Network Calculus Made Easy

Jean-Yves Le Boudec
Laboratoire de Réseaux de Communication
Ecole Polytechnique Fédérale de Lausanne (EPFL)
http://lrcwww.epfl.ch

Technical Report EPFL-DI 96/218

December 14, 1996

Abstract

We define general concepts that extend, and simplify, the network calculus developed by Cruz [1, 2, 3] for guaranteed quality of service networks. We introduce a definition of service curve, the extended service curve, which is the same for isolated queues and for networks of queues, and makes it possible to model nodes that otherwise do not fit in the calculus of Cruz. We introduce two key technical tools, the $\oplus$ and $\ominus$ operators, on arrival and service curves. We show how their systematic use simplifies the derivation of many fundamental results. We obtain a simple characterization of the output flow from a shaper. We show that, if the shaping and the arrival curves are concave (which is the case in practice), then the output of a shaper is still constrained by the same arrival curve as the input, in addition to being constrained by the shaper curve. Lastly, we define a general form of deterministic effective bandwidth that is particularly simple and efficient.

Keywords Guaranteed Quality of Service; ATM; Effective Bandwidth; Arrival Curves; Service Curves; Queueing Systems; Network Calculus; Max-Plus Algebra

Contents

1 Introduction

The support of guaranteed quality of service, as defined for example by the IETF in [16] requires the computation of the variable part of delay and other bounds for flows served in sequences of buffers. Fundamental work has been pioneered by Cruz in [1, 2, 3], where general bounds based on the concepts of arrival and service curves are derived. Other fundamental work for specific or general scheduling policies is described in [10, 11, 8, 6]. In this paper we define a few concepts and some notation (the $\oplus$ and $\ominus$ operators) that generalize, and simplify, the work mentioned above. This enables us to explain known results with simpler, intuitive methods, generalize the results to more general settings, and discover new results.

We start with an overview of arrival and service curve concepts, as they are defined by Cruz in [1, 2, 3] (Section 2).

The concept of service curve by Cruz is simple and appealing for a single queue; for a network of queues, the definition is a little puzzling as it explicitly privileges one buffer. In particular, a network service curve is not a queue service curve in the sense of Cruz. We propose an alternative definition, which we call *extended service curve*; it is obtained from Cruz's definition by dropping the condition that requires a queue backlog to be zero. The new definition applies to a general input-output system, be it a queuing system or not. The definition is the same for individual queues or network of queues; it extends the applicability of network calculus to nodes that do not originally fit within Cruz's framework. The definition is given in Section 3; we show that the properties obtained by Cruz continue to hold for extended service curves. Indeed, the proofs are simpler than for the original calculus of Cruz. Our results are based on the use of the

$\oplus$ and $\ominus$ operators, two convenient and simple tools to express and discover network calculus results.

The very simple properties of the operators, listed in Section 4, also enables us to derive a number of simple calculus rules for service and arrival curves. We also point to the importance of convexity, concavity, and sub-additiveness for such curves.

We show in Section 5 how a general family of schedulers [16, 6, 14] can be modelled with extended service curves. In general, such schedulers do not have a service curve in the sense of Cruz.

Then we apply in Section 6 our concepts to the notion of a general shaper. A general shaper, with shaping curve σ, is a system that forces a flow to have an output constrained by σ, at the expense of possibly delaying bits in a buffer. The fundamental result we find is that a general shaper offers an extended service curve of σ, provided that σ is sub-additive. We point out that σ is sub-additive as soon as it is concave. This result is the starting point for other new results. In particular, we define an accurate characterization for the output of a buffered shaper. A specific example of shaper is the buffered leaky bucket, which corresponds to the case where the shaping curve is affine, namley of the form $\sigma(t) = rt + b$. A strong result by Cruz is that, if the input flow is also constrained by some affine curves, a buffered leaky bucket does not remove the initial arrival constraints on the flow. Using our formalism, we are able to extend this result very simply to any arrival curve and any shaping curve, provided that both are concave. In spite of being much more general, our proof is also considerably simpler (consider for example the simplicity of the proofs for Theorems 7 and 8).

Lastly, in Section 7, we introduce the general concept of deterministic effective bandwidth, which was introduced in a narrower context in [17] on pages 270–273. We give a simple, general definition, and show that it is a convex function of the arrival curve. This enables us to determine that call acceptance regions based on deterministic delay constraints are convex. This is in contrast to call acceptance regions based on statistical multiplexing with large deviation asymptotics, in which case it is the complement in the positive orthant which is convex [5]. Then we apply the results of Section 6 to the tunneling of several flows into one single, aggregate flow defined by a peak rate, a sustainable rate and a burst tolerance. We find that, if a deterministic delay constraint is used, then there is always an optimal peak rate, which is the effective bandwidth of the arrival traffic.

As a convenience to the reader, we gather here some convention and notation used throughout the paper.

- Greek, lower case letters other than ϵ denote functions defined on $\mathbb{R}^+$, with values in $[0, +\infty]$.
- For any real number x, x^+ denotes $\max(x, 0)$.
- For any R, λ_R represents the function defined by $\lambda_R(t) = Rt$.
- In general, we denote arrival curves by α, service curves by β, and shaping curves by σ. An "arrival"curve that constrains the output of a system is noted α^*.
- $R(t)$ is the arrival function to a system; $R^*(t)$ the departure function.

2 Background on Arrival and Service Curves According to Cruz

We first recall a few definitions. Consider a system $\mathcal{S}$, which we view as a blackbox; $\mathcal{S}$ takes data in and outputs data after a variable delay. Assume every observation starts at time 0; we call $R(t)$ the arrival function, namely the number of bits seen on the input flow in time interval $[0, t]$. We use a continuous time model, so t as well as $R(t)$ are real numbers. We also call $R^*(t)$ the departure function (namely, the arrival function at the output of system $\mathcal{S}$). The *backlog* at time t is $R^*(t) - R(t)$; it is the amount of bits that are held inside the system, assuming we can observe input and output simultaneously. Similarly, the *virtual delay* at time t is

$$d(t) = \inf \{T : T \geq 0 \text{ and } R(t) \leq R^*(t+T)\} \tag{1}$$

It 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 the departure function is continuous (no batch departure) then $R^*(t + d(t)) = R(t)$.

The purpose of network calculus is to find computational rules for bounding virtual delays and backlog for arbitrary systems that represent networks. For that purpose, Cruz introduces the concepts of arrival and queue service curves, which we recall here.

- Given a wide-sense increasing function α, we say that a flow is constrained by α if and only if for all $s \leq t$: $R(t) - R(s) \leq \alpha(t-s)$. Function α is called an *arrival curve* for flow r.

  A flow controlled by a leaky bucket has an arrival curve of the form $\alpha(t) = b + rt$. Similarly, an ATM flow, constrained by the GCRA algorithm with parameters (T, τ) [9] has an arrival curve $\alpha(t) = B + Pt$, with P in b/s and B in bits given by: $B = \tau P + \delta$, $P = \frac{\delta}{T}$. In the formulas, δ is the cell size in bits. A flow conforming to the draft IETF specification for integrated service [16], with maximum packet size M, peak rate p, sustainable rate r and burst tolerance b, has an arrival curve defined by $\alpha(t) = \min(M + pt, b + rt)$.

- Let β be a wide-sense increasing, non-negative function. Cruz says that the buffer offers to a flow a *queue service curve* β if and only if for all time t there exists some $t_0 \leq t$ such that the backlog for the flow in the buffer at time t_0 is zero and $R^*(t) - R^*(t_0) \geq \beta(t - t_0)$.

  In other words, the flow receives a rate of service at least equal to $\beta(T)$ during all intervals $[0, T]$ included in the busy period starting at time 0. Here, a busy period is defined by the conditions $x(s) > 0$ or $r(s) > 0$, where $x(s)$ is the backlog for the flow at time s. We must have $\beta(0) = 0$ for β to be a service curve.

A first result by Cruz is as follows. Consider a data flow constrained by the arrival curve α, served in a buffer with a queue service curve of β. The virtual delay $d(t)$ at any time t satisfies the following inequality:

$$d(t) \leq \sup_{s \geq 0} (\inf \{T : T \geq 0 \text{ and } \alpha(t) \leq \beta(s+T)\}) \tag{2}$$

Moreover, the backlog $x(t)$ at any time t satisfies the following inequality:

$$x(t) \leq \sup_{s \geq 0} (\alpha(s) - \beta(s)) \tag{3}$$

Consider further a flow, with arrival curve, traversing a sequence of buffers $1, \ldots, i, \ldots, I$, with constant propagation delay between buffers, and with each offering a service curve β_i. An important result by Cruz is that the maximum end-to-end delay variation is bounded as in Equation (2), with

$$\beta(t) = \inf_{t_1 + \ldots + t_I = t} \{\beta_1(t_1) + \ldots + \beta_I(t_I)\} \tag{4}$$

Cruz defines a network service curve by saying that a series of buffers offers to a flow a *network service curve* of β if for all time t there exists some $t_0 \leq t$ such that the backlog for the flow in the *first* buffer at time t_0 is zero and $R^*(t) - R(t_0) \geq \beta(t - t_0)$. Function β in the equation above is shown by Cruz to be a network service curve. Note that the network service curve is not a queue service curve in the sense defined above, as we cannot generally find a time $t_0 > 0$ at which the backlog for the flow in the network is zero. It is this puzzling subtlety that leads us to an alternative definition, which turns out to be both simpler and more powerful.

Another result by Cruz is that the output of such a network is constrained by arrival curve α^*, given by

$$\alpha^*(t) = \sup_{v \geq 0} (\alpha(t + v) - \beta(v)) \tag{5}$$

Equations (2) and (4) are a strong result. Indeed, consider again a sequence of buffers. On one hand we can compute a bound D on the total delay by using Equations (2) and (4). On the other hand, we can also compute a bound D_i for the variable delay experienced at buffer i, using Equation (5) iteratively to characterize the input flow at buffer i. Then, in general, $\sum_i B_i > B$. Network calculus captures this global network effect.

In the rest of this paper, we generalize the nodes to which network calculus applies. In particular, we show that the above results remain true even if we add variable delays between buffers that cannot be modelled as queueing delays.

3 Extended Service Curves

In this section we define our extension to the network calculus of Cruz.

3.1 Definition of Extended Service Curve

We propose to replace the definition of queue or network service curve by Cruz by the following definition. We use the same notation as in Section 2

Definition 1 (Extended Service Curve) *Consider a flow with arrival function R, input to a bit processing system $\mathcal{S}$. Call R^* the output function. We say that $\mathcal{S}$ offers to the flow an* extended 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 $R^*(t) - R(t_0) \geq \beta(t - t_0)$*

The condition is similar to the queue service curve, or network service curve definition by Cruz, with the difference that it does not imply any condition on a buffer being empty. We are able to show and even generalize the results by Cruz without requiring such assumptions, thanks to the simple properties of the $\oplus$ and $\ominus$ operators. The resulting calculus not only extends to nodes for which a service curve in the sense of Cruz does not exist, but is also surprisingly simpler. We derive the main results for service curves in the rest of this Section and in the following one.

Of course, queue service curves and network service curves are extended service curves, as recalled in Section 2.

We now introduce the key example of node (variable delay node) that offers an extended service curve and cannot easily be modelled by the calculus of Cruz. We first define the burst delay function δ_T by

Definition 2 (Burst Delay function) $\delta_T(t) = 0$ *if* $0 \leq t \leq T$ *and* $\delta_T(t) = +\infty$ *if* $t > T$

Proposition 1 (Variable Delay Node) *Consider a bit processing system for which we only know that the maximum delay for any bit of the flow is bounded by a constant T. This system offers to the flow an extended service curve of δ_T.*

The proof is straightforward and is left to the reader. Figure 1 illustrates a node that offers δ_T as a queue-service-curve, in the sense of Cruz: bits in the buffer are held for a time $\leq T$, until the buffer is emptied instantaneously. We see on this example that a variable delay node does not in general have δ_T as a queue-service-curve, but that it does have δ_T as an extended service curve. This latter point is essential, because, as we show later, formulas for delay and output bounds also apply to extended service curves.

3.2 Basic Results on Bounds

In this Section we generalize Cruz's results to extended service curves.

Theorem 1 (Backlog Bound) *Assume a flow, constrained by arrival curve α, traverses a system that offers an extended service curve β. The backlog $R^*(t) - R(t)$ for all t satisfies:*

$$R^*(t) - R(t) \leq \sup_{s \geq 0}\{\alpha(s) - \beta(s)\}$$

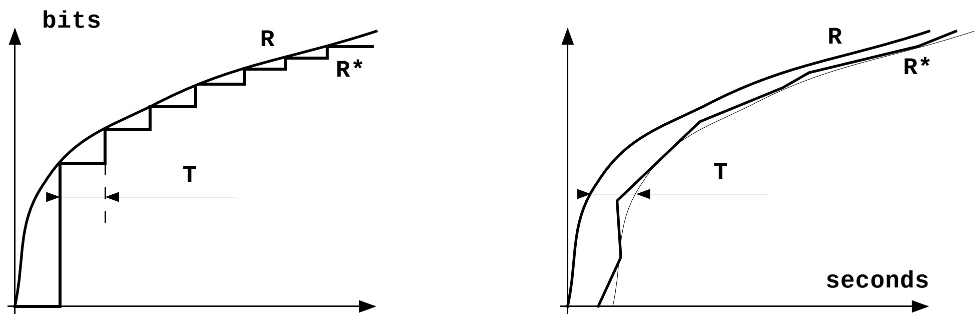

Figure 1: Arrival and Departure Functions for a node offering δ_T as a queue-service-curve (left) or as an extended service curve (right). On the right is also shown the arrival curve shifted by T to the right. The departure curve is limited by this curve.

Proof: For all t there exists some s such that

$$R^*(t) - R(t-s) \geq \beta(s)$$

Now

$$R(t) - R(t-s) \leq \alpha(s)$$

By subtraction, it comes that $R^*(t) - R(t) \leq \alpha(s) - \beta(s)$ □

If the system is a single buffer, then the backlog can be interpreted as the instantaneous 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 theorem says that the backlog is bounded by the vertical deviation between the arrival and service curves.

The next result is best described using the following notation.

Definition 3 (Minus Operator) *For two functions γ_1 and γ_2, define $\gamma_1 \ominus \gamma_2$ by*

$$\gamma_1 \ominus \gamma_2(t) = \sup_{u \geq 0} \{\gamma_1(t+u) - \gamma_2(u)\} \tag{6}$$

We can thus re-write the result in Equation (5) by saying that the *output* flow has an arrival curve $\alpha^* = \alpha \ominus \beta$. This result is also true for extended service curves:

Theorem 2 (Output Characterization) *Assume a flow, constrained by arrival curve α, traverses a system that offers an extended service curve of β. The output flow is constrained by the arrival curve $\alpha^* = \alpha \ominus \beta$.*

Proof: With the same notation as above, consider $R^*(t) - R^*(t-s)$, for $0 \leq t-s \leq t$. By definition of the extended service curve, item 1, applied at time $t-s$, there exists some $u \geq 0$ such that $0 \leq t-s-u$ and

$$R^*(t-s) - R(t-s-u) \geq \beta(u)$$

Thus

$$R^*(t) - R^*(t-s) \le R^*(t) - \beta(u) - R(t-s-u)$$

Now $R^*(t) \le R(t)$, therefore

$$R^*(t) - R^*(t-s) \le R(t) - R(t-s-u) - \beta(u) \le \alpha(s+u) - \beta(u)$$

and the latter term is bounded by $(\alpha \ominus \beta)(s)$ by definition of the $\ominus$ operator. $\square$

It is convenient to introduce the following notation.

Definition 4 (Horizontal Deviation h) *For two wide-sense increasing functions α and β, call $h(\alpha, \beta)$ the horizontal deviation between the two curves, namely*

$$h(\alpha, \beta) = \sup_{s \ge 0} \left(\inf \{ T : T \ge 0 \text{ and } \alpha(s) \le \beta(s+T) \} \right) \tag{7}$$

The definition is a little complex, but is supported by the following intuition. If α and β are continuous, strictly increasing,then for all t there exists at most one number $d(t)$ such that $\alpha(t) = \beta(t + d(t))$. If there is no such number then we let $d(t) = +\infty$. In this case, $h(\alpha, \beta) = \sup_t d(t)$.

In other words, $h(\alpha, \beta)$ is nothing else than the formula used for computing the maximum delay in Equation (2), which can now be re-written as: $d(t) \le h(\alpha, \beta)$, with $d(t)$ the virtual delay at time t, α the arrival curve, and β the queue or network service curve in the sense of Cruz. This result still remains if we consider extended service curves. Indeed:

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

Proof: For some fixed $t \ge 0$, call d the virtual delay at time t, written as $d(t)$ in Equation (1). We can assume $d > 0$ otherwise the proof is trivial. By definition of d, for all $\epsilon > 0$, with $\epsilon < d$ we have

$$R^*(t + d - \epsilon) < R(t)$$

By the extended service curve property at time $t + d - \epsilon$, there exists some $t_0 \ge 0$ such that

$$R^*(t + d - \epsilon) - R(t_0) \ge \beta(t + d - \epsilon - t_0)$$

This implies that $t_0 \le t$. Now

$$R(t) - R(t_0) \le \alpha(t - t_0)$$

Combining the three inequations gives

$$\beta(t + d - \epsilon - t_0) < \alpha(t - t_0)$$

This proves that $d - \epsilon \le h(\alpha, \beta)$ for all $0 < \epsilon \le d$. $\square$

Finally, the last basic result requires the following definition.

Definition 5 (Min-Plus convolution) *For two functions* γ_1 *and* γ_2*, define* $\gamma_1 \oplus \gamma_2$ *by*

$$\gamma_1 \oplus \gamma_2(t) = \inf_{u \text{ such that } 0 \le u \le t} \{\gamma_1(u) + \gamma_2(t-u)\} \tag{8}$$

We see that the network-service curve mentioned in Section 2 is the Min-Plus convolution of the service curves offered by the buffers. In all practical cases that we will meet, the inf is a min, which explains the name we choose for the $\oplus$ operator. The concatenation result by Cruz is also true for extended service curves:

Theorem 4 (Concatenation of Nodes) *Assume a flow traverses systems* $\mathcal{S}_1$ *and* $\mathcal{S}_2$ *in sequence. Assume that* $\mathcal{S}_i$ *offers an extended service curve of* β_i*,* $i = 1, 2$ *to the flow. Then the concatenation of the two systems offers an extended service curve of* $\beta_1 \oplus \beta_2$ *to the flow.*

Proof: Call R the arrival function, R_1 the output function from $\mathcal{S}_1$ (thus R_1 is the arrival function to system $\mathcal{S}_2$) and R^* the final output function. Consider some $t \ge 0$. There exists some $u \ge 0$ with $0 \le t - u$ and

$$R^*(t) - R_1(t-u) \ge \beta_2(u)$$

Similarly, there exists some $v \ge 0$ with $0 \le t - u - v$ and

$$R_1(t-u) - R(t-u-v) \ge \beta_1(v)$$

Thus, by summation:

$$R^*(t) - R(t-u-v) \ge \beta_2(u) + \beta_1(v) \ge (\beta_1 \oplus \beta_2)(u+v)$$

□

We have thus shown that the basic results recalled in Section 2 are valid also for extended service curves. The bounding results mentioned in Section 2 are thus still valid if we add random, bounded delays between the queues. Such delays are accounted for by introducing δ_T functions in the service curves. As a consequence of the commutativity of $\oplus$, such delays can be inserted in any order along a sequence of buffers, without altering the delay bounds.

In the rest of the paper we derive new or more general results, using the properties of the $\oplus$ and $\ominus$ operators.

4 Network Calculus

In this Section we review the basic or more advanced properties of the $\ominus$ and $\oplus$ operators, which we claim to be the basics of network calculus. We start by illustrating how simple results can be obtained with little effort.

4.1 Network Calculus in Practice

The following holds for the Min-Plus convolution.

Rule 1: $\oplus$ is associative and commutative, so for example $(\gamma_1 \oplus \gamma_2) \oplus \gamma_3 = \gamma_1 \oplus (\gamma_2 \oplus \gamma_3) = \gamma_1 \oplus \gamma_2 \oplus \gamma_3$.

Rule 2: If If $\gamma_1(0) = \gamma_2(0) = 0$ then $\gamma_1 \oplus \gamma_2 \leq \min(\gamma_1, \gamma_2)$. If γ_1 and γ_2 are concave (namely if $-\gamma_1$ and $-\gamma_2$ are convex) then $\gamma_1 \oplus \gamma_2 = \min(\gamma_1, \gamma_2)$.

Rule 3: If γ_1 and γ_2 are convex and piecewise linear, then $\gamma_1 \oplus \gamma_2$ is obtained by putting end-to-end the different linear pieces of the individual service curves, sorted by increasing slopes [4].

The proof of the above points is easy. For example, for rule 2, we use the fact that $\gamma_1 + \gamma_2$ is concave, and thus has a minimum on $[0, t]$ at either 0 or t.

The IETF uses a generic service curve model; it assumes that every node offers a service of the form

$$\beta(t) = R(t - T)^+ \tag{9}$$

for some delay T and rate R. We call such a service curve the "IETF service curve model". We show in Section 5 that there are a number of scheduling policies that offer such extended service curves. In [16], it is further assumed that the delay parameter T depends on the rate R according to $T = \frac{C}{R} + D$ for some constants C and D.

The extended service curve in Equation 9 is convex, therefore we can apply rule 3. Thus, the concatenation of nodes that offer an extended service curve of $\beta_i(t) = R_i(t - T_i)^+$ offers an extended service curve β of the same form, given by $\beta(t) = R(t - T)^+$, with $T = \sum_i T_i$ and $R = \min_i R_i$. As an application, the maximum delay for a flow conforming to the IETF specification for integrated service, with maximum packet size M, peak rate p, sustainable rate r and burst tolerance b, is obtained by computing $h(\alpha, \beta)$, with $\alpha(t) = \min(M + pt, b + rt)$. This gives [16]:

$$d(t) \leq \frac{b - M}{R} \left(\frac{p - R}{p - r} \right)^+ + \frac{M}{R} + T$$

Consider now another example: a series of nodes guarantee a service defined by $\beta_i(t) = (r_i t + b_i)1_{\{t>0\}}$ (thus $\beta_i(t) = 0$ if $t = 0$, as is required for a service curve). By application of rule 2, the concatenation of these nodes offers the service curve $\beta(t) = \min_i(r_i t + b_i)1_{\{t>0\}}$. Such nodes are leaky bucket regulators with unlimited output peak rates (Section 6).

If we want to compute the min-plus convolution for a mix of functions which is neither all-concave nor all-convex, then it may be useful to use some decompositions in order to apply rules 2 and 3.

As an illustration of how this calculus can simplify some computations, consider the service curve β at the concatenation of two nodes offering the following service curves: at node 1,

$\beta_1(t) = R(t-T)^+$; at node 2, $\beta_2(t) = (rt+b)1_{\{t>0\}}$. Note that $(\alpha \oplus \delta_T)(t) = \alpha(t-T)1_{\{t>T\}}$ for any wide-sense increasing function α. We can now decompose β_1 as:

$$\beta_1 = \delta_T \oplus \lambda_R$$

and thus

$$\beta = (\delta_T \oplus \lambda_R) \oplus \beta_2 = \delta_T \oplus (\lambda_R \oplus \beta_2)$$

Now by rule 2

$$\lambda_R \oplus \beta_2(t) = \min(Rt, rt+b)$$

and thus

$$\beta(t) = \min\left[R(t-T), r(t-T)+b\right]1_{\{t>T\}}$$

In practice, we use network calculus to compute bounds on delay variation and backlogs. Constant propagation delays between systems can be ignored for the computation of delay bounds, as they simply transform an input function into the functions translated in time by a fixed amount. Variable delays can be modelled simply by the addition of δ_T functions to the service curves. For the computation of the backlog inside a system $\mathcal{S}$, constant delays inside $\mathcal{S}$ cannot be ignored and should be modelled with a δ_T extended service curve.

4.2 Arrival Curves, Sub-Additive and Concave Functions

We now obtain general results about arrival curves. These will be useful in Section 6.

In general, we do not expect an arrival curve to be convex. Indeed, it is easy to see that if α_1 and α_2 are two arrival curves for the same flow, then so is $\alpha_1 \oplus \alpha_2$. Assume now that an arrival curve α is convex and piecewise linear; then $\alpha \oplus \alpha$ is also an arrival curve, and by iterating, we see that an arrival curve for the same flow is given by $\bar{\alpha}(t) = r_0 t$, where r_0 is the slope of α at $t = 0 + \epsilon$, for ϵ small enough. This example illustrates that any arrival curve is not necessarily very meaningful.

In general, it makes sense for an arrival curve to be *sub-additive*:

Definition 6 *We say that function α is sub-additive when $\alpha(s+t) \leq \alpha(s) + \alpha(t)$ for all $s, t \geq 0$.*

Note that for functions α such that $\alpha(0) = 0$, being sub-additive is equivalent to the condition $\alpha \oplus \alpha = \alpha$. We have the following result.

Proposition 2 (Transformation of Arrival Curve into Sub-Additive Arrival Curve) *Consider a flow constrained by an arrival curve α. Then there exists an arrival curve $\bar{\alpha} \leq \alpha$ that also constrains the flow and is sub-additive.*

Proof: The proof consists in constructing the arrival curve $\bar{\alpha}$. It follows from the discussion above that $\alpha \oplus \ldots \oplus \alpha = \alpha^{\oplus n}$ is also a service curve, and so is

$$\bar{\alpha} = \min_{n \geq 1} \alpha^{\oplus n} \tag{10}$$

(We used the convention $\alpha^{\oplus 1} = \alpha$) Since $\alpha^{\oplus n} \leq \alpha^{\oplus i}$ for $i \leq n$, note that $\alpha^{\oplus n} = \min_{1 \leq i \leq n} \alpha^{\oplus i}$ and thus $\bar{\alpha}(t) = \lim_{n \to +\infty} \alpha^{\oplus n}(t)$. We now show that $\bar{\alpha}$ is sub-additive. For all $s, t \geq 0$, and for all $\epsilon > 0$ there exist some decompositions $s_1 + \ldots + s_i = s$ and $t_1 + \ldots + t_j = t$ such that

$$\bar{\alpha}(s) + \bar{\alpha}(t) \geq \alpha(s_1) + \ldots + \alpha(s_i) + \alpha(t_1) + \ldots + \alpha(t_j) - \epsilon$$

Thus, by definition of $\bar{\alpha}$:

$$\bar{\alpha}(s) + \bar{\alpha}(t) \geq \bar{\alpha}(s_1 + \ldots + s_i + t_1 + \ldots + t_j) - \epsilon = \bar{\alpha}(s + t) - \epsilon$$

and this is true for arbitrary small values of ϵ. □

Note that we know explicitly from Equation 10 how to transform the arrival curve α into a sub-additive arrival curve $\bar{\alpha}$. Note also that if α is sub-additive, then $\bar{\alpha} = \alpha$.

We should thus restrict our choice of arrival curves to sub-additive functions. A simple inspection of the arrival curve may be sufficient. Indeed:

Proposition 3 *Any concave function ϕ defined on $\mathbb{R}^+$ such that $\phi(0) = 0$ is sub-additive.*

The proof is given in appendix.

Note that any affine function is concave, and so is the minimum of any number of concave functions. Thus the arrival curves we have seen in Section 2 are all concave

Sub-additivity is a condition for the theorems in Section 6 to hold. In practice, we will consider concave arrival curves. Note however that there are arrival curves that are sub-additive and not concave (build an example by considering an on-off pattern).

Lastly, assume we wish to determine an arrival curve from an observed arrival function $R(t)_{t \geq 0}$. The solution is simple:

Proposition 4 (Minimum Arrival Curve) *Consider a flow given by its arrival function $R(t)_{t \geq 0}$. Then*

- *the function $R \ominus R$ is an arrival curve for the flow*
- *$R \ominus R$ is sub-additive*
- *for any arrival curve α that constrains the flow, we have: $(R \ominus R) \leq \alpha$*

The proof is straightforward and left to the reader. Remember that the arrival curve $R \ominus R$ defined in the proposition is given by

$$(R \ominus R)(t) = \sup_{v \geq 0} R(t+v) - R(v)$$

The proposition says that any flow has one *minimum* arrival curve. It is typically the arrival curve we could compute based on measurements of the arrival flow.

4.3 More about the $\ominus$ Operator

The following rules apply to any general functions. We use the notation $\gamma_1 \leq \gamma_2$ with the meaning : for all $t \geq 0$, $\gamma_1(t) \leq \gamma_2(t)$.

Rule 4 If $\alpha_1 \leq \alpha_2$, then for all β : $\alpha_1 \ominus \beta \leq \alpha_2 \ominus \beta$. If $\beta_1 \leq \beta_2$, then for all α : $\alpha \ominus \beta_1 \geq \alpha \ominus \beta_2$

Rule 5 Function α is sub-additive if and only if $\alpha \ominus \alpha \leq \alpha$

Rule 6 $(\gamma_1 \ominus \gamma_2) \ominus \gamma_3 = \gamma_1 \ominus (\gamma_2 \oplus \gamma_3)$ for all functions γ_1, γ_2 and γ_3

Proof: We give only the proof for rule 6. The other proofs are simple and left to the reader. For a fixed value of t, let $A = [(\gamma_1 \ominus \gamma_2) \ominus \gamma_3](t)$ and $B = [\gamma_1 \ominus (\gamma_2 \oplus \gamma_3)](t)$. We show first that $A \geq B$. For all $s \geq 0$ and all $0 \leq u \leq s$ we have

$$A \geq \gamma(t+u) - \gamma_3(u)$$

where $\gamma = \gamma_1 \ominus \gamma_2$. Similarly, by definition of $\ominus$:

$$\gamma(t+u) \geq \gamma_1(t+u+s-u) - \gamma_2(s-u)$$

Putting the two formulas together gives

$$\gamma_2(s-u) + \gamma_3(u) \leq \gamma_1(t+s) - A$$

Since this is true for all $0 \leq u \leq s$, we have

$$(\gamma_2 \oplus \gamma_3)(s) \leq \gamma_1(t+s) - A$$

The above is true for all s, which shows that $B \leq A$.

Conversely, for all $\epsilon > 0$ there is some v such that

$$A \leq \gamma(t+v) - \gamma_3(v) + \epsilon$$

There is also some u such that

$$\gamma(t+v) \leq \gamma_1(t+v+u) - \gamma_2(u) + \epsilon$$

Thus

$$A \le \gamma_1(t+v+u) - \gamma_2(u) - \gamma_3(v) + 2\epsilon \le \gamma_1(t+v+u) - (\gamma_2 \oplus \gamma_3)(u+v) + 2\epsilon$$

Thus $A \le B + 2\epsilon$ for all $\epsilon > 0$, thus $A \le B$. $\square$

Consider Theorem 4 again and assume the input flow to system $\mathcal{S}_1$ is constrained by α. From Theorem 2 we can conclude that the output of system $\mathcal{S}_2$ is constrained by $\alpha \ominus (\beta_1 \oplus \beta_2)$. On the other hand, by recursive application of Theorem 2, it is also constrained by $(\alpha \ominus \beta_1) \ominus \beta_2$. Rule 6 tells us that the two constraints are the same.

As an example of application, consider the output a system with offering the IETF service curve model $\beta = \delta_T \oplus \lambda_R$ as an extended service curve. Assume the input is constrained by α. Then the output is constrained by $\alpha^* = \alpha \ominus (\lambda_R \oplus \delta_T) = (\alpha \ominus \lambda_R) \ominus \delta_T$. Note that

$$(\gamma \ominus \delta_T)(t) = \gamma(t+T)$$

for all γ (shift to the left).

The computation of $\alpha \ominus \lambda_R$ is easy if α is concave. Indeed, in such a case, define t_0 as

$$t_0 = \inf\{t \ge 0 : \alpha'(t) \le R\}$$

where α' is the right-handside derivative, and assume that $t_0 < +\infty$. Then by studying the variations of the function $u \to \alpha(t+u) - Ru$ we find that $(\alpha \ominus \lambda_R)(s) = \alpha(s)$ if $s \ge t_0$, and $(\alpha \ominus \lambda_R)(s) = \alpha(t_0) + (s - t_0)R$ if $s < t_0$.

Putting the pieces all together we see that the output function α^* is obtained from α by

- replacing α on $[0, t_0]$ by the linear function with slope R which has the same value as α for $t = t_0$, keeping the same values as α on $[t_0, +\infty[$,
- and shifting by T to the left.

Figure 2 illustrates the operation. Note that the two operations can be performed in any order since $\oplus$ is commutative.

4.4 Advanced Results on $\oplus$ and $\ominus$ Operators

In this Section we conclude the study of the $\oplus$ and $\ominus$ operators by giving advanced results that are easy to understand, but will still make it possible to derive powerful results in the next Section.

Proposition 5 *Let α be wide-sense increasing and sub-additive, with $\alpha(0) = 0$. Then for any function β: $h(\alpha, \alpha \oplus \beta) = h(\alpha, \beta)$*

We will also use Proposition 5 under the following form.

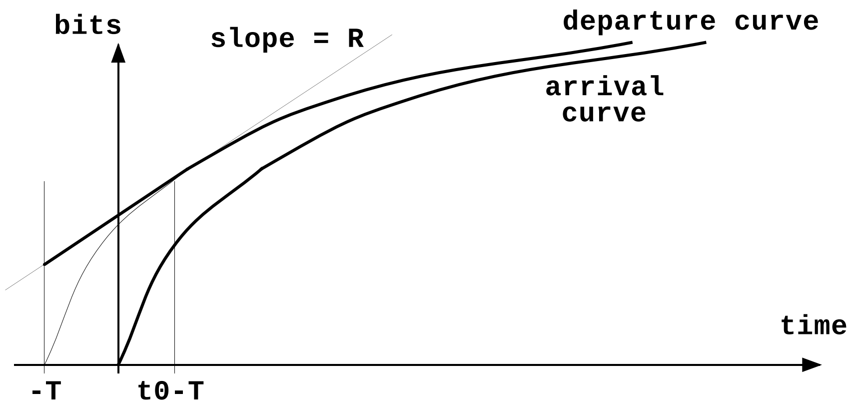

Figure 2: Derivation of output curve for a node offering the IETF service curve model $\beta(t) = R(t-T)^+$

Corollary 1 *Let α and σ be wide-sense increasing and sub-additive, with $\alpha(0) = \sigma(0) = 0$. Assume $\alpha \leq \sigma$. Then for any function β: $h(\alpha, \sigma \oplus \beta) = h(\alpha, \beta)$*

We can interpret this proposition and its corollary as follows. If we let a flow be served through a series of two buffers, with the first buffer offering a service curve at least equal to the arrival curve, then the delay bound computed from Theorem 3 is the same as if the first buffer were omitted. Indeed, the flow experiences no delay in the first buffer. Note that the latter interpretation cannot serve as a proof because the bound in Theorem 3 might be pessimistic in some cases.

Proof: The horizontal deviation $h(\alpha, \beta)$ is defined by $h(\alpha, \beta) = \sup_{t \geq 0} d(t)$, with

$$d(t) = \inf\{T \geq 0 : \alpha(t) \leq \beta(t+T)\}$$

Define similarly

$$d'(t) = \inf\{T \geq 0 : \alpha(t) \leq (\alpha \oplus \beta)(t+T)\}$$

First note that if we increase the function β, then $h(\alpha, \beta)$ may only decrease, in other words:

Rule 7 If $\beta_1 \leq \beta_2$ then $h(\alpha, \beta_1) \geq h(\alpha, \beta_2)$

Now $\alpha \oplus \beta \leq \beta$ since $\alpha(0) = 0$. This proves that $h(\alpha, \alpha \oplus \beta) \geq h(\alpha, \beta)$.
Conversely, the horizontal deviation $h(\alpha, \beta)$ is defined by $h(\alpha, \beta) = \sup_{t \geq 0} d(t)$, with

$$d(t) = \inf\{T \geq 0 : \alpha(t) \leq \beta(t+T)\}$$

Define similarly

$$d'(t) = \inf\{T \geq 0 : \alpha(t) \leq (\alpha \oplus \beta)(t+T)\}$$

For all $\epsilon > 0$, with $\epsilon < t + d'(t)$:

$$(\alpha \oplus \beta)(t + d'(t) - \epsilon) < \alpha(t)$$

Thus, there exists some u such that $0 \leq u \leq t + d'(t) - \epsilon$ and

$$\alpha(u) + \beta(t + d'(t) - u - \epsilon) < \alpha(t) \tag{11}$$

Now $u \leq t$ otherwise $\alpha(u) \geq \alpha(t)$. Let $v = t - u$; thus $v \geq 0$ and by sub-additivity of α,

$$\alpha(t) \leq \alpha(u) + \alpha(v)$$

We can rewrite Equation (11) as

$$\alpha(u) + \beta(v + d'(t) - \epsilon) < \alpha(t)$$

Putting together the last two inequations gives

$$\beta(v + d'(t) - \epsilon) < \alpha(v)$$

This proves that

$$d'(t) - \epsilon \leq d(v)$$

and thus

$$d'(t) - \epsilon \leq h(\alpha, \beta)$$

for all ϵ, thus $d'(t) \leq \Delta$. This is true for all t, thus we have proven that $h(\alpha, \alpha \oplus \beta) \leq h(\alpha, \beta)$.□

The proof of the corollary derives immediately from the theorem and from Rule 7.

We also have:

Proposition 6 *Let γ_1 be sub-additive, with $\gamma_1(0) = 0$. Then for any function γ_2, the following holds:*

$$\textit{if } \gamma_1 \leq \gamma_2 \textit{ then } \gamma_1 \ominus \gamma_2 \leq \gamma_1$$

Proof: Since $\gamma_2 \geq \gamma_1$ we have $\gamma_1 \ominus \gamma_2 \leq \gamma_1 \ominus \gamma_1$ (Rule 4). Now if γ_1 is sub-additive, $\gamma_1 \ominus \gamma_1 \leq \gamma_1$ (Rule 5). □

The proposition can be interpreted as follows. If we let a flow be served through a buffer that offers a service curve larger than the arrival curve, then the output flow is still constrained by the initial arrival curve. Indeed, the flow experiences no delay in the buffer .

5 Modelling Guaranteed Quality of Service Nodes

In Section 4, Equation (9), we have mentioned the common node model used by IETF. In this Section we show that this model can be used for a very large class of schedulers, even though such schedulers do not offer a queue-service-curve in the sense of Cruz. Of course, such nodes do offer an *extended* service curve of the type in Equation (9).

A number of scheduling policies has been proposed in the literature (see for example [10, 6, 7, 13, 14]). Consider the general form of scheduling proposed in [14] under the name of Guaranteed Rate (GR) scheduling. It is shown in [14] that Guaranteed Rate scheduling includes as particular cases: virtual clock scheduling [13], packet by packet generalized processor sharing [10] and self-clocked fair queuing [7].

Following [14], we say that a scheduling policy is of the *guaranteed rate* type, with rate R and delay v for a given flow if it guarantees that packet j of the flow is served at time $GCR(j) + v$, with $GRC(0) = 0$ and

$$GRC(j) = \max\{A(j), GRC(j-1)\} + \frac{l(j)}{R} \tag{12}$$

In the formula, $l(j)$ is the length in bits of packet j and $A(j)$ is the arrival time of packet j.

Theorem 5 (Modelling GR Nodes) *A node with guaranteed rate scheduling policy offers an extended service curve defined by*

$$\beta(t) = R(t - \frac{l}{R} - v)^+$$

In other words, $\beta = \lambda_R \oplus \delta_{\frac{l}{R}+v}$.

Proof: Call $D(i)$ the departure time for the last bit of packet i. If $\sup_i D(i)$ is finite, then if $t > \sup_i D(i)$ then $R(t) = R^*(t)$ since any packet is guaranteed to leave the system after a finite time. The extended service curve property is trivially true in that case. We can thus assume now that there is a packet index j such that $D(j-1) \leq t < D(j)$. Thus

$$R^*(t) = \sum_{i=1}^{j-1} l(i) \tag{13}$$

Define

$$i_0 = \max\{i \text{ such that } 1 \leq i \leq j \text{ and } GRC(i-1) \leq A(i)\}$$

Figure 3 illustrates a case with $j = 2$ and $i_0 = 1$. We first consider the case where $A(i_0) \leq D(j-1)$.

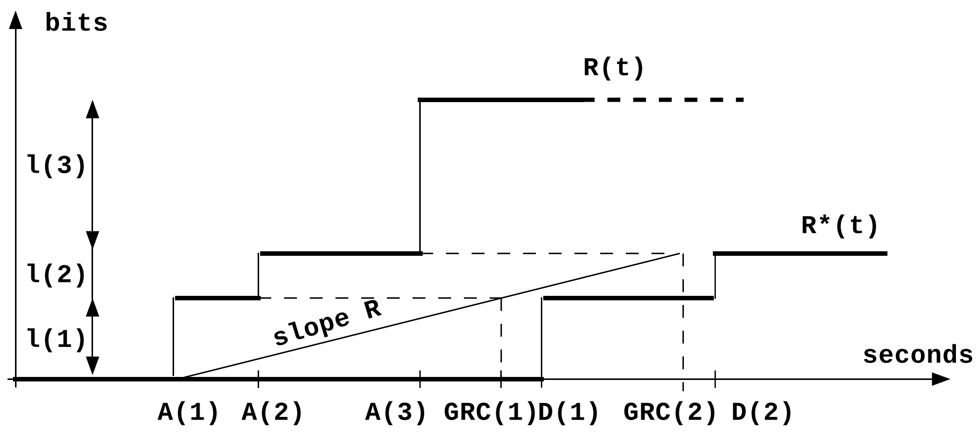

Figure 3: Arrival and Departure functions for GR scheduling

Now $GRC(i_0) = A(i_0) + \frac{l(i_0)}{R}$ and for all i such that $i_0 < i \leq j$ we have $GRC(i) = GRC(i-1) + \frac{l(j)}{R}$, thus

$$GRC(j) = A(i_0) + \frac{1}{R}\sum_{i=i_0}^{j} l(i) \tag{14}$$

Define $t_0 = A(i_0) - \epsilon$, where ϵ is small enough for $t + \epsilon \leq D(j)$ and $A(i_0) - \epsilon > A(i_0 - 1)$. We have:

$$R(t_0) = \sum_{i=1}^{i_0-1} l(i) \tag{15}$$

and

$$t - t_0 \leq D(j) - A(i_0) \leq GRC(j) + v - A(i_0)$$

and from Equation (14), it comes that

$$t - t_0 \leq v + \frac{1}{R}\sum_{i=i_0}^{j} l(i) \leq v + l + \frac{1}{R}\sum_{i=i_0}^{j-1} l(i)$$

Thus, by definition of β,

$$\beta(t - t_0) \leq \sum_{i=i_0}^{j-1} l(i) = R^*(t) - R(t_0)$$

From Equations (13) and (15), the last term in the formula above is precisely $R^*(t) - R(t_0)$ which proves the extended service curve property in the case where $A(i_0) \leq D(j-1)$.

Now if $A(i_0) > D(j-1)$ then necessarily $i_0 = j$. If $t \geq A(i_0)$, then the above reasoning applies and shows the extended service curve property. Otherwise, $D(i_0 - 1) \leq t < A(i_0)$, the queue is empty at t and the extended service curve property is true with $t_0 = t$. □

If the delay parameter v is equal to 0, then we can show that the queue is empty at time t_0, and, in that case, the extended service curve is also a queue service curve in the sense of Cruz. Otherwise, namely if $v > 0$, it is easy to build an example where the queue is not empty at time t_0 and the service curve property in the sense of Cruz is not true.

There are other forms of scheduling that do not belong to the GR type, for example, delay based schedulers [18]. Such schedulers can easily be shown to offer an extended service curve of δ_T for some T. Such schedulers are usually associated with a shaper. In Section 6, we will show that such a combined system offers an extended service curve of $\delta_T \oplus \sigma$, where σ is the shaping curve. In general, these systems do not offer a service curve in the sense of Cruz.

6 Shaping Devices

In this Section we apply the calculus introduced in the previous section to shaping and policing devices. We introduce the concept of policer that generalizes the leaky bucket controllers [17] and re-shapers [16].

Definition 7 *Consider a wide sense increasing function σ. We call* policer *a bit processing device such that, for any arbitrary input flow, we have:*

- *the output function has σ as arrival curve;*
- *if the input flow has σ as arrival curve, then the shaper is transparent to the flow.*

The second condition can be formalized as follows: call R the arrival function for a given, arbitrary flow, and R^* the output function; then $R(t) = R^*(t)$ for all t. We say that σ is a shaping curve. It is easy to verify that leaky bucket controllers are policers with a shaping curve of the form $\sigma(t) = (rt + b)1_{\{t>0\}}$.

A policer may exist in a variety of forms, since the definition does not specify what happens to input flows that are do not have σ as arrival curve. However, in all cases, a policer has the following feature. We consider only systems for which there is a minimum time granularity dt. For an ATM system, this corresponds to the sending of one cell, for other systems, it may be the bit processing time. During a time interval $[t, t + dt]$, the policer counts all bits that arrived since the beginning of the connection, over all possible windows of length $u \geq 0$, and checks that there are not more than $\sigma(u)$ bits. The number of bits dR^* allowed at the output, for a given number of input bits dR is thus

$$dR^* = \min\left\{dR, \min_{u\geq 0}\left[\sigma(t + dt, t - u) - R^*(t) - R^*(t - u)\right]\right\} \tag{16}$$

A *simple* policer declares as non conformant a number of bits equal to $dR^* - dR$ whenever $dR > dR^*$. We call *shaper* a *buffered* policer, namely a friendly policer that puts into a buffer all bits in excess (i.e. $dR^* - dR$ bits whenever $dR > dR^*$). It is outside the scope of this paper to study how to build shapers (see for example [10, 6, 14]). Real shaping systems may delay packets for a time longer than the granularity dt; such systems are modelled by the concatenation of a shaper as described here, and a delay node with service curve δ_T for some T.

The main result of this section is the following.

Theorem 6 (Service Curve offered by a General Shaper) *Consider a shaper with shaping curve σ, namely a device that serves bits at the maximum instant rate such that the output has σ as arrival curve, and otherwise keeps the bits in a buffer. Assume that σ is sub-additive and $\sigma(0) = 0$. This system offers to the flow a service curve equal to σ.*

Proof: In order to simplify the notation, we take the time granularity dt as time unit in this proof. Define the following

1. $x(t)$ is the buffer occupancy at the end of time slot t; by convention, we define $x(0) = 0$.
2. $R^*]u, t]$ is the number of bits transmitted during time slots $u + 1, \ldots, t$.
3. $\sigma(j)$ is the maximum number of bits that may come out during j consecutive time slots. In particular, $\sigma(0) = 0$.

We now show by induction on t that there exists some $s \geq 0$ such that $x(t - s) = 0$ and $R^*]t - s, t] = \sigma(s)$. This will imply that the shaper has σ as a queue-service-curve in the sense of Cruz, and thus as an extended service curve as well. Note that we are not actually showing a stronger result since, for a shaper with shaping curve σ, having σ as an extended service curve necessarily implies that σ is also a queue service curve.

The induction property is trivial if $x(t) = 0$ (take $s = 0$) and thus is true for $t = 1$.

Assume now that the induction property holds for $1, \ldots t - 1$. Call $M(t)$ the credit available at time slot t, namely $M(t) = \min_{u \geq 1} \{\sigma(u) - R^*]t - u, t]\}$.

If $M(t) > R^*]t - 1, t]$ then $x(t) = 0$), otherwise we would have served more bits during time slot t. The induction property at t is then trivially true in that case. Otherwise, call u^* the *maximum* u such that $1 \leq u \leq t$ and $\sigma(u) = R^*]t - u, t]$. We now need to show that $x(t - u^*) = 0$. Apply the induction property at time $t - u^*$. There exists some $s \geq 0$ such that $x(t - u^* - s) = 0$ and $R^*]t - u^* - s, t - u^*] = \sigma(s)$. Since σ is sub-additive, we have

$$\sigma(s) + \sigma(u^*) \geq \sigma(s + u^*)$$

It follows that

$$R^*]t - u^* - s, t] = \sigma(s) + \sigma(u^*) \geq \sigma(s + u^*)$$

and since the shaper is a policer, we have

$$R^*]t - u^* - s, t] \leq \sigma(s + u^*)$$

which shows that we have equality in the last inequation. From the definition of u^*, it follows that $s = 0$, which proves that $x(t - u^*) = 0$. □

Remember from Section 4.2 that it is not a restriction to assume that σ is sub-additive, and that a sufficient condition for σ to be sub-additive is that it is concave. Also, the condition $\sigma(0) = 0$ is not a restriction either, and we can always enforce it since we do not require σ to be continuous. Note that the converse of Theorem 6 is not true: offering a service curve of β is not sufficient to guarantee that the output is constrained by β, even if β is sub-additive.

The above result is key. It enables us to apply simple calculus rules from the previous Section. As an example, consider the issue of the buffer size required at a re-shaper. Re-shaping may be needed because the output of a buffer normally does not conform any more with the traffic contract specified at the input. Specifically, with the model in [16], a flow with arrival curve $\sigma(t) = \min(pt + M, rt + b)$ traverses a sequence of nodes, that offer a network service curve $\beta(t) = R(t - T)^+$. A re-shaper is placed after the sequence of nodes. From Theorem 1, the buffer requirement at the re-shaper is bounded by the vertical deviation between the shaping curve and the arrival curve of the flow when it reaches the buffer. The latter is given by Theorem 2. We can then easily find, as in [16], that the buffer requirement is bounded by B with

$$B = \begin{cases} \text{if } \frac{b-M}{p-r} < T & \text{then } b + Tr \\ \text{if } \frac{b-M}{p-r} \geq T \text{ and } p > R & \text{then } M + \frac{(b-M)(p-R)}{p-r} + TR \\ & \text{else } M + Tp \end{cases}$$

Let us now consider an application to a more general result. When re-shapers are introduced along a path, they act as additional buffers, that could increase the end-to-end delay. However, we can derive easily that they "come for free".

Proposition 7 (Shaping does not increase the delay bound) *Assume a flow, constrained by arrival curve α, is input to networks $\mathcal{S}_1$ and $\mathcal{S}_2$ in sequence. Assume a shaper, with curve σ is added between $\mathcal{S}_1$ and $\mathcal{S}_2$. Assume that σ is sub-additive. Then the delay bound given by Theorem 3 for the system without shaper is also valid for the system with shaper.*

The proof is now simple. Call β_i the extended service curve of network $\mathcal{S}_i$. The delay bound for the system with shaper is $h(\alpha, \beta_1 \oplus \alpha \oplus \beta_2)$ which, by associativity and commutativity, is also equal to $h(\alpha, \alpha \oplus \beta_1 \oplus \beta_2) = h(\alpha, \beta_1 \oplus \beta_2)$ from Proposition 5. □

A restricted form of Proposition 7 is found in [16], [6] or [3], where only leaky buckets are considered. We have here a more general, and simpler result. We will see in Section 7 that shapers are not limited to be simple leaky buckets.

As a further illustration for the power of the concepts introduced in Section 4, consider a less restrictive re-shaper [16] that is associated with a curve of the form

$$\sigma(t) = \min[Rt, rt + b] \tag{17}$$

Such a shaper is often called *buffered leaky bucket*. It enforces both a peak rate R and a sustainable rate r with burstiness parameter b. Consider now a series of nodes, and introduce such a shaper on a flow with maximum packet size M, peak rate p, sustainable rate r and burst tolerance b. With the same notation as in Proposition 7, the delay bound computed from Equation (2) is $d' = h(\alpha, \beta_1 \oplus \sigma \oplus \beta_2)$, with σ given by Equation (17) and $\alpha(t) = \min(M + pt, b + rt)$. From Section 4, $\sigma = \lambda_R \oplus (pt + b)_t$, where we note $(pt + b)_t$ the function $t \to pt + b$. Thus

$$d' = h(\alpha, (pt + b)_t \oplus \beta_1 \oplus \lambda_R \oplus \beta_2) = h(\alpha, \beta_1 \oplus \lambda_R \oplus \beta_2)$$

since $\alpha(t) \leq pt + b$ and by application of Corollary 1. Assume as in [16] that all buffers along the path guarantee a service curve of the form $\phi_i(t) = R_i(t - T_i)^+$ (so that β_1 and β_2 also have a similar form). Then, by application of Rule 3:

$$\beta_1 \oplus \lambda_R \oplus \beta_2 = \beta_1 \oplus \beta_2$$

as soon as at least one of the nodes has a rate $R_i \geq R$. We have thus just shown, without much computation, that a re-shaper of the form described in Equation (17) does not increase the delay bound, as long as R is as large as the minimum of the rates allocated to the flow along the path; this result is quoted in [16].

Let us apply now our calculus to a characterization of the output flow from a shaper. Of course, the output is constrained by the shaping curve. However, the fact that the shaper also offers its shaping curve as a service curve implies a much stronger result. Indeed:

Theorem 7 (Shaper Output Flow) *Assume a flow with arrival curve α is input to a shaper with shaping curve σ. The output flow is constrained by the arrival curve*

$$\alpha^* = [\min(\alpha, \sigma)] \ominus \sigma$$

Proof: For some arbitrary s and t, consider $R^*(t) - R^*(t - s)$, namely the number of bits output during time interval $[t - s, t]$. By Theorem 6, there exists some u such that

$$R^*(t - s) - R(t - s - u) \geq \sigma(u) \tag{18}$$

Now by definition of a shaper,

$$R^*(t) - R^*(t - s - u) \leq \sigma(s + u)$$

Since $R^* \leq R$, this gives

$$R^*(t) - R^*(t - s) \leq \sigma(s + u) - \sigma(u) \tag{19}$$

On the other hand, from Equation (18) we also have

$$R^*(t) - R^*(t - s) \leq R^*(t) - R(t - s - u) - \sigma(u) \leq R(t) - R(t - s - u) - \sigma(u)$$

Now the input during time interval $[t-s-u, t]$ is bounded by $\alpha(s+u)$. We have thus:

$$R^*(t) - R^*(t-s) \le \alpha(s+u) - \sigma(u)$$

Putting the last equation together with Equation (19) gives

$$R^*(t) - R^*(t-s) \le \min[\sigma(s+u), \alpha(s+u)] - \sigma(u) \tag{20}$$

The proposition now follows from Equation (20) and the definition of the $\ominus$ operator. $\square$

This characterization of the output flow seems to be quite accurate. Indeed, we are able to derive from it the following important result.

Theorem 8 (Shaping Conserves Arrival Constraints) *Assume a flow with arrival curve α is input to a shaper with shaping curve σ. Assume σ and α are concave, with $\alpha(0) = \sigma(0)$. Then the output flow is still constrained by the original arrival curve α.*

Note that the output flow is necessarily constrained by the shaping curve. The theorem states that any constraint on the input flow, enforced upstream of the shaper, is not undone by the shaper. The theorem generalizes a result by Cruz ([3], Section VI) which is obtained, after long derivations, in the specific case of leaky buckets and linear arrival curves.

Proof: From Theorem 7, the output is constrained by

$$\alpha^* = ([\min(\alpha, \sigma)] \ominus \sigma)$$

We have assumed that σ and α are concave, therefore, $\min(\alpha, \sigma)$ is concave (as minimum of concave functions) and thus sub-additive. The result now derives from Proposition 6 applied to $\gamma_1 = \min(\alpha, \sigma)$ and $\gamma_2 = \sigma$. $\square$

7 Deterministic Effective Bandwidth

In this Section we show how the concept of effective bandwidth can be defined simply in a deterministic context; we also illustrate an application of Theorem 6.

7.1 Effective Bandwidth of a Flow

We start by considering a trunk system that serves a flow in a work conserving manner, at a constant rate C. We assume the arrival flow is constrained by an arrival curve α and would like to characterize the minimum value of C that is required for a given α. This problem has been studied in [17], pp 270–273, in the specific case of a flow constrained by one leaky bucket. The authors in [17] find that, if we impose a fixed delay constraint D to the flow, then the condition on C is that $C \ge C_D$, where C_D depends on the leaky bucket parameters and the delay constraint. C_D is called the (deterministic) effective bandwidth of the flow, for a delay constraint of D. If N identical flows are superimposed, the effective bandwidth of the aggregate flow is NC_D; in contrast, for a heterogeneous mix of flows, there is no such additive property.

We show now how these results derive from a more general concept. Back to the general case, for a given arrival curve α, we wish to find a rate C such that the horizontal deviation $h(\alpha, \lambda_C)$ is not more than D. This is equivalent to requiring that $\alpha(s) \le C(s+D)$ for all $s \ge 0$, which in turn can be expressed as $C \ge \sup_{s\ge 0} \frac{\alpha(s)}{s+D}$. We have thus shown the following:

Proposition 8 (Effective Bandwidth) *The queue with constant rate C guarantees a delay bound of D to a flow with arrival curve α if $C \ge e_D(\alpha)$, with*

$$e_D(\alpha) = \sup_{s\ge 0} \frac{\alpha(s)}{s+D} \tag{21}$$

We call $e_D(\alpha)$ the *effective bit rate*, or *deterministic effective bandwidth* corresponding to the arrival curve α, for a delay constraint D. If α is differentiable, $e(D)$ is the slope of the tangent to the arrival curve, drawn from the time axis at $t = -D$ (Figure 4).

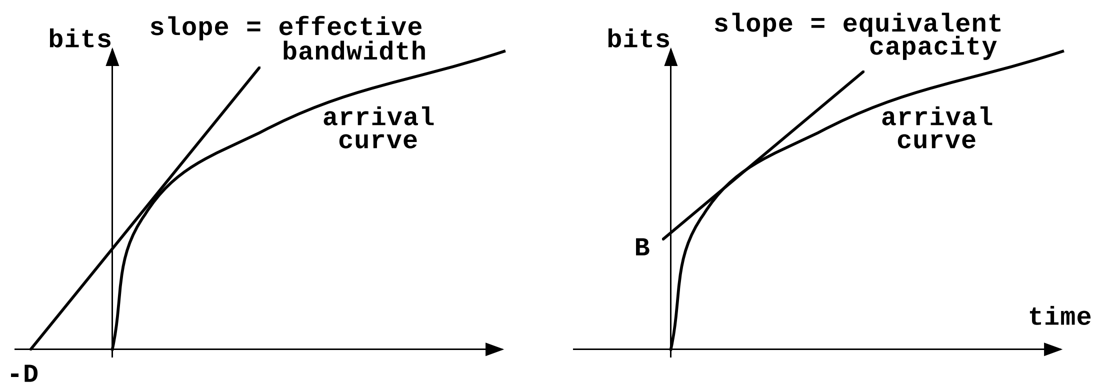

Figure 4: Effective Bandwidth for a delay constraint D and Equivalent Capacity for a buffer size B

We define the sustainable rate m as $m = \limsup_{s\to+\infty} \frac{\alpha(s)}{s}$ and the peak rate by $p = \sup_{s\ge 0} \frac{\alpha(s)}{s}$. Then $m \le e_D(\alpha) \le p$ for all D. Moreover, if α is concave, then $\lim_{D\to+\infty} e_D(\alpha) = m$.

For example, for a flow constrained according to the IETF specification, with maximum packet size M, peak rate p, sustainable rate r and burst tolerance b, the effective bandwidth is the maximum of r and the slopes of lines (QA_0) and (QA_1) on Figure 5; it is thus equal to:

$$e_D = \max\left\{\frac{M}{D}, r, p\left(1 - \frac{D - \frac{M}{p}}{x + D}\right)\right\} \tag{22}$$

with $x = \frac{b-M}{p-r}$.

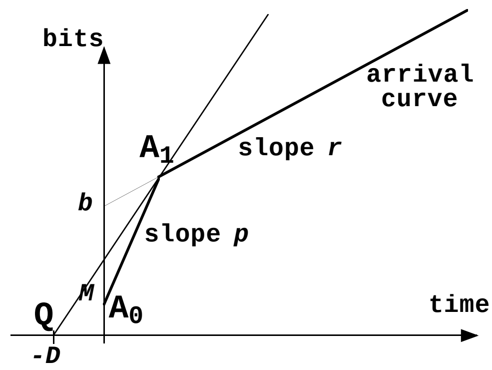

Figure 5: Effective Bandwidth for an arrival curve according to the IETF specification

It follows also directly from the definition in (21) that

$$e_D(\sum_i \alpha_i) \leq \sum_i e_D(\alpha_i)$$

In other words, the effective bandwidth for an aggregate flow is less than or equal to the sum of effective bandwidths. If the flows have all *identical* arrival curve, then the aggregate effective bandwidth is simply $I \times e_D(\alpha_1)$. It is this latter relation which is the origin of the term "effective bandwidth". The difference $\sum_i e_D(\alpha_i) - e_D(\sum_i \alpha_i)$ is a buffering gain; it tells us how much capacity is saved by sharing a buffer between the flows.

In more general terms, the effective bandwidth is a convex function of function α, namely $e_D(a\alpha_1 + (1-a)\alpha_2) \leq a e_D(\alpha_1) + (1-a) e_D(\alpha_2)$. This means also that the acceptance region is convex, for a call acceptance criterion based solely on a delay bound and zero data loss. This contrasts with acceptance regions based on statistical multiplexing with large deviation asymptotics. In such cases, it is the complement in the positive orthant which is convex [5].

Similar results hold if we replace delay constraints by the requirement that a fixed buffer size is not exceeded. Indeed, the queue with constant rate C, guarantees a maximum backlog of B (in bits) for a flow with arrival curve α if $C \geq f_B(\alpha)$, with

$$f_B(\alpha) = \sup_{s \geq 0} \frac{\alpha(s) - B}{s} \tag{23}$$

We call $f_B(\alpha)$ the *equivalent capacity*, by analogy to the work in [12]. Similar to effective bandwidth, the equivalent capacity of a heterogeneous mix of flows is less than or equal to the sum of equivalent capacities of the flows, provided that the buffers are also added up; in other words, $f_B(\alpha) \leq \sum_i f_{B_i}(\alpha_i)$, with $\alpha = \sum_i \alpha_i$ and $B = \sum_i B_i$. Figure 4 gives a graphical interpretation.

Note that formulas (21) or (23), or both, can be used to estimate the capacity required for a flow, based on a measured arrival curve (for example as in Proposition 4).

7.2 Tunneling

We show now how the network calculus presented in this paper can be used to estimate the resources required in a tunneling scenario. Tunneling refers to the multiplexing of several flows into a larger flow, which is handled in subsequent nodes as a single entity (called a virtual trunk in [15]). Tunneling occurs when a number of RSVP flows are multiplexed onto one single ATM connection, or over one RSVP flow itself. We expect tunneling to play an important role in the scalability of integrated services networks.

We assume in this section that a number of flows, with an aggregate arrival curve α, is multiplexed into a virtual trunk. The virtual trunk is viewed as a single flow by downstream nodes; as such, it is constrained by an arrival curve σ. We are interested in finding, for a given arrival curve α, the optimal values of σ, for a criterion that we define later. An exhaustive study of this problem is far beyond the scope of this paper. In contrast, we show how network calculus can be employed in this context; indeed, it is the study of this problem that motivated the derivation of the results in this paper, in particular Section 6.

If we assume a work conserving scheduling policy, then the multiplexing node acts as a general shaper. From Theorem 6, we can conclude that it offers a service curve σ to the aggregate flow. Assume that the constraint at the multiplexor node is to guarantee a maximum delay D. From Theorem 3, the requirement on the virtual trunk is that, for all $s \geq 0$, we have:

$$\sigma(s + D) \geq \alpha(s) \tag{24}$$

Optimal Parameters for an ATM Variable Bit Rate Virtual Trunk As an example, assume the virtual trunk is an ATM Variable Bit Rate (VBR) connection. The shaping curve for a VBR connection has the form $\sigma(t) = \min(Pt, St + B)$ where P is the peak rate, S the sustainable rate and B a burst tolerance parameter. Equation (24) is illustrated on Figure 6.

Equation (24) becomes:

$$\text{for all } s \geq 0 : \left\{ \begin{array}{l} (s + D)P \geq \alpha(s) \\ (t + D)S + B \geq \alpha(s) \end{array} \right. \tag{25}$$

The first condition in Formula (25) implies that $P \geq e_D(\alpha)$, which is thus a necessary and sufficient condition on P. In other words, we have shown that for a virtual trunk of the ATM VBR type, *there is an minimum peak rate P_0, which is the effective bandwidth of the arrival stream* and that this minimum peak rate is also optimal. More precisely, the latter statement means the following. We say that the parameter set (P, S, B) of the virtual trunk is feasible if the virtual trunk is able to carry the traffic with a delay less than the delay constraint D. The result is that, if (P, S, B) is feasible, then on one hand, necessarily $P \geq P_0 = e_D(\alpha)$, and on the other hand, (P_0, S, B) is also feasible. Another aspect of this result is that, from a bandwidth point of view, using a VBR trunk rather than a constant bit rate (CBR) trunk is all benefit since, by definition of the effective bandwidth, the CBR trunk would have a rate of at least P_0.

If we wish to solve the problem of finding a complete optimal parameter set for (B, M), then we need an optimality criterion. As an example, we assume that we wish to minimize a cost

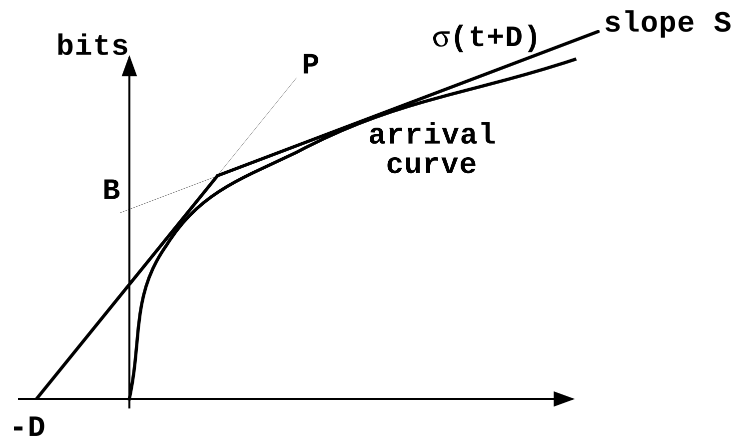

Figure 6: Virtual Trunk with parameters (P, S, B) satisfies the delay constraint D for traffic with arrival curve drawn on the picture.

function $c(P, M, B)$ which is affine and wide-sense increasing in each of its variables. The value of P should thus be set to P_0 and we wish to minimize $c(P_0, S, B) - c(P_0, 0, 0) = uS + vB$ for fixed values of u and v. We can already say is that the set of acceptable values (S, B) is defined by

$$\text{for all } s \geq 0: \quad B + (s + D)S - \alpha(s) \geq 0 \tag{26}$$

and is thus a convex region (as intersection of half-planes). Assume α is concave. Any point on the border of the region satisfies $B + (t + D)S - \alpha(s) = 0$ for some $s \geq s_0$, where s_0 reaches the minimum of $\frac{\alpha(s)}{s+D}$. It follows that any point on the border of the region necessarily satisfies $m < S \leq P_0$, where m is the sustainable rate $m = \lim_{s \to +\infty} \frac{\alpha(s)}{s}$. Also, for any given point (S, B) on the border we have

$$B = \sup_{s \geq 0} \{\alpha(s) - (s + D)S\} \tag{27}$$

and

$$S = \sup_{s \geq 0} \frac{\alpha(s) - B}{(s + D)} \tag{28}$$

The values of (S, B) are also normally limited not to exceed maximum values $S_{\max}$ resp. $B_{\max}$. We thus have to minimize $uS + vB$ in the region defined by (26) and $S \leq S_{\max}$, $B \leq B_{\max}$. By convexity of the region, the minimum is for a point on the border.

If $\frac{u}{v} \geq D$ then we first compute the point $(\bar{S}, \bar{B})$ on the border where the tangent is orthogonal to (u, v). We have:

$$\bar{S} = \sup_{s \geq 0} \frac{\alpha(s) - \alpha(\frac{u}{v} - D)}{(s + D - \frac{u}{v})} \tag{29}$$

$$\bar{B} = \alpha(\frac{u}{v} - D) - \frac{u}{v}\bar{S} \tag{30}$$

Finally, after some algebra, we find the optimal value (P_0, S_0, B_0) according to the algorithm in Figure 7.

$P_0 = \sup_{s \geq 0} \frac{\alpha(s)}{s+D}$

if $\frac{u}{v} < D$ {
    $S_0 = \min(P_0, S_{\max})$
    $B_0 = \sup_{s \geq 0}\{\alpha(s) - (s+D)S_0\}$
}
else {
    $\bar{S} = \sup_{s \geq 0} \frac{\alpha(s) - \alpha(\frac{u}{v} - D)}{(s + D - \frac{u}{v})}$
    $\bar{B} = \alpha(\frac{u}{v} - D) - \frac{u}{v}\bar{S}$

    if $\bar{S} > S_{\max}$ **and** $\bar{B} > B_{\max}$ there is no feasible solution

    else if $\bar{S} > S_{\max}$ {
        $S_0 = \min(P_0, S_{\max})$
        $B_0 = \sup_{s \geq 0}\{\alpha(s) - (s+D)S_0\}$
    }
    else if $\bar{B} > B_{\max}$ {
        $S_0 = \sup_{s \geq 0} \frac{\alpha(s) - B_{\max}}{s+D}$
        $B_0 = B_{\max}$
    }
    else {
        $S_0 = \bar{S}$
        $B_0 = \bar{B}$
    }
}

Figure 7: The algorithm for the optimal VBR virtual trunk parameters (P_0, S_0, B_0) for a given arrival curve α and a delay constraint D

The above results generalizes immediately to the case where the virtual trunk arrival curve is defined by any number of affine constraints (not just two as in this example). Similar results hold if a maximum buffer requirement is considered.

The algorithm in Figure 7 can be used to estimate the optimal characteristics of the virtual trunk, based on a measured arrival curve α.

8 Conclusion

We have shown how the calculus of Cruz can be extended, and simplified. All properties are elementary, and easy to interpret physically. We have shown on a few examples how they can be used effectively.

References

[1] 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.

[2] R.L. Cruz. A calculus for network delay, part ii: Network analysis. *IEEE Trans. Inform. Theory, vol 37-1*, pages 132–141, January 1991.

[3] R.L. Cruz. Quality of service guarantees in virtual circuit switched networks. *IEEE JSAC*, pages 1048–1056, August 1995.

[4] G. De Veciana, July 1996. Private Communication.

[5] Kelly F. Effective bandwidths of multiclass queues. *Queueing Systems (9)-1*, pages 5–15, 1991.

[6] 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.

[7] Golestani S. J. A self clocked fair queuing scheme for high speed applications. In *Proceedings of Infocom '94*, 1994.

[8] Kurose J. On computing per-session performance bounds in high-speed multi-hop computer networks. In *Proc ACM Sigmetrics / Performance '92*, 1992.

[9] Le Boudec J.-Y. The asynchronous transfer mode: A tutorial. *Computer Networks and ISDN Systems*, May 1992.

[10] Parekh A. K. and Gallager R. G. 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.

[11] Parekh A. K. and Gallager R. G. 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.

[12] Gun L. and R. Guérin. Bandwidth management and congestion control framework of the broadband network architecture. *Computer Networks and ISDN Systems, vol 26*, pages 61–78, 1993.

[13] Zhang L. A new traffic control algorithm for packet switching networks. In *Proceedings of ACM Sigcomm '90*, 1990.

[14] 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 Audion and Video, Durham NH*, April 1995.

[15] Giordano S., Le Boudec J.-Y., Oechslin P., and Robert S. Vbr over vbr: the homogeneous, loss-free case. In *Infocom '97*, 1997.

[16] S. Shenker, Partridge C., and Guérin R., August 1996. Specification of Guaranteed Quality of Service, draft-ietf-intserv-guaranteed-svc-06.

[17] P. Varayia and J. Walrand. *High Performance Communication Networks.* Morgan Kaufmann, San Francisco, 1996.

[18] H. Zhang. Service disciplines for guaranteed performance service in packet switching networks. *Proceedings of the IEEE*, 1996.

A Appendix

A.1 Proof of Proposition 3

Apply lemma 1 to $A = 0, B = s + t, x = s, y = t$. □

Lemma 1 *Let ϕ be a* concave *function defined on $[A, B]$ with A, B in $\mathbb{R}$. For all x, y in $[A, B]$ such that $x + y = A + B$ we have*

$$\phi(x) + \phi(y) \geq \phi(A) + \phi(B)$$

Proof: Define function γ by $\gamma(u) = \phi(x + u) - \phi(x)$ Functions γ and ϕ have right-handside derivatives γ' and ϕ' satisfying

$$\gamma'(u) = \phi'(x + u)$$

By concavity of ϕ, ϕ' is wide-sense decreasing, thus

$$\gamma'(u) \leq \phi'(A + u)$$

and therefore

$$\gamma(u) - \gamma(0) \leq \phi(A + u) - \phi(A)$$

for all u such that both $A + u$ and $x + u$ are in $[A, B]$. Apply the last formula to $u = B - x$ to obtain the required inequality.