# NETWORK CALCULUS
# Parts II and III

A Theory of Deterministic Queuing Systems
for the Internet

JEAN-YVES LE BOUDEC
PATRICK THIRAN

Online Version of the Book Springer Verlag -
LNCS 2050
Version January 16, 2002

# Part II

# Mathematical Background

# Chapter 3

# Basic Min-plus and Max-plus Calculus

In this chapter we introduce the basic results from Min-plus that are needed for the next chapters. Max-plus algebra is dual to Min-plus algebra, with similar concepts and results when minimum is replaced by maximum, and infimum by supremum. As basic results of network calculus use more min-plus algebra than max-plus algebra, we present here in detail the fundamentals of min-plus calculus. We briefly discuss the care that should be used when max and min operations are mixed at the end of the chapter. A detailed treatment of Min- and Max-plus algebra is provided in [26], here we focus on the basic results that are needed for the remaining of the book. Many of the results below can also be found in [11] for the discrete-time setting.

## 3.1 Min-plus Calculus

In conventional algebra, the two most common operations on elements of $\mathbb{Z}$ or $\mathbb{R}$ are their addition and their multiplication. In fact, the set of integers or reals endowed with these two operations verify a number of well known axioms that define algebraic structures: $(\mathbb{Z}, +, \times)$ is a commutative ring, whereas $(\mathbb{R}, +, \times)$ is a field. Here we consider another algebra, where the operations are changed as follows: addition becomes computation of the minimum, multiplication becomes addition. We will see that this defines another algebraic structure, but let us first recall the notion of minimum and infimum.

### 3.1.1 Infimum and Minimum

Let $\mathcal{S}$ be a nonempty subset of $\mathbb{R}$. $\mathcal{S}$ is bounded from below if there is a number $M$ such that $s \geq M$ for all $s \in \mathcal{S}$. The completeness axiom states that every nonempty subset $\mathcal{S}$ of $\mathbb{R}$ that is bounded from below has a greatest lower bound. We will call it *infimum* of $\mathcal{S}$, and denote it by $\inf \mathcal{S}$. For example the closed and open intervals

$[a, b]$ and $(a, b)$ have the same infimum, which is $a$. Now, if $\mathcal{S}$ contains an element that is smaller than all its other elements, this element is called *minimum* of $\mathcal{S}$, and is denoted by $\min \mathcal{S}$. Note that the minimum of a set does not always exist. For example, $(a, b)$ has no minimum since $a \notin (a, b)$. On the other hand, if the minimum of a set $\mathcal{S}$ exists, it is identical to its infimum. For example, $\min[a, b] = \inf[a, b] = a$. One easily shows that every finite nonempty subset of $\mathbb{R}$ has a minimum. Finally, let us mention that we will often use the notation $\wedge$ to denote infimum (or, when it exists, the minimum). For example, $a \wedge b = \min\{a, b\}$. If $\mathcal{S}$ is empty, we adopt the convention that $\inf \mathcal{S} = +\infty$.

If $f$ is a function from $\mathcal{S}$ to $\mathbb{R}$, we denote by $f(\mathcal{S})$ its range:

$$f(\mathcal{S}) = \{t \text{ such that } t = f(s) \text{ for some } s \in \mathcal{S}\}.$$

We will denote the infimum of this set by the two equivalent notations

$$\inf f(\mathcal{S}) = \inf_{s \in \mathcal{S}} \{f(s)\}.$$

We will also often use the following property.

**Theorem 3.1.1 ("Fubini" formula for infimum).** *Let $\mathcal{S}$ be a nonempty subset of $\mathbb{R}$, and $f$ be a function from $\mathcal{S}$ to $\mathbb{R}$. Let $\{\mathcal{S}_n\}_{n \in \mathbb{N}}$ be a collection of subsets of $\mathcal{S}$, whose union is $\mathcal{S}$. Then*

$$\inf_{s \in \mathcal{S}} \{f(s)\} = \inf_{n \in \mathbb{N}} \left\{ \inf_{s \in \mathcal{S}_n} \{f(s_n)\} \right\}.$$

**Proof:** By definition of an infimum, for any sets $\mathcal{S}_n$,

$$\inf \left\{ \bigcup_n \mathcal{S}_n \right\} = \inf_n \{\inf \mathcal{S}_n\}.$$

On the other hands, since $\cup_n \mathcal{S}_n = \mathcal{S}$,

$$f\left( \bigcup_{n \in \mathbb{N}} \mathcal{S}_n \right) = \bigcup_{n \in \mathbb{N}} f(\mathcal{S}_n)$$

so that

$$\begin{aligned}
\inf_{s \in \mathcal{S}} \{f(s)\} &= \inf f(\mathcal{S}) = \inf f\left( \bigcup_{n \in \mathbb{N}} \mathcal{S}_n \right) \\
&= \inf \left\{ \bigcup_{n \in \mathbb{N}} f(\mathcal{S}_n) \right\} = \inf_{n \in \mathbb{N}} \{\inf f(\mathcal{S}_n)\} \\
&= \inf_{n \in \mathbb{N}} \left\{ \inf_{s \in \mathcal{S}_n} \{f(s)\} \right\}.
\end{aligned}$$

□

### 3.1.2 Dioid $(\mathbb{R} \cup \{+\infty\}, \wedge, +)$

In traditional algebra, one is used to working with the algebraic structure $(\mathbb{R}, +, \times)$, that is, with the set of reals endowed with the two usual operations of addition and multiplication. These two operations possess a number of properties (associativity, commutativity, distributivity, etc) that make $(\mathbb{R}, +, \times)$ a commutative field. As mentioned above, in min-plus algebra, the operation of 'addition' becomes computation of the infimum (or of the minimum if it exists), whereas the one of 'multiplication' becomes the classical operation of addition. We will also include $+\infty$ in the set of elements on which min-operations are carried out, so that the structure of interest is now $(\mathbb{R} \cup \{+\infty\}, \wedge, +)$. Most axioms (but not all, as we will see later) defining a field still apply to this structure. For example, distribution of addition with respect to multiplication in conventional ('Plus-times') algebra

$$(3 + 4) \times 5 = (3 \times 5) + (4 \times 5) = 15 + 20 = 35$$

translates in min-plus algebra as

$$(3 \wedge 4) + 5 = (3 + 5) \wedge (4 + 5) = 8 \wedge 9 = 8.$$

In fact, one easily verifies that $\wedge$ and $+$ satisfy the following properties:

- **(Closure of $\wedge$)** For all $a, b \in \mathbb{R} \cup \{+\infty\}$, $a \wedge b \in \mathbb{R} \cup \{+\infty\}$.
- **(Associativity of $\wedge$)** For all $a, b, c \in \mathbb{R} \cup \{+\infty\}$, $(a \wedge b) \wedge c = a \wedge (b \wedge c)$.
- **(Existence of a zero element for $\wedge$)** There is some $\mathrm{e} = +\infty \in \mathbb{R} \cup \{+\infty\}$ such that for all $a \in \mathbb{R} \cup \{+\infty\}$, $a \wedge \mathrm{e} = a$.
- **(Idempotency of $\wedge$)** For all $a \in \mathbb{R} \cup \{+\infty\}$, $a \wedge a = a$.
- **(Commutativity of $\wedge$)** For all $a, b \in \mathbb{R} \cup \{+\infty\}$, $a \wedge b = b \wedge a$.
- **(Closure of $+$)** For all $a, b \in \mathbb{R} \cup \{+\infty\}$, $a + b \in \mathbb{R} \cup \{+\infty\}$.
- **(Associativity of $+$)** For all $a, b, c \in \mathbb{R} \cup \{+\infty\}$, $(a + b) + c = a + (b + c)$.
- **(The zero element for $\wedge$ is absorbing for $+$)** For all $a \in \mathbb{R} \cup \{+\infty\}$, $a + \mathrm{e} = \mathrm{e} = \mathrm{e} + a$.
- **(Existence of a neutral element for $+$)** There is some $u = 0 \in \mathbb{R} \cup \{+\infty\}$ such that for all $a \in \mathbb{R} \cup \{+\infty\}$, $a + u = a = u + a$.
- **(Distributivity of $+$ with respect to $\wedge$)** For all $a, b, c \in \mathbb{R} \cup \{+\infty\}$, $(a \wedge b) + c = (a + c) \wedge (b + c) = c + (a \wedge b)$.

A set endowed with operations satisfying all the above axioms is called a *dioid*. Moreover as $+$ is also commutative (for all $a, b \in \mathbb{R} \cup \{+\infty\}$, $a + b = b + a$), the structure $(\mathbb{R} \cup \{+\infty\}, \wedge, +)$ is a commutative dioid. All the axioms defining a dioid are therefore the same axioms as the ones defining a ring, except one: the axiom of idempotency of the 'addition', which in dioids replaces the axiom of cancellation of 'addition' in rings (i.e. the existence of an element $(-a)$ that 'added' to $a$ gives the zero element). We will encounter other dioids later on in this chapter.

### 3.1.3 A Catalog of Wide-sense Increasing Functions

A function $f$ is wide-sense increasing if and only if $f(s) \leq f(t)$ for all $s \leq t$. We will denote by $\mathcal{G}$ the set of non-negative wide-sense increasing sequences or functions and by $\mathcal{F}$ denote the set of wide-sense increasing sequences or functions such that $f(t) = 0$ for $t < 0$. Parameter $t$ can be continuous or discrete: in the latter case, $f = \{f(t), t \in \mathbb{Z}\}$ is called a sequence rather than a function. In the former case, we take the convention that the function $f = \{f(t), t \in \mathbb{R}\}$ is left-continuous. The range of functions or sequences of $\mathcal{F}$ and $\mathcal{G}$ is $\mathbb{R}^+ = [0, +\infty]$.

Notation $f + g$ (respectively $f \wedge g$) denotes the point-wise sum (resp. minimum) of functions $f$ and $g$:

$$\begin{aligned} (f+g)(t) &= f(t) + g(t) \\ (f \wedge g)(t) &= f(t) \wedge g(t) \end{aligned}$$

Notation $f \leq (=, \geq) g$ means that $f(t) \leq (=, \geq) g(t)$ for all $t$.

Some examples of functions belonging to $\mathcal{F}$ and of particular interest are the following ones. Notation $[x]^+$ denotes $\max\{x, 0\}$, $\lceil x \rceil$ denotes the smallest integer larger than or equal to $x$.

**Definition 3.1.1 (Peak rate functions $\lambda_R$).**

$$\lambda_R(t) = \begin{cases} Rt & \textit{if } t > 0 \\ 0 & \textit{otherwise} \end{cases}$$

*for some $R \geq 0$ (the 'rate').*

**Definition 3.1.2 (Burst delay functions $\delta_T$).**

$$\delta_T(t) = \begin{cases} +\infty & \textit{if } t > T \\ 0 & \textit{otherwise} \end{cases}$$

*for some $T \geq 0$ (the 'delay').*

**Definition 3.1.3 (Rate-latency functions $\beta_{R,T}$).**

$$\beta_{R,T}(t) = R[t-T]^+ = \begin{cases} R(t-T) & \textit{if } t > T \\ 0 & \textit{otherwise} \end{cases}$$

*for some $R \geq 0$ (the 'rate') and $T \geq 0$ (the 'delay').*

**Definition 3.1.4 (Affine functions $\gamma_{r,b}$).**

$$\gamma_{r,b}(t) = \begin{cases} rt + b & \textit{if } t > 0 \\ 0 & \textit{otherwise} \end{cases}$$

*for some $r \geq 0$ (the 'rate') and $b \geq 0$ (the 'burst').*

**Definition 3.1.5 (Step Function $v_T$).**

$$v_T(t) = 1_{\{t>T\}} = \begin{cases} 1 & \textit{if } t > T \\ 0 & \textit{otherwise} \end{cases}$$

*for some $T > 0$.*

**Definition 3.1.6 (Staircase Functions $u_{T,\tau}$).**

$$u_{T,\tau}(t) = \begin{cases} \lceil \frac{t+\tau}{T} \rceil & \textit{if } t > 0 \\ 0 & \textit{otherwise} \end{cases}$$

*for some $T > 0$ (the 'interval') and $0 \leq \tau \leq T$ (the 'tolerance').*

These functions are also represented in Figure 3.1. By combining these basic functions, one obtains more general piecewise linear functions belonging to $\mathcal{F}$. For example, the two functions represented in Figure 3.2 are written using $\wedge$ and $+$ from affine functions and rate-latency functions as follows, with $r_1 > r_2 > \ldots > r_I$ and $b_1 < b_2 < \ldots < b_I$

$$f_1 = \gamma_{r_1,b_1} \wedge \gamma_{r_2,b_2} \wedge \ldots \gamma_{r_I,b_I} = \min_{1 \leq i \leq I} \{\gamma_{r_i,b_i}\} \tag{3.1}$$

$$\begin{aligned} f_2 &= \lambda_R \wedge \{\beta_{R,2T} + RT\} \wedge \{\beta_{R,4T} + 2RT\} \wedge \ldots \\ &= \inf_{i \geq 0} \{\beta_{R,2iT} + iRT\}. \end{aligned} \tag{3.2}$$

We will encounter other functions later in the book, and obtain other representations with the min-plus convolution operator.

### 3.1.4 Pseudo-inverse of Wide-sense Increasing Functions

It is well known that any strictly increasing function is left-invertible. That is, if for any $t_1 < t_2$, $f(t_1) < f(t_2)$, then there is a function $f^{-1}$ such that $f^{-1}(f(t)) = t$ for all $t$. Here we consider slightly more general functions, namely, wide-sense increasing functions, and we will see that a pseudo-inverse function can defined as follows.

**Definition 3.1.7 (Pseudo-inverse).** *Let $f$ be a function or a sequence of $\mathcal{F}$. The pseudo-inverse of $f$ is the function*

$$f^{-1}(x) = \inf\{t \text{ such that } f(t) \geq x\}. \tag{3.3}$$

For example, one can easily compute that the pseudo-inverses of the four functions of Definitions 3.1.1 to 3.1.4 are

$$\begin{aligned} \lambda_R^{-1} &= \lambda_{1/R} \\ \delta_T^{-1} &= \delta_0 \wedge T \\ \beta_{R,T}^{-1} &= \gamma_{1/R,T} \\ \gamma_{r,b}^{-1} &= \beta_{1/r,b}. \end{aligned}$$

The pseudo-inverse enjoys the following properties:

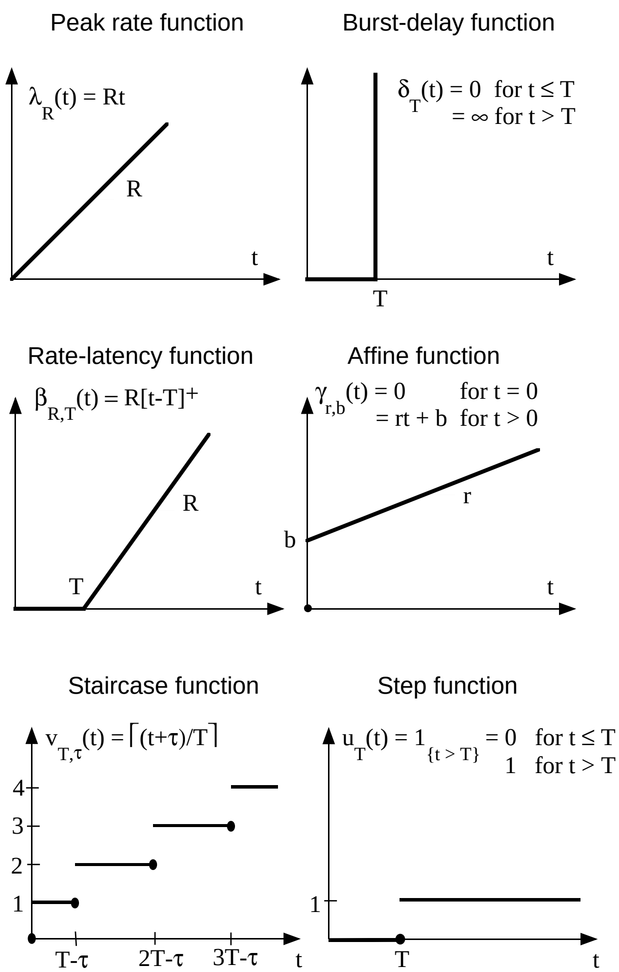



Figure 3.1: A catalog of functions of $\mathcal{F}$: Peak rate function (top left), burst-delay function (top right), rate-latency function (center left), affine function (center right), staircase function (bottom left) and step function (bottom right).

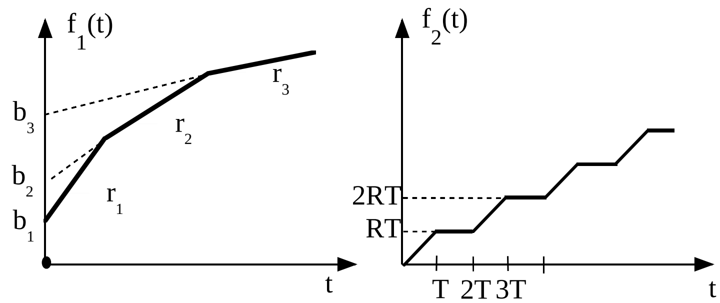



Figure 3.2: Two piecewise linear functions of $\mathcal{F}$ as defined by (3.1) (left) and (3.2) (right).

**Theorem 3.1.2 (Properties of pseudo-inverse functions).** *Let $f \in \mathcal{F}$, $x, t \geq 0$.*

- ***(Closure)*** $f^{-1} \in \mathcal{F}$ *and* $f^{-1}(0) = 0$.
- ***(Pseudo-inversion)*** *We have that*

$$f(t) \geq x \quad \Rightarrow \quad f^{-1}(x) \leq t \tag{3.4}$$
$$f^{-1}(x) < t \quad \Rightarrow \quad f(t) \geq x \tag{3.5}$$

- ***(Equivalent definition)***

$$f^{-1}(x) = \sup \{t \text{ such that } f(t) < x\} . \tag{3.6}$$

**Proof:** Define subset $\mathcal{S}_x = \{t \text{ such that } f(t) \geq x\} \subseteq \mathbb{R}^+$. Then (3.3) becomes $f^{-1}(x) = \inf \mathcal{S}_x$. (Closure) Clearly, from (3.3), $f^{-1}(x) = 0$ for $x \leq 0$ (and in particular $f^{-1}(0) = 0$). Now, let $0 \leq x_1 < x_2$. Then $\mathcal{S}_{x_1} \supseteq \mathcal{S}_{x_2}$, which implies that $\inf \mathcal{S}_{x_1} \leq \inf \mathcal{S}_{x_2}$ and hence that $f^{-1}(x_1) \leq f^{-1}(x_2)$. Therefore $f^{-1}$ is wide-sense increasing. (Pseudo-inversion) Suppose first that $f(t) \geq x$. Then $t \in \mathcal{S}_x$, and so is larger than the infimum of $\mathcal{S}_x$, which is $f^{-1}(x)$: this proves (3.4). Suppose next that $f^{-1}(x) < t$. Then $t > \inf \mathcal{S}_x$, which implies that $t \in \mathcal{S}_x$, by definition of an infimum. This in turn yields that $f(t) \geq x$ and proves (3.5). (Equivalent definition) Define subset $\tilde{\mathcal{S}}_x = \{t \text{ such that } f(t) < x\} \subseteq \mathbb{R}^+$. Pick $t \in \mathcal{S}_x$ and $\tilde{t} \in \tilde{\mathcal{S}}_x$. Then $f(\tilde{t}) < f(t)$, and since $f$ is wide-sense increasing, it implies that $\tilde{t} \leq t$. This is true for any $t \in \mathcal{S}_x$ and $\tilde{t} \in \tilde{\mathcal{S}}_x$, hence $\sup \tilde{\mathcal{S}}_x \leq \inf \mathcal{S}_x$. As $\tilde{\mathcal{S}}_x \cup \mathcal{S}_x = \mathbb{R}^+$, we cannot have $\sup \tilde{\mathcal{S}}_x < \inf \mathcal{S}_x$. Therefore

$$\sup \tilde{\mathcal{S}}_x = \inf \mathcal{S}_x = f^{-1}(x).$$

□

### 3.1.5 Concave, Convex and Star-shaped Functions

As an important class of functions in min-plus calculus are the convex and concave functions, it is useful to recall some of their properties.

**Definition 3.1.8 (Convexity in $\mathbb{R}^n$).** *Let $u$ be any real such that $0 \leq u \leq 1$.*

- *Subset $\mathcal{S} \subseteq \mathbb{R}^n$ is convex if and only if $ux + (1-u)y \in \mathcal{S}$ for all $x, y \in \mathcal{S}$.*
- *Function $f$ from a subset $\mathcal{D} \subseteq \mathbb{R}^n$ to $\mathbb{R}$ is convex if and only if $f(ux + (1-u)y) \leq uf(x) + (1-u)f(y)$ for all $x, y \in D$.*
- *Function $f$ from a subset $\mathcal{D} \subseteq \mathbb{R}^n$ to $\mathbb{R}$ is concave if and only if $-f$ is convex.*

For example, the rate-latency function (Fig 3.1, center left) is convex, the piecewise linear function $f_1$ given by (3.1) is concave and the piecewise linear function $f_2$ given by (3.2) is neither convex nor concave.

There are a number of properties that convex sets and functions enjoy [72]. Here are a few that will be used in this chapter, and that are a direct consequence of Definition 3.1.8.

- The convex subsets of $\mathbb{R}$ are the intervals.
- If $\mathcal{S}_1$ and $\mathcal{S}_2$ are two convex subsets of $\mathbb{R}^n$, their sum

  $$\mathcal{S} = \mathcal{S}_1 + \mathcal{S}_2 = \{s \in \mathbb{R}^n \mid s = s_1 + s_2 \text{ for some } s_1 \in \mathcal{S}_1 \text{ and } s_2 \in \mathcal{S}_2\}$$

  is also convex.
- Function $f$ from an interval $[a, b]$ to $\mathbb{R}$ is convex (resp. concave) if and only if $f(ux + (1-u)y) \leq$ (resp. $\geq$) $uf(x) + (1-u)f(y)$ for all $x, y \in [a, b]$ and all $u \in [0.1]$.
- The pointwise maximum (resp. minimum) of any number of convex (resp. concave) functions is a convex (resp. concave) function.
- If $\mathcal{S}$ is a convex subset of $\mathbb{R}^{n+1}$, $n \geq 1$, the function from $\mathbb{R}^n$ to $\mathbb{R}$ defined by

  $$f(x) = \inf\{\mu \in \mathbb{R} \text{ such that } (x, \mu) \in \mathcal{S}\}$$

  is convex.
- If $f$ is a convex function from $\mathbb{R}^n$ to $\mathbb{R}$, the set $\mathcal{S}$ defined by

  $$\mathcal{S} = \{(x, \mu) \in \mathbb{R}^{n+1} \text{ such that } f(x) \leq \mu\}$$

  is convex. This set is called the epigraph of $f$. It implies in the particular case where $n = 1$ that the line segment between $\{a, f(a)\}$ and $\{b, f(b)\}$ lies above the graph of the curve $y = f(x)$.

The proof of these properties is given in [72] and can be easily deduced from Definition 3.1.8, or even from a simple drawing. Chang [11] introduced *star-shaped* functions, which are defined as follows.

**Definition 3.1.9 (Star-shaped function).** *Function $f \in \mathcal{F}$ is star-shaped if and only if $f(t)/t$ is wide-sense decreasing for all $t > 0$.*

Star-shaped enjoy the following property:

**Theorem 3.1.3 (Minimum of star-shaped functions).** *Let $f, g$ be two star-shaped functions. Then $h = f \wedge g$ is also star-shaped.*

**Proof:** Consider some $t \geq 0$. If $h(t) = f(t)$, then for all $s > t$, $h(t)/t = f(t)/t \geq f(s)/s \geq h(s)/s$. The same argument holds of course if $h(t) = g(t)$. Therefore $h(t)/t \geq h(s)/s$ for all $s > t$, which shows that $h$ is star-shaped. □

We will see other properties of star-shaped functions in the next sections. Let us conclude this section with an important class of star-shaped functions.

**Theorem 3.1.4.** *Concave functions are star-shaped.*

**Proof:** Let $f$ be a concave function. Then for any $u \in [0, 1]$ and $x, y \geq 0$, $f(ux + (1-u)y) \geq uf(x) + (1-u)f(y)$. Take $x = t$, $y = 0$ and $u = s/t$, with $0 < s \leq t$. Then the previous inequality becomes $f(s) \geq (s/t)f(t)$, which shows that $f(t)/t$ is a decreasing function of $t$. □

On the other hand, a star-shaped function is not necessarily a concave function. We will see one such example in Section 3.1.7.

### 3.1.6 Min-plus Convolution

Let $f(t)$ be a real-valued function, which is zero for $t \leq 0$. If $t \in \mathbb{R}$, the integral of this function in the conventional algebra $(\mathbb{R}, +, \times)$ is

$$\int_0^t f(s)ds$$

which becomes, for a sequence $f(t)$ where $t \in \mathbb{Z}$,

$$\sum_{s=0}^{t} f(s).$$

In the min-plus algebra $(\mathbb{R} \cup \{+\infty\}, \wedge, +)$, where the 'addition' is $\wedge$ and the 'multiplication' is $+$, an 'integral' of the function $f$ becomes therefore

$$\inf_{s \in \mathbb{R} \text{ such that } 0 \leq s \leq t} \{f(s)\},$$

which becomes, for a sequence $f(t)$ where $t \in \mathbb{Z}$,

$$\min_{s \in \mathbb{Z} \text{ such that } 0 \le s \le t} \{f(s)\}.$$

We will often adopt a shorter notation for the two previous expressions, which is

$$\inf_{0 \le s \le t} \{f(s)\},$$

with $s \in \mathbb{Z}$ or $s \in \mathbb{R}$ depending on the domain of $f$.

A key operation in conventional linear system theory is the convolution between two functions, which is defined as

$$(f \otimes g)(t) = \int_{-\infty}^{+\infty} f(t-s)g(s)ds$$

and becomes, when $f(t)$ and $g(t)$ are two functions that are zero for $t < 0$,

$$(f \otimes g)(t) = \int_{0}^{t} f(t-s)g(s)ds.$$

In min-plus calculus, the operation of convolution is the natural extension of the previous definition:

**Definition 3.1.10 (Min-plus convolution).** *Let $f$ and $g$ be two functions or sequences of $\mathcal{F}$. The min-plus convolution of $f$ and $g$ is the function*

$$(f \otimes g)(t) = \inf_{0 \le s \le t} \{f(t-s) + g(s)\}. \tag{3.7}$$

*(If $t < 0$, $(f \otimes g)(t) = 0$).*

**Example.** Consider the two functions $\gamma_{r,b}$ and $\beta_{R,T}$, with $0 < r < R$, and let us compute their min-plus convolution. Let us first compute it for $0 \le t \le T$.

$$\begin{aligned}
(\gamma_{r,b} \otimes \beta_{R,T})(t) &= \inf_{0 \le s \le t} \left\{ \gamma_{r,b}(t-s) + R[s-T]^+ \right\} \\
&= \inf_{0 \le s \le t} \{\gamma_{r,b}(t-s) + 0\} = \gamma_{r,b}(0) + 0 = 0 + 0 = 0
\end{aligned}$$

Now, if $t > T$, one has

$$\begin{aligned}
&(\gamma_{r,b} \otimes \beta_{R,T})(t) \\
&= \inf_{0 \le s \le t} \left\{ \gamma_{r,b}(t-s) + R[s-T]^+ \right\} \\
&= \inf_{0 \le s \le T} \left\{ \gamma_{r,b}(t-s) + R[s-T]^+ \right\} \wedge \inf_{T \le s < t} \left\{ \gamma_{r,b}(t-s) + R[s-T]^+ \right\} \\
&\qquad \wedge \inf_{s=t} \left\{ \gamma_{r,b}(t-s) + R[s-T]^+ \right\} \\
&= \inf_{0 \le s \le T} \{b + r(t-s) + 0\} \wedge \inf_{T < s < t} \{b + r(t-s) + R(s-T)\} \\
&\qquad \wedge \{0 + R(t-T)\} \\
&= \{b + r(t-T)\} \wedge \left\{ b + rt - RT + \inf_{T < s < t} \{(R-r)s\} \right\} \wedge \{R(t-T)\} \\
&= \{b + r(t-T)\} \wedge \{b + r(t-T)\} \wedge \{R(t-T)\} \\
&= \{b + r(t-T)\} \wedge \{R(t-T)\}.
\end{aligned}$$

The result is shown in Figure 3.3. Let us now derive some useful properties for the

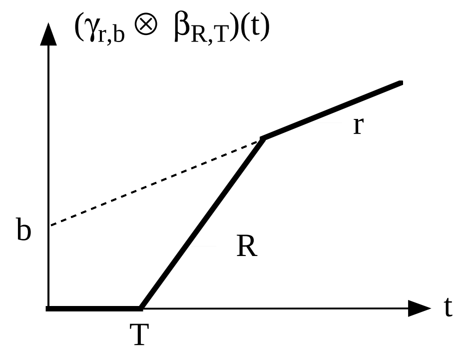



Figure 3.3: Function $\gamma_{r,b} \otimes \beta_{R,T}$ when $0 < r < R$.

computation of min-plus convolution.

**Theorem 3.1.5 (General properties of $\otimes$).** *Let $f, g, h \in \mathcal{F}$.*

- ***Rule 1 (Closure of $\otimes$)*** *$(f \otimes g) \in \mathcal{F}$.*
- ***Rule 2 (Associativity of $\otimes$)*** *$(f \otimes g) \otimes h = f \otimes (g \otimes h)$.*
- ***Rule 3 (The zero element for $\wedge$ is absorbing for $\otimes$)*** *The zero element for $\wedge$ belonging to $\mathcal{F}$ is the function $\varepsilon$, defined as $\varepsilon(t) = +\infty$ for all $t \geq 0$ and $\varepsilon(t) = 0$ for all $t < 0$. One has $f \otimes \varepsilon = \varepsilon$.*
- ***Rule 4 (Existence of a neutral element for $\otimes$)*** *The neutral element is $\delta_0$, as $f \otimes \delta_0 = f$.*
- ***Rule 5 (Commutativity of $\otimes$)*** *$f \otimes g = g \otimes f$.*
- ***Rule 6 (Distributivity of $\otimes$ with respect to $\wedge$)*** *$(f \wedge g) \otimes h = (f \otimes h) \wedge (g \otimes h)$.*
- ***Rule 7 (Addition of a constant)*** *For any $K \in \mathbb{R}^+$, $(f + K) \otimes g = (f \otimes g) + K$.*

The proof of these rules is easy. We prove the two first rules, the proof of the five others are left to the reader.

**Proof:** (Rule 1) Since $f$ is wide-sense increasing,

$$f(t_1 - s) + g(s) \leq f(t_2 - s) + g(s)$$

for all $0 \leq t_1 < t_2$ and all $s \in \mathbb{R}$. Therefore

$$\inf_{s\in\mathbb{R}} \{f(t_1 - s) + g(s)\} \leq \inf_{s\in\mathbb{R}} \{f(t_2 - s) + g(s)\}$$

and as $f(t) = g(t) = 0$ when $t < 0$, this inequality is equivalent to

$$\inf_{0\leq s\leq t_1} \{f(t_1 - s) + g(s)\} \leq \inf_{0\leq s\leq t_2} \{f(t_2 - s) + g(s)\},$$

which shows that $(f \otimes g)(t_1) \leq (f \otimes g)(t_2)$ for all $0 \leq t_1 < t_2$. (Rule 2) One has

$$\begin{aligned}
((f \otimes g) \otimes h)(t) &= \inf_{0\leq s\leq t} \left\{ \inf_{0\leq u\leq t-s} \{f(t-s-u) + g(u)\} + h(s) \right\} \\
&= \inf_{0\leq s\leq t} \left\{ \inf_{s\leq u'\leq t} \{f(t-u') + g(u'-s) + h(s)\} \right\} \\
&= \inf_{0\leq u'\leq t} \left\{ \inf_{0\leq s\leq u'} \{f(t-u') + g(u'-s) + h(s)\} \right\} \\
&= \inf_{0\leq u'\leq t} \left\{ f(t-u') + \inf_{0\leq s\leq u'} \{g(u'-s) + h(s)\} \right\} \\
&= \inf_{0\leq u'\leq t} \{f(t-u') + (g \otimes h)(u')\} \\
&= (f \otimes (g \otimes h))(t).
\end{aligned}$$

□

Rules 1 to 6 establish a structure of a commutative dioid for $(\mathcal{F}, \wedge, \otimes)$, whereas Rules 6 and 7 show that $\otimes$ is a linear operation on $(\mathbb{R}^+, \wedge, +)$. Now let us also complete these results by two additional rules that are helpful in the case of concave or convex functions.

**Theorem 3.1.6 (Properties of $\otimes$ for concave/convex functions).** *Let $f, g \in \mathcal{F}$.*

- ***Rule 8 (Functions passing through the origin)*** *If $f(0) = g(0) = 0$ then $f \otimes g \leq f \wedge g$. Moreover, if $f$ and $g$ are star-shaped, then $f \otimes g = f \wedge g$.*
- ***Rule 9 (Convex functions)*** *If $f$ and $g$ are convex then $f \otimes g$ is convex. In particular if $f, g$ are convex and piecewise linear, $f \otimes g$ is obtained by putting end-to-end the different linear pieces of $f$ and $g$, sorted by increasing slopes.*

Since concave functions are star-shaped, Rule 8 also implies that if $f, g$ are concave with $f(0) = g(0) = 0$, then $f \otimes g = f \wedge g$.

**Proof:** (Rule 8) As $f(0) = g(0) = 0$,

$$(f \otimes g)(t) = g(t) \wedge \inf_{0<s<t} \{f(t-s) + g(s)\} \wedge f(t) \leq f(t) \wedge g(t). \tag{3.8}$$

Suppose now that, in addition, $f$ and $g$ are star-shaped. Then for any $t > 0$ and $0 \leq s \leq t$ $f(t-s) \geq (1 - s/t)f(t)$ and $g(s) \geq (s/t)g(t)$, so that

$$f(t-s) + g(s) \geq f(t) + (s/t)(g(t) - f(t)).$$

Now, as $0 \le s/t \le 1$, $f(t) + (s/t)(g(t) - f(t)) \ge f(t) \wedge g(t)$ so that

$$f(t-s) + g(s) \ge f(t) \wedge g(t)$$

for all $0 \le s \le t$. Combining this inequality with (3.8), we obtain the desired result.
(Rule 9) The proof uses properties of convex sets and functions listed in the previous subsection. The epigraphs of $f$ and $g$ are the sets

$$\begin{aligned}\mathcal{S}_1 &= \{(s_1, \mu_1) \in \mathbb{R}^2 \text{ such that } f(s_1) \le \mu_1\} \\ \mathcal{S}_2 &= \{(s_2, \mu_2) \in \mathbb{R}^2 \text{ such that } g(s_2) \le \mu_2\}\end{aligned}$$

Since $f$ and $g$ are convex, their epigraphs are also convex, and so is their sum $\mathcal{S} = \mathcal{S}_1 + \mathcal{S}_2$, which can be expressed as

$$\mathcal{S} = \{(t, \mu) \in \mathbb{R}^2 | \text{ for some } (s, \xi) \in [0,t] \times [0,\mu], f(t-s) \le \mu - \xi, g(s) \le \xi\}.$$

As $\mathcal{S}$ is convex, function $h(t) = \inf\{\mu \in \mathbb{R} \text{ such that } (t, \mu) \in \mathcal{S}\}$ is also convex. Now $h$ can be recast as

$$\begin{aligned}&h(t)\\ &= \inf\{\mu \in \mathbb{R} \mid \text{for some} (s, \xi) \in [0,t] \times [0,\mu], f(t-s) \le \mu - \xi, g(s) \le \xi\} \\ &= \inf\{\mu \in \mathbb{R} \mid \text{for some } s \in [0,t], f(t-s) + g(s) \le \mu\} \\ &= \inf\{f(t-s) + g(s), s \in [0,t]\} \\ &= (f \otimes g)(t),\end{aligned}$$

which proves that $(f \otimes g)$ is convex.

If $f$ and $g$ are piecewise linear, one can construct the set $\mathcal{S} = \mathcal{S}_1 + \mathcal{S}_2$, which is the epigraph of $f \otimes g$, by putting end-to-end the different linear pieces of $f$ and $g$, sorted by increasing slopes [22].

Indeed, let $h'$ denote the function that results from this operation, and let us show that $h' = f \otimes g$. Suppose that there are a total of $n$ linear pieces from $f$ and $g$, and label them from 1 to $n$ according to their increasing slopes: $0 \le r_1 \le r_2 \le \ldots \le r_n$. Figure 3.4 shows an example for $n = 5$. Let $T_i$ denote the length of the projection of segment $i$ onto the horizontal axis, for $1 \le i \le n$. Then the length of the projection of segment $i$ onto the vertical axis is $r_i T_i$. Denote by $\mathcal{S}'$ the epigraph of $h'$, which is convex, and by $\partial\mathcal{S}'$ its boundary. Pick any point $(t, h'(t))$ on this boundary $\partial\mathcal{S}'$. We will show that it can always be obtained by adding a point $(t-s, f(t-s))$ of the boundary $\partial\mathcal{S}_1$ of $\mathcal{S}_1$ and a point $(s, g(s))$ of the boundary $\partial\mathcal{S}_2$ of $\mathcal{S}_2$. Let $k$ be the linear segment index to which $(t, h'(t))$ belongs, and assume, with no loss of generality, that this segment is a piece of $f$ (that is, $k \subseteq \partial\mathcal{S}_1$). We can express $h'(t)$ as

$$h'(t) = r_k(t - \sum_{i=1}^{k-1} T_i) + \sum_{i=1}^{k-1} r_i T_i. \tag{3.9}$$

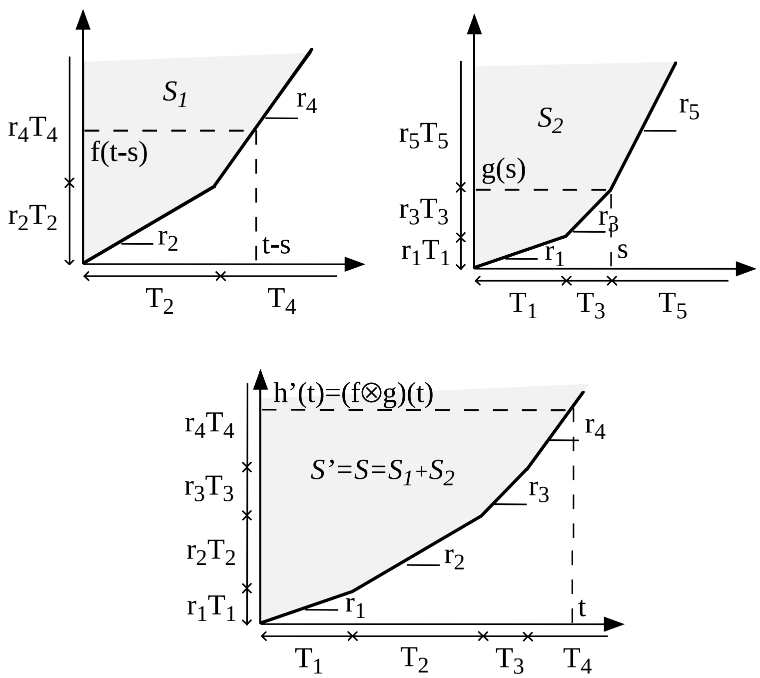



Figure 3.4: Convex, piecewise linear functions $f$ (and its epigraph $\mathcal{S}_1$ (top left)), $g$ (and its epigraph $\mathcal{S}_2$ (top right)), and $f \otimes g$ (and its epigraph $\mathcal{S} = \mathcal{S}_1 + \mathcal{S}_2$ (bottom)).

Now, let $s$ be the sum of the lengths of the horizontal projections of the segments belonging to $g$ and whose index is less than $k$, that is,

$$s = \sum_{i \subseteq \partial \mathcal{S}_2, 1 \le i \le k-1} T_i.$$

Then we can compute that

$$\begin{aligned} t - s &= t - \sum_{i=1}^{k-1} T_i + \sum_{i=1}^{k-1} T_i - \sum_{i \subseteq \partial \mathcal{S}_2, 1 \le i \le k-1} T_i \\ &= t - \sum_{i=1}^{k-1} T_i + \sum_{i \subseteq \partial \mathcal{S}_1, 1 \le i \le k-1} T_i \end{aligned}$$

and that

$$\begin{aligned} f(t-s) &= r_k(t - \sum_{i=1}^{k-1} T_i) + \sum_{i \subseteq \partial \mathcal{S}_1, 1 \le i \le k-1} r_i T_i \\ g(s) &= \sum_{i \subseteq \partial \mathcal{S}_2, 1 \le i \le k-1} r_i T_i. \end{aligned}$$

The addition of the right hand sides of these two equations is equal to $h'(t)$, because of (3.9), and therefore $f(t-s) + g(s) = h'(t)$. This shows that any point of $\partial \mathcal{S}'$ can be broken down into the sum of a point of $\partial \mathcal{S}_1$ and of a point of $\partial \mathcal{S}_2$, and hence that $\partial \mathcal{S}' = \partial \mathcal{S}_1 + \partial \mathcal{S}_2$, which in turn implies that $\mathcal{S}' = \mathcal{S}_1 + \mathcal{S}_2 = \mathcal{S}$. Therefore $h' = f \otimes g$. □

The last rule is easy to prove, and states that $\otimes$ is isotone, namely:

**Theorem 3.1.7 (Isotonicity of $\otimes$).** *Let $f, g, f', g' \in \mathcal{F}$.*

- ***Rule 10 (Isotonicity)*** *If $f \le g$ and $f' \le g'$ then $f \otimes f' \le g \otimes g'$.*

We will use the following theorem:

**Theorem 3.1.8.** *For $f$ and $g$ in $\mathcal{F}$, if in addition $g$ is continuous, then for any $t$ there is some $t_0$ such that*

$$(f \otimes g)(t) = f_l(t_0) + g(t - t_0) \tag{3.10}$$

*where $f_l(t_0) = \sup_{\{s<t_0\}} f(s)$ is the limit to the left of $f$ at $t_0$. If $f$ is left-continuous, then $f_l(t_0) = f(t_0)$.*

**Proof:** Fix $t$. There is a sequence of times $0 \le s_n \le t$ such that

$$\inf_{t_0 \le t} (f(t_0) + g(t - t_0)) = \lim_{n \to \infty} (f(s_n) + g(t - s_n)) \tag{3.11}$$

Since $0 \le s_n \le t$, we can extract a sub-sequence that converges towards some value $t_0$. We take a notation shortcut and write $\lim_{n \to \infty} s_n = t_0$. If $f$ is continuous, the right hand-side in 3.11 is equal to $f_l(t_0) + g(t - t_0)$ which shows the proposition. Otherwise $f$ has a discontinuity at $t_0$. Define $\delta = f(t_0) - f_l(t_0)$. We show that we can again extract a subsequence such that $s_n < t_0$. Indeed, if this would not be true, we would have $s_n \geq t_0$ for all but a finite number of indices $n$. Thus for $n$ large enough we would have

$$f(s_n) \geq f_l(t_0) + \delta$$

and by continuity of $g$:

$$g(t - s_n) \geq g(t - t_0) - \frac{\delta}{2}$$

thus

$$f(s_n) + g(t - s_n) \geq f_l(t_0) + g(t - t_0) + \frac{\delta}{2}$$

Now

$$f_l(t_0) + g(t - t_0) \geq \inf_{s \leq t} (f(s) + g(t - s))$$

thus

$$f(s_n) + g(t - s_n) \geq \inf_{s \leq t} (f(s) + g(t - s)) + \frac{\delta}{2}$$

which contradicts 3.11. Thus we can assume that $s_n \leq t_0$ for $n$ large enough and thus $\lim_{n\to\infty} f(s_n) = f_l(t_0)$. □

Finally, let us mention that it will sometimes be useful to break down a somewhat complex function into the convolution of a number of simpler functions. For example, observe that the rate-latency function $\beta_{R,T}$ can be expressed as

$$\beta_{R,T} = \delta_T \otimes \lambda_R. \tag{3.12}$$

### 3.1.7 Sub-additive Functions

Another class of functions will be important in network calculus are sub-additive functions, which are defined as follows.

**Definition 3.1.11 (Sub-additive function).** *Let $f$ be a function or a sequence of $\mathcal{F}$. Then $f$ is sub-additive if and only if $f(t+s) \leq f(t) + f(s)$ for all $s, t \geq 0$.*

Note that this definition is equivalent to imposing that $f \leq f \otimes f$. If $f(0) = 0$, it is equivalent to imposing that $f \otimes f = f$.

We will see in the following theorem that concave functions passing through the origin are sub-additive. So the piecewise linear function $f_1$ given by (3.1), being concave and passing through the origin, is sub-additive.

The set of sub-additive functions is however larger than that of concave functions: the piecewise linear function $f_2$ given by (3.2) is not concave, yet one check that it verifies Definition 3.1.11 and hence is sub-additive.

Contrary to concave and convex functions, it is not always obvious, from a quick visual inspection of the graph of a function, to establish whether it is sub-additive or not. Consider the two functions $\beta_{R,T} + K'$ and $\beta_{R,T} + K''$, represented respectively on the left and right of Figure 3.5. Although they differ only by the constants $K'$ and $K''$, which are chosen so that $0 < K'' < RT < K' < +\infty$, we will see $\beta_{R,T} + K'$ is sub-additive but not $\beta_{R,T} + K''$. Consider first $\beta_{R,T} + K'$. If $s + t \leq T$, then

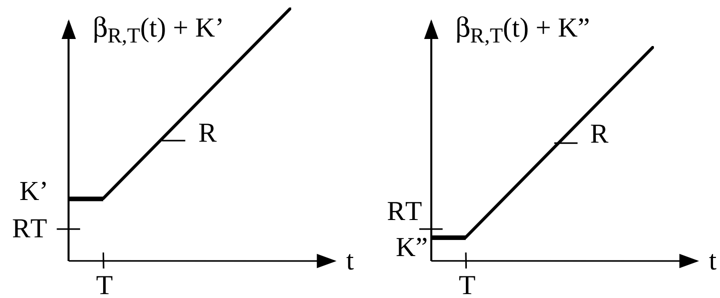



Figure 3.5: Functions $\beta_{R,T} + K'$ (left) and $\beta_{R,T} + K''$ (right). The only difference between them is the value of the constant: $K'' < RT < K'$.

$s, t \leq T$ and

$$\beta_{R,T}(s+t) + K' = K' < 2K' = (\beta_{R,T}(s) + K') + (\beta_{R,T}(t) + K').$$

On the other hand, if $s + t > T$, then, since $K' > RT$,

$$\begin{aligned} \beta_{R,T}(t+s) + K' &= R(t+s-T) + K' \\ &< R(s+t-T) + K' + (K' - RT) \\ &= (R(t-T) + K') + (R(s-T) + K') \\ &\leq (\beta_{R,T}(t) + K') + (\beta_{R,T}(s) + K'), \end{aligned}$$

which proves that $\beta_{R,T} + K'$ is sub-additive. Consider next $\beta_{R,T} + K''$. Pick $s = T$ and $t > T$. Then, since $K'' < RT$,

$$\begin{aligned} &\beta_{R,T}(t+s) + K'' = \\ &\qquad \beta_{R,T}(t+T) + K'' = Rt + K'' = R(t-T) + RT + K'' \\ &\qquad > R(t-T) + K'' + K'' = (\beta_{R,T}(t) + K'') + (\beta_{R,T}(s) + K''), \end{aligned}$$

which proves that $\beta_{R,T} + K''$ is not sub-additive.

Let us list now some properties of sub-additive functions.

**Theorem 3.1.9 (Properties of sub-additive functions).** *Let $f, g \in \mathcal{F}$.*

- ***(Star-shaped functions passing through the origin)*** *If $f$ is star-shaped with $f(0) = 0$, then $f$ is sub-additive.*
- ***(Sum of sub-additive functions)*** *If $f$ and $g$ are sub-additive, so is $(f + g)$.*

- ***(Min-plus convolution of sub-additive functions)*** *If $f$ and $g$ are sub-additive, so is $(f \otimes g)$.*

The first property also implies that concave functions passing through the origin are sub-additive. The proof of the second property is simple and left to the reader, we prove the two others.

**Proof:** (Star-shaped functions passing through the origin) Let $s, t \geq 0$ be given. If $s$ or $t = 0$, one clearly has that $f(s+t) = f(s) + f(t)$. Assume next that $s, t > 0$. As $f$ is star-shaped,

$$\begin{aligned} f(s) &\geq \frac{s}{s+t} f(s+t) \\ f(t) &\geq \frac{t}{s+t} f(s+t) \end{aligned}$$

which sum up to give $f(s) + f(t) \geq f(s+t)$. (Min-plus convolution of sub-additive functions) Let $s, t \geq 0$ be given. Then

$$\begin{aligned} &(f \otimes g)(s) + (f \otimes g)(t) \\ &= \inf_{0 \leq u \leq s} \{f(s-u) + g(u)\} + \inf_{0 \leq v \leq t} \{f(t-v) + g(v)\} \\ &= \inf_{0 \leq u \leq s} \inf_{0 \leq v \leq t} \{f(s-u) + f(t-v) + g(u) + g(v)\} \\ &\geq \inf_{0 \leq u \leq s} \inf_{0 \leq v \leq t} \{f(s+t-(u+v)) + g(u+v)\} \\ &= \inf_{0 \leq u+v \leq s+t} \{f(s+t-(u+v)) + g(u+v)\} \\ &= (f \otimes g)(t+s). \end{aligned}$$

□

The minimum of any number of star-shaped (resp. concave) functions is still a star-shaped (resp. concave) function. If one of them passes through the origin, it is therefore a sub-additive function: for example, as already mentioned earlier, the concave piecewise linear function $f_1$ given by (3.1) is sub-additive. On the other hand the minimum of two sub-additive functions is not, in general, sub-additive. Take for example the minimum between a rate latency function $\beta_{R',T}$ and function $f_2$ given by (3.2), when $R' = 2R/3$. with $R, T$ as defined in (3.2). Both functions are sub-additive, but one can check that $\beta_{R',T} \wedge f_2$ is not.

The first property of the previous theorem tells us that all star-shaped functions are sub-additive. One can check for example that $\beta_{R,T} + K'$ is a star-shaped function (which is not concave), but not $\beta_{R,T} + K''$. One can also wonder if, conversely, all sub-additive functions are star-shaped. The answer is no: take again function $f_2$ given by (3.2), which is sub-additive. It is not star-shaped, because $f(2T)/2T = R/2 < 2R/3 = f(3T)/3T$.

### 3.1.8 Sub-additive Closure

Given a function $f \in \mathcal{F}$, if $f(0) = 0$, then $f \geq f \otimes f \geq 0$. By repeating this operation, we will get a sequence of functions that are each time smaller and converges to some limiting function that, as we will see, is the largest sub-additive function smaller than $f$ and zero in $t = 0$, and is called sub-additive closure of $f$. The formal definition is as follows.

**Definition 3.1.12 (Sub-additive closure).** *Let $f$ be a function or a sequence of $\mathcal{F}$. Denote $f^{(n)}$ the function obtained by repeating $(n-1)$ convolutions of $f$ with itself. By convention, $f^{(0)} = \delta_0$, so that $f^{(1)} = f$, $f^{(2)} = f \otimes f$, etc. Then the sub-additive closure of $f$, denoted by $\overline{f}$, is defined by*

$$\overline{f} = \delta_0 \wedge f \wedge (f \otimes f) \wedge (f \otimes f \otimes f) \wedge \ldots = \inf_{n \geq 0} \left\{ f^{(n)} \right\}. \tag{3.13}$$

**Example.** Let us compute the sub-additive closure of the two functions $\beta_{R,T} + K'$ and $\beta_{R,T} + K''$, represented respectively on the left and right of Figure 3.5. Note first that Rule 7 of Theorem 3.1.5 and Rule 9 of Theorem 3.1.6 yield that for any $K > 0$,

$$(\beta_{R,T} + K) \otimes (\beta_{R,T} + K) = (\beta_{R,T} \otimes \beta_{R,T}) + 2K = \beta_{R,2T} + 2K.$$

Repeating this convolution $n$ times yields that for all integers $n \geq 1$

$$(\beta_{R,T} + K)^{(n)} = \beta_{R,nT} + nK.$$

Now, if $K = K' > RT$ and $t \leq nT$,

$$\begin{aligned} \beta_{R,nT} + nK' &= nK' > (n-1)RT + K' = R(nT - T) + K' \\ &\geq R[t - T]^+ + K' = \beta_{R,T} + K', \end{aligned}$$

whereas if $t > nT$

$$\begin{aligned} \beta_{R,nT} + nK' &= R(t - nT) + nK' = R(t - T) + (n-1)(K' - RT) + K' \\ &> R(t - T) + K' = \beta_{R,T} + K' \end{aligned}$$

so that $(\beta_{R,T} + K')^{(n)} \geq \beta_{R,T} + K'$ for all $n \geq 1$. Therefore (3.13) becomes

$$\overline{\beta_{R,T} + K'} = \delta_0 \wedge \inf_{n \geq 1} \left\{ (\beta_{R,T} + K')^{(n)} \right\} = \delta_0 \wedge (\beta_{R,T} + K'),$$

and is shown on the left of Figure 3.6. On the other hand, if $K = K'' < RT$, the infimum in the previous equation is not reached in $n = 1$ for every $t > 0$, so that the sub-additive closure is now expressed by

$$\overline{\beta_{R,T} + K''} = \delta_0 \wedge \inf_{n \geq 1} \left\{ (\beta_{R,T} + K'')^{(n)} \right\} = \delta_0 \wedge \inf_{n \geq 1} \left\{ (\beta_{R,nT} + nK'') \right\},$$

and is shown on the right of Figure 3.6.

Among all the sub-additive functions that are smaller than $f$ and that are zero in $t = 0$, there is one that is an upper bound for all others; it is equal to $\overline{f}$, as established by the following theorem.

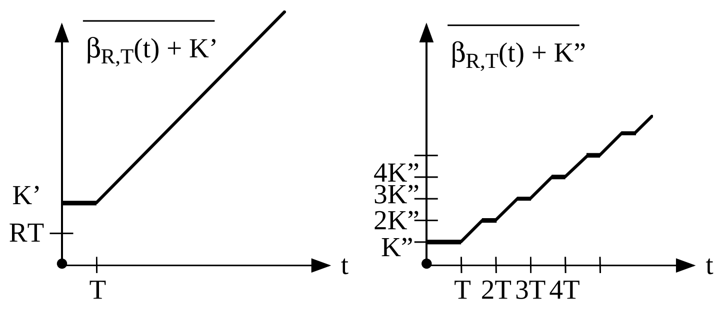



Figure 3.6: The sub-additive closure of functions $\beta_{R,T} + K'$ (left) and $\beta_{R,T} + K''$ (right), when $K'' < RT < K'$.

**Theorem 3.1.10 (Sub-additive closure).** *Let $f$ be a function or a sequence of $\mathcal{F}$, and let $\overline{f}$ be its sub-additive closure. Then (i) $\overline{f} \le f$, $\overline{f} \in \mathcal{F}$ and $\overline{f}$ is sub-additive. (ii) if function $g \in \mathcal{F}$ is sub-additive, with $g(0) = 0$ and $g \le f$, then $g \le \overline{f}$.*

**Proof:** (i) It is obvious from Definition 3.1.12, that $\overline{f} \le f$. By repeating $(n-1)$ times Rule 1 of Theorem 3.1.5, one has that $f^{(n)} \in \mathcal{F}$ for all $n \ge 1$. As $f^{(0)} = \delta_0 \in \mathcal{F}$ too, $\overline{f} = \inf_{n \ge 0}\{f^{(n)}\} \in \mathcal{F}$. Let us show next that $\overline{f}$ is sub-additive. For any integers $n, m \ge 0$, and for any $s, t \ge 0$,

$$
\begin{aligned}
f^{(n+m)}(t+s) &= (f^{(n)} \otimes f^{(m)})(t+s) = \inf_{0 \le u \le t+s}\{f^{(n)}(t+s-u) + f^{(m)}(u)\} \\
&\le f^{(n)}(t) + f^{(m)}(s)
\end{aligned}
$$

so that

$$
\begin{aligned}
\overline{f}(t+s) &= \inf_{n+m \ge 0}\{f^{(n+m)}(t+s)\} = \inf_{n,m \ge 0}\{f^{(n+m)}(t+s)\} \\
&\le \inf_{n,m \ge 0}\{f^{(n)}(t) + f^{(m)}(s)\} \\
&= \inf_{n \ge 0}\{f^{(n)}(t)\} + \inf_{m \ge 0}\{f^{(m)}(s)\} = \overline{f}(t) + \overline{f}(s)
\end{aligned}
$$

which shows that $\overline{f}$ is sub-additive. (ii) Next, suppose that $g \in \mathcal{F}$ is sub-additive, $g(0) = 0$ and $g \le f$. Suppose that for some $n \ge 1$, $f^{(n)} \ge g$. Clearly, this holds for $n = 0$ (because $g(0) = 0$ implies that $g \le \delta_0 = f^{(0)}$) and for $n = 1$. Now, this assumption and the sub-additivity of $g$ yield that for any $0 \le s \le t$, $f^{(n)}(t-s) + f(s) \ge g(t-s) + g(s) \ge g(t)$ and hence that $f^{(n+1)}(t) \ge g(t)$. By recursion on $n$, $f^{(n)} \ge g$ for all $n \ge 0$, and therefore $\overline{f} = \inf_{n \ge 0}\{f^{(n)}\} \ge g$. □

**Corollary 3.1.1 (Sub-additive closure of a sub-additive function).** *Let $f \in \mathcal{F}$. Then the three following statements are equivalent: (i) $f(0) = 0$ and $f$ is sub-additive (ii) $f \otimes f = f$ (iii) $\overline{f} = f$.*

**Proof:** (i) $\Rightarrow$ (ii) follows immediately from from Definition 3.1.11. (ii) $\Rightarrow$ (iii): first note that $f \otimes f = f$ implies that $f^{(n)} = f$ for all $n \geq 1$. Second, note that $(f \otimes f)(0) = f(0) + f(0)$, which implies that $f(0) = 0$. Therefore $\overline{f} = \inf_{n \geq 0}\{f^{(n)}\} = \delta_0 \wedge f = f$. (iii) $\Rightarrow$ (i) follows from Theorem 3.1.10. □

The following theorem establishes some additional useful properties of the sub-additive closure of a function.

**Theorem 3.1.11 (Other properties of sub-additive closure).** *Let $f, g \in \mathcal{F}$*

- ***(Isotonicity)*** *If $f \leq g$ then $\overline{f} \leq \overline{g}$.*
- ***(Sub-additive closure of a minimum)*** *$\overline{f \wedge g} = \overline{f} \otimes \overline{g}$.*
- ***(Sub-additive closure of a convolution)*** *$\overline{f \otimes g} \geq \overline{f} \otimes \overline{g}$. If $f(0) = g(0) = 0$ then $\overline{f \otimes g} = \overline{f} \otimes \overline{g}$.*

**Proof:** (Isotonocity) Suppose that we have shown that for some $n \geq 1$, $f^{(n)} \geq g^{(n)}$ (Clearly, this holds for $n = 0$ and for $n = 1$). Then applying Theorem 3.1.7 we get

$$f^{(n+1)} = f^{(n)} \otimes f \geq g^{(n)} \otimes g = g^{(n+1)},$$

which implies by recursion on $n$ that $\overline{f} \leq \overline{g}$. (Sub-additive closure of a minimum) One easily shows, using Theorem 3.1.5, that

$$(f \wedge g)^{(2)} = (f \otimes f) \wedge (f \otimes g) \wedge (g \otimes g).$$

Suppose that we have shown that for some $n \geq 0$, the expansion of $(f \wedge g)^{(n)}$ is

$$\begin{aligned}
&(f \wedge g)^{(n)} = \\
&\quad f^{(n)} \wedge (f^{(n-1)} \otimes g) \wedge (f^{(n-2)} \otimes g^{(2)}) \wedge \ldots \wedge g^{(n)} = \\
&\quad \inf_{0 \leq k \leq n} \left\{ f^{(n-k)} \otimes g^{(k)} \right\}.
\end{aligned}$$

Then

$$\begin{aligned}
(f \wedge g)^{(n+1)} &= (f \wedge g) \otimes (f \wedge g)^{(n)} = \left\{ f \otimes (f \wedge g)^{(n)} \right\} \wedge \left\{ g \otimes (f \wedge g)^{(n)} \right\} \\
&= \inf_{0 \leq k \leq n} \left\{ f^{(n+1-k)} \otimes g^{(k)} \right\} \wedge \inf_{0 \leq k \leq n} \left\{ f^{(n-k)} \otimes g^{(k+1)} \right\} \\
&= \inf_{0 \leq k \leq n} \left\{ f^{(n+1-k)} \otimes g^{(k)} \right\} \wedge \inf_{1 \leq k' \leq n+1} \left\{ f^{(n+1-k')} \otimes g^{(k')} \right\} \\
&= \inf_{0 \leq k \leq n+1} \left\{ f^{(n+1-k)} \otimes g^{(k)} \right\}
\end{aligned}$$

which establishes the recursion for all $n \geq 0$. Therefore

$$\begin{aligned}
\overline{f \wedge g} &= \inf_{n\geq 0}\inf_{0\leq k\leq n}\left\{f^{(n-k)}\otimes g^{(k)}\right\} = \inf_{k\geq 0}\inf_{n\geq k}\left\{f^{(n-k)}\otimes g^{(k)}\right\}\\
&= \inf_{k\geq 0}\inf_{l\geq 0}\left\{f^{(l)}\otimes g^{(k)}\right\} = \inf_{k\geq 0}\left\{\inf_{l\geq 0}\{f^{(l)}\}\otimes g^{(k)}\right\}\\
&= \inf_{k\geq 0}\left\{\overline{f}\otimes g^{(k)}\right\} = \overline{f}\otimes \inf_{k\geq 0}\{g^{(k)}\} = \overline{f}\otimes\overline{g}.
\end{aligned}$$

(Sub-additive closure of a convolution) Using the same recurrence argument as above, one easily shows that $(f\otimes g)^{(n)} = f^{(n)}\otimes g^{(n)}$, and hence that

$$\begin{aligned}
\overline{f\otimes g} &= \inf_{n\geq 0}\left\{(f\otimes g)^{(n)}\right\} = \inf_{n\geq 0}\left\{f^{(n)}\otimes g^{(n)}\right\}\\
&\geq \inf_{n,m\geq 0}\left\{f^{(n)}\otimes g^{(m)}\right\}\\
&= \left(\inf_{n\geq 0}\left\{f^{(n)}\right\}\right)\otimes\left(\inf_{m\geq 0}\left\{g^{(m)}\right\}\right) = \overline{f}\otimes\overline{g}. \qquad (3.14)
\end{aligned}$$

If $f(0)=g(0)=0$, Rule 8 in Theorem 3.1.6 yields that $f\otimes g \leq f\wedge g$, and therefore that $\overline{f\otimes g}\leq\overline{f\wedge g}$. Now we have just shown above that $\overline{f\wedge g}=\overline{f}\otimes\overline{g}$, so that

$$\overline{f\otimes g}\leq\overline{f}\otimes\overline{g}.$$

Combining this result with (3.14), we get $\overline{f\otimes g}=\overline{f}\otimes\overline{g}$. □

Let us conclude this section with an example illustrating the effect that a difference in taking $t$ continuous or discrete may have. This example is the computation of the sub-additive closure of

$$f(t)=\begin{cases} t^2 & \text{if} \quad t>0\\ 0 & \text{if} \quad t\leq 0\end{cases}$$

Suppose first that $t\in\mathbb{R}$. Then we compute that

$$(f\otimes f)(t) = \inf_{0\leq s\leq t}\left\{(t-s)^2+s^2\right\} = (t/2)^2+(t/2)^2 = t^2/2$$

as the infimum is reached in $s=t/2$. By repeating this operation $n$ times, we obtain

$$\begin{aligned}
f^{(n)}(t) = \inf_{0\leq s\leq t}&\left\{(t-s)^2+(f^{(n-1)})^2(s)\right\} =\\
&\inf_{0\leq s\leq t}\left\{(t-s)^2+s^2/(n-1)\right\} = t^2/n
\end{aligned}$$

as the infimum is reached in $s=t(1-1/n)$. Therefore

$$\overline{f}(t)=\inf_{n\geq 0}\{t^2/n\}=\lim_{n\to\infty}t^2/n=0.$$

Consequently, if $t\in\mathbb{R}$, the sub-additive closure of function $f$ is

$$\overline{f}=0,$$

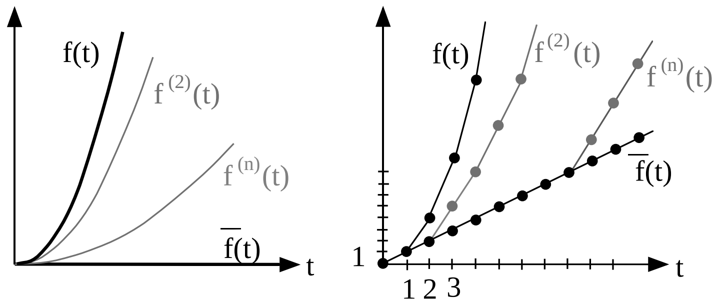



Figure 3.7: The sub-additive closure of $f(t) = t\lambda_1(t)$, when $t \in \mathbb{R}$ (left) and when $t \in \mathbb{Z}$ (right).

as shown on the left of Figure 3.7.

Now, if $t \in \mathbb{Z}$, the sequence $f(t)$ is convex and piecewise linear, as we can always connect the different successive points $(t, t^2)$ for all $t = 0, 1, 2, 3, \ldots$: the resulting graph appears as a succession of segments of slopes equal to $(2t + 1)$ (the first segment in particular has slope 1), and of projections on the horizontal axis having a length equal to 1, as shown on the right of Figure 3.7. Therefore we can apply Rule 9 of Theorem 3.1.6, which yields that $f \otimes f$ is obtained by doubling the length of the different linear segments of $f$, and putting them end-to-end by increasing slopes. The analytical expression of the resulting sequence is

$$(f \otimes f)(t) = \min_{0 \le s \le t} \left\{ (t - s)^2 + s^2 \right\} = \lceil t^2/2 \rceil.$$

Sequence $f^{(2)} = f \otimes f$ is again convex and piecewise linear. Note the first segment has slope $1$, but has now a double length. If we repeat $n$ times this convolution, it will result in a convex, piecewise linear sequence $f^{(n)}(t)$ whose first segment has slope 1 and horizontal length $n$:

$$f^{(n)}(t) = t \quad \text{ if } 0 \le t \le n,$$

as shown on the right of Figure 3.7. Consequently, the sub-additive closure of sequence $f$ is obtained by letting $n \to \infty$, and is therefore $\overline{f}(t) = t$ for $t \ge 0$. Therefore, if $t \in \mathbb{Z}$,

$$\overline{f} = \lambda_1.$$

### 3.1.9 Min-plus Deconvolution

The dual operation (in a sense that will clarified later on) of the min-plus convolution is the min-plus deconvolution. Similar considerations as the ones of Subsection 3.1.1 can be made on the difference between a $\sup$ and a $\max$. Notation $\vee$ stands for $\sup$ or, if it exists, for $\max$: $a \vee b = \max\{a, b\}$.

**Definition 3.1.13 (Min-plus deconvolution).** *Let $f$ and $g$ be two functions or sequences of $\mathcal{F}$. The min-plus deconvolution of $f$ by $g$ is the function*

$$(f \oslash g)(t) = \sup_{u \geq 0} \{f(t+u) - g(u)\} . \tag{3.15}$$

If both $f(t)$ and $g(t)$ are infinite for some $t$, then Equation (3.15) is not defined. Contrary to min-plus convolution, function $(f \oslash g)(t)$ is not necessarily zero for $t \leq 0$, and hence this operation is not closed in $\mathcal{F}$, as shown by the following example.

**Example.** Consider again the two functions $\gamma_{r,b}$ and $\beta_{R,T}$, with $0 < r < R$, and let us compute the min-plus deconvolution of $\gamma_{r,b}$ by $\beta_{R,T}$. We have that

$$\begin{aligned}
&(\gamma_{r,b} \oslash \beta_{R,T})(t) \\
&= \sup_{u \geq 0} \left\{\gamma_{r,b}(t+u) - R[u-T]^+\right\} \\
&= \sup_{0 \leq u \leq T} \left\{\gamma_{r,b}(t+u) - R[u-T]^+\right\} \vee \sup_{u>T} \left\{\gamma_{r,b}(t+u) - R[u-T]^+\right\} \\
&= \sup_{0 \leq u \leq T} \{\gamma_{r,b}(t+u)\} \vee \sup_{u>T} \{\gamma_{r,b}(t+u) - Ru + RT\} \\
&= \{\gamma_{r,b}(t+T)\} \vee \sup_{u>T} \{\gamma_{r,b}(t+u) - Ru + RT\} .
\end{aligned} \tag{3.16}$$

Let us first compute this expression for $t \leq -T$. Then $\gamma_{r,b}(t+T) = 0$ and (3.16) becomes

$$\begin{aligned}
&(\gamma_{r,b} \oslash \beta_{R,T})(t) \\
&= 0 \vee \sup_{T<u \leq -t} \{\gamma_{r,b}(t+u) - Ru + RT\} \\
&\qquad \vee \sup_{u>-t} \{\gamma_{r,b}(t+u) - Ru + RT\} \\
&= 0 \vee \sup_{T<u \leq -t} \{0 - Ru + RT\} \vee \sup_{u>-t} \{b + r(t+u) - Ru + RT\} \\
&= 0 \vee 0 \vee \{b + Rt + RT\} = [b + R(t+T)]^+ .
\end{aligned}$$

Let us next compute $(\gamma_{r,b} \oslash \beta_{R,T})(t)$ for $t > -T$. Then (3.16) becomes

$$\begin{aligned}
(\gamma_{r,b} \oslash \beta_{R,T})(t) &= \{b + r(t+T)\} \vee \sup_{u>T} \{b + r(t+u) - Ru + RT\} \\
&= \{b + r(t+T)\} \vee \{b + r(t+T)\} = b + r(t+T).
\end{aligned}$$

The result is shown in Figure 3.8.

Let us now state some properties of $\oslash$ (Other properties will be given in the next section).

**Theorem 3.1.12 (Properties of $\oslash$).** *Let $f, g, h \in \mathcal{F}$.*

- ***Rule 11 (Isotonicity of $\oslash$)*** *If $f \leq g$, then $f \oslash h \leq g \oslash h$ and $h \oslash f \geq h \oslash g$.*

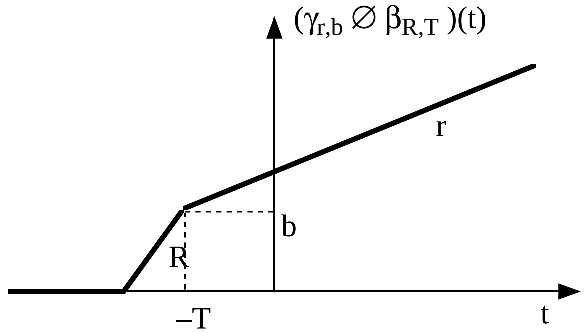



Figure 3.8: Function $\gamma_{r,b} \oslash \beta_{R,T}$ when $0 < r < R$.

- ***Rule 12 (Composition of*** $\oslash$***)*** $(f \oslash g) \oslash h = f \oslash (g \otimes h)$.
- ***Rule 13 (Composition of*** $\oslash$ ***and*** $\otimes$***)*** $(f \otimes g) \oslash g \le f \otimes (g \oslash g)$.
- ***Rule 14 (Duality between*** $\oslash$ ***and*** $\otimes$***)*** $f \oslash g \le h$ *if and only if* $f \le g \otimes h$.
- ***Rule 15 (Self-deconvolution)*** $(f \oslash f)$ *is a sub-additive function of* $\mathcal{F}$ *such that* $(f \oslash f)(0) = 0$.

**Proof:** (Rule 11) If $f \le g$, then for any $h \in \mathcal{F}$

$$\begin{aligned}(f \oslash h)(t) = \sup_{u \ge 0}\{f(t+u) - h(u)\} &\le \sup_{u \ge 0}\{g(t+u) - h(u)\} = (g \oslash h)(t)\\ (h \oslash f)(t) = \sup_{u \ge 0}\{h(t+u) - f(u)\} &\ge \sup_{u \ge 0}\{h(t+u) - g(u)\} = (h \oslash g)(t).\end{aligned}$$

(Rule 12) One computes that

$$\begin{aligned}((f \oslash g) \oslash h)(t) &= \sup_{u \ge 0}\{(f \oslash g)(t+u) - h(u)\}\\ &= \sup_{u \ge 0}\left\{\sup_{v \ge 0}\{f(t+u+v) - g(v)\} - h(u)\right\}\\ &= \sup_{u \ge 0}\left\{\sup_{v' \ge u}\{f(t+v') - g(v'-u)\} - h(u)\right\}\\ &= \sup_{u \ge 0}\sup_{v' \ge u}\{f(t+v') - \{g(v'-u) + h(u)\}\}\\ &= \sup_{v' \ge 0}\sup_{0 \le u \le v'}\{f(t+v') - \{g(v'-u) + h(u)\}\}\\ &= \sup_{v' \ge 0}\left\{f(t+v') - \inf_{0 \le u \le v'}\{g(v'-u) + h(u)\}\right\}\\ &= \sup_{v' \ge 0}\{f(t+v') - (g \otimes h)(v')\} = (f \oslash (g \otimes h))(t).\end{aligned}$$

(Rule 13) One computes that

$$\begin{aligned}
((f \otimes g) \oslash g)(t) &= \sup_{u \geq 0} \{(f \otimes g)(t+u) - g(u)\} \\
&= \sup_{u \geq 0} \inf_{0 \leq s \leq t+u} \{f(t+u-s) + g(s) - g(u)\} \\
&= \sup_{u \geq 0} \inf_{-u \leq s' \leq t} \{f(t-s') + g(s'+u) - g(u)\} \\
&\leq \sup_{u \geq 0} \inf_{0 \leq s' \leq t} \{f(t-s') + g(s'+u) - g(u)\} \\
&\leq \sup_{u \geq 0} \inf_{0 \leq s' \leq t} \left\{ f(t-s') + \sup_{v \geq 0}\{g(s'+v) - g(v)\} \right\} \\
&= \inf_{0 \leq s' \leq t} \left\{ f(t-s') + \sup_{v \geq 0}\{g(s'+v) - g(v)\} \right\} \\
&= \inf_{0 \leq s' \leq t} \{f(t-s') + (g \oslash g)(s')\} = (f \otimes (g \oslash g))(t).
\end{aligned}$$

(Rule 14) Suppose first that $(f \oslash g)(s) \leq h(s)$ for all $s$. Take any $s, v \geq 0$. Then

$$f(s+v) - g(v) \leq \sup_{u \geq 0} \{f(s+u) - g(u)\} = (f \oslash g)(s) \leq h(s)$$

or equivalently,

$$f(s+v) \leq g(v) + h(s).$$

Let $t = s + v$. The former inequality can be written as

$$f(t) \leq g(t-s) + h(s).$$

As it is verified for all $t \geq s \geq 0$, it is also verified in particular for the value of $s$ that achieves the infimum of the right-hand side of this inequality. Therefore it is equivalent to

$$f(t) \leq \inf_{0 \leq s \leq t} \{g(t-s) + h(s)\} = (g \otimes h)(t)$$

for all $t \geq 0$. Suppose now that for all $v$, $f(v) \leq (g \otimes h)(v)$. Pick any $t \in \mathbb{R}$. Then, since $g, h \in \mathcal{F}$,

$$f(v) \leq \inf_{0 \leq s \leq v} \{g(v-s) + h(s)\} = \inf_{s \in \mathbb{R}} \{g(v-s) + h(s)\} \leq g(t-v) + h(t).$$

Let $u = t - v$, the former inequality can be written as

$$f(t+u) - g(u) \leq h(t).$$

As this is true for all $u$, it is also verified in particular for the value of $u$ that achieves the supremum of the left-hand side of this inequality. Therefore it is equivalent to

$$\sup_{u \in \mathbb{R}} \{f(t+u) - g(u)\} \leq h(t).$$

Now if $u < 0$, $g(u) = 0$, so that $\sup_{u<0}\{f(t+u) - g(u)\} = f(t)$ and the former inequality is identical to

$$\sup_{u \geq 0}\{f(t+u) - g(u)\} \leq h(t)$$

for all $t$. (Rule 15) It is immediate to check that $(f \oslash f)(0) = 0$ and that $f \oslash f$ is wide-sense increasing. Now,

$$\begin{aligned}
&(f \oslash f)(s) + (f \oslash f)(t) \\
&\quad = \sup_{u \geq 0}\{f(t+u) - f(u)\} + \sup_{v \geq 0}\{f(s+v) - f(v)\} \\
&\quad = \sup_{u \geq 0}\{f(t+u) - f(u)\} + \sup_{w \geq -t}\{f(s+t+w) - f(t+w)\} \\
&\quad \geq \sup_{w \geq 0}\left\{\sup_{u \geq 0}\{f(t+u) - f(u) + f(s+t+w) - f(t+w)\}\right\} \\
&\quad \geq \sup_{w \geq 0}\{f(t+w) - f(w) + f(s+t+w) - f(t+w)\} \\
&\quad = (f \oslash f)(s+t).
\end{aligned}$$

□

Let us conclude this section by a special property that applies to self-deconvolution of sub-additive functions.

**Theorem 3.1.13 (Self-deconvolution of sub-additive functions).** *Let $f \in \mathcal{F}$. Then $f(0) = 0$ and $f$ is sub-additive if and only if $f \oslash f = f$.*

**Proof:** $(\Rightarrow)$ If $f$ is sub-additive, then for all $t, u \geq 0$, $f(t+u) - f(u) \leq f(t)$ and therefore for all $t \geq 0$,

$$(f \oslash f)(t) = \sup_{u \geq 0}\{f(t+u) - f(u)\} \leq f(t).$$

On the other hand, if $f(0) = 0$,

$$(f \oslash f)(t) = \sup_{u \geq 0}\{f(t+u) - f(u)\} \geq f(t) - f(0) = f(t).$$

Combining both equations, we get that $f \oslash f = f$. $(\Leftarrow)$ Suppose now that $f \oslash f = f$. Then $f(0) = (f \oslash f)(0) = 0$ and for any $t, u \geq 0$, $f(t) = (f \oslash f)(t) \geq f(t+u) - f(u)$ so that $f(t) + f(u) \geq f(t+u)$, which shows that $f$ is sub-additive. □

### 3.1.10 Representation of Min-plus Deconvolution by Time Inversion

Min-plus deconvolution can be represented in the time inverted domain by min-plus convolution, for functions that have a finite lifetime. Function $g \in \mathcal{G}$ has a finite lifetime if there exist some finite $T_0$ and $T$ such that $g(t) = 0$ if $t \leq T_0$ and $g(t) = g(T)$ for $t \geq T$. Call $\widehat{\mathcal{G}}$ the subset of $\mathcal{G}$, which contains functions having a finite lifetime. For function $g \in \widehat{\mathcal{G}}$, we use the notation $g(+\infty)$ as a shorthand for $\sup_{t\in\mathbb{R}}\{g(t)\} = \lim_{t\to+\infty} g(t)$.

**Lemma 3.1.1.** *Let $f \in \mathcal{F}$ be such that $\lim_{t\to+\infty} f(t) = +\infty$. For any $g \in \widehat{\mathcal{G}}$, $g \oslash f$ is also in $\widehat{\mathcal{G}}$ and $(g \oslash f)(+\infty) = g(+\infty)$.*

**Proof:** Define $L = g(+\infty)$ and call $T$ a number such that $g(t) = L$ for $t \geq T$. $f(0) \geq 0$ implies that $g \oslash f \leq g(+\infty) = g(L)$. Thus

$$(g \oslash f)(t) \leq L \text{ for } t \geq T. \tag{3.17}$$

Now since $\lim_{t\to+\infty} f(t) = +\infty$, there is some $T_1 > T$ such that $f(t) \geq L$ for all $t > T_1$. Now let $t > 2T_1$. If $u > T_1$, then $f(u) \geq L$. Otherwise, $u \leq T_1$ thus $t - u \geq t - T_1 > T_1$ thus $g(t-u) \geq L$. Thus in all cases $f(u) + g(t-u) \geq L$. Thus we have shown that

$$(g \otimes f)(t) \geq L \text{ for } t > 2T_1. \tag{3.18}$$

Combining (3.17) and (3.18) shows the lemma. □

**Definition 3.1.14 (Time Inversion).** *For a fixed $T \in [0, +\infty[$, the inversion operator $\Phi_T$ is defined on $\widehat{\mathcal{G}}$ by:*

$$\Phi_T(f)(g) = g(+\infty) - g(T-t)$$

Graphically, time inversion can be obtained by a rotation of $180^o$ around the point $(\frac{T}{2}, \frac{g(+\infty)}{2})$. It is simple to check that $\Phi_T(g)$ is in $\widehat{\mathcal{G}}$, that time inversion is symmetrical ($\Phi_T(\Phi_T(g)) = g$) and preserves the total value ($\Phi_T(g)(+\infty) = g(+\infty)$). Lastly, for any $\alpha$ and $T$, $\alpha$ is an arrival curve for $g$ if and only if $\alpha$ is an arrival curve for $\Phi_T(g)$.

**Theorem 3.1.14 (Representation of Deconvolution by Time Inversion).** *Let $g \in \widehat{\mathcal{G}}$, and let $T$ be such that $g(T) = g(+\infty)$. Let $f \in \mathcal{F}$ be such that $\lim_{t\to+\infty} f(t) = +\infty$. Then*

$$g \oslash f = \Phi_T(\Phi_T(g) \otimes f) \tag{3.19}$$

The theorem says that $g \oslash f$ can be computed by first inverting time, then computing the min-plus convolution between $f$, and the time-inverted function $g$, and then inverting time again. Figure 3.9 shows a graphical illustration.

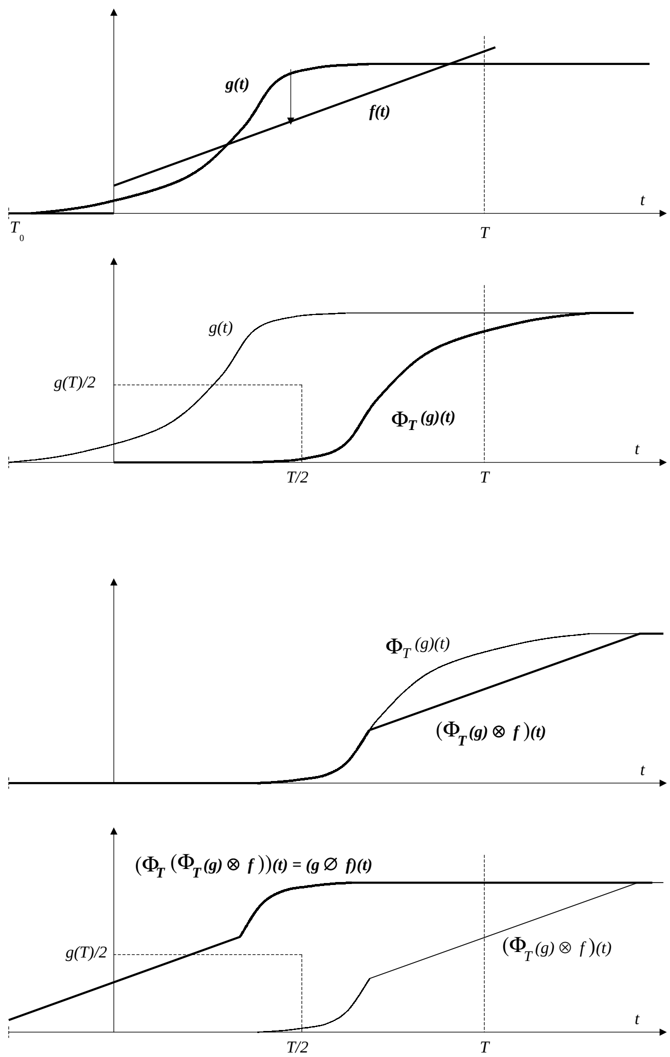



Figure 3.9: Representation of the min-plus deconvolution of $g$ by $f = \gamma_{r,b}$ by time-inversion. From top to bottom: functions $f$ and $g$, function $\Phi_T(g)$, function $\Phi_T(g) \otimes f$ and finally function $g \oslash f = \Phi_T(\Phi_T(g) \otimes f)$.

**Proof:** The proof consists in computing the right handside in Equation (3.19). Call $\hat{g} = \Phi_T(g)$. We have, by definition of the inversion

$$\Phi_T(\Phi_T(g) \otimes f) = \Phi_T(\hat{g} \otimes f) = (\hat{g} \otimes f)(+\infty) - (\hat{g} \otimes f)(T - t)$$

Now from Lemma 3.1.1 and the preservation of total value:

$$(\hat{g} \otimes f)(+\infty) = \hat{g}(+\infty) = g(+\infty)$$

Thus, the right-handside in Equation (3.19) is equal to

$$g(+\infty) - (\hat{g} \otimes f)(T - t) = g(+\infty) - \inf_{u \geq 0}\{\hat{g}(T - t - u) + f(u)\}$$

Again by definition of the inversion, it is equal to

$$g(+\infty) - \inf_{u \geq 0}\{g(+\infty) - g(t + u) + f(u)\} = \sup_{u \geq 0}\{g(t + u) - f(u)\}.$$

□

### 3.1.11 Vertical and Horizontal Deviations

The deconvolution operator allows to easily express two very important quantities in network calculus, which are the maximal vertical and horizontal deviations between the graphs of two curves $f$ and $g$ of $\mathcal{F}$. The mathematical definition of these two quantities is as follows.

**Definition 3.1.15 (Vertical and horizontal deviations).** *Let $f$ and $g$ be two functions or sequences of $\mathcal{F}$. The vertical deviation $v(f, g)$ and horizontal deviation $h(f, g)$ are defined as*

$$v(f, g) = \sup_{t \geq 0}\{f(t) - g(t)\} \qquad (3.20)$$

$$h(f, g) = \sup_{t \geq 0}\{\inf\{d \geq 0 \text{ such that } f(t) \leq g(t + d)\}\}. \qquad (3.21)$$

Figure 3.10 illustrates these two quantities on an example.

Note that (3.20) can be recast as

$$v(f, g) = (f \oslash g)(0) \qquad (3.22)$$

whereas (3.20) is equivalent to requiring that $h(f, g)$ is the smallest $d \geq 0$ such that for all $t \geq 0$, $f(t) \leq g(t + d)$ and can therefore be recast as

$$h(f, g) = \inf\{d \geq 0 \text{ such that } (f \oslash g)(-d) \leq 0\}.$$

Now the horizontal deviation can be more easily computed from the pseudo-inverse of $g$. Indeed, Definition 3.1.7 yields that

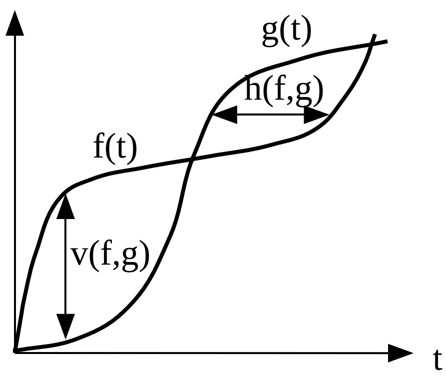



Figure 3.10: The horizontal and vertical deviations between functions $f$ and $g$.

$$\begin{aligned} g^{-1}(f(t)) &= \inf\{\Delta \text{ such that } g(\Delta) \geq f(t)\} \\ &= \inf\{d \geq 0 \text{ such that } g(t+d) \geq f(t)\} + t \end{aligned}$$

so that (3.21) can be expressed as

$$h(f,g) = \sup_{t \geq 0}\left\{g^{-1}(f(t)) - t\right\} = (g^{-1}(f) \oslash \lambda_1)(0). \tag{3.23}$$

We have therefore the following expression of the horizontal deviation between $f$ and $g$:

**Proposition 3.1.1 (Horizontal deviation).**

$$h(f,g) = \sup_{t \geq 0}\left\{g^{-1}(f(t)) - t\right\}.$$

## 3.2 Max-plus Calculus

Similar definitions, leading to similar properties, can be derived if we replace the infimum (or minimum, it is exists) by a supremum (or maximum, if it exists). We use the notation $\vee$ for denoting $\sup$ or $\max$. In particular, one can show that $(\mathbb{R} \cup \{-\infty\}, \vee, +)$ is also a dioid, and construct a max-plus convolution and deconvolution, which are defined as follows.

### 3.2.1 Max-plus Convolution and Deconvolution

**Definition 3.2.1 (Max-plus convolution).** *Let $f$ and $g$ be two functions or sequences of $\mathcal{F}$. The max-plus convolution of $f$ and $g$ is the function*

$$(f \overline{\otimes} g)(t) = \sup_{0 \leq s \leq t}\{f(t-s) + g(s)\}. \tag{3.24}$$

*(If $t < 0$, $(f\overline{\otimes}g)(t) = 0$).*

**Definition 3.2.2 (Max-plus deconvolution).** *Let $f$ and $g$ be two functions or sequences of $\mathcal{F}$. The max-plus deconvolution of $f$ by $g$ is the function*

$$(f\overline{\oslash}g)(t) = \inf_{u\geq 0}\left\{f(t+u) - g(u)\right\}. \tag{3.25}$$

### 3.2.2 Linearity of Min-plus Deconvolution in Max-plus Algebra

Min-plus deconvolution is, in fact, an operation that is linear in $(\mathbb{R}^+, \vee, +)$. Indeed, one easily shows the following property.

**Theorem 3.2.1 (Linearity of $\oslash$ in max-plus algebra).** *Let $f, g, h \in \mathcal{F}$.*

- ***Rule 16 (Distributivity of $\oslash$ with respect to $\vee$)*** $(f \vee g) \oslash h = (f \oslash h) \vee (g \oslash h)$.
- ***Rule 17 (Addition of a constant)*** *For any $K \in \mathbb{R}^+$, $(f + K) \oslash g = (f \oslash g) + K$.*

Min-plus convolution is not, however, a linear operation in $(\mathbb{R}^+, \vee, +)$, because in general

$$(f \vee g) \otimes h \neq (f \otimes h) \vee (g \otimes h).$$

Indeed, take $f = \beta_{3R,T}$, $g = \lambda_R$ and $h = \lambda_{2R}$ for some $R, T > 0$. Then using Rule 9, one easily computes (see Figure 3.11) that

$$\begin{aligned} f \otimes h &= \beta_{3R,T} \otimes \lambda_{2R} = \beta_{2R,T} \\ g \otimes h &= \lambda_R \otimes \lambda_{2R} = \lambda_R \\ (f \vee g) \otimes h &= (\beta_{3R,T} \vee \lambda_R) \otimes \lambda_{2R} = \beta_{2R,3T/4} \vee \lambda_R \\ &\neq \beta_{2R,T} \vee \lambda_R = (f \otimes h) \vee (g \otimes h). \end{aligned}$$

Conversely, we have seen that min-plus convolution is a linear operation in $(\mathbb{R}^+, \wedge, +)$, and one easily shows that min–plus deconvolution is not linear in $(\mathbb{R}^+, \wedge, +)$. Finally, let us mention that one can also replace $+$ by $\wedge$, and show that $(\mathbb{R} \cup \{+\infty\} \cup \{-\infty\}, \vee, \wedge)$ is also a dioid. **Remark** *However, as we have seen above, as soon as the three operations $\wedge$, $\vee$ and $+$ are involved in a computation, one must be careful before applying any distribution.*

## 3.3 Exercises

**Exercise 3.1.** *1. Compute $\alpha \otimes \delta$ for any function $\alpha$*

*2. Express the rate-latency function by means of $\delta$ and $\lambda$ functions.*

**Exercise 3.2.** *1. Compute $\bigotimes_i \beta_i$ when $\beta_i$ is a rate-latency function*

*2. Compute $\beta_1 \otimes \beta_2$ with $\beta_1(t) = R(t-T)^+$ and $\beta_2(t) = (rt+b)1_{\{t>0\}}$*

**Exercise 3.3.** *1. Is $\otimes$ distributive with respect to the $\min$ operator ?*

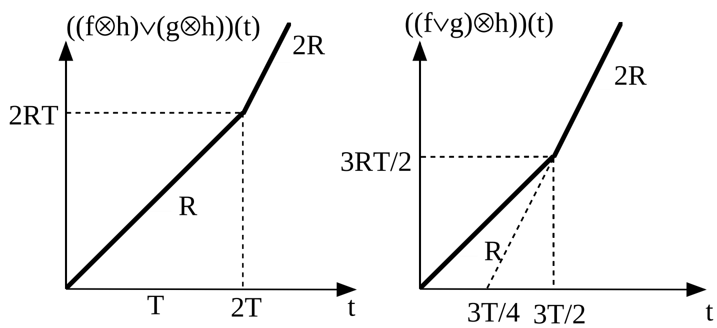



Figure 3.11: Function $(f \otimes h) \vee (g \otimes h)$ (left) and $(f \vee g) \otimes h$ (right) when $f = \beta_{3R,T}$, $g = \lambda_R$ and $h = \lambda_{2R}$ for some $R, T > 0$.

# Chapter 4

# Min-plus and Max-plus System Theory

In Chapter 3 we have introduced the basic operations to manipulate functions and sequences in Min-Plus or Max-Plus algebra. We have studied in detail the operations of convolution, deconvolution and sub-additive closure. These notions form the mathematical cornerstone on which a first course of network calculus has to be built.

In this chapter, we move one step further, and introduce the theoretical tools to solve more advanced problems in network calculus developed in the second half of the book. The core object in Chapter 3 were *functions and sequences* on which operations could be performed. We will now place ourselves at the level of *operators* mapping an input function (or sequence) to an output function or sequence. Max-plus system theory is developed in detail in [26], here we focus on the results that are needed for the remaining chapters of the book. As in Chapter 3, we focus here Min-Plus System Theory, as Max-Plus System Theory follows easily by replacing minimum by maximum, and infimum by supremum.

## 4.1 Min-plus and Max-plus Operators

### 4.1.1 Vector Notations

Up to now, we have only worked with scalar operations on scalar functions in $\mathcal{F}$ or $\mathcal{G}$. In this chapter, we will also work with vectors and matrices. The operations are extended in a straightforward manner.

Let $J$ be a finite, positive integer. For vectors $\vec{z}, \vec{z'} \in \mathbb{R}^{+J}$, we define $\vec{z} \wedge \vec{z}'$ as the coordinate-wise minimum of $\vec{z}$ and $\vec{z}'$, and similarly for the $+$ operator. We write $\vec{z} \leq \vec{z'}$ with the meaning that $z_j \leq z'_j$ for $1 \leq j \leq J$. Note that the comparison so defined is not a total order, that is, we cannot guarantee that either $\vec{z} \leq \vec{z'}$ or

$\vec{z'} \leq \vec{z}$ holds. For a constant $K$, we note $\vec{z} + K$ the vector defined by adding $K$ to *all* elements of $\vec{z}$.

We denote by $\mathcal{G}^J$ the set of $J$-dimensional wide-sense increasing real-valued functions or sequences of parameter $t$, and $\mathcal{F}^J$ the subset of functions that are zero for $t < 0$.

For sequences or functions $\vec{x}(t)$, we note similarly $(\vec{x} \wedge \vec{y})(t) = \vec{x}(t) \wedge \vec{y}(t)$ and $(\vec{x} + K)(t) = \vec{x}(t) + K$ for all $t \geq 0$, and write $\vec{x} \leq \vec{y}$ with the meaning that $\vec{x}(t) \leq \vec{y}(t)$ for all $t$.

For matrices $A, B \in \mathbb{R}^{+\,J} \times \mathbb{R}^{+\,J}$, we define $A \wedge B$ as the entry-wise minimum of $A$ and $B$. For vector $\vec{z} \in \mathbb{R}^{+\,J}$, the 'multiplication' of vector $\vec{z} \in \mathbb{R}^{+\,J}$ by matrix $A$ is – remember that in min-plus algebra, multiplication is the $+$ operation – by

$$A + \vec{z},$$

and has entries $\min_{1 \leq j \leq J}(a_{ij} + z_j)$. Likewise, the 'product' of two matrices $A$ and $B$ is denoted by $A + B$ and has entries $\min_{1 \leq j \leq J}(a_{ij} + b_{jk})$ for $1 \leq i, k \leq J$.

Here is an example of a 'multiplication' of a vector by a matrix, when $J = 2$

$$\begin{bmatrix} 5 & 3 \\ 1 & 3 \end{bmatrix} + \begin{bmatrix} 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 4 \\ 3 \end{bmatrix}$$

and an example of a matrix 'multiplication' is

$$\begin{bmatrix} 5 & 3 \\ 1 & 3 \end{bmatrix} + \begin{bmatrix} 2 & 4 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 4 & 3 \\ 3 & 3 \end{bmatrix}.$$

We denote by $\mathcal{F}^{J^2}$ the set of $J \times J$ matrices whose entries are functions or sequences of $\mathcal{F}$, and similarly for $\mathcal{G}^{J^2}$.

The min-plus convolution of a matrix $A \in \mathcal{F}^{J^2}$ by a vector $\vec{z} \in \mathcal{F}^J$ is the vector of $\mathcal{F}^J$ defined by

$$(A \otimes \vec{z})(t) = \inf_{0 \leq s \leq t} \{A(t-s) + \vec{z}(s)\}$$

and whose $J$ coordinates are thus

$$\min_{1 \leq j \leq J} \{a_{ij} \otimes z_j\}(t) = \inf_{0 \leq s \leq t} \min_{1 \leq j \leq J} \{a_{ij}(t-s) + z_j(s)\}.$$

Likewise, $A \otimes B$ is defined by

$$(A \otimes B)(t) = \inf_{0 \leq s \leq t} \{A(t-s) + B(s)\}$$

and has entries $\min_{1 \leq j \leq J}(a_{ij} \otimes b_{jk})$ for $1 \leq i, k \leq J$.

For example, we have

$$\begin{bmatrix} \lambda_r & \infty \\ \infty & \delta_T \end{bmatrix} \otimes \begin{bmatrix} \gamma_{r/2,b} \\ \delta_{2T} \end{bmatrix} = \begin{bmatrix} \lambda_r \wedge \gamma_{r/2,b} \\ \delta_{3T} \end{bmatrix}$$

and

$$\begin{bmatrix} \lambda_r & \infty \\ \infty & \delta_T \end{bmatrix} \otimes \begin{bmatrix} \gamma_{r/2,b} & \gamma_{r,b} \\ \delta_{2T} & \lambda_r \end{bmatrix} = \begin{bmatrix} \lambda_r \wedge \gamma_{r/2,b} & \lambda_r \\ \delta_{3T} & \beta_{r,T} \end{bmatrix}.$$

Finally, we will also need to extend the set of wide-sense increasing functions $\mathcal{G}$ to include non decreasing functions of two arguments. We adopt the following definition (a slightly different definition can be found in [11]).

**Definition 4.1.1 (Bivariate wide-sense increasing functions).** *We denote by $\tilde{\mathcal{G}}$ the set of bivariate functions (or sequences) such that for all $s' \le s$ and any $t \le t'$*

$$\begin{aligned} f(t,s) &\le f(t,s') \\ f(t,s) &\le f(t',s). \end{aligned}$$

*We call such functions bivariate wide-sense increasing functions.*

In the multi-dimensional case, we denote by $\tilde{\mathcal{G}}^J$ the set of $J \times J$ matrices whose entries are wide-sense increasing bivariate functions. A matrix of $A(t) \in \mathcal{F}^{J^2}$ is a particular case of a matrix $H(t,s) \in \tilde{\mathcal{G}}^J$, with $s$ set to a fixed value.

### 4.1.2 Operators

A system is an operator $\Pi$ mapping an input function or sequence $\vec{x}$ onto an output function or sequence $\vec{y} = \Pi(\vec{x})$. We will always assume in this book that $\vec{x}, \vec{y} \in \mathcal{G}^J$, where $J$ is a fixed, finite, positive integer. This means that each of the $J$ coordinates $x_j(t)$, $y_j(t)$, $1 \le j \le J$, is a wide-sense increasing function (or sequence) of $t$.

It is important to mention that Min-plus system theory applies to more general operators, taking $\mathbb{R}^J$ to $\mathbb{R}^J$, where neither the input nor the output functions are required to be wide-sense increasing. This requires minor modifications in the definitions and properties established in this chapter, see [26] for the theory described in a more general setting. In this book, to avoid the unnecessary overhead of new notations and definitions, we decided to expose min-plus system theory for operators taking $\mathcal{G}^J$ to $\mathcal{G}^J$.

Most often, the only operator whose output may not be in $\mathcal{F}^J$ is deconvolution, but all other operators we need will take $\mathcal{F}^J$ to $\mathcal{F}^J$.

Most of the time, the dimension of the input and output is $J = 1$, and the operator takes $\mathcal{F}$ to $\mathcal{F}$. We will speak of a *scalar* operator. In this case, we will drop the arrow on the input and output, and write $y = \Pi(x)$ instead.

We write $\Pi_1 \le \Pi_2$ with the meaning that $\Pi_1(\vec{x}) \le \Pi_2(\vec{x})$ for all $\vec{x}$, which in turn has the meaning that $\Pi_1(\vec{x})(t) \le \Pi_2(\vec{x})(t)$ for all $t$.

For a set of operators $\Pi_s$, indexed by $s$ in some set $S$, we call $\inf_{s \in S} \Pi_s$ the operator defined by $[\inf_{s\in S} \Pi_s](x(t)) = \inf_{s \in S}[\Pi_s(x(t))]$. For $S = \{1,2\}$ we denote it with $\Pi_1 \wedge \Pi_2$.

We also denote by $\circ$ the composition of two operators:

$$(\Pi_1 \circ \Pi_2)(\vec{x}) = \Pi_1(\Pi_2(\vec{x})).$$

We leave it to the alert reader to check that $\inf_{s\in S} \Pi_s$ and $\Pi_1 \circ \Pi_2$ do map functions in $\mathcal{G}^J$ to functions in $\mathcal{G}^J$.

### 4.1.3 A Catalog of Operators

Let us mention a few examples of scalar operators of particular interest. The first two have already been studied in detail in Chapter 3, whereas the third was introduced in Section 1.7. The fact that these operators map $\mathcal{G}^J$ into $\mathcal{G}^J$ follows from Chapter 3.

**Definition 4.1.2 (Min-plus convolution $\mathcal{C}_\sigma$).**

$$\begin{array}{rcrcl} \mathcal{C}_\sigma & : & \mathcal{F} & \rightarrow & \mathcal{F} \\ & & x(t) & \rightarrow & y(t) = \mathcal{C}_\sigma(x)(t) = (\sigma \otimes x)(t) = \inf_{0 \le s \le t} \left\{ \sigma(t-s) + x(s) \right\}, \end{array}$$

*for some $\sigma \in \mathcal{F}$.*

**Definition 4.1.3 (Min-plus deconvolution $\mathcal{D}_\sigma$).**

$$\begin{array}{rcrcl} \mathcal{D}_\sigma & : & \mathcal{F} & \rightarrow & \mathcal{G} \\ & & x(t) & \rightarrow & y(t) = \mathcal{D}_\sigma(x)(t) = (x \oslash \sigma)(t) = \sup_{u \ge 0} \left\{ x(t+u) - \sigma(u) \right\}, \end{array}$$

*for some $\sigma \in \mathcal{F}$.*

Note that Min-plus deconvolution produces an output that does not always belong to $\mathcal{F}$.

**Definition 4.1.4 (Packetization $\mathcal{P}_L$).**

$$\begin{array}{rcrcl} \mathcal{P}_L & : & \mathcal{F} & \rightarrow & \mathcal{F} \\ & & x(t) & \rightarrow & y(t) = \mathcal{P}_L(x)(t) = P^L(x(t)) = \inf_{i \in \mathbb{N}} \left\{ L(i) 1_{L(i+1) > x} \right\}, \end{array}$$

*for some wide-sense increasing sequence $L$ (defined by Definition 1.7.1).*

We will also need later on the following operator, whose name will be justified later in this chapter.

**Definition 4.1.5 (Linear idempotent operator $h_\sigma$).**

$$\begin{array}{rcrcl} h_\sigma & : & \mathcal{F} & \rightarrow & \mathcal{F} \\ & & x(t) & \rightarrow & y(t) = h_\sigma(x)(t) = \inf_{0 \le s \le t} \left\{ \sigma(t) - \sigma(s) + x(s) \right\}, \end{array}$$

*for some $\sigma \in \mathcal{F}$.*

The extension of the scalar operators to the vector case is straightforward. The vector extension of the convolution is for instance:

**Definition 4.1.6 (Vector min-plus convolution $\mathcal{C}_\Sigma$).**

$$\begin{array}{rcrcl} \mathcal{C}_\Sigma & : & \mathcal{F}^J & \rightarrow & \mathcal{F}^J \\ & & \vec{x}(t) & \rightarrow & \vec{y}(t) = \mathcal{C}_\Sigma(\vec{x})(t) = (\Sigma \otimes \vec{x})(t) = \inf_{0 \le s \le t} \left\{ \Sigma(t-s) + \vec{x}(s) \right\}, \end{array}$$

*for some $\Sigma \in \mathcal{F}^{J^2}$.*

If the $(i, j)$th entry of $\Sigma$ is $\sigma_{ij}$, the $i$th component of $\vec{y}(t)$ reads therefore

$$y_i(t) = \inf_{0 \le s \le t} \min_{1 \le j \le J} \{\sigma_{ij}(t-s) + x_j(s)\}$$

Let us conclude with the shift operator, which we directly introduce in the vector setting:

**Definition 4.1.7 (Shift operator $\mathcal{S}_T$).**

$$\begin{array}{rcccl} \mathcal{S}_T & : & \mathcal{G}^J & \rightarrow & \mathcal{G}^J \\ & & \vec{x}(t) & \rightarrow & \vec{y}(t) = \mathcal{S}_T(\vec{x})(t) = \vec{x}(t-T), \end{array}$$

*for some $T \in \mathbb{R}$.*

Let us remark that $\mathcal{S}_0$ is the identity operator: $\mathcal{S}_0(\vec{x}) = \vec{x}$.

### 4.1.4 Upper and Lower Semi-continuous Operators

We now study a number of properties of min-plus linear operators. We begin with that of upper-semi continuity.

**Definition 4.1.8 (Upper semi-continuous operator).** *Operator $\Pi$ is upper semi-continuous if for any (finite or infinite) set of functions or sequences $\{\vec{x}_n\}$, $\vec{x}_n \in \mathcal{G}^J$,*

$$\Pi\left(\inf_n \{\vec{x}_n\}\right) = \inf_n \{\Pi(\vec{x}_n)\}. \tag{4.1}$$

We can check that $\mathcal{C}_\sigma$, $\mathcal{C}_\Sigma$, $h_\sigma$ and $\mathcal{S}_T$ are upper semi-continuous. For example, for $\mathcal{C}_\Sigma$, we check indeed that

$$\begin{aligned} \mathcal{C}_\Sigma\left(\inf_n \{\vec{x}_n\}\right)(t) &= \inf_{0 \le s \le t} \left\{\Sigma(t-s) + \inf_n \{\vec{x}_n(s)\}\right\} \\ &= \inf_{0 \le s \le t} \inf_n \{\Sigma(t-s) + \vec{x}_n(s)\} \\ &= \inf_n \inf_{0 \le s \le t} \{\Sigma(t-s) + \vec{x}_n(s)\} \\ &= \inf_n \{\mathcal{C}_\Sigma(\vec{x}_n)(t)\}. \end{aligned}$$

Likewise, noting that $L(i+1) \le \inf_{n \in \mathbb{N}} \{x_n\}$ if and only if $L(i+1) \le x_n$ for all $n \in \mathbb{N}$, we get that

$$1_{\{L(i+1) \le \inf_{n \in \mathbb{N}} \{x_n\}\}} = \inf_{n \in \mathbb{N}} 1_{\{L(i+1) \le x_n\}}$$

and thus we get that $\mathcal{P}_L$ is upper semi-continuous:

$$\begin{aligned} \mathcal{P}_L\left(\inf_n \{x_n\}\right) &= \inf_{i \in \mathbb{N}} \left\{L(i) 1_{\{L(i+1) > \inf_n \{x_n\}\}}\right\} \\ &= \sup_{i \in \mathbb{N}} \left\{L(i) 1_{\{L(i+1) \le \inf_n \{x_n\}\}}\right\} \end{aligned}$$

$$\begin{aligned}
&= \sup_{i\in\mathbb{N}}\left\{\inf_n\left\{L(i)1_{\{L(i+1)\le x_n\}}\right\}\right\}\\
&= \inf_{i\in\mathbb{N}}\left\{\inf_n\left\{L(i)1_{\{L(i+1)> x_n\}}\right\}\right\}\\
&= \inf_{n}\left\{\inf_{i\in\mathbb{N}}\left\{L(i)1_{\{L(i+1)> x_n\}}\right\}\right\}\\
&= \inf_n\left\{\mathcal{P}_L(x_n)\right\}.
\end{aligned}$$

On the other hand, $\mathcal{D}_\sigma$ is not upper semi-continuous, because its application to an inf would involve the three operations $\sup$, $\inf$ and $+$, which do not commute, as we have seen at the end of the previous chapter.

It is easy to show that if $\Pi_1$ and $\Pi_2$ are upper semi-continuous, so are $\Pi_1 \wedge \Pi_2$ and $\Pi_1 \circ \Pi_2$.

The dual definition of upper semi-continuity is that of lower semi-continuity, which is defined as follows.

**Definition 4.1.9 (Lower semi-continuous operator).** *Operator $\Pi$ is lower semi-continuous if for any (finite or infinite) set of functions or sequences $\{\vec{x}_n\}$, $\vec{x}_n \in \mathcal{G}^J$,*

$$\Pi\left(\sup_n\{\vec{x}_n\}\right) = \sup_n\left\{\Pi(\vec{x}_n)\right\}. \tag{4.2}$$

It is easy to check that $\mathcal{D}_\sigma$ is lower semi-continuous, unlike other operators, except $\mathcal{S}_T$ which is also lower semi-continuous.

### 4.1.5 Isotone Operators

**Definition 4.1.10 (Isotone operator).** *Operator $\Pi$ is isotone if $\vec{x}_1 \le \vec{x}_2$ always implies $\Pi(\vec{x}_1) \le \Pi(\vec{x}_2)$.*

All upper semi-continuous operators are isotone. Indeed, if $\vec{x}_1 \le \vec{x}_2$, then $\vec{x}_1 \wedge \vec{x}_2 = \vec{x}_1$ and since $\Pi$ is upper semi-continuous,

$$\Pi(\vec{x}_1) = \Pi(\vec{x}_1 \wedge \vec{x}_2) = \Pi(\vec{x}_1) \wedge \Pi(\vec{x}_2) \le \Pi(\vec{x}_2).$$

Likewise, all lower semi-continuous operators are isotone. Indeed, if $\vec{x}_1 \le \vec{x}_2$, then $\vec{x}_1 \vee \vec{x}_2 = \vec{x}_2$ and since $\Pi$ is lower semi-continuous,

$$\Pi(\vec{x}_1) \le \Pi(\vec{x}_1) \vee \Pi(\vec{x}_2) = \Pi(\vec{x}_1 \vee \vec{x}_2) = \Pi(\vec{x}_2).$$

### 4.1.6 Linear Operators

In classical system theory on $(\mathbb{R}, +, \times)$, a system $\Pi$ is linear if its output to a linear combination of inputs is the linear combination of the outputs to each particular input. In other words, $\Pi$ is linear if for any (finite or infinite) set of inputs $\{x_i\}$, and for any constant $k \in \mathbb{R}$,

$$\Pi\left(\sum_i x_i\right) = \sum_i \Pi(x_i)$$

and for any input $x$ and any constant $k \in \mathbb{R}$,

$$\Pi\,(k\,\cdot\,x) = k\,\cdot\,\Pi(x).$$

The extension to min-plus system theory is straightforward. The first property being replaced by that of upper semi-continuity, a min-plus linear operator is thus defined as an upper semi-continuous operator that has the following property ("multiplication" by a constant):

**Definition 4.1.11 (Min-plus linear operator).** *Operator $\Pi$ is min-plus linear if it is upper semi-continuous and if for any $\vec{x} \in \mathcal{G}^J$ and for any $k \geq 0$,*

$$\Pi\,(\vec{x} + k) = \Pi\,(\vec{x}) + k. \tag{4.3}$$

One can easily check that $\mathcal{C}_\sigma$, $\mathcal{C}_\Sigma$, $h_\sigma$ and $\mathcal{S}_T$ are min-plus linear, unlike $\mathcal{D}_\sigma$ and $\mathcal{P}_L$. $\mathcal{D}_\sigma$ is not linear because it is not upper semi-continuous, and $\mathcal{P}_L$ is not linear because it fails to verify (4.3).

In classical linear theory, a linear system is represented by its impulse response $h(t,s)$, which is defined as the output of the system when the input is the Dirac function. The output of such a system can be expressed as

$$\Pi(x)(t) = \int_{-\infty}^{\infty} h(t,s)x(s)ds$$

Its straightforward extension in Min-plus system theory is provided by the following theorem [26]. To prove this theorem in the vector case, we need first to extend the burst delay function introduced in Definition 3.1.2, to allow negative values of the delay, namely, the value $T$ in

$$\delta_T(t) = \begin{cases} 0 & \text{if } t \leq T \\ \infty & \text{if } t > T, \end{cases}$$

is now taking values in $\mathbb{R}$. We also introduce the following matrix $D_T \in \mathcal{G}^J \times \mathcal{G}^J$.

**Definition 4.1.12 (Shift matrix).** *The shift matrix is defined by*

$$D_T(t) = \begin{bmatrix} \delta_T(t) & \infty & \infty & \cdots & \infty \\ \infty & \delta_T(t) & \infty & & \\ \infty & \infty & \delta_T(t) & \ddots & \vdots \\ \vdots & \vdots & & \ddots & \infty \\ \infty & \cdots & & \infty & \delta_T(t) \end{bmatrix}$$

*for some $T \in \mathbb{R}$.*

**Theorem 4.1.1 (Min-plus impulse response).** *$\Pi$ is a min-plus linear operator if and only if there is a unique matrix $H \in \tilde{\mathcal{G}}^J$ (called the* impulse response*), such that for any $\vec{x} \in \mathcal{G}^J$ and any $t \in \mathbb{R}$,*

$$\Pi(\vec{x})(t) = \inf_{s \in \mathbb{R}} \{H(t,s) + \vec{x}(s)\}. \tag{4.4}$$

**Proof:** If (4.4) holds, one immediately sees that $\Pi$ is upper semi-continuous and verifies (4.3), and therefore is min-plus linear. $\Pi$ maps $\mathcal{G}^J$ to $\mathcal{G}^J$ because $H \in \tilde{\mathcal{G}}^J$.

Suppose next that $\Pi$ is min-plus linear, and let us prove that there is a unique matrix $H(t,s) \in \tilde{\mathcal{G}}^J$ such that (4.4) holds.

Let us first note that $D_s(t) + \vec{x}(s) = \vec{x}(s)$ for any $s \geq t$. Since $\vec{x} \in \mathcal{G}^J$, we have

$$\inf_{s \geq t} \{D_s(t) + \vec{x}(s)\} = \inf_{s \geq t} \{\vec{x}(s)\} = \vec{x}(t).$$

On the other hand, all entries of $D_s(t)$ are infinite for $s < t$. We have therefore that

$$\inf_{s < t} \{D_s(t) + \vec{x}(s)\} = \infty$$

We can combine these two expressions as

$$\vec{x}(t) = \inf_{s \in \mathbb{R}} \{D_s(t) + \vec{x}(s)\},$$

or, dropping explicit dependence on $t$,

$$\vec{x} = \inf_{s \in \mathbb{R}} \{D_s + \vec{x}(s)\}.$$

Let $\vec{d}_{s,j}$ denote the $j$th column of $D_s$:

$$\vec{d}_{s,j} = \begin{bmatrix} \infty \\ \vdots \\ \infty \\ \delta_s \\ \infty \\ \vdots \\ \infty \end{bmatrix}$$

where $\delta_s$ is located at the $j$th position in this vector. Using repeatedly the fact $\Pi$ is min-plus linear, we get that

$$\begin{aligned} \Pi(\vec{x}) &= \Pi\left(\inf_{s \in \mathbb{R}} \{D_s + \vec{x}(s)\}\right) \\ &= \inf_{s \in \mathbb{R}} \{\Pi\left(D_s + \vec{x}(s)\right)\} \\ &= \inf_{s \in \mathbb{R}} \left\{\Pi\left(\min_{1 \leq j \leq J} \left\{\vec{d}_{s,j} + x_j(s)\right\}\right)\right\} \end{aligned}$$

$$
\begin{aligned}
&= \inf_{s\in\mathbb{R}}\left\{\min_{1\le j\le J}\left\{\Pi\left(\vec{d}_{s,j}+x_j(s)\right)\right\}\right\}\\
&= \inf_{s\in\mathbb{R}}\left\{\min_{1\le j\le J}\left\{\Pi\left(\vec{d}_{s,j}\right)+x_j(s)\right\}\right\}.
\end{aligned}
$$

Defining

$$H(t,s)=\left[\vec{h}_1(t,s)\quad\ldots\quad\vec{h}_j(t,s)\quad\ldots\quad\vec{h}_J(t,s)\right] \tag{4.5}$$

where

$$\vec{h}_j(t,s)=\Pi\left(\vec{d}_{s,j}\right)(t) \tag{4.6}$$

for all $t\in\mathbb{R}$, we obtain therefore that

$$\Pi(\vec{x})(t)=\inf_{s\in\mathbb{R}}\left\{\min_{1\le j\le J}\left\{\vec{h}_j(t,s)+x_j(s)\right\}\right\}=\inf_{s\in\mathbb{R}}\left\{H(t,s)+\vec{x}(s)\right\}.$$

We still have to check that $H(t,s)\in\tilde{\mathcal{G}}^J$. Since for any fixed $s$, $\Pi\left(\vec{d}_{s,j}\right)\in\mathcal{G}^J$, we have that for any $t\le t'$

$$\vec{h}_j(t,s)=\Pi\left(\vec{d}_{s,j}\right)(t)\le\Pi\left(\vec{d}_{s,j}\right)(t')=\vec{h}_j(t',s),$$

hence $H(t,s)\le H(t',s)$. On the other hand, if $s'\le s$, one easily check that $\vec{d}_{s,j}\le\vec{d}_{s',j}$. Therefore, since $\Pi$ is isotone (because it is linear and thus upper semi-continuous),

$$\vec{h}_j(t,s)=\Pi\left(\vec{d}_{s,j}\right)(t)\le\Pi\left(\vec{d}_{s',j}\right)(t)=\vec{h}_j(t,s')$$

and therefore $H(t,s)\le H(t,s')$ for any $s\ge s'$. This shows that $H(t,s)\in\tilde{\mathcal{G}}^J$.

To prove uniqueness, suppose that there is another matrix $H'\in\tilde{\mathcal{G}}^J$ that satisfies (4.4), and let $\vec{h'}_j$ denote its $j$th column. Then for any $u\in\mathbb{R}$ and any $1\le j\le J$, taking $\vec{x}=\vec{d}_{u,j}$ as the input, we get from (4.6) that for $t\in\mathbb{R}$

$$
\begin{aligned}
\vec{h}_j(t,u) &= \Pi\left(\vec{d}_{u,j}\right)(t)=\inf_{s\in\mathbb{R}}\left\{H'(t,s)+\vec{d}_{u,j}(s)\right\}\\
&= \inf_{s\in\mathbb{R}}\left\{\vec{h'}_j(t,s)+\delta_u(s)\right\}=\inf_{s\le u}\left\{\vec{h'}_j(t,s)\right\}=\vec{h'}_j(t,u).
\end{aligned}
$$

Therefore $H'=H$. □

We will denote a general min-plus linear operator whose impulse response is $H$ by $\mathcal{L}_H$. In other words, we have that

$$\mathcal{L}_H(\vec{x})(t)=\inf_{s\in\mathbb{R}}\left\{H(t,s)+\vec{x}(s)\right\}.$$

One can compute that the impulse response corresponding to $\mathcal{C}_\Sigma$ is

$$H(t,s)=\begin{cases}\Sigma(t-s) & \text{if } s\le t\\ \Sigma(0) & \text{if } s>t\end{cases},$$

to $h_\sigma$ is

$$H(t,s) = \begin{cases} \sigma(t) - \sigma(s) & \text{if } s \le t \\ 0 & \text{if } s > t \end{cases},$$

and to $\mathcal{S}_T$ is

$$H(t,s) = D_T(t-s).$$

In fact the introduction of the shift matrix allows us to write the shift operator as a min-plus convolution: $\mathcal{S}_T = \mathcal{C}_{D_T}$ if $T \ge 0$.

Let us now compute the impulse response of the compostion of two min-plus linear operators.

**Theorem 4.1.2 (Composition of min-plus linear operators).** *Let $\mathcal{L}_H$ and $\mathcal{L}_{H'}$ be two min-plus linear operators. Then their composition $\mathcal{L}_H \circ \mathcal{L}_{H'}$ is also min-plus linear, and its impulse repsonse denoted by $H \circ H'$ is given by*

$$(H \circ H')(t,s) = \inf_{u \in \mathbb{R}} \left\{ H(t,u) + H'(u,s) \right\}.$$

**Proof:** The composition $\mathcal{L}_H \circ \mathcal{L}_{H'}$ applied to some $\vec{x} \in \mathcal{G}^J$ is

$$\begin{aligned} \mathcal{L}_H(\mathcal{L}_{H'}(\vec{x}))(t) &= \inf_u \left\{ H(t,u) + \inf_s \left\{ H'(u,s) + \vec{x}(s) \right\} \right\} \\ &= \inf_u \inf_s \left\{ H(t,u) + H'(u,s) + \vec{x}(s) \right\} \\ &= \inf_s \left\{ \inf_u \left\{ H(t,s) + H'(u,s) \right\} + \vec{x}(s) \right\}. \end{aligned}$$

□

We can therefore write

$$\mathcal{L}_H \circ \mathcal{L}_{H'} = \mathcal{L}_{H \circ H'}.$$

Likewise, one easily shows that

$$\mathcal{L}_H \wedge \mathcal{L}_{H'} = \mathcal{L}_{H \wedge H'}.$$

Finally, let us mention the dual definition of a max-plus linear operator.

**Definition 4.1.13 (Max-plus linear operator).** *Operator $\Pi$ is max-plus linear if it is lower semi-continuous and if for any $\vec{x} \in \mathcal{G}^J$ and for any $k \ge 0$,*

$$\Pi(\vec{x} + k) = \Pi(\vec{x}) + k. \tag{4.7}$$

Max-plus linear operators can also be represented by their impulse response.

**Theorem 4.1.3 (Max-plus impulse response).** *$\Pi$ is a max-plus linear operator if and only if there is a unique matrix $H \in \tilde{\mathcal{G}}^J$ (called the* impulse response*), such that for any $\vec{x} \in \mathcal{G}^J$ and any $t \in \mathbb{R}$,*

$$\Pi(\vec{x})(t) = \sup_{s \in \mathbb{R}} \left\{ H(t,s) + \vec{x}(s) \right\}. \tag{4.8}$$

One can easily check that $\mathcal{D}_\sigma$ and $\mathcal{S}_T$ are max-plus linear, unlike $\mathcal{C}_\Sigma$, $h_\sigma$ and $\mathcal{P}_L$.

For example, $\mathcal{D}_\sigma(x)(t)$ can be written as

$$\mathcal{D}_\sigma(x)(t) = \sup_{u\geq 0}\{x(t+u)-\sigma(u)\} = \sup_{s\geq t}\{x(s)-\sigma(s-t)\} = \sup_{s\in\mathbb{R}}\{x(s)-\sigma(s-t)\}$$

which has the form (4.8) if $H(t,s) = -\sigma(s-t)$.

Likewise, $\mathcal{S}_T(x)(t)$ can be written as

$$\mathcal{S}_T\left(\vec{x}\right)(t) = \vec{x}(t-T) = \sup_{s\in\mathbb{R}}\{\vec{x}(s) - D_{-T}(s-t)\}$$

which has the form (4.8) if $H(t,s) = -D_{-T}(s-t)$.

### 4.1.7 Causal Operators

A system is causal if its output at time $t$ only depends on its input before time $t$.

**Definition 4.1.14 (Causal operator).** *Operator $\Pi$ is causal if for any $t$, $\vec{x}_1(s) = \vec{x}_2(s)$ for all $s \leq t$ always implies $\Pi(\vec{x}_1)(t) = \Pi(\vec{x}_2)(t)$.*

**Theorem 4.1.4 (Min-plus causal linear operator).** *A min-plus linear system with impulse response $H$ is causal if $H(t,s) = H(t,t)$ for $s > t$.*

**Proof:** If $H(t,s) = 0$ for $s > t$ and if $\vec{x}_1(s) = \vec{x}_2(s)$ for all $s \leq t$ then since $\vec{x}_1, \vec{x}_2 \in \mathcal{G}^J$,

$$\begin{aligned}
\mathcal{L}_H(\vec{x}_1)(t) &= \inf_{s\in\mathbb{R}}\{H(t,s)+\vec{x}_1(s)\} \\
&= \inf_{s\leq t}\{H(t,s)+\vec{x}_1(s)\} \wedge \inf_{s>t}\{H(t,s)+\vec{x}_1(s)\} \\
&= \inf_{s\leq t}\{H(t,s)+\vec{x}_1(s)\} \wedge \inf_{s>t}\{H(t,t)+\vec{x}_1(s)\} \\
&= \inf_{s\leq t}\{H(t,s)+\vec{x}_1(s)\} \\
&= \inf_{s\leq t}\{H(t,s)+\vec{x}_2(s)\} \\
&= \inf_{s\leq t}\{H(t,s)+\vec{x}_2(s)\} \wedge \inf_{s>t}\{H(t,t)+\vec{x}_2(s)\} \\
&= \inf_{s\leq t}\{H(t,s)+\vec{x}_2(s)\} \wedge \inf_{s>t}\{H(t,s)+\vec{x}_2(s)\} \\
&= \inf_{s\in\mathbb{R}}\{H(t,s)+\vec{x}_2(s)\} = \mathcal{L}_H(\vec{x}_2)(t).
\end{aligned}$$

□

$\mathcal{C}_\sigma$, $\mathcal{C}_\Sigma$, $h_\sigma$ and $\mathcal{P}_L$ are causal. $\mathcal{S}_T$ is causal if and only if $T \geq 0$. $\mathcal{D}_\sigma$ is not causal. Indeed if $\vec{x}_1(s) = \vec{x}_2(s)$ for all $s \leq t$, but that $\vec{x}_1(s) \neq \vec{x}_2(s)$ for all $s > t$, then

$$\begin{aligned}\mathcal{D}_\sigma(\vec{x}_1)(t) &= \sup_{u\geq 0}\{\vec{x}_1(t+u)-\sigma(u)\}\\ &\neq \sup_{u\geq 0}\{\vec{x}_2(t+u)-\sigma(u)\}\\ &= \mathcal{D}_\sigma(\vec{x}_1)(t)\end{aligned}$$

### 4.1.8 Shift-invariant Operators

A system is shift-invariant, or time-invariant, if a shift of the input of $T$ time units yields a shift of the output of $T$ time units too.

**Definition 4.1.15 (Shift-invariant operator).** *Operator $\Pi$ is shift-invariant if it commutes with all shift operators, i.e. if for any $\vec{x}\in\mathcal{G}$ and for any $T\in\mathbb{R}$*

$$\Pi(\mathcal{S}_T(\vec{x})) = \mathcal{S}_T(\Pi(\vec{x})).$$

**Theorem 4.1.5 (Shift-invariant min-plus linear operator).** *Let $\mathcal{L}_H$ and $\mathcal{L}_{H'}$ be two min-plus linear, shift-invariant operators.*

*(i) A min-plus linear operator $\mathcal{L}_H$ is shift-invariant if and only if its impulse response $H(t,s)$ depends only on the difference $(t-s)$.*

*(ii) Two min-plus linear, shift-invariant operators $\mathcal{L}_H$ and $\mathcal{L}_{H'}$ commute. If they are also causal, the impulse response of their composition is*

$$(H\circ H')(t,s) = \inf_{0\leq u\leq t-s}\{H(t-s-u)+H'(u)\} = (H\otimes H')(t-s).$$

**Proof:** (i) Let $\vec{h}_j(t,s)$ and $\vec{d}_{s,j}(t)$ denote (respectively) the $j$th column of $H(t,s)$ and of $D_s(t)$. Note that $\vec{d}_{s,j}(t) = \mathcal{S}_s(\vec{d}_{0,j})(t)$. Then (4.6) yields that

$$\begin{aligned}\vec{h}_j(t,s) &= \Pi\left(\vec{d}_{s,j}\right)(t) = \Pi\left(\mathcal{S}_s(\vec{d}_{0,j})\right)(t)\\ &= \mathcal{S}_s\left(\Pi(\vec{d}_{0,j})\right)(t) = \left(\Pi(\vec{d}_{0,j})\right)(t-s) = \vec{h}_j(t-s,0)\end{aligned}$$

Therefore $H(t,s)$ can be written as a function of a single variable $H(t-s)$.

(ii) Because of Theorem 4.1.2, the impulse response of $\mathcal{L}_H\circ\mathcal{L}_{H'}$ is

$$(H\circ H')(t,s) = \inf_u\{H(t,u)+H'(u,s)\}.$$

Since $H(t,u) = H(t-u)$ and $H'(u,s) = H'(u-s)$, and setting $v = u-s$, the latter can be written as

$$(H\circ H')(t,s) = \inf_u\{H(t-u)+H'(u-s)\} = \inf_v\{H(t-s-v)+H'(v)\}.$$

Similarly, the impulse response of $\mathcal{L}_{H'}\circ\mathcal{L}_H$ can be written as

$$(H'\circ H)(t,s) = \inf_u\{H'(t-u)+H(u-s)\} = \inf_v\{H(v)+H'(t-s-v)\}$$

where this time we have set $v = t - u$. Both impulse responses are identical, which shows that the two operators commute.

If they are causal, then their impulse response is infinite for $t > s$ and the two previous relations become

$$(H \circ H')(t,s) = (H' \circ H)(t,s) = \inf_{0 \le v \le t} \{H(t-s-v) + H'(v)\} = (H \otimes H')(t-s).$$

□

Min-plus convolution $\mathcal{C}_\Sigma$ (including of course $\mathcal{C}_\sigma$ and $\mathcal{S}_T$) is therefore shift-invariant. In fact, it follows from this theorem that the only min-plus linear, causal and shift-invariant operator is min-plus convolution. Therefore $h_\sigma$ is not shift-invariant.

Min-plus deconvolution is shift-invariant, as

$$\begin{aligned} \mathcal{D}_\sigma(\mathcal{S}_T(x))(t) &= \sup_{u \ge 0}\{\mathcal{S}_T(x)(t+u) - \sigma(u)\} = \sup_{u \ge 0}\{x(t+u-T) - \sigma(u)\} \\ &= (x \oslash \sigma)(t-T) = \mathcal{D}_\sigma(x)(t-T) = \mathcal{S}_T\left(\mathcal{D}_\sigma\right)(x)(t). \end{aligned}$$

Finally let us mention that $\mathcal{P}_L$ is not shift-invariant.

### 4.1.9 Idempotent Operators

An idempotent operator is an operator whose composition with itself produces the same operator.

**Definition 4.1.16 (Idempotent operator).** *Operator $\Pi$ is idempotent if its self-composition is $\Pi$, i.e. if*

$$\Pi \circ \Pi = \Pi.$$

We can easily check that $h_\sigma$ and $\mathcal{P}_L$ are idempotent. If $\sigma$ is sub-additive, with $\sigma(0) = 0$, then $\mathcal{C}_\sigma \circ \mathcal{C}_\sigma = \mathcal{C}_\sigma$, which shows that in this case, $\mathcal{C}_\sigma$ is idempotent too. The same applies to $\mathcal{D}_\sigma$.

## 4.2 Closure of an Operator

By repeatedly composing a min-plus operator with itself, we obtain the closure of this operator. The formal definition is as follows.

**Definition 4.2.1 (Sub-additive closure of an operator).** *Let $\Pi$ be a min-plus operator taking $\mathcal{G}^J \to \mathcal{G}^J$. Denote $\Pi^{(n)}$ the operator obtained by composing $\Pi$ $(n-1)$ times with itself. By convention, $\Pi^{(0)} = \mathcal{S}_0 = \mathcal{C}_{D_0}$, so $\Pi^{(1)} = \Pi$, $\Pi^{(2)} = \Pi \circ \Pi$, etc. Then the sub-additive closure of $\Pi$, denoted by $\overline{\Pi}$, is defined by*

$$\overline{\Pi} = \mathcal{S}_0 \wedge \Pi \wedge (\Pi \circ \Pi) \wedge (\Pi \circ \Pi \circ \Pi) \wedge \ldots = \inf_{n \ge 0} \left\{ \Pi^{(n)} \right\}. \tag{4.9}$$

In other words,

$$\overline{\Pi}(\vec{x}) = \vec{x} \wedge \Pi(\vec{x}) \wedge \Pi(\Pi(\vec{x})) \wedge \ldots$$

It is immediate to check that $\overline{\Pi}$ does map functions in $\mathcal{G}^J$ to functions in $\mathcal{G}^J$.

The next theorem provides the impulse response of the sub-additive closure of a min-plus linear operator. It follows immediately from applying recursively Theorem 4.1.2.

**Theorem 4.2.1 (Sub-additive closure of a linear operator).** *The impulse response of $\overline{\mathcal{L}}_H$ is*

$$\overline{H}(t,s) = \inf_{n\in\mathbb{N}} \inf_{u_n,\ldots,u_2,u_1} \{H(t,u_1) + H(u_1,u_2) + \ldots + H(u_n,s)\}. \tag{4.10}$$

*and $\overline{\mathcal{L}}_H = \mathcal{L}_{\overline{H}}$.*

For a min-plus linear, shift-invariant and causal operator, (4.10) becomes

$$\begin{aligned}
&\overline{H}(t-s)\\
&\quad = \inf_{n\in\mathbb{N}} \inf_{s\le u_n\le\ldots\le u_2\le u_1\le t} \{H(t-u_1) + H(u_1-u_2) + \ldots + H(u_n-s)\}\\
&\quad = \inf_{n\in\mathbb{N}} \inf_{0\le v_n\le\ldots\le v_2\le v_1\le t-s} \{H(t-s-v_1) + H(v_1-v_2) + \ldots + H(v_n)\}\\
&\quad = \inf_{n\in\mathbb{N}} \{H^{(n)}\}(t-s) \qquad\qquad (4.11)
\end{aligned}$$

where $H^{(n)} = H \otimes H \otimes \ldots \otimes H$ ($n$ times, $n \ge 1$) and $H^{(0)} = S_0$.

In particular, if all entries $\sigma_{ij}(t)$ of $\Sigma(t)$ are sub-additive functions, we find that

$$\overline{\mathcal{C}}_\Sigma = \mathcal{C}_\Sigma.$$

In the scalar case, the closure of the min-plus convolution operator $\mathcal{C}_\sigma$ reduces to the min-plus convolution of the sub-additive closure of $\sigma$:

$$\overline{\mathcal{C}}_\sigma = \mathcal{C}_{\overline{\sigma}}.$$

If $\sigma$ is a "good" function (i.e., a sub-additive function with $\sigma(0) = 0$), then $\overline{\mathcal{C}}_\sigma = \mathcal{C}_\sigma$.

The sub-additive closure of the idempotent operators $h_\sigma$ and $\mathcal{P}_L$ are easy to compute too. Indeed, since $h_\sigma(x) \le x$ and $\mathcal{P}_L(x) \le x$,

$$\overline{h_\sigma} = h_\sigma$$

and

$$\overline{\mathcal{P}}_L = \mathcal{P}_L.$$

The following result is easy to prove. We write $\Pi \le \Pi'$ to express that $\Pi(\vec{x}) \le \Pi'(\vec{x})$ for all $\vec{x} \in \mathcal{G}^J$.

**Theorem 4.2.2 (Sub-additive closure of an isotone operator).** *If $\Pi$ and $\Pi'$ are two isotone operators, and $\Pi \le \Pi'$, then $\overline{\Pi} \le \overline{\Pi'}$.*

Finally, let us conclude this section by computing the closure of the minimum between two operators.

**Theorem 4.2.3 (Sub-additive closure of $\Pi_1 \wedge \Pi_2$).** *Let $\Pi_1, \Pi_2$ be two isotone operators taking $\mathcal{G}^J \to \mathcal{G}^J$. Then*

$$\overline{\Pi_1 \wedge \Pi_2} = \overline{(\Pi_1 \wedge \mathcal{S}_0) \circ (\Pi_2 \wedge \mathcal{S}_0)}. \tag{4.12}$$

**Proof:** (i) Since $\mathcal{S}_0$ is the identity operator,

$$\begin{aligned}
\Pi_1 \wedge \Pi_2 &= (\Pi_1 \circ \mathcal{S}_0) \wedge (\mathcal{S}_0 \circ \Pi_2) \\
&\geq ((\Pi_1 \wedge \mathcal{S}_0) \circ \mathcal{S}_0) \wedge (\mathcal{S}_0 \circ (\Pi_2 \wedge \mathcal{S}_0)) \\
&\geq ((\Pi_1 \wedge \mathcal{S}_0) \circ (\Pi_2 \wedge \mathcal{S}_0)) \wedge ((\Pi_1 \wedge \mathcal{S}_0) \circ (\Pi_2 \wedge \mathcal{S}_0)) \\
&= (\Pi_1 \wedge \mathcal{S}_0) \circ (\Pi_2 \wedge \mathcal{S}_0).
\end{aligned}$$

Since $\Pi_1$ and $\Pi_2$ are isotone, so are $\Pi_1 \wedge \Pi_2$ and $(\Pi_1 \wedge \mathcal{S}_0) \circ (\Pi_2 \wedge \mathcal{S}_0)$. Consequently, Theorem 4.2.2 yields that

$$\overline{\Pi_1 \wedge \Pi_2} \geq \overline{(\Pi_1 \wedge \mathcal{S}_0) \circ (\Pi_1 \wedge \mathcal{S}_0)}. \tag{4.13}$$

(ii) Combining the two inequalities

$$\begin{aligned}
\Pi_1 \wedge \mathcal{S}_0 &\geq \Pi_1 \wedge \Pi_2 \wedge \mathcal{S}_0 \\
\Pi_2 \wedge \mathcal{S}_0 &\geq \Pi_1 \wedge \Pi_2 \wedge \mathcal{S}_0
\end{aligned}$$

we get that

$$\overline{(\Pi_1 \wedge \mathcal{S}_0) \circ (\Pi_1 \wedge \mathcal{S}_0)} \geq \overline{(\Pi_1 \wedge \Pi_2 \wedge \mathcal{S}_0) \circ (\Pi_1 \wedge \Pi_2 \wedge \mathcal{S}_0)}. \tag{4.14}$$

Let us show by induction that

$$((\Pi_1 \wedge \Pi_2) \wedge \mathcal{S}_0)^{(n)} = \min_{0 \leq k \leq n} \left\{ (\Pi_1 \wedge \Pi_2)^{(k)} \right\}.$$

Clearly, the claim holds for $n = 0, 1$. Suppose it is true up to some $n \in \mathbb{N}$. Then

$$\begin{aligned}
&((\Pi_1 \wedge \Pi_2) \wedge \mathcal{S}_0)^{(n+1)} \\
&\quad = ((\Pi_1 \wedge \Pi_2) \wedge \mathcal{S}_0) \circ ((\Pi_1 \wedge \Pi_2) \wedge \mathcal{S}_0)^{(n)} \\
&\quad = ((\Pi_1 \wedge \Pi_2) \wedge \mathcal{S}_0) \circ \left( \min_{0 \leq k \leq n} \left\{ (\Pi_1 \wedge \Pi_2)^{(k)} \right\} \right) \\
&\quad = \left( (\Pi_1 \wedge \Pi_2) \circ \min_{0 \leq k \leq n} \left\{ (\Pi_1 \wedge \Pi_2)^{(k)} \right\} \right) \wedge \left( \mathcal{S}_0 \circ \min_{0 \leq k \leq n} \left\{ (\Pi_1 \wedge \Pi_2)^{(k)} \right\} \right) \\
&\quad = \min_{1 \leq k \leq n+1} \left\{ (\Pi_1 \wedge \Pi_2)^{(k)} \right\} \wedge \min_{0 \leq k \leq n} \left\{ (\Pi_1 \wedge \Pi_2)^{(k)} \right\} \\
&\quad = \min_{0 \leq k \leq n+1} \left\{ (\Pi_1 \wedge \Pi_2)^{(k)} \right\}.
\end{aligned}$$

Therefore the claim holds for all $n \in \mathbb{N}$, and

$$\begin{aligned}(((\Pi_1 \wedge \Pi_2) \wedge \mathcal{S}_0) \circ ((\Pi_1 \wedge \Pi_2) \wedge \mathcal{S}_0))^{(n)} &= ((\Pi_1 \wedge \Pi_2) \wedge \mathcal{S}_0)^{(2n)} \\ &= \min_{0 \le k \le 2n} \left\{ (\Pi_1 \wedge \Pi_2)^{(k)} \right\}.\end{aligned}$$

Consequently,

$$\begin{aligned}\overline{(\Pi_1 \wedge \Pi_2 \wedge \mathcal{S}_0) \circ (\Pi_1 \wedge \Pi_2 \wedge \mathcal{S}_0)} &= \inf_{n \in \mathbb{N}} \min_{0 \le k \le 2n} \left\{ (\Pi_1 \wedge \Pi_2)^{(k)} \right\} \\ &= \inf_{k \in \mathbb{N}} \left\{ (\Pi_1 \wedge \Pi_2)^{(k)} \right\} \\ &= \overline{\Pi_1 \wedge \Pi_2}\end{aligned}$$

and combining this result with (4.13) and (4.14), we get (4.12). □

If one of the two operators is an idempotent operator, we can simplify the previous result a bit more. We will use the following corollary in Chapter 9.

**Corollary 4.2.1 (Sub-additive closure of $\Pi_1 \wedge h_M$).** *Let $\Pi_1$ be an isotone operator taking $\mathcal{F} \to \mathcal{F}$, and let $M \in \mathcal{F}$. Then*

$$\overline{\Pi_1 \wedge h_M} = \overline{(h_M \circ \Pi_1)} \circ h_M. \tag{4.15}$$

**Proof:** Theorem 4.2.3 yields that

$$\overline{\Pi_1 \wedge h_M} = \overline{(\Pi_1 \wedge \mathcal{S}_0) \circ h_M} \tag{4.16}$$

because $h_M \le \mathcal{S}_0$. The right hand side of (4.16) is the inf over all integers $n$ of

$$(\{\Pi_1 \wedge \mathcal{S}_0\} \circ h_M)^{(n)}$$

which we can expand as

$$\{\Pi_1 \wedge \mathcal{S}_0\} \circ h_M \circ \{\Pi_1 \wedge \mathcal{S}_0\} \circ h_M \circ \ldots \circ \{\Pi_1 \wedge \mathcal{S}_0\} \circ h_M.$$

Since

$$\begin{aligned}h_M \circ \{\Pi_1 \wedge \mathcal{S}_0\} \circ h_M &= \{h_M \circ \Pi_1 \circ h_M\} \wedge h_M \\ &= (\{h_M \circ \Pi_1\} \wedge \mathcal{S}_0) \circ h_M \\ &= \min_{0 \le q \le 1} \left\{ (h_M \circ \Pi_1)^{(q)} \right\} \circ h_M,\end{aligned}$$

the previous expression is equal to

$$\min_{0 \le q \le n} \left\{ (h_M \circ \Pi_1)^{(q)} \right\} \circ h_M.$$

Therefore we can rewrite the right hand side of (4.16) as

$$\begin{aligned}\overline{(\Pi_1 \wedge \mathcal{S}_0) \circ h_M} &= \inf_{n\in\mathbb{N}} \left\{ \min_{0\le q\le n} \left\{ (h_M \circ \Pi_1)^{(q)} \right\} \circ h_M \right\} \\ &= \inf_{q\in\mathbb{N}} \left\{ (h_M \circ \Pi_1)^{(q)} \right\} \circ h_M = \overline{(h_M \circ \Pi_1)} \circ h_M,\end{aligned}$$

which establishes (4.15).

Therefore we can rewrite the right hand side of (4.16) as

$$\begin{aligned}\overline{(\Pi_1 \wedge \mathcal{S}_0) \circ h_M} &= \inf_{n\in\mathbb{N}} \left\{ \min_{0\le q\le n} \left\{ (h_M \circ \Pi_1)^{(q)} \right\} \circ h_M \right\} \\ &= h_M \circ \inf_{q\in\mathbb{N}} \left\{ (h_M \circ \Pi_1)^{(q)} \right\} \circ h_M = h_M \circ \overline{(h_M \circ \Pi_1)} \circ h_M,\end{aligned}$$

which establishes (4.15). □

The dual of super-additive closure is that of super-additive closure, defined as follows.

**Definition 4.2.2 (Super-additive closure of an operator).** *Let* $\Pi$ *be an operator taking* $\mathcal{G}^J \to \mathcal{G}^J$. *The super-additive closure of* $\Pi$, *denoted by* $\underline{\Pi}$, *is defined by*

$$\underline{\Pi} = \mathcal{S}_0 \vee \Pi \vee (\Pi \circ \Pi) \vee (\Pi \circ \Pi \circ \Pi) \vee \ldots = \sup_{n\ge 0} \left\{ \Pi^{(n)} \right\}. \tag{4.17}$$

# 4.3 Fixed Point Equation (Space Method)

## 4.3.1 Main Theorem

We now have the tools to solve an important problem of network calculus, which has some analogy with ordinary differential equations in conventional system theory.

The latter problem reads as follows: let $\Pi$ be an operator from $\mathbb{R}^J$ to $\mathbb{R}^J$, and let $\vec{a} \in \mathbb{R}^J$. What is then the solution $\vec{x}(t)$ to the differential equation

$$\frac{d\vec{x}}{dt}(t) = \Pi(\vec{x})(t) \tag{4.18}$$

with the inital condition

$$\vec{x}(0) = \vec{a}. \tag{4.19}$$

Here $\Pi$ is an operator taking $\mathcal{G}^J \to \mathcal{G}^J$, and $\vec{a} \in \mathcal{G}^J$. The problem is now to find the largest function $\vec{x}(t) \in \mathcal{G}^J$, which verifies the recursive inequality

$$\vec{x}(t) \le \Pi(\vec{x})(t) \tag{4.20}$$

and the initial condition

$$\vec{x}(t) \le \vec{a}(t). \tag{4.21}$$

The differences are however important: first we have inequalities instead of equalities, and second, contrary to (4.18), (4.20) does not describe the evolution of the trajectory $\vec{x}(t)$ with time $t$, starting from a fixed point $\vec{a}$, but the successive iteration of $\Pi$ on the whole trajectory $\vec{x}(t)$, starting from a fixed, given function $\vec{a}(t) \in \mathcal{G}^J$.

The following theorem provides the solution this problem, under weak, technical assumptions that are almost always met.

**Theorem 4.3.1 (Space method).** *Let $\Pi$ be an upper semi-continuous operator taking $\mathcal{G}^J \to \mathcal{G}^J$. For any fixed function $\vec{a} \in \mathcal{G}^J$, the problem*

$$\vec{x} \leq \vec{a} \wedge \Pi(\vec{x}) \tag{4.22}$$

*has one maximum solution in $\mathcal{G}^J$, given by $\vec{x}^\star = \overline{\Pi}(\vec{a})$.*

The theorem is proven in [26]. We give here a direct proof that does not have the pre-requisites in [26]. It is based on a fixed point argument. We call the application of this theorem "Space method", because the iterated variable is not time $t$ (as in the "Time method" described shortly later) but the full sequence $\vec{x}$ itself. The theorem applies therefore indifferently whether $t \in \mathbb{Z}$ or $t \in \mathbb{R}$.

**Proof:** (i) Let us first show that $\overline{\Pi}(\vec{a})$ is a solution of (4.22). Consider the sequence $\{\vec{x}^n\}$ of decreasing sequences defined by

$$\begin{aligned} \vec{x}_0 &= \vec{a} \\ \vec{x}_{n+1} &= \vec{x}_n \wedge \Pi(\vec{x}_n), \qquad n \geq 0. \end{aligned}$$

Then one checks that

$$\vec{x}^\star = \inf_{n \geq 0} \{\vec{x}_n\}$$

is a solution to (4.22) because $\vec{x}^\star \leq \vec{x}_0 = \vec{a}$ and because $\Pi$ is upper-semi-continuous so that

$$\Pi(\vec{x}^\star) = \Pi(\inf_{n \geq 0} \{\vec{x}_n\}) = \inf_{n \geq 0} \{\Pi(\vec{x}_n)\} \geq \inf_{n \geq 0} \{\vec{x}_{n+1}\} \geq \inf_{n \geq 0} \{\vec{x}_n\} = \vec{x}^\star.$$

Now, one easily checks that $\vec{x}_n = \inf_{0 \leq m \leq n} \{\Pi^{(m)}(\vec{a})\}$, so

$$\vec{x}^\star = \inf_{n \geq 0} \{\vec{x}_n\} = \inf_{n \geq 0} \inf_{0 \leq m \leq n} \{\Pi^{(m)}(\vec{a})\} = \inf_{n \geq 0} \{\Pi^{(n)}(\vec{a})\} = \overline{\Pi}(\vec{a}).$$

This also shows that $\vec{x}^\star \in \mathcal{G}^J$.

(ii) Let $\vec{x}$ be a solution of (4.22). Then $\vec{x} \leq \vec{a}$ and since $\Pi$ is isotone, $\Pi(\vec{x}) \leq \Pi(\vec{a})$. From (4.22), $\vec{x} \leq \Pi(\vec{x})$, so that $\vec{x} \leq \Pi(\vec{a})$. Suppose that for some $n \geq 1$, we have shown that $\vec{x} \leq \Pi^{(n-1)}(\vec{a})$. Then as $\vec{x} \leq \Pi(\vec{x})$ and as $\Pi$ is isotone, it yields that $\vec{x} \leq \Pi^{(n)}(\vec{a})$. Therefore $\vec{x} \leq \inf_{n \geq 0}\{\Pi^{(n)}(\vec{a})\} = \overline{\Pi}(\vec{a})$, which shows that $\vec{x}^\star = \overline{\Pi}(\vec{a})$ is the maximal solution. $\square$

Similarly, we have the following result in Max-plus algebra.

**Theorem 4.3.2 (Dual space method).** *Let $\Pi$ be a lower semi-continuous operator taking $\mathcal{G}^J \to \mathcal{G}^J$. For any fixed function $\vec{a} \in \mathcal{G}^J$, the problem*

$$\vec{x} \geq \vec{a} \vee \Pi(\vec{x}) \tag{4.23}$$

*has one minimum solution, given by $\vec{x}^\star = \underline{\Pi}(\vec{a})$.*

### 4.3.2 Examples of Application

Let us now apply this theorem to five particular examples. We will first revisit the input-output characterization of the greedy shaper of Section 1.5.2, and of the variable capacity node described at the end of Section 1.3.2. Next we will apply it to two window flow control problems (with a fixed length window). Finally, we will revisit the variable length packet greedy shaper of Section 1.7.4.

#### Input-Output Characterization of Greedy Shapers

Remember that a greedy shaper is a system that delays input bits in a buffer, whenever sending a bit would violate the constraint $\sigma$, but outputs them as soon as possible otherwise. If $R$ is the input flow, the output is thus the maximal function $x \in \mathcal{F}$ satisfying the set of inequalities (1.13), which we can recast as

$$x \leq R \wedge \mathcal{C}_\sigma(x).$$

It is thus given by $R^* = \overline{\mathcal{C}_\sigma}(x) = \mathcal{C}_{\overline{\sigma}}(x) = \overline{\sigma} \otimes x$. If $\sigma$ is a "good" function, one therefore retrieves the main result of Theorem 1.5.1.

#### Input-Output Characterization of Variable Capacity Nodes

The variable capacity node was introduced at the end of Section 1.3.2, where the variable capacity is modeled by a cumulative function $M(t)$, where $M(t)$ is the total capacity available to the flow between times $0$ and $t$. If $m(t)$ is the instantaneous capacity available to the flow at time $t$, then $M(t)$ is the primitive of this function. In other words, if $t \in \mathbb{R}$,

$$M(t) = \int_0^t m(s)ds \tag{4.24}$$

and if $t \in \mathbb{Z}$ the integral is replaced by a sum on $s$. If $R$ is the input flow and $x$ is the output flow of the variable capacity node, then the variable capacity constraint imposes that for all $0 \leq s \leq t$

$$x(t) - x(s) \leq M(t) - M(s),$$

which we can recast using the idempotent operator $h_M$ as

$$x \leq h_M(x). \tag{4.25}$$

On the other hand, the system is causal, so that

$$x \leq R. \tag{4.26}$$

The output of the variable capacity node is therefore the maximal solution of system (4.25) and (4.26). It is thus given by

$$R^*(t) = \overline{h}_M(R)(t) = h_M(R)(t) = \inf_{0 \leq s \leq t}\{M(t) - M(s) + R(s)\}$$

because the sub-additive closure of an idempotent operator is the operator itself, as we have seen in the previous section.

**Static window flow control – example 1**

Let us now consider an example of a feedback system. This example is found independently in [10] and [64, 2]. A data flow $a(t)$ is fed via a window flow controller to a network offering a service curve $\beta$. The window flow controller limits the amount of data admitted into the network in such a way that the total backlog is less than or equal to $W$, where $W > 0$ (the window size) is a fixed number (Figure 4.1).

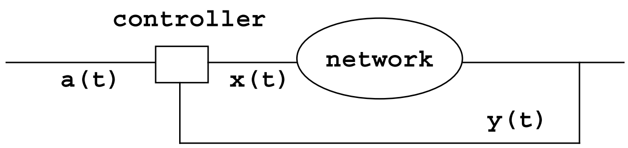



Figure 4.1: Static window flow control, from [10] or [64]

Call $x(t)$ the flow admitted to the network, and $y(t)$ the output. The definition of the controller means that $x(t)$ is the maximum solution to

$$\begin{cases} x(t) \leq a(t) \\ x(t) \leq y(t) + W \end{cases} \tag{4.27}$$

We do not know the mapping $\Pi : x \to y = \Pi(x)$, but we assume that $\Pi$ is isotone, and we assume that $y(t) \geq (\beta \otimes x)(t)$, which can be recast as

$$\Pi(x) \geq \mathcal{C}_\beta(x). \tag{4.28}$$

We also recast System (4.27) as

$$x \leq a \wedge \{\Pi(x) + W\}, \tag{4.29}$$

and direclty apply Theorem 4.3.1 to derive that the maximum solution is

$$x = \overline{(\Pi + W)}(a).$$

Since $\Pi$ is isotone, so is $\Pi + W$. Therefore, because of (4.28) and applying Theorem 4.2.2, we get that

$$x = \overline{(\Pi + W)}(a) \geq \overline{(\mathcal{C}_\beta + W)}(a). \tag{4.30}$$

Because of Theorem 4.2.1,

$$\overline{(\mathcal{C}_\beta + W)}(a) = \overline{\mathcal{C}}_{\beta+W}(a) = \mathcal{C}_{\overline{\beta+W}}(a) = \overline{(\beta + W)} \otimes a.$$

Combining this relationship with (4.30) we have that

$$y \geq \beta \otimes x \geq \beta \otimes \left( \overline{(\beta + W)} \otimes a \right) = \left( \beta \otimes \overline{(\beta + W)} \right)(a),$$

which shows that the complete, closed-loop system of Figure 4.1 offers to flow $a$ a service curve [10]

$$\beta_{\text{wfc1}} = \beta \otimes \overline{(\beta + W)}. \tag{4.31}$$

For example, if $\beta = \beta_{R,T}$ then the service curve of the closed-loop system is the function represented on Figure 4.2. When $RT \leq W$, the window does not add any restriction on the service guarantee offered by the open-loop system, as in this case $\beta_{\text{wfc1}} = \beta$. If $RT > W$ on the other hand, the service curve is smaller than the open-loop service curve.

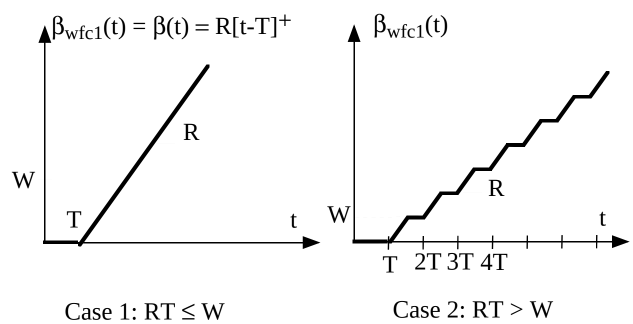



Figure 4.2: The service curve $\beta_{\text{wfc1}}$ of the closed-loop system with static window flow control, when the service curve of the open loop system is $\beta_{R,T}$ with $RT \leq W$ (left) and $RT > W$ (right).

**Static window flow control – example 2**

Let us extend the window flow control model to account for the existence of background traffic, which constraints the input traffic rate at time $t$, $dx/dt(t)$ (if $t \in \mathbb{R}$) or $x(t) - x(t-1)$ (if $t \in \mathbb{Z}$), to be less that some given rate $m(t)$. Let $M(t)$ denote the primitive of this prescribed rate function. Then the rate constraint on $x$ becomes (4.25). Function $M(t)$ is not known, but we assume that there is some function $\gamma \in \mathcal{F}$ such that

$$M(t) - M(s) \geq \gamma(t-s)$$

for any $0 \leq s \leq t$, which we can recast as

$$h_M \geq \mathcal{C}_\gamma. \tag{4.32}$$

This is used in [43] to derive a service curve offered by the complete system to the incoming flow $x$, which we shall also compute now by applying Theorem 4.3.1.

With the additional constraint (4.25), one has to compute the maximal solution of

$$x \le a \wedge \{\Pi(x) + W\} \wedge h_M(x), \tag{4.33}$$

which is

$$x = \overline{(\{\Pi + W\} \wedge h_M)}(a). \tag{4.34}$$

As in the previous subsection, we do not know $\Pi$ but we assume that it is isotone and that $\Pi \ge \mathcal{C}_\beta$. We also know that $h_M \ge \mathcal{C}_\gamma$. A first approach to get a service curve for $y$, is to compute a lower bound of the right hand side of (4.34) by time-invariant linear operators, which commute as we have seen earlier in this chapter. We get

$$\{\Pi + W\} \wedge h_M \ge \{\mathcal{C}_\beta + W\} \wedge \mathcal{C}_\gamma = \mathcal{C}_{\{\beta+W\}\wedge\gamma},$$

and therefore (4.34) becomes

$$x \ge \overline{\mathcal{C}}_{\{\beta+W\}\wedge\gamma}(a) = \mathcal{C}_{\overline{\{\beta+W\}\wedge\gamma}}(a) = \overline{(\{\beta + W\} \wedge \gamma)} \otimes a.$$

Because of Theorem 3.1.11,

$$\overline{\{\beta + W\} \wedge \gamma} = \overline{(\beta + W)} \otimes \overline{\gamma}$$

so that

$$y \ge \beta \otimes x \ge \left(\beta \otimes \overline{(\beta + W)} \otimes \overline{\gamma}\right) \otimes a$$

and thus a service curve for flow $a$ is

$$\beta \otimes \overline{(\beta + W)} \otimes \overline{\gamma}. \tag{4.35}$$

Unfortunately, this service curve can be quite useless. For example, if for some $T > 0$, $\gamma(t) = 0$ for $0 \le t \le T$, then $\overline{\gamma}(t) = 0$ for all $t \ge 0$, and so the service curve is zero.

A better bound is obtained by differing the lower bounding of $h_M$ by the time-invariant operator $\mathcal{C}_\gamma$ after having used the idempotency property in the computation of the sub-additive closure of the right hand side of (4.34), via Corollary 4.2.1. Indeed, this corollary allows us to replace (4.34) by

$$x = \left(\overline{(h_M \circ (\Pi + W))} \circ h_M\right)(a).$$

Now we can bound $h_M$ below by $\mathcal{C}_\gamma$ to obtain

$$\begin{aligned}
\overline{(h_M \circ (\Pi + W))} \circ h_M &\ge \overline{(\mathcal{C}_\gamma \circ \mathcal{C}_{\beta+W})} \circ \mathcal{C}_\gamma \\
&= \overline{\mathcal{C}}_{\gamma\otimes(\beta+W)} \circ \mathcal{C}_\gamma \\
&= \mathcal{C}_{\overline{\beta\otimes\gamma+W}} \circ \mathcal{C}_\gamma \\
&= \mathcal{C}_{\gamma\otimes\overline{(\beta\otimes\gamma+W)}}.
\end{aligned}$$

We obtain a better service curve than by our initial approach, where we had directly replaced $h_M$ by $\mathcal{C}_\gamma$:

$$\beta_{\text{wfc2}} = \beta \otimes \gamma \otimes \overline{(\beta \otimes \gamma + W)}. \tag{4.36}$$

is a better service curve than (4.35).

For example, if $\beta = \beta_{R,T}$ and $\gamma = \beta_{R',T'}$, with $R > R'$ and $W < R'(T + T')$, then the service curve of the closed-loop system is the function represented on Figure 4.3.

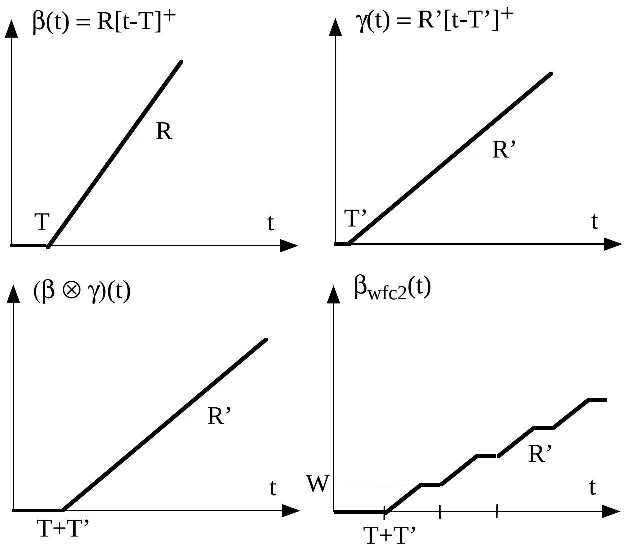



Figure 4.3: The service curve $\beta_{\text{wfc}_2}$ of the closed-loop system with window flow control (bottom right), when the service curve of the open loop system is $\beta = \beta_{R,T}$ (top left) and when $\gamma = \beta_{R',T'}$ (top right), with $R > R'$ and $W < R'(T + T')$.

**Packetized greedy shaper**

Our last example in this chapter is the packetized greedy shaper introduced in Section 1.7.4. It amounts to computing the maximum solution to the problem

$$x \le R \wedge \mathcal{P}_L(x) \wedge \mathcal{C}_\sigma(x)$$

where $R$ is the input flow, $\sigma$ is a "good" function and $L$ is a given sequence of cumulative packet lengths.

We can apply Theorem 4.3.1 and next Theorem 4.2.2 to obtain

$$x = \overline{\mathcal{P}_L \wedge \mathcal{C}_\sigma}(R) = \overline{\mathcal{P}_L \circ \mathcal{C}_\sigma}(R)$$

which is precisely the result of Theorem 1.7.4.

## 4.4 Fixed Point Equation (Time Method)

We conclude this chapter by another version of Theorem 4.3.1 that applies only to the disrete-time setting. It amounts to compute the maximum solution $\vec{x} = \overline{\Pi}(\vec{a})$ of (4.22) by iterating on time $t$ instead of interatively applying operator $\Pi$ to the full trajectory $\vec{a}(t)$. We call this method the "time method" (see also [11]). It is valid

| Operator | $\mathcal{C}_\sigma$ | $\mathcal{D}_\sigma$ | $\mathcal{S}_T$ | $h_\sigma$ | $\mathcal{P}_L$ |
|---|---|---|---|---|---|
| Upper semi-continuous | yes | no | yes | yes | yes |
| Lower semi-continuous | no | yes | yes | no | no |
| Isotone | yes | yes | yes | yes | yes |
| Min-plus linear | yes | no | yes | yes | no |
| Max-plus linear | no | yes | yes | no | no |
| Causal | yes | no | yes (1) | yes | yes |
| Shift-invariant | yes | yes | yes | no | no |
| Idempotent | no (2) | no (2) | no (3) | yes | yes |

(1) (if $T \geq 0$)
(2) (unless $\sigma$ is a 'good' function)
(3) (unless $T = 0$)

Table 4.1: A summary of properties of some common operators

under stronger assumptions than the space method, as we require here that operator $\Pi$ be min-plus linear.

**Theorem 4.4.1.** *Let $\Pi = \mathcal{L}_H$ be a min-plus linear operator taking $\mathcal{F}^J \to \mathcal{F}^J$, with impulse response $H \in \tilde{\mathcal{G}}^J$. For any fixed function $\vec{a} \in \mathcal{F}^J$, the problem*

$$\vec{x} \leq \vec{a} \wedge \mathcal{L}_H(\vec{x}) \tag{4.37}$$

*has one maximum solution, given by*

$$\begin{aligned} \vec{x}^\star(0) &= \vec{a}(0) \\ \vec{x}^\star(t) &= \vec{a}(0) \wedge \inf_{0 \leq u \leq t-1} \{H(t,u) + \vec{x}^\star(u)\}. \end{aligned}$$

**Proof:** Note that the existence of a maximum solution is given by Theorem 4.3.1. Define $\vec{x}^\star$ by the recursion in the Theorem. As $H \in \tilde{\mathcal{G}}^J$ it follows easily by induction that $\vec{x}^\star$ is a solution to problem (4.37). Conversely, for any solution $\vec{x}$, $\vec{x}(0) \leq a(0) = \vec{x}^\star(0)$ and if $\vec{x}(u) \leq \vec{x}^\star(u)$ for all $0 \leq u \leq t-1$, it follows that $\vec{x}(t) \leq \vec{x}^\star(t)$ which shows that $\vec{x}^\star$ is the maximal solution. $\square$

## 4.5 Conclusion

This chapter has introduced min-plus and max-plus operators, and discussed their properties, which are summarized in Table 4.5. The central result of this chapter, which will be applied in the next chapters, is Theorem 4.3.1, which enables us to compute the maximal solution of a set of inqualities involving the iterative application of an upper semi-continuous operator.

# Part III

# A Second Course in Network Calculus

# Chapter 5

# Optimal Multimedia Smoothing

In this chapter we apply network calculus to smooth multimedia data over a network offering reservation based services, such as ATM or RSVP/IP, for which we know one minimal service curve. One approach to stream video is to act on the quantization levels at the encoder output: this is called rate control, see e.g. [24]. Another approach is to smooth the video stream, using a smoother fed by the encoder, see e.g. [65, 68, 56]. In this chapter, we deal with this second approach.

A number of smoothing algorithms have been proposed to optimize various performance metrics, such as peak bandwidth requirements, variability of transmission rates, number of rate changes, client buffer size [27]. With network calculus, we are able to compute the minimal client buffer size required given a maximal peak rate, or even a more complex (VBR) smoothing curve. We can also compute the minimal peak rate required given a given client buffer size. We will see that the scheduling algorithm that must be implemented to reach these bounds is not unique, and we will determine the full set of video transmission schedules that minimize these resources and achieve these optimal bounds.

## 5.1 Problem Setting

A video stream stored on the server disk is directly delivered to the client, through the network, as shown on Figure 5.1. At the sender side, a smoothing device reads the encoded video stream $R(t)$ and sends a stream $x(t)$ that must conform to an arrival curve $\sigma$, which we assume to be a 'good' function, i.e. is sub-additive and such that $\sigma(0) = 0$. The simplest and most popular smoothing curve in practice is a constant rate curve (or equivalently, a peak rate constraint) $\sigma = \lambda_r$ for some $r > 0$.

We take the transmission start as origin of time: this implies that $x(t) = 0$ for $t \leq 0$.

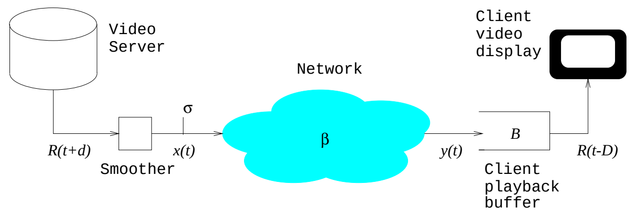



Figure 5.1: Video smoothing over a single network.

At the receiver side, the video stream $R$ will be played back after $D$ units of times, the *playback delay*: the output of the decoding buffer $B$ must therefore be $R(t-D)$.

The network offers a guaranteed service to the flow $x$. If $y$ denotes the output flow, it is not possible, in general, to express $y$ as a function of $x$. However we assume that the service guarantee can be expressed by a service curve $\beta$. For example, as we have seen in Chapter 1, the IETF assumes that RSVP routers offer a rate-latency service curve $\beta$ of the form $\beta_{L,C}(t) = C[t-L]^+ = \max\{0, C(t-L)\}$. Another example is a network which is completely transparent to the flow (i.e. which does not incur any jitter to the flow nor rate limitation, even if it can introduce a fixed delay, which we ignore in this chapter as we can always take it into account separately). We speak of a *null network*. It offers a service curve $\beta(t) = \delta_0(t)$.

To keep mathematical manipulations simple, we assume that the encoding buffer size is large enough to contain the full data stream. On the other hand, the receiver (decoding) buffer is a much more scarce resource. Its finite size is denoted by $B$.

As the stream is pre-recorded and stored in the video server, it allows the smoother to prefetch and send some of the data before schedule. We suppose that the smoother is able to look ahead data for up to $d$ time units ahead. This *look-ahead delay* can take values ranging from zero (in the most restrictive case where no prefetching is possible) up to the length of the full stream. The sum of the look-ahead delay and playback delay is called the *total delay*, and is denoted by $T$: $T = D + d$.

These constraints are described more mathematically in Section 5.2.

We will then apply Theorem 4.3.1 to solve the following problems:

(i) we first compute, in Section 5.3, the minimal requirements on the playback delay $D$, on the look-ahead delay $d$, and on the client buffer size $B$ guaranteeing a lossless transmission for given smoothing and service curves $\sigma$ and $\beta$.

(ii) we then compute, in Section 5.4, all scheduling strategies at the smoother that will achieve transmission in the parameter setting computed in Section 5.3. We call the resulting scheduling "optimal smoothing".

(iii) in the CBR case ($\sigma = \lambda_r$), for a given rate $r$ and for a rate-latency service curve ($\beta = \beta_{L,C}$), we will obtain, in Section 5.5, closed-form expressions of the minimal values of $D$, $T = D + d$ and $B$ required for lossless smoothing. We will also solve the dual problem of computing the minimal rate $r$ needed to deliver video for a given playback delay $D$, look-ahead delay $d$ and client buffer size $B$.

We will then compare optimal smoothing with greedy shaping in Section 5.6 and with separate delay equalization in Section 5.7. Finally, we will repeat problems (i) and (iii) when intermediate caching is allowed between a backbone network and an access network.

## 5.2 Constraints Imposed by Lossless Smoothing

We can now formalize the constraints that completely define the smoothing problem illustrated on Figure 5.1).

- **Flow** $x \in \mathcal{F}$: As mentioned above, the chosen origin of time is such that $x(t) = 0$ for $t \leq 0$, or equivalently

$$x(t) \leq \delta_0(t). \tag{5.1}$$

- **Smoothness constraint**: Flow $x$ is constrained by an arrival curve $\sigma(\cdot)$. This means that for all $t \geq 0$

$$x(t) \leq (x \otimes \sigma)(t) = \mathcal{C}_\sigma(x)(t). \tag{5.2}$$

- **Playback delay constraint (no playback buffer underflow)**: The data is read out from the playback buffer after $D$ unit of times at a rate given by $R(t-D)$. This implies that $y(t) \geq R(t-D)$. However we do not know the exact expression of $y$ as a function of $x$. All we know is that the network guarantees a service curve $\beta$, namely that $y(t) \geq (x \otimes \beta)(t)$. The output flow may therefore be as low as $(x \otimes \beta)(t)$, and hence we can replace $y$ in the previous inequality to obtain $(x \otimes \beta)(t) \geq R(t-D)$. Using Rule 14 in Theorem 3.1.12, we can recast this latter inequality as

$$x(t) \geq (R \oslash \beta)(t-D) = \mathcal{D}_\beta(R)(t-D) \tag{5.3}$$

  for all $t \geq 0$.

- **Playback buffer constraint (no playback buffer overflow)**: The size of the playback buffer is limited to $B$, and to prevent any overflow of the buffer, we must impose that $y(t) - R(t-D) \leq B$ for all $t \geq 0$. Again, we do not know the exact value of $y$, but we know that it can be as high as $x$, but not higher, because the network is a causal system. Therefore the constraint becomes, for all $t \geq 0$,

$$x(t) \leq R(t-D) + B. \tag{5.4}$$

- **Look-ahead delay constraint**: We suppose that the encoder can prefetch data from the server up to $d$ time units ahead, which translates in the following inequality:

$$x(t) \leq R(t+d). \tag{5.5}$$

## 5.3 Minimal Requirements on Delays and Playback Buffer

Inequalities (5.1) to (5.5) can be recast as two sets of inequalities as follows:

$$\begin{aligned} x(t) &\leq \delta_0(t) \wedge R(t+d) \wedge \{R(t-D)+B\} \wedge \mathcal{C}_\sigma(x)(t) && (5.6)\\ x(t) &\geq (R \oslash \beta)(t-D). && (5.7)\end{aligned}$$

There is a solution $x$ to the smoothing problem if and only if it simultaneously verifies (5.6) and (5.7). This is equivalent to requiring that the maximal solution of (5.6) is larger than the right hand side of (5.7) for all $t$.

Let us first compute the maximal solution of (5.6). Inequality (5.6) has the form

$$x \leq a \wedge \mathcal{C}_\sigma(x) \qquad (5.8)$$

where

$$a(t) = \delta_0(t) \wedge R(t+d) \wedge \{R(t-D)+B\}. \qquad (5.9)$$

We can thus apply Theorem 4.3.1 to compute the unique maximal solution of (5.8), which is $x_{\max} = \mathcal{C}_\sigma(a) = \sigma \otimes a$ because $\sigma$ is a 'good' function. Replacing $a$ by its expression in (5.9), we compute that the maximal solution of (5.6) is

$$x_{\max}(t) = \sigma(t) \wedge \{(\sigma \otimes R)(t+d)\} \wedge \{(\sigma \otimes R)(t-D)+B\}\,. \qquad (5.10)$$

We are now able to compute the smallest values of the playback delay $D$, of the total delay $T$ and of the playback buffer $B$ ensuring the existence of a solution to the smoothing problem, thanks to following theorem. The requirement on $d$ for reaching the smallest value of $D$ is therefore $d = T - D$.

**Theorem 5.3.1 (Requirements for optimal smoothing).** *The smallest values of $D$, $T$ and $B$ ensuring a lossless smoothing to a 'good' curve $\sigma$ through a network offering a service curve $\beta$ are*

$$\begin{aligned} D_{\min} &= h(R, (\beta \otimes \sigma)) = \inf\{t \geq 0\ : (R \oslash (\beta \otimes \sigma))(-t) \leq 0\} && (5.11)\\ T_{\min} &= h((R \oslash R), (\beta \otimes \sigma)) && (5.12)\\ &= \inf\{t \geq 0\ : ((R \oslash R) \oslash (\beta \otimes \sigma))(-t) \leq 0\} \\ B_{\min} &= v((R \oslash R), (\beta \otimes \sigma)) = ((R \oslash R) \oslash (\beta \otimes \sigma))(0). && (5.13)\end{aligned}$$

*where $h$ and $v$ denote respectively the horizontal and vertical distances given by Definition 3.1.15.*

**Proof:** The set of inequalities (5.6) and (5.7) has a solution if, and only if, the maximal solution of (5.6) is larger or equal to the right hand side of (5.7) at all times. This amounts to impose that for all $t \in \mathbb{R}$

$$\begin{aligned}
(R \oslash \beta)(t-D) - \sigma(t) &\leq 0 \\
(R \oslash \beta)(t-D) - (\sigma \otimes R)(t+d) &\leq 0 \\
(R \oslash \beta)(t-D) - (\sigma \otimes R)(t-D) &\leq B.
\end{aligned}$$

Using the deconvolution operator and its properties, the latter three inequalities can be recast as

$$\begin{aligned}
(R \oslash (\beta \otimes \sigma))(-D) &\leq 0 \\
((R \oslash R) \oslash (\beta \otimes \sigma))\,(-T) &\leq 0 \\
((R \oslash R) \oslash (\beta \otimes \sigma))\,(0) &\leq B.
\end{aligned}$$

The minimal values of $D$, $T$ and $B$ satisfying these three inequalities are given by (5.11), (5.12) and (5.13). These three inequalities are therefore the necessary and sufficient conditions ensuring the existence of a solution to the smoothing problem. □

## 5.4 Optimal Smoothing Strategies

An optimal smoothing strategy is a solution $x(t)$ to the lossless smoothing problem where $D$, $T = D + d$ and $B$ take their minimal value given by Theorem 5.3.1. The previous section shows that there exists at least one optimal solution, namely (5.10). It is however not the only one, as we will see in this section.

### 5.4.1 Maximal Solution

The maximal solution (5.10) requires only the evaluation of an infimum at time $t$ over the past values of $R$ and over the future values of $R$ up to time $t + d_{\min}$, with $d_{\min} = T_{\min} - D_{\min}$. Of course, we need the knowledge of the traffic trace $R(t)$ to dimension $D_{\min}$, $d_{\min}$ and $B_{\min}$. However, once we have these values, we do not need the full stream for the computation of the smoothed input to the network.

### 5.4.2 Minimal Solution

To compute the minimal solution, we reformulate the lossless smoothing problem slightly differently. Because of Rule 14 of Theorem 3.1.12, an inequality equivalent to (5.2) is

$$x(t) \geq (x \oslash \sigma)(t) = \mathcal{D}_\sigma(x)(t). \tag{5.14}$$

We use this equivalence to replace the set of inequalities (5.6) and (5.7) by the equivalent set

$$x(t) \leq \delta_0(t) \wedge R(t+d) \wedge \{R(t-D) + B\} \tag{5.15}$$

$$x(t) \geq (R \oslash \beta)(t-D) \vee \mathcal{D}_\sigma(x)(t). \tag{5.16}$$

One can then apply Theorem 4.3.2 to compute the *minimal* solution of (5.16), which is $x_{\min} = \mathcal{D}_\sigma(b) = b \oslash \sigma$ where $b(t) = \delta_0(t) \wedge R(t+d) \wedge \{R(t-D)+B\}$, because $\sigma$ is a 'good' function. Eliminating $b$ from these expressions, we compute that the minimal solution is

$$x_{\min}(t) = (R \oslash (\beta \otimes \sigma))(t-D), \tag{5.17}$$

and compute the constraints on $d$, $D$ and $B$ ensuring that it verifies (5.15): one would get the very same values of $D_{\min}$, $T_{\min}$ and $B_{\min}$ given by (5.11) (5.12) and (5.13).

It does achieve the values of $D_{\min}$ and $B_{\min}$ given by (5.11) and (5.13), but requires nevertheless the evaluation, at time $t$, of a supremum over all values of $R$ up to the end of the trace, contrary to the maximal solution (5.10). Min-plus deconvolution can however be represented in the time inverted domain by a min-plus convolution, as we have seen in Section 3.1.10. As the duration of the pre-recorded stream is usually known, the complexity of computing a min-plus deconvolution can thus be reduced to that of computing a convolution.

### 5.4.3 Set of Optimal Solutions

Any function $x \in \mathcal{F}$ such that

$$x_{\min} \le x \le x_{\max}$$

and

$$x \le x \otimes \sigma$$

is therefore also a solution to the lossless smoothing problem, for the same minimal values of the playback delay, look-ahead delay and client buffer size. This gives the set of all solutions. A particular solution among these can be selected to further minimize another metric, such as the ones discussed in [27], e.g. number of rate changes or rate variability.

The top of Figure 5.2 shows, for a synthetic trace $R(t)$, the maximal solution (5.10) for a CBR smoothing curve $\sigma(t) = \lambda_r(t)$ and a service curve $\beta(t) = \delta_0(t)$, whereas the bottom shows the minimal solution (5.17). Figure 5.3 shows the same solutions on a single plot, for the MPEG trace $R(t)$ of the top of Figure 1.2.4 representing the number of packet arrivals per time slot of 40 ms corresponding to a MPEG-2 encoded video when the packet size is 416 bytes for each packet.

An example of VBR smoothing on the same MPEG trace is shown on Figure 5.4, with a smoothing curve derived from the T-SPEC field, which is given by $\sigma = \gamma_{P,M} \wedge \gamma_{r,b}$, where $M$ is the maximum packet size (here $M = 416$ Bytes), $P$ the peak rate, $r$ the sustainable rate and $b$ the burst tolerance. Here we roughly have $P = 560$ kBytes/sec, $r = 330$ kBytes/sec and $b = 140$ kBytes The service curve is a rate-latency curve $\beta_{L,C}$ with $L = 1$ second and $r = 370$ kBytes/sec. The two traces have the same envelope, thus the same minimum buffer requirement (here, 928kBytes). However the second trace has its bursts later, thus, has a smaller minimum playback delay ($D_2 = 2.05$s versus $D_1 = 2.81$s).

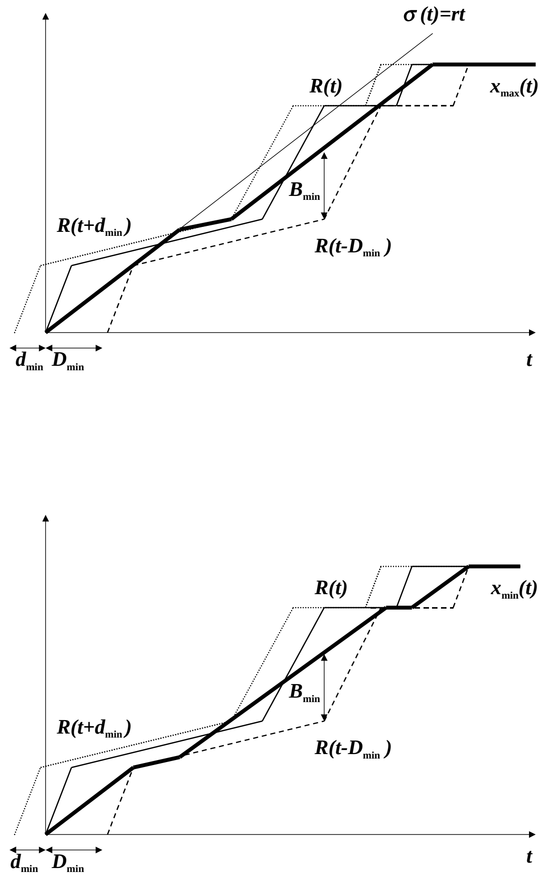



Figure 5.2: In bold, the maximal solution (top figure) and minimal solution (bottom figure) to the CBR smoothing problem with a null network.

## 5.5 Optimal Constant Rate Smoothing

Let us compute the above values in the case of a constant rate (CBR) smoothing curve $\sigma(t) = \lambda_r(t) = rt$ (with $t \geq 0$) and a rate-latency service curve of the network $\beta(t) = \beta_{L,C}(t) = C[t-L]^+$. We assume that $r < C$, the less interesting case where $r \geq C$ being handled similarly. We will often use the decomposition of a rate-latency function as the min-plus convolution of a pure delay function, with a constant rate function: $\beta_{L,C} = \delta_L \otimes \lambda_C$. We will also use the following lemma.

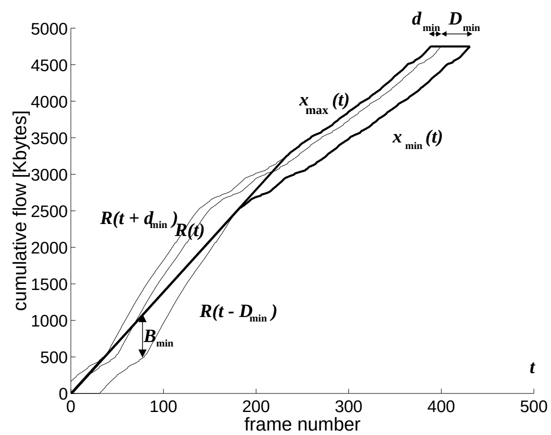



Figure 5.3: In bold, the maximal and minimal solutions to the CBR smoothing problem of an MPEG trace with a null network. A frame is generated every 40 msec.

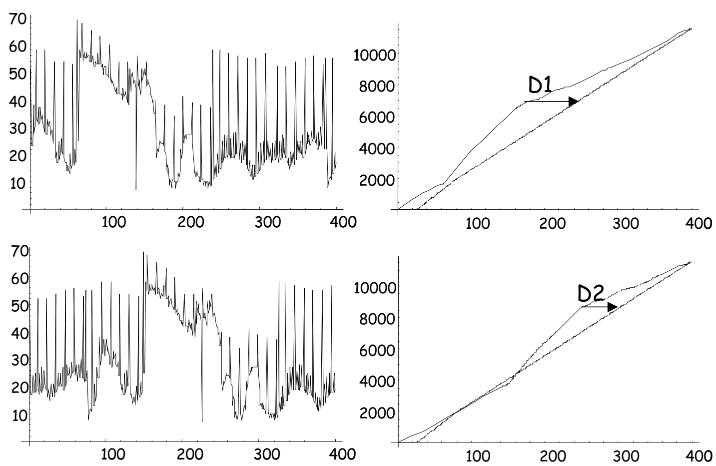



Figure 5.4: Two MPEG traces with the same arrival curve (left). The corresponding playback delays $D_1$ and $D_2$ are the horizontal deviations between the cumulative flows $R(t)$ and function $\sigma \otimes \beta$ (right).

**Lemma 5.5.1.** *If $f \in \mathcal{F}$,*

$$h(f, \beta_{L,C}) = L + \frac{1}{C}(f \oslash \lambda_C)(0). \tag{5.18}$$

**Proof:** As $f(t) = 0$ for $t \leq 0$ and as $\beta_{L,C} = \delta_L \otimes \lambda_C$, we can write for any $t \geq 0$

$$\begin{aligned}
(f \oslash \beta_{L,C})(-t) &= \sup_{u \geq 0}\{f(u-t) - (\delta_L \otimes \lambda_C)(u)\} \\
&= \sup_{u \geq 0}\{f(u-t) - \lambda_C(u-L)\} \\
&= \sup_{v \geq -t}\{f(v) - \lambda_C(v+t-L)\} \\
&= \sup_{v \geq 0}\{f(v) - \lambda_C(v+t-L)\} \\
&= \sup_{v \geq 0}\{f(v) - \lambda_C(v)\} - C(t-L) \\
&= (f \oslash \lambda_C)(0) - Ct + CL,
\end{aligned}$$

from which we deduce the smallest value of $t$ making the left-hand side of this equation non-positive is given by (5.18). □

In the particular CBR case, the optimal values (5.11), (5.12) and (5.13) become the following ones.

**Theorem 5.5.1 (Requirements for CBR optimal smoothing).** *If $\sigma = \lambda_r$ and $\beta = \beta_{L,C}$ with $r < C$, the smallest values of $D$, of $T$ and of $B$ are*

$$\begin{aligned}
D_{\min} &= L + \frac{1}{r}(R \oslash \lambda_r)(0) && (5.19) \\
T_{\min} &= L + \frac{1}{r}((R \oslash R) \oslash \lambda_r)(0) && (5.20) \\
B_{\min} &= ((R \oslash R) \oslash \lambda_r))(L) \leq rT_{\min}. && (5.21)
\end{aligned}$$

**Proof:** To establish (5.19) and (5.20), we note that $R$ and $(R \oslash R) \in \mathcal{F}$. Since $r < C$

$$\beta \otimes \sigma = \beta_{L,C} \otimes \lambda_r = \delta_L \otimes \lambda_C \otimes \lambda_r = \delta_L \otimes \lambda_r = \beta_{L,r}$$

so that we can apply Lemma 5.5.1 with $f = R$ and $f = (R \oslash R)$, respectively.

To establish (5.21), we develop (5.13) as follows

$$\begin{aligned}
((R \oslash R) \oslash (\beta \otimes \sigma))(0) &= ((R \oslash R) \oslash (\delta_L \otimes \lambda_r))(0) \\
&= \sup_{u \geq 0}\{(R \oslash R)(u) - \lambda_r(u-L)\} \\
&= ((R \oslash R) \oslash \lambda_r)(L) \\
&= \sup_{u \geq L}\{(R \oslash R)(u) - \lambda_r(u-L)\} \\
&= \sup_{u \geq L}\{(R \oslash R)(u) - \lambda_r(u)\} + rL
\end{aligned}$$

$$\begin{aligned} &\leq \sup_{u\geq 0}\{(R \oslash R)(u) - \lambda_r(u)\} + rL \\ &= ((R \oslash R) \oslash \lambda_r)(0) + rL = rT_{\min}. \end{aligned}$$

□

This theorem provides the minimal values of playback delay $D_{\min}$ and buffer $B_{\min}$, as well as the minimal look-ahead delay $d_{\min} = T_{\min} - D_{\min}$ for a given constant smoothing rate $r < C$ and a given rate-latency service curve $\beta_{L,C}$. We can also solve the dual problem, namely compute for given values of playback delay $D$, of the look-ahead delay $d$, of the playback buffer $B$ and for a given rate-latency service curve $\beta_{L,C}$, the minimal rate $r_{\min}$ which must be reserved on the network.

**Theorem 5.5.2 (Optimal CBR smoothing rate).** *If $\sigma = \lambda_r$ and $\beta = \beta_{L,C}$ with $r < C$, the smallest value of $r$, given $D \geq L$, $d$ and $B \geq (R \oslash R)(L)$, is*

$$\begin{aligned} r_{\min} &= \sup_{t>0}\left\{\frac{R(t)}{t+D-L}\right\} \vee \sup_{t>0}\left\{\frac{(R \oslash R)(t)}{t+D+d-L}\right\} \\ &\quad \vee \sup_{t>0}\left\{\frac{(R \oslash R)(t+L) - B}{t}\right\}. \end{aligned} \tag{5.22}$$

**Proof:** Let us first note that because of (5.19), there is no solution if $D < L$. On the other hand, if $D \geq L$, then (5.19) implies that the rate $r$ must be such that for all $t > 0$

$$D \geq L + \frac{1}{r}(R(t) - rt)$$

or equivalently $r \geq R(t)/(t + D - L)$. The latter being true for all $t > 0$, we must have $r \geq \sup_{t\geq 0}\{R(t)/(t + D - L)\}$. Repeating the same argument with (5.20) and (5.21), we obtain the minimal rate (5.22). □

In the particular case where $L = 0$ and $r < C$ the network is completely transparent to the flow, and can be considered as a null network: can replace $\beta(t)$ by $\delta_0(t)$. The values (5.19), (5.20) and (5.21) become, respectively,

$$D_{\min} = \frac{1}{r}(R \oslash \lambda_r)(0) \tag{5.23}$$

$$T_{\min} = \frac{1}{r}((R \oslash R) \oslash \lambda_r)(0) \tag{5.24}$$

$$B_{\min} = ((R \oslash R) \oslash \lambda_r))(0) = rT_{\min}. \tag{5.25}$$

It is interesting to compute these values on a real video trace, such as the first trace on top of Figure 1.2.4. Since $B_{\min}$ is directly proportional to $T_{\min}$ because of (5.25), we show only the graphs of the values of $D_{\min}$ and $d_{\min} = T_{\min} - D_{\min}$, as a function of the CBR smoothing rate $r$ on Figure 5.5. We observe three qualitative ranges of rates: (i) the very low ones where the playback delay is very large, and where look-ahead does not help in reducing it; (ii) a middle range where the playback delay can be kept quite small, thanks to the use of look-ahead and (iii) the high rates above the peak rate of the stream, which do not require any playback nor lookahead of the stream. These three regions can be found on every MPEG trace [75], and depend on the location of the large burst in the trace. If it comes sufficiently late, then the use of look-ahead can become quite useful in keeping the playback delay small.

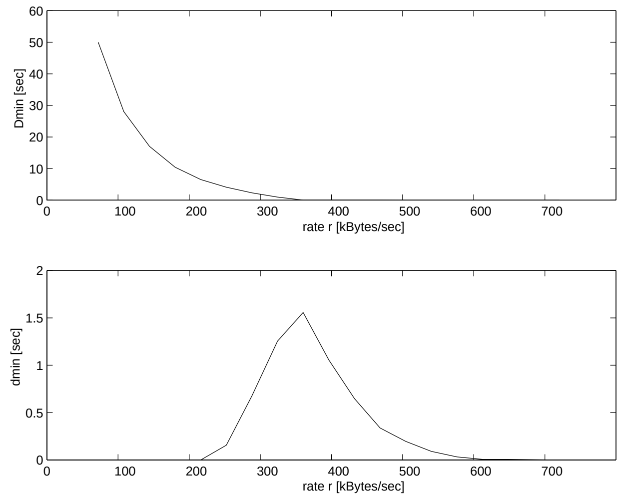



Figure 5.5: Minimum playback delay $D_{\min}$ and corresponding look-ahead delay $d_{\min}$ for a constant rate smoothing $r$ of the MPEG-2 video trace shown on top of Figure 1.2.4.

## 5.6 Optimal Smoothing versus Greedy Shaping

An interesting question is to compare the requirements on $D$ and $B$, due to the scheduling obtained in Section 5.4, which are minimal, with those that a simpler scheduling, namely the greedy shaper of Section 1.5, would create. As $\sigma$ is a 'good' function, the solution of a greedy shaper is

$$x_{\text{shaper}}(t) = (\sigma \otimes R)(t). \tag{5.26}$$

To be a solution for the smoothing problem, it must satisfy all constraints listed in Section 5.2. It already satisfies (5.1), (5.2) and (5.5). Enforcing (5.3) is equivalent to impose that for all $t \in \mathbb{R}$

$$(R \oslash \beta)(t - D) \leq (\sigma \otimes R)(t),$$

which can be recast as

$$((R \oslash R) \oslash (\beta \otimes \sigma))(-D) \leq 0. \tag{5.27}$$

This implies that the minimal playback delay needed for a smoothing using a greedy shaping algorithm is equal to the minimal total delay $T_{\min}$, the sum of the playback and lookahead delays, for the optimal smoothing algorithm. It means that the only way an optimal smoother allows to decrease the playback delay is its ability to look ahead and send data in advance. If this look-ahead is not possible ($d = 0$) as for example for a live video transmission, the playback delay is the same for the greedy shaper and the optimal smoother.

The last constraint that must be verified is (5.4), which is equivalent to impose that for all $t \in \mathbb{R}$

$$(\sigma \otimes R)(t) \leq R(t - D) + B,$$

which can be recast as

$$((R \otimes \sigma) \oslash R)(D) \leq B. \tag{5.28}$$

Consequently, the minimal requirements on the playback delay and buffer using a greedy shaper instead of an optimal smoother are given by the following theorem.

**Theorem 5.6.1 (Requirements for greedy shaper).** *If $\sigma$ is a 'good' function, then the smallest values of $D$ and $B$ for lossless smoothing of flow $R$ by a greedy shaper are*

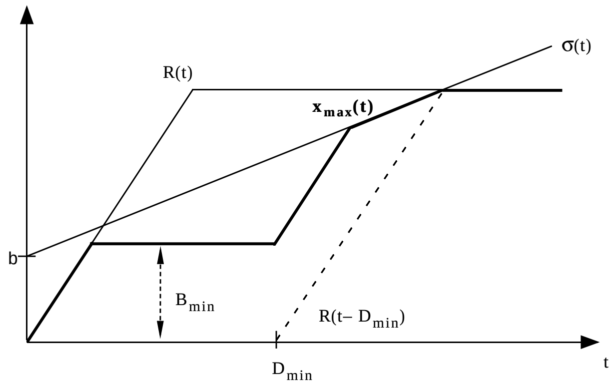
$$\begin{aligned} D_{\textit{shaper}} &= T_{\min} = h((R \oslash R), (\beta \otimes \sigma)) && (5.29)\\ B_{\textit{shaper}} &= ((R \otimes \sigma) \oslash R)(D_{\textit{shaper}}) \in [B_{\min}, \sigma(D_{\textit{shaper}})]. && (5.30) \end{aligned}$$

**Proof:** The expressions of $D_{\text{shaper}}$ and $B_{\text{shaper}}$ follow immediately from (5.27) and (5.28). The only point that remains to be shown is that $B_{\text{shaper}} \leq \sigma(D_{\text{shaper}})$, which we do by picking $s = u$ in the inf below:

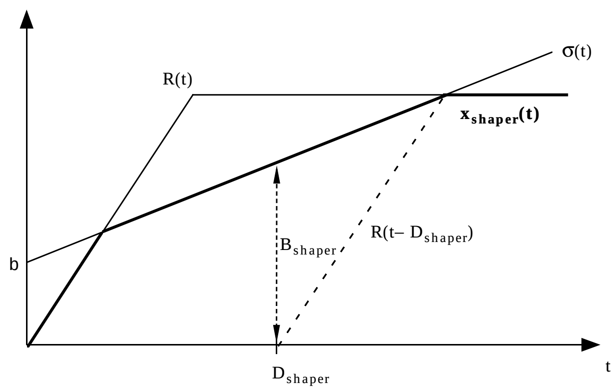
$$\begin{aligned} B_{\text{shaper}} &= (R \oslash (R \otimes \sigma))(D_{\text{shaper}}) \\ &= \sup_{u \geq 0} \left\{ \inf_{0 \leq s \leq u + D_{\text{shaper}}} \left\{ R(s) + \sigma(u + D_{\text{shaper}} - s) \right\} - R(u) \right\} \\ &\leq \sup_{u \geq 0} \left\{ R(u) + \sigma(u + D_{\text{shaper}} - u) - R(u) \right\} \\ &= \sigma(D_{\text{shaper}}). \end{aligned}$$

□

Consequently, a greedy shaper does not minimize, in general, the playback buffer requirements, although it does minimize the playback delay when look-ahead is possible. Figure 5.6 shows the maximal solution $x_{\max}$ of the optimal shaper (top) and the solution $x_{\text{shaper}}$ of the greedy shaper (bottom) when the shaping curve is a one leaky bucket affine curve $\sigma = \gamma_{r,b}$, when the look-ahead delay $d = 0$ (no look ahead possible) and for a null network ($\beta = \delta_0$). In this case the playback delays are identical, but not the playback buffers.




(a) Optimal smoothing solution, with
playback buffer requirements




(a) Greedy shaper solution, with
playback buffer requirements

Figure 5.6: In bold, the maximal solution (top figure) and minimal solution (bottom figure) to the smoothing problem with a null network, no look-ahead and an affine smoothing curve $\sigma = \gamma_{r,b}$ .

Another example is shown on Figure 5.7 for the MPEG-2 video trace shown on top of Figure 1.2.4. Here the solution of the optimal smoother is the minimal solution $x_{\min}$.

There is however one case where a greedy shaper does minimize the playback buffer: a constant rate smoothing ($\sigma = \lambda_r$) over a null network ($\beta = \delta_0$). Indeed, in this case, (5.25) becomes

$$B_{\min} = rT_{\min} = rD_{\text{shaper}} = \sigma(D_{\text{shaper}}),$$

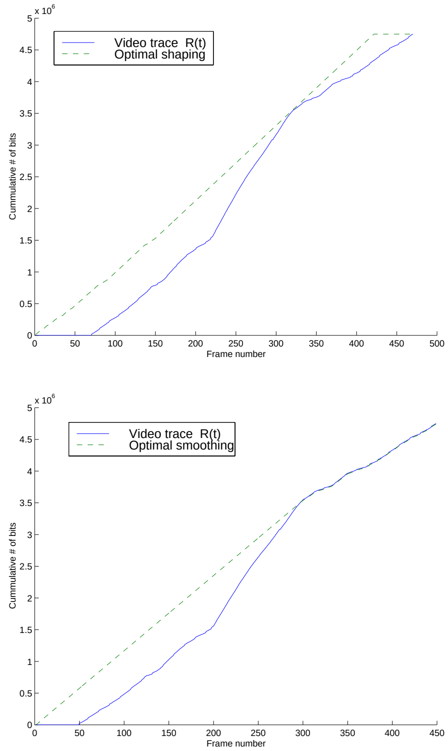



Figure 5.7: Example of optimal shaping versus optimal smoothing for the MPEG-2 video trace shown on top of Figure 1.2.4. The example is for a null network and a smoothing curve $\sigma = \gamma_{P,M} \wedge \gamma_{r,b}$ with $M = 416$ bytes, $P = 600$ kBytes/sec, $r = 300$ kBytes/sec and $b = 80$ kBytes. The figure shows the optimal shaper [resp. smoother] output and the original signal (video trace), shifted by the required playback delay. The playback delay is $2.76$ sec for optimal shaping (top) and $1.92$ sec for optimal smoothing (bottom).

and therefore $B_{\text{shaper}} = B_{\min}$. Consequently, if no look-ahead is possible and if the network is transparent to the flow, greedy shaping is an optimal CBR smoothing strategy.

## 5.7 Comparison with Delay Equalization

A common method to implement a decoder is to first remove any delay jitter caused by the network, by delaying the arriving data in a delay equalization buffer, before using the playback buffer to compensate for fluctuations due to pre-fetching. Figure 5.8 shows such a system. If the delay equalization buffer is properly configured, its combination with the guaranteed service network results into a fixed delay network, which, from the viewpoint we take in this chapter, is equivalent to a null network. Compared to the original scenario in Figure 5.1, there are now two separate buffers for delay equalization and for compensation of prefetching. We would like to understand the impact of this separation on the minimum playback delay $D_{\min}$.

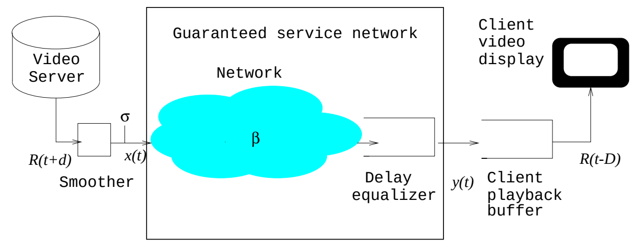



Figure 5.8: Delay equalization at the receiver.

The delay equalization buffer operates by delaying the first bit of data by an initial delay $D_{eq}$, equal to the worst case delay through the network. We assume that the network offers a rate-latency service curve $\beta_{L,C}$. Since the flow $x$ is constainted by the arrival curve $\sigma$ which is assumed to be a 'good' function, we know from Theorem 1.4.4, that the worst-case delay is

$$D_{eq} = h(\sigma, \beta_{L,C}).$$

On the other hand, the additional part of the playback delay to compensate for fluctuations due to pre-fetching, denoted by $D_{pf}$, is given by (5.11) with $\beta$ replaced by $\delta_0$:

$$D_{pf} = h(R, \delta_0 \otimes \sigma) = h(R, \sigma).$$

The sum of these two delays is, in general, larger than the optimal playback delay (without a separation between equalization and compensation for prefetching), $D_{\min}$, given by (5.11):

$$D_{\min} = h(R, \beta_{L,C} \otimes \sigma).$$

Consider the example of Figure 5.9, where $\sigma = \gamma_{r,b}$ with $r < C$. Then one easily computes the three delays $D_{\min}$, $D_{eq}$ and $D_{pf}$, knowing that

$$\begin{aligned} \beta_{L,C} \otimes \sigma &= \delta_L \otimes \lambda_C \otimes \gamma_{r,b} = \delta_L \otimes (\lambda_C \wedge \gamma_{r,b}) \\ &= (\delta_L \otimes \lambda_C) \wedge (\delta_L \otimes \gamma_{r,b}) = \beta_{L,C} \wedge (\delta_L \otimes \gamma_{r,b}). \end{aligned}$$

One clearly has $D_{\min} < D_{eq} + D_{pf}$: separate delay equalization gives indeed a larger overall playback delay. In fact, looking carefully at the figure (or working out the computations), we can observe that the combination of delay equalization and compensation for prefetching in a single buffer accounts for the busrtiness of the

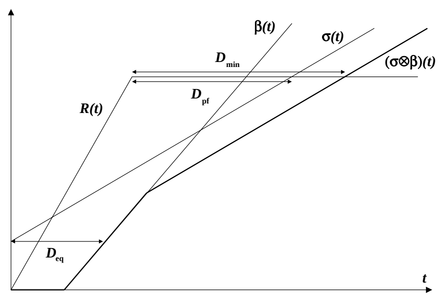



Figure 5.9: Delays $D_{\min}$, $D_{eq}$ and $D_{pf}$ for a rate-latency service curve $\beta_{L,C}$ and an affine smoothing curve $\sigma = \gamma_{r,b}$ .

(optimally) smoothed flow only once. This is another instance of the "pay bursts only once" phenomenon, which we have already met in Section 1.4.3.

We must however make – once again – an exception for a constant rate smoothing. Indeed, if $\sigma = \lambda_r$ (with $r < C$), then $D_{pf}$ is given by (5.23) and $D_{\min}$ by (5.19), so that

$$\begin{aligned} D_{eq} &= h(\lambda_r, \beta_{L,C}) = L \\ D_{pf} &= \frac{1}{r}(R \oslash \lambda_r)(0) \\ D_{\min} &= L + \frac{1}{r}(R \oslash \lambda_r)(0) \end{aligned}$$

and therefore $D_{\min} = D_{eq} + D_{pf}$. In the CBR case, separate delay equalization is thus able to attain the optimal playback delay.

## 5.8 Lossless Smoothing over Two Networks

We now consider the more complex setting where two networks separate the video server from the client: the first one is a backbone network, offering a service curve $\beta_1$ to the flow, and the second one is a local access network, offering a service curve $\beta_2$ to the flow, as shown on Figure 5.10. This scenario models intelligent, dynamic caching often done at local network head-ends. We will compute the requirements on $D$, $d$, $B$ and on the buffer $X$ of this intermediate node in Subsection 5.8.1. Moreover, we will see in Subsection 5.8.2 that for constant rate shaping curves and rate-latency service curves, the size of the client buffer $B$ can be reduced by implementing a particular smoothing strategy instead of FIFO scheduling at the intermediate node.

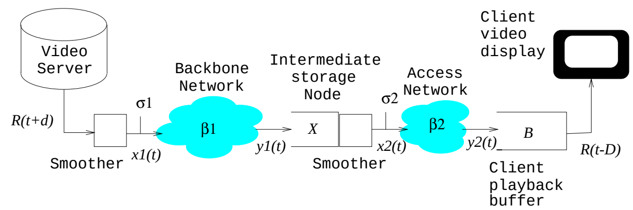



Figure 5.10: Smoothing over two networks with a local caching node.

Two flows need therefore to be computed: the first one $x_1(t)$ at the input of the backbone network, and the second one $x_2(t)$ at the input of the local access network, as shown on Figure 5.10.

The constraints on both flows are now as follows:

- **Causal flow** $x_1$: This constraint is the same as (5.1), but with $x$ replaced by $x_1$:
$$x_1(t) \le \delta_0(t), \tag{5.31}$$
- **Smoothness constraint**: Both flows $x_1$ and $x_2$ are constrained by two arrival curves $\sigma_1$ and $\sigma_2$:
$$x_1(t) \le (x_1 \otimes \sigma_1)(t) \tag{5.32}$$
$$x_2(t) \le (x_2 \otimes \sigma_2)(t). \tag{5.33}$$
- **No playback and intermediate server buffers underflow**: The data is read out from the playback buffer after $D$ unit of times at a rate given by $R(t-D)$, which implies that $y_2(t) \ge R(t-D)$. On the other hand, the data is retrieved from the intermediate server at a rate given by $x_2(t)$, which implies that $y_1(t) \ge x_2(t)$. As we do not know the expressions of the outputs of each network, but only a service curve $\beta_1$ and $\beta_2$ for each of them, we can replace $y_1$ by $x_1 \otimes \beta_1$ and $y_2$ by $x_2 \otimes \beta_2$, and reformulate these two constraints by
$$x_2(t) \le (x_1 \otimes \beta_1)(t) \tag{5.34}$$
$$x_2(t) \ge (R \oslash \beta_2)(t-D). \tag{5.35}$$
- **No playback and intermediate server buffers overflow**: The size of the playback and cache buffers are limited to $B$ and $X$, respectively, and to prevent any overflow of the buffer, we must impose that $y_1(t) - x_2(t) \le X$ and $y_2(t) - R(t-D) \le B$ for all $t \ge 0$. Again, we do not know the exact value

of $y_1$ and $y_2$, but we know that they are bounded by $x_1$ and $x_2$, respectively, so that the constraints becomes, for all $t \geq 0$,

$$x_1(t) \leq x_2(t) + X \tag{5.36}$$
$$x_2(t) \leq R(t - D) + B. \tag{5.37}$$

- **Look-ahead delay constraint**: this constraint is the same as in the single network case:
$$x_1(t) \leq R(t + d). \tag{5.38}$$

### 5.8.1 Minimal Requirements on the Delays and Buffer Sizes for Two Networks

Inequalities (5.31) to (5.38) can be recast as three sets of inequalities as follows:

$$x_1(t) \leq \delta_0(t) \wedge R(t + d) \wedge (\sigma_1 \otimes x_1)(t) \wedge (x_2(t) + X) \tag{5.39}$$
$$x_2(t) \leq \{R(t - D) + B\} \wedge (\beta_1 \otimes x_1)(t) \wedge (\sigma_2 \otimes x_2)(t) \tag{5.40}$$
$$x_2(t) \geq (R \oslash \beta_2)(t - D). \tag{5.41}$$

We use the same technique for solving this problem sa in Section 5.3, except that now the dimension of the system $J$ is 2 instead of 1.

With $T$ denoting transposition, let us introduce the following notations:

$$\begin{aligned} \vec{x}(t) &= [x_1(t) \quad x_2(t)]^T \\ \vec{a}(t) &= [\delta_0(t) \wedge R(t + d) \quad R(t - D) + B]^T \\ \vec{b}(t) &= [0 \quad (R \oslash \beta_2)(t - D)]^T \end{aligned}$$

$$\Sigma(t) = \begin{bmatrix} \sigma_1(t) & \delta_0(t) + X \\ \beta_1(t) & \sigma_2(t) \end{bmatrix}.$$

With these notations, the set of inequalities (5.39), (5.40) and (5.41) can therefore be recast as

$$\vec{x} \leq \vec{a} \wedge (\Sigma \otimes \vec{x}) \tag{5.42}$$
$$\vec{x} \geq \vec{b}. \tag{5.43}$$

We will follow the same approach as in Section 5.3: we first compute the maximal solution of (5.42) and then derive the constraints on $D$, $T$ (and hence $d$), $X$ and $B$ ensuring the existence of this solution. We apply thus Theorem 4.3.1 again, but this time in the two-dimensional case, to obtain an explicit formulation of the maximal solution of (5.42). We get

$$\vec{x}_{\max} = \overline{\mathcal{C}}_\Sigma(\vec{a}) = (\overline{\Sigma} \otimes \vec{a}) \tag{5.44}$$

where $\overline{\Sigma}$ is the sub-additive closure of $\Sigma$, which is, as we know from Section 4.2,

$$\overline{\Sigma} = \inf_{n \in \mathbb{N}} \{\Sigma^{(n)}\} \tag{5.45}$$

where $\Sigma^{(0)} = D_0$ and $\Sigma^{(n)}$ denotes the $n$th self-convolution of $\Sigma$. Application of (5.45) to matrix $\Sigma$ is straightforward, but involves a few manipulations which are skipped. Denoting

$$\begin{aligned} \alpha &= \sigma_1 \otimes \sigma_2 \otimes \inf_{n \in \mathbb{N}} \left\{ \beta_1^{(n+1)} + nX \right\} \\ &= \sigma_1 \otimes \sigma_2 \otimes \beta_1 \otimes \overline{\beta_1 + X}, \end{aligned} \tag{5.46}$$

we find that

$$\overline{\Sigma} = \begin{bmatrix} \sigma_1 \wedge (\alpha + X) & (\sigma_1 \otimes \sigma_2 + X) \wedge (\alpha + 2X) \\ \alpha & \sigma_2 \wedge (\alpha + X) \end{bmatrix}$$

and therefore the two coordinates of the maximal solution of (5.42) are

$$\begin{aligned} x_{1\max}(t) &= \sigma_1(t) \wedge \{\alpha(t) + X\} \wedge (\sigma_1 \otimes R)(t+d) \wedge \{(\alpha \otimes R)(t+d) + X\} \\ &\quad \wedge \{(\sigma_1 \otimes \sigma_2 \otimes R)(t-D) + B + X\} \\ &\quad \wedge \{(\alpha \otimes R)(t-D) + B + 2X\} \end{aligned} \tag{5.47}$$

$$\begin{aligned} x_{2\max}(t) &= \alpha(t) \wedge (\alpha \otimes R)(t+d) \wedge \{(\sigma_2 \otimes R)(t-D) + B\} \\ &\quad \wedge \{(\alpha \otimes R)(t-D) + B + X\}. \end{aligned} \tag{5.48}$$

Let us mention that an alternative (and computationally simpler) approach to obtain (5.47) and (5.48) would have been to first compte the maximal solution of (5.40), as a function of $x_1$, and next to replace $x_2$ in (5.39) by this latter value.

We can now express the constraints on $X$, $B$, $D$ and $d$ that will ensure that a solution exists by requiring that (5.48) be larger than (5.41). The result is stated in the following theorem, whose proof is similar to that of Theorem 5.3.1.

**Theorem 5.8.1.** *The lossless smoothing of a flow to (sub-additive) curves $\sigma_1$ and $\sigma_2$, respectively, over two networks offering service curves $\beta_1$ and $\beta_2$ has a solution if and only if the $D$, $T$, $X$ and $B$ verify the following set of inequalities, with $\alpha$ defined by (5.46):*

$$(R \oslash (\alpha \otimes \beta_2)(-D) \leq 0 \tag{5.49}$$

$$((R \oslash R) \oslash (\alpha \otimes \beta_2))(-T) \leq 0 \tag{5.50}$$

$$((R \oslash R) \oslash (\sigma_2 \otimes \beta_2))(0) \leq B \tag{5.51}$$

$$((R \oslash R) \oslash (\alpha \otimes \beta_2))(0) \leq B + X. \tag{5.52}$$

## 5.8.2 Optimal Constant Rate Smoothing over Two Networks

Let us compute the values of Theorem 5.8.1 in the case of two constant rate (CBR) smoothing curves $\sigma_1 = \lambda_{r_1}$ and $\sigma_2 = \lambda_{r_2}$. We assume that each network offers a rate-latency service curve $\beta_i = \beta_{L_i, C_i}$, $i = 1, 2$. We assume that $r_i \leq C_i$ In this case the optimal values of $D$, $T$ and $B$ become the following ones, depending on the value of $X$.

**Theorem 5.8.2.** *Let $r = r_1 \wedge r_2$. Then we have the following three cases depending on $X$:*
*(i) If $X \geq rL_1$, then $D_{\min}$, $T_{\min}$ and $B_{\min}$ are given by*

$$\begin{aligned}
D_{\min} &= L_1 + L_2 + \frac{1}{r}(R \oslash \lambda_r)(0) && (5.53)\\
T_{\min} &= L_1 + L_2 + \frac{1}{r}((R \oslash R) \oslash \lambda_r)(0) && (5.54)\\
B_{\min} &= ((R \oslash R) \oslash \lambda_{r_2})(L_2) \vee \{((R \oslash R) \oslash \lambda_r)(L_1 + L_2) - X\}\\
&\leq ((R \oslash R) \oslash \lambda_r)(L_2). && (5.55)
\end{aligned}$$

*(ii) If $0 < X < rL_1$ then $D_{\min}$, $T_{\min}$ and $B_{\min}$ are bounded by*

$$\begin{aligned}
&\frac{X}{r} + L_2 + \frac{L_1}{X}(R \oslash \lambda_{\frac{X}{L_1}})(0) \leq D_{\min}\\
&\quad\leq L_1 + L_2 + \frac{L_1}{X}(R \oslash \lambda_{\frac{X}{L_1}})(0) && (5.56)\\
&\frac{X}{r} + L_2 + \frac{L_1}{X}((R \oslash R) \oslash \lambda_{\frac{X}{L_1}})(0) \leq T_{\min}\\
&\quad\leq L_1 + L_2 + \frac{L_1}{X}((R \oslash R) \oslash \lambda_{\frac{X}{L_1}})(0) && (5.57)\\
&((R \oslash R) \oslash \lambda_{\frac{X}{L_1}})(L_1 + L_2) - r_2 L_1 \leq B_{\min}\\
&\quad\leq ((R \oslash R) \oslash \lambda_{\frac{X}{L_1}})(L_2) && (5.58)
\end{aligned}$$

*(iii) Let $K$ be duration of the stream. If $X = 0 < rL_1$ then $D_{\min} = K$.*

*Proof.* One easily verifies that $\delta_{L_1}^{(n+1)} = \delta_{(n+1)L_1}$ and that $\lambda_{C_1}^{(n+1)} = \lambda_{C_1}$. Since $\beta_1 = \beta_{L_1,C_1} = \delta_{L_1} \otimes \lambda_{C_1}$, and since $r = r_1 \wedge r_2 \leq C_1$, (5.46) becomes

$$\begin{aligned}
\alpha &= \lambda_r \otimes \inf_{n \in \mathbb{N}} \left\{ \delta_{(n+1)L_1} \otimes \lambda_{C_1} + nX \right\}\\
&= \delta_{L_1} \otimes \inf_{n \in \mathbb{N}} \left\{ \delta_{nL_1} \otimes \lambda_r + nX \right\}. && (5.59)
\end{aligned}$$

(i) If $X \geq rL_1$, then for $t \geq nL_1$

$$(\delta_{nL_1} \otimes \lambda_r)(t) + nX = \lambda_r(t - nL_1) + nX = rt + n(X - rL_1) \geq rt = \lambda_r(t)$$

whereas for $0 \leq t < nL_1$

$$(\delta_{nL_1} \otimes \lambda_r)(t) + nX = \lambda_r(t - nL_1) + nX = nX \geq nrL_1 > rt = \lambda_r(t).$$

Consequently, for all $t \geq 0$, $\alpha(t) \geq (\delta_{L_1} \otimes \lambda_r)(t)$. On the other hand, taking $n = 0$ in the infimum in (5.59) yields that $\alpha \leq \delta_{L_1} \otimes \lambda_r$. Combining these two inequalities, we get that

$$\alpha = \delta_{L_1} \otimes \lambda_r$$

and hence that

$$\alpha \otimes \beta_2 = \delta_{L_1} \otimes \lambda_r \otimes \delta_{L_2} \otimes \lambda_{r_2} = \delta_{L_1+L_2} \otimes \lambda_r = \beta_{L_1+L_2,r}. \tag{5.60}$$

Inserting this last relation in (5.49) to (5.52), and using Lemma 5.5.1 we establish (5.53), (5.54) and the equality in (5.55). The inequality in (5.55) is obtained by noticing that $r_2 \geq r$ and that

$$\begin{aligned}((R \oslash R) \oslash \lambda_r)(L_1 + L_2) - X &= \sup_{u \geq 0}\{(R \oslash R)(u + L_1 + L_2) - ru\} - X \\ &= \sup_{v \geq L_1}\{(R \oslash R)(v + L_2) - r(v - L_1)\} - X \\ &\leq \sup_{v \geq 0}\{(R \oslash R)(v + L_2) - rv\} + (rL_1 - X) \\ &\leq ((R \oslash R) \oslash \lambda_r)(L_2).\end{aligned}$$

(ii) If $0 < X < rL_1$, the computation of $\alpha$ does not provide a rate-latency curve anymore, but a function that can be bounded below and above by the two following rate-latency curves: $\beta_{L_1,X/L_1} \leq \alpha \leq \beta_{X/r,X/L_1}$. Therefore, replacing (5.60) by

$$\delta_{L_1+L_2} \otimes \lambda_{\frac{X}{L_1}} \leq \alpha \otimes \beta_2 \leq \delta_{\frac{X}{r}+L_2} \otimes \lambda_{\frac{X}{L_1}},$$

and applying Lemma 5.5.1 to both bounding rate-latency curves $\beta_{L_1,X/L_1}$ and $\beta_{X/r,X/L_1}$, we get respectively the lower and upper bounds (5.56) to (5.58).

(iii) If $X = 0$ and $rL_1 > 0$ then (5.59) yields that $\alpha(t) = 0$ for all $t \geq 0$. In this case (5.49) becomes $\sup_{u \geq 0}\{R(u - D)\} \leq 0$. This is possible only if $D$ is equal to the duration of the stream. □

It is interesting to examine these results for two particular values of $X$.

The first one is $X = \infty$. If the intermediate server is a greedy shaper whose output is $x_2(t) = (\sigma_2 \otimes y_1)(t)$, one could have applied Theorem 5.5.1 with $\sigma_2 = \lambda_r$ and $\beta = \beta_1 \otimes \sigma_2 \otimes \beta_2 = \delta_{L_1+L_2} \otimes \lambda_{r_2} = \beta_{L_1+L_2,r_2}$ to find out that $D$ and $T$ are still given by (5.53) and (5.54) but that $B = ((R \oslash R) \oslash \lambda_r)(L_1 + L_2)$ is larger than (5.55). Using the caching scheduling (5.48) instead of a greedy shaping one allows therefore to decrease the playback buffer size, but not the delays. The buffer $X$ of the intermediate node does not need to be infinite, but can be limited to $rL_1$.

The second one is $X = 0$. Then whatever the rate $r > 0$, if $L_1 > 0$, the playback delay is the length of the stream, which makes streaming impossible in practice. When $L_1 = L_2 = 0$ however (in which case we have two null networks) $X = rL_1 = 0$ is the optimal intermediate node buffer allocation. This was shown in [65](Lemma 5.3) using another approach. We see that when $L_1 > 0$, this is no longer the case.

## 5.9 Bibliographic Notes

The first application of network calculus to optimal smoohting is found in [51], for an unlimited value of the look-ahead delay. The minimal solution (5.17) is shown to be an optimal smoothing scheme. The computation of the minimum look-ahead delay, and of the maximal solution, is done in [75]. Network calculus allows to retrieve some results found using other methods, such as the optimal buffer allocation of the intermdiate node for two null networks computed in [65].

It also allows to extend these results, by computing the full set of optimal schedules and by taking into account non null networks, as well as by using more complex shaping curves $\sigma$ than constant rate service curves. For example, with the Resource Reservation Protocol (RSVP), $\sigma$ is derived from the T-SPEC field in messages used for setting up the reservation, and is given by $\sigma = \gamma_{P,M} \wedge \gamma_{r,b}$, where $M$ is the maximum packet size, $P$ the peak rate, $r$ the sustainable rate and $b$ the burst tolerance, as we have seen in Section 1.4.3.

The optimal T-SPEC field is computed in [51]. More precisely, the following problem is solved. As assumed by the Intserv model, every node offers a service of the form $\beta_{L,C}$ for some latency $L$ and rate $C$, with the latency parameter $L$ depending on the rate $C$ according to $L = \frac{C_0}{\rho} + D_0$. The constants $C_0$ and $D_0$ depends on the route taken by the flow throughout the network. Destinations choose a target admissible network delay $D_{net}$. The choice of a specific service curve $\beta_{L,C}$ (or equivalently, of a rate parameter $C$) is done during the reservation phase and cannot be known exactly in advance. The algorithm developed in [51] computes the admissible choices of $\sigma = \gamma_{P,M} \wedge \gamma_{r,b}$ and of $D_{net}$ in order to guarantee that the reservation that will subsequently be performed ensures a playback delay not exceeding a given value $D$.

# Chapter 6

# Aggregate Scheduling

## 6.1 Introduction

Aggregate scheduling arises naturally in many case. Let us just mention here the differentiated services framework (Section 2.4 on Page 105) and high speed switches with optical switching matrix and FIFO outputs. The state of the art for aggregate multiplexing is not very rich. In this chapter, we give a panorama of results, a number of which are new.

In a first step (Section 6.2), we evaluate how an arrival curve is transformed through aggregate multiplexing; we give a catalog of results, when the multiplexing node is either a service curve element with FIFO scheduling, or a Guaranteed Rate node (Section 2.1.3), or a service curve element with strict service curve property. This provides many simple, explicit bounds which can be used in practice.

In a second step (Section 6.3), we consider a global network using aggregate multiplexing (see assumptions below); given constraints at the inputs of the network, can we obtain some bounds for backlog and delay ? Here, the story is complex. The question of delay bounds for a network with aggregate scheduling was first raised by Chang [8]. For a given family of networks, we call *critical load factor* $\nu_{cri}$ a value of utilization factor below which finite bounds exist, and above which there exist unstable networks, i.e., networks whose backlog grow to infinity. For feed-forward networks with aggregate multiplexing, an iterative application of Section 6.2 easily shows that $\nu_{cri} = 1$. However, many networks are not feed-forward, and this result does not hold in general. Indeed, and maybe contrary to intuition, Andrews [3] gave some examples of FIFO networks with $\nu_{cri} < 1$. Still, the iterative application of Section 6.2, augmented with a time-stopping argument, provides lower bounds of $\nu_{cri}$ (which are less than 1).

In a third step (Section 6.4), we give a number of cases where we can say more. We recall the result in Theorem 2.4.1 on Page 107, which says that, for a general network with either FIFO service curve elements, or with GR nodes, we have $\nu_{cri} \geq \frac{1}{h-1}$ where $h$ is abound on the number of hops seen by any flow. Then, in

Section 6.4.1, we show that the unidirectional ring always always has $\nu_{cri} = 1$; thus, and this may be considered a surprise, the ring is not representative of non feed-forward topologies. This result is actually true under the very general assumption that the nodes on the ring are service curve elements, with any values of link speeds, and with any scheduling policy (even non FIFO) that satisfies a service curve property. As far as we know, we do not really understand why the ring is always stable, and why other topologies may not be. Last, and not least surprising, we present in Section 6.4.2 a particular case, originally found by Chlamtac, Faragó, Zhang, and Fumagalli [14], and refined by Zhang [79] and Le Boudec and Hébuterne [49] which shows that, for a homogeneous network of FIFO nodes with constant size packets, strong rate limitations at all sources have the effect of providing simple, closed form bounds.

Throughout the chapter, we make the following assumptions.

**Assumption and Notation**

- Consider a network with a fixed number $I$ of flows, following fixed paths. The collection of paths is called the topology of the network. A network node is modeled as a collection of output buffers, with no contention other than at the output buffers. Every buffer is associated with one unidirectional link that it feeds.

- Flow $i$ is constrained by one leaky bucket of rate $\rho_i$ and burstiness $\sigma_i$ at the input.

- Inside the network, flows are treated as an aggregate by the network; within an aggregate, packets are served according to some unspecified arbitration policy. We assume that the node is such that the aggregate of all flows receives a service curve at node $m$ equal to the rate-latency function with rate $r_m$ and latency $\frac{e_m}{r_m}$. This does not imply that the node is work-conserving. Also note that we do not require, unless otherwise specified, that the service curve property be strict. In some parts of the chapter, we make additional assumptions, as explained later.

  $e_m$ accounts for the latency on the link that exits node $m$; it also account for delays due to the scheduler at node $m$.

- We write $i \ni m$ to express that node $m$ is on the route of flow $i$. For any node $m$, define $\rho^{(m)} = \sum_{i \ni m} \rho_i$. The utilization factor of link $m$ is $\frac{\rho^{(m)}}{r_m}$ and the load factor of the network is $\nu = \max_m \frac{\rho^{(m)}}{r_m}$.

- The bit rate of the link feeding node $m$ is $C_m < +\infty$, with $C_m \geq r_m$.

We say that such a network is *stable* if the backlog at any node remains bounded.

## 6.2 Transformation of Arrival Curve through Aggregate Scheduling

Consider a number of flows served as an aggregate in a common node. Without loss of generality, we consider only the case of two flows. Within an aggregate, packets are served according to some unspecified arbitration policy. In the following subsections, we consider three additional assumptions.

### 6.2.1 Aggregate Multiplexing in a Strict Service Curve Element

The strict service curve property is defined in Definition 1.3.2 on Page 27. It applies to some isolated schedulers, but not to complex nodes with delay elements.

**Theorem 6.2.1 (Blind multiplexing).** *Consider a node serving two flows,* $1$ *and* $2$*, with some unknown arbitration between the two flows. Assume that the node guarantees a* strict *service curve* $\beta$ *to the aggregate of the two flows. Assume that flow* $2$ *is* $\alpha_2$*-smooth. Define* $\beta_1(t) := [\beta(t) - \alpha_2(t)]^+$*. If* $\beta_1$ *is wide-sense increasing, then it is a service curve for flow* $1$*.*

**Proof:** The proof is a straightforward extension of that of Proposition 1.3.4 on Page 26. □

We have seen an example in Section 1.3.2: if $\beta(t) = Ct$ (constant rate server or GPS node) and $\alpha_2 = \gamma_{r,b}$ (constraint by one leaky bucket) then the service curve for flow 1 is the rate-latency service curve with rate $C - r$ and latency $\frac{b}{C-r}$. Note that the bound in Theorem 6.2.1 is actually for a preemptive priority scheduler where flow 1 has low priority. It turns out that if we have no other information about the system, it is the only bound we can find. For completeness, we give the following case.

**Corollary 6.2.1 (Non preemptive priority node).** *Consider a node serving two flows,* $H$ *and* $L$*, with non-preemptive priority given to flow* $H$*. Assume that the node guarantees a* strict *service curve* $\beta$ *to the aggregate of the two flows. Then the high priority flow is guaranteed a service curve* $\beta_H(t) = [\beta(t) - l^L_{\max}]^+$ *where* $l^L_{\max}$ *is the maximum packet size for the low priority flow.*

*If in addition the high priority flow is* $\alpha_H$*-smooth, then define* $\beta_L$ *by* $\beta_L(t) = [\beta(t) - \alpha_H(t)]^+$*. If* $\beta_L$ *is wide-sense increasing, then it is a service curve for the low priority flow.*

**Proof:** The first part is an immediate consequence of Theorem 6.2.1. The second part is proven in the same way as Proposition 1.3.4. □

If the arrival curves are affine, then the following corollary of Theorem 6.2.1 expresses the burstiness increase due to multiplexing.

**Corollary 6.2.2 (Burstiness Increase due to Blind Multiplexing).** *Consider a node serving two flows in an aggregate manner. Assume the aggregate is guaranteed a* strict *service curve* $\beta_{R,T}$*. Assume also that flow* $i$ *is constrained by one leaky* *bucket with parameters* $(\rho_i, \sigma_i)$. *If* $\rho_1 + \rho_2 \leq R$ *the output of the first flow is constrained by a leaky bucket with parameters* $(\rho_1, b_1^*)$ *with*

$$b_1^* = \sigma_1 + \rho_1 T + \rho_1 \frac{\sigma_2 + \rho_2 T}{R - \rho_2}$$

Note that the burstiness increase contains a term $\rho_1 T$ that is found even if there is no multiplexing; the second term $\rho_1 \frac{\sigma_2+\rho_2 T}{R-\rho_2}$ comes from multiplexing with flow 2. Note also that if we further assume that the node is FIFO, then we have a better bound (Section 6.2.2).

**Proof:** From Theorem 6.2.1, the first flow is guaranteed a service curve $\beta_{R',T'}$ with $R' = R - \rho_2$ and $T' = \frac{\sigma_2+T\rho_2}{R-\rho_2}$. The result follows from a direct application of Theorem 1.4.3 on Page 29. □

**Do we need that the service curve property be strict ?** If we relax the assumption that the service curve property is strict, then the above results do not hold. A counter-example can be built as follows. All packets have the same size, 1 data unit, and input flows have a peak rate equal to 1. Flow 1 sends one packet at time $0$, and then stops. The node delays this packet forever. With an obvious notation, we have, for $t \geq 0$:

$$R_1(t) = \min(t, 1) \text{ and } R_1'(t) = 0$$

Flow 2 sends one packet every time unit, starting at time $t = 1$. The output is a continuous stream of packets, one per time unit, starting from time $1$. Thus

$$R_2(t) = (t-1)^+ \text{ and } R_2'(t) = R_2(t)$$

The aggregate flows are, for $t \geq 0$:

$$R(t) = t \text{ and } R'(t) = (t-1)t$$

In other words, the node offers to the aggregate flow a service curve $\delta_1$. Obviously, Theorem 6.2.1 does not apply to flow 1: if it would, flow 1 would receive a service curve $(\delta_1 - \lambda_1)^+ = \delta_1$, which is not true since it receives $0$ service. We can interpret this example in the light of Section 1.4.4 on Page 36: if the service curve property would be strict, then we could bound the duration of the busy period, which would give a minimum service guarantee to low priority traffic. We do not have such a bound on this example. In Section 6.2.2 we see that if we assume FIFO scheduling, then we do have a service curve guarantee.

### 6.2.2 Aggregate Multiplexing in a FIFO Service Curve Element

Now we relax the strict service curve property; we assume that the node guarantees to the aggregate flow a minimum service curve, and in addition assume that it handles packets in order of arrival at the node. We find some explicit closed forms bounds for some simple cases.

**Proposition 6.2.1 (FIFO Minimum Service Curves [18]).** *Consider a lossless node serving two flows, $1$ and $2$, in FIFO order. Assume that packet arrivals are instantaneous. Assume that the node guarantees a minimum service curve $\beta$ to the aggregate of the two flows. Assume that flow $2$ is $\alpha_2$-smooth. Define the family of functions $\beta^1_\theta$ by*

$$\beta^1_\theta(t) = [\beta(t) - \alpha_2(t-\theta)]^+ 1_{\{t>\theta\}}$$

*Call $R_1(t), R'_1(t)$ the input and output for flow $1$. Then for any $\theta \geq 0$*

$$R'_1 \geq R_1 \otimes \beta^1_\theta \tag{6.1}$$

*If $\beta^1_\theta$ is wide-sense increasing, flow $1$ is guaranteed the service curve $\beta^1_\theta$*

The assumption that packet arrivals are instantaneous means that we are either in a fluid system (one packet is one bit or one cell), or that the input to the node is packetized prior to being handled in FIFO order.

**Proof:** We give the proof for continuous time and assume that flow functions are left-continuous. All we need to show is Equation (6.1). Call $R_i$ the flow $i$ input, $R = R_1 + R_2$, and similarly $R'_i, R'$ the output flows.

Fix some arbitrary parameter $\theta$ and time $t$. Define

$$u := \sup\{v : \; R(v) \leq R'(t)\}$$

Note that $u \leq t$ and that

$$R(u) \leq R'(t) \text{ and } R(u^+) \geq R'(t) \tag{6.2}$$

where $R_r(u) = \inf_{v>u}[R(v)]$ is the limit to the right of $R$ at $u$.

(Case 1) consider the case where $u = t$. It follows from the above and from $R' \leq R$ that $R'_1(t) = R_1(t)$. Thus for any $\theta$, we have $R'_1(t) = R_1(t) + \beta^1_\theta(0)$ which shows that $R'_1(t) \geq (R_1 \otimes \beta^1_\theta)(t)$ in that case.

(Case 2), assume now that $u < t$. We claim that

$$R_1(u) \leq R'_1(t) \tag{6.3}$$

Indeed, if this is not true, namely, $R_1(u) > R'_1(t)$, it follows from the first part of Equation (6.2) that $R_2(u) < R'_2(t)$. Thus some bits from flow 2 arrived after time $u$ and departed by time $t$, whereas all bits of flow 1 arrived up to time $u$ have not yet departed at time $t$. This contradicts our assumption that the node is FIFO and that packets arrive instantaneously.

Similarly, we claim that

$$(R_2)_r(u) \geq R'_2(t) \tag{6.4}$$

Indeed, otherwise $x := R'_2(t) - (R_2)_r(u) > 0$ and there is some $v_0 \in (u, t]$ such that for any $v \in (u, v_0]$ we have $R_2(v) < R'_2(t) - \frac{x}{2}$. From Equation (6.2), we can find some $v_1 \in (u, v_0]$ such that if $v \in (u, v_1]$ then $R_1(v) + R_2(v) \geq R'(t) - \frac{x}{4}$. It follows that

$$R_1(v) \geq R'_1(t) + \frac{x}{4}$$

Thus we can find some $v$ with $R_1(v) > R'_1(t)$ whereas $R_2(v) < R'_2(t)$, which contradicts the FIFO assumption.

Call $s$ a time such that $R'(t) \geq R(s) + \beta(t-s)$. We have $R(s) \leq R'(t)$ thus $s \leq u$.

(Case 2a) Assume that $u < t - \theta$ thus also $t - s > \theta$. From Equation (6.4) we derive

$$R'_1(t) \geq R_1(s) + \beta(t-s) + R_2(s) - R'_2(t) \geq R_1(s) + \beta(t-s) + R_2(s) - (R_2)_r(u)$$

Now there exist some $\epsilon > 0$ such that $u + \epsilon \leq t - \theta$, thus $(R_2)_r(u) \leq R_2(t-\theta)$ and

$$R'_1(t) \geq R_1(s) + \beta(t-s) - \alpha_2(t-s-\theta)$$

It follows from Equation (6.3) that

$$R'_1(t) \geq R_1(s)$$

which shows that

$$R'_1(t) \geq R_1(s) + \beta^1_\theta(t-s)$$

(Case 2b) Assume that $u \geq t - \theta$. By Equation (6.3):

$$R'_1(t) \geq R_1(u) = R_1(u) + \beta^1_\theta(t-u)$$

□

We cannot conclude from Proposition 6.2.1 that $\inf_\theta \beta^1_\theta$ is a service curve. However, we can conclude something for the output.

**Proposition 6.2.2 (Bound for Output with FIFO).** *Consider a lossless node serving two flows, $1$ and $2$, in FIFO order. Assume that packet arrivals are instantaneous. Assume that the node guarantees to the aggregate of the two flows a minimum service curve $\beta$. Assume that flow $2$ is $\alpha_2$-smooth. Define the family of functions as in Proposition 6.2.1. Then the output of flow $1$ is $\alpha_1^*$-smooth, with*

$$\alpha_1^*(t) = \inf_{\theta \geq 0} \left(\alpha_1 \oslash \beta^1_\theta\right)(t)$$

**Proof:** Observe first that the network calculus output bound holds even if $\beta$ is not wide-sense increasing. Thus, from Proposition 6.2.1, we can conclude that $\alpha_1 \oslash \beta^1_\theta$ is an arrival curve for the output of flow 1. This is true for any $\theta$. □

We can apply the last proposition and obtain the following practical result.

**Theorem 6.2.2 (Burstiness Increase due to FIFO, General Case).** *Consider a node serving two flows, $1$ and $2$, in FIFO order. Assume that flow $1$ is constrained by one leaky bucket with rate $\rho_1$ and burstiness $\sigma_1$, and flow 2 is constrained by a sub-additive arrival curve $\alpha_2$. Assume that the node guarantees to the aggregate of the two flows a rate latency service curve $\beta_{R,T}$. Call $\rho_2 := \inf_{t>0} \frac{1}{t}\alpha_2(t)$ the maximum sustainable rate for flow $2$.*

*If $\rho_1 + \rho_2 < R$, then at the output, flow $1$ is constrained by one leaky bucket with rate $\rho_1$ and burstiness $b_1^*$ with*

$$b_1^* = \sigma_1 + \rho_1 \left(T + \frac{\hat{B}}{R}\right)$$

*and*

$$\hat{B} = \sup_{t \geq 0} [\alpha_2(t) + \rho_1 t - Rt]$$

*The bound is a worst case bound.*

**Proof:** (Step 1) Define $\beta_\theta^1$ as in Proposition 6.2.1. Define $B_2 = \sup_{t\geq 0} [\alpha_2(t) - Rt]$. Thus $B_2$ is the buffer that would be required if the latency $T$ would be $0$. We first show the following

$$\text{if } \theta \geq \frac{B_2}{R} + T \text{ then } \text{ for } t \geq \theta : \ \beta_\theta^1(t) = Rt - RT - \alpha_2(t-\theta) \tag{6.5}$$

To prove this, call $\phi(t)$ the right hand-side in Equation (6.5), namely, for $t \geq \theta$ define $\phi(t) = Rt - \alpha_2(t-\theta) - RT$. We have

$$\inf_{t>\theta} \phi(t) = \inf_{v>0} [Rv - \alpha_2(v) - RT + R\theta]$$

From the definition of $B_2$:

$$\inf_{t>\theta} \phi(t) = -B_2 + R\theta - RT$$

If $\theta \geq \frac{B_2}{R} + T$ then $\phi(t) \geq 0$ for all $t > \theta$. The rest follows from the definition of $\beta_\theta^1$.

(Step 2) We apply the second part of Proposition 6.2.1 with $\theta = \frac{\hat{B}}{R} + T$. An arrival curve for the output of flow $1$ is given by

$$\alpha_1^* = \lambda_{\rho_1,\sigma_1} \oslash \beta_\theta^1$$

We now compute $\alpha_1^*$. First note that obviously $\hat{B} \leq B_2$, and therefore $\beta_\theta^1(t) = Rt - RT - \alpha_2(t-\theta)$ for $t \geq \theta$. $\alpha_1^*$ is thus defined for $t > 0$ by

$$\alpha_1^*(t) = \sup_{s\geq 0} \left[\rho_1 t + \sigma_1 + \rho_1 s - \beta_\theta^1(s)\right] = \rho_1 t + \sigma_1 + \sup_{s\geq 0} \left[\rho_1 s - \beta_\theta^1(s)\right]$$

Define $\psi(s) := \rho_1 s - \beta_\theta^1(s)$. Obviously:

$$\sup_{s \in [0,\theta]} [\psi(s)] = \rho_1 \theta$$

Now from Step 1, we have

$$\begin{aligned} \sup_{s>\theta}[\psi(s)] &= \sup_{s>\theta} [\rho_1 s - Rs + RT + \alpha_2(s-\theta)] \\ &= \sup_{v>0} [\rho_1 v - Rv\alpha_2(v)] + (\rho_1 - R)\theta + RT \end{aligned}$$

From the definition of $\hat{B}$, the former is equal to

$$\sup_{s>\theta}[\psi(s)] = \hat{B} + (\rho_1 - R)\theta + RT = \rho_1\theta$$

which shows the burstiness bound in the theorem.

(Step 3) We show that the bound is attained. There is a time a $\hat{\theta}$ such that $\hat{B} = (\alpha_2)_r(\hat{\theta}) - (R - \rho_1)\hat{\theta}$. Define flow 2 to be greedy up to time $\hat{\theta}$ and stop from there on:

$$\begin{cases} R_2(t) = \alpha_2(t) \text{ for } t \leq \hat{\theta} \\ R_2(t) = (R_2)_r(\hat{\theta}) \text{ for } t > \hat{\theta} \end{cases}$$

Flow 2 is $\alpha_2$-smooth because $\alpha_2$ is sub-additive. Define flow 1 by

$$\begin{cases} R_1(t) = \rho_1 t \text{ for } t \leq \hat{\theta} \\ R_1(t) = \rho_1 t + \sigma_1 \text{ for } t > \hat{\theta} \end{cases}$$

Flow 1 is $\lambda_{\rho_1,\sigma_1}$-smooth as required. Assume the server delays all bits by $T$ at time $0$, then after time $T$ operates with a constant rate $R$, until time $\hat{\theta} + \theta$, when it becomes infinitely fast. Thus the server satisfies the required service curve property. The backlog just after time $\hat{\theta}$ is precisely $\hat{B} + RT$. Thus all flow-2 bits that arrive just after time $\hat{\theta}$ are delayed by $\frac{\hat{B}}{R} + T = \theta$. The output for flow 1 during the time interval $(\hat{\theta}+\theta, \hat{\theta}+\theta+t]$ is made of the bits that have arrived in $(\hat{\theta}, \hat{\theta}+t]$, thus there are $\rho_1 t + b_1^*$ such bits, for any $t$. □

The following corollary is an immediate consequence.

**Corollary 6.2.3 (Burstiness Increase due to FIFO).** *Consider a node serving two flows, $1$ and $2$, in FIFO order. Assume that flow $i$ is constrained by one leaky bucket with rate $\rho_i$ and burstiness $\sigma_i$. Assume that the node guarantees to the aggregate of the two flows a rate latency service curve $\beta_{R,T}$. If $\rho_1 + \rho_2 < R$, then flow $1$ has a service curve equal to the rate latency function with rate $R - \rho_2$ and latency $T + \frac{\sigma_2}{R}$ and at the output, flow $1$ is constrained by one leaky bucket with rate $\rho_1$ and burstiness $b_1^*$ with*

$$b_1^* = \sigma_1 + \rho_1 \left(T + \frac{\sigma_2}{R}\right)$$

Note that this bound is better than the one we used in Corollary 6.2.2 (but the assumptions are slightly different). Indeed, in that case, we would obtain the rate-latency service curve with the same rate $R - \rho_2$ but with a larger latency: $T + \frac{\sigma_2 + \rho_2 T}{R - \rho_2}$ instead of $T + \frac{\sigma_2}{R}$. The gain is due to the FIFO assumption.

### 6.2.3 Aggregate Multiplexing in a GR Node

We assume now that the node is of the Guaranteed Rate type (Section 2.1.3 on Page 86). A FIFO service curve element with rate-latency service curve satisfies this assumption, but the converse is not true (Theorem 2.1.2 on Page 88).

**Theorem 6.2.3.** *Consider a node serving two flows, $1$ and $2$ in some aggregate manner. Arbitration between flows is unspecified, but the node serves the aggregrate as a GR node with rate $R$ and latency $T$. Assume that flow $1$ is constrained by one leaky bucket with rate $\rho_1$ and burstiness $\sigma_1$, and flow 2 is constrained by a sub-additive arrival curve $\alpha_2$. Call $\rho_2 := \inf_{t>0} \frac{1}{t}\alpha_2(t)$ the maximum sustainable rate for flow $2$.*

*If $\rho_1 + \rho_2 < R$, then at the output, flow $1$ is constrained by one leaky bucket with rate $\rho_1$ and burstiness $b_1^*$ with*

$$b_1^* = \sigma_1 + \rho_1 \left( T + \hat{D} \right)$$

*and*

$$\hat{D} = \sup_{t>0} [\frac{\alpha_2(t) + \rho_1 t + \sigma_1}{R} - t]$$

**Proof:** From Theorem 2.1.3 on Page 88, the delay for any packet is bounded by $\hat{D} + T$. Thus an arrival curve at the output of flow 1 is $\alpha_1(t + \hat{D})$. □

**Corollary 6.2.4.** *Consider a node serving two flows, $1$ and $2$ in some aggregate manner. Arbitration between flows is unspecified, but the node serves the aggregrate as a GR node with rate $R$ and latency $T$. Assume that flow $i$ is constrained by one leaky bucket with rate $\rho_i$ and burstiness $\sigma_i$. If $\rho_1 + \rho_2 < R$, then, at the output, flow $1$ is constrained by one leaky bucket with rate $\rho_1$ and burstiness $b_1^*$ with*

$$b_1^* = \sigma_1 + \rho_1 \left( T + \frac{\sigma_1 + \sigma_2}{R} \right)$$

We see that the bound in this section is less good than Corollary 6.2.3 (but the assumptions are more general).

## 6.3 Stability and Bounds for a Network with Aggregate Scheduling

### 6.3.1 The Issue of Stability

In this section we consider the following global problem: Given a network with aggregate scheduling and arrival curve constraints at the input (as defined in the introduction) can we find good bounds for delay and backlog ? Alternatively, when is a network with aggregate scheduling stable (i.e., the backlog remains bounded) ? As it turns out today, this problem is open in many cases.

In the context of the following definition, we call "network" $\mathcal{N}$ a system satisfying the assumptions in the introduction, where all parameters except $\rho_i, \sigma_i, r_m, e_m$ are fixed. In some cases (Section 6.3.2), we may add additional constraints on these parameters.

**Definition 6.3.1 (Critical Load Factor).** *We say that $\nu_{cri}$ is the critical load factor for a network $\mathcal{N}$ if*

- *for all values of $\rho_i, \sigma_i, r_m, e_m$ such that $\nu < \nu_{cri}$, $\mathcal{N}$ is stable*
- *there exists values of $\rho_i, \sigma_i, r_m, e_m$ with $\nu > \nu_{cri}$ such that $\mathcal{N}$ is unstable.*

It can easily be checked that $\nu_{cri}$ is unique for a given network $\mathcal{N}$.

It is also easy to see that for all well defined networks, the critical load factor is $\leq 1$. However, Andrews gave in [3] an example of a FIFO network with $\nu_{cri} < 1$. The problem of finding the critical load factor, even for the simple case of a FIFO network of constant rate servers, seems to remain open. Hajek [34] shows that, in this last case, the problem can be reduced to that where every source $i$ sends a burst $\sigma_i$ instantly at time $0$, then sends at a rate limited by $\rho_i$.

In the rest of this section and in Section 6.4, we give lower bounds on $\nu_{cri}$ for some well defined sub-classes.

**Feed-Forward Networks** A feed-forward network is one in which the graph of unidirectional links has no cycle. Examples are interconnection networks used inside routers or multiprocessor machines. For a feed-forward network made of *strict* service curve element or GR nodes, $\nu_{cri} = 1$. This derives from applying the burstiness increase bounds given in Section 6.2 repeatedly, starting from network access points. Indeed, since there is no loop in the topology, the process stops and all input flows have finite burstiness.

**A Lower Bound on the Critical Load Factor** It follows immediately from Theorem 2.4.1 on Page 107 that for a network of GR nodes (or FIFO service curve elements), we have $\nu_{cri} \geq \frac{1}{h-1}$, where $h$ is the maximum hop count for any flow. A slightly better bound can be found if we exploit the values of the peak rates $C_m$ (Theorem 2.4.2).

### 6.3.2 The Time Stopping Method

For a non feed-forward network made of *strict* service curve element or GR nodes, we can find a lower bound on $\nu_{cri}$ (together with bounds on backlog or delay), using the time stopping method. It was introduced by Cruz in [20] together with bounds on backlog or delay. We illustrate the method on a specific example, shown on Figure 6.1. All nodes are constant rate servers, with unspecified arbitration between the flows. Thus we are in the case where all nodes are strict service curve elements, with service curves of the form $\beta_m = \lambda_{C_m}$.

The method has two steps. First, we assume that there is a finite burstiness bound for all flows; using Section 6.2 we obtain some equations for computing these bounds. Second, we use the same equations to show that, under some conditions, finite bounds exist.

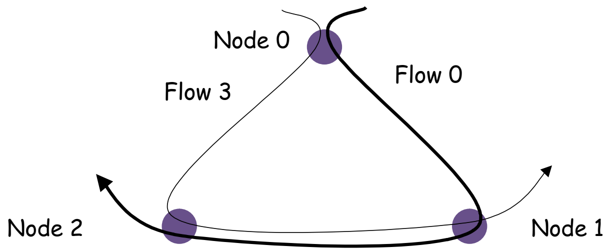



Figure 6.1: A simple example with aggregate scheduling, used to illustrate the bounding method. There are three nodes numbered $0, 1, 2$ and six flows, numbered $0, ..., 5$. For $i = 0, 1, 2$, the path of flow $i$ is $i, (i+1) \bmod 3, (i+2) \bmod 3$ and the path of flow $i+3$ is $i, (i+2) \bmod 3, (i+1) \bmod 3$. The fresh arrival curve is the same for all flows, and is given by $\alpha_i = \gamma_{\rho,\sigma}$. All nodes are constant rate, work conserving servers, with rate $C$. The utilization factor at all nodes is $6\frac{\rho}{C}$.

**First step: inequations for the bounds** For any flow $i$ and any node $m \in i$, define $\sigma_i^m$ as the maximum backlog that this flow would generate in a constant rate server with rate $\rho_i$. By convention, the fresh inputs are considered as the outputs of a virtual node numbered $-1$. In this first step, we assume that $\sigma_i^m$ is finite for all $i$ and $m \in i$.

By applying Corollary 6.2.2 we find that for all $i$ and $m \in i$:

$$\begin{cases} \sigma_i^0 \leq \sigma_i \\ \sigma_i^m = \sigma_i^{\text{pred}_i(m)} + \rho_i \frac{\sum_{j \ni m, j \neq i} \sigma_j^{\text{pred}_j(m)}}{C - \sum_{j \ni m, j \neq i} \rho_j} \end{cases} \tag{6.6}$$

where $\text{pred}_i(m)$ is the predecessor of node $m$. If $m$ is the first node on the path of flow $i$, we set by convention $\text{pred}_i(m) = -1$ and $\sigma_i^{-1} = \sigma_i$.

Now put all the $\sigma_i^m$, for all $(i, m)$ such that $m \in i$, into a vector $\vec{x}$ with one column and $n$ rows, for some appropriate $n$. We can re-write Equation (6.6) as

$$\vec{x} \leq A\vec{x} + \vec{a} \tag{6.7}$$

where $A$ is an $n \times n$, non-negative matrix and $\vec{a}$ is a non-negative vector depending only on the known quantities $\sigma_i$. The method now consists in assuming that the spectral radius of matrix $A$ is less than 1. In that case the power series $I + A + A^2 + A^3 + ...$ converges and is equal to $(I - A)^{-1}$, where $I$ is the $n \times n$ identity matrix. Since $A$ is non-negative, $(I - A)^{-1}$ is also non-negative; we can thus multiply Equation (6.6) to the left by $(I - A)^{-1}$ and obtain:

$$\vec{x} \leq (I - A)^{-1}\vec{a} \tag{6.8}$$

which is the required result, since $\vec{x}$ describes the burstiness of all flows at all nodes. From there we can obtain bounds on delays and backlogs.

Let us apply this step to our network example. By symmetry, we have only two unknowns $x$ and $y$, defined as the burstiness after one and two hops:

$$\begin{cases} x = b_0^0 = b_1^1 = \sigma_2^2 = b_3^0 = b_4^2 = b_5^1 \\ y = b_0^1 = b_1^2 = \sigma_2^0 = b_3^2 = b_4^1 = b_5^0 \end{cases}$$

Equation (6.6) becomes

$$\begin{cases} x \leq \sigma + \frac{\rho}{C-5\rho}(\sigma + 2x + 2y) \\ y \leq x + \frac{\rho}{C-5\rho}(2\sigma + x + 2y) \end{cases}$$

Define $\eta = \frac{\rho}{C-5\rho}$; we assume that the utilization factor is less than 1, thus $0 \leq \eta < 1$. We can now write Equation (6.7) with

$$\vec{x} = \begin{pmatrix} x \\ y \end{pmatrix},\ A = \begin{pmatrix} 2\eta & 2\eta \\ 1+\eta & 2\eta \end{pmatrix},\ \vec{a} = \begin{pmatrix} \sigma(1+\eta) \\ 2\sigma\eta \end{pmatrix}$$

Some remnant from linear algebra, or a symbolic computation software, tells us that

$$(I - A)^{-1} = \begin{pmatrix} \frac{1-2\eta}{1-6\eta+2\eta^2} & \frac{2\eta}{1-6\eta+2\eta^2} \\ \frac{1+\eta}{1-6\eta+2\eta^2} & \frac{1-2\eta}{1-6\eta+2\eta^2} \end{pmatrix}$$

If $\eta < \frac{1}{2}(3 - \sqrt{7}) \approx 0.177$ then $(I - A)^{-1}$ is positive. This is the condition for the spectral radius of $A$ to be less than 1. The corresponding condition on the utilization factor $\nu = \frac{6\rho}{C}$ is

$$\nu < 2\frac{8 - \sqrt{7}}{19} \approx 0.564 \tag{6.9}$$

Thus, for this specific example, if Equation (6.9) holds, and if the burstiness terms $x$ and $y$ are finite, then they are bounded as given in Equation (6.8), with $(I - A)^{-1}$ and $\vec{a}$ given above.

**Second Step: time stopping** We now prove that there is a finite bound if the spectral radius of $A$ is less than 1. For any time $\tau > 0$, consider the virtual system made of the original network, where all sources are stopped at time $\tau$. For this network the total number of bits in finite, thus we can apply the conclusion of step 1, and the burstiness terms are bounded by Equation (6.8). Since the right-handside Equation (6.8) is independent of $\tau$, letting $\tau$ tend to $+\infty$ shows the following.

**Proposition 6.3.1.** *With the notation in this section, if the spectral radius of $A$ is less than $1$, then the burstiness terms $b_i^m$ are bounded by the corresponding terms in Equation (6.8).*

Back to the example of Figure 6.1, we find that if the utilization factor $\nu$ is less than $0.564$, then the burstiness terms $x$ and $y$ are bounded by

$$\begin{cases} x \leq 2\sigma\frac{18-33\nu+16\nu^2}{36-96\nu+57\nu^2} \\ y \leq 2\sigma\frac{18-18\nu+\nu^2}{36-96\nu+57\nu^2} \end{cases}$$

The aggregate traffic at any of the three nodes is $\gamma_{6\rho,b}$-smooth with $b = 2(\sigma+x+y)$. Thus a bound on delay is (see also Figure 6.2):

$$d = \frac{b}{C} = 2\frac{\sigma}{C}\frac{108-198\nu+91\nu^2}{36-96\nu+57\nu^2}$$

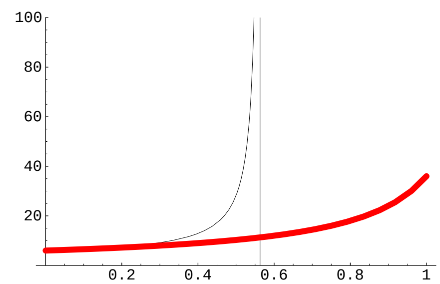



Figure 6.2: The bound $d$ on delay at any node obtained by the method presented here for the network of Figure 6.1 (thin line). The graph shows $d$ normalized by $\frac{\sigma}{C}$ (namely, $\frac{dC}{\sigma}$), plotted as a function of the utilization factor. The thick line is a delay bound obtained if every flow is re-shaped at every output.

**The critical load factor for this example** For the network in this example, where we impose the constraint that all $\rho_i$ are equal, we find $\nu_{cri} \geq \nu_0 \approx 0.564$, which is much less than $1$. Does it mean that no finite bound exists for $\nu_0 \leq \nu < 1$ ? The answer to this question is not clear.

First, the $\nu_0$ found with the method can be improved if we express more arrival constraints. Consider our particular example: we have not exploited the fact that the fraction of input traffic to node $i$ that originates from another node has to be $\lambda_C$-smooth. If we do so, we will obtain better bounds. Second, if we know that nodes have additional properties, such as FIFO, then we may be able to find better bounds. However, even so, the value of $\nu_{cri}$ seems to be unknown.

**The price for aggregate scheduling** Consider again the example on Figure 6.1, but assume now that every flow is reshaped at every output. This is not possible with differentiated services, since there is no per-flow information at nodes other than access nodes. However, we use this scenario as a benchmark that illustrates the price we pay for aggregate scheduling.

With this assumption, every flow has the same arrival curve at every node. Thus we can compute a service curve $\beta_1$ for flow 1 (and thus for any flow) at every node, using Theorem 6.2.1; we find that $\beta_1$ is the rate-latency function with rate $(C - 5\rho)$ and latency $\frac{5\sigma}{C-5\rho}$. Thus a delay bound for flow at any node, including the re-shaper, is $h(\alpha_1, \alpha_1 \otimes \beta_1) = h(\alpha_1, \beta_1) = \frac{6C}{C-5\rho}$ for $\rho \leq \frac{C}{6}$. Figure 6.2 shows this delay bound, compared to the delay bound we found if no reshaper is used. As we already know, we see that with per-flow information, we are able to guarantee a delay bound for any utilization factor $\leq 1$. However, note also that for relatively small utilization factors, the bounds are very close.

## 6.4 Stability Results and Explicit Bounds

In this section we give strong results for two specific case. The former is for a unidirectional ring of aggregate servers (of any type, not necessarily FIFO or strict service curve). We show that for all rings, $\nu_{cri} = 1$. The latter is for any topology, but with restrictions on the network type: packets are of fixed size and all links have the same bit rate.

### 6.4.1 The Ring is Stable

The result was initially obtained in [73] for the case of a ring of constant rate servers, with all servers having the same rate. We give here a more general, but simpler form.

**Assumption and Notation** We take the same assumptions as in Section 6.1 and assume in addition that the network topology is a unidirectional ring. More precisely:

- The network is a unidirectional ring of $M$ nodes, labelled $1, ..., M$. We use the notation $m \oplus k = (m + k - 1) \bmod M + 1$ and $m \ominus k = (m - k - 1) \bmod M + 1$, so that the successor of node $m$ on the ring is node $m \oplus 1$ and its predecessor is node $m \ominus 1$.

- The route of flow $i$ is $(0, i.\text{first}, i.\text{first} \oplus 1, ..., i.\text{first} \oplus (h_i - 1))$ where $0$ is a virtual node representing the source of flow $i$, $i.\text{first}$ is the first hop of flow $i$, and $h_i$ is the number of hops of flow $i$. At its last hop, flow $i$ exits the network. We assume that a flow does not wrap, namely, $h_i \leq M$. If $h_i = M$, then the flow goes around the all ring, exiting at the same node it has entered.

- Let $b = \sum_m b_m$ reflect the total latency of the ring.

- For any node $m$ let $\sigma^{(m)} = \sum_{i \ni m} \sigma_i$.
  Let $\sigma_{\max} = \max_{m=1}^{M} \sigma^{(m)}$ and $\sigma = \sum_i \sigma_i$. Note that $\sigma_{\max} \leq \sigma \leq M\sigma_{\max}$.
- Define $\eta = \min_m (r_m - \rho^{(m)})$.
- Let $\rho_0^{(m)} = \sum_{i.\text{first}=m} \rho_i$ and $\mu = \max_{m=0}^{M} \left[C_m - r_m + \rho^{(m)}\right]^+$. $\mu$ reflects the sum of the peak rate of transit links and the rates of fresh sources, minus the rate guaranteed to the aggregate of microflows. We expect high values of $\mu$ to give higher bounds.

**Theorem 6.4.1.** *If $\eta > 0$ (i.e. if the utilization factor is $< 1$) the backlog at any node of the unidirectional ring is bounded by*

$$M\frac{\mu}{\eta}\left(M\sigma_{\max} + b\right) + \sigma + b$$

**Proof:** The proof relies on the concept of chain of busy periods, combined with the time stopping method in Section 6.3.2.

For a node $m$ and a flow $i$, define $R_i^m(t)$ as the cumulative amount of data of flow $i$ at the output of node $m$. For $m = 0$, this defines the input function. Also define

$$x_m(t) = \sum_{i \ni m} \left(R_i^0(t) - R_i^m(t)\right) \tag{6.10}$$

thus $x_m(t)$ is the total amount of data that is present in the network at time $t$ and will go through node $m$ at some time $> t$.

We also define the backlog at node $m$ by

$$q_m(t) = \sum_{i \ni m, i.\text{first} \neq m} R_i^{m \ominus 1}(t) + \sum_{i.\text{first}=m} R_i^0(t) - \sum_{i \ni m} R_i^m(t)$$

Now obviously, for all time $t$ and node $m$:

$$q_m(t) \leq x_m(t) \tag{6.11}$$

and

$$x_m(t) \leq \sum_{n=1}^{M} q_n(t) \tag{6.12}$$

(Step 1) Assume that a finite bound $X$ exists. Consider a time $t$ and a node $m$ that achieves the bound: $x_m(t) = X$. We fix $m$ and apply Lemma 6.4.1 to all nodes $n$. Call $s_n$ the time called $s$ in the lemma. Since $x_n(s_n) \leq X$, it follows from the first formula in the lemma that

$$(t - s_n)\eta \leq M\sigma_{\max} + b \tag{6.13}$$

By combining this with the second formula in the lemma we obtain

$$q_n(t) \leq \mu \frac{M\sigma_{\max} + b}{\eta} + b_n + \sigma_0^{(n)}$$

Now we apply Equation (6.12) and note that $\sum_{n=1}^{M} \sigma_0^{(n)} = \sigma$, from which we derive

$$X \leq M\frac{\mu}{\eta}\left(M\sigma_{\max} + b\right) + \sigma + b \tag{6.14}$$

(Step 2) By applying the same reasoning as in Section 6.3.2, we find that Equation (6.14) is always true. The theorem follows from Equation (6.11). □

**Lemma 6.4.1.** *For any nodes $m, n$ (possibly with $m = n$), and for any time $t$ there is some $s$ such that*

$$\begin{cases} x_m(t) \leq x_n(s) - (t-s)\eta + M\sigma_{\max} + b \\ q_n(t) \leq (t-s)\mu + b_n + \sigma_0^{(n)} \end{cases}$$

*with* $\sigma_0^{(n)} = \sum_{i.\text{first}=n} \sigma_i$.

**Proof:** By definition of the service curve property at node $m$, there is some $s_1$ such that

$$\sum_{i \ni m} R_i^m(t) \geq \sum_{i \ni m, i.\text{first} \neq m} R_i^{m \ominus 1}(s_1) + \sum_{i.\text{first}=m} R_i^0(s_1) + r_m(t - s_1) - b_m$$

which we can rewrite as

$$\sum_{i \ni m} R_i^m(t) \geq -A + \sum_{i \ni m} R_i^0(s_1) + r_m(t - s_1) - b_m$$

with

$$A = \sum_{i \ni m, i.\text{first} \neq m} \left(R_i^0(s_1) - R_i^{m-1}(s_1)\right)$$

Now the condition $\{i \ni m, i.\text{first} \neq m\}$ implies that flow $i$ passes through node $m-1$, namely, $\{i \ni (m-1)\}$. Furthermore, each element in the summation that constitutes $A$ is nonnegative. Thus

$$A \leq \sum_{i \ni (m-1)} \left(R_i^0(s_1) - R_i^{m-1}(s_1)\right) = x_{m \ominus 1}(s_1)$$

Thus

$$\sum_{i \ni m} R_i^m(t) \geq -x_{m \ominus 1}(s_1) + \sum_{i \ni m} R_i^0(s_1) + r_m(t - s_1) - b_m \tag{6.15}$$

Now combining this with the definition of $x_m(t)$ in Equation (6.10) gives:

$$x_m(t) \leq x_{m \ominus 1}(s_1) + \sum_{i \ni m} \left(R_i^0(t) - R_i^0(s_1)\right) - r_m(t - s_1) + b_m$$

From the arrival curve property applied to all micro-flows $i$ in the summation, we derive:

$$x_m(t) \leq x_{m\ominus 1}(s_1) - (r_m - \rho^{(m)})(t - s_1) + \sigma^{(m)} + b_m$$

and since $r_m - \rho^{(m)} \geq \eta$ and $\sigma^{(m)} \leq \sigma_{\max}$ by definition of $\eta$ and $\sigma_{\max}$, we have

$$x_m(t) \leq x_{m\ominus 1}(s_1) - (t - s_1)\eta + \sigma_{\max} + b_m$$

We apply the same reasoning to node $m \ominus 1$ and time $s_1$, and so on iteratively until we reach node $n$ backwards from $m$. We thus build a sequence of times $s_0 = t, s_1, s_2, ..., s_j, ..., s_k$ such that

$$x_{m\ominus j}(s_j) \leq x_{m\ominus(j+1)}(s_{j+1}) - (t - s_{j+1})\eta + \sigma_{\max} + b_{m\ominus j} \tag{6.16}$$

until we have $m \ominus k = n$. If $n = m$ we reach the same node again by a complete backwards rotation and $k = M$. In all cases, we have $k \leq M$. By summing Equation (6.16) for $j = 0$ to $k - 1$ we find the first part of the lemma.

Now we prove the second part. $s = s_k$ is obtained by applying the service curve property to node $n$ and time $s_{k-1}$. Apply the service curve property to node $n$ and time $t$. Since $t \geq s_{k-1}$, we know from Proposition 1.3.2 on Page 24 that we can find some $s' \geq s$ such that

$$\sum_{i \ni n} R_i^n(t) \geq \sum_{i \ni n, i.\text{first} \neq n} R_i^{n-1}(s') + \sum_{i.\text{first}=n} R_i^0(s') + r_n(t - s') - b_n$$

Thus

$$\begin{aligned} q_n(t) \leq & \sum_{i \ni n, i.\text{first} \neq n} \left(R_i^{n\ominus 1}(t) - R_i^{n\ominus 1}(s')\right) + \\ & \sum_{i.\text{first}=n} (R_i^0(t) - R_i^0(s')) - r_n(t - s') + b_n \\ & \leq (C_n - r_n + \rho_0^{(n)})(t - s') + b_n + \sigma_0^{(n)} \leq (t - s')\mu + b_n + \sigma_0^{(n)} \end{aligned}$$

the second part of the formula follows from $s \leq s'$. □

**Remark:** A simpler, but weaker bound, is

$$M\frac{\mu}{\eta}(M\sigma + b) + \sigma + b$$

or

$$M\frac{\mu}{\eta}(M\sigma_{\max} + b) + M\sigma_{\max} + b \tag{6.17}$$

**The special case in [73]:** Under the assumption that all nodes are constant rate servers of rate equal to $1$ (thus $C_m = r_m = 1$ and $b_m$ is the latency of the link $m$), the following bound is found in [73]:

$$B_1 = \frac{Mb + M^2 \sigma_{\max}}{\eta} + b \tag{6.18}$$

In that case, we have $\mu \leq 1 - \eta$. By applying Equation (6.17), we obtain the bound

$$B_2 = \frac{M\mu b + \left[M^2 \mu + M\eta\right] \sigma_{\max}}{\eta} + b$$

since

$$\mu \leq 1 - \eta \tag{6.19}$$

and $0 < \eta \leq 1,\ M \leq M^2$, we have $B_2 < B_1$, namely, our bound is better than that in [73]. If there is equality in Equation (6.19) (namely, if there is a node that receives no transit traffic), then both bounds are equivalent when $\eta \to 0$.

### 6.4.2 Explicit Bounds for a Homogeneous ATM Network with Strong Source Rate Conditions

When analyzing a global network, we can use the bounds in Section 6.2.2, using the same method as in Section 2.4. However, as illustrated in [38], the bounds so obtained are not optimal: indeed, even for a FIFO ring, the method does *not* find a finite bound for all utilization factors less than (although we know from Section 6.4.1 that such finite bounds exist).

In this section we show in Theorem 6.4.2 some partial result that goes beyond the per-node bounds in Section 6.2.2. The result was originally found in [14, 49, 79].

Consider an ATM network with the assumptions as in Section 6.1, with the following differences

- Every link has one origin node and one end node. We say that a link $f$ is incident to link $e$ if the origin node of link $e$ is the destination node of link $f$. In general, a link has several incident links.

- All packets have the same size (called cell). All arrivals and departures occur at integer times (synchronized model). All links have the same bit rate, equal to $1$ cell per time unit. The service time for one cell is $1$ time unit. The propagation times are constant per link and integer.

- All links are FIFO.

**Proposition 6.4.1.** *For a network with the above assumption, the delay for a cell $c$ arriving at node $e$ over incident link $i$ is bounded by the number of cells arriving on incident links $j \neq i$ during the busy period, and that will depart before $c$.*

**Proof:** Call $R'(t)$ (resp. $R_j(t), R(t)$)the output flow (resp. input arriving on link $j$, total input flow). Call $d$ the delay for a tagged cell arriving at time $t$ on link $i$. Call $A_j$ the number of cells arriving on link $j$ up to time $t$ that will depart before the tagged cell, and let $A = \sum_j A_j$. We have

$$d = A - R'(t) \leq A - R(s) - (t - s)$$

where $s$ is the last time instant before the busy period at $t$. We can rewrite the previous equation as

$$d \leq \sum_{j \neq i} [A_j - R_j(s)] + [A_i(t) - R_i(s)] - (t - s)$$

Now the link rates are all equal to 1, thus $A_i - R_i(s) \leq t - s$ and

$$d \leq \sum_{j \neq i} [A_j - R_j(s)]$$

□

An "Interference Unit" is defined as a set $(e, \{j, k\})$ where $e$ is a link, $\{j, k\}$ is a set of two distinct flows that each have $e$ on their paths, and that arrive at $e$ over two different incident links (Figure 6.3). The Route Interference Number (RIN) of flow $j$ is the number of interference units that contain $j$. It is thus the number of other flows that share a common sub-path, counted with multiplicity if some flows share several distinct sub-paths along the same path. The RIN is used to define a sufficient condition, under which we prove a strong bound.

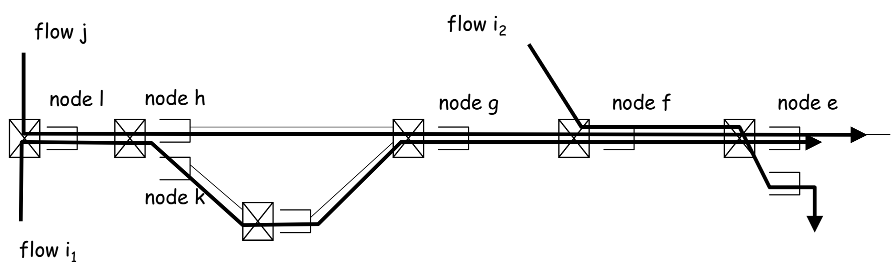



Figure 6.3: The network model and definition of an interference unit. Flows $j$ and $i_2$ have an interference unit at node $f$. Flows $j$ and $i_1$ have an interference unit at node $l$ and one at node $g$.

**Definition 6.4.1 (Source Rate Condition).** *The fresh arrival curve constraint (at network boundary) for flow $j$ is the stair function $v_{R+1,R+1}$, where $R$ is the RIN of flow $j$.*

The source rate condition is equivalent to saying that a flow generates at most one cell in any time interval of duration $\mathrm{RIN} + 1$.

**Theorem 6.4.2.** *If the source rate condition holds at all sources, then*

1. *The backlog at any node is bounded by $N - \max_i N_i$, where $N_i$ is the number of flows entering the node via input link $i$, and $N = \sum_i N_i$.*
2. *The end-to-end queuing delay for a given flow is bounded by its RIN.*
3. *There is at most one cell per flow present during any busy period.*

The proof of item 3 involves a complex analysis of chained busy periods, as does the proof of Theorem 6.4.1. It is given in a separate section. Item 3 gives an intuitive explanation of what happens: the source rate condition forces sources to leave enough spacing between cells, so that two cells of the same flow do not interfere, in some sense. The precise meaning of this is given in the proof. Items 1 and 2 derive from item 3 by a classical network calculus method (Figure 6.6).

**Proof of Theorem 6.4.2** As a simplification, we call “path of a cell“ the path of the flow of the cell. Similarly, we use the phrase “interference unit of $c$” with the meaning of interference unit of the flow of $c$.

We define a busy period as a time interval during which the backlog for the flow at the node is always positive. We now introduce a definition (super-chain) that will be central in the proof. First we use the following relation:

**Definition 6.4.2 (“Delay Chain” [14]).** *For two cells $c$ and $d$, and for some link $e$, we say that $c \preccurlyeq_e d$ if $c$ and $d$ are in the same busy period at $e$ and $c$ leaves $e$ before $d$.*

Figure 6.4 illustrates the definition.

**Definition 6.4.3 (Super-Chain [14]).** *Consider a sequence of cells $\underline{c} = (c_0, ..., c_i, ..., c_k)$ and a sequence of nodes $\underline{f} = (f_1, ..., f_k)$. We say that $(\underline{c}, \underline{f})$ is a super-chain if*

- *$f_1, ..., f_k$ are all on the path $P$ of cell $c_0$ (but not necessarily consecutive)*
- *$c_{i-1} \preccurlyeq_{f_i} c_i$ for $i = 1$ to $k$.*
- *the path of cell $c_i$ from $f_i$ to $f_{i+1}$ is a sub-path of $P$*

We say that the sub-path of $c_0$ that spans from node $f_1$ to node $f_k$ is the path of the super-chain.

**Definition 6.4.4 (Segment Interfering with a Super-Chain).** *For a given super-chain, we call “segment” a couple $(d, P)$ where $P$ is a sub-path of the path of the super-chain, $d$ is a cell whose path also has $P$ as a sub-path, and $P$ is maximal (namely, we cannot extend $P$ to be a common sub-path of both $d$ and the super-chain). We say that the segment $(d, P)$ is interfering with super-chain $(\underline{c}, \underline{f})$ if there is some $i$ on $P$ such that $d \preccurlyeq_{f_i} c_i$.*

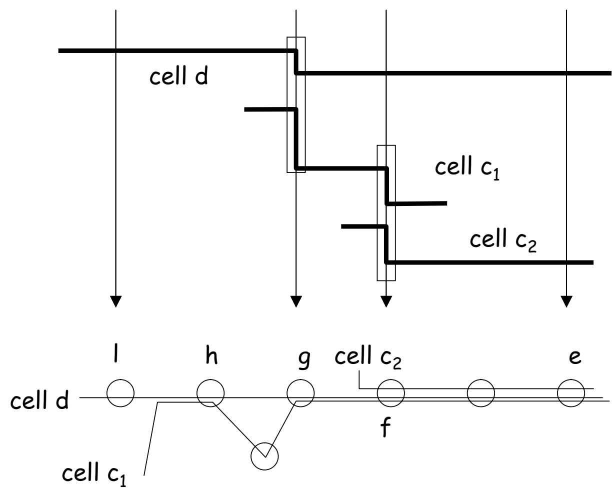



Figure 6.4: A time-space diagram illustrating the definitions of $d \preccurlyeq_g c_1$ and $c_1 \preccurlyeq_f c_2$. Time flows downwards. Rectangles illustrate busy periods.

**Lemma 6.4.2.** *Let $(\underline{c}, \underline{f})$ be a super-chain. Let $s_0$ be the arrival time of cell $c_0$ at link $f_1$ and $s'_k$ the departure time of cell $c_k$ from link $f_k$. Then $s'_k - s_0 \leq R_{1,k} + T_{1,k}$, where $R_{1,k}$ is the total number of segments interfering with $(\underline{c}, \underline{f})$ and $T_{1,k}$ is the total transmission and propagation time on the path of the super-chain.*

**Proof:** Consider first some node $f_j$ on the super-chain. Let $s_{j-1}$ (resp. $t_j$) be the arrival time of cell $c_{j-1}$ (resp. $c_j$) at the node. Let $t'_{j-1}$ (resp. $s'_j$) be the departure time of cell $c_{j-1}$ (resp. $c_j$) (Figure 6.5). Let $v_j$ be the last time slot before the busy period that $t_j$ is in. By hypothesis, $v_j + 1 \leq s_{j-1}$. Also define $\mathcal{B}_j$ (resp. $\mathcal{B}^0_j$) as the set of segments $(d, P)$ where $d$ is a cell arriving at the node after time $v_j$ on a link incident to the path of the super-chain (resp. on the path of the super-chain) and that will depart no later than cell $c_j$, and where $P$ is the maximal common sub-path for $d$ and the super-chain that $f_j$ is in. Also define $\mathcal{A}^0_j$ as the subset of those segments in $\mathcal{B}^0_j$ for which the cell departs after $c_{j-1}$. Let $B_j$ (resp. $B^0_j, A^0_j$) be the number of elements in $\mathcal{B}_j$ (resp. $\mathcal{B}^0_j, \mathcal{A}^0_j$), see Figure 6.5.

Since the rate of all incident links is 1, we have

$$B^0_j - A^0_j \leq s_{j-1} - v_j$$

Also, since the rate of the node is 1, we have:

$$s'_j - v_j = B_j + B^0_j$$

Combining the two, we derive

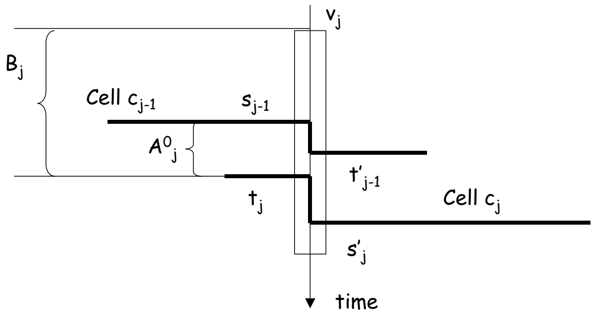



Figure 6.5: The notation used in the proof of Lemma 6.4.2.

$$s'_j - s_{j-1} = B_j + B_j^0 - (s_{j-1} - v_j) \leq B_j + A_j^0 \tag{6.20}$$

By iterative application of Equation (6.20) from $j = 1$ to $k$, we obtain

$$s'_k - s_0 \leq \sum_{j=1}^{k} (B_j + A_j^0) + T_{1,k}$$

Now we show that all sets in the collection $\{\mathcal{B}_j, \mathcal{A}_j^0,\ j = 1 \text{ to } k\}$ are two-by-two disjoint. Firstly, if $(d, P) \in \mathcal{B}_j$ then $f_j$ is the first node of $P$ thus $(d, P)$ cannot be in some other $\mathcal{B}_{j'}$ with $j \neq j'$. Thus the $\mathcal{B}_j$ are two-by-two disjoint. Second, assume $(d, P) \in \mathcal{B}_j$ and $(d, P) \in \mathcal{A}_{j'}^0$. It is obvious from their definitions that, for a fixed $j$, $\mathcal{B}_j$ and $\mathcal{A}_j^0$ are disjoint; thus $j \neq j'$. Since $f_j$ is the first node of $P$ and $j'$ is on $P$, it follows that $j < j'$. Now $d$ leaves $f_j$ before $c_j$ and leaves $f_{j'}$ after $c_{j'-1}$, which contradicts the FIFO assumption. Thus the $\mathcal{B}_j$ and $\mathcal{A}_{j'}^0$ are two-by-two disjoint. The same reasoning shows that it is not possible that $(d, P) \in \mathcal{A}_j \bigcap \mathcal{A}_{j'}$ with $j < j'$.

Now, by definition, every segment in either $\mathcal{B}_j$ or $\mathcal{A}_j^0$ is an interfering segment. Thus

$$\sum_{j=1}^{k} (B_j + A_j^0) \leq R_{1,k}$$

. $\square$

**Proposition 6.4.2.** *Assume the source rate condition holds. Let $(\underline{c}, \underline{f})$ be a super-chain.*

1. *For every interference unit of $c_0$ there is at most one cell interfering with the super-chain.*

2. *$c_k$ does not belong to the same flow as $c_0$.*

**Proof:** Define the time of a super-chain as the exit time for the last cell $c_k$ on the last node $f_k$. We use a recursion on the time $t$ of the super-chain.

If $t = 1$, the proposition is true because any flow has at most one cell on a link in one time slot. Assume now that the proposition holds for any super-chain with time $\leq t-1$ and consider a super-chain with time $t$.

First, we associate an interference unit to any segment $(d, P)$ interfering with the sub-chain, as follows. The paths of $d$ and $c_0$ may share several non contiguous sub-paths, and $P$ is one of them. Call $f$ the first node of $P$. To $d$ we associate the interference unit $(f, \{j_0, j\})$, where $j_0$ (resp. $j$) is the flow of $c_0$ (resp. $d$).

We now show that this mapping is injective. Assume that another segment $(d', P') \neq (d, P)$ is associated with the same interference unit $(f, \{j_0, j\})$. Without loss of generality, we can assume that $d$ was emitted before $d'$. $d$ and $d'$ belong to the same flow $j$, thus, since $P$ and $P'$ are maximal, we must have $P = P'$. By hypothesis, have an interference with the super-chain at a node on $P$. Let $f_l$ be a node on the super-chain and on $P$ such that $d \preccurlyeq_{f_l} c_l$. If $d'$ leaves node $f_l$ before $c_l$, then $d \preccurlyeq_{f_l} d'$, and thus $((d, d'), (f_l))$ is a super-chain. Since $d'$ is an interfering cell, necessarily, it must leave node $f_l$ before $t$, thus the proposition is true for super-chain $((d, d'), (f_l))$, which contradicts item 2. Thus $d'$ must leave node $f_l$ after cell $c_l$. But there is some other index $m \leq k$ such that $d \preccurlyeq_{f_m} c_m$, thus cell $d'$ leaves node $f_m$ before cell $c_m$. Define $l'$ as the smallest index with $l < l' \leq m$ such that $d'$ leaves node $f_{l'}$ after cell $c_{l'-1}$ and before $c_{l'}$. Then $((d, c_l, ..., c_{l'-1}, d'), (f_l, .., f_{l'}))$ is a super-chain with time $\leq t-1$ which would again contradict item 2 in the proposition. Thus, in all cases we have a contradiction, the mapping is injective, and item 1 is shown for the super-chain.

Second, let us count a bound on the maximum queuing delay of cell $c_0$. Call $u_0$ its emission time, $P_0$ the sub-path of $c_0$ from its source up to, but excluding, node $f_1$, and $T$ the total transmission and propagation time for the flow of $c_0$. The transmission and propagation time along $P_0$ is thus $T - T_{1,k}$. By Proposition 6.4.1, the queuing delay of $c_0$ at a node $f$ on $P_0$ is bounded by the number of cells $d \preccurlyeq_f c_0$ that arrive on a link not on $P_0$. By the same reasoning as in the previous paragraph, there is at most one such cell $d$ per interference unit of $c_0$ at $f$. Define $R$ as the number of interference units of the flow of $c_0$ on $P_1$. We have thus

$$s_0 \leq u_0 + R + T - T_{1,k} \tag{6.21}$$

Similarly, from Lemma 6.4.2, we have

$$s'_k \leq s_0 + R_{1,k} + T_{1,k}$$

Call $R'$ the number of interference units of the flow of $c_0$ on the path of the super-chain. It follows from the first part of the proof that $R_{1,k} \leq R'$, thus

$$s'_k \leq s_0 + R' + T_{1,k}$$

Combining with Equation (6.21) gives

$$s'_k \leq u_0 + R + R' + T \tag{6.22}$$

Now by the source condition, if $c_k$ belongs to the flow of $c_0$, its emission time $u'$ must satisfy

$$u' \geq u_0 + R + R' + 1$$

and thus

$$s'_k \geq u_0 + R + R' + 1 + T$$

which contradicts Equation (6.22). This shows that the second item of the proposition must hold for the super-chain. □

**Proof of Theorem 6.4.2:** Item 3 follows from Proposition 6.4.2, since if there would be two cells $d, d'$ of the same flow in the same busy period, then $((d, d'), (e))$ would be a super-chain.

Now we show how items 1 and 2 derive from item 3. Call $\alpha_i^*(t)$ the maximum number of cells that may ever arrive on incident link $i$ during $t$ time units inside a busy period. Since $\lambda_1$ is a service curve for node $e$, the backlog $B$ at node $e$ is bounded by

$$B \leq \sup_{t \geq 0} \left[ \sum_{i=1}^{I} \alpha_i^*(t) \ - t \right]$$

Now by item 3, $\alpha_i^*(t) \leq N_i$ and thus

$$\alpha_i^*(t) \leq \alpha_i(t) := \min[N_i, t]$$

Thus

$$B \leq \sup_{t \geq 0} \left[ \sum_{i=1}^{I} \alpha_i(t) \ - t \right]$$

Now define a renumbering of the $N_i$'s such that $N_{(1)} \leq N_{(2)} \leq ... \leq N_{(I)}$. The function $\sum_i \alpha_i(t) - t$ is continuous and has a derivative at all points except the $N_{(i)}$'s (Figure 6.6). The derivative changes its sign at $N_{(I)}$ (=$\max_{1 \leq i \leq I}(N_i)$) thus the maximum is at $N_{(I)}$ and its value is $N - N_{(I)}$, which shows item 1.

From Item 1, the delay at a node is bounded by the number of interference units of the flow at this node. This shows item 2. □

## 6.5 Bibliographic Notes

In [49], a stronger property is shown than Theorem 6.4.2: Consider a given link $e$ and a subset $A$ of $m$ connections that use that link. Let $n$ be a lower bound on the number of route interferences that any connection in the subset will encounter after this link. Then over any time interval of duration $m + n$, the number of cells belonging to $A$ that leave link $e$ is bounded by $m$.

It follows from item 1 in Theorem 6.4.2 that a better queuing delay bound for flow $j$ is:

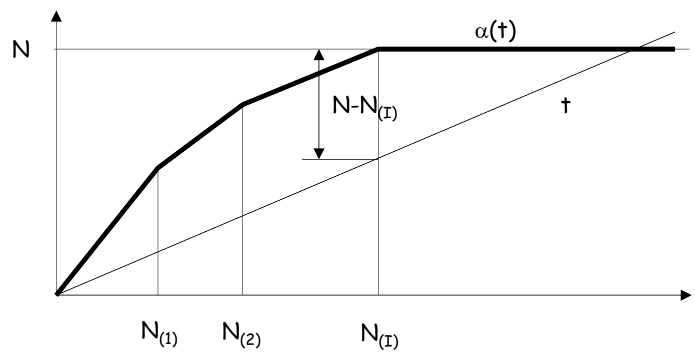



Figure 6.6: Derivation of a backlog bound.

$$\delta(j) = \sum_{e \text{ such that } e \in j} \left\{ \min_{i \text{ such that } 1 \leq i \leq I(e)} (N(e) - N_i(e)) \right\}$$

where $I(e)$ is the number of incident links at node $e$, $N_i(e)$ is the number of flows entering node $e$ on link i, and $N = \sum i = 1^{I(e)} N_i(e)$. In other words, the end-to-end queuing delay is bounded by the sum of the minimum numbers of route interference units for all flows at all nodes along the path of a flow. For asymmetric cases, this is less than the RIN of the flow.

## 6.6 Exercises

**Exercise 6.1.** *Consider the same assumptions as in Section 6.4.1 but with a linear network instead of a ring. Thus node $m$ feeds node $m+1$ for $m = 1, ..., M-1$; node $1$ receives only fresh traffic, whereas all traffic exiting node $M$ leaves the network. Assume that all service curves are strict. Find a bound which is finite for $\nu \leq 1$. Compare to Theorem 6.4.1.*

**Exercise 6.2.** *Consider the same assumptions as in Theorem 6.4.2. Show that the busy period duration is bounded by $N$.*

**Exercise 6.3.** *Consider the example of Figure 6.1. Apply the method of Section 6.3.2 but express now that the fraction of input traffic to node $i$ that originates from another node must have $\lambda_C$ as an arrival curve . What is the upper-bound on utilization factors for which a bound is obtained ?*

**Exercise 6.4.** *Can you conclude anything on $\nu_{cri}$ from Proposition 2.4.1 on Page 110 ?*

# Chapter 7

# Adaptive and Packet Scale Rate Guarantees

## 7.1 Introduction

In Chapter 1 we defined a number of service curve concepts: minimum service curve, maximum service curve and strict service curves. In this chapter we go beyond and define some concepts that more closely capture the properties of generalized processor sharing (GPS).

We start by a motivating section, in which we analyze some features of service curves that do not match GPS. Then we provide the theoretical framework of adaptive guarantees, which was first proposed in Okino's dissertation in [59] and by Agrawal, Cruz, Okino and Rajan in [1]. This framework is underlying the concept of packet scale rate guarantees, which is used in the definition of the Internet Expedited Forwarding service. We explain the relationship between the two and give practical applications.

In all of this chapter, we assume that flow functions are left-continuous, unless stated otherwise.

## 7.2 Adaptive Guarantee

### 7.2.1 Limitations of the Service Curve Abstraction

The definition of service curve introduced in Section 1.3 is an abstraction of nodes such as GPS and its practical implementations, as well as guaranteed delay nodes. This abstraction is used in many situations, described all along this book. However, it is not always sufficient.

Firstly, it does not provide a guarantee over any interval. Consider for example a node offering to a flow $R(t)$ the service curve $\lambda_C$. Assume $R(t) = B$ for $t > 0$, so the flow has a very large burst at time $0$ and then stops. A possible output is illustrated on Figure 7.1. It is perfectly possible that there is no output during the time interval $(0, \frac{B-\epsilon}{C}]$, even though there is a large backlog. This is because the server gave a higher service than the minimum required during some interval of time, and the service property allows it to be lazy after that.

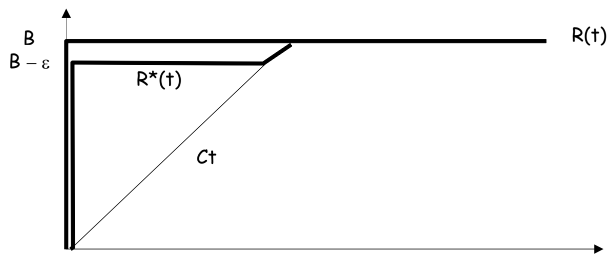



Figure 7.1: The service curve property is not sufficient.

Secondly, there are case where we would like to deduce a bound on the delay that a packet will suffer given the backlog that we can measure in the node. This is used for obtaining bounds in FIFO systems with aggregate scheduling. In Chapter 6 we use such a property for a constant delay server with rate $C$: given that the backlog at time $t$ is $Q$, the last bit present at time $t$ will depart before within a time of $\frac{Q}{C}$. If we assume instead that the server has a service curve $\lambda_C$, then we cannot draw such a conclusion. Consider for example Figure 7.1: at time $t > 0$, the backlog, $\epsilon$, can be made arbitrily small, whereas the delay $\frac{B-\epsilon}{C} - t$ can be made arbitrarily large.

A possible fix is the use of *strict service curve*, as defined in Definition 1.3.2 on Page 27. Indeed, it follows from the next section (and can easily be shown independently) that if a FIFO node offers a strict service curve $\beta$, then the delay at time $t$ is bounded by $\beta^{-1}(Q(t))$, where $Q(t)$ is the backlog at time $t$, and $\beta^{-1}$ is the pseudo-inverse (Definition 3.1.7 on Page 129).

We know that the GPS node offers to a flow a strict service curve equal of the form $\lambda_R$. However, we cannot model delay nodes with a strict service curve. Consider for example a node with input $R(t) = \epsilon t$, which delays all bits by a constant time $d$. Any interval $[s, t]$ with $s \geq d$ is within a busy period, thus if the node offers a strict service curve $\beta$ to the flow, we should have $\beta(t-s)\epsilon(t-s)$, and $\epsilon$ can be arbitrarily small. Thus, the strict service curve does not make much sense for a constant delay node.

### 7.2.2 Definition of Adaptive Guarantee

We know introduce a stronger concept, called *adaptive guarantee*, that better captures the properties of GPS [59, 1]. Before giving the formula, we motivate it on three examples.

Consider first a node offering a strict service curve $\beta$. Consider some fixed, but arbitrary times $s < t$. Assume that $\beta$ is continuous. If $[s, t]$ is within a busy period, we must have

$$R^*(t) \geq R^*(s) + \beta(t - s)$$

Else, call $u$ the beginning of the busy period at $t$. We have

$$R^*(t) \geq R(u) + \beta(t - u)$$

thus in all cases

$$R^*(t) \geq (R^*(s) + \beta(t - s)) \wedge \inf_{u \in [s,t]} (R(u) + \beta(t - u)) \tag{7.1}$$

Second, consider a node that guarantees a virtual delay $\leq d$. If $t - s \leq d$ then trivially

$$R^*(t) \geq R^*(s) + \delta_d(t - s)$$

and if $t - s > d$ then the virtual delay property means that

$$R^*(t) \geq R(t - d) = \inf_{u \in [s,t]} (R(u) + \delta_d(t - u))$$

thus we have the same relation as in Equation (7.1) with $\beta = \delta_d$.

Thirdly, consider a greedy shaper with shaping function $\sigma$ (assumed to be a good function). Then

$$R^*(t) = \inf_{u \leq t} [R(u) + \sigma(t - u)]$$

Breaking the inf into $u < s$ and $u \geq s$ gives

$$R^*(t) = \inf_{u < s} [R(u) + \sigma(t - u)] \wedge \inf_{u \in [s,t]} [R(u) + \sigma(t - u)] \tag{7.2}$$

Define $\tilde{\sigma} := \sigma \overline{\oslash} \sigma$, namely,

$$\tilde{\sigma}(u) = \inf_t [\sigma(t + u) - \sigma(u)] \tag{7.3}$$

For example, for a piecewise linear concave arrival curve (conjunction of leaky buckets), $\sigma(t) = \min_i (r_i u + b_i)$, we have $\tilde{\sigma}(u) = \min_i r_i u$. Back to Equation (7.2), we have

$$\sigma(t - u) \geq \sigma(s - u) + \tilde{\sigma}(t - s)$$

and finally

$$R^*(t) \geq (R^*(s) + \tilde{\sigma}(t - s)) \wedge \inf_{u \in [s,t]} (R(u) + \sigma(t - u)) \tag{7.4}$$

We see that these three cases fall under a common model:

**Definition 7.2.1 (Adaptive Service Curve).** *Let $\tilde{\beta}, \beta$ be in $\mathcal{F}$. Consider a system $\mathcal{S}$ and a flow through $\mathcal{S}$ with input and output functions $R$ and $R^*$. We say that $\mathcal{S}$ offers the* adaptive guarantee $(\tilde{\beta}, \beta)$ *if for any $s \le t$ it holds:*

$$R^*(t) \ge \left(R^*(s) + \tilde{\beta}(t-s)\right) \wedge \inf_{u \in [s,t]} [R(u) + \beta(t-u)]$$

*If $\tilde{\beta} = \beta$ we say that the node offers the adaptive guarantee $\beta$.*

The following proposition summarizes the examples discussed above:

**Proposition 7.2.1.**

- *If $\mathcal{S}$ offers to a flow a strict service curve $\beta$, then it also offers the* adaptive guarantee $\beta$*.*
- *If $\mathcal{S}$ guarantees a virtual delay bounded by $d$, then it also offers the* adaptive guarantee $\delta_d$
- *A greedy shaper with shaping curve $\sigma$, where $\sigma$ is a good function, offers the* adaptive guarantee $(\tilde{\sigma}, \sigma)$*, with $\tilde{\sigma}$ defined in Equation (7.3).*

Similar to [59], we use the notation $R \to (\tilde{\beta}, \beta) \to R^*$ to express that Definition 7.2.1 holds. If $\tilde{\beta} = \beta$ we write $R \to (\beta) \to R^*$.

Assume that $R$ is left-continuous and $\beta$ is continuous. It follows from Theorem 3.1.8 on Page 139 that the adaptive guarantee is equivalent to saying that for all $s \le t$, we have either

$$R^*(t) - R^*(s) \ge \tilde{\beta}(t-s)$$

or

$$R^*(t) \ge R(u) + \beta(t-u)$$

for some $u \in [s,t]$.

### 7.2.3 Properties of Adaptive Guarantees

**Theorem 7.2.1.** *Let $R \to (\tilde{\beta}, \beta) \to R^*$. If $\tilde{\beta} \le \beta$ then $\beta$ is a minimum service curve for the flow.*

**Proof:** Apply Definition 7.2.1 with $s = 0$ and use the fact that $\tilde{\beta} \le \beta$. □

**Theorem 7.2.2 (Concatenation).** *If $R \to (\tilde{\beta}_1, \beta_1) \to R_1$ and $R_1 \to (\tilde{\beta}_2, \beta_2) \to R^*$ then $R \to (\tilde{\beta}, \beta) \to R^*$ with*

$$\tilde{\beta} = \left(\tilde{\beta}_1 \otimes \beta_2\right) \wedge \tilde{\beta}_2$$

*and*

$$\beta = \beta_1 \otimes \beta_2$$

**Proof:** Consider some fixed but arbitrary times $s \leq t$ and let $u \in [s,t]$. We have

$$R_1(u) \geq \left[R_1(s) + \tilde{\beta}(u-s)\right] \wedge \inf_{v \in [s,u]} [R(v) + \beta_1(u-v)]$$

thus

$$\begin{aligned} R_1(u) + \beta_2(t-u) \quad & \geq \left[R_1(s) + \tilde{\beta}(u-s) + \beta_2(t-u)\right] \wedge \\ & \inf\nolimits_{v \in [s,u]} [R(v) + \beta_1(u-v) + \beta_2(t-u)] \end{aligned}$$

and

$$\begin{aligned} & \inf_{u \in [s,t]} [R_1(u) + \beta_2(t-u)] \geq \\ & \quad \inf_{u \in [s,t]} \left[R_1(s) + \tilde{\beta}(u-s) + \beta_2(t-u)\right] \\ & \quad \wedge \inf_{u \in [s,t], v \in [s,u]} [R(v) + \beta_1(u-v) + \beta_2(t-u)] \end{aligned}$$

After re-arranging the infima, we find

$$\begin{aligned} & \inf_{u \in [s,t]} [R_1(u) + \beta_2(t-u)] \geq \\ & \quad \left(R_1(s) + \inf_{u \in [s,t]} \left[\tilde{\beta}(u-s) + \beta_2(t-u)\right]\right) \wedge \\ & \quad \inf_{v \in [s,t]} \left(R(v) + \inf_{u \in [v,t]} [\beta_1(u-v) + \beta_2(t-u)]\right) \end{aligned}$$

which can be rewritten as

$$\begin{aligned} & \inf_{u \in [s,t]} [R_1(u) + \beta_2(t-u)] \geq \\ & \quad \left(R_1(s) + (\tilde{\beta_1} \otimes \beta_2)(t-s)\right) \wedge \\ & \quad \inf_{v \in [s,t]} [R(v) + \beta(t-v)] \end{aligned}$$

Now by hypothesis we have

$$R^*(t) \geq \left(R^*(s) + \tilde{\beta_2}(t-s)\right) \wedge \inf_{u \in [s,t]} [R(u) + \beta_2(t-u)]$$

Combining the two gives

$$\begin{aligned} & R^*(t) \geq \\ & \quad \left(R^*(s) + \tilde{\beta_2}(t-s)\right) \wedge \left(R_1(s) + (\tilde{\beta_1} \otimes \beta_2)(t-s)\right) \\ & \quad \wedge \inf_{v \in [s,t]} [R(v) + \beta(t-v)] \end{aligned}$$

Now $R_1(s) \geq R^*(s)$ thus

$$
\begin{aligned}
&R^*(t) \geq \\
&\quad \Big(R^*(s) + \tilde{\beta}_2(t-s)\Big) \wedge \Big(R^*(s) + (\tilde{\beta}_1 \otimes \beta_2)(t-s)\Big) \\
&\quad \wedge \inf_{v \in [s,t]} [R(v) + \beta(t-v)]
\end{aligned}
$$

□

**Corollary 7.2.1.** *If $R_{i-1} \to (\tilde{\beta}_i, \beta_i) \to R_i$ for $i = 1$ to $n$ then $R_0 \to (\tilde{\beta}, \beta) \to R_n$ with*

$$\tilde{\beta} = \Big(\tilde{\beta}_1 \otimes \beta_2 \otimes ... \otimes \beta_n\Big) \wedge \Big(\tilde{\beta}_2 \otimes \beta_3 \otimes ... \otimes \beta_n\Big) \wedge ... \wedge \Big(\tilde{\beta}_{n-1} \otimes \beta_n\Big) \wedge \tilde{\beta}_n$$

*and*

$$\beta = \beta_1 \otimes ... \otimes \beta_n$$

**Proof:** Apply Theorem 7.2.2 iteratively and use Rule 6 in Theorem 3.1.5 on Page 135. □

**Theorem 7.2.3 (Delay from Backlog).** *If $R \to (\tilde{\beta}, \beta) \to R^*$, then the virtual delay at time $t$ is bounded by $\tilde{\beta}^{-1}(Q(t))$, where $Q(t)$ is the backlog at time $t$, and $\tilde{\beta}^{-1}$ is the pseudo-inverse of $\tilde{\beta}$ (see Definition 3.1.7 on Page 129).*

Note that if the node is FIFO, then the virtual delay at time $t$ is the real delay for a bit arriving at time $t$.

**Proof:** If the virtual delay at time $t$ is larger than $t + \tau$ for some $\tau \geq 0$, then we must have

$$R^*(t+\tau) < R(t) \tag{7.5}$$

By hypothesis

$$R^*(t+\tau) \geq \Big(R^*(t) + \tilde{\beta}(\tau)\Big) \wedge \inf_{[u \in [t,t+\tau]} [R(u) + \beta(t+\tau-u)] \tag{7.6}$$

now for $u \in [t, t+\tau]$

$$R(u) + \beta(t+\tau-u) \geq R(t) + \beta(0) \geq R^*(t+\tau)$$

thus Equation (7.6) implies that

$$R^*(t+\tau) \geq R^*(t) + \tilde{\beta}(\tau)$$

combining with Equation (7.5) gives

$$Q(t) = R(t) - R^*(t) \geq \tilde{\beta}(\tau)$$

thus the virtual delay is bounded by $\sup\{\tau : \tilde{\beta}(\tau) > Q(t)\}$ which is equal to $\tilde{\beta}^{-1}(Q(t))$. □

Consider a system (*bit-by-bit system*) with $L$-packetized input $R$ and bit-by-bit output $R^*$, which is then $L$-packetized to produce a final packetized output $R'$. We call *combined system* the system that maps $R$ into $R'$. Assume both systems are first-in-first-out and lossless. Remember from Theorem 1.7.1 that the per-packet delay for the combined system is equal the maximum virtual delay for the bit-by-bit system.

**Theorem 7.2.4 (Packetizer and Adaptive Guarantee).** *If the bit-by-bit system offers to the flow the adaptive guarantee* $(\tilde{\beta}, \beta)$*, then the combined system offers to the flow the adaptive guarantee* $(\tilde{\beta}', \beta')$ *with*

$$\tilde{\beta}'(t) = [\tilde{\beta}(t) - l_{\max}]^+$$

*and*

$$\beta'(t) = [\beta(t) - l_{\max}]^+$$

*where* $l_{\max}$ *is the maximum packet size for the flow.*

**Proof:** Let $s \le t$. By hypothesis we have

$$R^*(t) \ge \left(R^*(s) + \tilde{\beta}(t-s)\right) \wedge \inf_{u \in [s,t]} [R(u) + \beta(t-u)]$$

We do the proof when the $\inf$ in the above formula is a minimum, and leave it to the alert reader to extend it to the general case. Thus assume that for some $u_0 \in [s,t]$:

$$\inf_{u \in [s,t]} [R(u) + \beta(t-u)] = R(u_0) + \beta(t-u_0)$$

it follows that either

$$R^*(t) - R^*(s) \ge \tilde{\beta}(t-s)$$

or

$$R^*(t) \ge R(u_0) + \beta(t-u_0)$$

Consider the former case. We have $R'(t) \ge R^*(t) - l_{\max}$ and $R'(s) \le R^*(s)$ thus

$$R'(t) \ge R^*(t) - l_{\max} \ge R'(s) + \tilde{\beta}(t-s) - l_{\max}$$

Now also obviously $R'(t) \ge R'(s)$, thus finally

$$R'(t) \ge R'(s) + \max[0, \tilde{\beta}(t-s) - l_{\max}] = R'(s) + \tilde{\beta}'(t-s)$$

Consider now the latter case. A similar reasoning shows that

$$R'(t) \ge R(u_0) + \beta(t-u_0) - l_{\max}$$

but also

$$R^*(t) \geq R(u_0)$$

now the input is $L$-packetized. Thus

$$R'(t) = P^L(R^*(t)) \geq P^L(R(u_0)) = R(u_0)$$

from which we conclude that $R'(t) \geq R(u_0) + \beta'(t - u_0)$.

Combining the two cases provides the required adaptive guarantee. □

## 7.3 Application to the Internet: Packet Scale Rate Guarantee

In this section we apply the concept of adaptive guarantee to practical schedulers used in the Internet.

### 7.3.1 Definition of Packet Scale Rate Guarantee

In Section 2.1.3 on Page 86 we have introduced the definition of guaranteed rate scheduler, which is the practical application of rate latency service curves. Consider a node where packets arrive at times $A_1 \geq 0, A_2, ...$ and leave at times $D_1, D_2, ....$ A guaranteed rate scheduler, with rate $r$ and latency $v$ requires that $D_i \leq T'_i + v$, where $T'_i$ is defined iteratively by $T'_0 = 0$ and

$$T'_i = \max\{A_i, T'_{i-1}\} + \frac{l_i}{r}$$

where $l_i$ is the length of the $i$th packet.

A *packet scale rate guarantee* is similar, but, much in the spirit of adaptive guarantees, avoids the limitations of the service curve concept discussed in Section 7.2.1. To that end, we would like that the deadline $T'_i$ is reduced whenever a packet happens to be served early. This is done by replacing $T'_{i-1}$ in the previous equation by $\min\{T'_i, D_i\}$. This gives the following definition.

**Definition 7.3.1 (Packet Scale Rate Guarantee).** *Consider a node that serves a flow of packets numbered $i = 1, 2, ....$ Call $A_i, D_i, l_i$ the arrival time, departure time, and length in bits for the $i$th packet, in order of arrival. Assume $A_1 \geq 0$.We say that the node offers to the flow a packet scale rate guarantee with rate $r$ and latency $v$ if the departure times satisfy*

$$D_i \leq F_i + v$$

*where $F_i$ is defined by:*

$$\begin{cases} F_0 = D_0 = 0 \\ F_i = \max\{A_i, \min(D_{i-1}, F_{i-1})\} + \frac{l_i}{r} \quad \text{for all } i \geq 1 \end{cases} \tag{7.7}$$

We now relate packet scale rate guarantee to an adaptive guarantee. We cannot expect an exact equivalence, since a packet scale rate guarantee does not specify what happens to bits at a time other than a packet departure or arrival. However, the concept of packetizer allows us to establish an equivalence.

**Theorem 7.3.1 (Equivalence with adaptive guarantee).** *Consider a node $\mathcal{S}$ with $L$-packetized input $R$ and with output $R^*$.*

1. *If $R \to (\beta) \to R^*$, where $\beta = \beta_{r,v}$ is the rate-latency function with rate $r$ and latency $v$, and if $\mathcal{S}$ is FIFO, then $\mathcal{S}$ offers to the flow the packet scale rate guarantee with rate $r$ and latency $v$.*
2. *Conversely, if $\mathcal{S}$ offers to the flow the packet scale rate guarantee with rate $r$ and latency $v$ and if $R^*$ is $L$-packetized, then $\mathcal{S}$ is the concatenation of a node $\mathcal{S}'$ offering the adaptive guarantee $\beta_{r,v}$ and the $L$-packetizer. If $\mathcal{S}$ is FIFO, then so is $\mathcal{S}'$.*

The proof is long and is given in a separate section (Section 7.3.3). Note that the packet scale rate guarantee does not mandate that the node be FIFO; it is possible that $D_i < D_{i-1}$ in some cases. However, part 1 of the theorem requires the FIFO assumption in order for a condition on $R, R^*$ to be translated into a condition on delays.

A special case of interest is when $v = 0$.

**Corollary 7.3.1.** *Consider a node with $L$-packetized input. Call $A_i, D_i$ the arrival and departure times for packet $i$, with $i = 1, 2, ...$ and $A_1 \geq 0$. Let $l_i$ be the size of packet $i$.*

1. *If the node guarantees a strict service curve $\lambda_r$ and is FIFO then*

$$\left\{ \begin{array}{l} D_0 = 0 \\ D_i \leq \max\{A_i, D_{i-1}\} + \frac{l_i}{r} \quad \text{for all } i \geq 1 \end{array} \right. \tag{7.8}$$

2. *Conversely if Equation (7.8) holds for all $i$, and if the output is $L$-packetized, then the node is the concatenation of a node guaranteeing a strict service curve $\lambda_r$ and an $L$-packetizer.*

**Proof:** Apply Theorem 7.3.1 with $v = 0$ and note that $D_{i-1} \leq F_{i-1}$ in Equation (7.7). □

**Definition 7.3.2.** *We call* minimum rate server*, with rate $r$, a node for which Equation (7.8) holds for all $i$*

Thus, roughly speaking, a minimum rate server guarantees that during any busy period, the instantaneous output rate is at least $r$. A GPS node with total rate $C$ and weight $w_i$ for flow $i$ is a minimum rate server for flow $i$, with rate $r_i = \frac{\phi_i C}{\sum_j \phi_j}$.

Since a packetizer does not add to the per-packet delay, we can immediately derive the following property from Theorem 7.2.3 and Theorem 7.3.1:

**Proposition 7.3.1 (Backlog from Delay).** *For a FIFO node offering the packet scale rate guarantee with rate $r$ and latency $v$, the delay for a packet present in the system at time $t$ is bounded by $\frac{Q(t)}{r} + v$, where $Q(t)$ is the backlog at time $t$.*

Lastly, we have a concatenation result for FIFO systems:

**Proposition 7.3.2.** *Consider a concatenation of FIFO systems numbered $1$ to $n$. The output of system $i-1$ is the input of system $i$, for $i > 1$. Assume system $i$ offers the packet scale rate guarantee with rate $R_i$ and latency $E_i$. The global system offers the packet scale rate guarantee with rate $R = \min_{i=1,\dots,n} R_i$ and latency $E = \sum_{i=1,\dots,n} E_i + \sum_{i=1,\dots,n-1} \frac{L_{\max}}{R_i}$.*

**Proof:** By Theorem 7.3.1–(2), we can decompose system $i$ into a concatenation $\mathcal{S}_i, \mathcal{P}_i$, where $\mathcal{S}_i$ offers the adaptive guarantee $\beta_{R_i,E_i}$ and $\mathcal{P}_i$ is a packetizer.

Call $\mathcal{S}$ the concatenation

$$\mathcal{S}_1, \mathcal{P}_1, \mathcal{S}_2, \mathcal{P}_2, ..., \mathcal{S}_{n-1}, \mathcal{P}_{n-1}, \mathcal{S}_n$$

By Theorem 7.3.1–(2), $\mathcal{S}$ is FIFO. By Theorem 7.2.4, it provides the adaptive guarantee $\beta_{R,E}$. By Theorem 7.3.1–(1), it also provides the packet scale rate guarantee with rate $R$ and latency $E$. Now $\mathcal{P}_n$ does not affect the finish time of the last bit of every packet.

□

### 7.3.2 Practical Realization of Packet Scale Rate Guarantee

We show in this section that a wide variety of schedulers provide the packet scale rate guarantee. More schedulers can be obtained by using the concatenation theorem in the previous section.

A simple but important realization is the priority scheduler.

**Proposition 7.3.3.** *Consider a non-preemptive priority scheduler in which all packets share a single FIFO queue with total output rate $C$. The high priority flow receives a packet scale rate guarantee with rate $C$ and latency $v = \frac{l_{max}}{C}$, where $l_{max}$ is the maximum packet size of all low priority packets.*

**Proof:** By Proposition 1.3.7, the high priority traffic receives a strict service curve $\beta_{r,c}$. □

We have already introduced in Section 2.1.3 a large number of schedulers that can be thought of as derived from GPS and we have modeled their behaviour with a rate-latency service curve. In order to give an adaptive guarantee for such schedulers, we need to define more.

**Definition 7.3.3 (Accuracy of a scheduler with respect to rate $r$).** *Consider a scheduler $S$ and call $D_i$ the time of the $i$-th departure. We say that the accuracy of $S$ with respect to rate $r$ is $(v_1, v_2)$ if there is a minimum rate server with rate $r$ and departure times $G_i$ such that for all $i$*

$$G_i - v_1 \le D_i \le G_i + v_2 \tag{7.9}$$

We interpret this definition as a comparison to a hypothetical GPS reference scheduler that would serve the same flows. The term $v_2$ determines the maximum per-hop delay bound, whereas $v_1$ has an effect on the jitter at the output of the scheduler. For example, it is shown in [6] that WF$^2$Q satisfies $v_1(\text{WF}^2\text{Q}) = l_{max}/r$, $v_2(\text{WF}^2\text{Q}) = l_{max}/C$, where $l_{max}$ is maximum packet size and $C$ is the total output rate. In contrast, for PGPS [61] $v_2(\text{PGPS}) = v_2(\text{WF}^2\text{Q})$, while $v_1(\text{PGPS})$ is linear in the number of queues in the scheduler. This illustrates that, while WF$^2$Q and PGPS have the same delay bounds, PGPS may result in substantially burstier departure patterns.

**Theorem 7.3.2.** *If a scheduler satisfies Equation (7.9), then it offers the packet scale rate guarantee with rate $r$ and latency $v = v_1 + v_2$.*

**Proof:** We first prove that for all $i \ge 0$

$$F_i \ge G_i - v_1 \tag{7.10}$$

where $F_i$ is defined by Equation (7.7). Indeed, if Equation (7.10) holds, then by Equation (7.9)):

$$D_i \le G_i + v_2 \le F_i + v_1 + v_2$$

which means that the scheduler offers the packet scale rate guarantee with rate $r$ and latency $v = v_1 + v_2$.

Now we prove Equation (7.10) by induction. Equation (7.10) trivially holds for $i = 0$.

Suppose now that it holds for $i - 1$, namely,

$$F_{i-1} \ge G_{i-1} - v_1$$

By hypothesis, Equation (7.9) holds:

$$D_{i-1} \ge G_{i-1} - v_1$$

thus

$$\min[F_{i-1}, D_{i-1}] \ge G_{i-1} - v_1 \tag{7.11}$$

Combining this with Equation (7.7), we obtain

$$F_i \ge G_{i-1} - v_1 + \frac{L(i)}{R} \tag{7.12}$$

Again from Equation (7.7) we have

$$\begin{array}{rl} F_i \ge & A_i + \frac{l_i}{r} \\ \ge & A_i - v_1 + \frac{l_i)}{r} \end{array} \tag{7.13}$$

Now by Equation (7.8)

$$G_i \leq \max[A_i, G_{i-1}] + \frac{l_i}{r} \tag{7.14}$$

Combining Equation (7.12)), Equation (7.13)) and (7.14) gives

$$F_i \geq G_i - v_1$$

□

### 7.3.3 Proof of Theorem 7.3.1

The first part of Theorem 7.3.1 is based on a max-plus representation of the packet scale rate guarantee, which maps the (min-plus) definition of an adaptive guarantee. The second part relies on the reduction to the minimum rate server.

We use the same notation as in Definition 7.3.1. $L(i) = \sum_{j=1}^{i} l_j$ is the cumulative packet length.

**Part 1:** Define the sequence of times $F_k$ by Equation (7.7). Consider now some fixed but arbitrary packet index $i \geq 1$. By the FIFO assumption, it is sufficient to show that

$$R^*(t) \geq L(i) \tag{7.15}$$

with $t = F_i + v$. Define

$$j = \max\{k \in \{1, ..., i\} : A_k \geq D_{k-1} \text{ or } A_k < D_{k-1} \leq F_{k-1}\}$$

Note that the set above is non-empty and $1 \leq j \leq i$. The definition of $j$ implies

$$A_j \geq D_{j-1} \text{ or } A_j < D_{j-1} \leq F_{j-1} \tag{7.16}$$

and

$$A_k < D_{k-1} \text{ and } F_{k-1} < D_{k-1} \text{ for } j+1 \leq k \leq i \tag{7.17}$$

Note that the set of indices $k$ to which the previous equation applies may be empty (in that case, $j = i$).

By Equation (7.16) and the definition of $F_j$, we have

$$F_j = s + \frac{l_j}{r} \tag{7.18}$$

with

$$s = A_j \vee D_{j-1}$$

Similarly, we derive from Equation (7.17) that for $j+1 \leq k \leq i$:

$$F_k = (A_k \vee F_{k-1}) + \frac{l_k}{r}$$

which can be re-written as

$$F_k = \left(A_k + \frac{l_k}{r}\right) \vee \left(F_{k-1} + \frac{l_k}{r}\right) \tag{7.19}$$

Now we obtain a max-plus expansion of $F_i$ as follows. We substitute $F_{i-1}$ from Equation (7.19) at $k = i - 1$ into Equation (7.19) at $k = i$ and obtain

$$F_i = \left(A_i + \frac{l_i}{r}\right) \vee \left(A_{i-1} + \frac{l_i + l_{i-1}}{r}\right) \vee \left(F_{i-2} + \frac{l_i + l_{i-1}}{r}\right)$$

We apply this iteratively until $k = j$ at which step we use Equation (7.18) instead of Equation (7.19). We obtain finally:

$$F_i = \left(s + \frac{L(i) - L(j-1)}{r}\right) \bigvee \max_{k=j+1}^{i} \left(A_k + \frac{L(i) - L(k-1)}{r}\right) \tag{7.20}$$

Let us apply the definition of an adaptive guarantee to the time interval $[s, t]$:

$$R^*(t) \geq A \wedge B$$

with

$$A := R^*(s) + r(t - s - v)^+ \text{ and } B := \inf_{u \in [s,t]} B(u)$$

where

$$B(u) := \big(R(u) + r(t - u - v)^+\big)$$

Firstly, since $s \geq D_{j-1}$, we have $R^*(s) \geq L(j-1)$. By Equation (7.20), $F_i \geq s + \frac{L(i)-L(j-1)}{r}$ thus $t \geq s + \frac{L(i)-L(j-1)}{r} + v$. It follows that

$$t - s - v \geq \frac{L(i) - L(j-1)}{r}$$

and thus $A \geq L(i)$.

Secondly, we show that $B \geq L(i)$ as well. Consider some $u \in [s, t]$. If $u \geq A_i$ then $R(u) \geq L(i)$ thus $B(u) \geq L(i)$. Otherwise, $u < A_i$; since $s \geq A_j$, it follows that $A_{k-1} \leq u < A_k$ for some $k \in \{j + 1, ..., i\}$ and $R(u) = L(k - 1)$. By Equation (7.20),

$$F_i \geq A_k + \frac{L(i) - L(k-1)}{r}$$

thus

$$t - u - v \geq \frac{L(i) - L(k-1)}{r}$$

It follows that $B(u) \geq L(i)$ also in that case. Thus we have shown that $B \geq L(i)$.

Combining the two shows that $R^*(t) \geq L(i)$ as required.

**Part 2:** We use a reduction to a minimum rate server as follows. Let $D'_i := \min(D_i, F_i)$ for $i \geq 0$. By Equation (7.7) we have

$$A_i \leq D'_i \leq \max(A_i, D'_{i-1}) + \frac{l_i}{r} \tag{7.21}$$

and

$$D'_i \leq D_i \leq D'_i + v \tag{7.22}$$

The idea of the proof is now to interpret $D'_i$ as the output time for packet $i$ out of a virtual minimum rate server. Of course, we cannot use Corollary 7.3.1.

Construct a virtual node $\mathcal{R}$ as follows. The input is the original input $R(t)$. The output is defined as follows. The number of bits of packet $i$ that are output up to time $t$ is $\psi_i(t)$, defined by

$$\begin{cases} \text{if } t > d'(i) \text{ then } \psi_i(t) = L(i) \\ \text{else if } a(i) < t \leq d'(i) \text{ then } \psi_i(t) = [L(i) - r(d'(i) - t)]^+ \\ \text{else } \psi_i(t) = 0 \end{cases}$$

so that the total output of $\mathcal{R}$ is $R_1(t) = \sum_{i \geq 1} \psi_i(t)$.

The start time for packet $i$ is thus $\max[A_i, D'_i - \frac{l_i}{r}]$ and the finish time is $D'_i$. Thus $\mathcal{R}$ is causal (but not necessarily FIFO, even if the original system would be FIFO). We now show that during any busy period, $\mathcal{R}$ has an output rate at least equal to $r$.

Let $t$ be during a busy period. Consider now some time $t$ during a busy period. There must exist some $i$ such that $A_i \leq t \leq D'_i$. Let $i$ be the smallest index such that this is true. If $A_i \geq D'_{i-1}$ then by Equation (7.21) $D'_i - t \leq \frac{l_i}{r}$ and thus $\psi'_r(t) = r$ where $\psi'_r$ is the derivative of $\psi_i$ to the right. Thus the service rate at time $t$ is at least $r$.

Otherwise, $A_i < D'i - 1$. Necessarily (because we number packets in order of increasing $A_i$'s – this is not a FIFO assumption) $A_{i-1} \leq A_i$; since $i$ is the smallest index such that $A_i \leq t < D'_i$, we must have $t \geq D'_{i-1}$. But then $D'_i - t \leq \frac{l_i}{r}$ and the service rate at time $t$ is at least $r$. Thus, node $\mathcal{R}$ offers the strict service curve $\lambda_r$ and

$$R \to (\lambda_r) \to R_1 \tag{7.23}$$

Now define node $\mathcal{D}$. Let $\delta(i) := D_i - D'_i$, so that $0 \leq \delta(i) \leq E$. The input of $\mathcal{D}$ is the output of $\mathcal{R}$. The output is as follows; let a bit of packet $i$ arrive at time $t$; we have $t \leq D'_i \leq D_i$. The bit is output at time $t' = \max[\min[D_{i-1}, D_i], t + \delta_i]$. Thus all bits of packet $i$ are delayed in $\mathcal{D}$ by at most $\delta(i)$, and if $D_{i-1} < D_i$ they depart after $D_i$. It follows that the last bit of packet $i$ leaves $\mathcal{D}$ at time $D_i$. Also, since $t' \geq t$, $\mathcal{D}$ is causal. Lastly, if the original system is FIFO, then $D_{i-1} < D_i$, all bits of packet $i$ depart after $D_{i-1}$ and thus the concatenation of $\mathcal{R}$ and $\mathcal{D}$ is FIFO. Note that $\mathcal{R}$ is not necessarily FIFO, even if the original system is FIFO.

The aggregate output of $\mathcal{D}$ is

$$R_2(t) \geq \sum_{i \geq 1} \psi_i(t - \delta(i)) \geq R_1(t - v)$$

thus the virtual delay for $\mathcal{D}$ is bounded by $v$ and

$$R_1 \to (\delta_v) \to R_2 \tag{7.24}$$

Now we plug the output of $\mathcal{D}$ into an $L$-packetizer. Since the last bit of packet $i$ leaves $\mathcal{D}$ at time $D_i$, the final output is $R^*$. Now it follows from Equation (7.23), Equation (7.24) and Theorem 7.2.2 that

$$R \to (\lambda_r \otimes \delta_v) \to R_2$$

□

## 7.4 Bibliographic Notes

The concept of adaptive service curve was introduced in Okino's dissertation in [59] and was published by Agrawal, Cruz, Okino and Rajan in [1], which contains most results in Section 7.2.3, as well as an application to a window flow control problem that extends Section 4.3.2 on Page 178. They call $\tilde{\beta}$ an "adaptive service curve" and $\beta$ a "partial service curve".

The packet scale rate guarantee was first defined in a framework dependent of adaptive service guarantees in [4]. It serves as a basis for the definition of the Expedited Forwarding capability defined for the Internet.

## 7.5 Exercises

**Exercise 7.1.** *Assume that* $R \to (\tilde{\beta}, \beta) \to R^*$.

1. *Show that the node offers to the flow a strict service curve equal to* $\tilde{\beta} \otimes \overline{\beta}$*, where* $\overline{\beta}$ *is the sub-additive closure of* $\beta$.

2. *If* $\tilde{\beta} = \beta$ *is a rate-latency function, what is the value obtained for the strict service curve ?*

**Exercise 7.2.** *Consider a system with input* $R$ *and output* $R^*$*. We call "input flow restarted at time* $t$*" the flow* $R_t$ *defined for* $u \geq 0$ *by*

$$R_t(u) = R(t + u) - R^*(t) = R(t, u] + Q(t)$$

*where* $Q(t) := R(t) - R^*(t)$ *is the backlog at time* $t$*. Similarly, let the"output flow restarted at time* $t$*" be the flow* $R^*_t$ *defined for* $u \geq 0$ *by*

$$R^*_t(u) = R^*(t + u) - R^*(t)$$

*Assume that the node guarantees a service curve* $\beta$ *to all couples of input, output flows* $(R_t, R^*_t)$*. Show that* $R \to (\beta) \to R^*$.

# Chapter 8

# Time Varying Shapers

## 8.1 Introduction

Throughout the book we usually assume that systems are idle at time $0$. This is not a limitation for systems that have a renewal property, namely, which visit the idle state infinitely often – for such systems we choose the time origin as one such instant.

There are cases however where we are interested in the effect at time $t$ of non zero initial conditions. This occurs for example for re-negotiable services, where the traffic contract is changed at periodic renegotiation moments. An example for this service is the Integrated Service of the IETF with the Resource reSerVation Protocol (RSVP), where the negotiated contract may be modified periodically [30]. A similar service is the ATM Available Bit Rate service (ABR). With a renegotiable service, the shaper employed by the source is time-varying. With ATM, this corresponds to the concept of Dynamic Generic Cell Rate Algorithm (DGCRA).. At renegotiation moments, the system cannot generally be assumed to be idle. This motivates the need for explicit formulae that describe the transient effect of non-zero initial condition.

In Section 8.2 we define time varying shapers. In general, there is not much we can say apart from a direct application of the fundamental min-plus theorems in Section 4.3. In contrast, for shapers made of a conjunction of leaky buckets, we can find some explicit formulas. In Section 8.3.1 we derive the equations describing a shaper with non-zero initial buffer. In Section 8.3.2 we add the constraint that the shaper has some history. Lastly, in Section 8.4, we apply this to analyze the case where the parameters of a shaper are periodically modified.

This chapter also provides an example of the use of time shifting.

## 8.2 Time Varying Shapers

We define a time varying shaper as follows.

**Definition 8.2.1.** *Consider a flow $R(t)$. Given a function of two time variables $H(t, s)$, a* time varying shaper *forces the output $R^*(t)$ to satisfy the condition*

$$R^*(t) \leq H(t, s) + R^*(s)$$

*for all $s \leq t$, possibly at the expense of buffering some data. An optimal time varying shaper, or greedy time varying shaper, is one that maximizes its output among all possible shapers.*

The existence of a greedy time varying shaper follows from the following proposition.

**Proposition 8.2.1.** *For an input flow $R(t)$ and a function of two time variables $H(t, s)$, among all flows $R^* \leq R$ satisfying*

$$R^*(t) \leq H(t, s) + R^*(s)$$

*there is one flow that upper bounds all. It is given by*

$$R^*(t) = \inf_{s \geq 0} \left[ \overline{H}(t, s) + R(s) \right] \tag{8.1}$$

*where $\overline{H}$ is the min-plus closure of $H$, defined in Equation (4.10) on Page 172.*

**Proof:** The condition defining a shaper can be expressed as

$$\begin{cases} R^* \leq \mathcal{L}_H(R^*) \\ R^* \leq R \end{cases}$$

where $\mathcal{L}_H$ is the min-plus linear operator whose impulse response is $H$ (Theorem 4.1.1). The existence of a maximum solution follows from Theorem 4.3.1 and from the fact that, being min-plus linear, $\mathcal{L}_H$ is upper-semi-continuous. The rest of the proposition follows from Theorem 4.2.1 and Theorem 4.3.1. □

The output of the greedy shaper is given by Equation (8.1). A time invariant shaper is a special case; it corresponds to $H(s, t) = \sigma(t - s)$, where $\sigma$ is the shaping curve. In that case we find the well-known result in Theorem 1.5.1.

In general, Proposition 8.2.1 does not help much. In the rest of this chapter, we specialize to the class of concave piecewise linear time varying shapers.

**Proposition 8.2.2.** *Consider a set of $J$ leaky buckets with time varying rates $r_j(t)$ and bucket sizes $b_j(t)$. At time $0$, all buckets are empty. A flow $R(t)$ satisfies the conjunction of the $J$ leaky bucket constraints if and only if for all $0 \leq s \leq t$:*

$$R(t) \leq H(t, s) + R(s)$$

*with*

$$H(t, s) = \min_{1 \leq j \leq J} \{ b_j(t) + \int_s^t r_j(u) du \} \tag{8.2}$$

**Proof:** Consider the level of the $j$th bucket. It is the backlog of the variable capacity node (Section 1.3.2) with cumulative function

$$M_j(t) = \int_0^t r_j(u)du$$

We know from Chapter 4 that the output of the variable capacity node is given by

$$R'_j(t) = \inf_{0 \le s \le t} \{M_j(t) - M_j(s) + R(s)\}$$

The $j$th leaky bucket constraint is

$$R(t) - R'_j(t) \le b_j(t)$$

Combining the two expresses the $j$th constraint as

$$R(t) - R(s) \le M_j(t) - M_j(s) + b_j(t)$$

for all $0 \le s \le t$. The conjunction of all these constraints gives Equation (8.2).

In the rest of this chapter, we give a practical and explicit computation of $\overline{H}$ for $H$ given in Equation (8.2), when the functions $r_j(t)$ and $b_j(t)$ are piecewise constant.

## 8.3 Time Invariant Shaper with Non-zero Initial Conditions

We consider in this section some time invariant shapers. We start with a general shaper with shaping curve $\sigma$, whose buffer is not assumed to be initially empty. Then we will apply this to analyze leaky bucket shapers with non-empty initial buckets.

### 8.3.1 Shaper with Non-empty Initial Buffer

**Proposition 8.3.1 (Shaper with non-zero initial buffer).** *Consider a shaper system with shaping curve $\sigma$. Assume that $\sigma$ is a good function. Assume that the initial buffer content is $w_0$. Then the output $R^*$ for a given input $R$ is*

$$R^*(t) = \sigma(t) \wedge \inf_{0 \le s \le t} \{R(s) + w_0 + \sigma(t-s)\} \quad \text{for all } t \ge 0 \tag{8.3}$$

**Proof:** First we derive the constraints on the output of the shaper. $\sigma$ is the shaping function thus, for all $t \ge s \ge 0$

$$R^*(t) \le R^*(s) + \sigma(t-s)$$

and given that the bucket at time zero is not empty, for any $t \ge 0$, we have that

$$R^*(t) \le R(t) + w_0$$

At time $s = 0$, no data has left the system; this is expressed with

$$R^*(t) \leq \delta_0(t)$$

The output is thus constrained by

$$R^* \leq (\sigma \otimes R^*) \wedge (R + w_0) \wedge \delta_0$$

where $\otimes$ is the min-plus convolution operation, defined by $(f \otimes g)(t) = \inf_s f(s) + g(t-s)$. Since the shaper is an optimal shaper, the output is the maximum function satisfying this inequality. We know from Lemma 1.5.1 that

$$\begin{aligned} R^* &= \sigma \otimes [(R + w_0) \wedge \delta_0] \\ &= [\sigma \otimes (R + w_0)] \wedge [\sigma \otimes \delta_0] \\ &= [\sigma \otimes (R + w_0)] \wedge \sigma \end{aligned}$$

which after some expansion gives the formula in the proposition. □.

Another way to look at the proposition consists in saying that the initial buffer content is represented by an instantaneous burst at time $0$.

The following is an immediate consequence.

**Corollary 8.3.1 (Backlog for a shaper with non-zero initial buffer).** *The backlog of the shaper buffer with the initial buffer content $w_0$ is given by*

$$w(t) = (R(t) - \sigma(t) + w_0) \vee \sup_{0<s\leq t} \{R(t) - R(s) - \sigma(t-s)\} \tag{8.4}$$

### 8.3.2 Leaky Bucket Shapers with Non-zero Initial Bucket Level

Now we characterize a leaky-bucket shaper system with non-zero initial bucket levels.

**Proposition 8.3.2 (Compliance with $J$ leaky buckets with non-zero initial bucket levels).** *A flow $S(t)$ is compliant with $J$ leaky buckets with leaky bucket specifications $(r_j, b_j)$, $j = 1, 2 \dots J$ and initial bucket level $q_j^0$ if and only if*

$$\begin{aligned} &S(t) - S(s) \leq \min_{1\leq j\leq J} [r_j \cdot (t-s) + b_j] \quad \text{for all } 0 < s \leq t \\ &S(t) \leq \min_{1\leq j\leq J} [r_j \cdot t + b_j - q_j^0] \quad \text{for all } t \geq 0 \end{aligned}$$

**Proof:** Apply Section 8.3.1 to each of the buckets. □

**Proposition 8.3.3 (Leaky-Bucket Shaper with non-zero initial bucket levels).** *Consider a greedy shaper system defined by the conjunction of $J$ leaky buckets $(r_j, b_j)$, with $j = 1, 2 \dots J$. Assume that the initial bucket level of the j-th bucket is $q_j^0$. The initial level of the shaping buffer is zero. The output $R^*$ for a given input $R$ is*

$$R^*(t) = \min[\sigma^0(t), (\sigma \otimes R)(t)] \quad \text{for all } t \geq 0 \tag{8.5}$$

*where $\sigma$ is the shaping function*

$$\sigma(u) = \min_{1 \le j \le J}\{\sigma_j(u)\} = \min_{1 \le j \le J}\{r_j \cdot u + b_j\}$$

*and $\sigma^0$ is defined as*

$$\sigma^0(u) = \min_{1 \le j \le J}\{r_j \cdot u + b_j - q_j^0\}$$

**Proof:** By Corollary 8.3.2 applied to $S = R^*$, the condition that the output is compliant with the $J$ leaky buckets is

$$\begin{array}{ll} R^*(t) - R^*(s) \le \sigma(t-s) & \text{for all } 0 < s \le t \\ R^*(t) \le \sigma^0(t) & \text{for all } t \ge 0 \end{array}$$

Since $\sigma^0(u) \le \sigma(u)$ we can extend the validity of the first equation to $s = 0$. Thus we have the following constraint:

$$R^*(t) \le [(\sigma \otimes R^*) \wedge (R \wedge \sigma^0)](t)$$

Given that the system is a greedy shaper, $R^*(\cdot)$ is the maximal solution satisfying those constraints. Using the same min-plus result as in Proposition 8.3.1, we obtain:

$$R^* = \sigma \otimes (R \wedge \sigma^0) = (\sigma \otimes R) \wedge (\sigma \otimes \sigma^0)$$

As $\sigma^0 \le \sigma$, we obtain

$$R^* = (\sigma \otimes R) \wedge \sigma^0$$

□

We can now obtain the characterization of a leaky-bucket shaper with non-zero initial conditions.

**Theorem 8.3.1 (Leaky-Bucket Shaper with non-zero initial conditions).** *Consider a shaper defined by $J$ leaky buckets $(r_j, b_j)$, with $j = 1, 2 \ldots J$ (leaky-bucket shaper). Assume that the initial buffer level of is $w_0$ and the initial level of the $j$th bucket is $q_j^0$. The output $R^*$ for a given input $R$ is*

$$R^*(t) = \min\{\sigma^0(t), w_0 + \inf_{u>0}\{R(u) + \sigma(t-u)\}\} \quad \text{for all } t \ge 0 \tag{8.6}$$

*with*

$$\sigma^0(u) = \min_{1 \le j \le J}(r_j \cdot u + b_j - q_j^0)$$

**Proof:** Apply Proposition 8.3.3 to the input $R' = (R + w_0) \wedge \delta_0$ and observe that $\sigma^0 \le \sigma$. □

An interpretation of Equation (8.6) is that the output of the shaper with non-zero initial conditions is either the output of the ordinary leaky-bucket shaper, taking into account the initial level of the buffer, or, if smaller, the output imposed by the initial conditions, independent of the input.

## 8.4 Time Varying Leaky-Bucket Shaper

We consider now time varying leaky-bucket shapers that are piecewise constant. The shaper is defined by a fixed number $J$ of leaky buckets, whose parameters change at times $t_i$. For $t \in [t_i, t_{i+1}) := I_i$, we have thus

$$r_j(t) = r_j^i \text{ and } b_j(t) = b_j^i$$

At times $t_i$, where the leaky bucket parameters are changed, we keep the leaky bucket level $q_j(t_i)$ unchanged.

We say that $\sigma_i(u) := \min_{1 \le j J}\{r_j^i u + b_j^i\}$ is the value of the time varying shaping curve during interval $I_i$. With the notation in Section 8.2, we have

$$H(t, t_i) = \sigma_i(t - t_i) \text{ if } t \in I_i$$

We can now use the results in the previous section.

**Proposition 8.4.1 (Bucket Level).** *Consider a piecewise constant time varying leaky-bucket shaper with output $R^*$. The bucket level $q_j(t)$ of the j-th bucket is, for $t \in I_i$:*

$$q_j(t) = \begin{array}{l} \left[R^*(t) - R^*(t_i) - r_j^i \cdot (t - t_i) + q_j(t_i)\right] \vee \\ \sup_{t_i < s \le t}\{R^*(t) - R^*(s) - r_j^i \cdot (t - s)\} \end{array} \tag{8.7}$$

**Proof:** We use a time shift, defined as follows. Consider a fixed interval $I_i$ and define

$$x^*(\tau) := R^*(t_i + \tau) - R^*(t_i)$$

Observe that $q_j(t_i + \tau)$ is the backlog at time $\tau$ (call it $w(\tau)$ at the shaper with shaping curve $\sigma(\tau) = r_j^i \cdot t$, fed with flow $x^*$, and with an initial buffer level $q_j(t_i)$. By Chapter 8.3.1 we have

$$w(\tau) = \left[x^*(\tau) - r_j^i \cdot \tau + q_j(t_i)\right] \vee \sup_{0 < s' \le \tau}\{x^*(\tau) - x^*(s') - r_j^i \cdot (\tau - s')\}$$

which after re-introducing $R^*$ gives Equation (8.7) □

**Theorem 8.4.1 (Time Varying Leaky-Bucket Shapers).** *Consider a piecewise constant time varying leaky-bucket shaper with time varying shaping curve $\sigma_i$ in the interval $I_i$. The output $R^*$ for a given input $R$ is*

$$R^*(t) = \min\left[\sigma_i^0(t - t_i) + R^*(t_i), \inf_{t_i < s \le t}\{\sigma_i(t - s) + R(s)\}\right] \tag{8.8}$$

*with $\sigma_i^0$ is defined by*

$$\sigma_i^0(u) = \min_{1 \le j \le J}\left[r_j^i \cdot u + b_i^j - q_j(t_i)\right]$$

*and $q_j(t_i)$ is defined recursively by Equation (8.7). The backlog at time $t$ is defined recursively by*

$$w(t) = \max \left[ \begin{array}{l} \sup_{t_i < s \leq t} \{R(t) - R(s) - \sigma_i(t-s)\}, \\ R(t) - R(t_i) - \sigma_i^0(t-t_i) + w(t_i) \end{array} \right] \quad t \in I_i \tag{8.9}$$

**Proof:** Use the same notation as in the proof of Proposition 8.4.1 and define in addition

$$x(\tau) := R(t_i + \tau) - R(t_i)$$

We can now apply Theorem 8.3.1, with initial bucket levels equal to $q_j(t_i)$ as given in Equation (8.7) and with an initial buffer level equal to $w(t_i)$. The input-output characterization of this system is given by Equation (8.6), thus

$$x^*(\tau) = \sigma_i^0(\tau) \wedge [\sigma_i \otimes x'](\tau)$$

where

$$x'(\tau) = \begin{cases} x(\tau) + w(t_i) & \tau > 0 \\ x(\tau) & \tau \leq 0 \end{cases}$$

Hence, re-introducing the original notation, we obtain

$$R^*(t) - R^*(t_i) = \left[ \sigma_i^0(t-t_i) \wedge \inf_{t_i < s \leq t} \{\sigma_i(t-s) + R(s) - R(t_i) + w(t_i)\} \right]$$

which gives Equation (8.8).

The backlog at time $t$ follows immediately. □

Note that Theorem 8.4.1 provides a representation of $\overline{H}$. However, the representation is recursive: in order to compute $R^*(t)$, we need to compute $R^*(t_i)$ for all $t_i < t$.

## 8.5 Bibliographic Notes

[67] illustrates how the formulas in Section 8.4 form the basis for defining a renegotiable VBR service. It also illustrates that, if some inconsistency exists between network and user sides whether leaky buckets should be reset or not at every renegotiation step, then this may result in inacceptable losses (or service degradation) due to policing.

[12] analyzes the general concept of time varying shapers.

# Chapter 9

# Systems with Losses

All chapters have dealt up to now with lossless systems. This chapter shows that network calculus can also be applied to lossy systems, if we model them as a lossless system preceded by a 'clipper' [16, 17], which is a controller dropping some data when a buffer is full, or when a delay constraint would otherwise be violated. By applying once again Theorem 4.3.1, we obtain a representation formula for losses. We use this formula to compute various bounds. The first one is a bound on the loss rate in an element when both an arrival curve of the incoming traffic and a minimum service curve of the element are known. We use it next to bound losses in a complex with a complex service curve (e.g., VBR shapers) by means of losses with simpler service curves (e.g., CBR shapers). Finally, we extend the clipper, which models data drops due to buffer overflow, to a 'compensator', which models data accrual to prevent buffer underflow, and use it to compute explicit solutions to Skorokhod reflection mapping problem with two boundaries.

## 9.1 A Representation Formula for Losses

### 9.1.1 Losses in a Finite Storage Element

We consider a network element offering a service curve $\beta$, and having a finite storage capacity (buffer) $X$. We denote by $a$ the incoming traffic.

We suppose that the buffer is not large enough to avoid losses for all possible input traffic patterns, and we would like to compute the amount of data lost at time $t$, with the convention that the system is empty at time $t = 0$. We model losses as shown in Figure 9.1, where $x(t)$ is the data that has actually entered the system in the time interval $[0, t]$. The amount of data lost during the same period is therefore $L(t) = a(t) - x(t)$.

The model of Figure 9.1 replaces the original lossy element, by an equivalent concatenation a controller or regulator that separates the incoming flow $a$ in two separate flows, $x$ and $L$, and that we call *clipper*, following the denomination introduced in [17], together with the original system, which is now lossless for flow $x$.

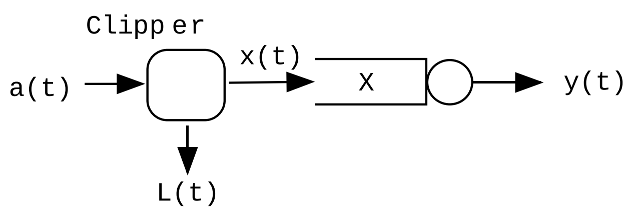



Figure 9.1: System with losses

The amount of data $(x(t)-x(s))$ that actually entered the system in any time interval $(s,t]$ is always bounded above by the total amount of data $(a(t)-a(s))$ that has arrived in the system during the same period. Therefore, for any $0 \le s \le t$, $x(t) \le x(s)+a(t)-a(s)$ or equivalently, using the linear idempotent operator introduced by Definition 4.1.5,

$$x(t) \le \inf_{0\le s\le t}\{a(t)-a(s)+x(s)\} = h_a(x)(t). \tag{9.1}$$

On the other hand, $x$ is the part of $a$ that does actually enter the system. If $y$ denotes its output, there is no loss for $x$ if $x(t)-y(t) \le X$ for any $t$. We do not know the exact mapping $y=\Pi(x)$ realized by the system, but we assume that $\Pi$ is isotone. So at any time $t$

$$x(t) \le y(t)+X = \Pi(x)(t)+X \tag{9.2}$$

The data $x$ that actually enters the system is therefore the maximum solution to (9.1) and (9.2), which we can recast as

$$x \le a \wedge \{\Pi(x)+X\} \wedge h_a(x), \tag{9.3}$$

and which is precisely the same equation as (4.33) with $W=X$ and $M=a$. Its maximal solution is given by

$$x = \overline{(\{\Pi+X\}\wedge h_a)}(a),$$

or equivalently, after applying Corollary 4.2.1, by

$$x = \left(\overline{(h_a\circ(\Pi+X))}\circ h_a\right)(a) = \left(\overline{(h_a\circ(\Pi+X))}\right)(a) \tag{9.4}$$

where the last equality follows from $h_a(a)=a$.

We do not know the exact mapping $\Pi$, but we know that $\Pi \ge \mathcal{C}_\beta$. We have thus that

$$x \geq \overline{(h_a \circ \mathcal{C}_{\beta+X})}(a). \tag{9.5}$$

The amount of lost data in the interval $[0, t]$ is therefore given by

$$\begin{aligned}
L(t) &= a(t) - x(t) \\
&= a(t) - \overline{h_a \circ \{\mathcal{C}_{\beta+X}\}}(a)(t) = a(t) - \inf_{n \in \mathbb{N}} \left\{ (h_a \circ \mathcal{C}_{\beta+X})^{(n)} \right\} (a)(t) \\
&= \sup_{n \in \mathbb{N}} \left\{ a(t) - (h_a \circ \mathcal{C}_{\beta+X})^{(n)} (a)(t) \right\} \\
&= \sup_{n \geq 0} \{ a(t) - \inf_{0 \leq s_{2n} \leq \ldots \leq s_2 \leq s_1 \leq t} \{ a(t) - a(s_1) + \beta(s_1 - s_2) + X \\
&\qquad + a(s_2) - \ldots + a(s_{2n}) \} \} \\
&= \sup_{n \in \mathbb{N}} \{ \sup_{0 \leq s_{2n} \leq \ldots \leq s_2 \leq s_1 \leq t} \{ a(s_1) - \beta(s_1 - s_2) - a(s_2) \\
&\quad + \ldots - a(s_{2n}) - nX \} \}.
\end{aligned}$$

Consequently, the loss process can be represented by the following formula:

$$L(t) \leq$$

$$\sup_{n \in \mathbb{N}} \left\{ \sup_{0 \leq s_{2n} \leq \ldots \leq s_2 \leq s_1 \leq t} \left\{ \sum_{i=1}^{n} \left[ a(s_{2i-1}) - a(s_{2i}) - \beta(s_{2i-1} - s_{2i}) - X \right] \right\} \right\} \tag{9.6}$$

If the network element is a greedy shaper, with shaping curve $\beta$, then $\Pi(x) = \mathcal{C}_\beta$, and the inequalities in (9.5) and (9.6) become equalities.

What the formula says is that losses up to time $t$ are obtained by summing the losses over all intervals $[s_{2i-1}, s_{2i}]$, where $s_{2i}$ marks the end of an overflow period, and where $s_{2i-1}$ is the last time before $s_{2i}$ when the buffer was empty. These intervals are therefore larger then the congestion intervals, and their number $n$ is smaller or eqaul to the number of congestion intervals. Figure 9.2 shows an example where $n = 2$ and where there are three congestion periods.

We will see in the next sections how the losses representation formula (9.6), can help us to obtain deterministic bounds on the loss process in some systems.

### 9.1.2 Losses in a Bounded Delay Element

Before moving to these applications, we first derive a representation formula for a similar problem, where data are discarded not because of a finite buffer limit, but because of a delay constraint: any entering data must have exited the system after at most $d$ unit of time, otherwise it is discarded. Such discarded data are called losses due to a delay constraint of $d$ time units.

As above, let $x$ be the part of $a$ that does actually enter the system, and let $y$ be its output. All the data $x(t)$ that has entered the system during $[0, t]$ must therefore have left at time $t + d$ at the latest, so that $x(t) - y(t + d) \leq 0$ for any $t$. Thus

$$x(t) \leq y(t + d) = \Pi(x)(t + d) = (\mathcal{S}_{-d} \circ \Pi)(x)(t), \tag{9.7}$$

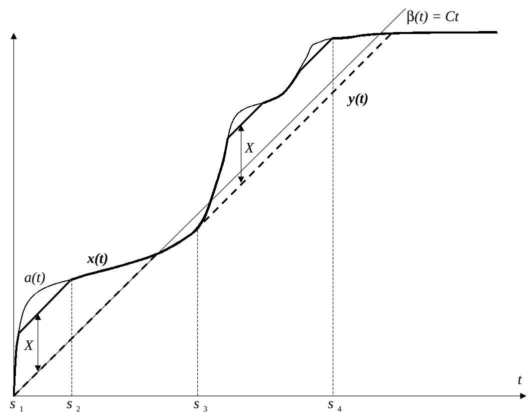



Figure 9.2: Losses in a constant rate shaper ($\beta = \lambda_C$). Fresh traffic $a$ is represented with a thin, solid line; accepted traffic $x$ is represented by a bold, solid line; the output process $y$ is represented by a bold, dashed line.

where $\mathcal{S}_{-d}$ is the shift operator (with forward shift of $d$ time units) given by Definition 4.1.7.

On the other hand, as in the previous example, the amount of data $(x(t) - x(s))$ that actually entered the system in any time interval $(s, t]$ is always bounded above by the total amount of data $(a(t) - a(s))$ that has arrived in the system during the same period. Therefore the data $x$ that actually enters the system is therefore the maximum solution to

$$x \le a \wedge (\mathcal{S}_{-d} \circ \Pi)(x) \wedge h_a(x), \tag{9.8}$$

which is

$$x = \overline{(\{\mathcal{S}_{-d} \circ \Pi\} \wedge h_a)}(a),$$

or equivalently, after applying Corollary 4.2.1, by

$$x = \left(\overline{h_a \circ (\{\mathcal{S}_{-d} \circ \Pi\})} \circ h_a\right)(a) = \left(\overline{h_a \circ \mathcal{S}_{-d} \circ \Pi}\right)(a). \tag{9.9}$$

Since $\Pi \ge \mathcal{C}_\beta$, we also have,

$$x \ge \left(\overline{h_a \circ \mathcal{S}_{-d} \circ \mathcal{C}_\beta}\right)(a). \tag{9.10}$$

The amount of lost data in the interval $[0, t]$ is therefore given by

$$L(t) \le \sup_{n \in \mathbb{N}} \left\{ a(t) - \left(h_a \circ \mathcal{S}_{-d} \circ \mathcal{C}_\beta\right)^{(n)}(a)(t) \right\}$$

which can be developed as

$$L(t) \leq \sup_{n\in\mathbb{N}} \left\{ \sup_{0\leq s_{2n}\leq\ldots\leq s_2\leq s_1\leq t} \left\{ \sum_{i=1}^{n} \left[a(s_{2i-1}) - a(s_{2i}) - \beta(s_{2i-1} + d - s_{2i})\right] \right\} \right\} \tag{9.11}$$

Once again, if $\Pi = \mathcal{C}_\beta$, then (9.11) becomes an equality.

We can also combine a delay constraint with a buffer constraint, and repeat the same reasoning, starting from

$$x \leq a \wedge \{\Pi(x) + X\} \wedge (\mathcal{S}_{-d} \circ \Pi)(x) \wedge h_a(x). \tag{9.12}$$

to obtain

$$\begin{aligned} L(t) \quad \leq \quad & \sup_{n\in\mathbb{N}}\{ \sup_{0\leq s_{2n}\leq\ldots\leq s_2\leq s_1\leq t} \{\sum_{i=1}^{n}[a(s_{2i-1}) - a(s_{2i}) \\ & -(\beta(s_{2i-1} + d - s_{2i}) \wedge \{\beta(s_{2i-1} - s_{2i}) + X\})]\}\}. \end{aligned} \tag{9.13}$$

This can be recast as a recursion on time if $t \in \mathbb{N}$, following the time method to solve (9.12) instead of the space method. This recurstion is established in [16].

## 9.2 Application 1: Bound on Loss Rate

Let us return to the case of losses due to buffer overflow, and suppose that in this section fresh traffic $a$ is constrained by an arrival curve $\alpha$.

The following theorem provide a bound on the loss rate $l(t) = L(t)/a(t)$, and is a direct consequence of the loss representation (9.6).

**Theorem 9.2.1 (Bound on loss rate).** *Consider a system with storage capacity $X$, offering a service curve $\beta$ to a flow constrained by an arrival curve $\alpha$. Then the loss rate $l(t) = L(t)/a(t)$ is bounded above by*

$$\hat{l}(t) = 1 - \inf_{0<s\leq t} \frac{\beta(s) + X}{\alpha(s)}. \tag{9.14}$$

**Proof:** With $\hat{l}(t)$ defined by (9.14), we have that for any $0 \leq u < v \leq t$,

$$1 - \hat{l}(t) = \inf_{0<s\leq t} \frac{\beta(s) + X}{\alpha(s)} \leq \frac{\beta(v-u) + X}{\alpha(v-u)} \leq \frac{\beta(v-u) + X}{a(v) - a(u)}$$

because $a(v) - a(u) \leq \alpha(v-u)$ by definition of an arrival curve. Therefore, for any $0 \leq u \leq v \leq t$,

$$a(v) - a(u) - \beta(v-u) - X \leq \hat{l}(t) \cdot [a(v) - a(u)].$$

For any $n \in \mathbb{N}_0 = \{1, 2, 3, ...\}$, and any sequence $\{s_k\}_{1 \leq k \leq 2n}$, with $0 \leq s_{2n} \leq \ldots \leq s_1 \leq t$, setting $v = s_{2i-1}$ and $u = s_{2i}$ in the previous equation, and summing over $i$, we obtain

$$\sum_{i=1}^{n} [a(s_{2i-1}) - a(s_{2i}) - \beta(s_{2i-1} - s_{2i}) - X] \leq \hat{l}(t) \cdot \sum_{i=1}^{n} [a(s_{2i-1}) - a(s_{2i})] .$$

Because the $s_k$ are increasing with $k$, the right hand side of this inequality is always less than, or equal to, $\hat{l}(t) \cdot a(t)$. Therefore we have

$$\begin{aligned} L(t) &\leq \sup_{n \in \mathbb{N}} \left\{ \sup_{0 \leq s_{2n} \leq \ldots \leq s_1 \leq t} \left\{ \sum_{i=1}^{n} [a(s_{2i-1}) - a(s_{2i}) - \beta(s_{2i-1} - s_{2i}) - X] \right\} \right\} \\ &\leq \hat{l}(t) \cdot a(t), \end{aligned}$$

which shows that $\hat{l}(t) \geq l(t) = L(t)/a(t)$. □

To have a bound independent of time $t$, we take the sup over all $t$ of (9.14), to get

$$\hat{l} = \sup_{t \geq 0} \hat{l}(t) = 1 - \inf_{t > 0} \frac{\beta(t) + X}{\alpha(t)}, \tag{9.15}$$

and retrieve the result of Chuang and Chang [15].

A similar result for losses due to delay constraint $d$, instead of finite buffer $X$, can be easily obtained, too:

$$\hat{l}(t) = 1 - \inf_{0 < s \leq t} \frac{\beta(s + d)}{\alpha(s)} \tag{9.16}$$

$$\hat{l} = 1 - \inf_{t > 0} \frac{\beta(t + d)}{\alpha(t)}. \tag{9.17}$$

## 9.3 Application 2: Bound on Losses in Complex Systems

As a particular application of the loss representation formula (9.6), we show how it is possible to bound the losses in a system offering a somewhat complex service curve $\beta$, by losses in simpler systems. The first application is the bound on the losses in a shaper by a system that segregates the resources (buffer, bandwidth) between a storage system and a policer. The second application deals with a VBR shaper, which is compared with two CBR shapers. For both applications, the losses in the original system are bounded along every sample path by the losses in the simpler systems. For congestion times however, the same conclusion does not always hold.

### 9.3.1 Bound on Losses by Segregation between Buffer and Policer

We will first compare the losses in two systems, having the same input flow $a(t)$.

The first system is the one of Figure 9.1 with service curve $\beta$ and buffer $X$, whose losses $L(t)$ are therefore given by (9.6).

The second system is made of two parts, as shown in Figure 9.3(a). The first part is a system with storage capacity $X$, that realizes some mapping $\Pi'$ of the input that is not explicitly given, but that is assumed to be isotone, and not smaller than $\Pi$ ($\Pi' \geq \Pi$). We also know that a first clipper discards data as soon as the total backlogged data in this system exceeds $X$. This operation is called *buffer discard*. The amount of buffer discarded data in $[0, t]$ is denoted by $L_{\text{Buf}}(t)$. The second part is a policer without buffer, whose output is the min-plus convolution of the accepted input traffic by the policer by $\beta$. A second clipper discards data as soon as the total output flow of the storage system exceeds the maximum input allowed by the policer. This operation is called *policing discard*. The amount of discarded data by policing in $[0, t]$ is denoted by $L_{\text{Pol}}(t)$.

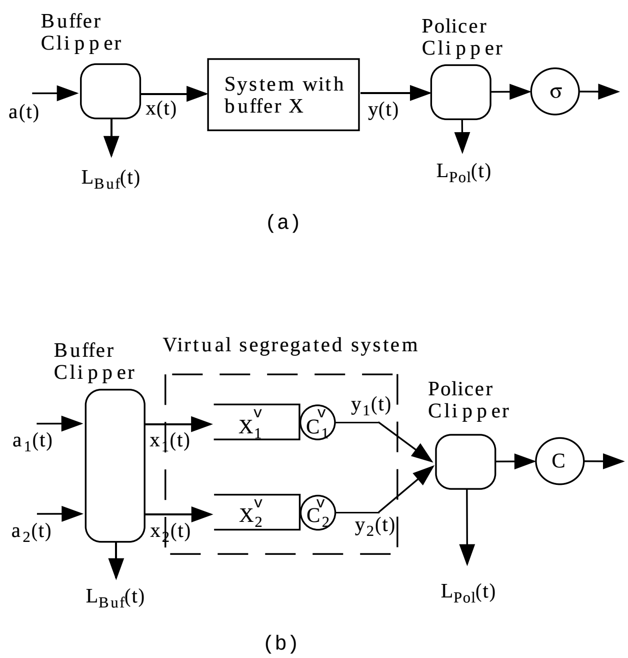



Figure 9.3: A storage/policer system with separation between losses due to buffer discard and to policing discard (a) A virtual segregated system for 2 classes of traffic, with buffer discard and policing discard, as used by Lo Presti et al [53] (b)

**Theorem 9.3.1.** *Let $L(t)$ be the amount of lost data in the original system, with service curve $\beta$ and buffer $X$.*

*Let $L_{\text{Buf}}(t)$ (resp. $L_{\text{Pol}}(t)$) be the amount of data lost in the time interval $[0,t]$ by buffer (resp. policing) discard, as defined above.*

*Then $L(t) \le L_{\text{Buf}}(t) + L_{\text{Pol}}(t)$.*

**Proof:** Let $x$ and $y$ denote respectively the admitted and output flows of the buffered part of the second system. Then the policer implies that $y = \beta \otimes x$, and any time $s$ we have

$$a(s) - L_{\text{Buf}}(s) - X = x(s) - X \le y(s) \le x(s) = a(s) - L_{\text{Buf}}(s).$$

which implies that for any $0 \le u \le v \le t$,

$$\begin{aligned} & y(v) - y(u) - \beta(v-u) \\ \ge\quad & (a(v) - L_{\text{Buf}}(v) - X) - (a(u) - L_{\text{Buf}}(u)) - \beta(v-u) \\ =\quad & a(v) - a(u) - \beta(v-u) - X - (L_{\text{Buf}}(v) - L_{\text{Buf}}(u)). \end{aligned}$$

We use the same reasoning as in the proof of Theorem 9.2.1: we pick any $n \in \mathbb{N}_0$ and any increasing sequence $\{s_k\}_{1\le k \le 2n}$, with $0 \le s_{2n} \le \ldots \le s_1 \le t$. Then we set $v = s_{2i-1}$ and $u = s_{2i}$ in the previous inequality, and we sum over $i$, to obtain

$$\begin{aligned} & \sum_{i=1}^{n} \left[ y(s_{2i-1}) - y(s_{2i}) - \beta(s_{2i-1} - s_{2i}) \right] \ge \\ & \qquad \sum_{i=1}^{n} \left[ a(s_{2i-1}) - a(s_{2i}) - \beta(s_{2i-1} - s_{2i}) - X \right] \\ & \qquad - \sum_{i=1}^{n} \left[ (L_{\text{Buf}}(s_{2i-1}) - L_{\text{Buf}}(s_{2i})) \right]. \end{aligned}$$

By taking the supremum over all $n$ and all sequences $\{s_k\}_{1\le k\le 2n}$, the left hand side is equal to $L_{\text{Pol}}(t)$, because of (9.6) (we can replace the inequality in (9.6) by an equality, because the output of the policer is $y = \beta \otimes x$). Since $\{s_k\}$ is a wide-sense increasing sequence, and since $L_{\text{Buf}}$ is a wide-sense increasing function, we obtain therefore

$$\begin{aligned} & L_{\text{Pol}}(t) \ge \\ & \quad \sup_{n\in\mathbb{N}} \left\{ \sup_{0 \le s_{2n} \le \ldots \le s_1 \le t} \left[ a(s_{2i-1}) - a(s_{2i}) - \beta(s_{2i-1} - s_{2i}) - X \right] \right\} - L_{\text{Buf}}(t) \\ = \quad & L(t) - L_{\text{Buf}}(t), \end{aligned}$$

which completes the proof. $\square$

Such a separation of resources between the “buffered system” and “policing system” is used in the estimation of loss probability for devising statistical CAC (Call Acceptance Control) algorithms as proposed by Elwalid et al [25], Lo Presti et al. [53]. The incoming traffic is separated in two classes. All variables relating to the first (resp. second) class are marked with an index 1 (resp. 2), so that $a(t) = a_1(t)+a_2(t)$. The original system is a CBR shaper ($\beta = \lambda_C$) and the storage system is a virtually segregated system as in Figure 9.3(b), made of 2 shapers with rates $C_1^v$ and $C_2^v$ and buffers $X_1^v$ and $X_2^v$. The virtual shapers are large enough to ensure that no loss occurs for all possible arrival functions $a_1(t)$ and $a_2(t)$. The total buffer space (resp. bandwidth) is larger than the original buffer space (resp. bandwidth): $X_1^v + X_2^v \geq X$ ($C_1^v + C_2^v \geq C$). However, the buffer controller discards data as soon as the total backlogged data in the virtual system exceeds $X$ and the policer controller discards data as soon as the total output rate of the virtual system exceeds $C$.

### 9.3.2 Bound on Losses in a VBR Shaper

In this second example, we consider of a "buffered leaky bucket" shaper [46] with buffer $X$, whose output must conform to a VBR shaping curve with peak rate $P$, sustainable rate $M$ and burst tolerance $B$ so that here the mapping of the element is $\Pi = \mathcal{C}_\beta$ with $\beta = \lambda_P \wedge \gamma_{M,B}$. We will consider two systems to bound these losses: first two CBR shapers in parallel (Figure 9.4(a)) and second two CBR shapers in tandem (Figure 9.4(b)). Similar results also holds for losses due to a delay constraint [50].

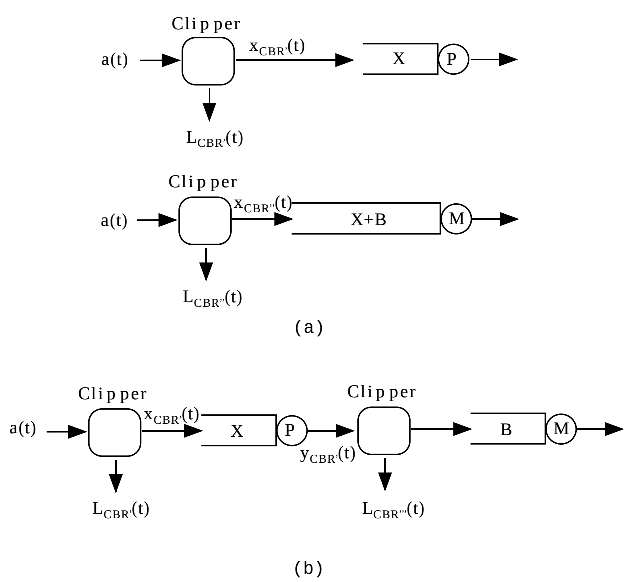



Figure 9.4: Two CBR shapers in parallel (a) and in tandem (b).

We will first show that the *amount of losses* during $[0, t]$ in this system is bounded by the sum of losses in two CBR shapers in parallel, as shown in Figure 9.4(a): the first one has buffer of size $X$ and rate $P$, whereas the second one has buffer of size $X + B$ and rate $M$. Both receive the same arriving traffic $a$ as the original VBR shaper.

**Theorem 9.3.2.** *Let $L_{\text{VBR}}(t)$ be the amount of lost data in the time interval $[0, t]$ in a VBR shaper with buffer $X$ and shaping curve $\beta = \lambda_P \wedge \gamma_{M,B}$, when the data that has arrived in $[0, t]$ is $a(t)$.*

*Let $L_{\text{CBR}'}(t)$ (resp. $L_{\text{CBR}''}(t)$) be the amount of lost data during $[0, t]$ in a CBR shaper with buffer $X$ (resp. $(X + B)$) and shaping curve $\lambda_P$ (resp. $\lambda_M$) with the same incoming traffic $a(t)$.*

*Then $L_{\text{VBR}}(t) \leq L_{\text{CBR}'}(t) + L_{\text{CBR}''}(t)$.*

**Proof:** The proof is again a direct application of (9.6). Pick any $0 \leq u \leq v \leq t$. Since $\beta = \lambda_P \wedge \gamma_{M,B}$,

$$\begin{aligned} & a(v) - a(u) - \beta(v - u) - X = \\ & \quad \{a(v) - a(u) - P(v - u) - X\} \vee \{a(v) - a(u) - M(v - u) - B - X\} \end{aligned}$$

Pick any $n \in \mathbb{N}_0$ and any increasing sequence $\{s_k\}_{1 \leq k \leq 2n}$, with $0 \leq s_{2n} \leq \ldots \leq s_1 \leq t$. Set $v = s_{2i-1}$ and $u = s_{2i}$ in the previous equation, and sum over $i$, to obtain

$$\begin{aligned} & \sum_{i=1}^{n} [a(s_{2i-1}) - a(s_{2i}) - \beta(s_{2i-1} - s_{2i}) - X] \\ & \quad = \sum_{i=1}^{n} [\{a(s_{2i-1}) - a(s_{2i}) - P(s_{2i-1} - s_{2i}) - X\} \\ & \qquad\qquad \vee \{a(s_{2i-1}) - a(s_{2i}) - M(s_{2i-1} - s_{2i}) - B - X\} \\ & \quad \leq \sum_{i=1}^{n} [a(s_{2i-1}) - a(s_{2i}) - P(s_{2i-1} - s_{2i}) - X] \\ & \qquad\qquad + \sum_{i=1}^{n} [a(s_{2i-1}) - a(s_{2i}) - M(s_{2i-1} - s_{2i}) - B - X] \\ & \quad \leq L_{\text{CBR}'}(t) + L_{\text{CBR}''}(t), \end{aligned}$$

because of (9.6). By taking the supremum over all $n$ and all sequences $\{s_k\}_{1 \leq k \leq 2n}$ in the previous inequality, we get the desired result. □

A similar exercise shows that the amount of losses during $[0, t]$ in the VBR system is also bounded above by the sum of losses in two CBR shapers in cascade as shown in Figure 9.4(b): the first one has buffer of size $X$ and rate $P$, and receives the same arriving traffic $a$ as the original VBR shaper, whereas its output is fed into the second one with buffer of size $B$ and rate $M$.

**Theorem 9.3.3.** *Let $L_{\text{VBR}}(t)$ be the amount of lost data in the time interval $[0, t]$ in a VBR shaper with buffer $X$ and shaping curve $\beta = \lambda_P \wedge \gamma_{M,B}$, when the data that has arrived in $[0, t]$ is $a(t)$.*

*Let $L_{\text{CBR}'}(t)$ (resp. $L_{\text{CBR}''}(t)$) be the amount of lost data during $[0, t]$ in a CBR shaper with buffer $X$ (resp. $B$) and shaping curve $\lambda_P$ (resp. $\lambda_M$) with the same incoming traffic $a(t)$ (resp. the output traffic of the first CBR shaper).*

*Then $L_{\text{VBR}}(t) \leq L_{\text{CBR}'}(t) + L_{\text{CBR}''}(t)$.*

The proof is left as an exercise.

Neither of the two systems in Figure 9.4 gives the better bound for any arbitrary traffic pattern. For example, suppose that the VBR system parameters are $P = 4$, $M = 1$, $B = 12$ and $X = 4$, and that the traffic is a single burst of data sent at rate $R$ during four time units, so that

$$a(t) = \begin{cases} R \cdot t & \text{if} \quad 0 \leq t \leq 4 \\ 4R & \text{if} \quad t \geq 4 \end{cases}$$

If $R = 5$, both the VBR system and the parallel set of the two $\text{CBR}'$ and $\text{CBR}''$ systems are lossless, whereas the amount of lost data after five units of time in the tandem of the two $\text{CBR}'$ and $\text{CBR}'''$ systems is equal to three.

On the other hand, if $R = 6$, the amount of lost data after five units of time in the VBR system, the parallel system ($\text{CBR}'$ and $\text{CBR}''$) and the tandem system ($\text{CBR}'$ and $\text{CBR}'''$) are respectively equal to four, eight and seven.

Interestingly enough, whereas both systems of Figure 9.4 will bound the *amount of losses in the original system*, it is no longer so for the *congestion periods*, i.e. the time intervals during which losses occur. The tandem system does not offer a bound on the congestion periods, contrary to the parallel system [50].

## 9.4 Solution to Skohorkhod's Reflection Problem with Two Boundaries

To obtain the model of Figure 9.1, we have added a regulator – called *clipper* – before the system itself, whose input $x$ is the maximal input ensuring a lossless service, given a finite storage capacity $X$. The clipper eliminates the fraction of fresh traffic $a$ that exceeds $x$. We now generalize this model by adding a second regulator *after* the lossless system, whose output is denoted with $y$, as shown on Figure 9.5. This regulator complements $y$, so that the output of the full process is now a given function $b \in \mathcal{F}$. The resulting process $N = y - b$ is the amount of traffic that needs to be fed to prevent the storage system to enter in starvation. $N$ compensates for possible buffer underflows, hence we name this second regulator *compensator*.

We can explicitly compute the loss process $L$ and the "compensation" process $N$, from the arrival process $a$ and the departure process $b$, using, once again, Theorem 4.3.1. We are looking for the maximal solution

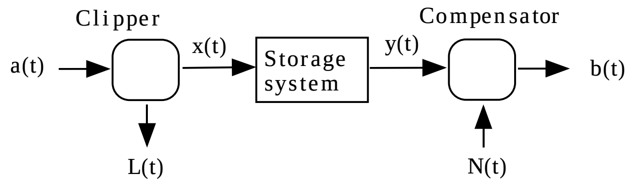



Figure 9.5: A storage system representing the variables used to solve Skorokhod's reflection problem with two boundaries

$$\vec{x}(t) = [x(t) \quad y(t)]^T,$$

where $T$ denotes transposition, to the set of inequalities

$$x(t) \leq \inf_{0 \leq s \leq t} \{a(t) - a(s) + x(s)\} \tag{9.18}$$

$$x(t) \leq y(t) + X \tag{9.19}$$

$$y(t) \leq x(t) \tag{9.20}$$

$$y(t) \leq \inf_{0 \leq s \leq t} \{b(t) - b(s) + y(s)\}. \tag{9.21}$$

The two first inequalities are identical to (9.1) and to (9.2). The two last inequalities are the dual constraints on $y$. We can therefore recast this system as

$$x \leq a \wedge h_a(x) \wedge \{y + X\} \tag{9.22}$$

$$y \leq b \wedge x \wedge h_b(x). \tag{9.23}$$

This is a system of min-plus linear inequalities, whose solution is

$$\vec{x} = \overline{\mathcal{L}}_H(\vec{a}) = \mathcal{L}_{\overline{H}}(\vec{a}),$$

where $H$ and $\vec{a}$ are defined as

$$\vec{a}(t) = [a(t) \quad b(t)]^T$$

$$H(t,s) = \begin{bmatrix} a(t) - a(s) & \delta_0(t-s) + X \\ \delta_0(t-s) & b(t) - b(s) \end{bmatrix}.$$

for all $0 \leq s \leq t$. Instead of computing $\overline{H}$, we go faster by first computing the maximal solution of (9.23). Using properties of the linear idempotent operator, we get

$$y = \overline{h}_b(x \wedge b) = h_b(x \wedge b) = h_b(x) \wedge h_b(b) = h_b(x).$$

Next we replace $y$ by $h_b(x)$ in (9.22), and we compute its maximal solution, which is

$$x = \overline{h_a \wedge \{h_b + X\}}(a).$$

We work out the sub-additive closure using Corollary 4.2.1, and we obtain

$$x = \overline{(h_a \circ \{h_b + X\})}(a) \tag{9.24}$$

and thus

$$y = \left(h_b \circ \overline{h_a \circ \{h_b + X\}}\right)(a). \tag{9.25}$$

After some manipulations, we get

$$N(t) = b(t) - y(t) =$$

$$\sup_{n \in \mathbb{N}} \left\{ \sup_{0 \le s_{2n+1} \le \ldots \le s_2 \le s_1 \le t} \left\{ \sum_{i=1}^{2n+1} (-1)^i (a(s_i) - b(s_i)) \right\} - nX \right\} \tag{9.26}$$

$$L(t) = a(t) - x(t) =$$

$$\sup_{n \in \mathbb{N}} \left\{ \sup_{0 \le s_{2n} \le \ldots \le s_2 \le s_1 \le t} \left\{ \sum_{i=1}^{2n} (-1)^{i+1} (a(s_i) - b(s_i)) \right\} - nX \right\}. \tag{9.27}$$

Interestingly enough, these two functions are the solution of the so-called Skorokhod reflection problem with two fixed boundaries [70, 35].

Let us describe this reflection mapping problem following the exposition of [42]. We are given a lower boundary that will be taken here as the origin, an upper boundary $X > 0$, and a *free process* $z(t) \in \mathbb{R}$ such that $0 \le z(0-) \le X$. Skorokhod's reflection problem looks for functions $N(t)$ (*lower boundary process*) and $L(t)$ (*upper boundary process*) such that

1. The *reflected process*

   $$W(t) = z(t) + N(t) - L(t) \tag{9.28}$$

   is in $[0, X]$ for all $t \ge 0$.

2. Both $N(t)$ and $L(t)$ are non decreasing with $N(0-) = L(0-) = 0$, and $N(t)$ (resp. $L(t)$) increases only when $W(t) = 0$ (resp. $W(t) = X$), i.e., with $1_A$ denoting the indicator function of $A$

   $$\int_0^\infty 1_{\{W(t)>0\}} dN(t) = 0 \tag{9.29}$$

   $$\int_0^\infty 1_{\{W(t)<X\}} dL(t) = 0 \tag{9.30}$$

The solution to this problem exists and is unique [35]. When only one boundary is present, explicit formulas are available. For instance, if $X \to \infty$, then there is only one lower boundary, and the solution is easily found to be

$$\begin{aligned} N(t) &= -\inf_{0 \le s \le t} \{z(s)\} \\ L(t) &= 0. \end{aligned}$$

If $X < \infty$, then the solution can be constructed by successive approximations but, to our knowledge, no solution has been explicitly obtained. The following theorem gives such explicit solutions for a continuous VF function $z(t)$. A VF function (VF standing for Variation Finie [35, 66]) $z(t)$ on $\mathbb{R}^+$ is a function such that for all $t > 0$

$$\sup_{n\in\mathbb{N}_0} \sup_{0=s_n<s_{n-1}<\ldots<s_1<s_0=t} \left\{ \sum_{i=0}^{n-1} |z(s_i) - z(s_{i+1})| \right\} < \infty.$$

VF functions have the following property [66]: $z(t)$ is a VF function on $\mathbb{R}^+$ if and only if it can be written as the difference of two wide-sense increasing functions on $\mathbb{R}^+$.

**Theorem 9.4.1 (Skorokhod's reflection mapping).** *Let the free process $z(t)$ be a continuous VF function on $\mathbb{R}^+$. Then the solution to Skorokhod's reflection problem on $[0,X]$ is*

$$N(t) = \sup_{n\in\mathbb{N}} \left\{ \sup_{0\le s_{2n+1}\le\ldots\le s_2\le s_1\le t} \left\{ \sum_{i=1}^{2n+1} (-1)^i z(s_i) \right\} - nX \right\} \quad (9.31)$$

$$L(t) = \sup_{n\in\mathbb{N}} \left\{ \sup_{0\le s_{2n}\le\ldots\le s_2\le s_1\le t} \left\{ \sum_{i=1}^{2n} (-1)^{i+1} z(s_i) \right\} - nX \right\}. \quad (9.32)$$

**Proof:** As $z(t)$ is a VF function on $[0,\infty)$, there exist two increasing functions $a(t)$ and $b(t)$ such that $z(t) = a(t) - b(t)$ for all $t \ge 0$. As $z(0) \ge 0$, we can take $b(0) = 0$ and $a(0) = z(0)$. Note that $a, b \in \mathcal{F}$.

We will show now that $L = a - x$ and $N = b - y$, where $x$ and $y$ are the maximal solutions of (9.22) and (9.23), are the solutions of Skorokhod's reflection problem.

First note that

$$W(t) = z(t) + N(t) - L(t) = (a(t) - b(t)) + (b(t) - y(t)) - (a(t) - x(t)) = x(t) - y(t)$$

is in $[0,X]$ for all $t \ge 0$ because of (9.19) and (9.20).

Second, because of (9.21), note that $N(0) = b(0) - y(0) = 0$ and that for any $t > 0$ and $0 \le s < t$, $N(t) - N(s) = b(t) - b(s) + y(s) - y(t) \ge 0$, which shows that $N(t)$ is non decreasing. The same properties can be deduced for $L(t)$ from (9.18).

Finally, if $W(t) = x(t) - y(t) > 0$, there is some $s^\star \in [0,t]$ such that $y(t) = y(s^\star) + b(t) - b(s^\star)$ because $y$ is the maximal solution satisfying (9.20) and (9.21). Therefore for all $s \in [s^\star, t]$,

$$0 \le N(t) - N(s) \le N(t) - N(s^\star) = b(t) - b(s^\star) + y(s^\star) - y(t) = 0$$

which shows that $N(t) - N(s) = 0$ and so that $N(t)$ is non increasing if $W(t) > 0$. A similar reasoning shows that $L(t)$ is non increasing if $W(t) < X$.

Consequently, $N(t)$ and $L(t)$ are the lower and upper reflected processes that we are looking for. We have already computed them: they are given by (9.26) and (9.27). Replacing $a(s_i) - b(s_i)$ in these two expressions by $z(s_i)$, we establish (9.31) and (9.32). $\square$

## 9.5 Bibliographic Notes

The clipper was introduced by Cruz and Tenaja, and was extended to get the loss representation formula presented in this chapter in [16, 50]. Explicit expressions when operator $\Pi$ is a general, time-varying operator, can be found in [16]. We expect results of this chapter to form a starting point for obtaining bounds on probabilities of loss or congestion for lossy shapers with complex shaping functions; the method would consist in applying known bounds to virtual systems and take the minimum over a set of virtual systems.

# Bibliography


[1] R. Agrawal, R. L. Cruz, C. Okino, and R. Rajan. A framework for adapative service guarantees. In *Proc. Allerton Conf on Comm, Control and Comp, Monticello, IL*, Sept 1998.

[2] R. Agrawal, R. L. Cruz, C. Okino, and R. Rajan. Performance bounds for flow control protocols. *IEEE/ACM Transactions on Networking (7) 3*, pages 310–323, June 1999.

[3] M. Andrews. Instability of fifo in session-oriented networks. In *Eleventh Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2000)*, January 2000.

[4] J. C. R. Bennett, Benson K., Charny A., Courtney W. F., and J.-Y. Le Boudec. Delay jitter bounds and packet scale rate guarantee for expedited forwarding. In *Proceedings of Infocom*, April 2001.

[5] J. C. R. Bennett, Benson K., Charny A., Courtney W. F., and J.-Y. Le Boudec. Delay jitter bounds and packet scale rate guarantee for expedited forwarding. *ACM/IEEE Transactions on Networking*, to appear, 2002.

[6] J.C.R. Bennett and H. Zhang. Wf2q: Worst-case fair weighted fair queuing. In *Proceedings of Infocom*, Mar 1996.

[7] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. An architecture for differentiated services, December 1998. RFC 2475, IETF.

[8] C. S. Chang. Stability, queue length and delay, part i: Deterministic queuing networks. Technical Report Technical Report RC 17708, IBM, 1992.

[9] C.-S. Chang, W.-J. Chen, and H.-Y. Hunag. On service guarantees for input buffered crossbar switches: A capacity decomposition approach by birkhoff and von neumann. In *Proc of IWQOS 99*, March 1999.

[10] C.S. Chang. On deterministic traffic regulation and service guarantee: A systematic approach by filtering. *IEEE Transactions on Information Theory*, 44:1096–1107, August 1998.

[11] C.S. Chang. *Performance Guarantees in Communication Networks*. Springer-Verlag, New York, 2000.

[12] C.S. Chang and R. L. Cruz. A time varying filtering theory for constrained traffic regulation and dynamic service guarantees. In *Preprint*, July 1998.

[13] A. Charny and J.-Y. Le Boudec. Delay bounds in a network with aggregate scheduling. In *First International Workshop on Quality of future Internet Services*, Berlin, Germany, September 2000.

[14] I. Chlamtac, A. Faragó, H. Zhang, and A. Fumagalli. A deterministic approach to the end-to-end analysis of packet flows in connection oriented networks. *IEEE/ACM transactions on networking*, (6)4:422–431, 08 1998.

[15] J.-F. Chuang, C.-M.and Chang. Deterministic loss ratio quality of service guarantees for high speed networks. *IEEE Communications Letters*, 4:236–238, July 2000.

[16] R. Cruz, C.-S. Chang, J.-Y. Le Boudec, and P. Thiran. A min-plus system theory for constrained traffic regulation and dynamic service guarantees. Technical Report SSC/1999/024, EPFL, July 1999.

[17] R. Cruz and M. Taneja. An analysis of traffic clipping. In *Proc 1998 Conf on Information Science & Systems, Princeton University*, 1998.

[18] R. L. Cruz. Sced+ : Efficient management of quality of service guarantees. In *IEEE Infocom'98, San Francisco*, March 1998.

[19] R.L. Cruz. A calculus for network delay, part i: Network elements in isolation. *IEEE Trans. Inform. Theory, vol 37-1*, pages 114–131, January 1991.

[20] R.L. Cruz. A calculus for network delay, part ii: Network analysis. *IEEE Trans. Inform. Theory, vol 37-1*, pages 132–141, January 1991.

[21] B. Davie, A. Charny, F. Baker, J. Bennett, K. Benson, J.-Y. Le Boudec, A. Chiu, W. Courtney, S. Davari, V. Firoiu, C. Kalmanek, K. K. Ramakrishnam, and D. Stiliadis. An expedited forwarding phb, April 2001. Work in Progress, Internet Draft, `ftp://ds.internic.net/internet-drafts/draft-ietf-diffserv-rfc2598bis-02.txt`.

[22] G. De Veciana, July 1996. Private Communication.

[23] A. Demers, S. Keshav, and S. Shenker. Analysis and simulation of a fair queuing algorithm. *Journal of Internetworking Research and Experience*, pages 3–26, Oct 1990.

[24] N. G. Duffield, K. K. Ramakrishan, and A. R. Reibman. Save: An algorithm for smoothed adaptative video over explicit rate networks. *IEEE/ACM Transactions on Networking*, 6:717–728, Dec 1998.

[25] A. Elwalid, Mitra D., and R. Wenworth. A new approach for allocating buffers and bandwidth to heterogeneous, regulated traffic in ATM node. *IEEE Journal of Selected Areas in Communications*, 13:1048–1056, August 1995.

[26] Baccelli F., Cohen G., Olsder G. J., , and Quadrat J.-P. *Synchronization and Linearity, An Algebra for Discrete Event Systems*. John Wiley and Sons, 1992.

[27] W.-C. Feng and J. Rexford. Performance evaluation of smoothing algorithms for transmitting variable-bit-rate video. *IEEE Transactions on Multimedia*, 1:302–312, Sept 1999.

[28] L. Georgiadis, R. Guérin, V. Peris, and R. Rajan. Efficient support of delay and rate guarantees in an internet. In *Proceedings of Sigcomm'96*, pages 106–116, August 1996.

[29] P. Goyal, S. S. Lam, and H. Vin. Determining end-to-end delay bounds in heterogeneous networks. In *5th Int Workshop on Network and Op. Sys support for Digital Audio and Video, Durham NH*, April 1995.

[30] R. Guérin and V. Peris. Quality-of-service in packet networks - basic mechanisms and directions. *Computer Networks and ISDN, Special issue on multimedia communications over packet-based networks*, 1998.

[31] R. Guérin and V. Pla. Aggregation and conformance in differentiated service networks – a case study. Technical Report Research Report, U Penn, http://www.seas.upenn.edu:8080/ guerin/publications/aggreg.pdf, August 2000.

[32] Jeremy Gunawardena. From max-plus algebra to nonexpansive mappings: a nonlinear theory for discrete event systems. *pre-print*, 1999.

[33] Sariowan H., Cruz R. L., and Polyzos G. C. Scheduling for quality of service guarantees via service curves. In *Proceedings ICCCN'95*, pages 512–520, Sept 1995.

[34] B. Hajek. Large bursts do not cause instability. *IEEE Trans on Aut Control*, 45:116–118, Jan 2000.

[35] J. M. Harrison. *Brownian Motion and Stochastic Flow Systems*. Wiley, New-York, 1985.

[36] J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski. Assured forwarding phb group, June 1999. RFC 2597, IETF.

[37] Golestani S. J. A self clocked fair queuing scheme for high speed applications. In *Proceedings of Infocom '94*, 1994.

[38] F. Farkas J. Y. Le Boudec. A delay bound for a network with aggregate scheduling. In *Proceedings of the Sixteenth UK Teletraffic Symposium on Management of Quality of Service*, page 5, Harlow, UK, May 2000.

[39] Y. Jiang. Delay bounds for a network of guaranteed rate servers with fifo aggregation. Pre-print, National University of Singapore, 2001.

[40] C Kalmanek, H. Kanakia, and R. Restrick. Rate controlled servers for very high speed networks. In *IEEE Globecom'90, vol 1*, pages 12–20, 1990.

[41] Keshav. *Computer Networking: An Engineering Approach*. Prentice Hall, Englewood Cliffs, New Jersey 07632, 1996.

[42] T. Konstantopoulos and V. Anantharam. Optimal flow control schemes that regulate the burstiness of traffic. *IEEE/ACM Transactions on Networking*, 3:423–432, August 1995.

[43] Cruz R. L. and Okino C. M. Service guarantees for window flow control. In *34th Allerton Conf of Comm., Cont., and Comp. Monticello, IL*, Oct 1996.

[44] Gun L. and R. Guérin. Bandwidth management and congestion control framework of the broadband network architecture. *Bandwidth management and congestion control framework of the broadband network architecture, vol 26*, pages 61–78, 1993.

[45] Zhang L. A new traffic control algorithm for packet switching networks. In *Proceedings of ACM Sigcomm '90*, 1990.

[46] J.-Y. Le Boudec. Application of network calculus to guaranteed service networks. *IEEE Transactions on Information Theory*, 44:1087–1096, May 1998.

[47] J.-Y. Le Boudec. Some properties of variable length packet shapers. In *Proc ACM Sigmetrics / Performance '01*, 2001.

[48] J.-Y. Le Boudec and A. Charny. Packet scale rate guarantee for non-fifo nodes. Technical Report DSC200138, EPFL-DSC, http://dscwww.epfl.ch./EN/publications/documents/tr01_038.pdf, July 2001.

[49] J.-Y. Le Boudec and G. Hebuterne. Comment on a deterministic approach to the end-to-end analysis of packet flows in connection oriented network. *IEEE/ACM Transactions on Networking*, February 2000.

[50] J.-Y. Le Boudec and P. Thiran. Network calculus viewed as a min-plus system theory applied to communication networks. Technical Report SSC/1998/016, EPFL, April 1998.

[51] J.-Y. Le Boudec and O. Verscheure. Optimal smoothing for guaranteed service. Technical Report DSC2000/014, EPFL, March 2000.

[52] J. Liebeherr, D.E. Wrege, and Ferrari D. Exact admission control for networks with bounded delay services. *ACM/IEEE transactions on networking*, 4:885–901, 1996.

[53] F. Lo Presti, Z.-L. Zhang, D. Towsley, and J. Kurose. Source time scale and optimal buffer/bandwidth trade-off for regulated traffic in a traffic node. *IEEE/ACM Transactions on Networking*, 7:490–501, August 1999.

[54] S. H. Low and P. P. Varaiya. A simple theory of traffic and resource allocation in atm. In *Globecom'91*, pages 1633–1637, December 1991.

[55] J. Y. Le Boudec M. Vojnovic. Stochastic analysis of some expedited forwarding networks. In *Proceedings of Infocom 2002*, New-York, June 2002.

[56] J. M. McManus and K.W. Ross. Video-on-demand over ATM: Constant-rate transmission and transport. *IEEE Journal on Selected Areas in Communications*, 7:1087–1098, Aug 1996.

[57] J. Naudts. *A Scheme for Multiplexing ATM Sources*. Chapman Hill, 1996.

[58] J. Naudts. Towards real-time measurement of traffic control parameters. *Computer networks*, 34:157–167, 2000.

[59] Clayton M. Okino. A framework for performance guarantees in communication networks, 1998. Ph.D. Dissertation, UCSD.

[60] A. K. Parekh and R. G. Gallager. A generalized processor sharing approach to flow control in integrated services networks: The single node case. *IEEE/ACM Trans. Networking, vol 1-3*, pages 344–357, June 1993.

[61] A. K. Parekh and R. G. Gallager. A generalized processor sharing approach to flow control in integrated services networks: The multiple node case. *IEEE/ACM Trans. Networking, vol 2-2*, pages 137–150, April 1994.

[62] Vinod Peris. Architecture for guaranteed delay service in high speed networks, 1997. Ph.D. Dissertation, University of Maryland, `http://www.isr.umd.edu`.

[63] Fabrice P. Guillemin Pierre E. Boyer, Michel J. Servel. The spacer-controller: an efficient upc/npc for atm networks. In *ISS '92, Session A9.3, volume 2*, October 1992.

[64] Agrawal R. and Rajan R. Performance bounds for guaranteed and adaptive services, December 1996. IBM Technical Report RC 20649.

[65] J. Rexford and D. Towsley. Smoothing variable-bit-rate video in an internetwork. *IEEE/ACM Transactions on Networking*, 7:202–215, April 1999.

[66] H. L. Royden. *Real Analysis*. Mc-Millan, New-York, 2 edition, 1968.

[67] J. Y. Le Boudec S. Giordano. On a class of time varying shapers with application to the renegotiable variable bit rate service. *Journal on High Speed Networks*, 9(2):101–138, June 2000.

[68] J. D. Salehi, Z.-L. Zhang, J. F. Kurose, and D. Towsley. Supporting stored video: Reducing rate variability and end-to-end resource requirements through optimal smoothing. *IEEE/ACM Transactions on Networking*, 6:397–410, Dec 1998.

[69] H. Sariowan. A service curve approach to performance guarantees in integrated service networks, 1996. Ph.D. Dissertation, UCSD.

[70] A. Skorokhod. Stochastic equations for diffusion processes in a bounded region. *Theory of Probability and its Applications*, 6:264–274, 1961.

[71] D. Stiliadis and A. Varma. Rate latency servers: a general model for analysis of traffic scheduling algorithms. In *IEEE Infocom '96*, pages 647–654, 1991.

[72] Rockafellar R. T. *Convex Analysis*. Princeton University Press, Princeton, 1970.

[73] L. Tassiulas and L. Georgiadis. Any work conserving policy stabilizes the ring with spatial reuse. *IEEE/ACM Transactions on Networking*, pages 205–208, April 1996.

[74] Lothar Thiele, Samarjit Chakraborty, and Martin Naedele. Real-time calculus for scheduling hard real-time systems. In *ISCAS*, Geneva, May 2000.

[75] P. Thiran, J.-Y. Le Boudec, and F. Worm. Network calculus applied to optimal multimedia smoothing. In *Proc of Infocom 2001*, April 2001.

[76] D Verma, H. Zhang, and D. Ferrari. Guaranteeing delay jitter bounds in packet switching networks. In *Proceedings of Tricomm '91, Chapel Hill*, pages 35–46, April 1991.

[77] H. Zhang. Service disciplines for guaranteed performance service in packet switching networks. *Proceedings of the IEEE*, 1996.

[78] H. Zhang and D. Ferrari. Rate controlled service disciplines. *Journal of High Speed Networks*, 3 No 4:389–412, August 1994.

[79] Hongbiao Zhang. A note on deterministic end-to-end delay analysis in connection oriented networks. In *Proc of IEEE ICC'99, Vancouver, pp 1223–1227*, 1999.

[80] Z.-L. Zhang and Duan Z. Fundamental trade-offs in aggregate packet scheduling. In *SPIE Vol. 4526*, August 2001.

# Index