allocated quantum is no smaller than the maximum packet size; and it can be implemented in very efficient ways, such as the Aliquem implementation [2].

We are interested in delay bounds for the worst case, as is typical in the context of deterministic networking. To this end, a standard approach is network calculus. Specifically, with network calculus, the service offered to a flow of interest by a system is abstracted by means of a service curve. A bound on the worst-case delay is obtained by combining the service curve with an arrival curve for the flow of interest. An arrival curve is a constraint on the amount of data that the flow of interest can send; such a constraint is necessary

to the existence of a finite delay bound. The exact definitions are recalled in Section II-A.

The network calculus approach was applied to DRR in [3], where a *strict* service curve is obtained. A strict service curve is a special case of a service curve hence can be used to derive delay (and backlog) bounds. The result was obtained under general assumptions such as per flow maximum packet size and assuming a server that offers any kind of strict service curve rather than a constant-rate server. They show that their delay bounds are smaller than or equal to all previous works [4]–[6]. We call this the strict service curve of *Boyer et al.*

The strict service curve of Boyer et al. does not make any assumptions on the interfering traffic. Hence, the resulting delay bounds are valid, even in degraded operational mode, i.e., when interfering traffic behaves in an unpredictable way. However, in real-time systems, there is also interest in finding worst-case delay bounds for non-degraded operational mode, i.e., when interfering traffic behaves as expected and satisfies known arrival curve constraints. For such cases, significantly smaller delay bounds were presented at a recent RTSS conference [7]. The main improvement in [7] is derived as follows. First, the network calculus delay bound is computed using the strict service curve of Boyer et al.; then, it is improved by what we call the correction term of *Soni et al.* The correction term is obtained by subtracting two terms: The former gives the maximum possible interference caused by any interfering flow in a backlogged period of the flow of interest and is derived from a detailed analysis of DRR; and the latter gives the effective interference caused by an interfering flow in a backlogged period of the flow of interest, given the knowledge of an arrival curve of that interfering flow. Unfortunately, the method is semi-rigorous and cannot be fully validated. Indeed, our first contribution is to show that the correction term of Soni et al. is incorrect; we do so by exhibiting a counter-example that satisfies their assumptions and that has a larger delay (Section III).

Our next contribution is obtaining a better strict service curve for DRR when there is no arrival curve constraint on interfering traffic. To do so, we rely on the method of pseudo-inverse, as it enables us to capture all details of DRR; a similar method was used to obtain a strict service curve for Interleaved Weighted Round-Robin in [8]. We also provide simplified lower bounds that can be used when analytic, closed-form

expressions are important. One such lower bound is precisely the strict service curve of Boyer et al. (Fig. 3), hence the worst-case delay bounds obtained with our strict service curve are guaranteed to be less than or equal to those of Boyer et al.

Our following contribution is a new iterative method for obtaining better strict service curves for DRR that account for the arrival curve constraints of interfering flows. Our method is rigorous and is based on pseudo-inverses and output arrival curves of interfering flows. We also provide, due to the computational aspect, two simpler variants. The delay bounds obtained with our method are fully proven. Furthermore, we compute them for the same case studies as in Soni et al. [7] (including two illustration networks and an industrial-sized one), and we find that they are smaller than the incorrect ones that use the correction term of Soni et al. Hence as of today, it follows that our delay bounds are the best proven delay bounds for DRR, with or without constraints on interfering traffic.

The remainder of the paper is organized as follows. After giving some necessary background in Section II, we describe the counter example to Soni et al. in Section III. In Section IV, we present our new strict service curves for DRR, with no knowledge of interfering traffic. In Section V, we present our new strict service curves for DRR that account for the interfering arrival curve constraints. In Section VII, we use numerical examples to illustrate the improvement in delay bounds obtained with our new strict service curves.

II. BACKGROUND

A. Network Calculus Background

We use the framework of network calculus [9]–[11]. Let \mathcal{F} denote the set of wide-sense increasing functions $f : \mathbb{R}^+ \mapsto \mathbb{R}^+ \cup \{+\infty\}$. A flow is represented by a cumulative arrival function $A \in \mathcal{F}$ and $A(t)$ is the number of bits observed on the flow between times 0 and t . We say that a flow has $\alpha \in \mathcal{F}$ as *arrival curve* if for all $s \leq t$, $A(t) - A(s) \leq \alpha(t - s)$. An arrival curve α can always be assumed to be sub-additive, i.e., to satisfy $\alpha(s + t) \leq \alpha(s) + \alpha(t)$ for all s, t . A periodic flow that sends up to a bits every b time units has, as arrival curve, the stair function, defined by $\nu_{a,b}(t) = a \lceil \frac{t}{b} \rceil$. Another frequently used arrival curve is the token-bucket function $\alpha = \gamma_{r,b}$, with rate r and burst b , defined by $\gamma_{r,b}(t) = rt + b$ for $t > 0$ and $\gamma_{r,b}(t) = 0$ for $t = 0$. Both of these arrival curves are sub-additive.

Consider a system S and a flow through S with input and output functions A and D ; we say that S offers $\beta \in \mathcal{F}$ as a *strict service curve* to the flow if the number of bits of the flow output by S in any *backlogged* interval $(s, t]$ is $D(t) - D(s) \geq \beta(t - s)$. A strict service curve β can always be assumed to be super-additive (i.e., to satisfy $\beta(s + t) \geq \beta(s) + \beta(t)$ for all s, t) and wide-sense increasing (otherwise, it can be replaced by its super-additive and non-decreasing closure [11]). A frequently used strict service curve is the rate-latency function $\beta_{R,T} \in \mathcal{F}$, with rate R and latency T , defined by $\beta_{R,T}(t) = R[t - T]^+$, where we use the notation $[x]^+ = \max\{x, 0\}$. It is super-additive.

Assume that a flow, constrained by a sub-additive arrival curve α , traverses a system that offers a strict service curve β and that respects the ordering of the flow (per-flow FIFO). The delay of the flow is upper bounded by the horizontal deviation defined by $h(\alpha, \beta) = \sup_{t \geq 0} \{\inf\{d \geq 0 \mid \alpha(t) \leq \beta(t + d)\}\}$. Also, the output flow is constrained by an arrival curve $\alpha^* = \alpha \otimes \beta$. The computation of $h(\alpha, \beta)$ and α^* can be restricted to $t \in [0, t^*]$ for $t^* \geq \inf_{s > 0} \{\alpha(s) \leq \beta(s)\}$ [11, Prop. 5.13], [12].

For f and g in \mathcal{F} , the min-plus convolution is defined by $(f \otimes g)(t) = \inf_{0 \leq s \leq t} \{f(t - s) + g(s)\}$ and the min-plus deconvolution by $(f \oslash g)(t) = \sup_{s \geq 0} \{f(t + s) - g(s)\}$ [9]–[11]. We will use the min-plus convolution of a stair function with a linear function, as shown in Fig. 1.

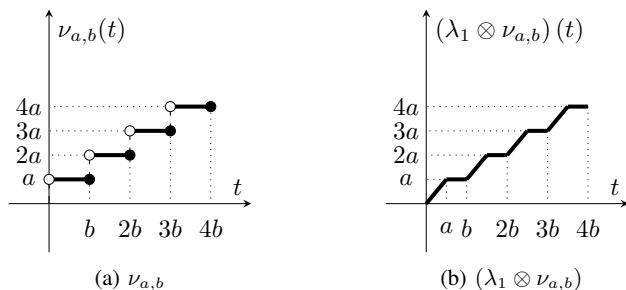


Fig. 1: Left: the stair function $\nu_{a,b} \in \mathcal{F}$ defined for $t \geq 0$ by $\nu_{a,b}(t) = a \lceil \frac{t}{b} \rceil$. Right: min-plus convolution of $\nu_{a,b}$ with the function $\lambda_1 \in \mathcal{F}$ defined by $\lambda_1(t) = t$ for $t \geq 0$, when $a \leq b$. The discontinuities are smoothed and replaced with a unit slope.

The non-decreasing closure f_\uparrow of a function $f : \mathbb{R}^+ \mapsto \mathbb{R}^+ \cup \{+\infty\}$ is the smallest function in \mathcal{F} that upper bounds f and is given by $f_\uparrow(t) = \sup_{s \leq t} f(s)$.

The lower pseudo-inverse f^\downarrow of a function $f \in \mathcal{F}$ is defined by $f^\downarrow(y) = \inf\{x \mid f(x) \geq y\} = \sup\{x \mid f(x) < y\}$ and satisfies [13, Sec. 10.1]:

$$\forall x, y \in \mathbb{R}^+, y \leq f(x) \Rightarrow x \geq f^\downarrow(y) \quad (1)$$

The network calculus operations can be automated in tools such as RealTime-at-Work (RTaW) [14], an interpreter that provides efficient implementations of min-plus convolution, min-plus deconvolution, non-decreasing closure, horizontal deviation, the composition of two functions, and a maximum and minimum of functions for piecewise-linear functions. All computations use infinite precision arithmetic (with rational numbers).

B. Deficit Round-Robin

A DRR subsystem serves n inputs, has one queue per input, and uses Algorithm 1 for serving packets. Each queue i is assigned a quantum Q_i . DRR runs an infinite loop of *rounds*. In one round, if queue i is non-empty, a service for this queue starts and its *deficit* is increased by Q_i . The service ends when either the deficit is smaller than the head-of-the-line packet or the queue becomes empty. In the latter case, the deficit is set back to zero. The `send` instruction is assumed to be the only one with a non-null duration. Its actual duration depends on

the packet size but also on the amount of service available to the entire DRR subsystem.

Algorithm 1 Deficit Round-Robin

Input: Integer quantum Q_1, Q_2, \dots, Q_n

Data: Integer deficits: d_1, d_2, \dots, d_n

```

1: for  $i \leftarrow 1$  to  $n$  do
2:    $d_i \leftarrow 0$ ;
3: end for
4: while True do
5:   for  $i \leftarrow 1$  to  $n$  do
6:     if (not empty( $i$ )) then
7:        $\triangleright$  A service for queue  $i$  starts.
8:        $d_i \leftarrow d_i + Q_i$ ;
9:       while (not empty( $i$ ))
10:        and (size(head( $i$ ))  $\leq d_i$ ) do
11:          $d_i \leftarrow d_i - \text{size}(\text{head}(i))$ ;
12:         send(head( $i$ ));
13:         removeHead( $i$ );
14:       end while
15:        $\triangleright$  A service for queue  $i$  ends.
16:     if (empty( $i$ )) then
17:        $d_i \leftarrow 0$ ;
18:     end if
19:   end if
20: end for
21: end while

```

In [3] as in much of the literature on DRR, the set of packets that use a given queue is called a *flow*; a flow may however be an aggregate of multiple flows, called micro-flows [15]; and an aggregate flow is called a *class* in [7]. In this paper, and in order to be consistent with the network calculus conventions, we use the former terminology and consider that a DRR input corresponds to one flow. When comparing our results to [7], the reader is invited to remember that a DRR flow in this paper corresponds to a DRR class in [7].

The DRR subsystem is itself placed in a larger system and can compete with other queuing subsystems. A common case is when the DRR subsystem is at the highest priority on a non-preemptive server with line rate c . Due to non-preemption, the service offered to the DRR subsystem might not be instantly available. This can be modelled by means of a rate-latency strict service curve (see Section II-A for the definition), with rate c and latency $\frac{c}{L^{\max}}$ where L^{\max} is the maximum packet size of lower priority. If the DRR subsystem is not at the highest priority level, this can be modelled with a more complex strict service curve [11, Section 8.3.2]. This motivates us to assume that the aggregate of all flows in the DRR subsystem receives a strict service curve β , which we call “aggregate strict service curve”. If the DRR subsystem has exclusive access to a transmission line of rate c , then $\beta(t) = ct$ for $t \geq 0$. We assume that $\beta(t)$ is finite for every (finite) t . (Note that the aggregate strict service curve β should not be confused with the strict service curves (also called “residual”

strict service curves in [3]) that we obtain in this paper for every flow.)

Here, we use the language of communication networks, but the results equally apply to real-time systems: Simply map flow to task, map packet to job, map packet size to job-execution time, and map strict service curve to “delivery curve” [16], [17].

C. Strict Service Curve of Boyer et al.

The strict service curve of Boyer et al. for DRR is given in [3], and we rewrite it using our notation. For flow i , let d_i^{\max} be its maximum residual deficit, defined by $d_i^{\max} = l_i^{\max} - \epsilon$ where l_i^{\max} is an upper bound on the packet size and ϵ is the smallest unit of information seen by the scheduler (e.g., one bit, one byte, or one 32-bit word). Also, let $Q_{\text{tot}} = \sum_{j=1}^n Q_j$. Then, for every flow i , their strict service curve is the rate-latency service curve $\beta_{R_i, T_i}(\beta(t))$ with rate $R_i = \frac{Q_i}{Q_{\text{tot}}}$ and latency $T_i = \sum_{j \neq i} d_j^{\max} + (1 + \frac{d_i^{\max}}{Q_i}) \sum_{j \neq i} Q_j$ (see Section II-A for the definition of a rate-latency function).

D. Correction Term of Soni et al.

When interfering flows are constrained by some arrival curves, Soni et al. give a correction term that improves the obtained delay bounds using the strict service curve of Boyer et al. in [7], which we now rewrite using our notation. Assume that every flow i has an arrival curve α_i , and the server is a constant-rate server with a rate equal to c . Let $D_i^{\text{Boyer-et-al}}$ be the network calculus delay bound for flow i obtained by combining α_i with the strict service curve of Boyer et al., as explained in Section II-C. The delay bound proposed in [7] is $D_i^{\text{Soni-et-al}} = D_i^{\text{Boyer-et-al}} - C_i^{\text{Soni-et-al}}$ with

$$C_i^{\text{Soni-et-al}} = \sum_{j, j \neq i} \frac{\left[S_j(D_i^{\text{Boyer-et-al}}) - \alpha_j(D_i^{\text{Boyer-et-al}}) \right]^+}{c} \quad (2)$$

where $S_j(t) \stackrel{\text{def}}{=} (Q_j + d_j^{\max}) 1_{t \geq h_i} + Q_j (1 + \lfloor \frac{c(t-H_i)}{Q_{\text{tot}}} \rfloor) 1_{t \geq H_i}$, $h_i = \frac{\sum_{j \neq i} Q_j + d_j^{\max}}{c}$ and $H_i = h_i + \frac{Q_i - d_i^{\max} + \sum_{j \neq i} Q_j}{c}$. In the correction term $C_i^{\text{Soni-et-al}}$, function S_j represents a lower bound on the maximum interference caused by flow j in a backlogged period of flow i ; the term with the arrival curve α_j represents the actual interference caused by flow j .

Two additional improvements are used in [7]. The former, called *grouping*, uses the fact that, if a collection of flows is known to arrive on the same link, the rate limitation imposed by the link can be used to derive, for the aggregate flow, an arrival curve that is smaller than the sum of arrival curves of the constituent flows. This improvement is also known under the name of line shaping and is used, for example, in [18]–[20]. The other improvement, called *offsets*, uses the fact that, if several periodic flows have the same source and if their offsets are known, the temporal separation imposed by the offsets can be used to compute, for the aggregate flow, an arrival curve that is also smaller than the sum of arrival curves of the constituent flows (the latter would correspond to an adversarial choice of the offsets). Both improvements reduce

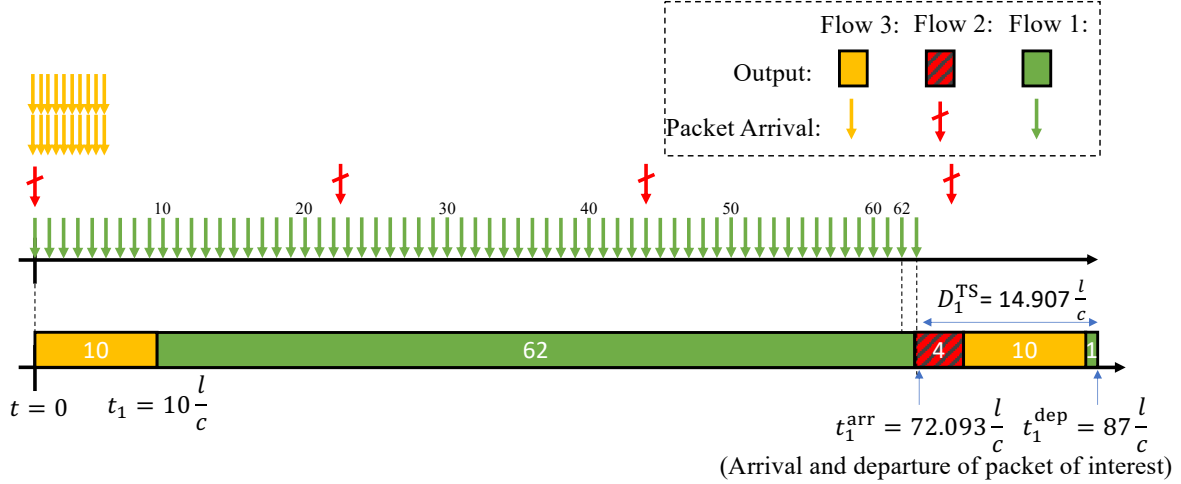


Fig. 2: Trajectory scenario for the packet of interest of flow 1 (Section III-B). This packet arrives at t_1^{arr} and departs at t_1^{dep} .

the arrival curves, hence the delay bounds. Note that both improvements are independent of the correction term (and, unlike the correction term, are correct); they can be applied to any method used to compute delay bounds, as we do in Section VII.

III. THE CORRECTION TERM OF SONI ET AL. IS INCORRECT

In this section, we show that the delay bound of Soni et al., namely the correction term given in equation (14) in [7], is invalid. For flow 1 in a system, we denote the delay bound of Soni et al. by $D_1^{\text{Soni-et-al}}$, and we denote the delay experienced by a packet of flow 1 in the trajectory scenario by D_1^{TS} .

A. System Parameters

Consider a constant-rate server, with a rate equal to c , that uses the DRR scheduling policy. All flows have packets of constant size l , and have quanta $Q_1 = 100l$, $Q_2 = 5l$, and $Q_3 = 10l$.

Each flow is constraint by a token-bucket arrival curve:

- 1) $\alpha_1(t) = \gamma_{r_1, b_1}$ with $r_1 = 0.86c$ and $b_1 = l$.
- 2) $\alpha_2(t) = \gamma_{r_2, b_2}$ with $r_2 = 0.0401c$ and $b_2 = l$.
- 3) $\alpha_3(t) = \gamma_{r_3, b_3}$ with $0 \leq r_3 < \frac{Q_3}{Q_{\text{tot}}}c$ and $b_3 = 20l$.

Assuming a token-bucket $\gamma_{r, b}$ for a flow implies that this flow has a minimum packet-arrival time equal to $\frac{l}{r}$. Also, observe that $r_i < \frac{Q_i}{Q_{\text{tot}}}c$ for $i = 1, 2, 3$. We compute the delay bound of Soni et al. for flow 1, as explained in II-D, and we obtain $D_1^{\text{Soni-et-al}} = 14.03383\frac{l}{c} - 1.236215\frac{\epsilon}{c}$.

B. Trajectory Scenario

We now construct a possible trajectory for our system. First, we give the inputs of our three flows. All queues are empty, and the server is idle at time $t = 0$. Then,

- 1) Flow 3 arrives first and $A_3(t) = \min(\alpha_3(t), 20l)$ for $t > 0$ (yellow arrows in Fig. 2).

- 2) Flow 1 arrives shortly after flow 3 and $A_1(t) = \min(\alpha_1(t), 63l)$ for $t > 0$ (green arrows in Fig. 2).
- 3) Flow 2 arrives shortly after flows 1 and 3 and $A_2(t) = \min(\alpha_2(t), 4l)$ for $t > 0$ (red arrows in Fig. 2).

Then, for the output, we have the following:

1) Flow 3 arrives first and has 20 ready packets. As its deficit was zero before this service and $Q_3 = 10l$, the server serves 10 packets of this flow. The end of the service for flow 3 is $t_1 = 10\frac{l}{c}$ (the first yellow part in Fig. 2).

2) Then, there is an emission opportunity for flow 1 and $A_1(t_1) = 9.6l$, which means flow 1 has 9 ready packets at time t_1 . The server starts serving packets of this flow. At the end of service of these first 9 packets, at $t_2 = 19\frac{l}{c}$, flow 1 has another 8 ready packets; hence, the server still serves packets of flow 1. This continues and 62 packets of flow 1 are served in this emission opportunity; the emission opportunity ends at $t_4 = 72\frac{l}{c}$ (the first green part in Fig. 2).

3) Then, there is an emission opportunity for flow 2 and $A_2(t_4) = 3.8872l$, which means flow 2 has 3 ready packets at time t_4 . At the end of service of 3 packets, another packet is also ready for flow 2. In total, 4 packets of flow 2 are served in this emission opportunity (the red part in Fig. 2).

4) A packet for flow 1 arrives at $t_1^{\text{arr}} = 72 + \frac{0.08l}{r_1} \approx 72.093\frac{l}{c}$. This packet should wait for flow 2 and flow 3 to use their emission opportunities, and then it can be served. We call this *the packet of interest* of flow 1, for which we capture the delay (the first blue arrow, at t_1^{arr} , on Fig. 2).

5) For flow 3, again 10 packets are served (the second yellow part in Fig. 2).

6) Finally, the packet of interest is served and its departure time is $t_1^{\text{dep}} = 87\frac{l}{c}$.

It follows that the delay for the packet of interest is $D_1^{\text{TS}} = t_1^{\text{dep}} - t_1^{\text{arr}} = 15\frac{l}{c} - \frac{0.08l}{r_1} \approx 14.907\frac{l}{c}$. Note that $D_1^{\text{TS}} > D_1^{\text{Soni-et-al}}$. To fix ideas, if $l = 100$ bytes and $c = 100$ Mb/s, the delay bounds are $D_1^{\text{Boyer-et-al}} = 146.228\mu\text{s}$, $D_1^{\text{Soni-et-al}} = 112.172\mu\text{s}$, and $D_1^{\text{TS}} = 119.256\mu\text{s}$.

C. The Contradiction with the Bound of Soni et al.

We found a trajectory scenario such that $D_1^{\text{Soni-et-al}}$ is not a valid delay bound. Let us explain why the approach of Soni et al., presented in [7], gives an invalid delay bound. In [7], it is implicitly assumed that as the delay for a packet of flow 1 is upper bounded by $D_1^{\text{Boyer-et-al}}$ (the obtained delay bound using the strict service curve of Boyer et al. for flow 1), only packets of interfering flows arriving within a duration $D_1^{\text{Boyer-et-al}}$ will get a chance to delay a given packet of flow 1. However, in the trajectory scenario given in Section III-B, all packets of flow 2 (an interfering flow for flow 1) arriving within the time interval $[0, 75\frac{l}{c}]$ with the duration $75\frac{l}{c} \gg D_1^{\text{Boyer-et-al}} = 18.3\frac{l}{c} - 2.15\frac{c}{c}$ delay the packet of interest of flow 1.

IV. NEW DRR STRICT SERVICE CURVE

Our next result is a non-convex strict service curve for DRR; it dominates the state-of-the-art rate-latency strict service curve for DRR by Boyer et al. We also give simpler, lower approximations of it. Specifically, we also find a convex strict service curve and two rate-latency strict service curves.

Theorem 1 (Non-convex Strict Service Curve for DRR). *Let S be a server shared by n flows that uses DRR, as explained in Section II-B, with quantum Q_i for flow i . Recall that the server offers a strict service curve β to the aggregate of the n flows. For any flow i , d_i^{\max} is the maximum residual deficit (defined in Section II-C).*

Then, for every i , S offers to flow i a strict service curve β_i^0 given by $\beta_i^0(t) = \gamma_i(\beta(t))$ with

$$\gamma_i(x) = (\lambda_1 \otimes \nu_{Q_i, Q_{\text{tot}}}) \left([x - \psi_i(Q_i - d_i^{\max})]^+ \right) \quad (3)$$

$$+ \min([x - \sum_{j \neq i} (Q_j + d_j^{\max})]^+, Q_i - d_i^{\max})$$

$$Q_{\text{tot}} = \sum_{j=1}^n Q_j \quad (4)$$

$$\psi_i(x) \stackrel{\text{def}}{=} x + \sum_{j, j \neq i} \phi_{i,j}(x) \quad (5)$$

$$\phi_{i,j}(x) \stackrel{\text{def}}{=} \left\lfloor \frac{x + d_i^{\max}}{Q_i} \right\rfloor Q_j + (Q_j + d_j^{\max}) \quad (6)$$

Here, $\nu_{a,b}$ is the stair function, λ_1 is the unit rate function and \otimes is the min-plus convolution, all described in Fig. 1.

The proof is in Section VI. See Fig. 3 for some illustrations of β_i^0 . Observe that γ_i in (3) is the strict service curve obtained when the aggregate strict service curve is $\beta = \lambda_1$ (i.e., when the aggregate is served at a constant, unit rate). In the common case where β is equal to a rate-latency function, say $\beta_{c,T}$, we have $\beta_i^0(t) = \gamma_i(c(t-T))$ for $t \geq T$ and $\beta_i^0(t) = 0$ for $t \leq T$, namely, β_i^0 is derived from γ_i by a re-scaling of the x axis and a right-shift.

The function $\phi_{i,j}(x)$, defined in (6), is the maximum interference that flow j can create in any backlogged period of flow i , such that flow i receives a service x . Using $\phi_{i,j}$ as it is results in the strict service curve of Theorem 1, which has a complex

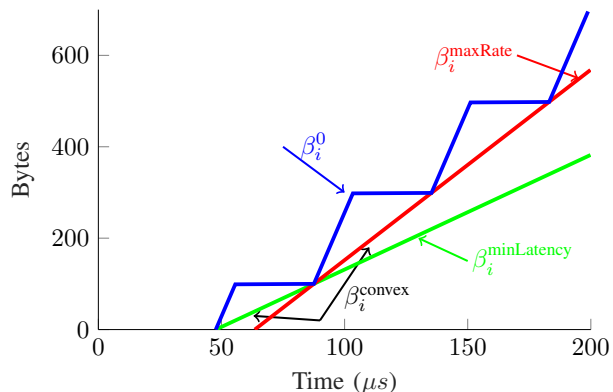


Fig. 3: Strict service curves for DRR for an example with three input flows, quanta = $\{199, 199, 199\}$ bytes, maximum residual deficits $d^{\max} = \{99, 99, 99\}$ bytes, and $\beta(t) = ct$ with $c = 100$ Mb/s (i.e., the aggregate of all flows is served at a constant rate). The figure shows the non-convex DRR strict service curve β_i^0 of Theorem 1; it also shows the two rate-latency strict service curves β_i^{\maxRate} (same as that Boyer et al.) and β_i^{\minLatency} in Corollary 1 and the convex service curve $\beta_i^{\text{convex}} = \max(\beta_i^{\maxRate}, \beta_i^{\minLatency})$ in Corollary 2.

expression. If there is interest in simpler expressions, any lower bounding function is a strict service curve. In Theorem 2, we show that any upper bounding of function $\phi_{i,j}$, (which gives a lower bound on γ_i) results in a lower bound of β_i^0 , which is a valid, though less good, strict service curve for DRR.

Theorem 2 (Lower Bounds of Non-convex Strict Service Curves for DRR). *Make the same assumptions as in Theorem 1. Also, for flow i , consider functions $\phi'_{i,j} \in \mathcal{F}$ such that for every other flow $j \neq i$, $\phi'_{i,j} \geq \phi_{i,j}$. Let ψ'_i be defined as in (5) by replacing functions $\phi_{i,j}$ with $\phi'_{i,j}$ for every flow $j \neq i$.*

Then, let γ'_i be the lower-pseudo inverse of ψ'_i , i.e., $\gamma'_i = \psi'^{\downarrow}_i$.

Let $\beta_i^{0'}$ be the result of Theorem 1 by replacing functions $\phi_{i,j}$, ψ_i , and γ_i with $\phi'_{i,j}$, ψ'_i , and γ'_i .

Then, S offers to flow i a strict service curve $\beta_i^{0'}$ and $\beta_i^{0'} \leq \beta_i^0$.

The proof is in Section VI-B. There is often interest in service curves that are piecewise-linear and convex, a simple case is a rate-latency function. Specifically, convex piecewise-linear functions are stable under addition and maximum, and the min-plus convolution can be computed in automatic tools very efficiently [11, Sec. 4.2]. Observe that, if the aggregate service curve β is a rate-latency function, replacing γ_i by a rate-latency (resp. convex) lower-bounding function also yields a rate-latency (resp. convex) function for β_i^0 , and vice-versa. Therefore, we are interested in rate-latency (resp. convex) functions that lower bound γ_i . We now give two lower bounds of the non-convex strict service curve for DRR using Theorem 2 that are common: a convex lower bound and two rate-latency lower bounds.

To obtain a rate-latency strict service curve, we use two affine upper bounds of $\phi_{i,j}$: $\phi_{i,j}^{\maxRate}$, which results in a rate-

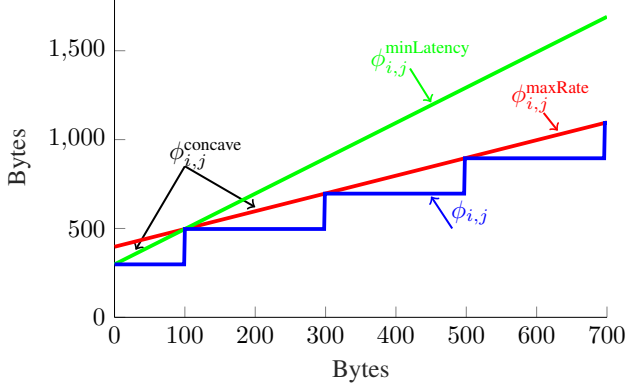


Fig. 4: Illustration of functions $\phi_{i,j}$, $\phi_{i,j}^{\maxRate}$, $\phi_{i,j}^{\minLatency}$, and $\phi_{i,j}^{\text{concave}}$ defined in (6), (7), (8), and (13), respectively. These functions are obtained for the example of Fig. 3.

latency function with the maximum rate, and $\phi_{i,j}^{\minLatency}$, which results in a rate-latency function with the minimum latency (Fig. 4). They are defined by

$$\phi_{i,j}^{\maxRate}(x) \stackrel{\text{def}}{=} \frac{Q_j}{Q_i} (x + d_i^{\max}) + Q_j + d_j^{\max} \quad (7)$$

$$\phi_{i,j}^{\minLatency}(x) \stackrel{\text{def}}{=} \frac{Q_j}{Q_i - d_i^{\max}} x + Q_j + d_j^{\max} \quad (8)$$

Corollary 1 (Rate-Latency Strict Service Curve for DRR). *With the assumption in Theorem 1 and the definitions (7)-(8), S offers to every flow i strict service curves $\gamma_i^{\maxRate}(\beta(t))$ and $\gamma_i^{\minLatency}(\beta(t))$ with*

$$\gamma_i^{\maxRate} = \beta_{R_i^{\max}, T_i^{\max}} \quad (9)$$

$$\gamma_i^{\minLatency} = \beta_{R_i^{\min}, T_i^{\min}} \quad (10)$$

$$R_i^{\max} = \frac{Q_i}{Q_{\text{tot}}} \text{ and } T_i^{\max} = \sum_{j,j \neq i} \phi_{i,j}^{\maxRate}(0) \quad (11)$$

$$R_i^{\min} = \frac{Q_i - d_i^{\max}}{Q_{\text{tot}} - d_i^{\max}} \text{ and } T_i^{\min} = \sum_{j,j \neq i} \phi_{i,j}^{\minLatency}(0) \quad (12)$$

The right-hand sides in (9) and (10) are the rate-latency functions defined in Section II-A.

The above result is obtained by using Theorem 2 with $\phi_{i,j}^{\maxRate}$ and $\phi_{i,j}^{\minLatency}$; hence, $\gamma_i \geq \gamma_i^{\maxRate}$ and $\gamma_i \geq \gamma_i^{\minLatency}$. Also, observe that the strict service curve of Boyer et al., explained in Section II-C, is equal to $\gamma_i^{\maxRate}(\beta(t))$. It follows that β_i^0 dominates it; hence, obtained delay bound using β_i^0 are guaranteed to be less than or equal to those of Boyer et al.

A better upper bound on $\phi_{i,j}$ can be obtained by taking its concave closure (i.e., the smallest concave upper bound) that is equal to the minimum of $\phi_{i,j}^{\maxRate}$ and $\phi_{i,j}^{\minLatency}$:

$$\phi_{i,j}^{\text{concave}}(x) = \min \left(\phi_{i,j}^{\maxRate}(x), \phi_{i,j}^{\minLatency}(x) \right) \quad (13)$$

Corollary 2 (Convex Strict Service Curve for DRR). *With the assumption in Theorem 1 and the definitions (9)-(10), S offers to every flow i a strict service curve $\gamma_i^{\text{convex}}(\beta(t))$ with*

$$\gamma_i^{\text{convex}}(x) = \max \left(\gamma_i^{\maxRate}(x), \gamma_i^{\minLatency}(x) \right) \quad (14)$$

The above result is obtained by using Theorem 2 with $\phi_{i,j}^{\text{concave}}$. Also, it can be shown that it is the largest convex lower bound of γ_i . When β is a rate-latency function, this provides a convex piecewise-linear function, which has all the good properties mentioned earlier.

V. NEW DRR STRICT SERVICE CURVES THAT ACCOUNT FOR ARRIVAL CURVES OF INTERFERING FLOWS

The next result improves the strict service curves presented in Section IV by taking into account arrival curve constraints of interfering flows.

A. *Service Curves Derived from Non-Concave $\phi_{i,j}$*

Theorem 3 (Strict Service Curve for DRR Derived from Non-Concave Function $\phi_{i,j}$). *Let S be a server with the assumptions in Theorem 1. Also, assume that every flow i has an arrival curve $\alpha_i \in \mathcal{F}$.*

Let γ_i and $\phi_{i,j}$ be as defined as in (3) and (6). Recall that β_i^0 is defined in Theorem 1. For every integer $m \geq 1$ and every flow i , define $\beta_i^m \in \mathcal{F}$ by

$$\beta_i^m = \gamma_i \circ (\beta + \delta_i^{m-1})_{\uparrow} \quad (15)$$

with

$$\delta_i^{m-1}(t) \stackrel{\text{def}}{=} \sum_{j,j \neq i} [\phi_{i,j}(\beta_i^{m-1}(t)) - (\alpha_j \circ \beta_j^{m-1})(t)]^+ \quad (16)$$

Then 1) for every $m \geq 0$, β_i^m is a strict service curve for flow i in S and 2) $\beta_i^0 \leq \beta_i^1 \leq \beta_i^2 \leq \dots$

In (15), \uparrow is the non-decreasing closure, defined in Section II-A, and \circ is the composition of functions; also, note that β and δ_i^{m-1} are functions of the time t .

The proof is in Section VI-C. The essence of Theorem 3 is as follows. Equations (15) and (16) give new strict service curves β_i^m for every flow i ; they are derived from already available strict service curves β_i^{m-1} and from arrival curves on the input flows α_j . Then, this is iteratively applied, starting with the strict service curves that do not make any assumptions on interfering traffic, as obtained in Section IV. Item 2) states that, at every step, we have obtained a collection of strict service curves that can only improve on the previous steps.

We use Theorem 3 as follows. The strict service curve β_i^m obtained at the m th iteration gives a valid delay bound for flow i , obtained as the horizontal deviation $h(\alpha_i, \beta_i^m)$. For every flow i , the sequence $h(\alpha_i, \beta_i^m)$ decreases (in the wide sense) with m and is non-negative, hence converges. As the number of flows n is finite, we can stop computing strict service curves when the delay bound decreases insignificantly for all flows. Note that, at every iteration, we have obtained valid delay bounds, hence the stopping criterion does not affect the exactness of the final outcome.

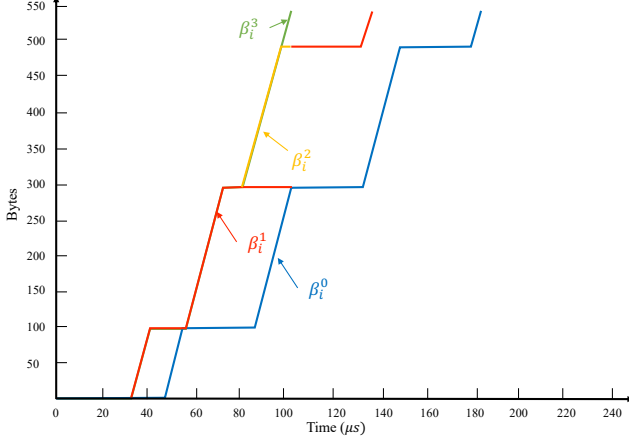


Fig. 5: Strict service curves of Theorem 3 for flow 2 of the example of Fig. 3; all flows have token-bucket arrival curves with $r = \{5, 1, 1\} \frac{1}{512}$ Mb/s and $b = \{5l, l, l\}$. When applying the scheme of Theorem 3, after 3 iterations, the strict service curves of all flows become stationary in the horizon of the figure, and the scheme stops. The sufficient horizon t^* in this example is $200\mu\text{s}$. Obtained with the RTaW online tool.

The computation of service curves in Theorem 3 and of the resulting delay bounds can be restricted to a finite horizon. Indeed, all computations in Theorem 3 are causal except for the min-plus deconvolution $\alpha_j \circ \beta_j^{m-1}$. But, as mentioned in Section II-A, such a computation and the computation of delay bounds can be limited to $t \in [0; t^*]$ for any positive t^* such that $\alpha_j(t^*) \leq \beta_j^{m-1}(t^*)$ for every $m \geq 1$ and $j = 1 : n$. To find such a t^* , we can use any lower bound on β_j^{m-1} , an obvious choice being β_j^0 . We then compute $t_j^* = \inf_{s>0} \{\alpha_j(s) \leq \beta_j^0(s)\}$ and take, as *sufficient horizon*, $t^* = \max_j t_j^*$. The computations in Theorem 3 can then be limited to this horizon or any upper bound on it. In some cases that we tested, the iterative scheme becomes stationary in such a finite horizon (Fig. 5).

The computations can be performed with a tool such as RealTime-at-Work (RTaW) [14], which uses an exact representation of functions with finite horizon, by means of rational numbers with exact arithmetic.

B. Simpler Versions of Theorem 3

Computation of the strict service curves of Theorem 3 can be costly. We first explain some sources of complexity and how to address them. We then propose two simpler versions.

One source of complexity lies in the initial strict service curves β_i^0 . For every flow i , β_i^0 can be replaced by its simpler lower bounds. As presented in Section IV, β_i^0 can be replaced by its convex closure $\gamma_i^{\text{convex}}(\beta(t))$, or rate-latency functions $\gamma_i^{\text{minLatency}}(\beta(t))$ and $\gamma_i^{\text{maxRate}}(\beta(t))$.

Another source of complexity is function $\phi_{i,j}$ (and the resulting function γ_i). Function $\phi_{i,j}$, as defined in (6), is non-concave and non-linear (because it uses floor operations). This might create discontinuities that can make the computation hard, see Fig. 5. To address this problem, we first give a generic Theorem to obtain simpler versions of Theorem 3,

by upper bounding $\phi_{i,j}$, at the expense of perhaps less good results.

Theorem 4 (Other Versions of Strict Service Curves for DRR of Theorem 3). *Make the same assumptions as in Theorem 3. Also, for a flow i , let $\phi_{i,j}'$ and γ_i' be defined as in Theorem 2.*

For every integer $m \geq 0$, let $\beta_i^{m'}$ be the result of Theorem 3 by replacing functions $\phi_{i,j}$ and γ_i with $\phi_{i,j}'$ and γ_i' , respectively.

Then, for every integer $m \geq 0$, S offers to flow i a strict service curve $\beta_i^{m'}$ and $\beta_i^{0'} \leq \beta_i^{1'} \leq \beta_i^{2'} \leq \dots$

The proof is not given in detail, as it is similar to the proof of Theorem 3 after replacing functions $\phi_{i,j}$ and γ_i with $\phi_{i,j}'$ and γ_i' , respectively. By applying Theorem 4 with $\phi_{i,j}^{\text{concave}}$, we obtain the following result.

Corollary 3 (Strict Service Curve for DRR Derived from Concave Function $\phi_{i,j}^{\text{concave}}$). *Theorem 3 holds by replacing $\phi_{i,j}$ and γ_i with $\phi_{i,j}^{\text{concave}}$ and γ_i^{convex} defined in (13) and (14), and the initial strict service curves β_i^0 as in Corollary 2.*

The sequence of strict service curves obtained by Corollary 3 is thus defined by $\beta_i^{\text{concave},0} = \gamma_i^{\text{convex}} \circ \beta = \beta_i^{\text{convex}}$ and for $m \geq 1$:

$$\beta_i^{\text{concave},m} = \gamma_i^{\text{convex}} \circ \left(\beta + \delta_i^{\text{concave},m-1} \right)_\uparrow \quad (17)$$

with $\delta_i^{\text{concave},m-1} =$

$$\sum_{j,j \neq i} \left[\phi_{i,j}^{\text{concave}} \circ \beta_i^{\text{concave},m-1} - \alpha_j \circ \beta_j^{\text{concave},m-1} \right]^+ \quad (18)$$

Let us explain why computing Corollary 3 is simpler. The first reason is the computing of the composition of $\phi_{i,j}^{\text{concave}}$ with another function. Observe that for a function f , $\phi_{i,j}^{\text{concave}}(f(t))$ is equal to $\min \left(\frac{Q_i}{Q_i} f(t) + \phi_{i,j}^{\text{maxRate}}(0), \frac{Q_i}{Q_i - d_i^{\text{max}}} f(t) + \phi_{i,j}^{\text{minLatency}}(0) \right)$, which includes only multiplication, addition, and minimum operations. The second reason is the computing of the min-plus deconvolution. As illustrated in Fig. 6, computing the min-plus deconvolution with such strict service curves is much simpler than with those in Fig. 5. The last reason is computing the composition of γ_i^{convex} with another function. Observe that for a function f , $\gamma_i^{\text{convex}}(f(t))$ is equal to $\max \left(R_i^{\text{max}} [f(t) - T_i^{\text{max}}]^+, R_i^{\text{min}} [f(t) - T_i^{\text{min}}]^+ \right)$, which again includes only multiplication, addition, and maximum operations.

By combining the application of Theorem 4 to $\phi_{i,j}^{\text{maxRate}}$ and $\phi_{i,j}^{\text{minLatency}}$, we obtain the following result that has a complexity slightly different than the previous results. It performs simpler min-plus computations, but the number of such computations is doubled. On the examples we tested, the run time is similar to that of Corollary 3 when arrival curves are concave (token buckets) but is smaller when they are non-concave (stair functions).

Corollary 4 (Strict Service Curve for DRR Derived from Combining Affine Functions $\phi_{i,j}^{\text{maxRate}}$ and $\phi_{i,j}^{\text{minLatency}}$). *Theorem*

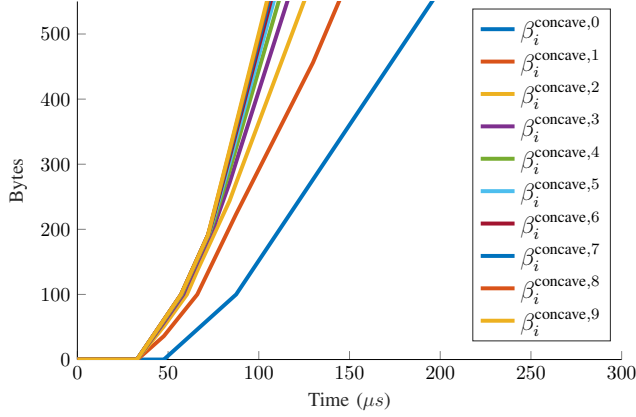


Fig. 6: Strict service curves of Corollary 3 for flow 2 of the example of Fig. 5. The iterative scheme is stopped when the computed delay bounds for all flows decreases by less than $0.25\mu\text{s}$. Obtained with MATLAB.

3 holds by replacing $\phi_{i,j}$ and γ_i with $\phi_{i,j}^{\text{maxRate}}$ and $\gamma_i^{\text{maxRate}}$ defined in (7) and (9). It also holds by replacing $\phi_{i,j}$ and γ_i with $\phi_{i,j}^{\text{minLatency}}$ and $\gamma_i^{\text{minLatency}}$ defined in (8) and (10). In both cases, let the initial strict service curves β_i^0 be defined as in Corollary 2. Last, the theorem also holds if, at each step m , we take the maximum of the two strict service curves obtained in each case.

The sequence of strict service curves obtained by Corollary 4 is thus defined by $\beta_i^{\text{affine},0} = \gamma_i^{\text{convex}} \circ \beta = \beta_i^{\text{concave},0} = \beta_i^{\text{convex}}$ and for $m \geq 1$, $\beta_i^{\text{affine},m} = \max(\beta_i^{\prime m}, \beta_i^{\prime\prime m})$ with

$$\begin{aligned} \beta_i^{\prime m} &= \gamma_i^{\text{minLatency}} \circ (\beta + \delta_i^{\text{minLatency},m-1})_{\uparrow}, \\ \beta_i^{\prime\prime m} &= \gamma_i^{\text{maxRate}} \circ (\beta + \delta_i^{\text{maxRate},m-1})_{\uparrow}. \end{aligned} \quad (19)$$

Also, $\delta_i^{\text{minLatency},m-1}$ and $\delta_i^{\text{maxRate},m-1}$ are equal to

$$\begin{aligned} \sum_{j,j \neq i} \left[\phi_{i,j}^{\text{minLatency}} \circ \beta_i^{\text{affine},m-1} - \alpha_j \otimes \beta_j^{\text{affine},m-1} \right]^+ \text{ and} \\ \sum_{j,j \neq i} \left[\phi_{i,j}^{\text{maxRate}} \circ \beta_i^{\text{affine},m-1} - \alpha_j \otimes \beta_j^{\text{affine},m-1} \right]^+, \end{aligned} \quad (20)$$

respectively. All the good properties presented for Corollary 3 are true here as well (see Fig. 7).

In Fig. 8, we apply the three methods in Theorem 3 and Corollaries 3 and 4 to one illustrative example. The obtained strict service curves of all schemes become stationary in the horizon of interest, hence the schemes stop. The scheme of Theorem 3 becomes stationary after 3 iterations, and the schemes of Corollaries 3 and 4 become stationary after 16 iterations.

VI. PROOFS

A. Proof of Theorem 1

The idea of the proof is as follows. We consider a backlogged period $(s, t]$ of flow of interest i , and we let p be the

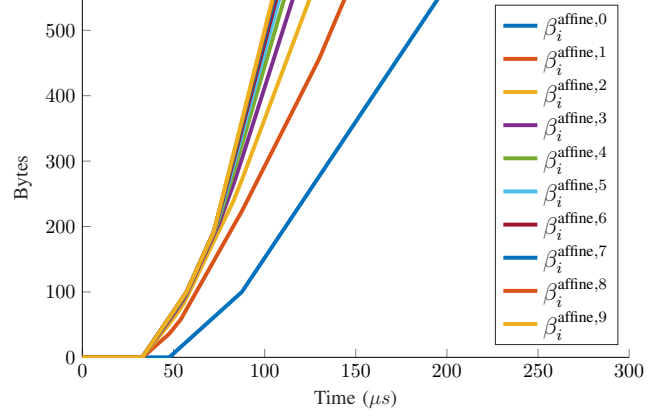


Fig. 7: Strict service curves of Corollary 4 for flow 2 of the example of Fig. 5. The iterative scheme is stopped when the computed delay bounds for all flows decreases by less than $0.25\mu\text{s}$. The delay bounds obtained with the strict service curves in the last iteration are the same as with Corollary 3; however, the strict service curves of the first iterations slightly differ from those of Fig. 6. Obtained with MATLAB.

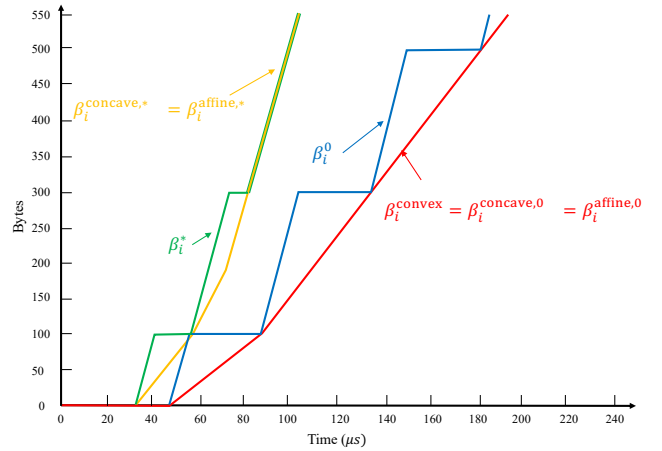


Fig. 8: Strict service curves of Theorem 3 and Corollaries 3 and 4 for flow 2 of the example of Fig. 5. When applying these schemes, after a number of iterations, the strict service curves of all flows become stationary in the horizon of the figure that is larger than the sufficient horizon $t^* = 200\mu\text{s}$. Therefore, the schemes stop, and we denote these strict service curves by $*$. Obtained with the RTaW online tool.

number of complete service opportunities for flow i in this period, where a complete service opportunity starts at line 7 and ends at line 15 of Algorithm 1. p is upper bounded by a function of the amount of service received by flow i , given in (22). Given this, the amount of service received by every other flow j is upper bounded by a function of the amount of service received by flow i , given in (24). Using this result gives an implicit inequality for the total amount of service in (26). By using the technique of pseudo-inverse, this inequality is inverted and provides a lower bound for the amount of service received by the flow of interest.

From [3, Sub-goal 1], the number p of complete service

opportunities for flow of interest, i , in $(s, t]$, satisfies

$$D_i(t) - D_i(s) \geq pQ_i - d_i^{\max} \quad (21)$$

Therefore, as p is integer:

$$p \leq \left\lfloor \frac{D_i(t) - D_i(s) + d_i^{\max}}{Q_i} \right\rfloor \quad (22)$$

Furthermore, it is shown in the proof of [3, Sub-goal 2] that

$$D_j(t) - D_j(s) \leq (p+1)Q_j + d_j^{\max} \quad (23)$$

Using (22) we obtain

$$D_j(t) - D_j(s) \leq \underbrace{\left\lfloor \frac{D_i(t) - D_i(s) + d_i^{\max}}{Q_i} \right\rfloor}_{\phi_{i,j}(D_i(t) - D_i(s))} Q_j + (Q_j + d_j^{\max}) \quad (24)$$

Next, as the interval $(s, t]$ is a backlogged period, by the definition of the strict service curve for the aggregate of flows we have

$$\beta(t-s) \leq (D_i(t) - D_i(s)) + \sum_{j \neq i} (D_j(t) - D_j(s)) \quad (25)$$

We upper bound the amount of service to every other flow j by applying (24):

$$\beta(t-s) \leq (D_i(t) - D_i(s)) + \underbrace{\sum_{j, j \neq i} \phi_{i,j}(D_i(t) - D_i(s))}_{\psi_i(D_i(t) - D_i(s))} \quad (26)$$

Then we invert (26) using (1) and obtain

$$D_i(t) - D_i(s) \geq \psi_i^\downarrow(\beta(t-s)) \quad (27)$$

Lastly, we want to compute ψ_i^\downarrow . Observe that, by plugging $\phi_{i,j}$ in (5), $\psi_i(x) = x + \left\lfloor \frac{x + d_i^{\max}}{Q_i} \right\rfloor \left(\sum_{j \neq i} Q_j \right) + \sum_{j \neq i} (Q_j + d_j^{\max})$; as there is no plateau in ψ_i , its lower-pseudo inverse is simply its inverse which is obtained by flipping the axis (Fig. 9), and is obtained as

$$\begin{aligned} \psi_i^\downarrow(x) &= (\lambda_1 \otimes \nu_{Q_i, Q_{\text{tot}}}) \left([x - \psi_i(Q_i - d_i^{\max})]^+ \right) \\ &+ \min \left(\left[x - \sum_{j \neq i} (Q_j + d_j^{\max}) \right]^+, Q_i - d_i^{\max} \right) \end{aligned} \quad (28)$$

ψ_i^\downarrow is illustrated in Fig. 9. In (28), observe that the term with min expresses the finite part at the beginning between 0 and $\psi_i(Q_i - d_i^{\max})$; also, observe that the term with the min-plus convolution expresses the rest (see Fig. 1.b with $a = Q_i$ and $b = Q_{\text{tot}} = \sum_{j=1}^n Q_j$). \square

B. Proof of Theorem 2

First observe that, since $\phi'_{i,j} \in \mathcal{F}$, it follows that $\beta'_i \in \mathcal{F}$. Second, as for every $j \neq i$, $\phi_{i,j} \leq \phi'_{i,j}$, we have $\psi_i \leq \psi'_i$. In [13, Sec. 10.1], it is shown that $\forall f, g \in \mathcal{F}, f \geq g \Rightarrow f^\downarrow \leq g^\downarrow$. Applying this with $f = \psi'_i$ and $g = \psi_i$ gives that $\psi_i^\downarrow \leq \psi'_i^\downarrow$. It follows $\beta'_i(t) = \psi_i^\downarrow(\beta(t)) \leq \psi'_i^\downarrow(\beta(t)) = \beta_i(t)$. The conclusion follows from the fact that any lower bound in \mathcal{F} of a strict service curve is a strict service curve. \square

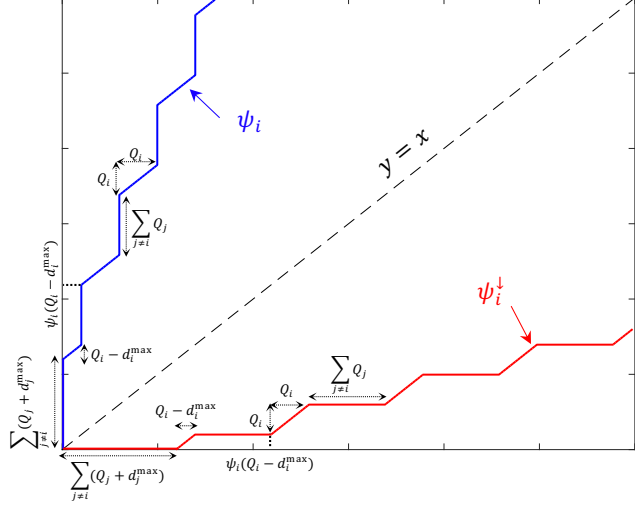


Fig. 9: Illustration of ψ_i and its lower-pseudo inverse ψ_i^\downarrow , defined in (5) and (28), respectively.

C. Proof of Theorem 3

The idea of the proof is as follows. First we give a lemma on the operation of DRR, which follows from some of the results in the proof of Theorem 1. Then, we study the mapping that transforms the collection of service curves in one iteration of the theorem. We show that this mapping transforms a collection of strict service curves into another collection of strict service curves and that it is isotone. Last, we show that the output of the first iteration is an upper bound of the input, which, using isotony, concludes the proof.

Lemma 1. Assume that the output of flow j is constrained by an arrival curve $\alpha_j^* \in \mathcal{F}$, i.e., D_j is constrained by α_j^* . Then, for any backlogged period $(s, t]$ of flow i , the amount of service for flow $j \neq i$ in $(s, t]$ is upper bounded by

$$\begin{aligned} D_j(t) - D_j(s) &\leq \phi_{i,j}(D_i(t) - D_i(s)) \\ &- [\phi_{i,j}(D_i(t) - D_i(s)) - \alpha_j^*(t-s)]^+ \end{aligned} \quad (29)$$

where $\phi_{i,j}$ is defined in (6).

Proof. First, by (24), we have

$$D_j(t) - D_j(s) \leq \phi_{i,j}(D_i(t) - D_i(s)) \quad (30)$$

Second, as α_j^* is an arrival curve for D_j , we have

$$D_j(t) - D_j(s) \leq \alpha_j^*(t-s) \quad (31)$$

Combine (30) and (31) to derive

$$D_j(t) - D_j(s) \leq \min(\phi_{i,j}(D_i(t) - D_i(s)), \alpha_j^*(t-s)) \quad (32)$$

Observe that for any two numbers x, y , we have $\min(x, y) = x - [x - y]^+$. Apply this to the previous equation and obtain (29). \square

The proof continues with the study of the mapping $\Pi : \mathcal{F}^n \rightarrow \mathcal{F}^n$ that transforms $(\beta_1^{m-1}, \dots, \beta_n^{m-1})$ into $(\beta_1^m, \dots, \beta_n^m)$, namely, $\Pi(\beta_1, \dots, \beta_n) = (\beta'_1, \dots, \beta'_n)$ with

$$\beta'_i = \gamma_i \circ \left(\beta + \sum_{j,j \neq i} [\phi_{i,j} \circ \beta_i - (\alpha_j \circ \beta_j)]^+ \right)_\uparrow \quad (33)$$

Lemma 2. Let $(\beta_1, \dots, \beta_n) \in \mathcal{F}^n$ and $(\beta'_1, \dots, \beta'_n) = \Pi(\beta_1, \dots, \beta_n)$. Assume that, for every i , β_i is a strict service curve for flow i . Then β'_i is also a strict service curve for every flow i .

Proof. Consider a backlogged period $(s, t]$ of flow of interest i . As the interval $(s, t]$ is a backlogged period, and since β is a strict service curve for the aggregate of flows, we have

$$\beta(t-s) \leq (D_i(t) - D_i(s)) + \sum_{j \neq i} (D_j(t) - D_j(s)) \quad (34)$$

Since β_j is a service curve for flow j , it follows that $\alpha_j \circ \beta_j$ is an arrival curve for the output of flow j (Section II-A). Apply Lemma 1 with $\alpha_j^* = \alpha_j \circ \beta_j$ and obtain

$$\begin{aligned} & \beta(t-s) + \sum_{j,j \neq i} [\phi_{i,j} (D_i(t) - D_i(s)) - (\alpha_j \circ \beta_j)(t-s)]^+ \\ & \leq (D_i(t) - D_i(s)) + \underbrace{\sum_{j,j \neq i} \phi_{i,j} (D_i(t) - D_i(s))}_{\psi_i(D_i(t) - D_i(s))} \end{aligned} \quad (35)$$

Then, as β_i is a strict service curve for flow i , we have $D_i(t) - D_i(s) \geq \beta_i(t-s)$. Combining it with the above equation it follows that

$$\begin{aligned} & \beta(t-s) + \sum_{j,j \neq i} [\phi_{i,j} (\beta_i(t-s)) - \alpha_j^*(t-s)]^+ \\ & \leq \psi_i (D_i(t) - D_i(s)) \end{aligned} \quad (36)$$

As ψ_i is an increasing function, it follows that the right-hand side is an increasing function of $(t-s)$. Then, by applying [11, Lemma 3.1], it follows that the inequality holds for the non-decreasing closure of the left-hand side (with respect to $t-s$), namely

$$\begin{aligned} & \left(\beta + \sum_{j,j \neq i} [\phi_{i,j} \circ \beta_i - \alpha_j \circ \beta_j]^+ \right)_\uparrow (t-s) \\ & \leq \psi_i (D_i(t) - D_i(s)) \end{aligned} \quad (37)$$

Then, we use the lower pseudo-inverse technique to invert (37) as in (1):

$$D_i(t) - D_i(s) \geq \psi_i^\downarrow \left(\left(\beta + \sum_{j,j \neq i} [\phi_{i,j} \circ \beta_i - \alpha_j \circ \beta_j]^+ \right)_\uparrow (t-s) \right) \quad (38)$$

Hence, the right-hand side is a strict service curve for flow i . Observe that $\gamma_i = \psi_i^\downarrow$. \square

By Theorem 1, β_i^0 is a strict service curve for flow i ; it follows from Lemma 2 that, for every $m \geq 0$, β_i^m is also a strict service curve for flow i , i.e., item 1) is shown. To show item 2), we first show that Π is isotone.

Lemma 3. Let $(\beta_1, \dots, \beta_n) \in \mathcal{F}^n$ and $(\hat{\beta}_1, \dots, \hat{\beta}_n) \in \mathcal{F}^n$ such that $\beta_i \leq \hat{\beta}_i$ for every i . Let $(\beta'_1, \dots, \beta'_n) = \Pi(\beta_1, \dots, \beta_n)$ and $(\hat{\beta}'_1, \dots, \hat{\beta}'_n) = \Pi(\hat{\beta}_1, \dots, \hat{\beta}_n)$. Then $\beta'_i \leq \hat{\beta}'_i$ for every i .

Proof. Let $\delta_i = \sum_{j,j \neq i} [\phi_{i,j} \circ \beta_i - (\alpha_j \circ \beta_j)]^+$ and $\hat{\delta}_i = \sum_{j,j \neq i} [\phi_{i,j} \circ \hat{\beta}_i - (\alpha_j \circ \beta_j)]^+$, so that $\beta'_i = \gamma_i \circ (\beta + \delta_i)_\uparrow$ and $\hat{\beta}'_i = \gamma_i \circ (\beta + \hat{\delta}_i)_\uparrow$.

As $\phi_{i,j}$ is wide-sense increasing we have $\phi_{i,j} \circ \beta_i \leq \phi_{i,j} \circ \hat{\beta}_i$. Also, by [11, Section 2.4], we have $\alpha_j \circ \beta_j \geq \alpha_j \circ \hat{\beta}_j$. Hence,

$$\phi_{i,j} \circ \beta_i - (\alpha_j \circ \beta_j) \leq \phi_{i,j} \circ \hat{\beta}_i - (\alpha_j \circ \hat{\beta}_j), \text{ thus} \quad (39)$$

$$[\phi_{i,j} \circ \beta_i - (\alpha_j \circ \beta_j)]^+ \leq [\phi_{i,j} \circ \hat{\beta}_i - (\alpha_j \circ \hat{\beta}_j)]^+ \quad (40)$$

The above inequality holds for every $j \neq i$ thus $\delta_i \leq \hat{\delta}_i$ and

$$\beta + \delta_i \leq \beta + \hat{\delta}_i \leq (\beta + \hat{\delta}_i)_\uparrow \quad (41)$$

Then, by applying [11, Lemma 3.1], it follows that the above inequality holds for the non-decreasing closure of the left-hand side:

$$(\beta + \delta_i)_\uparrow \leq (\beta + \hat{\delta}_i)_\uparrow \quad (42)$$

Using the above inequality and as γ_i is increasing, it follows $\beta'_i = \gamma_i \circ ((\beta + \delta_i)_\uparrow) \leq \gamma_i \circ ((\beta + \hat{\delta}_i)_\uparrow) = \hat{\beta}'_i$. \square

We next show that $\beta_i^0 \leq \beta_i^1$ for every i . As $\delta_i^0 \geq 0$, we have

$$\beta_i^0 = \gamma_i \circ \beta \leq \gamma_i \circ (\beta + \delta_i^0)_\uparrow = \beta_i^1 \quad (43)$$

Then, using the above equation and Lemma 3, we obtain that $\beta_i^m \leq \beta_i^{m+1}$ for every m and i , i.e., item 2) of Theorem 3 holds. \square

VII. NUMERICAL EVALUATION

In this section, we compare the obtained delay bounds by using our new strict service curves for DRR, presented in Sections IV and V, to those of Boyer et al. and to Soni et al. We use all network configurations that were presented by Soni et al. [7], specifically, two illustration networks and an industrial-sized one. For the illustration networks, we use the exact same configuration of flows and switches that Soni et al. use. For the industrial-sized network, Soni kindly replied to our e-mail request by saying that, for confidentiality reasons, they do not have the rights to provide more details about the network configuration than what is already given in [7]. Consequently, we use the same network but randomly choose the missing information (explained in detail in Section VII-B).

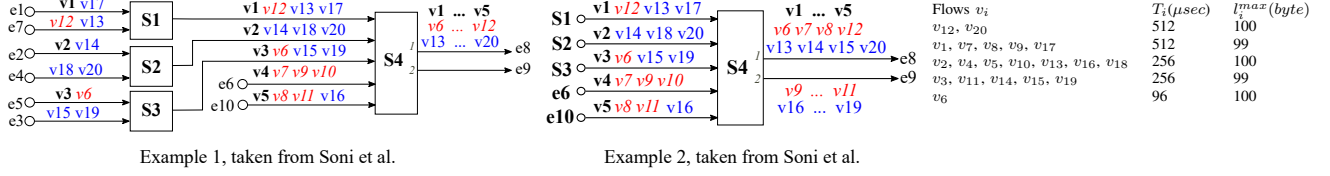


Fig. 10: Networks of Examples 1 and 2, taken from Soni et al [7]. Examples 1 and 2 differ only by the configuration of the switch S_4 .

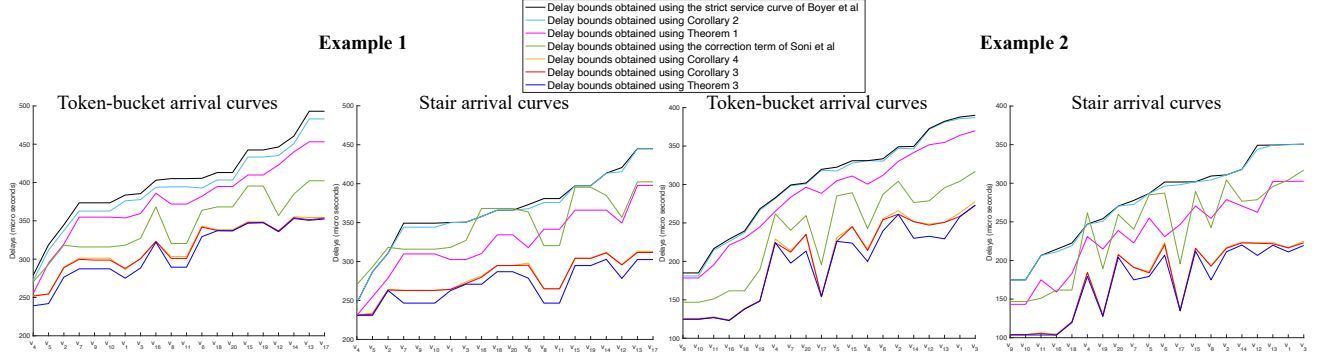


Fig. 11: Delay bounds of flow v_1, v_2, \dots, v_{20} in Example 1 and Example 2 of Fig. 10. In each example, we follow [7] and assume once that flows are constrained by token-bucket arrival curves, and once that flows are constrained by stair arrival curves. The delay bounds of Soni et al. are taken from [7], and other results are computed with the RTaW online tool. First, delay bounds obtained with our new strict service curves for DRR, with no knowledge on the interfering traffic, are always better than those of Boyer et al. Second, delay bounds obtained with our new strict service curve for DRR that accounts for arrival curve of interfering flows are always better than those of Soni et al. The delay bounds obtained with Corollary 4 are visually indistinguishable from those obtained with Corollary 3. In each plot, flows are ordered by values of Boyer’s bound.

A. Illustration Networks

Example 1 and 2 are illustrated in Fig. 10. We use the exact same network with the exact same configuration for flows and switches that Soni et al. use in [7]. Examples 1 and 2 differ only by the configuration of the switch S_4 . Flows $\{v_1 \dots v_5\}$, $\{v_6 \dots v_{12}\}$, and $\{v_{13} \dots v_{20}\}$ are assigned to class C_1 , C_2 , and C_3 , respectively. There is one DRR scheduler at every switch output port; what we called “flow” earlier in the paper corresponds here to a class hence $n = 3$. Inside a class, arbitration is FIFO (all packets of all flows of a given class are in the same FIFO queue). Also, as in [7], we assume that queuing is on output ports only. All classes have the same quantum equal to 199 bytes. The rate of the links are equal to $c = 100$ Mb/s, and every switch S_i has a switching latency equal to $16\mu\text{s}$. Every flow v_i has a maximum packet size l_i^{\max} and minimum packet arrival T_i . Hence, flow v_i is constrained by a token-bucket arrival curve with rate equal to $\frac{l_i^{\max}}{T_i}$ and burst equal to l_i^{\max} ; also, it is constrained by a stair arrival curve given by $l_i^{\max} \lceil \frac{t}{T_i} \rceil$.

For the sake of comparison, as Soni et al. do not consider grouping and offsets (explained in Section II-D) in these two examples, we also do not consider them. This means that the arrival curve we use for bounding the input of a class at a switch is simply equal to the arrival curves expressed for every member flow. Arrival curves are propagated using the delay bounds computed at the upstream nodes. We illustrate

the reported values in [7] for the delay bounds of Soni et al. For the other results, we use the RTaW online tool (Fig. 11). As explained in Section II-A, RTaW provides all the necessary operations to implement our new strict service curves for DRR. First, observe that delay bounds obtained with our new strict service curves for DRR, with no knowledge on the interfering traffic, are always better than those of Boyer et al. Second, delay bounds obtained with our new strict service curve for DRR that accounts for arrival curve of interfering flows are always better than the (incorrect) ones of Soni et al. The obtained delay bounds of Theorem 3 are better than or equal to those of Corollaries 3 and 4; also, they are almost equal for both Corollaries. When using token-bucket arrival curves, the run-time of Theorem 3 is in the order of 3 minutes; for Corollaries 3 and 4 it is in the order of 30 seconds; when using stair arrival curves, the run-time of Theorem 3 is in the order of 5 minutes; for Corollaries 3 and 4 it is in the order of 1 minute and 45 seconds, respectively.

B. Industrial-Sized Network

We use the network of Fig. 12; it corresponds to a test configuration provided by Airbus in [19]. The industrial-sized case study that Soni et al. use in [7] is based on this network in [21]. We combine the available information in both papers to understand this network. It includes 96 end-systems, 8 switches, 984 flows, and 6412 possible paths. The rate of

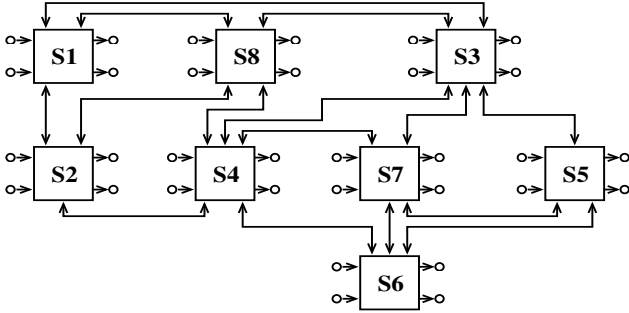


Fig. 12: Industrial-sized network topology. The figure is taken from [21].

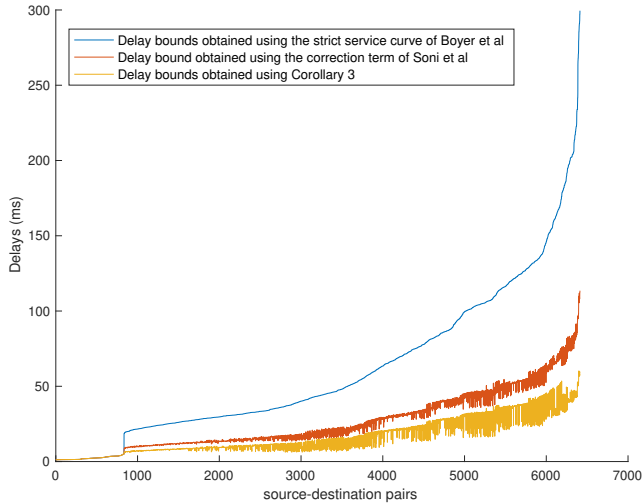


Fig. 13: Delay bounds of the industrial case for all source-destination pairs in the system. The delay bounds obtained with Corollary 4 are visually indistinguishable from those obtained with Corollary 3 and are not displayed. Source-destination paths are ordered by values of Boyer’s bound.

the links are equal to $c = 100$ Mb/s, and every switch S_i has a switching latency equal to $16\mu s$. We find that each switch has 6 input and 6 output end-systems. Three classes of flows are considered: critical flows, multimedia flows, and best-effort flows. There is one DRR scheduler at every switch output port with $n = 3$ classes. At every DRR scheduler, the quanta are 3070 bytes for the critical class, 1535 bytes for the multimedia class, and 1535 bytes for the best-effort class. 128 multicast flows, with 834 destinations, are critical; they have a maximum packet-size equal to 150 bytes and their minimum packet arrival time is between 4 and 128ms. 500 multicast flows, with 3845 destinations, are multimedia and their class has a quantum equal to 1535 bytes; they have a maximum packet-size equal to 500 bytes; and their minimum packet-arrival time is between 2 and 128ms. 266 multicast flows, with 1733 destinations, are best-effort; they have a maximum packet-size equal to 1535 bytes; and their minimum packet arrival time is between 2 and 128ms. For every flow, the path from the source to a destination can traverse at most 4 switches. Specifically, 1797, 2787, 1537, and 291 source-

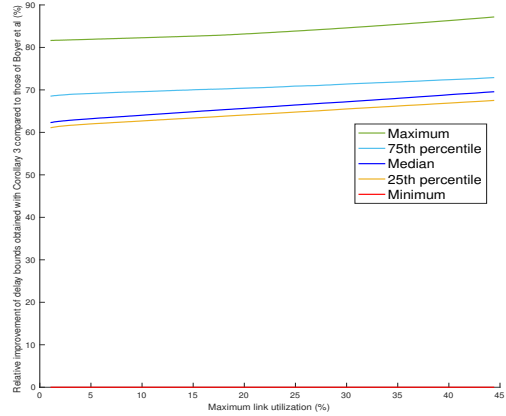


Fig. 14: Relative improvements of delay bounds obtained with Corollary 3 compared to those of Boyer et al. for one random configuration of the industrial case, when only varying the link rate $c \in [36.8, 1600]$ Mb/s to simulate different workloads (the rate c cannot be less than 36.8 Mb/s as the delay bound of some source-destination pairs become unbounded.). For each choice of the link rate c , the relative improvements for all source-destination pairs are computed; then, the maximum, the 75th percentile, the median, the 25th percentile, and the minimum of them are plotted. The link between S_4 and S_3 achieves the maximum utilization.

destination paths have 1, 2, 3, and 4 hops, respectively. We choose the paths randomly and satisfy all these constraints.

Due to the limited expressiveness of the language used by the RTaW online tool, we could not implement the industrial-size network there. Therefore, we used MATLAB, which has the required expressiveness, but does not have support for min-plus computations; hence, we could only implement Corollaries 3 and 4. The obtained delay bounds and run-times (103 and 106 minutes, respectively, on a 2.6 GHz 6-Core Intel Core i7 computer) are almost the same for both Corollaries. We also computed the delay bounds obtained with the strict service curve of Boyer et al. and with the correction term of Soni et al. In all cases, and as in [7], the arrival curve used for bounding the input of a class at a switch incorporates the effects of delay bounds computed upstream, as well as grouping (line shaping) and offset (Section II-D); and the offsets are such that they create maximum separation, as with [7]. We find that our bounds significantly improve upon the existing bounds, even the incorrect ones (Fig. 13). Also, we vary the link rate c to simulate different workloads (Fig. 14).

VIII. CONCLUSION

The method of the pseudo-inverse enables us to perform a detailed analysis of DRR and to obtain strict service curves that significantly improve the previous results. Our results use the network calculus approach and are mathematically proven, unlike some previous delay bounds that we have proved to be incorrect. Our method assumes that the aggregate service provided to the DRR subsystem is modelled with a strict service curve. Therefore it can be recursively applied to hierarchical DRR schedulers as found, for instance, with class-based queuing.

REFERENCES

- [1] M. Shreedhar and G. Varghese, "Efficient fair queuing using deficit round-robin," *IEEE/ACM Transactions on Networking*, vol. 4, no. 3, pp. 375–385, 1996.
- [2] L. Lenzi, E. Mingozzi, and G. Stea, "Aliquem: a novel drr implementation to achieve better latency and fairness at $o(1)$ complexity," in *IEEE 2002 Tenth IEEE International Workshop on Quality of Service (Cat. No.02EX564)*, 2002, pp. 77–86.
- [3] M. Boyer, G. Stea, and W. M. Sofack, "Deficit round robin with network calculus," in *6th International ICST Conference on Performance Evaluation Methodologies and Tools*, 2012, pp. 138–147.
- [4] S. S. Kanhere and H. Sethu, "On the latency bound of deficit round robin," in *Proceedings. Eleventh International Conference on Computer Communications and Networks*, 2002, pp. 548–553.
- [5] D. Stiliadis, "Traffic scheduling in packet-switched networks: Analysis, design, and implementation," Ph.D. dissertation, 1996, aAI9637506.
- [6] L. Lenzi, E. Mingozzi, and G. Stea, "Full exploitation of the deficit round robin capabilities by efficient implementation and parameter tuning."
- [7] A. Soni, X. Li, J. Scharbag, and C. Fraboul, "Optimizing network calculus for switched ethernet network with deficit round robin," in *2018 IEEE Real-Time Systems Symposium (RTSS)*, 2018, pp. 300–311.
- [8] S. M. Tabatabaee, J. Y. Le Boudec, and M. Boyer, "Interleaved weighted round-robin: A network calculus analysis," in *2020 32nd International Teletraffic Congress (ITC 32)*, 2020, pp. 64–72.
- [9] J.-Y. Le Boudec and P. Thiran, *Network Calculus: A Theory of Deterministic Queuing Systems for the Internet*. Springer Science & Business Media, 2001, vol. 2050.
- [10] C. S. Chang, *Performance Guarantees in Communication Networks*. New York: Springer-Verlag, 2000.
- [11] A. Bouillard, M. Boyer, and E. Le Corronc, *Deterministic Network Calculus: From Theory to Practical Implementation*. Wiley-ISTE.
- [12] K. Lampka, S. Bondorf, and J. Schmitt, "Achieving efficiency without sacrificing model accuracy: Network calculus on compact domains," in *2016 IEEE 24th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS)*, 2016, pp. 313–318.
- [13] J. Liebeherr, "Duality of the max-plus and min-plus network calculus," *Foundations and Trends in Networking*, vol. 11, no. 3-4, pp. 139–282, 2017.
- [14] "RealTime-at-Work online Min-Plus interpreter for Network Calculus," <https://www.realtimeatwork.com/minplus-playground>, accessed: year-month-day.
- [15] A. Charny and J.-Y. Le Boudec, "Delay bounds in a network with aggregate scheduling," in *Quality of Future Internet Services*. Springer, 2000, pp. 1–13.
- [16] D. B. Chokshi and P. Bhaduri, "Modeling fixed priority non-preemptive scheduling with real-time calculus," in *2008 14th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications*, Aug 2008, pp. 387–392.
- [17] L. Thiele, S. Chakraborty, and M. Naedele, "Real-time calculus for scheduling hard real-time systems," in *2000 IEEE International Symposium on Circuits and Systems (ISCAS)*, vol. 4, May 2000, pp. 101–104 vol.4.
- [18] A. Mifdaoui and T. Leydier, "Beyond the Accuracy-Complexity Tradeoffs of CompositionalAnalyses using Network Calculus for Complex Networks," in *10th International Workshop on Compositional Theory and Technology for Real-Time Embedded Systems (co-located with RTSS 2017)*, Paris, France, December 2017, pp. 1–8. [Online]. Available: <https://hal.archives-ouvertes.fr/hal-01690096>
- [19] J. Grieu, "Analyse et évaluation de techniques de commutation ethernet pour l'interconnexion des systèmes avioniques," September 2004. [Online]. Available: <https://oatao.univ-toulouse.fr/7385/>
- [20] A. Bouillard, "Trade-off between accuracy and tractability of network calculus in fifo networks," 2020.
- [21] H. Charara, J. Scharbag, J. Ermont, and C. Fraboul, "Methods for bounding end-to-end delays on an afdx network," in *18th Euromicro Conference on Real-Time Systems (ECRTS'06)*, 2006, pp. 10 pp.–202.