Packetization and Aggregate Scheduling

Anne Bouillard, Nadir Farhi, Bruno Gaujal

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Packetization and Aggregate Scheduling

Anne Bouillard — Nadir Farhi — Bruno Gaujal

N° 7685

Juillet 2011

Distributed and High Performance Computing

Rapport de recherche

ISSN 0249-6399 ISRN INRIA/RR--7685--FR+ENG

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE
INRIA
centre de recherche
PARIS - ROCQUENCOURT

Packetization and Aggregate Scheduling

Anne Bouillard* , Nadir Farhi† , Bruno Gaujal‡

Theme : Distributed and High Performance Computing
Networks, Systems and Services, Distributed Computing
Équipes-Projets TREC et MESCAL

Rapport de recherche n° 7685 — Juillet 2011 — 21 pages

Abstract: We present a new formalism for data packetization in Network Calculus. Packet curves are introduced to model constraints on the packet lengths of data flows. Indeed, a more precise knowledge of the packet characteristics can be efficiently exploited to get tighter performance bounds, specially when dealing with scheduling policies based on packet count, such as round-robin. A second use of packet curves is the packetization of a superposition of periodic flows. Finally, we show that packet curves can be used to compute a global service curve for the aggregate of several flows, with different service curves, sharing a unique queue.

Key-words: Network Calculus, packetization, scheduling, quality of service guarantees

This work has been supported by ANR project PEGASE, 2009-SEGI-009.

* ENS/INRIA TREC `Anne.Bouillard@ens.fr`
† Université Paris-Est, IFSTTAR `nadir.farhi@ifsttar.fr`
‡ INRIA/LIG MESCAL `bruno.gaujal@imag.fr`

Centre de recherche INRIA Paris – Rocquencourt
Domaine de Voluceau, Rocquencourt, BP 105, 78153 Le Chesnay Cedex
Téléphone : +33 1 39 63 55 11 — Télécopie : +33 1 39 63 53 30

Paquetisation et ordonnancement agrégé

Résumé : Nous présentons dans ce rapport un nouveau formalisme de paquétisation de donnée en *Network Calculus*. La notion de courbe de paquets est introduite pour modéliser les contraintes sur les longueurs des paquets de flux de données. En effet, une connaissance plus précise des caractéristiques de paquets peuvent être exploitées efficacement pour obtenir de meilleurs bornes de performances, spécialement dans le cas de politiques de service basées sur le comptage de paquets, comme la politique *round-robin*. Deuxièmement, les courbes de paquets permettent de calculer des caractéristiques (courbes d'arrivées) plus précises de flux périodiques superposés. Enfin, nous montrons comment ces courbes peuvent être utilisées pour calculer une courbe de service globale pour des flux partageant différemment une même ressource.

Mots-clés : Network calculus, paquétisation, ordonnancement, garanties de service.

Introduction

The purpose of this article is to present a new data packetization approach in network calculus. Network calculus [1, 2, 3] is a theory based on min-plus algebra [4] and developed for the calculus of performance bounds in computer and communication networks. Remarkably, this theory is almost only based on two objects: *arrival curves* and *service curves*, that are used to express constraints on arrival flows and service capacities. Performance bounds are then derived by cleverly handling arrival and service curves, and by taking into account the service policies. Although several alternative approaches of the network calculus exist, such as trajectory methods [5] or model checking [6], the network calculus approach is applied on a range of fields, e.g. internet Quality of Service (QoS) [7], wireless sensor networks [8], with several advantages on other approaches.

Network calculus has recently received a lot of attention because its algebraic framework provides an efficient and elegant way to compose elementary network elements into more complex systems in order to get worst-case performances upper bounds. Unfortunately, those bounds are often over-pessimistic. Indeed, as soon as several flows and servers are composed together, tight bounds cannot be obtained from purely algebraic methods. This phenomenon has been observed under several assumptions (blind multiplexing [9] or FIFO [10]). Some exact methods have been derived using linear programming [11] in general acyclic networks, but are algorithmically costly and there are no general results for networks with cyclic dependencies.

While network composition and flow aggregation has received a lot of attention, few works in network calculus concern the packet nature of flows. The main technique to deal with packets so far is called packetization and only uses the maximal and minimal sizes of the packets. This is rather unsatisfactory because most actual flows in communicating embedded systems are made of packets (often of different sizes) and the interaction between the flows inside a node of the system is also often packet-based (for example when no preemption is possible).

In this paper, we propose a more refined modeling of packet flows, that may have different packet lengths. We propose a new object that we call *packet curve*, that captures information about the distribution of packets in a flow the same way as arrival time constraints of data are captured by arrival curves, in network calculus theory. As mentioned before, when packets may have different lengths, only the minimum and the maximum lengths have been taken into account in the calculation of performance bounds. We show here that the whole available information on the packet lengths, given by a packet curve, can be taken into account in that calculation.

In Section 3 we provide closed form formulas for the packets curves in one important case, the superposition of several periodic flows.

In Section 4, we apply the approach using packet curves to calculate residual services of arrival flows routed under the round-robin policy, where packets of each flow may have different lengths, and where information on the sequence of packet lengths of a given flow is given in packet curves. Although this approach is quite efficient for the round-robin service discipline, we will also see that the approach is not as good to other service policies such as packet-based fixed priority or packet-based FIFO.

In Section 5, we treat the problem of determining a global minimum strict service curve for the aggregation of flows that guarantee some given minimum services for each flow. This problem only has some meaning when data is set in packets, in which case, the server is supposed to be reinitiated each time it starts to serve a new packet flow. Under the general case where packets of one flow may have different lengths, we show the role of packet curves on the calculus of global minimum service curves.

1 Network Calculus Preliminaries

Network calculus is based on (min,plus) algebra [4]. Data arrivals and services are modeled by (min,plus) functions and (min,plus) operators such as (min,plus) convolution and deconvolution, and used to express and handle constraints on data arrivals and service. More precisely, the set of functions considered is $\mathcal{F} \stackrel{\text{def}}{=} \{f : \mathbb{R}_+ \to \mathbb{R}_+ \cup \{+\infty\} \mid f(0) = 0 \text{ and } f \text{ is non-decreasing}\}$, where $\mathbb{R}_+$ is the set of non-negative reals, and the two operators are defined as follows: let $f, g \in \mathcal{F}$, then, $\forall t \in \mathbb{R}_+$,

- (min,plus)-convolution: $f * g(t) = \inf_{0 \leq s \leq t} f(s) + g(t-s)$;
- (min,plus)-deconvolution: $f \oslash g(t) = \sup_{s \geq 0} f(t+s) - g(s)$.

The set $\mathcal{F}$ equipped with the minimum and the (min,plus)-convolution is a semi-ring with zero element $\epsilon : t \mapsto +\infty$ and unit element $e : 0 \mapsto 0; t \mapsto +\infty$. We also define the power of a function as $f^0 = e$ and $\forall n \in \mathbb{N} \setminus \{0\}$, $f^n = f * f^{n-1}$.

Two important notions in network calculus theory are *arrival curves* and *service curves*. One of the main objectives of this theory is to calculate upper bounds of end-to-end delays and data backlogs on servers. This section provides a brief review of the basic results of network calculus. A more detailed presentation can be found in [1, 2].

Consider a data flow arriving at a server. For $t \in \mathbb{R}_+$, the cumulative amount of data between times 0 and t is denoted by $A(t) \in \mathcal{F}$. The function A is then non-decreasing, and $A(0) = 0$.

Definition 1 ((Maximum) arrival curve)**.** A function α (resp. γ) is a maximal (resp. minimal) arrival curve for A if

$$\forall s, t \in \mathbb{R}_+,\ s \leq t, \quad A(t) - A(s) \leq \alpha(t-s) \quad \big(\text{resp. } A(t) - A(s) \geq \gamma(t-s)\big).$$

Let A be an arrival flow at a given network server. We denote the output flow from this server by $\bar{A}$.

Definition 2 (Minimum simple service curve)**.** The node offers a minimum simple service curve β if $\bar{A} \geq A * \beta$.

A *backlogged period* of a server is an interval $(s, t]$ such that $\forall u \in (s, t]$, $A(u) > \bar{A}(u)$.

Definition 3 (Minimum strict service curve)**.** A minimum service curve β is *strict* if during any backlogged period $(s, t]$ of the server,

$$\bar{A}(t) - \bar{A}(s) \geq \beta(t-s).$$

Basic results of network calculus give upper bounds of the worst-case backlog, the worst-case delay and the output burstiness of a server. Those bounds are computed using a maximum arrival curve α for the input flow A and a minimum service curve β for the server.

- The backlog at time t is defined by $B(t) = A(t) - \bar{A}(t)$. The maximum backlog $B_{\max} = \sup_{t\in\mathbb{R}_+} B(t)$ is bounded as follows:

$$B_{\max} \leq \sup_{s\geq 0}[\alpha(s) - \beta(s)] = \alpha \oslash \beta(0).$$

- The virtual delay at time t is defined by $d_v(t) = \inf\{d \geq 0 \mid \bar{A}(t+d) \geq A(t)\}$. The maximum virtual delay $d_{\max} = \sup_{t\in\mathbb{R}_+} d_v(t)$ satisfies:

$$d_{\max} \leq \sup_{t\geq 0}\{\inf\{d \geq 0 \mid \beta(t+d) \geq \alpha(t)\}\}.$$

- Output burstiness: the curve $\alpha \oslash \beta$ is a maximum arrival curve for the output flow $\bar{A}$.

The difference between simple and strict service curves is important when dealing with *residual service curves*: when several flows share the same server, it may be necessary (in the case of arbitrary multiplexing or fixed priorities, for example) to have strict service curves to compute a service curve for a single flow (basically removing the arrival curve of the cross-traffic from the service curve). Unfortunately, in the case of arbitrary multiplexing, the curve obtained is not a strict service curve [12]. The next theorem provides a residual strict service curve when there already exist individual strict service curves. Doing this, those individual services are improved.

Theorem 1. *Let $A_1, A_2, \dots, A_n$ be n arrival flows to a given server that offers a minimum strict service curve β. Let $\alpha_1, \alpha_2, \dots, \alpha_n$ be maximum arrival curves for $A_1, A_2, \dots, A_n$ respectively. Let $\beta_1, \beta_2, \dots, \beta_n$ be minimum strict service curves offered by this server to $A_1 A_2, \dots, A_n$ respectively. Then, for all $i = 1, \dots, n$,*

$$\max\left((\beta - \sum_{j\neq i}\min(\alpha_j \oslash \beta_j, \alpha_j \oslash (\beta - \sum_{k\neq j}\alpha_k)^+))^+, \beta_i\right)$$

is a minimum strict service curve for A_i.

Proof. We know that β is a strict service curve, so, from [2], $(\beta - \sum_{j\neq i}\alpha_j)^+$ is a simple service curve for A_i. As a consequence, $\alpha_i \oslash (\beta - \sum_{j\neq i}\alpha_j)^+$ is a maximum arrival curve for $\bar{A}_i$. Moreover, as β_i is also a strict service curve for A_i, $\alpha_i \oslash \beta_i$ is an arrival curve for $\bar{A}_i$. Set $\tilde{\alpha}_i = \min(\alpha_i \oslash (\beta - \sum_{j\neq i}\alpha_j)^+, \alpha_i \oslash \beta_i)$.

Now, fix $i \in \{1, \dots, n\}$. Let $(s, t]$ be a backlogged period for A_i. This trivially implies that $(s, t]$ is a backlogged period for the aggregate flow. Then

$$\sum_{j=1}^{n} \bar{A}_j(t) - \bar{A}_j(s) \geq \beta(t-s),$$

and

$$\bar{A}_i(t) - \bar{A}_i(s) \geq (\beta(t-s) - \sum_{j\neq i}\bar{A}_j(t) - \bar{A}_j(s))^+.$$

It then suffices to replace $\bar{A}_j(t) - \bar{A}_j(s)$ by the constraints computed in the first part of the proof:

$$\bar{A}_i(t) - \bar{A}_i(s) \geq (\beta(t-s) - \sum_{j \neq i} \tilde{\alpha}_j(t-s))^+$$

and $(\beta - \sum_{j \neq i} \min(\alpha_j \oslash \beta_j, \alpha_j \oslash (\beta - \sum_{k \neq j} \alpha_k)^+))^+$ is a strict service curve for A_i.

The maximum of two strict service curves for a server is also a strict service curve [13], so one can conclude.

□

2 Packetization

We introduce here two new concepts: packet operators and packet curves. We give a short review on packetization and some new results for non-preemptive service, that we will use in the next sections. The objective of the new formulas introduced here is to provide a network calculus approach to compute residual services (and also delay and backlog bounds) in the case where packet data flows are served under non-preemptive packet service disciplines, with packets having different lengths.

We are concerned here with data arrival flows that arrive in packets. Thus, two types of flows can be distinguished: the flow of the amount of data itself (bits), independent of how it is clustered in packets, that we simply call the *data flow*, and the flow of the number of packets, that we call the *packet flow*. The idea is to define operators as well as minimum and maximum curves that allow us to switch from the data flow space to the packet flow space, and vice-versa. This is similar to packetization [1, 2], but we will go one step further here by introducing the constraints on packets under the concept of *packet curves*.

Packetizers describe how data is clustered in packets by an increasing sequence of packet lengths [1, 2]. We replace this sequence by a minimum and/or maximum curves that give the minimum and/or maximum number of packets in a given amount of data. This new approach is more powerful than packetization and is more in line with the network calculus approach based on constraint curves. Packet curves are used in this article to compute residual services under non-preemptive policies where packets of any data flow may have different lengths.

2.1 Packet operators

For a non-decreasing function $f : D \to E$, the non-decreasing function f^{-1} called pseudo-inverse of f, is the smallest non-decreasing function such that: $\forall x \in E$, $f^{-1}(x) = \inf\{t \in D, f(t) \geq x\}$. Thus $\forall t \in D$, $f^{-1} \circ f(t) \leq t$.

Definition 4 (Packet operator)**.** Let A be an arrival flow. The *packet operator* of A is the function $\mathcal{P} : \mathbb{R} \to \mathbb{N}$ such that for an amount x of arrival data, $\mathcal{P}(x)$ is the number of *entire* packets in x.

The amount of data contained in $\mathcal{P}(x)$ packets is at most x (since $\mathcal{P}^{-1} \circ \mathcal{P}(x) \leq x$) and by definition, $\mathcal{P}$ is *non-decreasing and right-continuous.* Let $P(t)$ be the cumulative number of entire packets arrived up to time t. We have $P = \mathcal{P} \circ A(t)$ and P is the *packet flow* of A.

The amount of data contained in n packets is $\mathcal{P}^{-1}(n)$ and $(\mathcal{P}^{-1}(n))_{n \in \mathbb{N}}$ is the non-decreasing sequence of cumulative packet lengths. If we denote this sequence by M, then the operator $\mathcal{P}^{-1} \circ \mathcal{P}$ is an M-packetizer [14, 15, 1].

Let $\mathcal{P}$ be the packet operator associated to a flow A arriving at a given server and let $\bar{A}$ denote the output flow of A from the server. If the flow A is served with the FIFO policy, then the packet operator associated to $\bar{A}$ is simply the packet operator associated to A. That is, if we denote by $\bar{\mathcal{P}}$ the packet operator associated to the output flow $\bar{A}$, then we have $\bar{\mathcal{P}} = \mathcal{P}$. That means that FIFO servers do not repacketize data. This is true only when one flow is served and when the data is served with FIFO service. We will see below that when more than one flow are served, the server often repacketizes data of the aggregate flow, depending on the service policy applied. In the whole paper, we will assume that the service policy is FIFO per flow, but not necessarily for the aggregate flow.

We recall a well-known result on packetization, based on an approach that only takes into account the maximum length of packets, $\ell^{\max}$.

Theorem 2. (Packetization, [2, Theorem 1.7.1])

1. *If β is a minimum service curve for A, then $(\beta - \ell^{\max})^+$ is a minimum service curve for $\mathcal{P}^{-1} \circ \mathcal{P} \circ A$, where $(.)^+ = \max(., 0)$.*
2. *If α is a maximum arrival curve for A, then $\alpha + \ell_{\max}$ is a maximum arrival curve for $\mathcal{P}^{-1} \circ \mathcal{P} \circ A$.*

2.2 Packet curves

For a given arrival flow A, we usually do not know the entire function A, but have some information about it, namely the average in time and the maximal variance of arrival data. This provides maximum arrival curves used to compute performance bounds. Similarly, for an arrival flow A, we are not always able to know exactly the associated packet operator $\mathcal{P}$. However, we may have some information about the distribution of short and long packets on the data. Using this information we define minimum and maximum *packet curves* that give minimum and maximum number of packets in a given amount of data.

Definition 5 (Packet curve)**.** A curve π (resp. Π) is a minimum (resp. maximum) packet curve for $\mathcal{P}$ if $\forall 0 \leq x \leq y$,

$$\mathcal{P}(y) - \mathcal{P}(x) \geq \pi(y - x) \quad \text{(resp. } \mathcal{P}(y) - \mathcal{P}(x) \leq \Pi(y - x)\text{)}.$$

Let π be a minimum packet curve for an arrival flow A, and let $\ell^{\max}$ and L denote respectively the maximum and the average packet lengths. Then we have $\pi(\ell^{\max}) \leq 1$ and $L \leq \lim_{x \to +\infty} x/\pi(x)$. A realistic example for a minimum packet curve π is $\pi(x) = \max_{i \in \{1,\ldots,k\}} U_i(x - V_i)^+$, $\forall x \in \mathbb{R}_+$, where $(U_i)_i$ and $(V_i)_i$ are increasing sequences, with $U_0 = V_0 = 0$ and $U_1 \geq 1/\ell^{\max}, V_1 = \ell^{\max}$. A realistic maximum packet curve is $\Pi(x) = \min_{i \in \{1,\ldots,k\}} \mu_i x + \nu_i$, where $(\mu_i)_i$ is a decreasing sequence and $(\nu_i)_i$ is an increasing sequence, with $\nu_0 \geq 1$ and $\mu_0 \leq 1/\ell_{\min}$.

Example 1. Consider a data arrival flow whose packets are of lengths either 1 or 2. In addition, in three successive packets, there must be at least one packet of length 1, and at least one packet of length 2. The minimum and the maximum lengths are thus given by $\ell^{\min} = 1$ and $\ell^{\max} = 2$. The best minimal packet curve for this would be the following (with $n \in \mathbb{N}$)

$$\pi_1(5n + x) = \begin{cases} 3n & \text{if } 0 \leq x < 2, \\ 3n + 1 & \text{if } 2 \leq x < 4, \\ 3n + 2 & \text{if } 4 \leq x < 5, \end{cases}$$

and the best maximum packet curve would be

$$\Pi_1(4n+x) = \begin{cases} 3n+1 & \text{if } 0 \leq x < 1, \\ 3n+2 & \text{if } 1 \leq x < 2, \\ 3n+3 & \text{if } 2 \leq x < 4. \end{cases}$$

One could also choose $\pi_2(x) = \max(0, 1/2(x-2), 3/5(x-2/3))$ and $\Pi_2(x) = \min(x+1, 3/4x+3/2)$. Those functions are depicted on Figure 1.

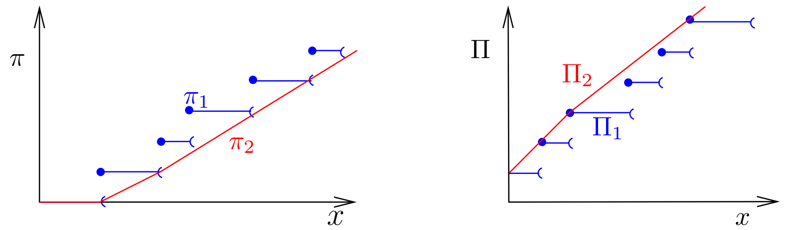

Figure 1: The packet curves π_1, π_2, Π_1 and Π_2.

The following results are direct consequences of the definitions.

Proposition 1. *If α is a maximum arrival curve for A, and Π is a maximum packet curve for $\mathcal{P}$, then $\Pi \circ \alpha$ is a maximum arrival curve for P.*

Proof. Let $s, t \in \mathbb{R}_+$, $s \leq t$, $P(t) - P(s) = \mathcal{P}(A(t)) - \mathcal{P}(A(s)) \leq \Pi(A(t) - A(s)) \leq \Pi(\alpha(t-s))$ as Π is non-decreasing. □

If $\mathcal{P}$ is given, it could be interesting to know the best minimal and maximal packet curves. For every $x, y \in \mathbb{R}_+$, we have $\pi(x) \geq \mathcal{P}(y+x) - \mathcal{P}(y)$, then the best choice for $\pi(x)$ would be $\inf_{y \geq 0} \mathcal{P}(y+x) - \mathcal{P}(y)$. Similarly, $\Pi(x) \leq \mathcal{P}(y+x) - \mathcal{P}(y)$ and we can choose $\Pi(x) = \sup_{y \geq 0} \mathcal{P}(y+x) - \mathcal{P}(y) = \mathcal{P} \oslash \mathcal{P}(x)$.

In the following two results (Theorem 3 and Theorem 4), we show how to switch, from the service viewpoint, between a data flow and its associated packet flow.

Theorem 3. *Let β be a minimum (strict) service curve for A, and π be a minimum packet curve for $\mathcal{P}$, then $\pi \circ \beta$ is a minimum (strict) service curve for P.*

Proof. If β is a minimum service curve for A, then, $\forall t \in \mathbb{R}_+$, $\bar{A}(t) \geq \inf_{0 \leq s \leq t} A(s) + \beta(t-s)$. Since $\mathcal{P}$ is right-continuous, we have

$$\begin{aligned} \bar{P}(t) = \mathcal{P}(\bar{A}(t)) &\geq \mathcal{P}(\inf_{0 \leq s \leq t} A(s) + \beta(t-s)) \\ &\geq \inf_{0 \leq s \leq t} \mathcal{P}(A(s) + \beta(t-s)) \\ &\geq \inf_{0 \leq s \leq t} [\mathcal{P}(A(s)) + \mathcal{P}(A(s) + \beta(t-s)) - \mathcal{P}(A(s))] \\ &\geq \inf_{0 \leq s \leq t} P(s) + \pi(\beta(t-s)). \end{aligned}$$

If β is a strict service curve, then let $(s, t]$ be a backlogged period of A.

$$\bar{P}(t) - \bar{P}(s) = \bar{\mathcal{P}}(\bar{A}(t)) - \bar{\mathcal{P}}(\bar{A}(s)) \geq \pi(\bar{A}(t) - \bar{A}(s)) \geq \pi(\beta(t-s)).$$

□

Theorem 4. *Let β be a minimum simple (resp. strict) service curve for a packet flow $P = \mathcal{P} \circ A$ and π and Π be minimum and maximum packet curves for $\mathcal{P}$. Then, $(\Pi)^{-1} \circ \lfloor \beta \rfloor$ (resp. $(\Pi)^{-1} \circ \lceil \beta \rceil$) is a minimum simple (resp. strict) service curve for $(\mathcal{P})^{-1} \circ \mathcal{P} \circ A$.*

Proof. First note that $(\mathcal{P})^{-1} \circ \mathcal{P} \circ A = (\mathcal{P})^{-1} \circ P$.

Let $p \leq q \in \mathbb{N}$ and $x = \mathcal{P}^{-1}(p)$ and $y = \mathcal{P}^{-1}(q)$. Then, we have $p = \mathcal{P}(x)$ and $q = \mathcal{P}(y)$, and $q - p \leq \Pi(y - x)$ and $\Pi^{-1}(q - p) \leq y - x = \mathcal{P}^{-1}(q) - \mathcal{P}^{-1}(p)$.

Now, if β is a service curve for the packet flow P, then for every $t \in \mathbb{R}_+$, there exists s such that $\bar{P}(t) \geq P(s) + \lfloor \beta(t - s) \rfloor$ (as the infimum in the convolution formula in on a finite set of integers is thus finite, it is a minimum). Then, $\mathcal{P}^{-1}(\bar{P}(t)) - \mathcal{P}^{-1}(P(s)) \geq \Pi^{-1}(\bar{P}(t) - P(s)) \geq \Pi^{-1}(\lfloor \beta(t - s) \rfloor)$.

If β is a strict service curve for the packet flow P, for every s, t such that $(s, t]$ is a backlogged period, $\bar{P}(t) - \bar{P}(s) \geq \lceil \beta(t - s) \rceil$ (as $\bar{P}$ takes its values in $\mathbb{N}$), and then, $\mathcal{P}^{-1}(\bar{P}(t)) - \mathcal{P}^{-1}(\bar{P}(s)) \geq \Pi^{-1}(\lceil \beta(t - s) \rceil)$. □

2.3 Aggregate packetization

As mentioned above, when two flows are aggregated and served by the same server, it may be useful to compute the global packet curve of the aggregate flow. In Proposition 2, we compute a global packet curve of the aggregate flow for any order of arrival (or of service). In other words, it will also be a packet curve for the aggregated departure flow.

Proposition 2. (Aggregated packetizing) *If π_1 and π_2 are respectively packet curves for $\mathcal{P}_1$ and $\mathcal{P}_2$, then $\pi_1 * \pi_2$ is a packet curve for $\mathcal{P}$ and $\bar{\mathcal{P}}$, the respective packet operators for the aggregate arrival and departure flows.*

Proof. Let x and y be two amounts of data of the aggregate output flow, with $0 \leq x \leq y$. There exists x_1 and x_2, such that $x_1 + x_2 = x$ where x_1 is the amount of data of the first flow and x_2 is the amount of data in the second flow. Similarly, there exists y_1 and y_2, with $y_1 + y_2 = y$, giving the amount of data of flows 1 and 2 respectively. We have $x_1 \leq y_1$ and $x_2 \leq y_2$.

Then we have $\bar{\mathcal{P}}(y) - \bar{\mathcal{P}}(x) = \bar{\mathcal{P}}_1(y_1) + \bar{\mathcal{P}}_2(y_2) - \bar{\mathcal{P}}_1(x_1) - \bar{\mathcal{P}}_2(x_2) = \mathcal{P}_1(y_1) + \mathcal{P}_2(y_2) - \mathcal{P}_1(x_1) - \mathcal{P}_2(x_2) \geq \pi_1(y_1 - x_1) + \pi_2(y_2 - x_2) \geq \min_{0 \leq z \leq y - x}[\pi_1(z) + \pi_2(y - x - z)] = (\pi_1 * \pi_2)(y - x)$. □

3 Superposition of periodic flows

In this section we apply the previous construction for a flow made of a superposition of periodic flows. This case study is useful especially for real-time systems where it is very common. It also illustrates the gain of the packet curves over classical packetization in describing the behavior of packets.

We consider that the arrival flow is composed of a superposition of N periodic elementary flows. Let us denote by T_n the period of elementary flow n, by S_n the size of all the packets in flow n and $\phi_n (< T_n)$ the phase of flow n (*i.e.* the arrival time of the first packet from elementary flow n.)

The elementary flows are ordered with non-decreasing phases:

$$0 \leq \phi_1 \leq \phi_2 \leq \cdots \leq \phi_N.$$

The data arrival flow is

$$A(t) = \sum_{n=1}^{N} \left\lfloor \frac{t + T_n - \phi_n}{T_n} \right\rfloor S_n.$$

Note that every term of this sum is non-negative. Let us now consider L the affine lower approximation of A, and U the affine upper approximation of A, defined by

$$L(t) \stackrel{\text{def}}{=} \sum_{n=1}^{N} \frac{t - \phi_n}{T_n} S_n \qquad \text{and} \qquad U(t) \stackrel{\text{def}}{=} \sum_{n=1}^{N} \frac{t - \phi_n + T_n}{T_n} S_n.$$

We have

$$L(t) < A(t) \leq U(t).$$

A simple inversion gives

$$L^{-1}(x) = \frac{x + \sum_{n=1}^{N} \phi_n \rho_n}{\sum_{n=1}^{N} \rho_n} \qquad \text{and} \qquad U^{-1}(x) = \frac{x + \sum_{n=1}^{N} (\phi_n - T_n)\rho_n}{\sum_{n=1}^{N} \rho_n},$$

where $\rho_n = S_n / T_n$.

The packet flow $P(t)$ is

$$P(t) = \sum_{n=1}^{N} \left\lfloor \frac{t + T_n - \phi_n}{T_n} \right\rfloor.$$

As $P(t) = \mathcal{P}(A(t))$, and A is right-continuous, $A \circ A^{-1}(x) = x$ and $\mathcal{P}(x) = P(A^{-1}(x))$. Therefore, the packet operator $\mathcal{P}$ satisfies

$$\sum_{n=1}^{N} \left\lfloor \frac{U^{-1}(x) + T_n - \phi_n}{T_n} \right\rfloor \leq \mathcal{P}(x) < \sum_{n=1}^{N} \left\lfloor \frac{L^{-1}(x) + T_n - \phi_n}{T_n} \right\rfloor.$$

Hence

$$\sum_{n=1}^{N} \left\lfloor \frac{U^{-1}(x) + T_n - \phi_n}{T_n} \right\rfloor \leq \mathcal{P}(x) \leq \sum_{n=1}^{N} \left\lfloor \frac{L^{-1}(x) + T_n - \phi_n}{T_n} \right\rfloor - 1. \tag{1}$$

Which can also be written

$$\sum_{n=1}^{N} \left\lfloor \frac{U^{-1}(x) - \phi_n}{T_n} \right\rfloor + N \leq \mathcal{P}(x) \leq \sum_{n=1}^{N} \left\lfloor \frac{L^{-1}(x) - \phi_n}{T_n} \right\rfloor + N - 1. \tag{2}$$

Then, the best maximum and minimum packets curves are obtained by using the formulas introduced in the previous section:

$$\pi(x) = \inf_{y \geq 0} \mathcal{P}(y + x) - \mathcal{P}(y),$$
$$\Pi(x) = \sup_{y \geq 0} \mathcal{P}(y + x) - \mathcal{P}(y) = \mathcal{P} \oslash \mathcal{P}(x).$$

The simplest case is when the flow is made of a single period flow ($N = 1$). In that case, $\mathcal{P}(x) = \lfloor x/S_1 \rfloor$ (the left and the right bounds in (2) coincide), and the gain over classical packetization is nil.

On the other hand, if $N \neq 1$ then the gain over packetization on the arrival curve can be arbitrary large.

If the phases ϕ_n are not known, simple maximum and minimum packets curves can be computed.

$$\begin{aligned}
\mathcal{P}(y) - \mathcal{P}(x) &\leq \sum_{n=1}^{N} \left\lfloor \frac{L^{-1}(y) - \phi_n}{T_n} \right\rfloor - \sum_{n=1}^{N)} \left\lfloor \frac{U^{-1}(x) - \phi_n}{T_n} \right\rfloor - 1 \\
&\leq \sum_{n=1}^{N} \left\lceil \frac{L^{-1}(y) - U^{-1}(x)}{T_n} \right\rceil - 1 \\
&= \sum_{n=1}^{N} \left\lceil \frac{y - x + \sum_{i=1}^{N} T_i\rho_i}{T_n \sum_{i=1}^{N} \rho_i} \right\rceil - 1.
\end{aligned}$$

as for the minimum packet curve,

$$\begin{aligned}
\mathcal{P}(y) - \mathcal{P}(x) &\geq \sum_{n=1}^{N} \left\lfloor \frac{U^{-1}(y) - \phi_n}{T_n} \right\rfloor - \sum_{n=1}^{N} \left\lfloor \frac{L^{-1}(x) - \phi_n}{T_n} \right\rfloor + 1 \\
&\geq \sum_{n=1}^{N} \left\lfloor \frac{y - x - \sum_{i=1}^{N} T_i\rho_i}{T_n \sum_{i=1}^{N} \rho_i} \right\rfloor + 1.
\end{aligned}$$

Example 2. Let us take two flows ($N = 2$) with phases $\phi_1 = 1, \phi_2 = 2$, periods $T_1 = 3, T_2 = 4$, and packet lengths $S_1 = 2, S_2 = 3$.

$$\begin{aligned}
A(t) &= 2 \lfloor (t+2)/3 \rfloor + 3 \lfloor (t+2)/4 \rfloor \\
\pi(x) &= \lfloor 4(x-5)/17 \rfloor + \lfloor 3(x-5)/17 \rfloor + 1 \\
\Pi(x) &= \lceil 4(x+5)/17 \rceil + \lceil 3(x+5)/17 \rceil - 1.
\end{aligned}$$

4 Non-preemptive service

4.1 General approach

In this section, we explain how packet curves are used in aggregate services with non-preemptive policies. Let $(A_i)_{i\in I}$ be a finite family of arrival flows at a given server, with a service curve β for the aggregate flow $A = \sum_{i\in I} A_i$. The flows A_i, $i \in I$ are served under a non-preemptive service policy. In the remaining of the paper, we will always denote by P (resp. P_i) the arrival packet flows of A (resp. A_i), by $\bar{A}$ (resp. $\bar{A}_i$) its output flow and by $\bar{P}$ (resp. $\bar{P}_i$) its output packet flow. Moreover, the packet operator of A_i (and $\bar{A}_i$) is denoted by $\mathcal{P}_i$. Then we proceed as follows (see Figure 2 for two flows).

1. From maximum arrival curves $\alpha_i, i \in I$ and maximum packet curves $\Pi_i, i \in I$ of $A_i, i \in I$, maximum arrival curves for the packet flows $P_i, i \in I$ are $\Pi_i \circ \alpha_i, i \in I$, given by Proposition 1.

2. We compute a service curve $\beta' = *_{i=1}^{n} \pi_i \circ \beta$ for the aggregate packet flow P using Theorem 3.

3. Using β', we calculate the residual services β'_i for the flows P_i, $i \in I$. The flows P_i being packet flows, this curve concerns the number of packets served and not the quantity of data served.

4. From the residual services of P_i, we deduce β_i, the residual services for A_i using Theorem 4.

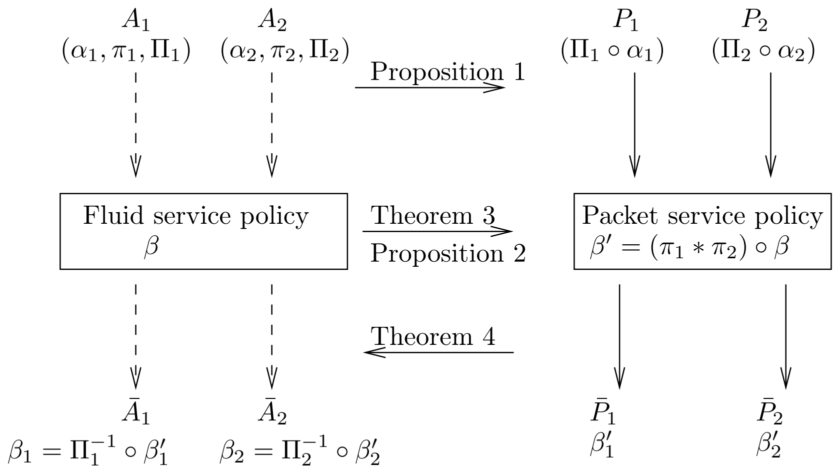

Figure 2: Non-preemptive service calculus scheme.

4.2 Packet curve approach *versus* the fluid approach

We have defined a procedure to compute residual service curves taking into account the size of packets more precisely than with the existing approaches (namely using Theorem 2, or existing results, which we call the fluid approach). A natural question is to compare the two approaches. We will see that in some cases, when the service policy does not involve counting of packets, the fluid approach is better, whereas our approach will be better with policies that are based on packets counting. Indeed, when only the amount of data arrived until a certain date is involved, the way the packets arrive is not important and then, only the maximum packet size will matter. This is illustrated with an example. Here, for the sake of simplicity, we will only consider affine and rate-latency curves.

Fixed priorities

Consider the case of fixed priorities. Two flows arrive in a server offering a strict service curve $\beta : t \mapsto R(t-T)^+$. Flow i admits an arrival curve $\alpha_i : t \mapsto \sigma_i + \rho_i t$, a minimum packet curve $\pi : x \mapsto U_i(x - V_i)^+$ and a maximum packet curve $\Pi : x \mapsto \nu_i + \mu_i x$. Flow 1 has the priority. The maximum packet size of flow 2 is $\ell_2^{\max} \leq V_2$ and the maximum packet size for the two flows is $\ell^{\max}$.

The remaining strict service curve β_1^f for flow 1, computed with the fluid approach, is

$$\beta_1^f(t) = (\beta - \ell_2^{\max})^+ = R\left(t - T - \frac{\ell_2^{\max}}{R}\right)^+.$$

With the packet-based approach, one has first to compute the service curve for the packet flow: $\beta'(t) = (\pi_1 * \pi_2)(\beta(t)) = \min(U_1, U_2)(\beta(t) - V_1 - V_2) = R\min(U_1, U_2)(t - T - \frac{V_1+V_2}{R})^+$. Now, $(\beta' - 1)^+$ is a strict service curve for packet flow 1: $(\beta' - 1)^+(t) = R\min(U_1, U_2)(t - T - \frac{V_1+V_2}{R} - \frac{1}{R\min(U_1,U_2)})^+$. Finally, one can apply Π_1^{-1} to get back to the data flow. Thus a strict service curve β_1^p for flow 1, computed with the packet-based approach is

$$\beta_1^p(t) = \frac{R\min(U_1, U_2)}{\mu_1}\left(t - T - \frac{V_1 + V_2}{R} - \frac{1 + \nu_1}{R\min(U_1, U_2)}\right)^+.$$

We have $\ell_2^{\max} \leq V_2$ and, since $\mu_1 \geq U_1$, $\frac{R\min(U_1,U_2)}{\mu_1} \leq R$. So $\beta_1^f \geq \beta_1^p$ and the fluid approach is better than the packet-based one.

By the fluid approach of [12], the remaining strict service curve for flow 2 is

$$\beta_2^f(t) = (\beta - \alpha_1 - \ell_2^{\max})^+ = (R - \rho_1)\left(t - T - \frac{\sigma + \rho_1 T + \ell_2^{\max}}{R - \rho_1}\right)^+,$$

whereas, with the packet-based approach, it is given by

$$\beta_2^p(t) = \frac{R\min(U_1, U_2) - \mu_1\rho_1}{\mu_2} \left[t - (T + \frac{V_1 + V_2}{R} + \frac{\mu_1\sigma_1 + \nu_1 + \nu_2 + \rho_1\mu_1(T + (V_1 + V_2)/2)}{R\min(U_1, U_2) - \mu_1\rho_1}\right]^+.$$

Then, $\beta_2^f \geq \beta_2^p$. Indeed, the computation with packet curves converts the data flow into the packet flow as choosing long packets first, and then converts the packet flow into a data flow using the short packets first. The two steps are done independently, so long packets are converted into short packets.

For any service policy which is based on the quantity of data rather than on the number of packets (for instance, there is no difference for the service if one packet of length ten arrives or if ten packets of size one arrive at the same time), the fluid approach will be better than the packet based one. Another example of this is the FIFO policy.

Some service policies are based on counting packets rather than on the amount of data. An example is the round-robin policy.

Round-Robin service discipline

Round-robin is a service policy that assigns service to each flow in a circular order, *without priority.* The order is respected whenever possible: if one flow is out of packets, the next flow, according to the defined order, takes its place. A separate flow is considered for every data stream, and the server serves a packet from any non-empty queue encountered, following a cyclic order.

Proposition 3. *Consider n packet flows $P_1, \ldots, P_n$ arriving to a server with strict service curve β. If the service policy is round-robin, then for all $i = 1, \ldots, n$, the curve β_i defined by*

$$\beta_i(t) = \left(\frac{1}{n}\beta(t) - 1\right)^+$$

is a strict service curve for P_i.

Proof. As β is a strict service curve, we have $\forall s \leq t$ such that $(s,t]$ is a backlogged period for P_1 (and then for the whole process aggregate flow),

$$\sum_{i=1}^{n}(\bar{P}_i(t) - \bar{P}_i(s)) \geq \beta(t-s).$$

As the service policy is Round-Robin, we know that $\sum_{i=1}^{n}(\bar{P}_i(t)-\bar{P}_i(s)) \leq n(\bar{P}_1(t)-\bar{P}_1(s))+n$. Indeed, one packet of each flow is served in rounds, and there are always available packets for P_1. Then,

$$\bar{P}_1(t) - \bar{P}_1(s) \geq \frac{\beta(t-s)-n}{n},$$

and $\left(\frac{1}{n}\beta(t)-1\right)^+$ is a strict service curve for P_1. □

Example 3. Let us take a simple case with $n=2$, $\beta(t) = R(t-T)^+$, $\alpha_1(t) = \sigma_1 + \rho_1 t$, $\alpha_2(t) = \sigma_2 + \rho_2 t$. Then it is easy to check that

$$\beta_1(t) = \beta_2(t) = \frac{R}{2}\left(t - T - \frac{2}{R}\right)^+$$

is a strict minimum service curve for P_1 and P_2.

Now we compute the service curve for the arrival flows with packets of different lengths. We suppose that minimum and maximum packet curves π_i and Π_i, and maximum arrival curves $\alpha_i, 1 \leq i \leq n$ are respectively associated to the data flows $A_i, 1 \leq i \leq n$. We first do the computation by applying the packet-based approach explained in the subsection 4.1. Then we present a method based on the adaptation of Proposition 3 to the case where packets have different lengths and where packet curves are given. Finally, we use the method that we called the fluid method, and make a comparison of those three methods.

Theorem 5 below is the application of the packet-based approach to the round-robin discipline.

Theorem 5. *If β is a strict service curve for the aggregate data flow, then for all $i \in \{1,\dots,n\}$, the curve*

$$\beta_i = \max\left[(\tilde{\beta} - \sum_{j\neq i}\min\left[\tilde{\alpha}_j \oslash \tilde{\beta}_j, \tilde{\alpha}_j \oslash (\tilde{\beta} - \sum_{k\neq j}\tilde{\alpha}_k)^+\right])^+, \tilde{\beta}_i\right],$$

with

$$\tilde{\beta}_i = (\Pi_i)^{-1}\left(\frac{1}{n}((*_{i=1}^{n}\pi_i)\circ\beta) - 1\right)^+,$$

$\tilde{\alpha}_i = \alpha_i + \ell_i^{\max}$ *and* $\tilde{\beta} = (\beta - \ell^{\max})^+$, *is a minimum strict service curve for* $(\mathcal{P}_i)^{-1} \circ \mathcal{P}_i \circ A_i$.

Note that we use the packet curves only for computing $\tilde{\beta}_i$, as the curves β_i are based on the blind multiplexing, which is not a packet-based policy.

Proof. We apply the scenario given in Section 4 to compute $\tilde{\beta}_i$:

- By Proposition 2, $*_{i=1}^{n}\pi_i$ is a minimum packet curve for $\bar{A}$;

- By Theorem 3, $(*_{i=1}^{n}\pi_i)\circ\beta$ is a minimum strict service curve for P;
- By Proposition 3, $\left(\frac{1}{n}((*_{i=1}^{n}\pi_i)\circ\beta)-1\right)^{+}$ is a minimum strict service curve for P_i for $1\leq i\leq n$;
- By Theorem 4, $(\Pi_i)^{-1}\left(\frac{1}{n}((*_{i=1}^{n}\pi_i)\circ\beta)-1\right)^{+}$ is a minimum strict service curve for $(\mathcal{P}_i)^{-1}\circ\mathcal{P}_i\circ A_i$ for $1\leq i\leq n$.

Then we apply Theorem 1 taking into account the maximum packet lengths for each flow (Theorem 2).

□

Another *ad-hoc* method would be to adapt the proof of Proposition 3 with packet curves directly.

Proposition 4. *Consider a server offering a strict service curve β crossed by n flows with round-robin service policy. Let ψ be a curve defined by*

$$\psi = Id + \sum_{i=2}^{n}(\pi_i^{-1}\circ\Pi_1 + 1),$$

where Id denotes the identity map. Then $\psi^{-1}(\beta-\ell^{\max})$ is a strict service curve for A_1, and $(\psi^{-1}(\beta-\ell^{\max}))^{+}$ is a strict service curve for $\mathcal{P}_1^{-1}\circ\mathcal{P}_1\circ A_1$.

Proof. We recall that we assume the packet curves to be right-continuous. Let $(s,t]$ be a backlogged period for flow A_1. Then, $\sum_{i=1}^{n}\bar{A}_i(t)-\bar{A}_i(s)\geq\beta(t-s)$. Moreover

$$\sum_{i=1}^{n}\mathcal{P}_i^{-1}\circ\mathcal{P}_i\circ\bar{A}_i(t)-\mathcal{P}_i^{-1}\circ\mathcal{P}_i\circ\bar{A}_i(s)\geq\beta(t-s)-\ell^{\max}.$$

During $(s,t]$, at most $\Pi_1(\mathcal{P}_1^{-1}\circ\mathcal{P}_1\circ\bar{A}_1(t)-\mathcal{P}_1^{-1}\circ\mathcal{P}_1\circ\bar{A}_1(s))$ packets have been served for flow 1. So, the round-robin policy imposes that at most $\Pi_1(\mathcal{P}_1^{-1}\circ\mathcal{P}_1\circ\bar{A}_1(t)-\mathcal{P}_1^{-1}\circ\mathcal{P}_1\circ\bar{A}_1(s))+1$ packets have been served for any other flow i (with $i\neq 1$). Then

$$\sum_{i=2}^{n}\mathcal{P}_i^{-1}\circ\mathcal{P}_i\circ\bar{A}_i(t)-\mathcal{P}_i^{-1}\circ\mathcal{P}_i\circ\bar{A}_i(s)$$
$$\leq\sum_{i=2}^{n}\pi_i^{-1}(\Pi_1(\mathcal{P}_1^{-1}\circ\mathcal{P}_1\circ\bar{A}_1(t)-\mathcal{P}_1^{-1}\circ\mathcal{P}_1\circ\bar{A}_1(s))+1),$$

Thus

$$\sum_{i=1}^{n}\mathcal{P}_i^{-1}\circ\mathcal{P}_i\circ\bar{A}_i(t)-\mathcal{P}_i^{-1}\circ\mathcal{P}_i\circ\bar{A}_i(s)\leq\psi(\mathcal{P}_1^{-1}\circ\mathcal{P}_1\circ\bar{A}_1(t)-\mathcal{P}_1^{-1}\circ\mathcal{P}_1\circ\bar{A}_1(s)),$$

So,

$$\mathcal{P}_1^{-1}\circ\mathcal{P}_1\circ\bar{A}_1(t)-\mathcal{P}_1^{-1}\circ\mathcal{P}_1\circ\bar{A}_1(s)\geq\psi^{-1}(\beta(t-s)-\ell^{\max}).$$

□

Using the fluid method, one would get

$$\tilde{\beta}_i^f = \left(\frac{\ell_i^{\min}}{n\ell^{\max}}\beta - \ell^{\max}\right)^+.$$

Note that the fluid method requires the knowledge of $\ell_i^{\min}$ and $\ell^{\max}$.

Example 4. We take a simple example with $n = 2, \beta(t) = R(t-T)^+$, $\alpha_i(t) = \sigma_i + \rho_i t, \pi_i(x) = U_i(x - V_i)^+$ and $\Pi_i(x) = \nu_i + \mu_i x$ for $1, 2$. We compare $\tilde{\beta}_1$ obtained by the three approaches, denoted respectively by $\tilde{\beta}_1^p$ the curve obtained by the packet-based approach (Theorem 5), by $\tilde{\beta}_1^r$ the curve obtained by Proposition 4 and by $\tilde{\beta}_1^f$ the one obtained by the fluid approach. We obtain the curves $\tilde{\beta}_1^p(t) = R_1^p(t-T_1^p)^+$, $\tilde{\beta}_1^{ah}(t) = R_1^{ah}(t-T_1^{ah})^+$ and $\tilde{\beta}_1^f(t) = R_1^f(t-T_1^f)^+$, with

$$\begin{array}{ll} R_1^p = \frac{R\min(U_1,U_2)}{2\mu_1}, & T_1^p = T + \frac{V_1+V_2}{R} + \frac{2+2\nu_1}{R\min(U_1,U_2)}, \\ R_1^{ah} = \frac{RU_2}{\mu_1+U_2}, & T_1^{ah} = T + \frac{U_2\ell^{\max}+U_2V_2+\nu_1+1}{U_2R}, \\ R_1^f = \frac{R\ell_1^{\min}}{2\ell^{\max}}, & T_1^f = T + \frac{2(\ell^{\max})^2}{R\ell_1^{\min}}. \end{array}$$

Simple computations lead to $R_1^f \leq R_1^p \leq R_1^{ah}$. As a consequence, the long term service rate obtained with the *ad-hoc* method is better than the one obtained with the generic packet-based method, which is in turn better than the one obtained with the fluid methods. The latencies cannot be compared. However, since the curves are strict, we also know that the curve $\max(\tilde{\beta}_1^p, \tilde{\beta}_1^r, \tilde{\beta}_1^f)$ is a strict service curve for $\mathcal{P}_1^{-1} \circ \mathcal{P}_1 \circ A_1$.

5 Shared queues

In this section, we consider the following problem: let A_1 and A_2 be two packetized data flows arriving to the same server. The server serves the packets one by one, and each time it finishes the service of one packet, it picks another packet of either one flow or the other (we still assume FIFO service per flow). Now, the servers provide a different service for those flows: packets of flow A_1 have a strict minimum service curve β_1 and packets of flow A_2 have a strict minimum service curve β_2. When finishing the service of one packet, if the next one is from a different flow, then the service is reinitiated (as if the switching time is the beginning of a backlogged period); see Figure 3.

The aim of this section is to give means to compute a global strict service curve for the aggregate flow in this server. Note that this question is not relevant if there is no packetization, or if packets could have arbitrarily small sizes, in which case, the only possible service curve would be zero.

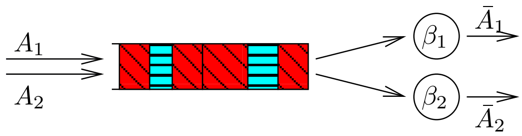

Figure 3: Shared queues: the two servers cannot be active at the same time.

Let us first study the strict service curve of one flow, say A_1, in the case where the service is re-initiated at each packet. This is the worst-case scenario, as β_1 can always be considered super-additive ($\forall s,t \geq 0,\ \beta_1(s)+\beta_1(t) \leq \beta_1(s+t)$). More precisely, we first focus on the latest time t_a at which the amount of data a can be served.

Lemma 1. *Consider a server that offers a strict service curve β, whose service is reinitiated for each packet (the start of the service of a packet is seen as the start of a backlogged period), and let A be a flow crossing that server, with a maximum arrival curve α, minimum and maximum packet curves π and Π. Then, $\tilde{\beta}$ defined by*

$$\tilde{\beta}^{-1}(x) = \left(\beta^{(\lfloor \Pi(x) \rfloor)}\right)^{-1} (\Pi^{-1}(\lfloor \Pi(x) \rfloor)) + \beta^{-1}(\pi^{-1}(0^+))$$

is a strict service curve for A (with no reinitialization of the service).

Proof. As A has a maximum packet curve Π, we know that the number of packets in the amount of data x is at most $\lfloor \Pi(x) \rfloor$. Then, the amount of data in $\lfloor \Pi(x) \rfloor$ packets is at least $\Pi^{-1}(\lfloor \Pi(x) \rfloor)$. After serving the $\lfloor \Pi_1(x) \rfloor$ first packets, to serve the amount of data x, the server needs the time to serve one packet. In the worst case, this packet is of maximum size, $\ell^{\max} \leq \pi^{-1}(0^+)$. If we denote by $\tilde{\beta}$ the resulting minimum strict service curve for A, taking into account the reinitialization of the service, then we have

$$t_x \overset{def}{=} \tilde{\beta}^{-1}(x) = \left(\beta^{(\lfloor \Pi(x) \rfloor)}\right)^{-1} (\Pi^{-1}(\lfloor \Pi(x) \rfloor)) + \beta^{-1}(\pi^{-1}(0^+)).$$

□

Example 5. Suppose that $\beta(t) = R(t-T)^+$. As β is convex,

$$\beta^{(\lfloor \Pi(x) \rfloor)}(t) = \lfloor \Pi(x) \rfloor \beta(\frac{t}{\lfloor \Pi(x) \rfloor}) = R(t - \lfloor \Pi(x) \rfloor T)^+.$$

Then

$$\left(\beta^{(\lfloor \Pi(x) \rfloor)}\right)^{-1} (\Pi^{-1}(\lfloor \Pi(x) \rfloor)) = \frac{1}{R}\Pi^{-1}(\lfloor \Pi(x) \rfloor) + \lfloor \Pi(x) \rfloor T$$

and

$$\begin{aligned} t_x &= \tfrac{1}{R}\Pi^{-1}(\lfloor \Pi(x) \rfloor) + \lfloor \Pi(x) \rfloor T + T + V/R \\ &\leq \tfrac{1}{R}x + \Pi(x)T + T + V/R. \end{aligned}$$

One can then choose

$$\tilde{\beta}(t) = \frac{R}{1+\mu TR}\left(t - (\nu T + T + V/R)\right)^+.$$

If only the information about the largest and smallest packet lengths is known, then one can apply the formula with $\pi(x) = \frac{1}{\ell^{\max}}(x-\ell^{\max})^+$ and $\Pi(x) = 1 + \frac{x}{\ell^{\min}}$. Then, one finds: $\tilde{\beta}^f(t) = \frac{R\ell^{\min}}{\ell^{\min}+RT}(1 - 2T - \ell^{\max})$. The service rate is then larger with the packet approach than with the fluid approach. When Π and π are well-chosen ($\pi(0) = 1$ and $\pi^{-1}(0+) = \ell^{\max}$, $\tilde{\beta} \geq \tilde{\beta}^f$.

Theorem 6. *Let us consider n servers. Each server i, $i \in \{1,\dots,n\}$, offers a strict service curve β_i and is crossed by a data flow A_i with minimum and maximum packet curves π_i and Π_i respectively. If those servers share a unique queue, then the system for the aggregate flow offers a strict service curve $\tilde{\beta} = *_{i=1}^n \tilde{\beta}_i$, where $\tilde{\beta}_i$ is defined as in Lemma 1.*

Proof. In a time interval of length t, if the time of service of each server is t_i with $\sum_{i=1}^n t_i = t$, then, the quantity of data processed is $\sum_{i=1}^n \tilde{\beta}_i(t_i)$. Then the quantity of data processed is at least $\inf_{t_i \mid \sum_i t_i = t} \sum_{i=1}^n \tilde{\beta}_i(t_i)$. □

This solution is rather pessimistic, as it will take into account only the slowest server, but if only one flow is served, the server does not need to be re-initiated, then the service will be better than the one computed.

We now present a better solution when $n = 2$. Let us still compute the last instant at which the amount of data x can be served (thus one first compute $\tilde{\beta}^{-1}$). This amount can be decomposed in $x = x_1 + x_2$ where at least one of x_1 or x_2 represents a number of entire packets. If the number of packets served are respectively n_1 and n_2, then the number of switches between serving flow A_1 and serving flow A_2 is at most $2\min(n_1, n_2)$.

As a consequence, the time to serve the quantity of data x_1, not taking into account the possible non-entire packet is at most

$$t_{x_1} = \left(\beta_1^{\min(\lfloor \Pi_1(x_1)\rfloor, \lfloor \Pi_2(x_2)\rfloor)}\right)^{-1} (\Pi_1^{-1}(\lfloor \Pi_1(x_1)\rfloor)).$$

The sequence β^k is non-increasing with k. Then, $(\beta^k)^{-1}$ is non-decreasing with k and

$$\left(\beta_1^{\min(\lfloor \Pi_1(x_1)\rfloor, \lfloor \Pi_2(x_2)\rfloor)}\right)^{-1} = \min\left(\beta_1^{\lfloor \Pi_1(x_1)\rfloor}\right)^{-1}, \left(\beta_1^{\lfloor \Pi_2(x_2)\rfloor}\right)^{-1}.$$

Now, taking into account the two flows and the non-entire packet, one gets

$$\tilde{\beta}^{-1}(x) = \max_{x_1+x_2=x} (\tau_1(x_1, x_2) + \tau_2(x_1, x_2) + \tau_3), \tag{3}$$

where

$$\begin{aligned}
\tau_1(x_1, x_2) &= \left(\beta_1^{\min(\lfloor \Pi_1(x_1)\rfloor, \lfloor \Pi_2(x_2)\rfloor)}\right)^{-1} (\Pi_1^{-1}(\lfloor \Pi_1(x_1)\rfloor)),\\
\tau_2(x_1, x_2) &= \left(\beta_2^{\min(\lfloor \Pi_1(x_1)\rfloor, \lfloor \Pi_2(x_2)\rfloor)}\right)^{-1} (\Pi_2^{-1}(\lfloor \Pi_2(x_2)\rfloor)) \text{ and}\\
\tau_3 &= \max(\beta_1^{-1}(\pi_1^{-1}(0^+)), \beta_2^{-1}(\pi_2^{-1}(0^+))).
\end{aligned}$$

Example 6. Consider the simplest case where every curve is affine or rate-latency. Similar computations as in Example 5 give

$$\begin{aligned}
\tau_1(x_1, x_2) &= \frac{\Pi_1^{-1}(\lfloor \Pi_1(x_1)\rfloor}{R_1} + \min(\lfloor \Pi_1(x_1)\rfloor, \lfloor \Pi_2(x_2)\rfloor) T_1\\
&\leq \frac{x_1}{R_1} + T_1 \min(\nu_1 + x_1\mu_1, \nu_2 + (x - x_1)\mu_2).
\end{aligned}$$

Similarly τ_2 satisfies

$$\tau_2(x_1, x_2) \leq \frac{x_2}{R_2} + T_2 \min(\nu_1 + x_1\mu_1, \nu_2 + (x - x_1)\mu_2).$$

and

$$\tau_3 = \max(\frac{V_1}{R_1} + T_1, \frac{V_2}{R_2} + T_2).$$

Then we get

$$t_x \leq \max_{0 \leq x_1 \leq x} \left\{ \frac{x_1}{R_1} + \frac{x - x_1}{R_2} + (T_1 + T_2) \min(\nu_1 + x_1\mu_1, \nu_2 + (x - x_1)\mu_2) \right\} + \tau_3.$$

Then the following four cases can occur:

- If $(T_1+T_2)\mu_2 \leq \frac{R_2-R_1}{R_1R_2}$, then the maximum is obtained for $x_1 = x$ and we have

$$t_x \leq \frac{x}{R_1} + (T_1+T_2)\min(\nu_1+\mu_1 x, \nu_2) + \tau_3.$$

Then,

$$\tilde{\beta}(t) = \max \left\{ \begin{array}{l} \frac{R_1}{1+(T_1+T_2)\mu_1 R_1}(t-[(T_1+T_2)\nu_1+\tau_3])^+, \\ \\ R_1(t-[(T_1+T_2)\nu_2+\tau_3])^+ \end{array} \right\}.$$

Intuitively, this case occurs when the service of the second flow is high and packets of that flow are long compared to the service of flow 1. Then, asymptotically, only packets of flow 1 are served (and there is no switching between the flows), and the service rate is R_1.

- If $(T_1+T_2)\mu_1 \leq \frac{R_1-R_2}{R_1R_2}$, this case is symmetric regarding the previous one:

$$\tilde{\beta}(t) = \max\{\frac{R_2}{1+(T_1+T_2)\mu_2 R_2}(t-[(T_1+T_2)\nu_2+\tau_3])^+, \qquad R_2(t-[(T_1+T_2)\nu_1+\tau_3])^+\}.$$

- Otherwise and if $\nu_1 \geq \nu_2$, then we have $\mu_2 > 0$ and
  - For $x \leq \frac{\nu_1-\nu_2}{\mu_2}$, the maximum is obtained for $x_1 = 0$ and we get

$$t_x \leq \frac{1+(T_1+T_2)R_2\mu_2}{R_2}x + (T_1+T_2)\nu_2 + \tau_3.$$

  - For $x > \frac{\nu_1-\nu_2}{\mu_2}$, the maximum is obtained for $x_1 = \frac{\nu_2-\nu_1+x\mu_2}{\mu_1+\mu_2}$ and we get

$$t_x \leq \frac{1}{\mu_1+\mu_2}\left((T_1+T_2)\mu_1\mu_2 + \frac{\mu_2}{R_1} + \frac{\mu_1}{R_2}\right)x + \frac{1}{\mu_1+\mu_2}\left((\nu_1-\nu_2)\frac{R_1-R_2}{R_1R_2} + (T_1+T_2)(\mu_2\nu_1+\mu_1\nu_2)\right) + \tau_3.$$

Therefore we have

$$t_x \leq \min\left[\frac{1+(T_1+T_2)R_2\mu_2}{R_2}x + (T_1+T_2)\nu_2 + \tau_3, \quad \frac{1}{\mu_1+\mu_2}\left((T_1+T_2)\mu_1\mu_2 + \frac{\mu_2}{R_1} + \frac{\mu_1}{R_2}\right)x + \frac{1}{\mu_1+\mu_2}\left((\nu_1-\nu_2)\frac{R_1-R_2}{R_1R_2} + (T_1+T_2)(\mu_2\nu_1+\mu_1\nu_2)\right) + \tau_3.\right]$$

Hence

$$\tilde{\beta}(t) = \max\left[\frac{R_2}{1+(T_1+T_2)R_2\mu_2}(t-((T_1+T_2)\nu_2+\tau_3))^+, \tilde{R}(t-\tilde{T})^+\right].$$

where

$$\tilde{R} = \frac{(\mu_1+\mu_2)R_1R_2}{R_2\mu_2 + R_1\mu_1 + R_1R_2(T_1+T_2)\mu_1\mu_2},$$

$$\tilde{T} = \frac{(\nu_1-\nu_2)(R_1-R_2) + (T_1+T_2)(\mu_1\nu_2+\mu_2\nu_1)R_1R_2}{(\mu_1+\mu_2)R_1R_2} + \tau_3.$$

- The last case is the symmetric of the third case. Asymptotically, the service is the same for the two last cases. There is an alternation in the service of the flows.

6 Conclusion

In this paper, we show that the packet nature of the traffic can be taken into account in Network calculus by introducing packets curves that constrain the packet flow in the same way as arrival curves constrain the data flow. This generalizes the crude approach that only uses the maximal and minimal sizes of packets and allows one to get more precise bounds on the performance of several scheduling policies such as round robin or servers with set up times between flows. In the future, we plan to use this approach to provide performance guarantees for embedded systems where packets play a critical role (for example when wormhole routing is used).

References

[1] C. S. Chang, Performance Guarantees in Communication Networks, TNCS, 2000.

[2] J.-Y. Le Boudec, P. Thiran, Network Calculus: A Theory of Deterministic Queuing Systems for the Internet, Vol. LNCS 2050, Springer-Verlag, 2001, revised version 4, May 10, 2004.
URL `http://ica1www.epfl.ch/PS_files/netCalBookv4.pdf`

[3] A. Kumar, D. Manjunath, J. Kuri, Communication Networking: An Analytical Approach, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2004.

[4] F. Baccelli, G. Cohen, G.-J. Olsder, J.-P. Quadrat, Synchronization and Linearity, Wiley, 1992.

[5] S. Martin, P. Minet, L. George, End-to-end response time with fixed priority scheduling: trajectory approach versus holistic approach: Research articles, Int. J. Commun. Syst. 18 (2005) 37–56.

[6] M. Hendriks, M. Verhoef, Timed automata based analysis of embedded system architectures, Parallel and Distributed Processing Symposium, International 0 (2006) 165.

[7] V. Firoiu, J.-Y. Le Boudec, D. Towsley, Z.-L. Zhang, Theories and Models for Internet Quality of Service, Proceedings of the IEEE 90 (9) (2002) 1565–1591.

[8] J. Schmitt, U. Roedig, Sensor Network Calculus - A Framework for Worst Case Analysis, in: Proceedings of the International Conference on Distributed Computing in Sensor Systems (DCOSS05), Marina del Rey, USA, IEEE Computer Society Press, 2005.

[9] J. B. Schmitt, F. A. Zdarsky, M. Fidler, Delay bounds under arbitrary multiplexing: When network calculus leaves you in the lurch ..., in: Proc. of INFOCOM'2008, 2008.

[10] L. Lenzini, E. Mingozzi, G. Stea, A methodology for computing end-to-end delay bounds in fifo-multiplexing tandems, Performance Evaluation 65 (11-12) (2008) 922–943.

[11] A. Bouillard, L. Jouhet, E. Thierry, Tight Performance Bounds in the Worst Case Analysis of Feed Forward Networks, in: Proc. of INFOCOM'10, IEEE, 2010.

[12] A. Bouillard, L. Jouhet, E. Thierry, Service curves in Network Calculus: dos and don'ts, Tech. Rep. RR-7094, INRIA (2009).

[13] A. Bouillard, L. Jouhet, É. Thierry, Comparison of different classes of service curves in network calculus, in: Proc. of WODES, 2010, pp. 216–321.

[14] R. Agrawal, R. Rajan, A general framework for analyzing schedulers and regulators in integrated services network, In Proc. 34th Annu. Allerton Commun., Contr., and Computing Conf. (1996) 239–248.

[15] R. L. Cruz, Lecture notes on quality of service guaranties.

Centre de recherche INRIA Paris – Rocquencourt
Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)

Centre de recherche INRIA Bordeaux – Sud Ouest : Domaine Universitaire - 351, cours de la Libération - 33405 Talence Cedex
Centre de recherche INRIA Grenoble – Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier
Centre de recherche INRIA Lille – Nord Europe : Parc Scientifique de la Haute Borne - 40, avenue Halley - 59650 Villeneuve d'Ascq
Centre de recherche INRIA Nancy – Grand Est : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex
Centre de recherche INRIA Rennes – Bretagne Atlantique : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex
Centre de recherche INRIA Saclay – Île-de-France : Parc Orsay Université - ZAC des Vignes : 4, rue Jacques Monod - 91893 Orsay Cedex
Centre de recherche INRIA Sophia Antipolis – Méditerranée : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399