Imperfect Bandwidth-Sharing Policies using Network Calculus

Anne Bouillard
Huawei Technologies France SASU,
Boulogne-Billancourt, France
anne.bouillard@huawei.com

***Abstract*—Bandwidth-sharing policies aim at enforcing fairness among several classes of traffic by reserving one share of the available bandwidth for each class. They recently attracted a lot of attention, in particular in the context of time-sensitive networking. One advantage of these policies is their simplicity of implementation, especially the Round-Robin policies. Moreover, if a traffic class does not use all its reserved bandwidth, the other classes can use the unused part. Recent works greatly improve the delay bounds by considering this phenomenon for the Deficit Round-Robin policy, when the knowledge about the incoming traffic is available.**

There was also an attempt to use this approach for Weighted Round-Robin policy. Unfortunately, this was incorrect, mainly due to the variability of the packet lengths and the share of the bandwidth is not perfectly known. In this paper, we propose a generalization of bandwidth-sharing policies, that we call *imperfect* bandwidth sharing. We compute per-class service guarantees that correct and improves the state of the art on Round-Robin policies, and assess through numerical experiments the algorithmic and performance gain of our solution.

***Index Terms*—Network calculus, weighted round robin, scheduling, performance evaluation.**

I. Introduction

Bandwidth-sharing policies are service policies that share the available bandwidth of a server among different classes of traffic. The aim is to enforce fairness, by preventing a class from starving too long. One advantage of these policies is the ease of implementation. Representative example of such policies are the Round-Robin (RR) ones [1], that are used in the context of Time-sensitive networking [2], Networks on Chips [3]...

RR policies are used when traffic is separated in multiple classes. Packets of each class are served in rounds: while RR serves one packet each class at a time, Weighted Round-Robin (WRR) can serve several packets of a class at each round, introducing more flexibility for the bandwidth allocated to each class. Interleaved WRR (IWRR) interleaves the service of packets of the classes inside each round to reduce the per-class inter-service time. In Deficit Round-Robin (DRR), the amount of data served at each round is based on the quantity of data rather than on a packet count. This latter policy enforces a perfect share of the bandwidth: for each class, the bandwidth reserved, and guaranteed, is proportional to its per-round allocated service. This is similar for WRR when packets have a per-class fixed packet length, but not for variable packet lengths, and we will call this policy *imperfect bandwidth-sharing*.

ISBN 978-3-903176-57-7 © 2023 IFIP

One proposed analysis of bandwidth-sharing policies is Network Calculus [4], whose aim is to compute performance (end-to-end delays or buffer occupancy) upper bounds in networks. Broadly speaking, the analysis of a large topology with several traffic classes is done in two steps. The first step is modeling, for each server (for example a router, a switch, the output port of a switch...) and each class a function characterizing its service guarantee, called a *service curve*. Usually, within each class, the scheduling is FIFO (First-in-First-Out). After characterizing each per-class service, FIFO per-class networks are obtained, and the second step is the analysis of these FIFO networks. Several methods have been studied, for example TFA++ [5], [6], based on the algebraic properties of Network Calculus, PLP [7], based on linear programming.

In this paper, we focus on the first step: finding per-class service curves for (imperfect) bandwidth-sharing policies. Perfect bandwidth sharing policies have already received a lot of attention: Generalized Processor-Sharing (GPS) [8], an idealization of bandwidth-sharing policies, was among the first service policies to be analyzed, for constant-rate servers. The analysis was recently generalized [9], [10]. Strict service curves have first been derived for DRR [11] and WRR [12] and IWRR [13] without taking into account the characterization of the incoming traffic. Recently approaches have been adapted to account for the incoming traffic [10], [14], demonstrating great improvement of the delay bounds in realistic topologies [15].

An attempt to adapt the approach of [10] to WRR was done in [16]. The authors assume that the worst-case is to consider only small packet lengths for the flow of interest and large packets for the cross traffic. In fact, large packets of the flow of interest can create large backlogs for the cross-traffic, which in turn increases the delays of the flow of interest. The authors have recently proposed a corrected version [17] that is more similar to [15]. However, this new approach can be very pessimistic as it can lead to infinite delay bounds even if the system is stable.

Here is the list of contributions of the paper:

1) We provide a counter-example to [16], demonstrating that WRR is not a (perfect) bandwidth-sharing policy.
2) We define *imperfect* bandwidth-sharing policies, that unifies the approaches of [10] and [15].

3) We derive strict service curves for WRR and IWRR that corrects [16] and improves [17]. In particular, we obtain finite bounds whenever the server is stable.
4) We improve the service curve of [17] for IWRR.
5) We give a heuristic that runs in quadratic time (instead of exponential) and is a good approximation of WRR, especially in for constant-length packets.

The rest of the paper is organized as follows: in Section II, we define the necessary background of Network Calculus. In Section III, we detail some approaches of the state of the art of bandwidth-sharing policies. In Section IV, we define *imperfect* bandwidth-sharing policies, and derive per-class service curves. The algorithm and an approximation heuristic is derived in Section V. Finally, we demonstrate the efficiency of the analysis though numerical evaluation in Section VI.

II. Network Calculus framework

In this section, we present the necessary background for our analysis. More details can be found in [18], [12].

We consider a server crossed by n classes of traffic numbered from 1 to n. We denote by $\mathbb{N}_n$ the set $\{1,\ldots,n\}$ and for all subset $M \subseteq \mathbb{N}_n$, its complementary in $\mathbb{N}_n$ is denoted $\overline{M}(=\mathbb{N}_n \setminus M)$. For a sequence $(x_i)_{i\in\mathbb{N}_n}$, and $M \subseteq \mathbb{N}_n$, we denote $x^M = \sum_{j\in M} x_j$.

1) Arrival and departure processes: For each class i, $A_i : \mathbb{R}_+ \to \mathbb{R}_+$ is its cumulative arrival process: $\forall t \geq 0$, $A_i(t)$ is the amount a data of class i arrived up to time t in the server (*i. e.,* in the time interval $[0,t)$). By abuse of the notation, and for compactness, we will also denote by $A_i(s,t) = A_i(t) - A_i(s)$ the amount of data arrived in the time interval $[s,t)$.

We say that A_i is constrained by the arrival curve α_i if $\forall s \leq t$, $A_i(s,t) \leq \alpha_i(t-s)$. One classical example of arrival curve is the token-bucket γ_{b_i,r_i} with arrival rate r_i and burst b_i: $\alpha_i = \gamma_{b_i,r_i} : 0 \mapsto 0; t > 0 \mapsto b_i + r_i t$.

We denote by A the aggregate arrival process: $A = \sum_{i=1}^n A_i$. More generally, for any subset M of $\mathbb{N}_n$, $A^M = \sum_{i\in M} A_i$ is the aggregated process of classes in M.

Let us denote by $D_i : \mathbb{R}_+ \to \mathbb{R}_+$ the departure cumulative process of class i. We assume causality and no loss: no data is lost or created in the server, and then, $D_i \leq A_i$. Similarly, D is the aggregation of the departure process: $D = \sum_{i=1}^n D_i$, and $D^M = \sum_{i\in M} D_i$.

The traffic of data is usually not fluid, but made of packets. For each class i, we denote by $\ell_i^{\min}$ and $\ell_i^{\max}$ minimum and maximum packet length of class i.

2) Server and service process: The network calculus framework offers a variety of service curves, and one difficulty is choosing the most suitable to the context. In this paper, we will use variable capacity nodes (vcn) with a slight adaptation of the terminology and strict service curves.

A server is a vcn if there exists a function $C : \mathbb{R}_+ \to \mathbb{R}_+$ such that for all $t \geq 0$, $C(t)$ is the service is offered to the incoming traffic up to time t. Similar to the arrival processes, $C(s,t) = C(t) - C(s)$ is the service offered during the interval $[s,t)$. More precisely, if the system is never empty in the time interval $(s,t]$, *i.e.* $A(u) > D(u)$ for all $u \in (s,t]$, then the quantity of data exiting the system during $[s,t)$ is exactly $D(s,t) = C(s,t)$. The interval $(s,t]$ is called a *backlogged period*. Note that the interval is open on the left because one could have $A(s) = D(s)$, which would be the *start* of the backlogged period.

Often, $C(s,t)$ is not exactly known. Also, we see from the definition that it is only important to know $C(s,t)$ during backlogged period. Otherwise, C can be fixed arbitrarily. We say that a server *guarantees a vcn C* if for all $s \leq t$ such that $(s,t]$ is a backlogged period, $D(s,t) \geq C(s,t)$. Note that in this case, C is not necessarily deduced from a univariate function C such that $C(s,t) = C(t) - C(t)$, but we can always assume super-additivity: $C(s,u) \geq C(s,t) + C(t,u)$ for all $0 \leq s \leq t \leq u$.

We assume that the server offers a finite long-term service rate: for all $s \geq 0$, $\lim_{t\to\infty} \frac{C(s,t)}{t-s} = R < \infty$. We also assume the existence of a function $\beta : \mathbb{R}_+ \to \mathbb{R}_+$ such that $\forall s \leq t$, $C(s,t) \geq \beta(t-s)$. This function β is called a strict service curve[1]. One classical example of service curve is the *rate-latency* service curve with rate R and latency T, where $\beta_{R,T} : t \mapsto R(t-T)_+$ and $(x)_+ = \max(0,x)$.

3) Performance guarantees: The aim of network calculus is to compute performance guarantees, such as delay or backlog (or buffer usage) upper bounds from the arrival and service curves, as well as guarantees for the departure processes (used to propagate the computation for multi-server systems). The following theorem is fundamental in Network Calculus.

Theorem 1. *Assume a server offering a service curve β crossed by a class of data with arrival curve α.*

- *The backlog of the system is upper bounded by* $b = \sup_{t\geq 0} \alpha(t) - \beta(t) = vDev(\alpha,\beta)$*;*
- *The* virtual *delay is upper bounded by* $d = \inf\{s \geq 0 \mid \forall t \in \mathbb{R}_+,\ \alpha(t) \leq \beta(t+s)\}$*;*
- *An arrival curve of the departure process is* $\alpha \oslash \beta(t) = \sup_{s\geq 0} \alpha(t+s) - \beta(s)$.

The delay bound is valid if we assume that data are served in their arrival order, which is not always the case, especially for multi-class systems: the delay bound depends on the scheduling policy. One way to tackle the problem of finding valid delay bounds for each class of traffic is finding service curves for each class of traffic. Indeed, inside each class of traffic, the service policy is FIFO, and the delay bound can be computed with Theorem 1.

The main goal of this paper is then to compute per-class (strict) service curve β_i for each class of traffic i, so Theorem 1 can be applied for each class.

III. Network Calculus and Round-Robin policies

In RR scheduling, traffic classes are served in rounds: at each round, every class can be served a certain amount of data, relying either on a bit-count (DRR) or on a packet-count

[1] This is rather the definition of a vcn service curve, but it has been shown that under the finite long-term service rate assumption [12, Th. 9.5], this is equivalent to the strict service curve. Here, we will reserve the term *variable capacity node* for the bivariate service guarantee for more clarity.

(RR or WRR). Moreover, there can be internal sub-rounds inside each round, that lead to the *interleaved* round-robin policies, where one packet per class is served at each sub-round, until the per-round service limit is reached. Interleaved policies prevent any class from starving too long. IWRR was already described in the seminal paper [1], and is precisely given in [13].

In this section, we provide some results from the state of the art, first when the characteristics of the traffic (*i.e.*, the arrival curves) are unknown, then when they are known. We finally provide a counter-example to [16].

Consider a server offering a strict service curve β.

A. Traffic-agnostic service curves for Round-robin policies

1) General Processor-Sharing (GPS) [8]: The idealized bandwidth-sharing policy is GPS. Each class i is offered a proportion $\phi_i \geq 0$ of the service, with $\sum_{i=1}^n \phi_i = 1$. A strict service curve for class i is then

$$\beta_i^{GPS} = \phi_i \beta.$$

We can observe that $\sum_{i=1}^n \beta_i^{GPS} = \beta$: there is no cost in dividing the bandwidth among the classes.

2) Deficit Round Robin (DRR) [11]: Each class i is associated a quantum q_i representing the amount of data served during each round (this is an approximation due to the packetization). Denoting $Q = \sum_{i=1}^n q_i$, there exists a constant H_i depending on $(q_j, \ell_j^{\max})_{j\in\mathbb{N}_n}$ such that a strict service curve offered to class i is

$$\beta_i^{DRR} = \frac{q_i}{Q}\Big(\beta - H_i\Big)_+.$$

One can remark that $\beta \geq \sum_{i=1}^n \beta_i^{DRR} \geq \big(\beta - \frac{\sum_{i=1}^n q_i H_i}{Q}\big)_+$. There is a cost to the sharing of the bandwidth, represented by $\frac{\sum_{i=1}^n q_i H_i}{Q}$, but on the long run, the sum of the per-class service rates is the service rate of the server. There is no apparent loss of bandwidth in this representation. We call this type of policy *perfect bandwidth-sharing*.

3) Weighted Round Robin (WRR): Each class i is offered to serve w_i packets at each round. There exists a constant H'_i depending on $(w_j, \ell_i^{\max}, \ell_j^{\min})_{j\in\mathbb{N}_n}$ such that

$$\beta_i^{WRR} = \frac{w_i \ell_i^{\min}}{\sum_{j\neq i} w_j \ell_j^{\max} + w_i \ell_i^{\min}}\Big(\beta - H'_i\Big)_+. \tag{1}$$

In this third case, we observe that $\sum_{i=1}^n \beta_i^{WRR} \leq \big(\sum_{i=1}^n \frac{w_i \ell_i^{\min}}{\sum_{j\neq i} w_j \ell_j^{\max} + w_i \ell_i^{\min}}\big)\beta$. For variable-length packets, there exists $\rho < 1$ such that $\sum_i \beta_i^{WRR} \leq \rho\beta$ and the residual service curves cannot take account for the full usage of the bandwidth. We call this type of policy *imperfect bandwidth sharing*. When queues are saturated, the bandwidth is fully used, yet its exact partition between classes is imperfectly known, contrary to perfect bandwidth-sharing.

In [13], the authors derive several IWRR strict service curves for fixed-length packets, and a constant rate server ($\beta = \beta_{R,0}$): a tight service curve that alternates between idle periods (slope 0) and full service (slope R), and a piece-wise linear service curve. In particular, one service curve can be similar to that of Eq. (1), replacing H'_i by $H''_i < H'_i$.

The service curves given above can be computed without the knowledge of the arrival curves of the classes, so they are called *traffic-agnostic*. Service curves can be improved with this knowledge.

B. Cross-traffic aware service curves for RR policies

Bandwidth-sharing policy (Def. 1) has been defined in [10] and includes GPS and DRR. To differentiate from the imperfect bandwidth-sharing defined in Section IV, we call it *perfect bandwidth-sharing*.

1) GPS and DRR:

Definition 1. *A server has a* perfect bandwidth-sharing *policy if there exist positive numbers $(\phi_i)_{1\leq i\leq n}$ and non-negative numbers $(H_{i,j})_{1\leq i,j\leq n}$ such that for all $i \neq j$, for all backlogged period $(s,t]$ of class i,*

$$\phi_j D_i(s,t) \geq \phi_i (D_j(s,t) - H_{i,j})_+. \tag{2}$$

This definition recovers GPS (when $H_{i,j} = 0$ for all i, j) and DRR. In the framework of network calculus, the first work about GPS in [8] solves the case for a constant-rate server. This result is generalized to convex service curves and concave arrival curves in [9], and an alternative proof is given in [10], that is also valid for all perfect bandwidth sharing policies.

Theorem 2. *There exist non-negative constants $(H_{i,M})_{i,M\subseteq\mathbb{N}_n}$ depending on $(\phi_j, H_{i,j})_{i,j}$ only such that a strict service curve for class i is*

$$\beta_i = \sup_{M\subseteq\mathbb{N}_n\setminus\{i\}} \frac{\phi_i}{\Phi_{\overline{M}}}\Big(\beta - \sum_{j\in M}\alpha_j - H_{i,M}\Big)_+, \tag{3}$$

with $H_{i,M} = 0$ for the GPS policy.

Another method has been proposed in [15]. First the authors derive a more precise traffic-agnostic service curve as $\beta_i^{DRR} = \psi_i(\beta)$, where ψ_i is a non-decreasing function. They propose two solutions: the tighter model where ψ_i is a function modeling precisely the alternation of idle (slope 0) and service (slope R) periods for each class, and the linear model, where ψ_i is a piece-wise linear convex function. This is due to a more precise modeling of a bandwidth-sharing policy, where Eq. (2) is replaced by $D_i(s,t) \geq \psi_{i,j}(D_j(s,t))$. To obtain traffic-aware service curves, they combine this with the result stating that if β_k are per-class service curves for classes $k \in M$, then $(\beta - \sum_{k\in M}\alpha_k \oslash \beta_k)_+$ is a strict service curve for the aggregations of classes in $\overline{M}$. This result can be combined with the traffic-agnostic result, and the per-class service curves can be computed iteratively with the formula for class i:

$$\beta_i = \sup_{M\subseteq\mathbb{N}_n\setminus\{i\}} \psi_{i,\overline{M}}\Big(\beta - \sum_{j\in M}\alpha_j \oslash \beta_j\Big)_+, \tag{4}$$

for some functions $\psi_{j,\overline{M}}$ depending on the $\psi_{i,j}$'s only.

Experimentally, these latter bounds are better than with Th. 2, even in the linear model. However, no proof has been given. We show on a simplified example that these two methods may not be comparable.

Example 1. *Consider two classes, with $\beta : t \mapsto R(t-T)_+$, and $\alpha_1(t) = b_1 + r_1 t$. Assume that for $i \in \{1,2\}$, $\phi_{3-i} D_i(s,t) \geq \phi_i(D_{3-i}(s,t) - H_i)_+$, and that $\phi_1 R > r_1$, and $\phi_1 + \phi_2 = 1$. A first strict service curve for class i is $\beta_i^1 = \phi_i(\beta - H_i)_+$. Another strict service curve for class 2 can be derived as*

- *[15]: $\beta_2^2 = (\beta - \alpha_1 \oslash \beta_1^1)_+ = (R-r_1)(t - T + \frac{b_1 + r_1(2T + \frac{H_1}{R})}{R - r_1})_+$;*
- *[10]: $\beta_2^3 = (\beta - \alpha_1 - \phi_1 H_1)_+ = (R - r_1)(t - T - \frac{b_1 + r_1 T + \phi_1 H_1}{R - r_1})_+$.*

Let us compare β_2^2 and β_2^3.

- *If $H_1 = 0$, as in GPS, then the price of the latency of β is paid twice in [15], and [10] is better than [15];*
- *If $T = 0$ then [15] is better than [10] if $\phi_1 > \frac{r_1}{R}$, which is often the case.*

2) WRR is not a perfect bandwidth-sharing policy: In [16], the authors claim that WRR is a bandwidth-policy, and that Theorem 2 can be applied with $\phi_i = w_i \ell_i^{\min}$ and $\phi_j = w_j \ell_j^{\max}$ for all $j \neq i$. They moreover claim a simplification of the result compared to [10] : $H_{j,M} = \sum_{i \in \overline{M} \setminus \{j\}} H_{j,i}$.

In the case of a two-class server,

$$\beta_2 = \max\left(\frac{\phi_2}{\phi_2 + \phi_1}(\beta - H_{2,1})_+, (\beta - \alpha_1)_+\right)$$

is a strict service curve for class 2, and in particular, $(\beta - \alpha_1)_+$ is a strict service curve.

Let us first give an example that contradicts this result.

Example 2. *Assume that for $i \in \{1,2\}$, $w_i = 1$, $\ell_i^{\min} = 1$, $\ell_i^{\max} = 3$, that $\alpha_1 = \gamma_{3,1/2}$ and that the servers serves exactly one packet per time unit (it guarantees the strict service curve $\beta = \beta_{1,0}$). The arrival and departure processes are represented in Table I: packets of class 1 are numbered a_i and packets of class 2 b_i. The intuition is the following: from time 0 to time 6, five long packets arrive for class 2 and five short packets for class 1. It takes up to time 23 to serve these packets. In the meantime, five long packets of class 1 arrive, conform to the arrival curve α_1, creating a backlog that will be served from time 23. This backlog will reduce the service, hence increase the delay for class 2, which is not taken into account in $(\beta - \alpha_1)_+$. Class 2 is not backlogged from time 23 to 24, and five small packets arrive at time 24, so time 24 is a start of a backlogged period for class 2. These packets are served alternately with the five long packets of class 1. Packet b_{11} is served at time 43. One the one hand, $D_2(43) - D_2(24) = 5$. On the other hand, $(\beta - \alpha_1)_+(t) = (t - 3 - \frac{1}{2}t)_+ = \frac{1}{2}(t-6)_+$ and $(\beta - \alpha_1)_+(19) = 6.5 > 5$. So $(\beta - \alpha_1)_+$ is not a strict service curve for class 2.*

Beyond disproving the result from [16], this example also shows that WRR is not a bandwidth-sharing policy as defined in Def. 1. Indeed, from Th. 2, there would exists $H \geq 0$, independent on α_1 and α_2 such that $(\beta - \alpha_1 - H)_+$ is a strict service curve for class 2. To prove this does not hold, the example can also be modified by increasing the number of packets arriving in the both phases, creating a larger backlog at the start of second phase and reduce the service for class 2.

Computing a correct arrival curve, requires either knowing α_2 (to compute a backlog upper bound), or using both $\ell_i^{\min}$ and $\ell_i^{\max}$ for each class.

IV. Imperfect bandwidth-sharing policy

In this section, we define the imperfect bandwidth-sharing policy, in order to give a general method to compute a strict service curves for each class of traffic for scheduling policies such as WRR or IWRR. Compared to [10], we will generalize the definition, inspired by the work of [15], to allow a more precise modeling of these policies. Nevertheless, for the sake of conciseness, we focus on linear modelings only.

A. Definition

Definition 2. *The server has an imperfect bandwidth-sharing policy if there exist non-negative and non-decreasing functions $(\xi_{i,j})_{1 \leq i,j \leq n}$, such that for all class i, for all backlogged period $(s,t]$ of class i, for all class j,*

$$\xi_{i,j}(D_i(s,t)) \geq D_j(s,t).$$

We can assume that $\xi_{i,i}$ is the identity function. One could enforce that $\xi_{i,j}(0) = 0$, but this is not necessary.

One can first remark that this is a generalization of the perfect bandwidth-sharing policies: in that case we have $\xi_{i,j} = \gamma_{H_{i,j}, \frac{\phi_j}{\phi_i}}$. Therefore, perfect bandwidth-sharing could also be defined with functions $\xi_{i,j}$ replacing parameters $\phi_{i,j}$ and $H_{i,j}$. If $r_{i,j}$ is the long-term growth rate of $\xi_{i,j}$, perfect bandwidth-sharing policies impose that $r_{i,j} = 1/r_{j,i}$, which is not the case for imperfect bandwidth-sharing policies.

B. Examples of imperfect bandwidth-sharing policies

Let us now give some examples of imperfect bandwidth-sharing policies.

1) WRR is an imperfect bandwidth-sharing policy: With the notations defined in Section III, the result can be directly adapted from [16],

$$\frac{w_j \ell_j^{\max}}{w_j \ell_i^{\min}} D_i(s,t) + w_j \ell_j^{\max} \geq D_j(s,t).$$

From the remark above, a perfect bandwidth-sharing would impose that $\ell_i^{\max} = \ell_i^{\min}$ for all class i.

2) IWRR is an imperfect bandwidth-sharing policy: An adaptation from [16] would be straightforward, but function $\xi_{i,j}$ can be improved.

Lemma 1. *Consider a server with IWRR policy. Then for all $i, j \in \mathbb{N}_n$, for all backlogged period $(s,t]$ of class i,*

$$\frac{w_j \ell_j^{\max}}{w_j \ell_i^{\min}} D_i(s,t) + h_{i,j} \ell_j^{\max} \geq D_j(s,t),$$

with $h_{i,j} = w_j - w_i$ if $w_j > w_i$ and $h_{i,j} = w_j\left(1 - \frac{w_j - 1}{w_i}\right)$ if $w_i \geq w_j$.

Proof. Consider $(s,t]$ a backlogged period for class i, and assume that p packets are completely served. Let us compute

Arrival process:

packet	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}	b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8	b_9	b_{10}	b_{11}
length	1	1	1	1	1	3	3	3	3	3	3	3	3	3	3	3	1	1	1	1	1
arr. time	0	0	0	3	6	10	16	22	28	34	0	0	0	0	0	0	24	24	24	24	24

Departure process:

packet	b_1	a_1	b_2	a_2	b_3	a_3	b_4	a_4	b_5	a_5	b_6	a_6	b_7	a_7	b_8	a_8	b_9	a_9	b_{10}	a_{10}	b_{11}
length	3	1	3	1	3	1	3	1	3	1	3	3	1	3	1	3	1	3	1	3	1
dep. time	3	4	7	8	11	12	15	16	19	20	23	26	27	30	31	34	35	38	39	42	43

TABLE I: Arrival and departure processes for the counter-example to [16].

an upper bound K of the number of packets (partially) served for class j during this time interval. From [16], $K \geq \lfloor \frac{p}{w_i} \rfloor w_j + (w_j - w_i)_+ + \min((p \mod w_i) + 1, w_j)$.

The first term corresponds to consecutive rounds of service of w_i packets of class i, during which w_j packets of class j are served. The two other terms account for the remaining packets, either at the start of a round (packets are served in turn) or at the end (packets of class j might be served in a row if $w_i < w_j$). If $w_i < w_j$, then $K \leq \frac{w_j}{w_i} p + w_j - w_i + 1$. If $w_i \geq w_j$, note that for all $x < w_i$, $\min(x \mod w_i + 1, w_j) \leq \frac{w_j}{w_i} x + w_j - \frac{w_j}{w_i}(w_j - 1)$. Then $K \leq \frac{w_j}{w_i} p + w_j - \frac{w_j}{w_i}(w_j - 1)$.

We have $D_i(s,t) \geq p\ell_i^{\min}$ and $D_j(s,t) \leq K\ell_j^{\max} \leq \left(\frac{w_j}{w_i} p + w_j - \frac{w_j}{w_i}(w_j - 1)\right)\ell_j^{\max}$, hence the result after rewritings. ◻

Note that our formula differs from that of [16] because there, $\min((p \mod w_i) + 1, w_j)$ is bounded by p.

3) Packets distribution: We can also take advantage of more knowledge about the packet-length distribution in each class. For example, packets could be split into chunks of similar size, except for the last chunk that could be smaller. If ℓ is the length of the chunk, and $h_i = \left\lceil \frac{\ell_i^{\min}}{\ell} \right\rceil$, we know that there is at most one chunk shorter than ℓ every h_i chunks, and obtain the following inequality, proved in Appendix A.

$$\frac{h_i w_j}{(h_i - 1) w_i} D_i(s,t) + w_j \ell + \frac{w_j}{w_i} \ell \geq D_j(s,t). \tag{5}$$

C. Vcn guarantees and strict service curve for each class

We now derive vcn guarantees for each class. We proceed in several steps. First, Lemma 2 is used to compute a per-class vcn guarantee from a vcn guarantee for the aggregation of a subset of classes. Second, from per-class vcn guarantees of a subset of classes, one can compute a vcn guarantee of an aggregation of classes (Theorem 3).

1) A cross-traffic agnostic vcn for each class: Suppose that we are able to compute a vcn guarantee C_M^{ag} for the aggregate classes in $M \subseteq \mathbb{N}_n$ of classes. Lemma 2 shows how to derive a vcn guarantee for each class $i \in M$.

Lemma 2. *If C_M^{ag} is a vcn guarantee for the aggregate classes in M, then for all $i \in M$,*

$$C_i = \psi_{i,M} \circ C_M^{ag} \tag{6}$$

is a vcn guarantee for class i with $\psi_{i,M} = (\sum_{j \in M} \xi_{i,j})^{-1}$, and $f^{-1}(x) = \inf\{z \geq 0 \mid f(z) \geq x\}$.

Proof. For all $s, t \in \mathbb{R}_+$ such that $(s,t]$ is a backlogged for class i, for all $j \in M$, $\xi_{i,j}(D_i(s,t)) \geq D_j(s,t)$. Therefore, $\sum_{j \in M} \xi_{i,j}(D_i(s,t)) \geq \sum_{j \in M} D_j(s,t)$. Since $(s,t]$ is also a backlogged period for the aggregate classes in M,

$$\sum_{j \in M} D_j(s,t) \geq C_M^{ag}(s,t),$$

and then $\left(\sum_{j \in M} \xi_{i,j}\right)(D_i(s,t)) \geq C_M^{ag}(s,t)$. Denoting $\psi_{i,M} = (\sum_{j \in M} \xi_{i,j})^{-1}$, we obtain $D_i(s,t) \geq \psi_{i,M}(C_M^{ag}(s,t))$, which concludes the proof. ◻

When $M = \mathbb{N}_n$, per-class traffic-agnostic guarantees are computed.

2) A vcn guarantees for aggregated classes: The next step is to compute a vcn guarantee for classes $M \subseteq \mathbb{N}_n$, so that Lemma 2 can be applied to any subset of classes.

Theorem 3. *Assume that C is a vcn guarantee for the server, with strict service curve β, and that there exists $M \subseteq \mathbb{N}_n$ such that each class $j \in M$ has a vcn guarantee $C_j \geq \psi_j \circ C$, for a non-decreasing function ψ_j such that $\psi_j(0) = 0$. Then $C_{\overline{M}}^{ag}$ is a vcn guarantee for the aggregated classes in $\overline{M}$, where, for all $s \leq t$,*

$$C_{\overline{M}}^{ag}(s,t) \geq C(s,t) - \sum_{j \in M} \sup\left\{\alpha_j(t - s + v_j) - \psi_j(Y_j) \mid \forall j \in M,\ v_j \geq 0, Y_j \geq \max(\beta(v_j), \beta(t - s + v_j) - C(s,t))\right\}. \tag{7}$$

Proof. Let s and t be such that $(s,t]$ is a backlogged period for classes $\overline{M}$. Then, $D(s,t) \geq C(s,t)$.

For all $j \in M$, let $p_j = \sup\{v \leq s \mid A_j(v) = D_j(v)\}$ be the last start of backlogged period of class j before time s. On the one hand, from the hypothesis,

$$D_j(p_j, s) \geq C_j(p_j, s) \geq \psi_j(C(p_j, s)). \tag{8}$$

On the other hand, for all $j \in M$, $D_j(p_j,t) = D_j(t) - D_j(p_j) = D_j(t) - A_j(p_j) \leq A_j(t) - A_j(p_j) = A_j(p_j,t) \leq \alpha_j(t - p_j)$.

Recall that $D^{\overline{M}} = \sum_{j \neq M} D_j$. Combining the two previous inequalities, we obtain

$$\begin{aligned} D^{\overline{M}}(s,t) &\geq C(s,t) - \sum_{j \in M} D_j(s,t) \\ &= C(s,t) - \sum_{j \in M} \left(D_j(p_j,t) - D_j(p_j,s)\right) \\ &\geq C(s,t) - \sum_{j \in M} \left(\alpha_j(t - p_j) - \psi_j(C(p_j,s))\right) \quad (9) \\ &\geq C(s,t) - \sum_{j \in M} \sup_{p_j \leq s} \left(\alpha_j(t - p_j) - \psi_j(C(p_j,s))\right). \quad (10) \end{aligned}$$

Since $(p_j, s]$ is a backlogged period for class j and $(s,t]$ a backlogged period for classes $\overline{M}$, then $(p_j, t]$ is a backlogged period for the system, and the inequalities $C(x,y) \geq \beta(x-y)$ must hold for $(x,y) \in \{(p_j,s),(p_j,t)\}$, so $C(p_j,s) \geq \beta(s-p_j)$ and, since C is super-additive, $C(p_j,t) \geq C(p_j,s) + C(s,t) \geq \beta(t-p_j)$. We obtain the formulation of the statement in Eq. (7) with $v_j = s - p_j$, and $Y_j = C(p_j, s)$. □

In order to solve the optimization given in Eq. (7) easily, we specify this theorem to the linear model, where $\beta = \beta_{R,T}$ is a rate-latency function and $\alpha_j = \gamma_{b_j,r_j}$ are token-bucket functions.

Corollary 1. *With the same hypotheses and notations as above, if moreover $\beta = \beta_{R,T}$, and $\alpha_j = \gamma_{b_j,r_j}$, then $C^{ag}_{\overline{M}}$ is a vcn guarantee for the aggregated classes in $\overline{M}$, where, for all $s \leq t$,*

$$C^{ag}_{\overline{M}}(s,t) \geq \Big((1 - \frac{r^M}{R})C(s,t) - (b^M + r^M T + q^M)\Big)_+.$$

where $q_j = \sup_{t\geq 0} r_j t - \psi_j(Rt)$ and $x^M = \sum_{j\in M} x_j$.

Proof. Since $C(s,t) \geq R(t-s-T)_+$, there exists $T_1 \leq T$ such that $C(s,t) = R(t-s-T_1)$. Then, we have

$$Y_j \geq (R(t-s+v_j-T)_+ - R(t-s-T_1))_+ \geq R(v_j - T + T_1)_+.$$

For all classes $j \in M$, let us bound the supremum of Equation (7) using this bound on Y_j:

$$\begin{aligned}
&\sup\{\alpha_j(t-s+v_j) - \psi_j(Y_j) \mid \\
&\qquad v_j \geq 0, Y_j \geq \max(\beta(v_j), \beta(t-s+v_j) - C(s,t))\} \\
&\leq \sup\{b_j + r_j(t-s+v_j) - \psi_j(R(v_j - T + T_1)_+) \mid v_j \geq 0\} \\
&= b_j + r_j(t-s+T-T_1) + \sup_{v_j\geq 0}\{r_j(v_j) - \psi_j(Rv_j)\} \\
&= b_j + r_j(t-s+T-T_1) + q_j.
\end{aligned}$$

When summing those terms, we obtain

$$C^{ag}_{\overline{M}}(s,t) \geq C(s,t) - (b^M + r^M(t-s+T-T_1) + q^M).$$

Now, one can express T_1 as $T_1 = t - s - \frac{C(s,t)}{R}$, and

$$\begin{aligned}
C^{ag}_{\overline{M}}(s,t) &\geq C(s,t) - (b^M + r^M(T + \frac{C(s,t)}{R}) + q^M) \\
&= (1 - \frac{r^M}{R})C(s,t) - (b^M + r^M T + q^M).
\end{aligned}$$

We conclude by remarking that $C^{ag}_{M} \geq 0$. □

A strict service curve can be deduced by bounding the variable capacity node of the server by its strict service curve: $C(s,t) \geq \beta(t-s)$ for all $s \leq t$.

Corollary 2. *With the same assumptions as Cor. 1, a strict service curve for classes $\overline{M}$ is $\beta^{ag}_{\overline{M}} = (\beta - \sum_{j\in M}(\alpha_j + q_j))_+$.*

Example 3. *Consider again Example 1. We find $q_1 = \frac{r_1 H_1}{R}$, so that Corollary 2 gives*

$$\beta_2^4 = (R - r_1)\Big(t - T + \frac{b_1 + r_1(T + \frac{H_1}{R})}{R - r_1}\Big)_+,$$

and [15] is improved by $\frac{r_1}{R-r_1}T$ compared to β_2^2 and while [10] is improved by $(\phi_1 - \frac{r_1}{R})H_1$ compared to β_2^3.

Note that Corollaries 1 and 2 do not require that the ψ_j's are linear, so it can also be applied to the tight model of [15].

D. Dealing with the stability issue

The results of the previous section may not be enough to compute finite performance bounds, as illustrated next.

Example 4. *Consider the RR server with 2 classes, with $\ell^{\min} = 1$ and $\ell^{\max} = 2$. Then, $\xi_{1,2} = \xi_{2,1} = \gamma_{2,2}$: $D_1(s,t) \geq 2(D_2(s,t) - 2)_+$. Assume that $\beta = \beta_{1,0}$ and that $\alpha_1 = \alpha_2 = \gamma_{b,r}$. We have $\psi_1 = (\xi_{1,2} + Id)^{-1} = \beta_{1/3,2}$. The service rate guaranteed to class 1 is 1/3 (and similarly for class 2 by symmetry). Finite bounds for q_1 can be found only if $r \leq 1/3$, and finite performance bounds can be finite, using this theorem only if the load of the system is less than 0.667.*

We now solve this issue in the linear model. The idea is to use backlog upper bounds for each subset of classes. Assume the server is stable: $\sum_{i\in\mathbb{N}_n} r_i < R$. We can compute an upper bound of the backlog in the server: $B = \sup_{t\geq 0} \sum_i \alpha_i(t) - \beta(t)$. This is also an upper bound of the backlog for any subset of class.

More precisely, we can improve Corollary 1 if we know a backlog bound B_M for the aggregation of classes in M.

Theorem 4. *With the same hypotheses and notations as in Cor. 1, and if B_M is an upper bound for the backlog of classes in M at any time, then $C^{\overline{M}}_B$ is a vcn guarantee for classes $\overline{M}$ where $C^{ag}_{\overline{M}}$ is defined for all $0 \leq s \leq t$ as*

$$C^{ag}_{\overline{M}}(s,t) = \big(C(s,t) - r^M_j(t-s) - B_M\big)_+.$$

Proof. We use a result from [12, Theorem 5.4], stating that if 1) $(\gamma_{b_j,r_j})_{j\in M}$ are respective arrival curves for classes $j \in M$, and this is their only constraints; 2) the service curve β is continuous; 3) the server is causal; 4) B_M is a backlog upper bound for the aggregate classes $j \in M$, then γ_{B_M, r^M} is an arrival curve for the departure process. The assumptions are satisfied by our server, so for all $s \leq t$, $D^M(s,t) \leq B_M + r^M(t-s)$.

For all $s \leq t$ in the same backlogged period for classes $\overline{M}$, $D^{\overline{M}}(s,t) \geq 0$ and

$$\begin{aligned}
D^{\overline{M}}(s,t) &\geq C(s,t) - D^M(s,t) \\
&\geq C(s,t) - B_M - r^M(t-s).
\end{aligned}$$

□

One can now regroup Theorem 4 and Corollary 1.

Corollary 3. *With the same notations as in Theorem 4, a vcn guarantee for aggregate classes $\overline{M}$ is*

$$C^{ag}_{\overline{M}}(s,t) \geq \Big((1 - \frac{r^M}{R})C(s,t) - \min(b^M + q^M, B_M) - r^M T\Big)_+.$$

Proof. We start by deriving another variable capacity node from Theorem 4. First notice that

$$C(s,t) \geq R(t-s-T)_+ \geq R(t-s-T),$$

so $r^M(t-s) \leq \frac{r^M}{R}C(s,t)+r^M T$. As a consequence, $C^{ag}_{\overline{M}}(s,t) \geq (C(s,t)-\frac{r^M}{R}C(s,t)-r^M T-B_M)_+ = ((1-\frac{r^M}{R})C(s,t)-r^M T-B_M)_+$. We conclude by taking the maximum with the vcn guarantee from Cor. 1. □

Example 5. *Continuing from Example 4, a backlog upper bound for the server is* $2b$*, so class 2 is guaranteed a variable capacity node* $C_1 \geq ((1-r)C - 2b)_+$ *from Corollary 3, and then combining with Lemma 2, we obtain* $C_1(s,t) \geq \max((1-r)C-2b)_+, \frac{1}{3}(C(s,t)-2)_+)$*, which ensures finite performance bounds for any server load under 1.*

V. Algorithms for Imperfect Bandwidth Sharing

We now focus on the algorithmic aspect of computing service curves for imperfect bandwidth-sharing scheduling. We first give an iterative algorithm that has an exponential-time complexity. and then give a simpler heuristic.

A. Iterative algorithm

Let us denote $\chi_{\overline{M}} : t \mapsto \big((1-\frac{r^M}{R})x - \min(b^M+q^M, B_M) - r^M T\big)_+$. If we combine Lemma 2 and Corollary 3 to subset M and class $i \notin M$, one gets that $\psi_{i,\overline{M}} \circ \chi_{\overline{M}} \circ C$ is a vcn guarantee for class i, and that $\beta_i = \psi_{i,\overline{M}} \circ \chi_{\overline{M}} \circ \beta$ is a strict service curve for class i. We denote $\beta^{ag}_{\overline{M}} = \chi_{\overline{M}} \circ \beta$.

We have the following dependencies:

- β_i depends on β^{ag}_M;
- β^{ag}_M depends on $(\beta_j)_{j\in\overline{M}}$ and B_M;
- B_M depends on β^{ag}_M.

Algorithm 1 is an iterative scheme to solve these inter-dependencies and compute per-class strict service curves.

First, one can define constants that depend only on the server and traffic parameters: the arrival curves $(\alpha_j)_{j\in\mathbb{N}_n}$, the service curve β, and $(\xi_{i,j})_{i,j\in\mathbb{N}_n}$ as defined in Def. 2. We can also pre-compute and store the following constants:

- $\psi_{i,M} = (\sum_{j\in M}\xi_{i,j})^{-1}$ (see Lemma 2);
- $\alpha^M = \sum_{j\in M}\alpha_j$, the aggregate arrival curves.

Second, the variables used in the algorithm, and basic functions to compute them are the following:

- Aggregate backlog for the subset of classes M, $M \subseteq \mathbb{N}_n$: $B_M = \text{AggBack}(M, \beta^{ag}_M) = vDev(\alpha^M, \beta^{ag}_M)$ (Th. 1);
- Per-class service curve for class $i \in \mathbb{N}_n$: $\beta_i = \text{ClsSC}(i, M, \beta^{ag}_M)$ (Lemma 2);
- Aggregate service curve for subset $M \subseteq \mathbb{N}_n$: $\beta^{ag}_M = \text{AggSC}(M, (\beta_j)_{j\notin M}, B_{\overline{M}})$ (Corollary 3).

Lines (1-2) initialize the variables, to the maximum backlog for the server for the aggregate backlog B_M, null functions for aggregate service curves and a first application of Lemma 2 to $M = \mathbb{N}_n$ for the per-class service curves. Then we iterate the loop (lines 3-8) until a stopping criterion is met. It can be arbitrarily defined, since the service curve computed are always valid. The inner loop improves the service curves for each class and subset of classes (we use that the maximum of strict service curve is also a strict service curves).

The complexity of each round of Algorithm 1 is exponential in the number of classes. Indeed, it as an internal loop on all the subsets of classes. A first way to improve is to remove useless computations. If, for example $\beta^{ag}_{\overline{M}}$ is not improved at line 5, lines (6-8) become useless, and so on. This can be done by introducing Boolean variables to state if a variable has been improved, and can then improve other variables in the subsequent steps.

Algorithm 1: Iterative scheme for computing per-class strict service curves

Data: $(\alpha_j)_{j\in\mathbb{N}_n}$, β, $(\xi_{i,j})_{i,j}$, Stop a stopping criterion.
Result: A per-class strict service curve for all class j

1 **foreach** $M \subseteq \mathbb{N}_n$ **do** $B_M \leftarrow B$; $\beta^{ag}_M \leftarrow \beta_{0,0}$; ;
2 **foreach** $j \in \mathbb{N}_n$ **do** $\beta_j \leftarrow \text{ClsSC}(j, \mathbb{N}_n, \beta)$;
3 **while** *not* Stop **do**
4   **foreach** $M \subseteq \mathbb{N}_n$ **do**
5         $\beta^{ag}_{\overline{M}} \leftarrow \max(\beta^{ag}_{\overline{M}}, \text{AggSC}(\overline{M}, (\beta_j)_{j\in M}, B_M))$;
6   **foreach** $j \notin M$ **do**
7             $\beta_j \leftarrow \max(\beta_j, \text{ClsSC}(j, M, \beta^{ag}_{\overline{M}}))$;
8         $B_{\overline{M}} \leftarrow \min(B_{\overline{M}}, \text{AggBack}(\overline{M}, \beta^{ag}_{\overline{M}}))$;
9 **return** $(\beta_j)_{j\in\mathbb{N}_n}$.

Another solution is to perform the computations in parallel. Indeed, elementary operations for improving β^{ag}_M, β_i and B_M for each $i \in \mathbb{N}_n$ or $M \subseteq \mathbb{N}_n$ can be done in parallel, using a shared memory, similarly to what is done in [19] or [20]. We do not detail the approach here, but rather present a heuristic to reduce the complexity.

B. A polynomial time heuristic

Tight per-class service curves can be computed in polynomial time for GPS [9]. Intuitively, the classes that use the less their share of service give the benefit of unused bandwidth to the other classes of traffic, but not the reverse. This enables to compute an order on the classes so that (let us assume this is the natural order) only the computation of $\beta^{ag}_{\{k,\dots,n\}} = (\beta - \sum_{j=1}^{k-1}\alpha_j)_+$ is needed to compute tight per-class GPS strict service curves[2]:

$$\beta_i = \sup_{j\leq i}\frac{\phi_i}{\sum_{k=j}^{n}\phi_k}\Big(\beta - \sum_{\ell=1}^{j-1}\alpha_\ell\Big)_+.$$

Algo. 2 presents the modification of Algo. 1 using this intuition and improving the per-round complexity: we only update the variables for n inclusion-increasing subsets. At each step, we add one element to M, and update the variables as in Algo 1. If the complexity of each elementary operation is constant, the complexity of Algo. 2 is quadratic.

VI. Numerical evaluation

In this section, we compare with the state of the art: the performance bounds with cross-traffic-agnostic approaches and the one from [17]. We first deal with constant packet

[2]Better service curves can be found, by computing the maximum for all subset of classes, but the performances bounds (delay or backlog) will not be improved.

Algorithm 2: Heuristic for computing per-class strict service curves

Initialization: Lines (1-2) of Algo. 1;
$M \leftarrow \emptyset$;
while $M \neq \mathbb{N}_n$ **do**
  foreach $j \notin M$ **do** $t_j \leftarrow \sup\{t \mid \alpha_j(t) > \beta_j(t)\}$;
  $i \leftarrow \text{argmin}\{t_j,\ j \in M\}$;
  $M \leftarrow M \cup \{i\}$;
  Update the variables: Lines (5-8) of Algo. 1;
return $(\beta_j)_{j \in \mathbb{N}_n}$.

TABLE II: Comparison of the state of the art and 1-round heuristic

number of classes	average pessimism	1 %-approximation proportion	acceleration factor
4	0.11 %	97.5 %	4.9
5	0.18 %	96.1 %	7.6
6	0.20 %	94.8 %	13.2
7	0.25 %	93.8 %	18.3
8	0.27 %	92.8 %	23.3

lengths (which is a perfect bandwidth-sharing policy, similar to DRR) to assess the quality of the heuristic. We then focus on imperfect bandwidth-sharing policies and take inspiration on the numerical results from [17].

A. Constant packet length

For each class, packets have a fixed length. We consider servers with 4 to 8 classes, and for each case, we generate 10,000 random instances of token-bucket arrival curves (bursts and arrival rates generated uniformly at random in $[0, 100]$ kb and $[0, 1]$ Mb.s^{-1} and service rates with the load generated uniformly at random. The packet length of class 1 is 3040 b, and is 12 kb for the other classes, and $w_1 = 5$, $w_i = 2$ for $i > 1$.

We do not observe improvement of the delays compared to the state or the art. Indeed, we did not introduce any latency in the server. But even introducing a latency did not improve the delays: this might be due to the iteration process. However, we can compare this bound with the heuristic given in Algorithm 2. Table II summarizes the results. The average pessimism of the bound is less than 0.3 %, and more than 92 % of the instances approximate the bound of Algo. 1 with less than 1 % relative error. The pessimism slightly increases with the class number. However, the computation time is reduced by up to 23 times.

B. Variable packet length

Consider the example from [16]: a 4-class server with characteristics summarized in Table III. The server is a constant-rate server with service rate R, that varies from 3 Mb.s^{-1} (server load is 1) to infinity. While in [17], the authors focus of small server loads (less than 0.2), to ensure the stability of the system, we can compute finite bounds for all server loads up to 1. Let us denote by ρ the server load ($\rho = \sum_i r_i/R$). Figures 1a and 1b show the comparison between several methods for the

TABLE III: Characteristics of the server use of Section VI-B

class	1	2	3	4
b_i (b)	30208	19968	24576	27648
r_i (Mb.s^{-1})	0.65	0.85	0.95	0.55
$\ell_i^{\min}$ (b)	4096	3072	4608	3072
$\ell_i^{\max}$ (b)	8704	5632	6656	8192
w_i	4	6	7	10

delay bound of classes 2 and 3 for WRR ad IWRR. The agnostic method (Lemma 2 with $M = \mathbb{N}_n$) cannot compute finite delay bounds for respectively $\rho > 0.37$ and $\rho > 0.57$ with WRR. The same stability region (with smaller delays) could be observed for IWRR. The state-of-the-art method from [17] computes similar bounds for WRR and IWRR, and finite delay bounds are computed for ρ up to 0.85, and we can verify that Algorithm 1 can compute finite bounds for all loads. The new modeling of IWRR improves the bounds compared to WRR. The growth rate of the delay bounds (on their linear parts) is smaller with IWRR. This can be explained by the fact that class 3 can serve more packets in a round, and never waits for the service of more than one packet of each class, which results in a reduction of the latency proportional to the service rate. The heuristic also gives a good approximation of the delay bounds. In particular, for class 3, where the difference with Algo. 1 is negligible.

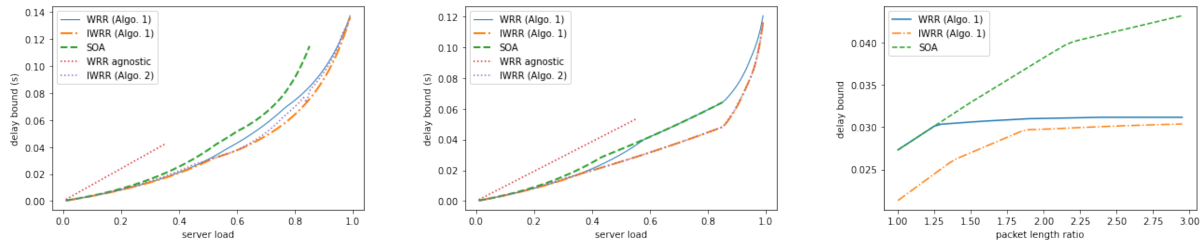
Second, we study the influence of the packet lengths on the delay bounds: every parameter is the same as in Table III, except $\ell_i^{max} = \mu \ell_i^{\min}$, with $\mu \in [1, 3]$, and $\rho = 0.5$. Figure 1c compares the delay bounds of class 2 with IWRR, WRR and state of the art [17]. The delay bound from the state-of-the-art increases strictly with the ratio μ. In contrast, the WRR delay bound first increases, and then remain constant. This happens when the backlog bounds become useful in Algo. 1. This bound is not based on the packet lengths anymore. The same phenomenon can be observed for IWRR, although the delay bounds are smaller, hence the initial part is longer.

VII. Conclusion

In this paper we have presented a method to compute strict service curves for WRR, under the more general framework of imperfect bandwidth-sharing. It corrects and improves the state of the art. Numerical experiments show the gain that can be observed, either algorithmically for constant packet lengths or in the bounds in the general case. Further research will focus 1) towards using the knowledge of the packet-length distribution more generally than Eq. (5) using for example [21]; 2) on balancing the per-round periodic scheme of IWRR, to reduce again the latency of such scheduling. Although the computations presented in this paper are focused on one server, the analysis presented for DRR, in particular [20] and [15] can be applied almost directly (the only difference is to compute the arrival curves at the input instead of the output of the servers).

References

[1] M. Katevenis, S. Sidiropoulos, and C. Courcoubetis, "Weighted round-robin cell multiplexing in a general-purpose atm switch chip," *IEEE*



(a) Delay bounds with different methods for class 2.

(b) Delay bounds with different methods for class 3.

(c) class 2 whith increasing packet length ratio μ.

Fig. 1: Delay bounds with different methods.

Journal on Selected Areas in Communications, vol. 9, no. 8, pp. 1265–1279, 1991.

[2] "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.

[3] 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.

[4] R. Cruz, "Quality of Service Guarantees in Virtual Circuit Switched Networks," *IEEE Journal on selected areas in communication*, vol. 13, pp. 1048–1056, 1995.

[5] A. Mifdaoui and T. Leydier, "Beyond the Accuracy-Complexity Trade-offs of Compositional Analyses 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)*, 2017, pp. 1–8.

[6] L. Thomas, J.-Y. Le Boudec, and A. Mifdaoui, "On Cyclic Dependencies and Regulators in Time-Sensitive Networks," in *IEEE Real-Time Systems Symposium, RTSS 2019*. IEEE, 2019, pp. 299–311. [Online]. Available: https://doi.org/10.1109/RTSS46320.2019.00035

[7] A. Bouillard, "Trade-off between accuracy and tractability of network calculus in FIFO networks," *Perform. Evaluation*, vol. 153, p. 102250, 2022. [Online]. Available: https://doi.org/10.1016/j.peva.2021.102250

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

[9] A. Burchard and J. Liebeherr, "A General Per-Flow Service Curve for GPS," in *30th International Teletraffic Congress, ITC 2018, Vienna, Austria, September 3-7, 2018 - Volume 2*. IEEE, 2018, pp. 31–36. [Online]. Available: https://doi.org/10.1109/ITC30.2018.10058

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

[11] 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*, ICST/IEEE, 2012, pp. 138–147. [Online]. Available: https://doi.org/10.4108/valuetools.2012.250202

[12] A. Bouillard, M. Boyer, and E. Le Corronc, *Deterministic Network Calculus: From Theory to Practical Implementation*. ISTE, 2018.

[13] S. M. Tabatabaee, J. L. Boudec, and M. Boyer, "Interleaved weighted round-robin: A network calculus analysis," *IEICE Trans. Commun.*, vol. 104-B, no. 12, pp. 1479–1493, 2021. [Online]. Available: https://doi.org/10.1587/transcom.2021iti0001

[14] S. M. Tabatabaee and J.-Y. Le Boudec, "Deficit round-robin: A second network calculus analysis," in *2021 IEEE 27th Real-Time and Embedded Technology and Applications Symposium (RTAS)*, 2021, pp. 171–183.

[15] S. M. Tabatabaee and J. L. Boudec, "Deficit round-robin: A second network calculus analysis," *IEEE/ACM Trans. Netw.*, vol. 30, no. 5, pp. 2216–2230, 2022. [Online]. Available: https://doi.org/10.1109/TNET.2022.3164772

[16] V. Constantin, P. Nikolaus, and J. B. Schmitt, "Improving performance bounds for weighted round-robin schedulers under constrained cross-traffic," in *IFIP Networking Conference*. IEEE, 2022, pp. 1–9. [Online]. Available: https://doi.org/10.23919/IFIPNetworking55013.2022.9829772

[17] ——, "Improving performance bounds for weighted round-robin schedulers under constrained cross-traffic," *CoRR*, vol. abs/2202.08381, 2022. [Online]. Available: https://arxiv.org/abs/2202.08381

[18] J.-Y. Le Boudec and P. Thiran, *Network Calculus: A Theory of Deterministic Queuing Systems for the Internet*. Springer-Verlag, 2001, vol. LNCS 2050, revised version 4, May 10, 2004.

[19] S. Plassart and J. L. Boudec, "Equivalent versions of total flow analysis," *CoRR*, vol. abs/2111.01827, 2021. [Online]. Available: https://arxiv.org/abs/2111.01827

[20] S. M. Tabatabaee, A. Bouillard, and J. L. Boudec, "Worst-case delay analysis of time-sensitive networks with deficit round-robin," *CoRR*, vol. abs/2208.11400, 2022. [Online]. Available: https://doi.org/10.48550/arXiv.2208.11400

[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*, ICST/IEEE, 2012, pp. 136–137. [Online]. Available: https://doi.org/10.4108/valuetools.2012.250208

APPENDIX

A. *Proof of formula* (5)

Assume that packets are split into chunks of size ℓ and that $\ell_i^{\min}$ is the minimum packet length of class i. Then one packet is divided in at least $h_i = \lceil \frac{\ell_i^{\min}}{\ell} \rceil$ chunks, the last one potentially having a smaller length. So in m consecutive packets, at most $\lceil \frac{m}{h_i} \rceil \leq \frac{m}{h_i} + \frac{h_i - 1}{h_i}$ have length less than ℓ, and at least $m - \lceil \frac{m}{h_i} \rceil \geq \frac{(h_i-1)(m-1)}{h_i}$ have length exactly ℓ.

Consider $(s,t]$, a backlogged period for class i, and assume that there are exactly p complete rounds of service for class i. This means that at least pw_i packets are served, and consequently

$$D_i(s,t) \geq \frac{(h_i - 1)(pw_i - 1)}{h_i}\ell. \tag{11}$$

There are at most $p+1$ (possibly incomplete) rounds of service for class i during $(s,t]$, and then at most $w_j(p+1)$ packets served: $D_j(s,t) \leq w_j(p+1)\ell$, so we have a lower bound for p: $p \geq \frac{D_j(s,t)}{w_l \ell} - 1$. We can replace p by this bound in Eq. (11), and after basic transformation,

$$\frac{h_i w_j}{(h_i - 1)w_i} D_i(s,t) + w_j \ell + \frac{w_j}{w_i}\ell \geq D_j(s,t).$$