A SHORT TUTORIAL ON NETWORK CALCULUS
I: FUNDAMENTAL BOUNDS IN COMMUNICATION NETWORKS

Jean-Yves Le Boudec and Patrick Thiran

ICA-DSC, EPFL
CH-1015 Lausanne, Switzerland
`http://icawww.epfl.ch`

ABSTRACT

Network Calculus is a collection of results based on Min-Plus algebra, which applies to deterministic queuing systems found in communication networks. It can be used for example to understand the computations for delays used in the IETF guaranteed service, why re-shaping delays can be ignored in shapers or spacer-controllers, a common model for schedulers, etc. This short tutorial presents the basic results of network calculus and their application to some fundamental performance bounds in communication networks.

1. INTRODUCTION

Network Calculus is a set of recent developments which provide a deep insight into flow problems encountered in networking. The foundation of network calculus lies in the mathematical theory of dioids, and in particular, the Min-Plus dioid (also called Min-Plus algebra). With network calculus, we are able to understand some fundamental properties of integrated services networks, of window flow control, of scheduling and of buffer or delay dimensioning. These two companion papers [1] are a very short introduction to this theory.

Network calculus can be viewed as the system theory that applies to computer networks. The main difference with traditional system theory, as the one which was so successfully applied to design electronic circuits, is that here we consider another algebra, where the operations are changed as follows: addition becomes computation of the minimum, multiplication becomes addition.

Let us illustrate this difference with an example. Consider a very simple circuit, such as the RC cell represented in Figure 1. If the input signal is the voltage $x(t) \in \mathbb{R}$, then the output $y(t) \in \mathbb{R}$ of this simple circuit is the convolution of x by the impulse response of this circuit, which is here $h(t) = \exp(-t/RC)/RC$ for $t \geq 0$:

$$y(t) = (h \otimes x)(t) = \int_0^t h(t-s)x(s)ds.$$

Consider now a node of a communication network, which is idealized as a (greedy) shaper. A (greedy) shaper is a device that forces an input flow $x(t)$ to have an output $y(t)$ that conforms to a given set of rates according to a traffic enveloppe σ (the shaping curve), at the expense of possibly delaying bits in the buffer. Here the input and output 'signals' are cumulative flow, defined as the number of bits seen on the data flow in time interval $[0, t]$. These functions are non-decreasing with time t. We will denote by $\mathcal{G}$ the set of non-negative wide-sense increasing functions and by $\mathcal{F}$ denote the set of wide-sense increasing functions (or sequences) such that $f(t) = 0$ for $t < 0$. Parameter t can be continuous or discrete. We will see in this paper that x and y are linked by the relation

$$y(t) = (\sigma \otimes x)(t) = \inf_{s:0 \leq s \leq t} \{\sigma(t-s) + x(s)\}\,. \tag{1}$$

This relation defines the min-plus convolution between σ and x.

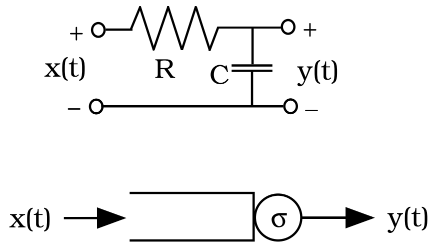

Figure 1: Traditional system theory for an elementary circuit (top) and min-plus system theory for a shaper (bottom).

This paper reviews the basic concepts of network calculus, namely the way we characterize the 'signals' (i.e. the flows) via arrival curves (Section 2) and the 'system' (e.g., the network node) via a service curve (Section 3). These tools will enable us to derive some deterministic performance bounds on quantities such delays and backlogs (Section 4), which are defined as follows, for a lossless system with input flow $x(t)$ and output flow $y(t)$:

Definition 1 (Backlog and Delay) *The* backlog *at time t is* $x(t) - y(t)$*, the* virtual delay *at time t is*

$$d(t) = \inf\left\{\tau \geq 0 : x(t) \leq y(t+\tau)\right\}.$$

The backlog is the amount of bits that are held inside the system; if the system is a single buffer, it is the queue length. In contrast, if the system is more complex, then the backlog is the number of bits "in transit", assuming that we can observe input and output simultaneously. The virtual delay at time t is the delay that would be experienced by a bit arriving at time t if all bits received before it are served before it. If we plot $x(t)$ and $y(t)$ versus t, the backlog is the vertical deviation between these two curves. The virtual delay is the horizontal deviation.

We will conclude the paper with 'the linear time-invariant system' of communication network: the shaper. The interested reader is also referred to the pioneering work of Cruz [5], Chang [3], Agrawal and Rajan[4].

2. ARRIVAL CURVES

To provide guarantees to data flows requires some specific support in the network; as a counterpart, the traffic sent by sources needs to be limited. With integrated services networks (ATM or the integrated services internet), this is done by using the concept of arrival curve, defined below.

Definition 2 (Arrival Curve) *Given a wide-sense increasing function α defined for $t \geq 0$ (namely $\alpha \in \mathcal{F}$), we say that a flow x is constrained by α if and only if for all $s \leq t$:*

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

Note that this is equivalent to imposing that for all $t \geq 0$

$$x(t) \leq \inf_{0 \leq s \leq t}\left\{\alpha(t-s) + x(s)\right\} = (\alpha \otimes x)(t) \quad (2)$$

The simplest arrival curve is $\alpha(t) = Rt$. Then the constraint means that, on any time window of width τ, the number of bits for the flow is limited by $R\tau$. We say in that case that the flow is peak rate limited. This occurs if we know that the flow is arriving on a link whose physical bit rate is limited by R bits/sec. A flow where the only constraint is a limit on the peak rate is often (improperly) called a "constant bit rate" (CBR) flow.

More generally, because of their relationship with leaky buckets, we will often use *affine* arrival curves $\gamma_{r,b}$, defined by: $\gamma_{r,b}(t) = rt + b$ for $t > 0$ and 0 otherwise. Having $\gamma_{r,b}$ as an arrival curve allows a source to send b bits at once, but not more than r bits/s over the long run. Parameters b and r are called the burst tolerance (in units of data) and the rate (in units of data per time unit). The Integrated services framework of the Internet (Intserv) uses arrival curves, such as

$$\alpha(t) = \min\{M + pt, rt + b\} = \gamma_{p,M}(t) \wedge \gamma_{r,b}(t)$$

where M is interpreted as the maximum packet size, p as the peak rate, b as the burst tolerance, and r as the sustainable rate Figure 2. Notation $\wedge$ stands for minimum or infimum. In Intserv jargon, the 4-uple (p, M, r, b) is also called a T-SPEC (traffic specification). ATM uses similar curves.

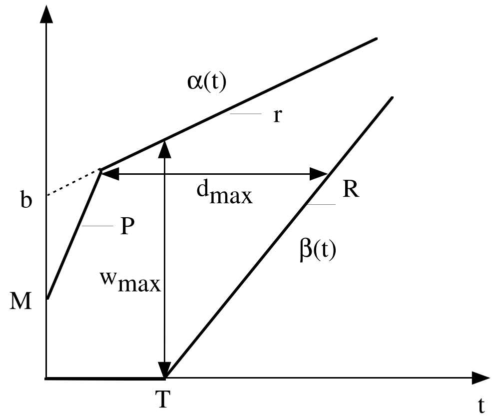

Figure 2: Arrival curve α for ATM VBR and for Intserv flows, rate-latency service curve β and vertical and horizontal devaitions between both curves.

One can always replace an arrival curve α by its sub-additive closure, which is defined as

$$\overline{\alpha} = \inf\{\delta_0, \alpha, \alpha \otimes \alpha, \ldots, \alpha^{(n)}, \ldots\}$$

where $\alpha^{(n)} = \alpha \otimes \ldots \otimes \alpha$ (n times) and δ_0 is the "impulse" function defined by $\delta_0(t) = \infty$ for $t > 0$ and $\delta_0(0) = 0$). One can show indeed that $x \leq x \otimes \alpha$ if and only if $x \leq x \otimes \overline{\alpha}$. If $\alpha(0) = 0$ and α is sub-additive (meaning that for all $s, t \geq 0$, $\alpha(s+t) \leq \alpha(s) + \alpha(t)$), then $\overline{\alpha} = \alpha$. As an example, one can check that $\overline{\gamma}_{r,b} = \gamma_{r,b}$.

Finally, it is possible to compute from measurements of a given flow $x(t)$ its minimal arrival curve, which is $(x \oslash x)(t)$ where $\oslash$ denotes the min-plus deconvolution operator defined by

$$(x \oslash \sigma)(t) = \sup_{u \geq 0}\left\{x(t+u) - \sigma(u)\right\}, \quad (3)$$

for a given function $\sigma \in \mathcal{F}$. Note that if $x, \sigma \in \mathcal{F}$, then $(x \otimes \sigma) \in \mathcal{F}$ but in general $(x \oslash \sigma) \notin \mathcal{F}$ (it belongs to $\mathcal{G}$). One can check however that $(x \oslash x) \in \mathcal{F}$. Let us also mention that the name deconvolution is justified by the fact that for any $x, y, z \in \mathcal{F}$, $x \leq y \otimes z$ if and only if $x \oslash z \leq y$.

3. SERVICE CURVES

We have seen that one first principle in integrated services networks is to put arrival curve constraints on flows. In order to provide reservations, network nodes in return need to offer some guarantees to flows. This is done by packet schedulers. The details of packet scheduling are abstracted using the concept of service curve, which we introduce in this section.

Definition 3 (Service Curve) *Consider a system $\mathcal{S}$ and a flow through $\mathcal{S}$ with input and ouptut function x and y. We say that $\mathcal{S}$ offers to the flow a* service curve *β if and only if for all $t \geq 0$, there exists some $t_0 \geq 0$, with $t_0 \leq t$, such that*

$$y(t) - x(t_0) \geq \beta(t - t_0).$$

Again, we can recast this definition as

$$y(t) \geq \inf_{0 \leq s \leq t} \{\beta(t-s) + x(s)\} = (\beta \otimes x)(t) \quad (4)$$

Let us consider a few examples. A simple one is a GPS (Generalized Processor Sharing) node which, by offering a service curve $\beta(t) = Rt$, guarantees that each flow is served at least at rate R bits/s during a busy period.

A second example is a guaranteed delay node. Here the only information we have about the network node is that the maximum delay for the bits of a given flow x is bounded by some fixed value T, and that the bits of the flow are served in first in, first out order. This is used with a family of schedulers called "earliest deadline first" (EDF), and can be translated as $y(t) \geq x(t-T)$ for all $t \geq T$. Using the "impulse" function δ_T defined by $\delta_T(t) = 0$ if $0 \leq t \leq T$ and $\delta_T(t) = +\infty$ if $t > T$, we have that $(x \otimes \delta_T)(t) = x(t-T)$. We have therefore shown that a guaranteed delay node offers a service curve $\beta = \delta_T$.

As a last example, the IETF assumes that RSVP routers offer a service curve of the form

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

as shown on Figure 2. We call this curve the rate-latency service curve.

Finally, let us mention the following result, which is well-known in traditional system theory, and which is easy to establish in network calculus:

Theorem 1 (Concatenation of Nodes) *Assume a flow traverses systems $\mathcal{S}_1$ and $\mathcal{S}_2$ in sequence. Assume that $\mathcal{S}_i$ offers a service curve of β_i, $i = 1, 2$ to the flow. Then the concatenation of the two systems offers a service curve of $\beta_1 \otimes \beta_2$ to the flow.*

As an example, consider two nodes offering each a rate-latency service curve β_{R_i,T_i}, $i = 1, 2$, as is commonly assumed with Intserv. A simple computation gives

$$\beta_{R_1,T_1} \otimes \beta_{R_2,T_2} = \beta_{R_1 \wedge R_2, T_1+T_2}. \quad (5)$$

Thus concatenating RSVP routers amounts to adding the latency components and taking the minimum of the rates.

We are now also able to give another interpretation of the rate-latency service curve model. We can compute that $\beta_{R,T} = \delta_T \otimes \gamma_{R,0}$; thus we can view a node offering a rate-latency service curve as the concatenation of a guaranteed delay node, with delay T and a CBR or GPS node with rate R.

4. THREE FUNDAMENTAL BOUNDS

In this section we see the main simple network calculus results. They are all bounds for lossless systems with service guarantees [4]. The proofs are straightforward applications of the definitions of service and arrival curves.

The first theorem says that the backlog is bounded by the vertical deviation between the arrival and service curves:

Theorem 2 (Backlog Bound) *Assume a flow, constrained by arrival curve α, traverses a system that offers a service curve β. The backlog $x(t) - y(t)$ for all t satisfies:*

$$x(t) - y(t) \leq \sup_{s \geq 0} \{\alpha(s) - \beta(s)\}$$

We now use the concept of horizontal deviation, which is a little complex, but is supported by the following intuition. Call $\Delta(s) = \inf\{\tau \geq 0 : \alpha(s) \leq \beta(s+\tau)\}$. From Definition 1, $\Delta(s)$ is the virtual delay for a hypothetical system which would have α as input and β as output, assuming that such a system exists (namely, assuming that ($\alpha \leq \beta$). Let $h(\alpha, \beta)$ be the supremum of all values of $\Delta(s)$. The second theorem gives a bound on delay for the general case.

Theorem 3 (Delay Bound) *Assume a flow, constrained by arrival curve α, traverses a system that offers a service curve of β. The virtual delay $d(t)$ for all t satisfies: $d(t) \leq h(\alpha, \beta)$.*

Theorem 4 (Output Flow) *Assume a flow, constrained by arrival curve α, traverses a system that offers a service curve of β. The output flow is constrained by the arrival curve $\alpha^* = \alpha \oslash \beta$.*

As a first application of the previous results, consider a VBR flow, defined by TSPEC (M, p, r, b) (hence $\alpha(t) = \{M + pt\} \wedge \{rt + b\}$) and served in one node which guarantees a service curve equal to the rate-latency function $\beta(t) = R[t-T]^+$. This example is the standard model used in Intserv

(Figure 2). Let us apply Theorems 2 and 3. Assume that $R \geq r$ namely the reserved rate is as large as the sustainable rate of the flow. The buffer required for the flow is bounded by

$$w_{\max} = b + r \max\left(\frac{b-M}{p-r}, T\right)$$

The maximum delay for the flow is bounded by

$$d_{\max} = \frac{M + \frac{b-M}{p-r}(p-R)^+}{R} + T.$$

We can also apply Theorem 4 and find an arrival curve α^* for the output flow.

As a second application, let us show how these bounds, combined with Theorem 1, allow us to understand a phenomenon known in the Insterv community as "Pay Bursts Only Once". Consider the concatenation of two nodes offering each a rate-latency service curve β_{R_i,T_i}, $i = 1, 2$, as is commonly assumed with Intserv. Assume the fresh input is constrained by $\gamma_{r,b}$. Assume that $r < R_1$ and $r < R_2$. We are interested in the delay bound, which we know is a worst case. Let us compare the results obtained by applying Theorem 3 (i) to the network service curve (5), resulting in a delay bound D_0; (ii) iteratively to every node, resulting in two individual bounds D_1 and D_2.

(i) The delay bound D_0 can be computed by application of Theorem 3:

$$D_0 = \frac{b}{R_1 \wedge R_2} + T_1 + T_2.$$

(ii) Now apply the second method. A bound on the delay at node 1 is (Theorem 3): $D_1 = b/R_1 + T_1$. The output of the first node is constrained by by $\alpha^*(t) = b + rt + rT_1$, because of Theorem 4. A bound on the delay at the second buffer is therefore $D_2 = (b + rT_1)/R_2 + T_2$. Consequently,

$$D_1 + D_2 = \frac{b}{R_1} + \frac{b + rT_1}{R_2} + T_1 + T_2$$

It is easy to see that $D_0 < D_1 + D_2$. In other words, the bounds obtained by considering the global service curve are better than the bounds obtained by considering every buffer in isolation.

5. GREEDY SHAPERS

We call *policer* with curve σ a device that counts the bits arriving on an input flow and decides which bits conform with an arrival curve of σ. We call *shaper*, with shaping curve σ, a bit processing device that forces its output to have σ as arrival curve. We call *greedy shaper* a shaper which delays the input bits in a buffer, whenever sending a bit would violate the constraint σ, but outputs them as soon as possible.

With ATM and sometimes with Intserv, traffic sent over one connection, or flow, is policed at the network boundary. Policing is performed in order to guarantee that users do not send more than specified by the contract of the connection. Traffic in excess is either discarded, or marked with a low priority for loss in the case of ATM, or passed as best effort traffic in the case of Intserv. In the latter case, with IPv4, there is no marking mechanism, so it is necessary for each router along the path of the flow to perform the policing function again.

Policing devices inside the network are normally buffered, they are thus shapers. Shaping is also often needed because the output of a buffer normally does not conform any more with the traffic contract specified at the input.

The main result on greedy shapers is the following.

Theorem 5 (Input/output characterization of greedy shapers) *Consider a greedy shaper with shaping curve σ. Assume that the shaper buffer is empty at time 0, and that it is is large enough so that there is no data loss. For an input flow x, the output y is given by*

$$y = \bar{\sigma} \otimes x \tag{6}$$

where $\bar{\sigma}$ is the sub-additive closure of σ.

A simple proof of this theorem will be given in [1]. Remember that if σ is sub-additive and $\sigma(0) = 0$, $\bar{\sigma} = \sigma$. An immediate consequence of this theorem is that a greedy shaper offers to the incoming flow a service curve equal to σ. The input-output characterization of greedy shapers $y = \sigma \otimes x$ is however much stronger than the service curve property.

6. REFERENCES

[1] J. Y. Le Boudec, P. Thiran, S. Giordano, 'A short tutorial on Network Calculus II: Min-plus system theory applied to Communication Networks', *Proc. ISCAS'2000*, Geneva, May 2000.

[2] J.-Y. Le Boudec, 'Application of Network Calculus To Guaranteed Service Networks,' *IEEE Trans. Information Theory*, vol 44(3), May 1998.

[3] C.S. Chang. 'A filtering theory for deterministic traffic regulation', in *Proceedings Infocom'97*, Kobe, Japan, April 1997.

[4] R. Agrawal, R. L. Cruz, C. Okino and R. Rajan, 'Performance Bounds for Flow Control Protocols', *IEEE Trans. on Networking*, vol 7(3), pp 310–323, June 99.

[5] R. L. Cruz. 'Quality of service guarantees in virtual circuit switched networks', *IEEE Journal on Selected Areas in Communication*, pp. 1048–1056, August 1995.