Foundations and Trends® in Networking
Vol. 11, No. 3-4 (2016) 139–282

DOI: 10.1561/1300000059

Duality of the Max-Plus and Min-Plus Network Calculus

Jörg Liebeherr
University of Toronto
jorg@ece.utoronto.ca

Contents

Abstract

The network calculus is a framework for the analysis of communication networks, which exploits that many computer network models become tractable for analysis if they are expressed in a min-plus or max-plus algebra. In a min-plus algebra, the network calculus characterizes amounts of traffic and available service as functions of time. In a max-plus algebra, the network calculus works with functions that express the arrival and departure times or the required service time for a given amount of traffic. While the min-plus network calculus is more convenient for capacity provisioning in a network, the max-plus network calculus is more compatible with traffic control algorithms that involve the computation of timestamps. Many similarities and relationships between the two versions of the network calculus are known, yet they are largely viewed as distinct analytical approaches with different capabilities and limitations. We show that there exists a one-to-one correspondence between the min-plus and max-plus network calculus, as long as traffic and service are described by functions with real-valued domains and ranges. Consequently, results from one version of the network calculus can be readily applied for computations in the other version. The ability to switch between min-plus and max-plus analysis without any loss of accuracy provides additional flexibility for characterizing and analyzing traffic control algorithms. This flexibility is exploited for gaining new insights into link scheduling algorithms that offer rate and delay guarantees to traffic flows.

J. Liebeherr. *Duality of the Max-Plus and Min-Plus Network Calculus*. Foundations and Trends® in Networking, vol. 11, no. 3-4, pp. 139–282, 2016.
DOI: 10.1561/1300000059.

1

Introduction

Network calculus is a methodology for performance evaluation of communication networks that expresses the analysis of networks in a min-plus or max-plus algebra. In these algebras, the conventional addition and multiplication operations are replaced by the minimum or maximum operation, respectively, and addition. On the one hand, algebras with a minimum or maximum operation have weaker properties than algebras endowed with an addition and a multiplication. For instance, the minimum and the maximum do not have inverse operations. On the other hand, taking minimums and maximums creates strong ordering properties that can be analytically exploited. Network algorithms that involve sequencing of traffic, *e.g.*, scheduling with a sorted queue, or ordering of events, *e.g.*, window flow control, can often be described by linear systems in a min-plus or max-plus algebra, but are non-linear in an algebra with addition and multiplication.

The deterministic analysis of networks by Cruz in [13, 14] and its application to Generalized Processor Sharing scheduling by Parekh and Gallager in [28, 29] mark the beginning of network calculus research. The research was motivated by the emergence of communication networks that provide service guarantees even in adversarial worst-case scenarios. Within a few years, researchers recognized that non-traditional algebras, so-called *dioids*, for modelling discrete-event dynamic systems [3] can provide the foundation for a systems theory for communication networks [1, 6, 8]. The dioid algebras are applied to non-decreasing functions that represent cumulative arrival, departure and service processes in a network. The essence of the systems theory is that the departure traffic at a network element can be characterized by a convolution of functions describing the cumulative arrivals and the available service. The convolution operation is performed either in a min-plus or max-plus dioid algebra, leading to the min-plus and max-plus versions of the network calculus. Detailed models have been developed for many types of network elements, such as buffered links with FIFO or more complex scheduling algorithms, delay elements, traffic regulators, and many more. Comprehensive discussions can be found in textbooks on the topic [7, 9].

Network calculus analysis can select either a min-plus or max-plus algebra setting, yet, overwhelmingly, the literature presents derivations in a min-plus framework. In such a setting, arrivals and departures are represented as functions of time, where a function value $F(t)$ represents the amount of arriving or departing traffic until time t. This representation is convenient when performing computations with multiplexed traffic flows, since an aggregate of traffic flows that are characterized by functions $F_1(t), F_2(t), \ldots, F_N(t)$ is simply the sum $\sum_j F_j(t)$. Expressions for multiplexed traffic flows are needed when determining capacity requirements for a network, *e.g.*, the maximum number of flows that can be supported in a network subject to given service requirements. The representation of traffic by functions of time is less ideal when describing network control algorithms that assign timestamps to traffic. An example is a traffic regulator that determines the earliest time when a packet can be admitted to a network, or a scheduling algorithm that assigns deadlines for the departure time of packets. Obtaining timestamps from a function of the form $F(t)$ requires to solve an inverse problem. In a max-plus framework, arrivals and departures are characterized by functions $F(\nu)$ that give the arrival time or departure time of the ν-th bit or packet. For example, at a traffic regulator, the timestamp that determines when the ν-th bit or packet can be admitted is simply the value of the departure time function at ν. On the other hand, expressions for multiplexed traffic in the max-plus algebra are cumbersome (as we will see in §3).

Ideally, network analysis should be able to reconcile the advantages of the min-plus and max-plus network calculus algebras. That is, it should be able to employ functions $F(\nu)$ when working with network mechanisms involving timestamps and functions $F(t)$ when multiplexing traffic. Such mix-and-match computations require that the functions $F(t)$ and $F(\nu)$ can be related to each other. Mappings between expressions in the min-plus and max-plus network calculus, and vice versa, exist in the literature (see §13), however, since the mappings are (generally) not one-to-one, performing them comes at a loss of accuracy. At present, the prevailing view is that "many concepts [of the min-plus algebra] can be mirrored in the max-plus algebra," [20, p. 63], but also that not every result in the min-plus algebra can be extended to a max-plus setting [9, Remark 6.2.7] and that there is a lacking correspondence between concepts in the min-plus and max-plus algebra [7, §1.10]. On the other hand, according to dioid theory, the underlying min-plus and max-plus algebras of integer or real numbers are isomorphic [3, 21]. Thus, the question arises why the isomorphism does not extend to the min-plus and max-plus network calculus, which are based on these algebras? Our objective is to explore this question. We find that there exists a one-to-one relationship between the min-plus and max-plus network calculus, as long as both approaches are using functions that have a real-valued, that is, continuous-time or continuous-space, domains. Some of the previously observed differences between max-plus and min-plus analysis can be traced to the use of functions with a discrete-valued domain. After establishing the duality between the two versions of the network calculus, we proceed to characterize scheduling algorithms with rate and delay guarantees by service curves of the network calculus.

The remainder is structured as follows. In §2, we show that the max-plus convolution operation emerges when we describe the departures at a work-conserving link in terms of the arrivals and the link capacity. We observe that the expression for the departures is sensitive to the choice of measuring traffic in discrete units (bits, bytes, or packets) or by a real-valued metric. In §3–8, we present a self-contained description of the max-plus network calculus. In §9, we summarize the definitions and main results of the min-plus network calculus, which are later used for comparisons between the two network calculus versions. In §10, we show that the min-plus algebra and max-plus algebra for non-decreasing functions endowed with a minimum (or maximum) and a convolution operation are isomorphic to each other. We use the isomorphism in §11 to establish a duality of service curves, traffic envelopes, and performance bounds. In §12, we express scheduling algorithms for rate guarantees in terms of the continuous-space max-plus network calculus, and establish a connection between well-known scheduling algorithms and expressions in the max-plus algebra. In §13, we discuss the related literature with a focus on prior work on the max-plus network calculus, existing mappings between the min-plus and max-plus network calculus, and its relationship to lattice theory. We present conclusions in §14.

2

Motivation

Consider a work-conserving buffered link with a fixed transmission rate of C bits per second, as shown in Figure 2.1, which experiences arrivals of a sequence of packets. Arrivals that exceed the transmission rate are stored in a First-In-First-Out (FIFO) buffer, and become the backlog of the buffered link. A work-conserving buffered link transmits at rate C as long as there is a backlog in the buffer.

Let $T_A^p(n)$ and ℓ_n denote the arrival time and the size (in bytes or bits), respectively, of the n-th packet. The departure time of the n-th packet at the buffered link, denoted by $T_D^p(n)$, is determined by adding the transmission time $\frac{\ell_n}{C}$ of the packet either to its arrival time or the departure time of the

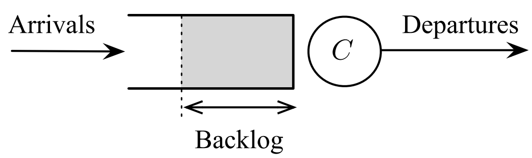


Figure 2.1: Work-conserving buffered link with rate C.

previous packet, whichever occurs later. We can write this as

$$T_D^p(n) = \max\{T_A^p(n), T_D^p(n-1)\} + \frac{\ell_n}{C}\,, \tag{2.1}$$

with $T_D^p(0) = 0$. If we expand the recursion in the expression, we obtain

$$\begin{aligned} T_D^p(n) &= \max\Big\{T_A^p(n) + \frac{\ell_n}{C}\ ,\ T_A^p(n-1) + \frac{\ell_{n-1}+\ell_n}{C}\,, \\ &\qquad\qquad \ldots\ , T_A^p(1) + \frac{\ell_1 + \ldots + \ell_n}{C}\Big\} \\ &= \max_{0 \le k \le n-1} \Big\{T_A^p(n-k) + \frac{\ell_{n-k} + \ldots + \ell_n}{C}\Big\}\,. \end{aligned} \tag{2.2}$$

We now perform a change of units, where, instead of marking the arrivals and departures of packets, we track individual bits. We use $T_A(\nu)$ and $T_D(\nu)$, respectively, to denote the arrival and departure times of the ν-th bit in the packet sequence, where $\nu = 0$ is the first bit in the sequence. If we express the recursion of (2.1) using a bit-level description, we obtain for $\nu \ge 0$ that

$$T_D(\nu) = \max\{T_A(\nu), T_D(\nu-1)\} + \frac{1}{C}\,, \tag{2.3}$$

where we set $T_D(\nu) = -\infty$ for $\nu < 0$. Expanding the recursion results in

$$\begin{aligned} T_D(\nu) &= \max\Big\{T_A(\nu) + \frac{1}{C}\,,\, T_A(\nu-1) + \frac{2}{C}\,,\ \ldots\ ,\, T_A(0) + \frac{\nu}{C}\Big\} \\ &= \max_{\kappa=0,1,\ldots,\nu} \{T_A(\nu-\kappa) + \frac{\kappa+1}{C}\}\,. \end{aligned} \tag{2.4}$$

Suppose we introduce an operation $\overline{\otimes}$ for two functions f and g such that

$$f \,\overline{\otimes}\, g(\nu) = \max_{\kappa=0,1,\ldots,\nu} \{f(\nu-\kappa) + g(\kappa)\}\,,$$

and define a function $\gamma_S(\nu) = \frac{\nu+1}{C}$. Then we can write (2.4) as

$$T_D(\nu) = T_A \,\overline{\otimes}\, \gamma_S(\nu)\,. \tag{2.5}$$

We refer to the $\overline{\otimes}$ operation as *max-plus convolution*. Alternatively, defining $\gamma_S'(\nu) = \frac{\nu}{C}$, the departure time can be characterized as

$$T_D(\nu) = T_A \,\overline{\otimes}\, \gamma_S'(\nu) + \frac{1}{C}\,. \tag{2.6}$$

In a bit-level description of packetized arrivals, all bits belonging to the same packet arrive instantaneously. If we denote the cumulative size of the first n packets by L_n, with $L_n = \sum_{j=1}^{n} \ell_j$ and $L_o = 0$, the arrival times of the bits of the n-th packet are given by

$$T_A(L_{n-1}) = T_A(L_n - l_n + 1) = \ldots = T_A(L_n - 1)\,,$$

for each $n \geq 1$. The departure time of a packet in a bit-level description is the departure time of the last bit of the packet, so that $T_D^p(n) = T_D(L_n - 1)$ for the n-th packet. With packetized arrivals and departures, the departure time of the n-th packet, expressed with (2.5), is

$$T_D(L_n - 1) = T_A \,\overline{\otimes}\, \gamma_S(L_n - 1)\,,$$

which is identical to the packet-level expression in (2.2).

The convolution expressions in (2.5) and (2.6) contain a discretization artifact due to the bit-level description. In (2.5), a term $\frac{1}{C}$ is added to the service function, and in (2.6), the departure time $T_D(L_n - 1)$ is increased by the same amount. To observe the artifact, consider that the unit in which we measure traffic is $\frac{1}{k}$-th of a bit. Then, the recursion in (2.3) and (2.4) becomes

$$\begin{aligned} T_D(L_n - \tfrac{1}{k}) &= \max\{T_A(L_n - \tfrac{1}{k}), T_D(L_n - \tfrac{2}{k})\} + \frac{1}{kC} \\ &= \max_{\kappa = 0, \frac{1}{k}, \frac{2}{k} \ldots, L_n - \frac{1}{k}} \Big\{ T_A(L_n - \tfrac{1}{k} - \kappa) + \frac{\kappa + \frac{1}{k}}{C} \Big\}\,. \end{aligned} \tag{2.7}$$

Letting $k \to \infty$, we obtain a fluid-flow description of traffic, where traffic volume is measured by a real number. Then, the values $\nu \in [L_{n-1}, L_n)$ designate the data in the n-th packet. If we use the notation $T_D(L_n^-) = \sup_{\kappa < L_n} T_D(\kappa)$, and then take the limit $k \to \infty$ in (2.7), we obtain

$$T_D(L_n^-) = \sup_{0 \leq \kappa \leq L_n^-} \Big\{ T_A(L_n^- - \kappa) + \frac{\kappa}{C} \Big\}\,.$$

Defining the max-plus convolution for functions f and g with real-valued arguments as $f \,\overline{\otimes}\, g(\nu) = \sup_{0 \leq \kappa \leq \nu}\{f(\nu - \kappa) + g(\kappa)\}$, the departure time of the n-th packet is given by

$$T_D(L_n^-) = T_A \,\overline{\otimes}\, \gamma_S(L_n^-)\,, \tag{2.8}$$

with $\gamma_S(\nu) = \frac{\nu}{C}$. We see that the term $\frac{1}{kC}$ disappears when traffic volume is measured by real numbers. This is the main reason why, in the remainder, we prefer to work with fluid-flow arrival functions on a real-valued domain.

3

Modelling Traffic Arrivals in the Space Domain

Consider an arrival scenario of traffic over a time period of 2 milliseconds as depicted in Figure 3.1. Each vertical bar in the figure represents an arrival event of one or more packets. The arrivals are summarized in Table 3.1 in terms of the packet index n, the packet arrival time $T_A^p(n)$, and the packet size ℓ_n. There are seven arrival events with eight packet arrivals. Packet arrivals are assumed to be instantaneous and multiple packet arrivals may occur at the same time. For instance, the arrival event at time $t = 0.3$ ms consists of two packet arrivals, one with size 1000 bits and one with size 500 bits.

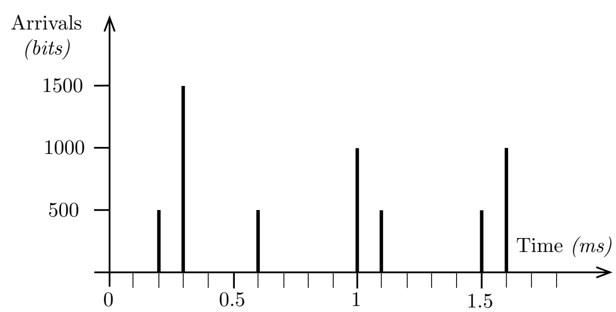
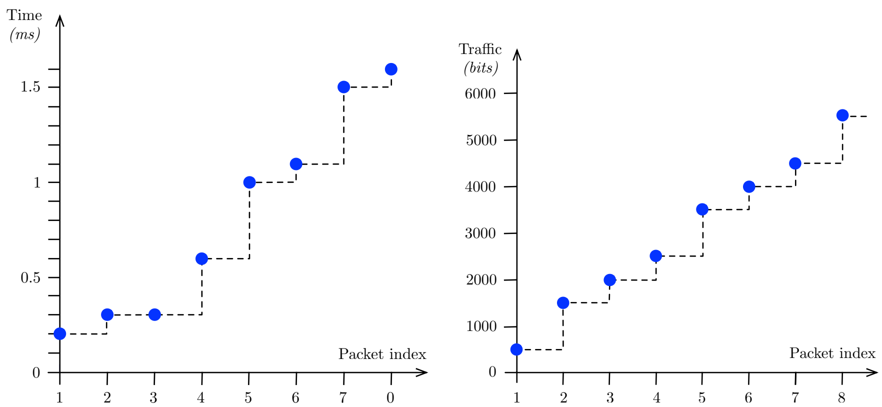


Figure 3.1: Arrival scenario. Each arrival event is indicated by a vertical line whose length indicates the total amount of arriving traffic.

Packet index n	1	2	3	4	5	6	7	8
$T_A^p(n)$ (ms)	0.2	0.3	0.3	0.6	1	1.1	1.5	1.6
ℓ_n (bits)	500	1000	500	500	1000	500	500	1000

Table 3.1: Parameters of the packet arrival scenario.



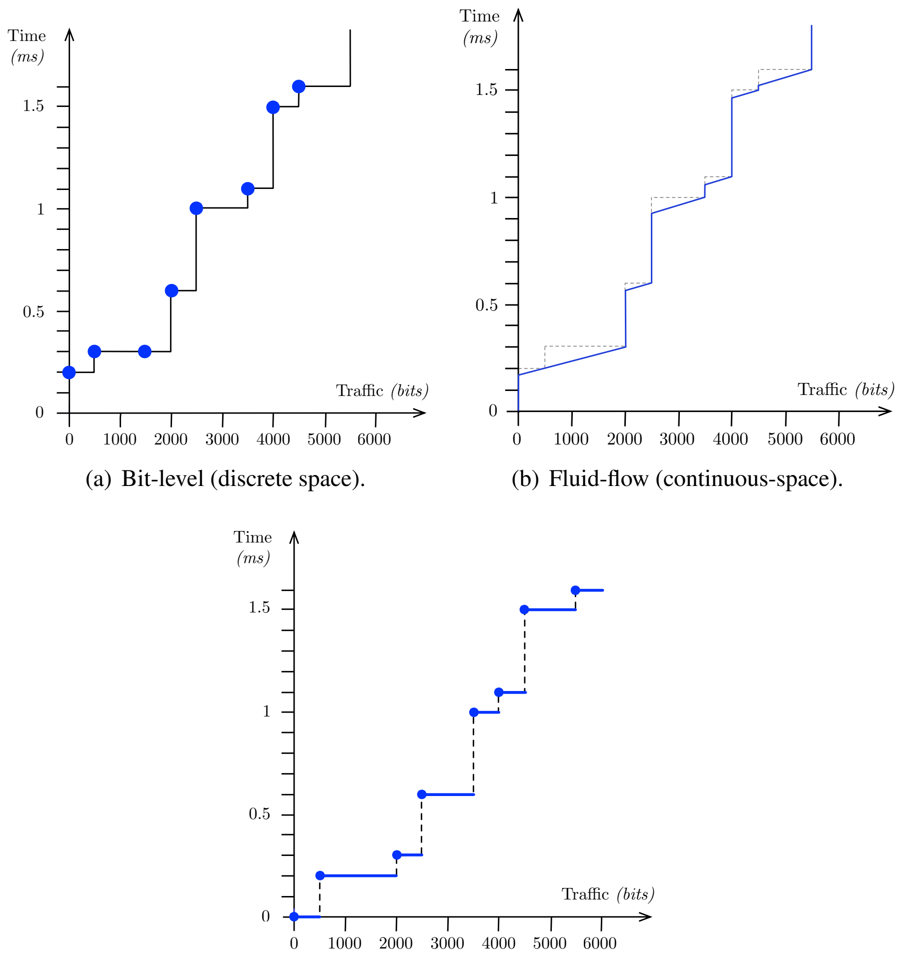
(a) Arrival time function $T_A^p(n)$.

(b) Cumulative packet size function L_n.

Figure 3.2: Packet-level arrival characterization.

Figure 3.2(a) illustrates $T_A^p(n)$, the arrival time of the n-th packet, as a function of the packet index n. The packet-level arrival time function does not encode information on the amount of data in a packet. Unless all packets have the same size, information on packet sizes must be provided separately, for example, by the cumulative packet size function L_n defined in §2. The function is shown in Figure 3.2(b).

A bit-level representation of arrivals dispenses with the need to maintain two functions to describe arriving traffic. Letting $T_A(\nu)$ denote the arrival time of the $(\nu - 1)$-th bit, we obtain the arrival time function T_A given in Figure 3.3(a). The function is defined for discrete values $\nu = 0, 1, \ldots$, and is therefore referred to as a *discrete-space function*. The markers in the figure indicate the function values that correspond to packet arrival instants. Due to our convention of counting bits starting at zero, the first packet covers values $\nu = 0, 1, \ldots, 499$ bits, the second covers $\nu = 500, \ldots, 1499$ bits, and so



(a) Bit-level (discrete space).

(b) Fluid-flow (continuous-space).

(c) Right-continuous (continuous-space).

Figure 3.3: Representations of the arrival time function T_A.

forth. In Figure 3.3(a), for $\nu \geq 5500$ bits, we set the arrival time function to $T_A(\nu) = \infty$, indicating that those bits will never arrive. An alternative is to limit the domain of $T_A(\nu)$ to $\nu < 5500$ bits.

When traffic is expressed using non-negative real numbers, the arrival time function is a *continuous-space* function. With continuous-space functions, we can express arrivals that occur continuously at a constant or variable rate. Figure 3.3(b) shows an arrival function T_A that is generated when the packet arrivals in Table 3.1 have passed through a link with a rate of 15 Mbps. In this case, the data of a packet does not arrive instantaneously but over a time interval whose length depends on the packet size. The dotted lines in the figure indicate the intervals over which arrivals occur. We continue to use 'bits' as the unit of traffic in a continuous-space setting, allowing bits to take any non-negative real value.

A continuous-space arrival time function with instantaneous arrivals of entire packets results in discontinuities ('jumps') that require a choice of the function value at the time instant of a jump. Depending on this choice, the arrival time function is right-continuous or left-continuous. As a convention, we will always interpret continuous-space arrival time functions T_A as right-continuous functions. Compared to the left-continuous alternative, a right-continuous function T_A is the more conservative choice, since it assigns a bit value a timestamp that is never earlier than that of the left-continuous version. Figure 3.3(c) illustrates a right-continuous function arrival time function T_A with instantaneous packet arrivals. For emphasis, the function values at points where T_A is not continuous are indicated by large dots.

When working with a non-decreasing function F that has jumps, we sometimes need to refer to the function value immediately to the left or to the right of a value x. This can be done with the notation $F(x^-)$ and $F(x^+)$, defined as

$$F(x^-) = \sup_{u<x} F(u) \quad \text{and} \quad F(x^+) = \inf_{u>x} F(u)\,. \tag{3.1}$$

A right-continuous function always satisfies $F(x) = F(x^+)$, and a left-continuous function satisfies $F(x) = F(x^-)$ for each $x \in \mathbb{R}$. If a right-continuous function has a jump at x, then $F(x^-)$ is the function value immediately before the jump, and $F(x)$ the function value after the jump. For a left-continuous function F with a jump at x, $F(x)$ is the function value

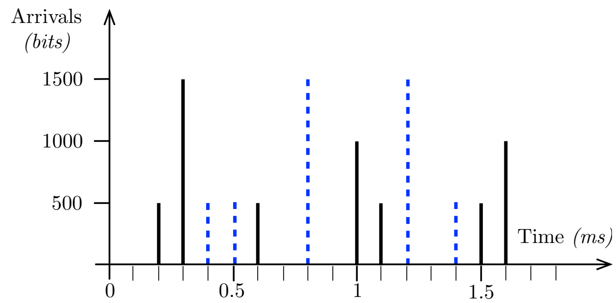


Figure 3.4: Arrival scenario with data aggregation (A solid line indicates a Flow 1 arrival, and a dashed line is a Flow 2 arrival).

before the jump and $F(x^+)$ the value after the jump.

Data aggregation. Let us consider an aggregate of several flows. When we represent arrivals as functions of time, flows are aggregated by simply adding the arrival functions. Let $A_j(t)$ denote the cumulative arriving traffic from flow j that arrives until time t. Then, with arrival functions $A_1, \ldots, A_N$ for a set of flows, the arrival function of the aggregate is $\sum_{j=1}^{N} A_j(t)$. Determining aggregate arrivals in the space domain is less obvious. Let us use T_A for the arrival time function of the aggregate of a set of flows with arrival time functions $T_{A_1}, \ldots, T_{A_N}$. The time instant $T_A(\nu)$ denotes the earliest time when arrivals from all flows add up to ν bits. Each flow j contributes ν_j bits to this number with $0 \leq \nu_j \leq \nu$, which gives $\nu = \nu_1 + \ldots + \nu_N$. The last arrival of the tuple $(\nu_1, \ldots, \nu_N)$ gets us to the total number of ν, which is $\max_{j=1,\ldots,N} T_{A_j}(\nu_j)$. Since there can be many tuples that add up to ν, the time $T_A(\nu)$ is determined by the tuple that reaches the desired number of ν bits first. This gives us

$$T_A(\nu) = \inf_{\substack{\nu_1,\ldots,\nu_N \\ \nu=\nu_1+\ldots+\nu_N}} \max_{j=1,\ldots,N} T_{A_j}(\nu_j)\,. \tag{3.2}$$

For illustration consider the arrival scenario in Figure 3.4 with two flows. The vertical lines in the figure indicate packet arrivals. Arrivals from Flow 1 and Flow 2 are represented by solid and dashed lines, respectively. We can construct the values of the aggregate arrival time function T_A by visual inspection, where we move along the time axis and collect packet arrivals until the desired number of bits is obtained. For example, the arrival time function for $\nu = 5000$ bits is reached by adding the first six packet arrivals, yielding $T_A(5000) = 0.8$ ms. At this time, the contributions from Flow 1 and Flow 2 are $\nu_1 = \nu_2 = 2500$ bits. There are other values of ν_1 and ν_2 that also result in $\nu = 5000$ bits. We can quickly confirm that, when considering all combinations and computing the time $\max\{T_{A_1}(\nu_1), T_{A_2}(\nu_2)\}$, 0.8 ms is the earliest time that satisfies $\nu_1 + \nu_2 = 5000$.

Remark: If the arrival time functions $T_{A_1}, \ldots, T_{A_N}$ are continuous and strictly increasing, there exists for each ν a tuple $(\nu_1^*, \ldots, \nu_N^*)$ with $\nu = \nu_1^* + \ldots + \nu_N^*$ such that

$$T_A(\nu) = T_{A_1}(\nu_1^*) = \ldots = T_{A_N}(\nu_N^*)\,.$$

In other words, there exists a tuple that attains the infimum and that equalizes the contributing arrival time functions. Since arrival time functions are in general right-continuous and non-decreasing, a minimizing tuple may not exist. Here, we get the weaker property that, for each ν, there exist $\nu_1, \ldots, \nu_N$ with $\nu = \nu_1 + \ldots + \nu_N$ such that

$$\begin{aligned} &\exists j \in \{1, \ldots, N\} : \\ &\quad T_{A_j}(\nu_j) = T_A(\nu) \text{ and } \forall i \neq j : T_{A_i}(\nu_i^-) \leq T_A(\nu) \leq T_{A_i}(\nu_i)\,. \end{aligned} \tag{3.3}$$

4

Max-Plus Algebra of Functions

We next present an algebra, referred to as *max-plus algebra*, which contains the max-plus convolution in (2.8) as one of its operations. The algebra is built around the replacement of the conventional addition and multiplication operations on the real line, respectively, by a maximum and an addition. For example, for the expression $4 \times 1 + 3 \times 3 = 13$ the replacement results in $(4+1) \vee (3+3) = \max\{4+1, 3+3\} = 6$, where we use the symbol '$\vee$' to indicate the maximum operation. Both operations have neutral elements, $-\infty$ for the maximum ($a \vee -\infty = a$) and 0 for the addition ($a + 0 = a$), and $-\infty$ is an absorbing element for the second operation ($a + (-\infty) = -\infty$). Clearly, the operations $\vee$ and $+$ are commutative and associative. Also, the $+$-operation distributes over the $\vee$-operation. A major difference to the conventional algebra with addition and multiplication is that the maximum is idempotent ($a \vee a = a$) and that there are no inverse elements for the maximum operation. An algebra induced by the minimum and addition operations is referred to as a *dioid* or *idempotent semiring* [3]. The dioids are $(\mathbb{Z} \cup -\infty \cup \infty, \vee, +)$ for integer numbers and $(\mathbb{R} \cup -\infty \cup \infty, \vee, +)$ for real numbers.[1]

[1] The literature on semiring and dioid algebras [3, 21] does not always assume that the dioid operations are commutative. If they are, they are referred to as *commutative dioids*. A further distinction is that of a *complete dioid*, which requires that the first operation, the maximum in our case, is closed for an infinite number of elements, and that the distribution law extends to infinitely many elements. Obviously, the dioids $(\mathbb{Z}\cup-\infty\cup\infty, \vee, +)$ and $(\mathbb{R}\cup-\infty\cup\infty, \vee, +)$ satisfy these properties.

We are interested in applying the max-plus algebra to functions, such as the arrival time functions T_A. A common characteristic of these functions is that they are non-decreasing, that is, $T_A(\nu + \kappa) \geq T_A(\nu)$ for all $\kappa \geq 0$. In a continuous-time domain, we always assume that functions are right-continuous. Since the argument of functions in the space domain designates a quantity of data, only non-negative arguments are meaningful. Nonetheless, we define functions over the entire real axis. We consider non-decreasing functions $F : \mathbb{R} \to \mathbb{R}_o^+ \cup \{\infty\} \cup \{-\infty\}$, where function values are timestamps. Timestamps are non-negative real numbers, with two exceptions. The value $F(\nu) = \infty$ is interpreted as 'will never occur in the future', and a timestamp of $F(\nu) = -\infty$ is interpreted as 'has never occurred in the past'. Consistent with this reasoning, we set $T_A(\nu) = -\infty$ for $\nu < 0$. If we run into a situation where we have to add infinite values with different signs, we use the convention that $-\infty + \infty = -\infty$.

We consider two classes of continuous-space functions. We use $\mathcal{T}$ to denote the set of right-continuous non-decreasing functions that are defined over the entire real axis and take values in $\mathbb{R} \cup \{\infty\} \cup \{-\infty\}$. Functions that evaluate to $-\infty$ on the negative half axis ($F(\nu) = -\infty$ for $\nu < 0$) and non-negative on the positive half axis ($F(\nu) \geq 0$ for $\nu \geq 0$) are referred to as *one-sided* functions. We define $\mathcal{T}_o$ as the set of functions $F \in \mathcal{T}$ that are one-sided. The same classes of functions can be defined in a discrete-time domain for functions $F : \mathbb{Z} \to \mathbb{R}_o^+ \cup \{\infty\} \cup \{-\infty\}$.

The max-plus algebra of functions in $\mathcal{T}$ and $\mathcal{T}_o$ involves the following three operations for two functions F and G:

- **Maximum:** $F \vee G(\nu) = \max\{F(\nu), G(\nu)\}$,
- **Max-plus convolution:** $F \overline{\otimes} G(\nu) = \sup_{\kappa \in \mathbb{R}} \{F(\kappa) + G(\nu - \kappa)\}$,
- **Max-plus deconvolution:** $F \overline{\oslash} G(\nu) = \inf_{\kappa \in \mathbb{R}} \{F(\nu + \kappa) - G(\kappa)\}$.

We can replace a supremum by a maximum, when the supremum is taken over a finite range. If $F, G \in \mathcal{T}_o$, we can limit the ranges in the computation of the max-plus convolution and deconvolution by

$$F \overline{\otimes} G(\nu) = \sup_{0 \leq \kappa \leq \nu} \{F(\kappa) + G(\nu - \kappa)\} \ ,$$
$$F \overline{\oslash} G(\nu) = \inf_{\kappa \geq 0} \{F(\nu + \kappa) - G(\kappa)\} \, .$$

For the convolution on $\mathcal{T}_o$, the range can be limited since $F(\kappa) = -\infty$ for $\kappa < 0$ and $F(\nu - \kappa) = -\infty$ for $\kappa < \nu$. For the deconvolution on $\mathcal{T}_o$, we have that $F(\nu + \kappa) - G(\kappa) = \infty$ for all $\kappa < 0$. Note that we used the limited range of the convolution for functions in $\mathcal{T}_o$ when we first introduced the max-plus convolution in §2.

We define the function $\overline{\delta}$ as

$$\overline{\delta}(\nu) = \begin{cases} -\infty\,, & \nu < 0\,, \\ 0\,, & \nu \geq 0\,, \end{cases}$$

and we define $\overline{\delta}_T(\nu) = \overline{\delta}(\nu) + T$ for any $T > 0$. The following lemmas list properties of the max-plus convolution and the max-plus deconvolution operation. We write '$F \,\overline{\otimes}\, G$' to mean '$F \,\overline{\otimes}\, G(\nu)$ for all $\nu \in \mathbb{R}$'. Also, we write '$F \leq G$' to mean '$F(\nu) \leq G(\nu)$ for all $\nu \in \mathbb{R}$'.

Lemma 4.1 (PROPERTIES OF THE MAX-PLUS CONVOLUTION)**.** Consider functions $F, G, H \in \mathcal{T}$.

(a) **Closure.** If F and G lie in $\mathcal{T}$ (or $\mathcal{T}_o$), so does $F \,\overline{\otimes}\, G$.

(b) **Associativity.** $(F \,\overline{\otimes}\, G) \,\overline{\otimes}\, H = F \,\overline{\otimes}\, (G \,\overline{\otimes}\, H)$.

(c) **Commutativity.** $F \,\overline{\otimes}\, G = G \,\overline{\otimes}\, F$.

(d) **Distributivity.** $(F \vee G) \,\overline{\otimes}\, H = (F \,\overline{\otimes}\, H) \vee (G \,\overline{\otimes}\, H)$.

(e) **Neutral element.** $F \,\overline{\otimes}\, \overline{\delta} = F$.

(f) **Time shift.** For each $T > 0$, $F \,\overline{\otimes}\, \overline{\delta}_T(\nu) = F(\nu) + T$.

(g) **Monotonicity.** If $F \leq G$ then $F \,\overline{\otimes}\, H \leq G \,\overline{\otimes}\, H$.

(h) **Boundedness.** $F \,\overline{\otimes}\, G \geq F$ for all $F \in \mathcal{T}$ and $G \in \mathcal{T}_o$. In particular, if $F \in \mathcal{T}_o$ then $F \,\overline{\otimes}\, F \geq F$.

(i) **Existence of maximum.** For $F, G \in \mathcal{T}_o$, for each value of ν, there exists a μ^* with $0 \leq \mu^* \leq \nu$, such that $F \,\overline{\otimes}\, G(\nu) = F(\mu^*) + G(\nu - \mu^*)$.

These properties establish that $(\mathcal{T}, \vee, \overline{\otimes})$ and $(\mathcal{T}_o, \vee, \overline{\otimes})$ are dioids, for both discrete-space and continuous-space functions. The closure property confirms that the max-plus convolution preserves the properties of non-decreasing and one-sided function. Associativity is useful since the outcome

of a sequence of max-plus convolutions does not depend on the order in which the max-plus convolution operator is applied. In practice this means that we can drop parentheses, and simply write $F \overline{\otimes} G \overline{\otimes} H$. A consequence of commutativity is that we can rearrange the order in a sequence of min-plus convolutions, *e.g.*, arranging $F_1 \overline{\otimes} F_2 \overline{\otimes} \ldots \overline{\otimes} F_N$ as $F_N \overline{\otimes} F_{N-1} \overline{\otimes} \ldots \overline{\otimes} F_1$ without altering the result.
The max-plus deconvolution is not endowed with the same nice properties as the convolution.

Lemma 4.2 (PROPERTIES OF THE MAX-PLUS DECONVOLUTION)**.** Consider functions $F, G, H \in \mathcal{T}_o$.

(a) **Not closed.** There exist F and G such that $F \overline{\oslash} G \notin \mathcal{T}_o$ or even $F \overline{\oslash} G \notin \mathcal{T}$.

(b) **Not associative.** There exist F, G, and H such that $(F \overline{\oslash} G) \overline{\oslash} H \neq F \overline{\oslash} (G \overline{\oslash} H)$.

(c) **Not commutative.** There exist F and G such that $F \overline{\oslash} G \neq G \overline{\oslash} F$.

(d) **Composition of $\overline{\otimes}$ and $\overline{\oslash}$.** $(F \overline{\oslash} G) \overline{\oslash} H = F \overline{\oslash} (G \overline{\otimes} H)$.

(e) **Duality.** $F \geq G \overline{\otimes} H$ if and only if $F \overline{\oslash} H \geq G$.

The proofs of Lemmas 4.1 and 4.2 are found in the appendix.

We next discuss superadditive functions, which play a special role in the max-plus algebra. A function F is said to be *superadditive* if for all $\mu, \nu \in \mathbb{R}$,

$$F(\mu + \nu) \geq F(\mu) + F(\nu) \,. \tag{4.1}$$

We generally assume that superadditive functions lie in the set $\mathcal{T}_o$. Define

$$\begin{aligned} F^{\overline{(0)}} &= \overline{\delta} \,, \\ F^{\overline{(n)}} &= F^{\overline{(n-1)}} \overline{\otimes} F \,. \end{aligned}$$

Then, the superadditive closure of F, denoted by $F^{\overline{*}}$, is defined as

$$F^{\overline{*}} = \lim_{n \to \infty} \left(\overline{\delta} \vee F \vee F^{\overline{(2)}} \vee \cdots \vee F^{\overline{(n)}} \right) = \sup_{n > 0} F^{\overline{(n)}} \,.$$

We immediately see that $F^{\overline{*}} \geq F$. To show that the superadditive closure deserves its name, *i.e.*, that $F^{\overline{*}}$ is superadditive, we write

$$\begin{aligned}
F^{\overline{*}}(\nu+\mu) &= \sup_{k+m>0} F^{\overline{(k+m)}}(\nu+\mu) \\
&= \sup_{k+m>0} (F^{\overline{(k)}} \,\overline{\otimes}\, F^{\overline{(m)}})(\nu+\mu) \\
&= \sup_{k+m>0} \sup_{\kappa\in\mathbb{R}} \left\{ F^{\overline{(k)}}(\kappa) + F^{\overline{(m)}}(\nu+\mu-\kappa) \right\} \\
&\geq \sup_{k+m>0} \left\{ F^{\overline{(k)}}(\mu) + F^{\overline{(m)}}(\nu) \right\} \\
&= \sup_{k>0} F^{\overline{(k)}}(\mu) + \sup_{m>0} F^{(m)}(\nu) \\
&= F^{\overline{*}}(\mu) + F^{\overline{*}}(\nu)\,.
\end{aligned}$$

In the fourth line, we have restricted the inner supremum to $\kappa = \mu$. If $F \in \mathcal{T}_o$, then the superadditive closure is simplified to

$$F^{\overline{*}} = \lim_{n\to\infty} F^{\overline{(n)}}\,.$$

In the max-plus algebra, superadditive functions have a number of unique properties, as given by the next lemma.

Lemma 4.3. For a function $F \in \mathcal{T}_o$, the following four statements are equivalent:

(a) F is superadditive.

(b) $F = F \,\overline{\otimes}\, F$.

(c) $F = F \,\overline{\oslash}\, F$.

(d) $F = F^{\overline{*}}$.

Proof.
$(a) \Rightarrow (b)$: Since $F \geq \overline{\delta}$, the monotonicity of $\overline{\otimes}$ gives

$$F \,\overline{\otimes}\, F \geq F \,\overline{\otimes}\, \overline{\delta} = F\,.$$

Next, by substituting ν for $\nu+\mu$ in (4.1) we get

$$F(\nu) \geq F(\nu-\mu) + F(\mu)\,, \quad \forall \nu, \mu \in \mathbb{R}\,,$$

which is equivalent to

$$F(\nu) \geq \sup_{\mu \in \mathbb{R}} \{F(\nu - \mu) + F(\mu)\} = F \,\overline{\otimes}\, F(\nu)\,, \quad \forall \nu \in \mathbb{R}\,.$$

$(b) \Rightarrow (d)$: Since $F \in \mathcal{T}_o$, we have $F^{\overline{*}} = \lim_{n\to\infty} F^{\overline{(n)}}$. With $F = F \,\overline{\otimes}\, F$ we get that $F^{\overline{*}} = F$.

$(d) \Rightarrow (a)$: This follows from the fact that the superadditive closure is superadditive.

$(a) \Rightarrow (c)$: Every function $F \in \mathcal{T}_o$ satisfies

$$\begin{aligned} F \,\overline{\oslash}\, F(\nu) &= \inf_{\kappa \geq 0} \{F(\nu + \kappa) - F(\kappa)\} \\ &\leq F(\nu) - F(0) \\ &\leq F(\nu)\,, \end{aligned}$$

where we set $\kappa = 0$ in the second line, and used that $F(0) \geq 0$ in the third line. If F is superadditive, we can also write

$$\begin{aligned} F \,\overline{\oslash}\, F(\nu) &= \inf_{\kappa \geq 0} \{F(\nu + \kappa) - F(\kappa)\} \\ &\geq \inf_{\kappa \geq 0} F(\nu) \\ &= F(\nu)\,, \end{aligned}$$

where we use the superadditivity of F in the second line. Hence, a superadditive function $F \in \mathcal{T}_o$ satisfies $F = F \,\overline{\oslash}\, F$.

$(c) \Rightarrow (a)$: $F \,\overline{\oslash}\, F = F$ means that for all $\nu \in \mathbb{R}$ we have

$$F(\nu + \kappa) - F(\kappa) \geq F(\nu)\,, \quad \forall \kappa \geq 0\,,$$

so the superadditive property is satisfied for $\kappa \geq 0$. For $\kappa < 0$ we note that $F(\kappa) = -\infty$. In this case, if $\nu + \kappa \geq 0$ we have

$$F(\nu + \kappa) \geq 0 \quad \text{and} \quad F(\nu) + F(\kappa) = -\infty\,,$$

and if $\nu + \kappa < 0$, we have

$$F(\nu + \kappa) = -\infty \quad \text{and} \quad F(\nu) + F(\kappa) = -\infty\,.$$

In both cases, $F(\nu + \kappa) \geq F(\nu) + F(\kappa)$. Hence, the superadditive property holds for every $\kappa \in \mathbb{R}$.

□

5

Backlog and Delay in the Space Domain

We are interested in observing and analyzing network traffic that passes through one or more network devices. We study the amount of traffic that resides in a network device at any time, which is referred to as *backlog*, as well as the time duration that traffic resides in a device, referred to as *delay*. The concept of a *network element* serves as a general model for anything that can impose delay on network traffic, such as a packet switch, a buffered link, a network cable, or any combination thereof. In Figure 5.1, we show arrivals and departures to a network element. In the figure, γ_S refers to a function that characterizes the service offered by the network element. We refer to the analysis of backlog, delays, and other performance metrics at a network element in a max-plus algebra as *max-plus network calculus*. We next develop its building blocks and main results.

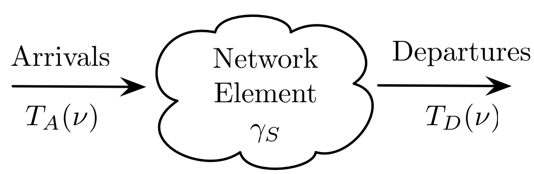


Figure 5.1: Network element with arrivals and departures.

Arrivals at a network element are described in terms of an arrival time function $T_A \in \mathcal{T}_o$. We allow T_A to be a discrete-space or continuous-space function, but our discussion generally assumes that T_A is right-continuous. The departures at a system are given by $T_D \in \mathcal{T}_o$, where we have $T_D(\nu) \geq T_A(\nu)$, that is, a bit cannot depart earlier than its arrival time. The delay at a network element is expressed in terms of the difference between arrival and departure times, that is, for all $\nu \geq 0$,

$$W(\nu) = T_D(\nu) - T_A(\nu) \, . \tag{5.1}$$

$W(\nu)$ provides the delay of the ν-th bit, where we allow the bit value to be a real number. The backlog describes the number of bits that have arrived, but not yet departed. If expressed as a function of bits, there is a choice whether we evaluate the backlog at the time when a bit arrives at or when it departs from the network element. Denoting by $B^a(\nu)$ and $B^d(\nu)$, respectively, the backlog when the ν-th bit arrives and departs, we have

$$\begin{aligned} B^a(\nu) &= \inf \left\{ \kappa > 0 \mid T_D(\nu - \kappa) \leq T_A(\nu) \right\} \, , \\ B^d(\nu) &= \inf \left\{ \kappa > 0 \mid T_A(\nu + \kappa) \geq T_D(\nu) \right\} \, . \end{aligned} \tag{5.2}$$

We refer to the two quantities as the *arrival backlog* and the *departure backlog*. Figure 5.2 illustrates the arrival time function T_A and the departure time function T_D for the arrival scenario from Table 3.1 at a buffered link that operates at 4 Mbps. The figure shows the delay $W(\nu)$ and the backlog $B(\nu)$ for a value of ν that is somewhere between 1000 and 2000. The delay is simply the vertical distance between the departure time function and the arrival time function. When T_A and T_D are strictly increasing, then $B^a(\nu)$ and $B^d(\nu)$ are the horizontal distances between T_D and T_A, where, for $B^a(\nu)$, the horizontal distance is measured at $T_A(\nu)$ and for $B^d(\nu)$, the distance is measured at $T_D(\nu)$. When T_A and T_D are not strictly increasing, $B^a(\nu)$ is the horizontal distance between the point $(\nu, T_A(\nu))$ and the curve T_D, and $B^d(\nu)$ is the horizontal distance between the point $(\nu, T_D(\nu))$ and the curve T_A.

We define a *busy sequence* to be a maximal contiguous set of bits that experience non-zero delays. That is, an interval $I = [\kappa, \mu)$ is a busy sequence if $W(\nu) > 0$ for $\kappa < \nu < \mu$ and $W(\kappa^-) = W(\mu) = 0$, where $W(\kappa^-)$ is as defined in (3.1). An *idle sequence* is a maximal contiguous set of bits that experience no delay. Since inserting $T_A(\nu) = T_D(\nu)$ in the definitions of the

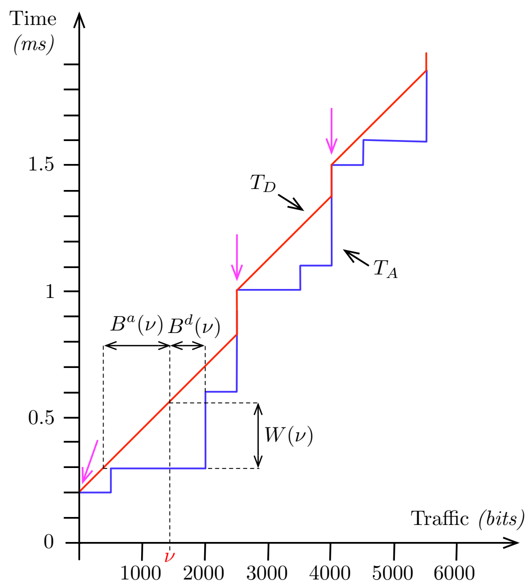


Figure 5.2: Arrival and departure time functions at a buffered link.

arrival and departure backlog gives

$$W(\nu) = 0 \quad \Longrightarrow \quad B^a(\nu) = 0 \quad \text{and} \quad B^d(\nu) = 0\,,$$

we see that the bits of an idle sequence experience no arrival or departure backlog. Note that the converses do not hold. For example, an arrival ν to an element without backlog ($B^a(\nu) = 0$) may experience a delay, *e.g.*, when the network element imposes a fixed delay on all traffic. Also, a bit that leaves no backlog behind when it departs ($B^d(\nu) = 0$) may have been delayed at the network element.

In some derivations, we need to find the beginning of a busy sequence with respect to a value $\nu \geq 0$. For this we define

$$\underline{\nu} = \sup\{\kappa \mid 0 \leq \kappa \leq \nu\,, W(\kappa) = 0\}\,. \tag{5.3}$$

The beginning of the busy sequence is expressed as the least upper bound on the bit value $\leq \nu$ that experiences no delay. Note that the supremum in the definition of $\underline{\nu}$ accounts for two cases. One case is that $\underline{\nu}$ does not experience any delay ($W(\underline{\nu}) = 0$). The second case is that $W(\underline{\nu}) > 0$ and $W(\underline{\nu}^-) = 0$, which allows the delay at the beginning of a busy sequence to be positive, as long as it is preceded by a value with no delay. Here, the supremum $\underline{\nu}$ is not in the set over which the supremum is computed. If there is no previous idle sequence, that is, ν is in a busy sequence that starts at zero ($W(\kappa) > 0$ for all $\kappa \in [0, \nu]$), the supremum falls back to the lower bound of the interval and we obtain that $\underline{\nu} = 0$. If ν is in an idle sequence ($W(\nu) = 0$), then $\underline{\nu} = \nu$. In Figure 5.2 we have indicated the beginning of the busy sequences for the arrival scenario with downward pointing arrows. Observe that the beginning of a busy sequence does not experience a delay. While this is always the case for the buffered link, it may not hold true for other network elements. We will encounter such scenarios in §11.5. (There, in Figure 11.3(a), the beginning of the busy sequences at ν_3 and ν_5 have a non-zero delay.)

With the definition of the beginning of a busy sequence at hand, we can prove that the departure time function at a work-conserving link with rate C, as given in (2.8), applies to arbitrary values of ν. Recall that the convolution expression in (2.8) for packetized arrivals was obtained only for $T_D(L_1^-), T_D(L_2^-), \ldots$, that is, for the departure times of the end of each packet.

Lemma 5.1. Given a right-continuous arrival time function T_A, the departure time function at a work-conserving buffered link with rate $C > 0$ is given for all $\nu \geq 0$ by

$$T_D(\nu) = \sup_{0 \leq \kappa \leq \nu} \left\{ T_A(\kappa) + \frac{\nu - k}{C} \right\}.$$

This expression is of course equal to $T_D = T_A \overline{\otimes} \gamma_S$ with $\gamma_S(\nu) = \frac{\nu}{C}$.

Proof. Pick two arbitrary values κ and ν with $0 \leq \kappa \leq \nu$. Since traffic is served in a FIFO fashion, the departure time for bit value ν cannot be earlier than the arrival time for κ plus the service time of $\nu - \kappa$ bits. This gives

$$T_D(\nu) \geq T_A(\kappa) + \frac{\nu - k}{C}.$$

Since the above relationship holds for all values of κ, it holds for the supremum, yielding

$$T_D(\nu) \geq \sup_{0 \leq \kappa \leq \nu} \left\{ T_A(\kappa) + \frac{\nu - \kappa}{C} \right\}.$$

To show that the inequality above also holds for the other direction, we need to use the work-conserving property of the buffered link, which states that the link generates output at rate C at any time where there is a non-zero backlog. Note that this concept of backlog is defined as a function of time, which is not captured by the space-domain backlog expressions $B^a(\nu)$ or $B^d(\nu)$. Using $B(t)$ to denote the backlog at time t, we can establish a relationship between the time-domain backlog and the space-domain delay. (In the min-plus network calculus, where $A(t)$ and $D(t)$ denote the cumulative arrivals in the time interval $[0, t)$, we have $B(t) = A(t) - D(t)$.) Since the departure time of a bit at a work-conserving link with rate C depends on the backlog found upon its arrival, we have

$$T_D(\nu) = T_A(\nu) + \frac{B(T_A(\nu))}{C} \,,$$

which we can rewrite as

$$W(\nu) = \frac{B(T_A(\nu))}{C} \,.$$

This lets us conclude

$$W(\nu) > 0 \quad \Longleftrightarrow \quad B(T_A(\nu)) > 0 \,. \tag{5.4}$$

Let $\underline{\nu}$ be defined as in (5.3). Suppose that $W(\nu) > 0$. Then, from the relationship between $W(\nu)$ and $B(t)$ we can infer that $B(t) > 0$ for $t \in [T_A(\underline{\nu}+\varepsilon), T_A(\nu))$ for any $\varepsilon > 0$ with $\underline{\nu}+\varepsilon < \nu$. We also have $B(t) > 0$ for $t \in [T_A(\nu), T_D(\nu))$ since ν is part of the backlog until it departs. Therefore, in the interval $[T_A(\underline{\nu} + \varepsilon), T_D(\nu))$, the link transmits continuously at rate C with a total output of $\nu - \underline{\nu} + \varepsilon$ bits. This means that the departure time of ν is

$$T_D(\nu) = T_A(\underline{\nu} + \varepsilon) + \frac{\nu - \underline{\nu} - \varepsilon}{C} \,.$$

If $W(\nu) = 0$, we have $T_D(\nu) = T_A(\nu)$. Hence, we get the inequality

$$T_D(\nu) \le \sup_{0 \le \kappa \le \nu} \Big\{ T_A(\kappa) + \frac{\nu - \kappa}{C} \Big\} \,.$$

$\square$

Using (5.1), we obtain an exact expression for the delay, given by

$$W(\nu) = \sup_{0 \le \kappa \le \nu} \Big\{ \frac{\nu - k}{C} - \big(T_A(\nu) - T_A(\kappa)\big) \Big\} \,. \tag{5.5}$$

Discrete-space considerations. When traffic is represented by discrete-space functions, departure time functions and busy sequences may not fully coincide with the corresponding continuous-space version. To see this, consider a discrete-space arrival time function $T_A(\nu)$ with $\nu = 0, 1, 2, \ldots$ at a buffered link with rate C. Since the service time of a single bit is $\frac{1}{C}$, we get $T_D(\nu) \geq T_A(\nu) + \frac{1}{C}$. As a consequence, we have $T_D(\nu) > T_A(\nu)$ for every value of ν, and $W(\nu) > 0$ (and $B^a(\nu) > 0$) for all ν. Hence, a discrete-space representation does not comply with the definition of a busy sequence in (5.3). Another consequence is that the discrete-space and continuous-space departure time functions at the buffered link have the relationship

$$T_D^{\text{discrete}}(\nu) = T_D^{\text{continuous}}(\nu) + \frac{1}{C}\,.$$

On the other hand, at network elements where the service time does not depend on the amount of data being serviced, the discrete-space and continuous-space departure time functions are identical. An example of this is a network element that imposes a fixed delay, with $T_D(\nu) = T_A(\nu) + T$ for some delay value $T > 0$, where

$$T_D^{\text{discrete}}(\nu) = T_D^{\text{continuous}}(\nu)$$

holds. This discrepancy between continuous and discrete domains does not exist in the min-plus network calculus, where arrivals and departures are characterized by discrete-time or continuous-time functions. This means that, in the max-plus framework, we have to be more careful about distinguishing a discretized or continuous analysis. It should be noted that the difference between continuous-space and discrete-space descriptions are limited. In the case of the buffered link, the difference is the transmission time of a single bit in a bit-level description. As seen in §2, the difference shrinks when traffic is measured in smaller units.

6

Max-Plus Traffic Envelopes and Traffic Regulators

We next explore space-invariant characterizations of traffic that bound the traffic of a flow in terms of traffic envelopes. In the max-plus network calculus, traffic envelopes describe the shortest time interval that must elapse for a given amount of traffic.

Definition 6.1. A function λ_E is a *max-plus traffic envelope* for an arrival time function T_A, if for all $\nu \geq 0$ and $\mu \geq 0$,

$$\lambda_E(\mu) \leq T_A(\nu + \mu) - T_A(\nu) \,. \tag{6.1}$$

A max-plus traffic envelope specifies that at most μ bits of traffic may arrive within $\lambda_E(\mu)$ time units. We write $T_A \sim \lambda_E$, if λ_E is a max-plus traffic envelope for T_A. Among other properties, every max-plus traffic envelope goes through the origin ($\lambda_E(0) = 0$). A traffic envelope is both space-invariant and time-invariant. Space-invariance refers to the fact that $\lambda_E(\mu)$ bounds the arrival time period of μ consecutive bits, regardless where the bits are located in a traffic flow. By time-invariance we mean that any traffic envelope λ_E for an arrival time function T_A is also a traffic envelope for a time-shifted version of T_A. We can express this as

$$T_A \sim \lambda_E \quad \Longrightarrow \quad T_A + \tau \sim \lambda_E \,, \text{ for } \tau \geq 0 \,,$$

where $T_A + \tau$ (more precisely, $T_A(\nu) + \tau$) shifts the arrival times of T_A by τ time units into the future. For negative arguments $\mu < 0$, we set $\lambda_E(\mu) = -\infty$, which ensures that $\lambda_E \in \mathcal{T}_o$.

Theorem 6.1. An envelope $\lambda_E \in \mathcal{T}_o$ for an arrival time function T_A with $T_A \sim \lambda_E$ satisfies

$$T_A = T_A \,\overline{\otimes}\, \lambda_E \,. \tag{6.2}$$

Proof. First observe that $\lambda_E \in \mathcal{T}_o$ ensures that $\lambda_E \geq \overline{\delta}$. Then, the monotonicity of the max-plus convolution gives

$$T_A \,\overline{\otimes}\, \lambda_E \geq T_A \,\overline{\otimes}\, \overline{\delta} = T_A \,.$$

For the other direction of the inequality, we rewrite (6.1) as

$$\lambda_E(\mu) + T_A(\nu - \mu) \leq T_A(\nu) \,, \qquad \forall \mu \leq \nu \,,$$

which is equivalent to

$$\sup_{0 \leq \mu \leq \nu} \{\lambda_E(\mu) + T_A(\nu - \mu)\} \leq T_A(\nu) \,,$$

and, therefore, $\lambda_E \,\overline{\otimes}\, T_A \leq T_A$. □

If we iterate (6.2), we obtain

$$\begin{aligned} T_A &= T_A \otimes \lambda_E \\ &= T_A \,\overline{\otimes}\, \lambda_E \,\overline{\otimes} \cdots \overline{\otimes}\, \lambda_E \\ &= T_A \,\overline{\otimes}\, \lambda_E^{\overline{*}} \,. \end{aligned}$$

Therefore, if λ_E is a max-plus traffic envelope, so is its superadditive closure $\lambda_E^{\overline{*}}$. The reverse also holds, that is, if $T_A \sim F^{\overline{*}}$ then $T_A \sim F$. This follows from $\lambda_E^{\overline{*}} \geq \lambda_E$. If we compare two max-plus traffic envelopes for the same arrival time function, *e.g.*, $T_A \sim \lambda_E^1$ and $T_A \sim \lambda_E^2$, then λ_E^1 is a better envelope than λ_E^2 if $\lambda_E^1 > \lambda_E^2$. The interpretation is that, λ_E^1 permits a larger time interval for the arrival of the same number of bits.

Since $\lambda_E^{*} \geq \lambda_E$, the superadditive closure $\lambda_E^{\overline{*}}$ is never a worse envelope for T_A than λ_E. Consequently, we can always improve a max-plus envelope by replacing it with its superadditive closure. Put differently, any good choice

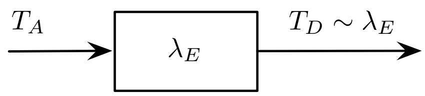


Figure 6.1: Traffic regulator for a max-plus traffic envelope λ_E.

of a max-plus traffic envelope λ_E for an arrival time function A is superadditive, since $\overline{\lambda}_E^* = \lambda_E$ in this case.

Let us address the question of the best achievable traffic envelope for a given arrival time function T_A. If we write (6.1) as $\lambda_E \leq T_A \overline{\oslash} T_A$, the best envelope is the largest function satisfying the inequality. We refer to this envelope as *the empirical max-plus traffic envelope* of T_A, and denote it by $\lambda_A^{\mathcal{E}}$. The empirical envelope is constructed by setting $\lambda_A^{\mathcal{E}} = T_A \overline{\oslash} T_A$. Since no envelope for T_A can be larger than $\lambda_A^{\mathcal{E}}$, we can safely infer that $\lambda_A^{\mathcal{E}}$ is superadditive.

Next we discuss network elements that enforce envelope functions for departing traffic. A traffic regulator for a max-plus traffic envelope λ_E is a network element which ensures that $T_D \sim \lambda_E$ for each arrival time function T_A. Figure 6.1 illustrates a traffic regulator. We consider a type of traffic regulator that buffers non-compliant traffic until it becomes compliant. Such a traffic regulator is called a *greedy shaper*. The attribute ‘greedy’ refers to the fact that non-compliant traffic is released from the buffer as early as allowed by the traffic envelope.

Theorem 6.2. A network element is a greedy shaper for a superadditive max-plus traffic envelope λ_E if and only if for an arrival time function T_A, the departure time function T_D satisfies

$$T_D = T_A \overline{\otimes} \lambda_E .$$

The theorem suggests how to implement a greedy shaper, namely, set the departure time of a bit to the max-plus convolution of the arrival time function and the traffic envelope.

Proof. We first prove $T_D \geq T_A \overline{\otimes} \lambda_E$. By a change of variable $\kappa = \nu + \mu$ we can rewrite (6.1) as

$$\lambda_E(\kappa - \mu) \leq T_A(\kappa) - T_A(\nu) , \qquad \forall \nu, \kappa \text{ with } 0 \leq \mu \leq \kappa .$$

Since $T_D(\nu) \geq T_A(\nu)$ we can rewrite this as

$$T_D(\kappa) \geq T_A(\nu) + \lambda_E(\kappa - \mu)\,, \qquad \forall \nu, \kappa \text{ with } 0 \leq \mu \leq \kappa\ ,$$

which is equivalent to

$$T_D(\kappa) \geq \sup_{0 \leq s \leq t} \{T_A(\nu) + \lambda_E(\kappa - \mu)\} = T_A \,\overline{\otimes}\, \lambda_E(\kappa)\,.$$

Next we show that $T_D \leq T_A \,\overline{\otimes}\, \lambda_E$. Fix a $\nu \geq 0$. Assume for the moment that $W(\nu) = 0$, that is, $T_D(\nu) = T_A(\nu)$. Since $\lambda_E(0) = 0$, we have $T_D(\nu) \leq T_A(\nu) + \lambda_E(0)$, and therefore, we get

$$T_D(\nu) \leq \sup_{0 \leq \mu \leq \nu} \{T_A(\mu) + \lambda_E(\nu - \mu)\}\ .$$

Now we consider $W(\nu) > 0$, that is, ν is delayed. Then, the previous arrivals have not complied to the envelope and the departures have saturated the envelope for an interval $[\eta, \nu]$ with $T_D(\nu) - T_D(\eta) = \lambda_E(\nu - \eta)$. If we let μ be such that the interval where the envelope is saturated is maximal, we get

$$\mu = \inf \{\eta < \nu \,|\, T_D(\nu) - T_D(\eta) = \lambda_E(\nu - \eta)\}\ .$$

We will show that $T_D(\mu) = T_A(\mu)$ must hold. If $\mu = 0$, then the envelope has been saturated in the entire interval $[0, \nu]$. In this case, we can use that every max-plus traffic envelope satisfies $\lambda_E(0) = 0$, and obtain $T_A(0) = T_D(0)$.

If $\mu > 0$, then for any choice of $\kappa < \mu$ we have

$$T_D(\nu) - T_D(\kappa) \geq \lambda_E(\nu - \kappa)\,.$$

Since, by superadditivity of λ_E, it holds that

$$\lambda_E(\nu - \kappa) \geq \lambda_E(\nu - \mu) + \lambda_E(\mu - \kappa)\,,$$

and since $T_D(\nu) - T_D(\mu) = \lambda_E(\nu - \mu)$, we can conclude that

$$T_D(\mu) - T_D(\kappa) \geq \lambda_E(\mu - \kappa)\,.$$

Since this relationship holds for all $\kappa < \mu$, according to the operation of the greedy shaper, we have that μ is not delayed, that is, $T_D(\mu) = T_A(\mu)$. Then we continue with

$$\begin{aligned} T_D(\nu) &= T_D(\mu) + \lambda_E(\nu - \mu) \\ &= T_A(\mu) + \lambda_E(\nu - \mu) \\ &\leq \sup_{0 \leq \kappa \leq \nu} \{T_A(\kappa) + \lambda_E(\nu - \kappa)\} \\ &= T_A \,\overline{\otimes}\, \lambda_E(\nu)\,. \end{aligned}$$

To show that $T_D = T_A \overline{\otimes} \lambda_E$ for a superadditive function λ_E implies $T_D \sim \lambda_E$, we write

$$T_D \otimes \lambda_E = T_A \otimes \lambda_E \otimes \lambda_E = T_A \otimes \lambda_E = T_D \,.$$

The first equality inserts $T_D = T_A \otimes \lambda_E$, the second uses that superadditive functions satisfy $\lambda_E = \lambda_E \otimes \lambda_E$, and the last inequality uses again that $T_D = T_A \otimes \lambda_E$. Then, the claim follows since $T_D = T_D \otimes \lambda_E$ is equivalent to $T_D \sim \lambda_E$.

Finally, we show that a traffic regulator that satisfies $T_D = T_A \overline{\otimes} \lambda_E$ releases the same number of bits at least as soon as any alternative realization. This makes the case that $T_D = T_A \overline{\otimes} \lambda_E$ is a defining property of a greedy shaper. Suppose we have a superadditive traffic envelope λ_E. For a given arrival time function T_A, let T_D^{greedy} denote the departure time function of the traffic regulator for envelope λ_E that satisfies $T_D = T_A \overline{\otimes} \lambda_E$. Let T_D^{alt} denote an alternative realization of a traffic regulator for λ_E. Since both versions enforce the envelope λ_E we can write

$$T_D^{\text{alt}} = T_D^{\text{alt}} \overline{\otimes} \lambda_E \geq T_A \overline{\otimes} \lambda_E = T_D^{\text{greedy}} \,,$$

where the first equality follows from Theorem 6.1, the inequality follows from $T_A \leq T_D^{\text{alt}}$ and the monotonicity of the convolution, and the last equality uses our assumption that the departures are equal to $T_A \overline{\otimes} \lambda_E$. Thus, for any arrival time function T_A, we have $T_D^{\text{alt}} \geq T_D^{\text{greedy}}$, meaning that the departure times T_D^{greedy} cannot be improved. □

Examples:

(a) Consider a simple traffic regulator which enforces that the output has a fixed rate r. This is done with an envelope $\lambda_E(\nu) = \frac{\nu}{r}$ with $r > 0$. If we allow bits to depart e time units earlier than allowed by the fixed rate guarantee, we obtain the envelope

$$\lambda_E(\nu) = \left[\frac{\nu}{r} - e\right]^+ \,, \tag{6.3}$$

where $e > 0$ is the permitted earliness with respect to a fixed rate output. We will refer to e as *earliness allowance*. Later, we will see that this max-plus traffic envelope corresponds to the well-known token bucket traffic regulator with burst size er and rate r.

(b) As a generalization of the previous traffic envelope, suppose that there are two constraints

$$\lambda_{E_1}(\nu) = \left[\frac{\nu}{r_1} - e_1\right]^+ \quad \text{and} \quad \lambda_{E_2}(\nu) = \left[\frac{\nu}{r_2} - e_2\right]^+ ,$$

with $r_1 > r_2$ and $e_1 < e_2$, that must both be satisfied. This can be achieved by the traffic envelope

$$\lambda_E(\nu) = \max\left\{\left[\frac{\nu}{r_1} - e_1\right]^+, \left[\frac{\nu}{r_2} - e_2\right]^+\right\} , \tag{6.4}$$

that is, the latest timestamp computed by either of the two traffic envelopes determines the earliest departure time of a bit.

(c) When traffic arrivals consist of fixed-sized packets of ℓ bits, a fixed rate traffic regulator can be viewed as releasing at most one packet every θ time units. Also allowing a slack for being early by e time units, the resulting max-plus traffic envelope is

$$\lambda_E(\nu) = \left[\left\lfloor\frac{\nu}{\ell}\right\rfloor \theta - e\right]^+ .$$

The above discussion does not touch on the complexity of implementing a traffic regulator. Obviously, computing the max-plus convolution $T_A \overline{\otimes} \lambda_E(\nu)$ for each value of ν can become computationally impractical. As we will see in §12.2, for packet-level arrivals, the max-plus convolution can be computed efficiently if λ_E has the form of (6.3) or (6.4).

7

Service Curves in the Max-Plus Network Calculus

In §2, we saw that the output at a work-conserving buffered link with rate C can be expressed as the max-plus convolution of the arrival time function and the function $\gamma_S(\nu) = \frac{\nu}{C}$, that is, $T_D = T_A \,\overline{\otimes}\, \gamma_S$. A generalization of this relationship to other network elements leads us to the concept of a *max-plus service curve*.

7.1 Max-plus service curves

We start out with the definition of service curves in the max-plus algebra.

Definition 7.1. Consider a network element and an arbitrary arrival time function $T_A \in \mathcal{T}_o$, as shown in Figure 5.1. Let $T_D \in \mathcal{T}_o$ be the departure time function induced by arrivals T_A at the network element. A function $\gamma_S \in \mathcal{T}_o$ is an *exact max-plus service curve* for the network element, if

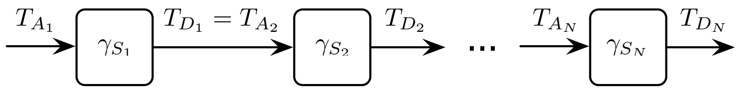
$$T_D = T_A \,\overline{\otimes}\, \gamma_S \,.$$

A *lower max-plus service curve* γ_S satisfies

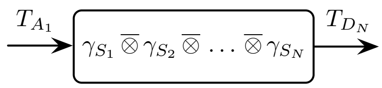
$$T_D \leq T_A \,\overline{\otimes}\, \gamma_S \,,$$

and an *upper max-plus service curve* γ_S satisfies

$$T_D \geq T_A \,\overline{\otimes}\, \gamma_S \,.$$



(a) Sequence of network elements.



(b) Equivalent network element with network service curve.

Figure 7.1: Network service curve.

Max-plus service curves express guarantees offered by a network element. Since the guarantees are expressed in terms of the times of departures, a lower service curve describes a lower bound on the available service at the network element and an upper service curve characterizes an upper bound on the service. An example of an exact service curve is the work-conserving buffered link with rate C, with $\gamma_S(\nu) = \frac{\nu}{C}$ according to Lemma 5.1. Another example is the greedy shaper, with $\gamma_S = \lambda_E$ according to Theorem 6.2.

The max-plus convolution of max-plus service curves results again in a service curve. This can be exploited to compute service guarantees given by a sequence of network elements.

Theorem 7.1. Given a set of N network elements in sequence as shown in Figure 7.1(a), where each element offers a max-plus service curve $\gamma_{S_n} \in \mathcal{T}_o$ ($n = 1, \ldots, N$). Let the arrival and departure time functions of the n-th element be given by T_{A_n} and T_{D_n}, with $T_{D_{n-1}} = T_{A_n}$ for $1 < n \leq N$. Then

$$\gamma^{\text{net}} = \gamma_{S_1} \overline{\otimes} \gamma_{S_2} \overline{\otimes} \ldots \overline{\otimes} \gamma_{S_N}$$

is a max-plus service curve for the entire sequence of network elements.

The service curve γ^{net} is referred to as *network service curve*. The theorem allows us to replace the tandem network in Figure 7.1(a) by a single network element with service curve $\gamma_{S_1} \overline{\otimes} \gamma_{S_2} \overline{\otimes} \ldots \overline{\otimes} \gamma_{S_N}$ as shown in Figure 7.1(b). If the service curves of all elements are exact service curves, the service curve for the entire network is also exact. Likewise for the lower and upper service curves.

Proof. We prove the concatenation property for exact service curves.

$$\begin{aligned} T_{D_N} &= T_{A_N} \overline{\otimes} \gamma_{S_N} \\ &= (T_{A_{N-1}} \overline{\otimes} \gamma_{S_{N-1}}) \overline{\otimes} \gamma_{S_N} \\ &\vdots \qquad\qquad \vdots \\ &= (\ldots (T_{A_1} \overline{\otimes} \gamma_{S_1}) \overline{\otimes} \ldots \overline{\otimes} \gamma_{S_{N-1}}) \overline{\otimes} \gamma_{S_N} \\ &= T_{A_1} \overline{\otimes} (\gamma_{S_1} \overline{\otimes} \ldots \overline{\otimes} \gamma_{S_{N-1}} \overline{\otimes} \gamma_{S_N}) \,. \end{aligned}$$

Here, we repeatedly inserted $T_{D_n} = T_{A_n} \overline{\otimes} \gamma_{S_n}$ and replaced $T_{D_{n-1}} = T_{A_n}$. In the last line, we moved parentheses, which is permitted since the convolution operation is associative. □

As we have seen in the example of the buffered link, a network element with a rate guarantee of R bits per second can be expressed by the service curve

$$\gamma_{S_1}(\nu) = \frac{\nu}{R} \,.$$

A delay server which guarantees a (minimum, maximum, or exact) delay of T time units can be expressed with the service curve

$$\gamma_{S_2}(\nu) = T \,,$$

which follows immediately from

$$T_A \overline{\otimes} \gamma_{S_2}(\nu) = \sup_{0 \le \kappa \le \nu} \{T_A(\nu - \kappa) + T\} = T_A(\nu) + T \,.$$

A combined rate and delay guarantee is given by the service curve

$$\gamma_{S_3}(\nu) = \frac{\nu}{R} + T \,,$$

which results from the max-plus convolution of a rate server with rate R and a delay server with delay T, that is, $\gamma_{S_3} = \gamma_{S_1} \overline{\otimes} \gamma_{S_2}$. Such a network element is referred to as a *latency-rate server*. In the max-plus algebra, the latency-rate service curve takes the form of an affine function.

7.2 Residual max-plus service curves

At a work-conserving buffered link with a fixed rate C that receives arrivals from multiple traffic flows, the *residual service* or *leftover service* expresses

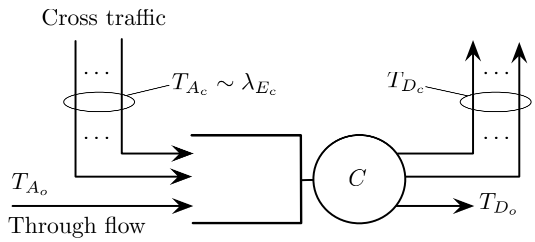


Figure 7.2: Buffered link with through flow and cross traffic.

the capacity that is available to one of these flows. When evaluating the residual service for a flow, that flow is referred to as the *through flow*, and the aggregate of all other flows are summarily referred to as *cross traffic*. See Figure 7.2 for an illustration. If cross traffic arrivals are bounded by a max-plus traffic envelope λ_c, we can characterize the residual service of the through flow by a max-plus service curve.

With multiple flows at a buffered link, there is a need for a scheduling algorithm that selects backlogged traffic for transmission. We will analyze the residual service of the through flow without assuming that we know which scheduling algorithm is operating at the link. Such an analysis is referred to as *blind-to-multiplexing* or *blind multiplexing*. The residual service computed under blind multiplexing can be refined when the specifics of the scheduling algorithm are taken into consideration.

Theorem 7.2 (RESIDUAL SERVICE CURVE)**.** Given a work-conserving buffered link with a fixed rate C with a through flow and cross traffic. If λ_c is a max-plus traffic envelope for the cross traffic, then

$$\gamma_S(\nu) = \frac{1}{C}\left(\inf\Big\{\mu \geq 0 \mid \lambda_c(\mu) \geq \frac{\nu+\mu}{C}\Big\} + \nu\right)$$

is a max-plus lower service curve for the through flow.

We use T_{A_o} and T_{D_o} to denote the arrival time and departure time functions, respectively, of the through flow, and T_{A_c} and T_{D_c} to describe the corresponding functions for the cross traffic. The aggregate of through flow and cross traffic is denoted by T_A for the arrivals, and by T_D for the departures.

Proof. Consider an arbitrary bit value $\nu_o \geq 0$ of the through flow. Let $\underline{\nu}$ be the beginning of the busy sequence in which ν_o is located. The value $\underline{\nu}$ is composed of $\underline{\nu}_o$ bits from the through flow, and $\underline{\nu}_c$ bits from the cross traffic, that is, $\underline{\nu} = \underline{\nu}_o + \underline{\nu}_c$ with $\nu_o \geq \underline{\nu}_o$. At the beginning of the busy sequence, both the through flow and the cross traffic have no delay, that is, $T_{A_o}(\underline{\nu}_o) = T_{D_o}(\underline{\nu}_o)$, $T_{A_c}(\underline{\nu}_c) = T_{D_c}(\underline{\nu}_c)$, and $T_A(\underline{\nu}) = T_D(\underline{\nu})$. The departure time $T_{D_o}(\nu_o)$ is bounded by

$$T_{D_o}(\nu_o) \leq \\ T_{A_o}(\underline{\nu}_o) + \frac{1}{C}\Big(\nu_o - \underline{\nu}_o + \Big\{\text{Cross traffic arriving in } [T_{A_o}(\underline{\nu}_o), T_{D_o}(\nu_o)]\Big\}\Big) \,.$$

The term $\nu_o - \underline{\nu}_o$ describes the traffic of the through flow that must be transmitted by $T_{D_o}(\nu_o)$. Here enters an assumption that the transmission order of the through traffic occurs in the same order as its arrivals. Such a transmission order is referred to as *locally FIFO*. The term that follows states that all cross traffic arrivals since $T_A(\underline{\nu}_o)$ must have been transmitted by $T_D(\nu_o)$. This ensures that, at time $T_D(\nu)$, there is no cross traffic backlog. Time $T_{D_o}(\nu_o)$ is therefore the first time after $T_{A_o}(\nu_o)$ when all cross traffic arrivals since $T_{A_o}(\underline{\nu}_o)$ as well as $\nu_o - \underline{\nu}_o$ bits from the through flow have been transmitted. Since cross traffic arrivals are bounded by the max-plus envelope λ_c, we can write

$$\Big\{\text{Cross traffic arriving in } [T_{A_o}(\underline{\nu}_o), T_{D_o}(\nu_o)]\Big\} \leq \\ \inf\Big\{\mu \geq 0 \,\Big|\, \lambda_c(\mu) \geq \frac{\nu_o - \underline{\nu}_o + \mu}{C}\Big\} \,.$$

Defining a function μ_c as

$$\mu_c(\nu) = \inf\Big\{\mu \geq 0 \,\Big|\, \lambda_c(\mu) \geq \frac{\nu + \mu}{C}\Big\} \,,$$

we have that $\mu_c(\nu_o - \underline{\nu}_o)$ is an upper bound on the cross traffic that must be transmitted by $T_{D_o}(\nu_o)$. With this, we can write the bound for the departure time $T_D(\nu_o)$ as

$$T_D(\nu_o) \leq T_A(\underline{\nu}_o) + \frac{\nu_o - \underline{\nu}_o + \mu_c(\nu_o - \underline{\nu}_o)}{C} \,. \tag{7.1}$$

If we define a service curve by

$$\gamma_S(\nu) = \frac{\mu_c(\nu) + \nu}{C} \,, \tag{7.2}$$

we obtain from (7.1) that

$$\begin{aligned}
T_D(\nu_o) &\le T_A(\underline{\nu}_o) + \gamma_S(\nu_o - \underline{\nu}_o) \\
&\le \sup_{0 \le \kappa \le \nu_o} \{T_A(\kappa) + \gamma_S(\nu_o - \kappa)\} \\
&= T_A \,\overline{\otimes}\, \gamma_S(\nu_o)\,.
\end{aligned}$$

□

Since the proof of Theorem 7.2 has not used any knowledge of the scheduling algorithm, it is valid for any work-conserving scheduling algorithm. This justifies the attribute 'blind multiplexing' or 'blind-to-multiplexing.' It amounts to viewing the through flow as having a lower priority than the cross traffic.

As another remark, the proof of Theorem 7.2 used $\underline{\nu}_o$ to denote the beginning of the busy sequence in which ν_o is located, but did not provide an explicit expression for it. This can be done by specifying the amount of cross flow traffic that arrives before ν_o as

$$\nu_c^{\nu_o} = \sup\{\kappa \mid T_{A_c}(\kappa) \le T_{A_o}(\nu_o)\}\,.$$

Then, setting $\nu = \nu_o + \nu_c^{\nu_o}$ we can obtain $\underline{\nu}$ from (5.3). The quantities $\underline{\nu}_o$ and $\underline{\nu}_c$ are determined by (3.2) as the values κ and μ that define the infimum in

$$T_A(\underline{\nu}) = \inf_{\substack{\kappa,\mu \\ \underline{\nu}=\kappa+\mu}} \max\{T_{A_o}(\kappa), T_{A_c}(\mu)\}\,.$$

Example: We derive the residual service curve for the cross traffic envelope

$$\lambda_c(\nu) = \left[\frac{\nu}{r} - e\right]^+ .$$

The computation of μ_c yields

$$\begin{aligned}
\mu_c &= \inf\Big\{\mu \ge 0 \,\Big|\, \frac{\nu+\mu}{C} < \Big[\frac{\mu}{r} - e\Big]^+\Big\} \\
&= \inf\Big\{\mu \ge er \,\Big|\, \frac{\nu+\mu}{C} < \frac{\mu}{r} - e\Big\} \\
&= \begin{cases} \dfrac{r\nu + erC}{C-r} & \text{if } C > r\,, \\ \infty & \text{if } C \le r\,. \end{cases}
\end{aligned}$$

We dropped '$[\ldots]^+$' in the second line, since the inequality in the infimum is satisfied only if $\frac{\mu}{r} - e > 0$. With this, the service curve becomes

$$\gamma_S(\nu) = \begin{cases} \dfrac{\nu + er}{C - r} & \text{if } C > r\,, \\ \infty & \text{if } C \leq r\,. \end{cases} \tag{7.3}$$

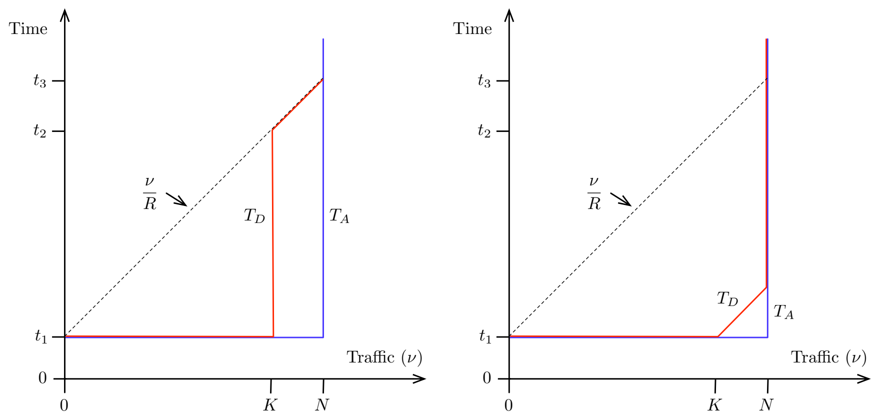
Packetizer in the space domain. A packetizer is a network element that reconstitutes packet-level traffic by storing the bits belonging to the same packet until all data of that packet has arrived, and then releasing all bits of the packet simultaneously. Consider the departure time function T_D of fluid-flow traffic that departs from a network element. We can obtain a packetized departure time function by simply assigning to each bit in a packet the departure time of the last bit in the packet. Using the cumulative packet size function L_n, and noting that the n-th packet covers the range $\nu \in [L_{n-1}, L_n)$, we can express the packetized departure function by a function P^L, defined as

$$P^L(T_D(\nu)) = T_D(L_n^-)\,, \quad \text{for } \nu \in [L_{n-1}, L_n)\,.$$

The packetized departure time of bit ν is simply equal to the departure time of the end of the packet, $T_D(L_n^-)$. Note that the construction of the packet departure times in §2 is consistent with this definition of a packetizer.

7.3 Strict and adaptive max-plus service curves

Sometimes the service guarantees expressed by lower max-plus service curves are too weak. As an example, suppose we want to express a minimum rate guarantee, where a flow can receive an arbitrary amount of available service, with the assurance that the service rate never drops below a guaranteed rate R. If this guarantee is expressed by the lower service curve $\gamma_S(\nu) = \frac{\nu}{R}$, the actual service rate can deviate substantially from a constant rate. This is illustrated in Figure 7.3(a), which shows an arrival time function T_A and a departure time function T_D. The arrivals occur in a single burst of size N at time t_1. The departures are such that an amount $K < N$ is serviced immediately at time t_1. After that, there are no departures in the time interval $[t_1, t_2]$. After t_2, departures occur at a constant rate R, until all traffic is transmitted by time t_3. The departures in Figure 7.3(a) do not resemble at all those of



(a) Lower service curve. (b) Strict service curve.

Figure 7.3: Feasible arrival and departure time functions with service curve $\gamma_S(\nu) = \frac{\nu}{R}$.

a rate guarantee since no traffic is serviced in the interval $[t_1, t_2]$. However, the condition for the lower service curve $T_D(\nu) \leq T_A \overline{\otimes} \gamma_S(\nu)$ is satisfied for each $\nu \in [0, N]$. The reason for this discrepancy is that the rate guarantee offered by a lower max-plus service curve is given for the interval $[0, N]$. Hence, if a flow first has an output rate that is higher than the guaranteed rate, followed by an output rate that is lower than the guarantee, the guarantee may still be satisfied for each value of ν. (Note that an exact service curve $\gamma_S(\nu) = \frac{\nu}{R}$ ensures that the guaranteed rate is available during any time interval. However, an exact service guarantee does not allow a flow to exceed the guaranteed rate R, even if the network element has excess capacity.)

The inability of a lower max-plus service curves to express reasonable rate guarantees is a motivation to refine the service curve definition. One such refinement is a strict service curve.

Definition 7.2. A *strict max-plus service curve* is a function $\gamma_S \in \mathcal{T}_o$ such that for all ν and μ in the same busy sequence ($\underline{\nu} \leq \mu < \nu$), it holds that

$$\begin{aligned} T_D(\nu) - T_D(\mu) &\leq \gamma_S(\nu - \mu)\,, \text{ if } \underline{\nu} < \mu\,, \\ \text{and } \; T_D(\nu) - T_A(\mu) &\leq \gamma_S(\nu - \mu)\,, \text{ if } \underline{\nu} = \mu\,. \end{aligned} \tag{7.4}$$

Here, $\underline{\nu}$ is the beginning of the busy sequence in which ν is located, as defined in (5.3). A strict max-plus service curve makes a guarantee for any sequence of bits in the same busy sequence. If $\underline{\nu} < \mu$, then μ is in the middle of a busy sequence. In this case, a strict service curve bounds the lag between the departure times of ν and μ. If μ is at the beginning of a busy sequence ($\underline{\nu} = \mu$), the departure time of ν is linked to the arrival time of μ. We can view the second condition on the service in (7.4) as anchoring the service guarantees within the busy period to an absolute time. Without the second condition, the departure time of the first bit in a busy sequence could be delayed arbitrarily.

In Figure 7.3(b), we illustrate the departures for a strict service curve $\gamma_S(\nu) = \frac{\nu}{R}$. We consider the same arrival scenario as in Figure 7.3(a). Also, we assume that the initial departures at time t_1 are again equal to K, which exceeds the guaranteed service rate $\frac{\nu}{R}$. Different from the lower service curve, after time t_1, the strict service curve cannot leverage the fact that it has previously transmitted at a higher rate. The first condition in (7.4) enforces that departures continue at least at rate $\frac{\nu}{R}$ until the backlog has been cleared. This results in a departure time function which is at most as large as the function T_D depicted in Figure 7.3(b).

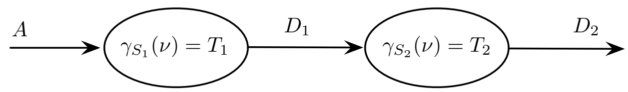
To emphasize the difference between lower and strict service curves, the former are sometimes referred to as *weak service curves*. Every strict max-plus service curve is a lower (or weak) max-plus service curve. To see this consider an arbitrary $\nu > 0$. If $W(\nu) = 0$, that is, $T_A(\nu) = T_D(\nu)$, we obtain

$$\begin{aligned} T_D(\nu) &\le T_A(\nu) + \gamma_S(0) \\ &\le \sup_{0 \le \kappa \le \nu} \{T_A(\kappa) + \gamma_S(\nu - \kappa)\} \\ &= T_A \,\overline{\otimes}\, \gamma_S(\nu)\,. \end{aligned}$$

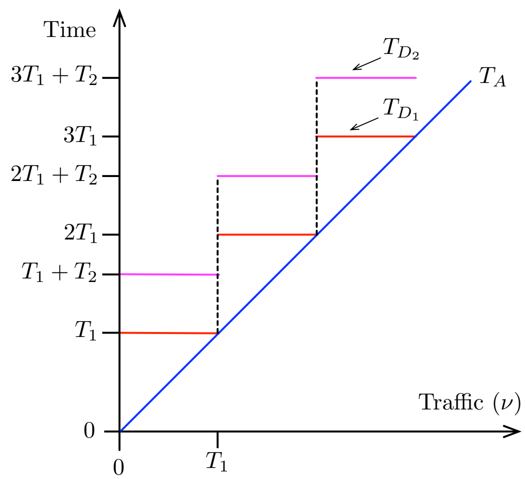
If $W(\nu) > 0$, setting $\mu = \underline{\nu}$ in the second equation of (7.4) with $\underline{\nu}$ as defined in (5.3), we obtain

$$\begin{aligned} T_D(\nu) &\le T_A(\underline{\nu}) + \gamma_S(\nu - \underline{\nu}) \\ &\le \sup_{0 \le \kappa \le \nu} \{T_A(\kappa) + \gamma_S(\nu - \kappa)\} \\ &= T_A \,\overline{\otimes}\, \gamma_S(\nu)\,. \end{aligned}$$

As a remark, strong service curves are always defined as lower bounds on the service. In fact, the only service curve for which (7.4) holds with equality for all ν is the constant-rate service curve $\gamma_S(\nu) = \frac{\nu}{R}$.



(a) Network elements.



(b) Arrival and departure scenario.

Figure 7.4: Sequence of two network elements with strict service curves.

A drawback of strict service curves is that they are not well equipped to express delay guarantees. From the second inequality in (7.4), we can see that a strict max-plus service curve $\gamma_S(\nu) = T$ limits the time period from the first arrival to the last departure in a busy sequence to T. Hence, bits appearing later in a busy sequence must satisfy a shorter delay bound than those at the beginning of a busy sequence. This is not useful for expressing a guarantee that grants all bits the same delay bound.

A second drawback of strict service curves is that the convolution of strict service curves does not maintain strictness. Consider for example a sequence of two network elements as shown in Figure 7.4(a). We give these network elements a strange service policy: They first buffer all arriving traffic and then flush the entire accumulated backlog in one instant after a time limit of T_1

for the first element and after T_2 for the second element, with $T_1 \neq T_2$. The service curves $\gamma_{S_1}(\nu) = T_1$ and $\gamma_{S_2}(\nu) = T_2$, for the first and second network element, respectively, satisfy (7.4) and are therefore strict service curves. For the max-plus convolution, we obtain $\gamma_{S_1} \overline{\otimes} \gamma_{S_2}(\nu) = T_1 + T_2$. Now let us consider an arrival scenario with constant rate arrivals where $T_A(\nu) = \nu$. The arrival time function, and resulting departure time functions T_{D_1} and T_{D_2} are shown in Figure 7.4(b). The end-to-end delay of the entire network is given by $W(\nu) = T_{D_2} - T_A$. The figure illustrates that $W(\nu) > 0$ for all $\nu \geq 0$, that is, the initial busy sequence has infinite length. As a consequence, the service curve $\gamma_{S_1} \overline{\otimes} \gamma_{S_2}(\nu) = T_1 + T_2$ violates the second inequality in (7.4) for $\mu = 0$ and $\nu > T_1$, since $T_{D_2}(T_1 + \kappa) \geq T_1 + T_2$ for all $\kappa > 0$.

The shortcomings of strict service curves can be avoided by revising – actually weakening – the strict service curve guarantee. The revised definition uses a modified max-plus convolution operation for functions $F, G \in \mathcal{T}_o$, given by

$$F \underset{\mu}{\overline{\otimes}} G(\nu) = \sup_{\mu \leq \kappa \leq \nu} \left\{ F(\kappa) + G(\nu - \kappa) \right\}.$$

Definition 7.3. Given a network element with an arrival time function T_A and resulting departure time function T_D. A function $\gamma_S \in \mathcal{T}_o$ is an *adaptive max-plus service curve* for the network element if, for all $\nu \geq 0$,

$$T_D(\nu) \leq \inf_{\mu \leq \nu} \left\{ \max \left[T_D(\mu) + \gamma_S(\nu - \mu) \, , T_A \underset{\mu}{\overline{\otimes}} \gamma_S(\nu) \right] \right\}. \tag{7.5}$$

We can think of the condition as requiring that for all $\mu \leq \nu$ one of the two inequalities

$$T_D(\nu) \leq T_D(\mu) + \gamma_S(\nu - \mu) \quad \text{or} \quad T_D(\nu) \leq T_A \underset{\mu}{\overline{\otimes}} \gamma_S(\nu) \tag{7.6}$$

is satisfied. Note that the first inequality agrees with the first condition for a strict service curve.

By setting $\mu = 0$ in the definition, we see that every adaptive service curve is a lower service curve, since either

$$T_D(\nu) \leq \gamma_S(\nu) \qquad \text{or} \qquad T_D(\nu) \leq T_A \underset{\mu}{\overline{\otimes}} \gamma_S(\nu) \, .$$

Since both $\gamma_S \leq T_A \overline{\otimes} \gamma_S$ and $T_A \underset{\mu}{\overline{\otimes}} \gamma_S \leq T_A \overline{\otimes} \gamma_S$, the bound for a lower service curve is satisfied in both cases.

To see that every strict service curve is an adaptive service curve, pick two values μ and ν in the same busy period with $\underline{\nu} \leq \mu < \nu$. If $\mu > \underline{\nu}$, then the first condition in (7.4) gives that

$$T_D(\nu) \leq T_D(\mu) + \gamma_S(\nu - \mu)\,.$$

If $\mu = \underline{\nu}$ then the second condition in (7.4) gives

$$\begin{aligned} T_D(\nu) &\leq T_A(\underline{\nu}) + \gamma_S(\nu - \underline{\nu}) \\ &\leq T_A \overline{\underset{\mu}{\otimes}} \gamma_S(\nu)\,, \end{aligned}$$

which satisfies the second inequality in (7.6).

In the scenario in Figure 7.3(a), an adaptive service curve $\gamma_S(\nu) = \frac{\nu}{R}$ yields

$$\begin{aligned} T_D(\nu) &\leq \inf_{\mu \leq \nu} \Big\{ \max\Big[T_D(\mu) + \frac{\nu - \mu}{R}\,, \sup_{\mu \leq \kappa \leq \nu} \{ T_A(\kappa) + \frac{\nu - \kappa}{R} \} \Big] \Big\} \\ &= \min\Big(\inf_{\mu < 0} \Big\{ t_1 + \frac{\nu - \mu}{R} \Big\}\,, \\ &\qquad \inf_{0 \leq \mu \leq \nu} \Big\{ \max\Big[T_D(\mu) + \frac{\nu - \mu}{R}\,, t_1 + \frac{\nu - \mu}{R} \Big] \Big\} \Big) \\ &= \min\Big(t_1 + \frac{\nu}{R}, \inf_{0 \leq \mu \leq \nu} \Big\{ T_D(\mu) + \frac{\nu - \mu}{R} \Big\} \Big)\,. \end{aligned} \tag{7.7}$$

In the second line, we split the infimum into two parts, $\mu < 0$ and $0 \leq \mu \leq \nu$. The first infimum has only one term since $T_D(\mu) + \frac{\nu - \mu}{R} = -\infty$ for $\mu < 0$. Also, since all arrivals occur at time t_1, we set $T_A(\kappa) = t_1$ for all $\kappa \geq 0$, and obtain

$$\sup_{\mu \leq \kappa \leq \nu} \{ T_A(\kappa) + \frac{\nu - \kappa}{R} \Big\} = \sup_{\mu \leq \kappa \leq \nu} \Big\{ t_1 + \frac{\nu - \kappa}{R} \Big\} = t_1 + \frac{\nu - \mu}{R}\,.$$

In the third line, we use that $T_D(\mu) \geq t_1$ for $\mu \geq 0$. If we write the infimum in (7.7) with an existential quantifier, we see that (7.7) expresses the two conditions

$$\begin{aligned} & T_D(\nu) - t_1 \leq \frac{\nu}{R}\,, \\ & T_D(\nu) - T_D(\mu) \leq \frac{\nu - \mu}{R}\,, \qquad \forall \mu, 0 \leq \mu \leq \nu\,, \end{aligned}$$

which are the same conditions as those of the strict service curve in Definition 7.2 for the given arrival scenario and service curve $\gamma_S(\nu) = \frac{\nu}{R}$ with

$\underline{\nu} = 0$ and $T_A(0) = t_1$. Hence, an adaptive rate guarantee provides the same bounds on the departures as a strict rate guarantee. Generally, however, adaptive service curves provide a weaker service guarantee than strict service curves. For example, for the adaptive service curve $\gamma_S(\nu) = T$, equation (7.5) evaluates to

$$\begin{aligned} T_D(\nu) &\leq \inf_{\mu \leq \nu} \Big\{ \max \Big[T_D(\mu) + T \,, \sup_{\mu \leq \kappa \leq \nu} \{ T_A(\kappa) + T \} \Big] \Big\} \\ &= \inf_{\mu \leq \nu} \Big\{ \max \{ T_D(\mu) + T \,, T_A(\nu) + T \} \Big\} \\ &= \max \{ \inf_{\mu \leq \nu} \{ T_D(\mu) \} \,, T_A(\nu) \} + T \\ &= T_A(\nu) + T \,, \end{aligned} \tag{7.8}$$

where we used the fact that $\inf_{\mu \leq \nu} T_D(\mu) = -\infty$, since $T_D \in \mathcal{T}_o$. The bound on the departures is identical to that of a lower service curve, *i.e.*, $T_D(\nu) \leq T_A(\nu) + T$, and not as stringent as the bound imposed by a strict service curve. Both the lower and the adaptive service curves ensure that each ν can experience a delay of up to T time units after its arrival time.

Another advantage of adaptive max-plus service curves is that they can be concatenated. That is, if γ_{S_1} and γ_{S_2} are adaptive max-plus service curves, then $\gamma_{S_1} \overline{\otimes} \gamma_{S_2}$ is also adaptive max-plus. This opens a door to a performance analysis of a network. Given a sequence of network elements each offering an adaptive max-plus service curve, the service guarantee of the entire group of network elements can be obtained by performing a max-plus convolution. We defer the proof of this property to §11.2.

8

Performance Bounds

Consider a network element with a lower max-plus service curve γ_S, which experiences arrivals that are bounded by a max-plus traffic envelope λ_E. We next present an envelope for the departure function T_D, as well as bounds for the delay and the backlog at the network element. All bounds are computed from the deconvolution $\lambda_E \overline{\oslash} \gamma_S$.

Theorem 8.1. Given a flow with max-plus traffic envelope λ_E and a network element that offers the flow a lower max-plus service curve γ_S.

(a) **Envelope for departure time function T_D:** The function $[\lambda_E \overline{\oslash} \gamma_S(\nu)]^+$ is a max-plus traffic envelope for the departure time function T_D for all $\nu \geq 0$, that is,

$$T_D \sim [\lambda_E \oslash \gamma_S]^+ .$$

(b) **Delay bound:** A lower bound on the delay $W(\nu)$ for all $\nu \geq 0$ is

$$W(\nu) \leq -\lambda_E \overline{\oslash} \gamma_S(0) . \tag{8.1}$$

(c) **Backlog bound:** At a network element where the order of departures is the same as the order of arrivals (locally FIFO), an upper bound on the arrival backlog $B^a(\nu)$ for all $\nu \geq 0$ is given by

$$B^a(\nu) \leq \inf \left\{ b \geq 0 \,|\, \lambda_E \overline{\oslash} \gamma_S(b) \geq 0 \right\} . \tag{8.2}$$

Since we assume $T_D \in \mathcal{T}_o$ for the departure time function, the requirement for a locally FIFO order of departures, as expressed for the backlog bound, may appear gratuitous. Indeed, if arrivals and departures at the network element consist of only a single flow, $T_D \in \mathcal{T}_o$ implies a locally FIFO order of the departures. On the other hand, if T_A and T_D express arrivals and departure times for an aggregate of flows, as given in (3.2), we may have $T_D \in \mathcal{T}_o$, even though the traffic of the aggregate does not depart in a locally FIFO fashion.

Proof.
(a) Fix values $\mu, \nu \geq 0$ and derive

$$\begin{aligned} T_D(\mu+\nu) - T_D(\mu) &\geq T_D(\nu+\mu) - T_A \,\overline{\otimes}\, \gamma_S(\mu) \\ &= T_D(\mu+\nu) - \sup_{0 \leq \kappa \leq \mu} \{T_A(\mu-\kappa) + \gamma_S(\kappa)\} \\ &\geq \inf_{\kappa \geq 0} \{T_A(\mu+\nu) - T_A(\mu-\kappa) - \gamma_S(\kappa)\} \\ &\geq \inf_{\kappa \geq 0} \{\lambda_E(\nu+\kappa) - \gamma_S(\kappa)\} \\ &= \lambda_E \,\overline{\oslash}\, \gamma_S(\nu)\,. \end{aligned}$$

In the first line we insert $T_D \leq T_A \,\overline{\otimes}\, \gamma_S$. Then, we expand the max-plus convolution. The third line holds since $T_D \geq T_A$. The next line uses that λ_E is a max-plus traffic envelope. Lastly, since $T_D(\mu+\nu) - T_D(\mu) \geq 0$ always holds, we get that $[\lambda_E \,\overline{\oslash}\, \gamma_S]^+$ is an envelope.

Note that the bound $[\lambda_E \,\overline{\oslash}\, \gamma_S]^+(\nu)$ for the envelope is used only for non-negative values of ν. Since envelopes are assumed to be in $\mathcal{T}_o$, they are assigned $-\infty$ for $\nu < 0$.
(b) The delay bound is derived for $\nu \geq 0$ by

$$\begin{aligned} W(\nu) &= T_D(\nu) - T_A(\nu) \\ &\leq T_A \,\overline{\otimes}\, \gamma_S(\nu) - T_A(\nu) \\ &= \sup_{0 \leq \kappa \leq \nu} \{T_A(\nu-\kappa) + \gamma_S(\kappa) - T_A(\nu)\} \\ &\leq \sup_{\kappa \geq 0} \{\gamma_S(\kappa) - \lambda_E(\kappa)\} \\ &= -\inf_{\kappa \geq 0} \{\lambda_E(\kappa) - \gamma_S(\kappa)\} \\ &= -\lambda_E \,\overline{\oslash}\, \gamma_S(0)\,. \end{aligned}$$

The first three lines spell out the definition of the delay, and use $T_D \leq T_A \overline{\otimes} \gamma_S$. The fourth lines uses that λ_E is a max-plus traffic envelope, and extends the range over which the supremum is taken. The next line takes advantage of $\sup_{x \geq 0} F(x) = -\inf_{x \geq 0} F(-x)$ for any function F, which then yields the deconvolution expression.
(c) We will show that

$$\lambda_E \overline{\oslash} \gamma_S(b) \geq 0 \quad \Longrightarrow \quad \forall \nu \geq 0 : B^a(\nu) \leq b\,.$$

Then, the infimum over all non-negative b satisfying the inequality yields the best bound achievable in this fashion. We start by deriving

$$\begin{aligned} \lambda_E \overline{\oslash} \gamma_S(b) \geq 0 \quad &\Longleftrightarrow \quad \inf_{\kappa \geq 0} \{\lambda_E(b+\kappa) - \gamma_S(\kappa)\} \geq 0 \\ &\Longleftrightarrow \quad \inf_{\kappa \geq b} \{\lambda_E(\kappa) - \gamma_S(\kappa - b)\} \geq 0 \\ &\Longleftrightarrow \quad \forall \kappa \geq b : \lambda_E(\kappa) \geq \gamma_S(\kappa - b)\,. \end{aligned}$$

Therefore, if $\lambda_E \overline{\oslash} \gamma_S(b) \geq 0$, then for each $\nu \geq b$

$$\begin{aligned} T_D(\nu - b) &\leq T_A \overline{\otimes} \gamma_S(\nu - b) \\ &\leq T_A \overline{\otimes} \lambda_E(\nu) \\ &= T_A(\nu)\,. \end{aligned}$$

In the second line we have used that $F \overline{\otimes} H(\nu) \geq G \overline{\otimes} H(\nu - b)$ whenever $F(\nu) \geq G(\nu - b)$, due to the monotonicity of the max-plus convolution. The last line follows by (6.2). For values $\nu < b$, we have $T_D(\nu - b) = -\infty$. Since in a locally FIFO system, $T_D(\nu - b) = T_A(\nu)$ implies $B^a(\nu) \leq b$, we have established that b is a bound on the backlog for all $\nu \geq 0$.

□

Example: Consider the max-plus traffic envelope

$$\lambda_E(\nu) = \left[\frac{\nu}{r} - e\right]^+ ,$$

for some $r > 0$ and $e > 0$, and the lower max-plus service curve

$$\gamma_S(\nu) = \frac{\nu}{C}\,,$$

for a constant $C > r$. To compute the deconvolution

$$\lambda_E \,\overline{\oslash}\, \gamma_S(\nu) = \inf_{\kappa \geq 0} \left\{ \left[\frac{\nu + \kappa}{r} - e \right]^+ - \frac{\kappa}{C} \right\}$$

we distinguish the cases

$$0 \leq \kappa < er - \nu \quad \text{and} \quad \kappa \geq er - \nu \,.$$

Since $0 \leq \kappa < er - \nu$ is only feasible if $\nu < er$, we evaluate the ranges $\nu < er$ and $\nu \geq er$ separately. For $\nu < er$, we get

$$\begin{aligned} \lambda_E \,\overline{\oslash}\, \gamma_S(\nu) &= \min \left(\inf_{0 \leq \kappa < er - \nu} \left\{ -\frac{\kappa}{C} \right\} , \inf_{\kappa \geq er - \nu} \left\{ \frac{\nu + \kappa}{r} - e - \frac{\kappa}{C} \right\} \right) \\ &= \frac{\nu - er}{C} \,. \end{aligned}$$

Note that this term is negative. For $\nu \geq er$, we derive

$$\begin{aligned} \lambda_E \,\overline{\oslash}\, \gamma_S(\nu) &= \inf_{\kappa \geq 0} \left\{ \frac{\nu + \kappa}{r} - e - \frac{\kappa}{C} \right\} \\ &= \frac{\nu}{r} - e \,. \end{aligned}$$

Here, the infimum occurs at $\kappa = 0$ since the function inside the infimum is increasing in κ. So, we have

$$\lambda_E \,\overline{\oslash}\, \gamma_S(\nu) = \begin{cases} \dfrac{\nu - er}{C} \,, & \text{if } \nu < er \,, \\ \dfrac{\nu}{r} - e \,, & \text{if } \nu \geq er \,, \end{cases}$$

which yields the bounds

$$\begin{aligned} T_D &\sim \left[\frac{\nu}{r} - e \right]^+ \,, \\ W(\nu) &\leq \frac{er}{C} \,, \\ B^a(\nu) &\leq er \,. \end{aligned}$$

The output bound can sometimes be improved when the service curve is exact. According to Theorem 6.2, if an exact service curve γ_S is superadditive, it acts as a traffic regulator, which enforces that the departure time function satisfies $T_D \sim \gamma_S$. Therefore, if the service curve $\gamma_S(\nu) = \frac{\nu}{C}$ is exact, the departure envelope is given by the service curve, that is, $T_D \sim \frac{\nu}{C}$, which may improve upon the departure envelope of a lower service curve.

With an adaptive service curve γ_S, it is possible to bound delays even when arrivals are not limited by a max-plus traffic envelope, as long as we can give an *a priori* bound on the backlog.

Theorem 8.2. Given a network element with adaptive service curve γ_S. If the arrival backlog is bounded by $B^a(\nu) \leq b^{\max}$ for all $\nu \geq 0$, then the delay is bounded for all $\nu \geq 0$ by

$$W(\nu) \leq \gamma_S(b^{\max})\,.$$

Proof. We use that $B^a(\nu) \leq \nu$ for all non-negative values ν. Setting $\mu = \nu - B^a(\nu)$, the adaptive service curve as defined in (7.5) gives

$$\begin{aligned} T_D(\nu) \leq \max\Big\{ & T_D(\nu - B^a(\nu)) + \gamma_S(B^a(\nu))\,, \\ & \sup_{\nu - B^a(\nu) \leq \kappa \leq \nu} \big\{T_A(\kappa) + \gamma_S(\nu - \kappa)\big\}\Big\} \\ \leq{} & \max\{T_D(\nu - B^a(\nu)), T_A(\nu)\} + \gamma_S(B^a(\nu)) \\ ={} & T_A(\nu) + \gamma_S(B^a(\nu))\,, \end{aligned}$$

where the second inequality uses that $T_A(\kappa) \leq T_A(\nu)$ and $\gamma_S(\nu - \kappa) \leq \gamma_S(B^a(\nu))$ for the given range of values of κ. In the last step, we used that by definition of the backlog B^a, $T_D(\nu - B^a(\nu)) \leq T_A(\nu)$ holds. Then the bound follows from the definition of the delay, $W(\nu) = T_D(\nu) - T_A(\nu)$, and $b^{\max} \geq B^a(\nu)$. $\square$

9

A Summary of the Min-Plus Network Calculus

Next we review some key concepts and results of the min-plus network calculus, without any derivations or proofs. There are numerous articles and textbooks that cover the min-plus network calculus [1, 7, 9]. The description of the min-plus network calculus will serve us as a reference when we address the mapping between the min-plus and max-plus versions of the network calculus.

Min-plus algebra. A min-plus algebra refers to a dioid $(\mathbb{R} \cup \infty, \min, +)$ or $(\mathbb{Z} \cup \infty, \min, +)$, where the operations are the minimum $(\wedge)$ and the addition $(+)$. The properties are analogous to those of the corresponding max-plus dioids, with ∞ as the neutral element for the minimum and the absorbing element for the addition.

Continuous-time functions describing arrivals, departures, or service are left-continuous, non-negative, and non-decreasing functions $F : \mathbb{R} \to \mathbb{R}_o^+ \cup \{\infty\}$. The set of left-continuous, non-negative and non-decreasing functions is denoted by $\mathcal{F}$. Functions are said to be one-sided if $F(t) = 0$ for $t \leq 0$. The subset of one-sided functions in $\mathcal{F}$ is denoted by $\mathcal{F}_o$. For two functions $F, G \in \mathcal{F}$, the *min-plus convolution* $F \otimes G$ is defined as

$$F \otimes G(t) = \inf_{s \in \mathbb{R}} \{F(s) + G(t-s)\} \ .$$

For $F, G \in \mathcal{F}_o$, the convolution simplifies to

$$F \otimes G(t) = \begin{cases} \inf_{0 \le s \le t} \{F(s) + G(t-s)\} \ , & t > 0 \, , \\ 0 \, , & t \le 0 \, . \end{cases}$$

We have that $(\mathcal{F}, \min, \otimes)$ and $(\mathcal{F}_o, \min, \otimes)$ are also dioids, where the neutral element of the $\otimes$-operation is the function δ, defined as

$$\delta(t) = \begin{cases} \infty \, , & t > 0 \, , \\ 0 \, , & t \le 0 \, . \end{cases}$$

A delay function with delay d, δ_d, is defined as $\delta_d(t) = \delta(t-d)$.

Consider traffic processes $F, G, H \in \mathcal{F}_o$.

(a) **Closure.** If F and G lie in $\mathcal{F}$ ($\mathcal{F}_o$), so does $F \otimes G$.

(b) **Associativity.** $(F \otimes G) \otimes H = F \otimes (G \otimes H)$.

(c) **Commutativity.** $F \otimes G = G \otimes F$.

(d) **Distributivity.** $(F \wedge G) \otimes H = (F \otimes H) \wedge (G \otimes H)$.

(e) **Neutral element.** $F \otimes \delta = F$.

(f) **Time shift.** For all times t and $d > 0$ we have $F \otimes \delta_d(t) = F(t-d)$.

(g) **Monotonicity.** If $F \le G$ then $F \otimes H \le G \otimes H$.

(h) **Boundedness.** $F \otimes G \le F$, in particular, $F \otimes F \le F$.

(i) **Existence of minimum.** For each value of t, there exists an s^* with $0 \le s^* \le t$, such that $F \otimes G(t) = F(s^*) + G(t - s^*)$.

For two functions $F, G \in \mathcal{F}_o$ the min-plus deconvolution is defined as

$$F \oslash G(t) = \sup_{s \ge 0} \{F(t+s) - G(s)\} \ .$$

The operation is not closed in $\mathcal{F}_o$, not commutative, and not associative.

A function F is subadditive if $F(t+\tau) \le F(t) + F(\tau)$ for all $t \ge 0$ and $\tau \ge 0$. For a non-decreasing function F, we define $F^{(n)}$ for $n \ge 0$ as

$$F^{(0)} = \delta \, ,$$
$$F^{(n)} = F \otimes F^{(n-1)} \, .$$

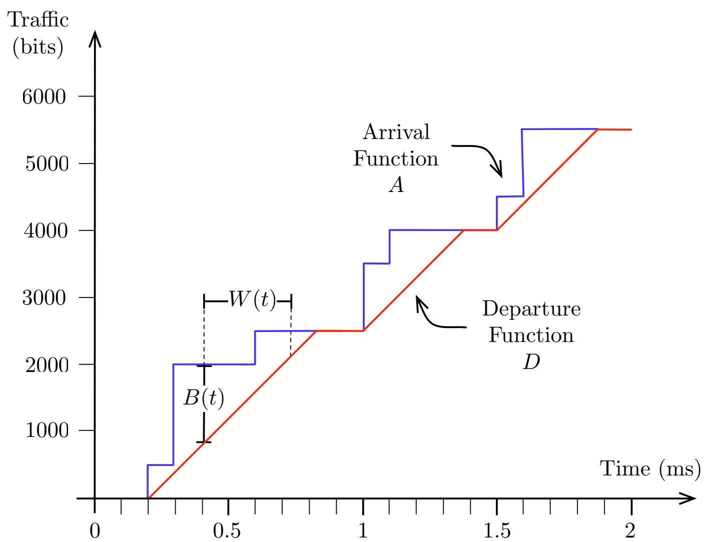


Figure 9.1: Arrival and departure function A and D at a buffered link with rate $C = 4$ Mbps.

The *subadditive closure* F^* of F is defined as

$$F^* = \inf_{n \geq 0} F^{(n)} .$$

Every subadditive closure is subadditive and $F^* \leq F$ always holds.

Subadditivity of a function $F \in \mathcal{F}_o$ expresses itself in a number of ways due to the equivalences

$$F \text{ is subadditive } \iff F = F \otimes F \iff F = F \oslash F \iff F = F^* .$$

Arrivals, departures, backlog, and virtual delay. Arrivals at a network element are described by a function $A \in \mathcal{F}_o$, such that $A(t)$ denotes the cumulative arrivals in the time interval $[0, t)$. The output of a network element in $[0, t)$ is characterized by a departure function $D \in \mathcal{F}_o$, with $D(t) \leq A(t)$. In Figure 9.1 we show the arrival and departure functions A and D for the arrival scenario in Figure 3.1 at a buffered link with a rate of $C = 4$ Mbps. The backlog at time t, $B(t)$, consists of the arrivals that have not yet departed, given by

$$B(t) = A(t) - D(t) . \tag{9.1}$$

In Figure 9.1, the backlog is the vertical distance between the arrival and the departure function. A maximal time interval with $B(t) > 0$ is referred to as *busy period*, and a maximal time interval without backlog ($B(t) = 0$) is called an *idle period*.

With a locally FIFO transmission order, the delay is determined by the transmission time of all traffic that have an earlier arrival time. Consider a time t and suppose that the arrivals prior to t are given by $A(t) = x$. Clearly, when the departure function D exceeds the value x, then all traffic that has arrived before time t has been transmitted. Hence, the delay induced by this traffic is the earliest time y such that $D(t + y) \geq x$. Denoting the delay at time t by $W(t)$, we have

$$W(t) = \inf \{y > 0 \mid D(t + y) \geq A(t)\} . \tag{9.2}$$

In Figure 9.1, $W(t)$ corresponds to the horizontal distance between the arrival and the departure functions at time t. Since $W(t)$ is defined even if there is no arrival at time t, it is often referred to as the *virtual delay* at time t.

Traffic envelopes. A function $E \in \mathcal{F}_o$ is a *min-plus traffic envelope* for an arrival function A, if

$$A = A \otimes E .$$

We write $A \sim E$ to indicate that E is a min-plus traffic envelope for A. For $A \sim E$, $E(\tau)$ expresses the maximum amount of traffic that may arrive in any time interval of length τ, that is, $E(\tau) \geq A(t + \tau) - A(t)$ for all $t \geq 0$. The smaller an envelope, the better, since it provides a tighter description of the enveloped function. Since $A \sim E$ if and only if $A \sim E^*$, and since $E^* \leq E$, where E^* is the subadditive closure of E, all reasonable choices for min-plus traffic envelopes are subadditive. A frequently used min-plus traffic envelope is the *token bucket* $E(t) = (b + rt)I_{t>0}$ with burst size $b \geq 0$ and rate $r > 0$.[1] A token bucket bounds the long-term rate of traffic to r, but permits instantaneous bursts of up to size b. An extension is a token bucket with a peak rate constraint $P > r$, which has the envelope $E(t) = \min\{Pt, b + rt\}I_{t>0}$. Here, a burst cannot be sent instantaneously, since the traffic rate is limited to the peak rate P. The best (tightest) envelope for an arrival function A is given by its empirical envelope $\mathcal{E}_A$, which is computed with $\mathcal{E}_A = A \oslash A$.

[1] I_{expr} denotes an indicator function, with $I_{\text{expr}} = 1$ if 'expr' is true, and $I_{\text{expr}} = 0$ otherwise.

Min-plus service curves. For a network element with arrival function $A \in \mathcal{F}_o$, and corresponding departure function $D \in \mathcal{F}_o$, $S \in \mathcal{F}_o$ is a (an) $\left\{\begin{array}{l}\text{exact}\\ \text{lower}\\ \text{upper}\end{array}\right\}$ min-plus service curve for the network element, if $\left\{\begin{array}{l}D = A \otimes S\\ D \geq A \otimes S\\ D \leq A \otimes S\end{array}\right\}$.

A work-conserving buffered link has an exact service curve $S(t) = C[t]^+$. The service curve of a delay element with delay d is $S = \delta_d$. Service curves of the form $S(t) = R[t-T]^+$ for constants $R > 0$ and $T > 0$ are referred to as latency-rate servers. The service provided by a sequence of network elements with service curves $S_1, \ldots, S_N$ can be equivalently expressed by a single network element with service curve $S_1 \otimes \ldots \otimes S_N$. This is analogous to the concatenation of service curves in the max-plus network calculus.

Given a network element with exact service curve $S(t) = C[t]^+$ that is shared by two flows, a 'through flow' and a 'cross flow'. If the arrivals of the cross flow are bounded by the min-plus traffic envelope E_c, a lower service curve of the through flow is given by

$$S(t) = [Ct - E_c(t)]^+ .$$

This lower service curve is the *residual min-plus service curve*. The residual service curve expresses a lower bound on the link capacity that is left unused by the cross flow.

Min-plus lower service curves of the form $S(t) = C[t]^+$ have the same issue with satisfying rate guarantees as the corresponding max-plus service curves, see §7.3. The issue arises when a flow acquires more than the guaranteed rate for an extended time period, followed by a time period without any service. For example, if a permanently backlogged network element ($A = \delta$) is served at rate $2C$ in the time interval $[0, T]$ and not served at all in the time interval $[T, 2T)$, the condition $D(t) \geq A \otimes S(t)$ is still satisfied for all $t \in [0, 2T]$. Since T is arbitrary, a flow may be served at a rate of zero for arbitrarily long time intervals, without violating the lower service curve. Rate guarantees can be strengthened by using *strict min-plus service curves* or *adaptive min-plus service curves*. A strict min-plus service curve guarantees that for any time interval $[t_1, t_2]$ where the backlog is non-zero, that is, $B(s) > 0$ for $s \in (t_1, t_2)$, the departures at the network element satisfy $D(t) - D(s) \geq S(t-s)$. Every strict min-plus service curve is a lower min-plus service curve. The limitation of strict min-plus service curves is that the concatenation of two strict min-plus service curves S_1 and S_2, $S_1 \otimes S_2$, may not be strict, and that strict min-plus service curves that enforce a delay bound d ($S = \delta_d$) require that busy periods are bounded by d.

The issues with strict min-plus service curves can be avoided with an adaptive min-plus service curve. The definition uses a modified min-plus convolution operation for $F, G \in \mathcal{F}_o$, given by

$$F \underset{s}{\otimes} G(\nu) = \inf_{s \le u \le t} \left\{ F(u) + G(t-u) \right\} .$$

Then, a network element with an arrival function A and departure function D has an adaptive min-plus service curve S, if for all $t \ge 0$,

$$D(t) \ge \sup_{s \le t} \Big\{ \min \Big[D(s) + S(t-s) \, , A \underset{s}{\otimes} S(t) \Big] \Big\} . \tag{9.3}$$

We note that the adaptive min-plus service curves originally proposed in [2, 26] are more general in that the function S appearing in the two parts of the minimum can be a different function in each part (see §13). For the earlier discussed scenario of a permanently backlogged network element, an adaptive min-plus service curve $S(t) = C[t]^+$ ensures that the strict guarantee $D(t) - D(s) \ge S(t-s)$ is satisfied for any time interval $[s, t]$. On the other hand, an adaptive service curve $S = \delta_d$ ensures that $D(t) \ge A(t-d)$ without restricting the length of a busy period. Moreover, for a sequence of two network elements with adaptive min-plus service curves S_1 and S_2, $S_1 \otimes S_2$ is an adaptive min-plus service curve for the sequence of network elements.

Performance bounds. Consider a network element with lower min-plus service curve S and arrivals A that are bounded by a min-plus traffic envelope, $A \sim E$. The network element satisfies the following bounds:

(a) **Envelope for departure function D:** The function $(E \oslash S(\tau)) I_{\tau > 0}$ is a min-plus traffic envelope for the departure function D, that is, $D \sim E \oslash S$.

(b) **Backlog bound:** The backlog $B(t)$ is bounded for arbitrary times $t > 0$ by

$$B(t) \le E \oslash S(0) . \tag{9.4}$$

(c) **Delay bound:** If arrivals are served in a locally FIFO fashion, then the virtual delay $W(t)$ is bounded for all times $t > 0$ by

$$W(t) \leq \inf \{d \geq 0 \,|\, E \oslash S(-d) \leq 0\} \ . \tag{9.5}$$

If we compare the bounds in the min-plus network calculus with the corresponding bounds of the max-plus network calculus from §8, we observe that they are all constructed using the deconvolution of the min-plus or max-plus algebra, respectively, of an arrival envelope and a service curve. In the max-plus setting, the computation of the delay bound is more straightforward than that of the backlog bound, and it is the other way around in the min-plus setting. In §11.3, we will see that the three bounds for the departure envelope, the backlog, and the delay in the min-plus calculus are equivalent to the corresponding bounds for the max-plus network calculus from §8.

10

Isomorphism between the Min-Plus and Max-Plus Algebra

The review of the min-plus network calculus in §9 has exposed many similarities between the max-plus network calculus and the min-plus network calculus. For example, the min-plus convolution and deconvolution appear to have the same properties as their max-plus versions, as long as we exchange infima by suprema and flip the direction of inequalities. Departure envelopes, delay bounds, and backlog bounds at a network element were computed by performing deconvolutions of service curves and traffic envelopes, as was the case in the max-plus network calculus. We also encountered differences between the min-plus and max-plus network calculus. For example, expressing an aggregate of traffic flows is more intuitive in a min-plus framework, where it is a simple sum. Also, when comparing the expressions for the residual max-plus and min-plus service curves, the min-plus version appears much simpler. These superficial observations provide the motivation for studying the relationship between the min-plus and max-plus versions of the network calculus in greater detail. We will find that the analyses of the two approaches are essentially mirror images of each other, in the sense that results in one version of the network calculus can be mapped to results in the other version. The mapping exists between continuous-time functions in the time domain and continuous-space functions in the space domain.

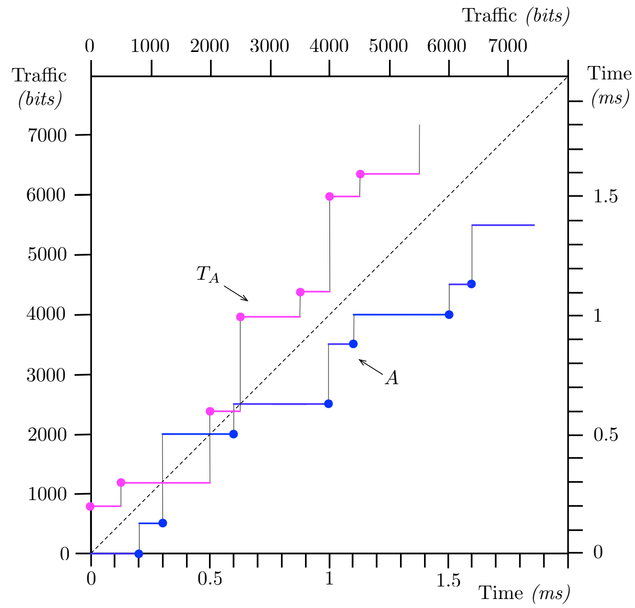


Figure 10.1: Arrival function A and arrival time function T_A as diagonal reflections.

We begin by comparing an arrival function A with an arrival time function T_A. In Figure 10.1 we depict A and T_A for the arrival scenario from Table 3.1. In the figure, the units for the arrival function A are at the bottom and on the left, and the units for the arrival time function T_A are on top and on the right. We can observe that the two characterizations of arrivals are diagonal reflections of each other.

If a function F is continuous and strictly increasing, its diagonal reflection is the inverse function F^{-1}, defined as

$$F(x) = y \quad \Longrightarrow \quad F^{-1}(y) = x\,.$$

By definition, $F^{-1}(F(x)) = x$ and $F(F^{-1}(y)) = y$. Hence, if an arrival function A is continuous and strictly increasing, A and T_A are related by $T_A = A^{-1}$ and $A = T_A^{-1}$. The same extends to the mapping of departure characterizations (D and T_D), traffic envelopes (E and λ_E), and service characterizations (S and γ_S), where we refer to Table 10.1 for the conventions of our notation. However, arrival functions A and arrival time functions T_A generally are not continuous and strictly increasing. Instead, they have 'jumps'

	Time domain (min-plus algebra)	Space domain (max-plus algebra)
Arrivals:	A	T_A
Departures:	D	T_D
Traffic envelope:	E	λ_E
Service curve:	S	γ_S

Table 10.1: Summary of notation.

and 'plateaus'. In Figure 10.1 we see that each packet arrival creates a jump of A, and the elapsed time between two arrival events creates a plateau. For the arrival time function T_A, it is the other way around. In functions with jumps and plateaus, the construction of an inverse is ambiguous. To illustrate this, consider the plateau $A(t) = 2500$, which is assumed for all times in the range 0.6 ms $< t \leq 1$ ms. While the inverse function value cannot be determined at a plateau, always choosing the smallest value or the largest value at a plateau constructs the desired reflection at the diagonal. We refer to functions that are established in this fashion as *pseudo-inverse functions*. If the pseudo-inverse function is constructed by picking the smallest value ($t = 0.6$ for $A(t) = 2500$), we refer to it as *lower pseudo-inverse*, and denote it by $A^{\downarrow}$. If the function is created by always selecting the largest value ($t = 1$ for $A(t) = 2500$), we call it the *upper pseudo-inverse*, and denote it by $A^{\uparrow}$. Pseudo-inverses can be constructed for all non-decreasing functions, be they discrete-time, discrete-space, fluid-flow, or other. The properties of pseudo-inverse functions will provide us with the tools for mapping the network calculus between the time domain and the space domain.

10.1 Properties of pseudo-inverse functions

We now present a formal definition of pseudo-inverse functions. For a non-decreasing function $F : \mathbb{R} \to \mathbb{R} \cup \{-\infty\} \cup \{\infty\}$, the lower pseudo-inverse $F^{\downarrow}$ and the upper pseudo-inverse $F^{\uparrow}$ are given by

$$F^{\downarrow}(y) = \inf \{x \mid F(x) \geq y\} = \sup \{x \mid F(x) < y\} \,, \quad (10.1)$$

$$F^{\uparrow}(y) = \sup \{x \mid F(x) \leq y\} = \inf \{x \mid F(x) > y\} \,. \quad (10.2)$$

Note that we define two versions for each pseudo-inverse. The two versions are identical as long as F is non-decreasing. The attributes 'lower' and 'upper' for the pseudo-inverses are justified since $F^{\uparrow} \geq F^{\downarrow}$. If F is strictly increasing and continuous, we have $F^{\downarrow} = F^{\uparrow} = F^{-1}$.

We can use Figure 10.1 to check that $T_A^{\uparrow}$ tracks the reflection of A at the diagonal, and that $A^{\downarrow}$ tracks the reflection of T_A. We therefore have the mapping

$$T_A(\nu) = A^{\uparrow}(\nu) \quad \text{and} \quad A(t) = T_A^{\downarrow}(t) . \tag{10.3}$$

Shortly, we will make this observation precise.

For functions that have jumps as well as plateaus, $F^{\downarrow}(F(x))$ and $F(F^{\uparrow}(y))$ generally do not recover the argument. Instead, the pseudo-inverses of a non-decreasing function F satisfy the following weaker properties, for all $x, y \in \mathbb{R}$ and arbitrary $\varepsilon > 0$:

(P1) $F^{\downarrow}(F(x)) \leq x < F^{\downarrow}(F(x) + \varepsilon)$.

(P2) $F^{\uparrow}(F(x) - \varepsilon) < x \leq F^{\uparrow}(F(x))$.

(P3) If F is right-continuous, then $F(F^{\downarrow}(y) - \varepsilon) < y \leq F(F^{\downarrow}(y))$.

(P4) If F is left-continuous, then $F(F^{\uparrow}(y)) \leq y < F(F^{\uparrow}(y) + \varepsilon)$.

The properties follow directly from the definitions of the pseudo-inverses. For example, we obtain property *(P1)* from

$$F^{\downarrow}(F(x)) = \inf \{y \mid F(y) \geq F(x)\} \leq x \, ,$$
$$F^{\downarrow}(F(x) + \varepsilon) = \inf \{y \mid F(y) \geq F(x) + \varepsilon\} > x \, .$$

Properties *(P1)* and *(P2)* hold for both left- and right-continuous functions. The other properties require that F is left- or right-continuous. We show this for property *(P4)*, by writing

$$F(F^{\uparrow}(y)) = F\left(\sup \{x \mid F(x) \leq y\}\right) \leq y \, ,$$
$$F(F^{\uparrow}(y) + \varepsilon) = F\left(\sup \{x \mid F(x) \leq y\} + \varepsilon\right) > y \, .$$

The above expressions assume that

$$\sup \{x \mid F(x) \leq y\} \in \{x \mid F(x) \leq y\} \; ,$$

which is ensured as long as F is left-continuous.

Figure 10.2 presents an illustration of the properties. Figure 10.2(a) shows a non-decreasing function F, where $F(x)$ is located on a plateau. In this case, the lower pseudo-inverse $F^{\downarrow}(F(x))$ maps to a value on the horizontal axis that is less than or equal to x, as indicated by the dashed arrow. The pseudo-inverse $F^{\downarrow}(F(x)+\varepsilon)$ maps to a value greater than x. This gives us property *(P1)*. If $F(x)$ is not on a plateau, we have $F^{\downarrow}(F(x)) = x$, and *(P1)* remains valid. Figure 10.2(b) shows the corresponding mapping for the upper pseudo-inverse, which yields the inequalities in property *(P2)*.

Figure 10.2(c) shows a construction of property *(P3)* for a right-continuous function F. We start by selecting a value y on the vertical axis. The figure depicts the interesting case, where y is not a function value of F, and instead lies in the middle of a jump of F. Taking the lower pseudo-inverse $F^{\downarrow}(y)$, and then $F(F^{\downarrow}(y))$ maps to a value on the vertical axis that is greater than or equal to y. On the other hand, $F(F^{\downarrow}(y)-\varepsilon)$ is less than y. Figure 10.2(d) illustrates property *(P4)* for a left-continuous function F, where $F(F^{\uparrow}(y)) \leq y$.

Sometimes we must evaluate the pseudo-inverses in the immediate neighborhood of a function value. Using $F(x^+)$ and $F(x^-)$, as defined in (3.1), and expressing them as

$$F(x^-) = \sup_{\varepsilon>0} F(x-\varepsilon) \quad \text{and} \quad F(x^+) = \inf_{\varepsilon>0} F(x+\varepsilon)\,,$$

we see that the strict inequalities in properties *(P1)*–*(P4)* may become equalities.

Properties *(P1)* and *(P4)* capture the defining properties of the pseudo-inverses. Another, equivalent, way to express the relationships between functions and their pseudo-inverses are properties *(P5)*–*(P8)*.

(P5) $F(x) > y \Longrightarrow F^{\uparrow}(y) \leq x$.

(P6) $F(x) \leq y \Longrightarrow F^{\uparrow}(y) \geq x$.

(P7) $F(x) < y \Longrightarrow F^{\downarrow}(y) \geq x$.

(P8) $F(x) \geq y \Longrightarrow F^{\downarrow}(y) \leq x$.

To show that *(P5)* holds, we note that $F(x) > y$ lets us conclude that $F(x) - \varepsilon \geq y$ for some $\varepsilon > 0$. Applying $F^{\uparrow}$ to both sides maintains the direction

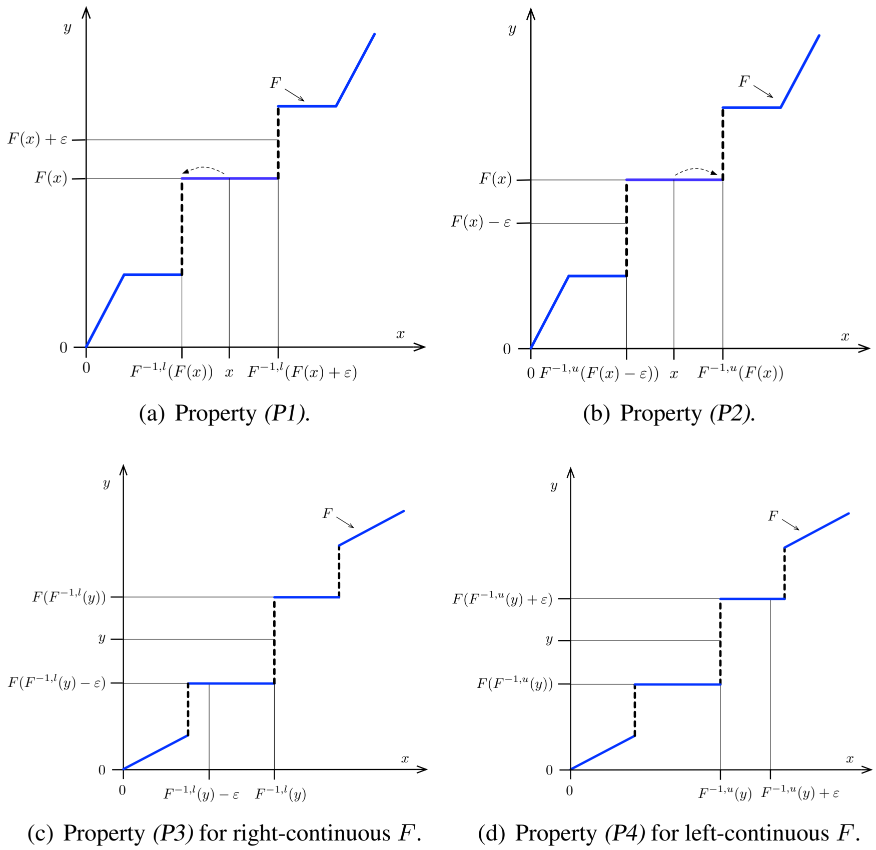


Figure 10.2: Illustrations of the properties of pseudo-inverses of non-decreasing functions.

of the inequality. By property *(P2)*, we obtain $x > F^{\uparrow}(F(x) - \varepsilon) \geq F^{\uparrow}(y)$. For *(P6)*, we can apply $F^{\uparrow}$ to both sides of the inequality $F(x) \leq y$. From property *(P2)*, we then get $x \geq F^{\uparrow}(F(x)) \geq F^{\uparrow}(y)$. The proofs for properties *(P7)* and *(P8)* are similar.

Even though pseudo-inverses are less powerful than actual inverse functions, they are sufficient to create perfect diagonal reflections between the arrival function A and the arrival time function T_A in Figure 10.1. In fact, the pseudo-inverses create a one-to-one mapping between the min-plus and max-plus network calculus. The mapping will be based on a set of fundamental properties of pseudo-inverses with respect to left-continuous and right-continuous functions.

Lemma 10.1 (FUNDAMENTAL PROPERTIES OF PSEUDO-INVERSES)**.** Given non-decreasing functions F and G.

(a) $F^{\downarrow}$ and $F^{\uparrow}$ are non-decreasing.

(b) $F^{\downarrow}$ is left-continuous and $F^{\uparrow}$ is right-continuous.

(c) If F is left-continuous, then $F = \left(F^{\uparrow}\right)^{\downarrow}$.

(d) If F is right-continuous, then $F = \left(F^{\downarrow}\right)^{\uparrow}$.

(e) If $F \geq G$, then $F^{\uparrow} \leq G^{\uparrow}$ and $F^{\downarrow} \leq G^{\downarrow}$.

The second property confirms that taking the lower pseudo-inverse produces a left-continuous function, and the upper pseudo-inverse produces a right-continuous function. Note that this holds for the pseudo-inverses of any non-decreasing function, be it left-continuous, right-continuous, or even continuous. The third and fourth properties state that taking first one type of pseudo-inverse and then the other type of pseudo-inverse of a function recovers the original function. This can be used to justify our observation in Figure 10.1 that A and T_A are diagonal reflections of each other, and the relationship between A and T_A in (10.3). Since A is left-continuous and T_A is right-continuous, the third and fourth properties yield

$$A = \left(A^{\uparrow}\right)^{\downarrow} = T_A^{\downarrow}\,,$$
$$T_A = \left(T_A^{\downarrow}\right)^{\uparrow} = A^{\uparrow}\,.$$

Hence, T_A can be expressed by the pseudo-inverse $A^{\uparrow}$, and A by the pseudo-inverse $T_A^{\downarrow}$. The fifth property, which states that pseudo-inverses reverse the order relationship of functions, is instrumental for mapping service curves and traffic envelopes between the min-plus and the max-plus algebras.

For the proofs of the properties, we view left-continuous functions as functions of time, *e.g.*, $F(t)$, and right-continuous functions as space-domain functions, *e.g.*, $F(\nu)$.

Proof. (of Lemma 10.1)
(a) For the first property, there is nothing to show since the claim follows directly from the definition of the pseudo-inverses in (10.1) and (10.2).
(b) To show that $F^{\uparrow}$ is right-continuous we prove that the limit from the right is equal to the function value. We derive

$$\begin{aligned} F^{\uparrow}((y^*)^+) &= \lim_{\substack{y \to y^* \\ y > y^*}} \inf \{x \mid F(x) > y\} \\ &= \inf_{y > y^*} \inf \{x \mid F(x) > y\} \\ &= \inf_{\substack{x,y \\ y > y^* \\ F(x) > y}} \{x\} \\ &= \inf \{x \mid F(x) > y^*\} \\ &= F^{\uparrow}(y^*)\,. \end{aligned}$$

The first line expresses the limit from the right of the upper pseudo-inverse for an arbitrary value y^*. The second line uses that $F^{\uparrow}$ is non-decreasing. In the next line, we summarize the two infima and collect their conditions. Since the last two conditions give $F(x) > y > y^*$, we can drop y entirely without changing the expression. Rewriting the result in the following line gives $F^{\uparrow}(y^*)$.

The derivation for the lower pseudo-inverse is done in an analogous fashion, by showing that $F^{\downarrow}((y^*)^-) = F^{\downarrow}(y^*)$.
(c) For an arbitrary time $t \in \mathbb{R}$, inserting the definitions of the left and right pseudo-inverses for $(F^{\uparrow})^{\downarrow}$ yields

$$\begin{aligned} (F^{\uparrow})^{\downarrow}(t) &= \inf\{\nu \mid F^{\uparrow}(\nu) \geq t\} \\ &= \inf\{\nu \mid \sup\{s \mid F(s) \leq \nu\} \geq t\}\,. \end{aligned}$$

We introduce notation for the two sets appearing in the equation and define

$$\begin{aligned} M_t &= \{\nu \mid \sup\{s \mid F(s) \leq \nu\} \geq t\}\,, \\ M_{t,\nu} &= \{s \mid F(s) \leq \nu\}\,. \end{aligned}$$

We prove the claim by showing that $F(t) = \inf M_t$. First, we argue that $F(t) \in M_t$, and then we show that $\hat{\nu} \notin M_t$ if $\hat{\nu} < F(t)$. If $\nu = F(t)$, the condition for membership in M_t becomes

$$\sup M_{t,F(t)} = \sup\{s \mid F(s) \leq F(t)\} \geq t\,.$$

The supremum above is either t, or a number $s^* > t$ with $F(s^*) = F(t)$. In both cases, the condition that the supremum is greater than or equal to t is satisfied, and we conclude that $F(t) \in M_t$. If a $\hat{\nu} < F(t)$, the condition for membership in M_t reads

$$\sup M_{t,\hat{\nu}} = \sup\{s \mid F(s) \leq \hat{\nu}\} \geq t\,.$$

Let $s^* = \sup M_{t,\hat{\nu}}$ be the supremum. Since F is left-continuous, we have that $s^* \in M_{t,\hat{\nu}}$, and, therefore, $F(s^*) \leq \hat{\nu}$. Together with the assumption on $\hat{\nu}$, we now have $F(s^*) \leq \hat{\nu} < F(t)$. Since F is non-decreasing, $F(s^*) \leq \hat{\nu} < F(t)$ implies that $s^* < t$. Hence, with $\hat{\nu}$, the condition for membership in M_t cannot be satisfied, and we have that $\hat{\nu} \notin M_t$.

(d) The proof is analogous to the left-continuous case shown above. For an arbitrary $\nu \in \mathbb{R}$, we have

$$\begin{aligned} \left(F^{\downarrow}\right)^{\uparrow}(\nu) &= \sup\{t \mid F^{\downarrow}(t) \leq \nu\} \\ &= \sup\{t \mid \inf\{k \mid F(k) \geq t\} \leq \nu\}\,. \end{aligned}$$

If we define $M_\nu = \{t \mid \inf\{k \mid F(k) \geq t\} \leq \nu\}$, we can proceed by showing that $F(\nu) = \sup M_\nu$.

(e) The proof of the third property uses the definitions of the pseudo-inverses.

$$\begin{aligned} F^{\uparrow}(\nu) &= \sup\{t \mid F(t) \leq \nu\} \\ &\leq \sup\{t \mid G(t) \leq \nu\} \\ &= G^{\uparrow}(\nu)\,, \end{aligned}$$

and

$$\begin{aligned} F^{\downarrow}(t) &= \inf\{\nu \mid F(\nu) \geq t\} \\ &\leq \inf\{\nu \mid G(\nu) \geq t\} \\ &= G^{\downarrow}(t)\,, \end{aligned}$$

where the inequality in both cases follows from $F \geq G$, since F and G are non-decreasing. $\square$

10.2 Mapping of algebras

Next we exploit the pseudo-inverses to map an arbitrary expression in the min-plus algebra to an expression in the max-plus algebra, and vice versa. The mapping is one-to-one and onto, thus creating an isomorphism between the algebras.

We address the mapping between the min-plus algebra of left-continuous functions in the sets $\mathcal{F}$ and $\mathcal{F}_o$ and the max-plus algebra of right-continuous functions in $\mathcal{T}$ and $\mathcal{T}_o$.

All functions that characterize arrivals, departures, traffic envelopes and service curves of the min-plus network calculus lie in $\mathcal{F}_o$, and in $\mathcal{T}_o$ for the max-plus network calculus. The relationship between the sets $\mathcal{F}_o$ and $\mathcal{T}_o$ is as follows.

Theorem 10.2 (MAPPING FUNCTIONS BETWEEN THE TIME DOMAIN AND SPACE DOMAIN)**.**

(a) If $F \in \mathcal{F}_o$ then $F^{\uparrow} \in \mathcal{T}_o$.

(b) If $F \in \mathcal{T}_o$ then $F^{\downarrow} \in \mathcal{F}_o$.

(c) $\delta^{\uparrow} = \bar{\delta}$ and $\bar{\delta}^{\downarrow} = \delta$.

Proof. By Lemma 10.1(a), $F^{\uparrow}$ and $F^{\downarrow}$ are non-decreasing, as long as F is non-decreasing. To prove the first claim, we evaluate the upper pseudo-inverse of a function $F \in \mathcal{F}_o$ at $\nu < 0$ and obtain

$$F^{\uparrow}(\nu) = \sup\{t \mid F(t) \leq \nu\} = -\infty\,,$$

since the supremum is taken over an empty set. For $\nu \geq 0$, we get that $F^{\uparrow}(\nu) \geq 0$ since $F(0) \leq \nu$. We therefore have $F^{\uparrow} \in \mathcal{T}_o$.

For the second claim, consider the lower pseudo-inverse of a function $F \in \mathcal{T}_o$, and evaluate it for $t \leq 0$. This yields

$$F^{\downarrow}(t) = \inf\{\nu \mid F(\nu) \geq t\} = 0\,,$$

since $F(0) \geq t$ and $F(\nu) = -\infty$ for $\nu < 0$. For values $t \geq 0$, we obtain that $F^{\downarrow}(t) \geq 0$. Hence, we get $F^{\downarrow} \in \mathcal{F}_o$.

For the third claim we write

$$\delta^{\uparrow}(\nu) = \sup\{t \mid \delta(t) \leq \nu\} = \begin{cases} -\infty\,, & \text{if } \nu < 0\,, \\ 0\,, & \text{if } \nu \geq 0\,, \end{cases}$$

$$\bar{\delta}^{\downarrow}(t) = \inf\{t \mid \bar{\delta}(\nu) \geq t\} = \begin{cases} 0\,, & \text{if } t \leq 0\,, \\ \infty\,, & \text{if } t > 0\,, \end{cases}$$

where we have used that $\sup \emptyset = -\infty$ and $\inf \emptyset = \infty$. □

Now we consider the relationship between the operations in the two algebras.

Theorem 10.3 (MAPPING MIN-PLUS ALGEBRA OPERATIONS)**.**
Let F and G be non-negative non-decreasing left-continuous functions, that is $F, G \in \mathcal{F}$.

(a) $\big(F \wedge G\big)^{\uparrow} = F^{\uparrow} \vee G^{\uparrow}$.

(b) $\big(F \otimes G\big)^{\uparrow} = F^{\uparrow} \,\overline{\otimes}\, G^{\uparrow}$.

(c) $\big(F \oslash G\big)^{\uparrow} = F^{\uparrow} \,\overline{\oslash}\, G^{\uparrow}$.

(d) $\big(F + G\big)^{\uparrow}(\nu) = \inf\limits_{0 \leq \kappa \leq \nu} \max\{F^{\uparrow}(\kappa), G^{\uparrow}(\nu - \kappa)\}\,, \quad \forall \nu \in \mathbb{R}$.

(e) If $F \in \mathcal{F}_o$ is subadditive then $F^{\uparrow}$ is superadditive.

Operations of the max-plus algebra and relations involving their operations can be mapped to the min-plus algebra with the help of the lower pseudo-inverse. The results are summarized in the next theorem.

Theorem 10.4 (MAPPING MAX-PLUS ALGEBRA OPERATIONS)**.**
Let F and G be non-decreasing right-continuous functions ($F, G \in \mathcal{T}$).

(a) $\big(F \vee G\big)^{\downarrow} = F^{\downarrow} \wedge G^{\downarrow}$.

(b) $\big(F \,\overline{\otimes}\, G\big)^{\downarrow} = F^{\downarrow} \otimes G^{\downarrow}$.

(c) $\left(F \,\overline{\oslash}\, G\right)^{\downarrow} = F^{\downarrow} \oslash G^{\downarrow}$.

(d) $\left(\inf_{0 \le s \le t} \max\{F(s), G(t-s)\}\right)^{\downarrow} = F^{\downarrow}(t) + G^{\downarrow}(t)$.

(e) If $F \in \mathcal{T}_o$ is superadditive then $F^{\downarrow}$ is subadditive.

With these properties we can map any expression in one algebra to the other algebra. For example, the aggregation of a set of flows with arrival functions $A_1, \ldots, A_N$ and arrival time functions $T_{A_1}, \ldots, T_{A_N}$ is governed by

$$\left(\sum_{j=1}^{N} A_j(t)\right)^{\uparrow} = \inf_{\substack{\nu_1,\ldots,\nu_N \\ \nu=\nu_1+\ldots+\nu_N}} \max_{j=1,\ldots,N} T_{A_j}(\nu_j)\,,$$

$$\left(\inf_{\substack{\nu_1,\ldots,\nu_N \\ \nu=\nu_1+\ldots+\nu_N}} \max_{j=1,\ldots,N} T_{A_j}(\nu_j)\right)^{\downarrow} = \sum_{j=1}^{N} A_j(t)\,,$$

which justifies the expression in (3.2).

We now present the proofs for the mapping of operations from the time domain to the space domain. The proofs for the mapping from the space domain to the time domain are analogous. We assume throughout that time-domain functions belong to $\mathcal{F}$, and space-domain functions belong to $\mathcal{T}$.

Proof. (of Theorem 10.3)
(a) The derivation expands the definition of the upper pseudo-inverse.

$$\begin{aligned}
\left(F \wedge G\right)^{\uparrow}(\nu) &= \sup\{t \mid \min\left(F(t), G(t)\right) \le \nu\} \\
&= \sup\{t \mid F(t) \le \nu \ \text{ or } \ G(t) \le \nu\} \\
&= \max\{\sup\{t \mid F(t) \le \nu\}\,, \sup\{t \mid G(t) \le \nu\}\} \\
&= \left(F^{\uparrow} \vee G^{\uparrow}\right)(\nu)\,.
\end{aligned}$$

(b) First we show that $(F \otimes G)^{\uparrow} \le F^{\uparrow} \,\overline{\otimes}\, G^{\uparrow}$.

$$\begin{aligned}
\left(F \otimes G\right)^{\uparrow}(\nu) &= \sup\{t \mid F \otimes G(t) \le \nu\} \\
&= \sup\{t \mid \inf_{0 \le s \le t}\{F(s) + G(t-s)\} \le \nu\} \\
&= \sup\{t \mid F(s_t) + G(t - s_t) \le \nu\}
\end{aligned}$$

$$
\begin{aligned}
&\le \sup\{t \mid F(s_t) \le \kappa_t \ \text{ and } \ G(t - s_t) \le \nu - \kappa_t\} \\
&\le \sup\{t \mid F^{\uparrow}(\kappa_t) \ge s_t \ \text{ and } \ G^{\uparrow}(\nu - \kappa_t) \ge t - s_t\} \\
&\le \sup\{t \mid F^{\uparrow}(\kappa_t) + G^{\uparrow}(\nu - \kappa_t) \ge t\} \\
&\le \sup\{t \mid \exists \kappa, 0 \le \kappa \le \nu : F^{\uparrow}(\kappa) + G^{\uparrow}(\nu - \kappa) \ge t\} \\
&= \sup\{t \mid \sup_{0 \le \kappa \le \nu} \{F^{\uparrow}(\kappa) + G^{\uparrow}(\nu - \kappa)\} \ge t\} \\
&= \sup\{t \mid F^{\uparrow} \overline{\otimes} G^{\uparrow}(\nu) \ge t\} \\
&= F^{\uparrow} \overline{\otimes} G^{\uparrow}(\nu)\,.
\end{aligned}
$$

In the third line, we denote by s_t with $s_t \le t$ the point where the infimum is attained. The fact that it is attained follows from property (i) of the min-plus convolution operation given in §9. Then we define $\kappa_t = F(s_t)$. In the fifth line, we take advantage of the order-reversing property of the pseudo-inverse.

For the reverse direction we derive

$$
\begin{aligned}
F^{\uparrow} \overline{\otimes} G^{\uparrow}(\nu) &= \sup_{0 \le \kappa \le \nu} \{\sup\{\tau \mid F(\tau) \le \kappa\} + \sup\{s \mid G(s) \le \nu - \kappa\}\} \\
&= \sup_{0 \le \kappa \le \nu} \{\sup\{t \mid t = \tau + s \ \text{ and } \ F(\tau) \le \kappa \ \text{ and } \ G(s) \le \nu - \kappa\}\} \\
&\le \sup_{0 \le \kappa \le \nu} \{\sup\{t \mid t = \tau + s \ \text{ and } \ F(\tau) + G(s) \le \nu\}\} \\
&= \sup_{0 \le \kappa \le \nu} \{\sup\{t \mid \inf_{0 \le \tau \le t} \{F(\tau) + G(t - \tau) \le \nu\}\} \\
&= \sup_{0 \le \kappa \le \nu} \{\sup\{t \mid F \otimes G(t) \le \nu\}\} \\
&= \big(F \otimes G\big)^{\uparrow}(\nu)\,.
\end{aligned}
$$

(c) We first show that $(F \oslash G)^{\uparrow} \ge F^{\uparrow} \overline{\oslash} G^{\uparrow}$.

$$
\begin{aligned}
\big(F \oslash G\big)^{\uparrow}(\nu) &= \inf\{t \mid F \oslash G(t) > \nu\} \\
&= \inf\{t \mid \sup_{s \ge 0} \{F(t + s) - G(s)\} > \nu\} \\
&= \inf\{t \mid F(t + s_t) - G(s_t) > \nu\} \\
&= \inf\{t \mid F(t + s_t) > \nu + \kappa_t \ \text{ and } \ G(s_t) \le \kappa_t\}
\end{aligned}
$$

$$
\begin{aligned}
&\geq \inf\{t \mid F^{\uparrow}(\nu+\kappa_t) \leq t + s_t \text{ and } G^{\uparrow}(\kappa_t) \geq s_t\} \\
&\geq \inf\{t \mid F^{\uparrow}(\nu+\kappa_t) - G^{\uparrow}(\kappa_t) \leq t\} \\
&\geq \inf\{t \mid \exists \kappa \geq 0 : F^{\uparrow}(\nu+\kappa) - G^{\uparrow}(\kappa) \leq t\} \\
&= \inf\{t \mid \inf_{\kappa \geq 0}\{F^{\uparrow}(\nu+\kappa) + G^{\uparrow}(\kappa)\} \leq t\} \\
&= \inf\{t \mid F^{\uparrow} \,\overline{\oslash}\, G^{\uparrow}(\nu) \leq t\} \\
&= F^{\uparrow} \,\overline{\oslash}\, G^{\uparrow}(\nu)\,.
\end{aligned}
$$

In the third line, we let $s_t \leq t$ be the time where the infimum is attained. In the next line we define $\kappa_t = G(s_t)$. After that, we apply the order-reversing properties *(P5)* and *(P6)*.

Now we argue that $F^{\uparrow} \,\overline{\oslash}\, G^{\uparrow} \geq (F \oslash G)^{\uparrow}$.

$$
\begin{aligned}
F^{\uparrow} \,\overline{\oslash}\, G^{\uparrow}(\nu) &= \inf_{\kappa \geq 0}\big\{\sup\{\tau \mid F(\tau) \leq \nu+\kappa\} - \sup\{s \mid G(s) \leq \kappa\}\big\} \\
&= \inf_{\kappa \geq 0}\big\{\sup\{\tau \mid F(\tau) \leq \nu+\kappa\} - s_\kappa\big\} \\
&= \inf_{\kappa \geq 0}\big\{\sup\{t \mid F(t+s_\kappa) \leq \nu+\kappa\}\big\} \\
&\geq \inf_{\kappa \geq 0}\big\{\sup\{t \mid F(t+s_\kappa) - G(s_\kappa) \leq \nu\}\big\} \\
&= \sup\big\{t \mid \forall \kappa \geq 0 : F(t+s_\kappa) - G(s_\kappa) \leq \nu\big\} \\
&\geq \sup\big\{t \mid \forall s \geq 0 : F(t+s) - G(s) \leq \nu\big\} \\
&= \sup\big\{t \mid \sup_{s \geq 0}\{F(t+s) - G(s)\} \leq \nu\big\} \\
&= \sup\big\{t \mid F \oslash G(t) \leq \nu\big\} \\
&= \big(F \oslash G\big)^{\uparrow}(\nu)\,.
\end{aligned}
$$

In the second line, we set $s_\kappa = G^{\uparrow}(\kappa)$. Since G is left-continuous, s_κ is in the set and we have $G(s_\kappa) \leq \kappa$. In the third line, we perform the substitution $t = \tau - s_\kappa$. For the fourth line, we use that $G(s_\kappa) \leq \kappa$. Since we substitute κ by something that is not greater we have restricted the supremum. For the next line, we express the infimum equivalently by a universal quantifier inside the supremum. For the second inequality, we relax the constraint on s_κ and further restrict the supremum.

(d) The claim follows from the following derivation.

$$\begin{aligned}
&\inf_{0\le\kappa\le\nu} \max\{F^{\uparrow}(\kappa), G^{\uparrow}(\nu-\kappa)\} \\
&\qquad\qquad = \inf_{0\le\kappa\le\nu} \max\{\inf\{\tau \mid F(\tau) > \kappa\}, \inf\{s \mid G(s) > \nu-\kappa\}\} \\
&\qquad\qquad = \inf\{t \mid \exists\kappa, 0 \le \kappa \le \nu : F(t) > \kappa \text{ and } G(t) > \nu-\kappa\} \\
&\qquad\qquad = \inf\{t \mid F(t) + G(t) > \nu\} \\
&\qquad\qquad = \big(F+G\big)^{\uparrow}(\nu)\,.
\end{aligned}$$

For the second line, since both F and G are non-decreasing, the maximum is the infimum over t such that $F(t) > \kappa$ and $G(t) > \nu - \kappa$.
(e) We use that a subadditive function $F \in \mathcal{F}_o$ satisfies $F = F \otimes F$, and that a superadditive function $F \in \mathcal{T}_o$ is characterized by $F = F \,\overline{\otimes}\, F$.

$$\begin{aligned}
F \in \mathcal{F}_o \text{ subadditive} &\iff F \otimes F = F \\
&\implies \big(F \otimes F\big)^{\uparrow} = F^{\uparrow} \\
&\iff F^{\uparrow} \,\overline{\otimes}\, F^{\uparrow} = F^{\uparrow} \\
&\iff F^{\uparrow} \in \mathcal{T}_o \text{ superadditive}\,.
\end{aligned}$$

□

In summary, Theorem 10.2 constructs a bijection between the sets $\mathcal{F}_o$ and $\mathcal{T}_o$. Theorems 10.3 and 10.4 establish an exact correspondence between the operations in the min-plus algebra and the max-plus algebra. The upper and lower pseudo-inverse hence create a pair of isomorphisms between the two algebraic structures. According to Lemma 10.1(d), the isomorphisms are order-reversing.

11

Min-plus and Max-plus Duality in the Network Calculus

We next use the mapping between the min-plus and the max-plus algebras to show that traffic envelopes and service curve definitions in the max-plus network calculus are consistent with the corresponding definitions in the min-plus network calculus. When considering backlog and delay, we find that a precise mapping of these quantities does not exist.

11.1 Mapping of traffic envelopes

Given a min-plus traffic envelope E and an arrival function A that complies to the envelope ($A \sim E$), we can construct a max-plus traffic envelope λ_E for the space domain. Likewise, given $T_A \sim \lambda_E$, we can construct a min-plus traffic envelope E for the time domain. This mapping of envelopes follows as a corollary from Theorems 10.3 and 10.4.

Corollary 11.1. Given an arrival function $A \in \mathcal{F}_o$ and a min-plus traffic envelope $E \in \mathcal{F}_o$, then

$$A \sim E \Longrightarrow A^{\uparrow} \sim E^{\uparrow} .$$

Given an arrival time function $T_A \in \mathcal{T}_o$ and a max-plus traffic envelope $\lambda_E \in \mathcal{T}_o$, then

$$T_A \sim \lambda_E \Longrightarrow T_A{}^{\downarrow} \sim \lambda_E{}^{\downarrow} .$$

Proof. Recall that min-plus and max-plus traffic envelopes in $\mathcal{F}_o$ and $\mathcal{T}_o$, respectively, can be characterized in terms of convolutions, *i.e.*,

$$A \sim E \quad \Longleftrightarrow \quad A = A \otimes E \,,$$
$$T_A \sim \lambda_E \quad \Longleftrightarrow \quad T_A = T_A \,\overline{\otimes}\, \lambda_E \,.$$

Using Theorem 10.3(b), we obtain from $A = A \otimes E$ that

$$A^{\uparrow} = (A \otimes E)^{\uparrow} = A^{\uparrow} \,\overline{\otimes}\, E^{\uparrow} \,,$$

and, therefore, $A^{\uparrow} \sim E^{\uparrow}$. Likewise, with Theorem 10.4(b), $T_A = T_A \,\overline{\otimes}\, \lambda_E$ implies that

$$T_A^{\downarrow} = (T_A \,\overline{\otimes}\, \lambda_E)^{\downarrow} = T_A^{\downarrow} \otimes \lambda_E^{\downarrow} \,,$$

which gives $T_A{}^{\downarrow} \sim \lambda_E{}^{\downarrow}$.

□

Example: We perform the mapping of the min-plus traffic envelope for a token bucket, given by $E(t) = (b + rt)I_{t>0}$ with $b, r > 0$. Since $E \in \mathcal{F}_o$, we get from Theorem 10.2(a) that $E^{\uparrow} \in \mathcal{T}_o$, and, therefore, $E^{\uparrow}(\nu) = -\infty$ for $\nu < 0$. For $\nu \geq 0$, we compute $E^{\uparrow}$ as

$$\begin{aligned} E^{\uparrow}(\nu) &= \sup\{t \mid (b + rt)I_{t>0} \leq \nu\} \\ &= \max\bigl\{\sup\{t > 0 \mid b + rt \leq \nu\}, 0\bigr\} \\ &= \max\Bigl\{\frac{\nu - b}{r}, 0\Bigr\} \,. \end{aligned}$$

Setting $e = \frac{b}{r}$ and comparing the right-inverse $E^{\uparrow}$ with λ_E from (6.3), we observe that the min-plus traffic envelope for the token bucket corresponds to the max-plus traffic envelope of a rate controller with earliness allowance e.

11.2 Mapping of service curves

We can relate service curves in the time and space domains in a similar fashion as traffic envelopes.

Corollary 11.2. Given a network element with left-continuous arrival and departure functions A and D.

If S is a (an) $\begin{Bmatrix} \text{exact} \\ \text{lower} \\ \text{upper} \end{Bmatrix}$ min-plus service curve, then $S^{\uparrow}$ is a (an) $\begin{Bmatrix} \text{exact} \\ \text{lower} \\ \text{upper} \end{Bmatrix}$ max-plus service curve, that is,

$$\begin{Bmatrix} D = A \otimes S & \Longrightarrow & D^{\uparrow} = A^{\uparrow} \overline{\otimes} S^{\uparrow} \\ D \geq A \otimes S & \Longrightarrow & D^{\uparrow} \leq A^{\uparrow} \overline{\otimes} S^{\uparrow} \\ D \leq A \otimes S & \Longrightarrow & D^{\uparrow} \geq A^{\uparrow} \overline{\otimes} S^{\uparrow} \end{Bmatrix}.$$

Corollary 11.3. Given a network element with right-continuous arrival time function T_A and departure time function T_D.

If γ_S is a (an) $\begin{Bmatrix} \text{exact} \\ \text{lower} \\ \text{upper} \end{Bmatrix}$ max-plus service curve, then $\gamma_S^{\downarrow}$ is a (an) $\begin{Bmatrix} \text{exact} \\ \text{lower} \\ \text{upper} \end{Bmatrix}$ min-plus service curve, that is,

$$\begin{Bmatrix} T_D = T_A \overline{\otimes} \gamma_S & \Longrightarrow & T_D^{\downarrow} = T_A^{\downarrow} \otimes \gamma_S^{\downarrow} \\ T_D \leq T_A \overline{\otimes} \gamma_S & \Longrightarrow & T_D^{\downarrow} \geq T_A^{\downarrow} \otimes \gamma_S^{\downarrow} \\ T_D \geq T_A \overline{\otimes} \gamma_S & \Longrightarrow & T_D^{\downarrow} \leq T_A^{\downarrow} \otimes \gamma_S^{\downarrow} \end{Bmatrix}.$$

The mappings follow directly from Theorems 10.3(b) and 10.4(b), in conjunction with the order-reversing property of pseudo-inverses in Lemma 10.1(e).

We now present examples of service curves that are mapped from the time domain to the space domain. We note that for every min-plus service curve $S \in \mathcal{F}_o$, we have $S^{\uparrow} \in \mathcal{T}_o$, and, therefore, $S^{\uparrow}(\nu) = -\infty$ if $\nu < 0$.

Examples:

(a) Let us first consider a min-plus delay service curve with δ_d (as defined in §9). We realize that for all finite $\nu \geq 0$, we have

$$S^{\uparrow}(\nu) = \sup\{t \mid \delta_d(t) \leq \nu\} = d\,.$$

Therefore,

$$S^{\uparrow}(\nu) = \begin{cases} -\infty\,, & \text{if } \nu < 0\,, \\ d\,, & \text{if } \nu \geq 0\,, \end{cases}$$

which corresponds to $\overline{\delta}_d$ in §4.

(b) Next we map the latency-rate service curve $S(t) = R(t-T)I_{t>T}$ with $R, T > 0$ from the time domain to the space domain. The computation of the upper pseudo-inverse for $\nu \geq 0$ yields

$$\begin{aligned} S^{\uparrow}(\nu) &= \sup\{t \mid R(t-T)I_{t>T} \leq \nu\} \\ &= \sup\{t > T \mid R(t-T) \leq \nu\} \\ &= \frac{\nu}{R} + T\,. \end{aligned}$$

This gives the pseudo-inverse function

$$S^{\uparrow}(\nu) = \begin{cases} -\infty\,, & \text{if } \nu < 0\,, \\ \dfrac{\nu}{R} + T\,, & \text{if } \nu \geq 0\,. \end{cases}$$

We see that latency-rate service curves in the time domain become affine functions in the space domain.

(c) The residual service curve $S(t) = [Ct - E_c(t)]^+$ expresses a lower bound on the available service of a flow at a work-conserving link with rate C, where the cross traffic complies to the min-plus traffic envelope E_c. Note that the difference $Ct - E_c(t)$ is generally not a non-decreasing function. This creates an issue since all of our results for mapping expressions between the time domain and the space domain assume that functions are non-decreasing. We sidestep this issue by assuming that the traffic envelope E_c is concave. This ensures that the inverse E_c^{-1} exists and that the inverse of $S(t) = [Ct - E_c(t)]^+$ is convex and strictly increasing for all t where $Ct > E_c(t)$. With this assumption, we derive the pseudo-inverse of S by

$$\begin{aligned} S^{\uparrow}(\nu) &= \inf\left\{t \mid [Ct - E_c(t)]^+ > \nu\right\} \\ &= \inf\{t \mid E_c(t) < Ct - \nu\} \\ &= \inf\{t \mid \lambda_c(Ct - \nu) \geq t\} \\ &= \frac{1}{C}\left(\inf\left\{x \mid \lambda_c(x) \geq \frac{x+\nu}{C}\right\} + \nu\right)\,. \end{aligned}$$

In the second line, we used that $Ct \geq E_c(t)$ must be satisfied for $\nu \geq 0$. Also, since $Ct - E_c(t)$ is convex and strictly increasing, we can relax the strict inequality. In the next line, we use that $E_c(x) \leq y$ if and only if $E_c^{-1}(y) \geq x$, which is stronger than property *(P6)*. In the last line, we substitute $x = Ct - \nu$, and obtain the residual service curve from (7.2).

For specific traffic envelopes E_c, we can compute the space-domain version directly. Let us consider $E_c(t) = \{b+rt\}I_{t>0}$ with $r < C$. For $\nu \geq 0$, we compute

$$\begin{aligned} S^{\uparrow}(\nu) &= \sup\Big\{t \,\Big|\, \big[Ct - \{b+rt\}I_{t>0}\big]^{+} \leq \nu\Big\} \\ &= \max\Big\{\sup\Big\{t > \frac{b}{C-r} \,\Big|\, (C-r)t - b \leq \nu\Big\}, \\ &\qquad\qquad \sup\Big\{t \leq \frac{b}{C-r} \,\Big|\, 0 \leq \nu\Big\}\Big\} \\ &= \frac{\nu + b}{C-r}\,. \end{aligned}$$

For $b = er$ this service curve is identical to the residual max-plus service curve from (7.3), which we derived for a buffered link where cross traffic is bounded by a rate r with earliness allowance e.

The mapping between service curves in the time and space domain can be extended to adaptive service curves. The following lemma relates the modified convolution operators used in the definitions of the min-plus and max-plus versions of the adaptive service curve.

Lemma 11.4. Given two non-decreasing functions F and G.

(a) If $F, G \in \mathcal{F}_o$, then, for all $x \geq 0$ we have $\big(F \underset{x}{\otimes} G\big)^{\uparrow} = F^{\uparrow} \underset{F(x)}{\overline{\otimes}} G^{\uparrow}$.

(b) If $F, G \in \mathcal{T}_o$, then, for all $x \geq 0$ we have $\big(F \underset{x}{\overline{\otimes}} G\big)^{\downarrow} = F^{\downarrow} \underset{F(x)}{\otimes} G^{\downarrow}$.

Proof. We prove the first claim. The prove of the second claim is done in the same fashion.

$$\begin{aligned} \Big(F \underset{x}{\otimes} G\Big)^{\uparrow}(\nu) &= \sup\{t \mid F \underset{x}{\otimes} G(t) \leq \nu\} \\ &= \sup\Big\{t \mid \inf_{x \leq s \leq t}\{F(s) + G(t-s)\} \leq \nu\Big\} \\ &= \sup\{t \mid F(s_t) + G(t - s_t) \leq \nu\} \end{aligned}$$

$$
\begin{aligned}
&= \sup\{t \mid F(s_t) \le \kappa_t \text{ and } G(t-s_t) \le \nu - \kappa_t\} \\
&\le \sup\{t \mid F^{\uparrow}(\kappa_t) \ge s_t \text{ and } G^{\uparrow}(\nu-\kappa_t) \ge t - s_t\} \\
&\le \sup\{t \mid F^{\uparrow}(\kappa_t) + G^{\uparrow}(\nu-\kappa_t) \ge t\} \\
&\le \sup\{t \mid \exists \kappa, F(x) \le \kappa \le \nu : F^{\uparrow}(\kappa) + G^{\uparrow}(\nu-\kappa) \ge t\} \\
&= \sup\Big\{t \mid \sup_{F(x)\le\kappa\le\nu} \{F^{\uparrow}(\kappa) + G^{\uparrow}(\nu-\kappa)\} \ge t\Big\} \\
&= \sup\{t \mid F^{\uparrow} \underset{F(x)}{\overline{\otimes}} G^{\uparrow}(\nu) \ge t\} \\
&= F^{\uparrow} \underset{F(x)}{\overline{\otimes}} G^{\uparrow}(\nu)\,.
\end{aligned}
$$

In the third line, we used property (i) of the min-plus convolution from §9. In the fourth line we set $\kappa_t = F(s_t)$, and in the fifth line we apply property *(P6)*. For the seventh line, we use $F(s_t) \ge F(x)$. For the other direction we derive

$$
\begin{aligned}
F^{\uparrow} \underset{F(x)}{\overline{\otimes}} G^{\uparrow}(\nu) &= \sup_{F(x)\le\kappa\le\nu} \Big\{\sup\{\tau \ge 0 \mid F(\tau) \le \kappa\} \\
&\qquad\qquad + \sup\{s \ge 0 \mid G(s) \le \nu - \kappa\}\Big\} \\
&= \sup_{0\le\kappa\le\nu} \Big\{\sup\{t \ge 0 \mid t = \tau + s, \tau \ge 0, s \ge 0 \\
&\qquad \text{and } F(\tau) \le \kappa \text{ and } F(x) \le \kappa \text{ and } G(s) \le \nu - \kappa\}\Big\} \\
&\le \sup\{t \mid t = \tau + s, \tau \ge 0, s \ge 0 \\
&\qquad\qquad \text{and } \max[F(\tau), F(x)] + G(s) \le \nu\} \\
&= \sup\{t \mid \forall \tau, 0 \le \tau \le t : \max[F(\tau), F(x)] + G(t-\tau) \le \nu\} \\
&= \sup\Big\{t \mid \sup_{0\le\tau\le t} \{\max[F(\tau), F(x)] + G(t-\tau)\} \le \nu\Big\} \\
&\le \sup\Big\{t \mid \sup_{x\le\tau\le t} \{F(\tau) + G(t-\tau)\} \le \nu\Big\} \\
&= \sup\{t \mid F \underset{x}{\otimes} G(t) \le \nu\} \\
&= \big(F \underset{x}{\otimes} G\big)^{\uparrow}(\nu)\,.
\end{aligned}
$$

After inserting the definition of the pseudo-inverses (first step) we express the sum of two suprema inside a single supremum (second step). In the third step, we summarize the inequalities, thereby relaxing the constraint on t. Doing so, we eliminate any appearance of κ, which allows us to drop the outer supremum. Then, we substitute $s = t - \tau$ (fourth step) and express the quantifier by a supremum (fifth step). The next line uses that

$$\sup_{0\leq\tau\leq t}\{\max[F(\tau), F(x)] + G(t-\tau)\} \geq \sup_{x\leq\tau\leq t}\{F(\tau) + G(t-\tau)\},$$

since both F and G are non-decreasing function. Replacing the first expression by the first therefore relaxes the constraint on t. The remaining steps merely insert the $\underset{x}{\otimes}$-operator and the definition of the upper pseudo-inverse. $\square$

Theorem 11.5. If S is an adaptive min-plus service curve, then $S^{\uparrow}$ is an adaptive max-plus service curve. Likewise, if γ_S is an adaptive max-plus service curve, then $\gamma_S^{\downarrow}$ is an adaptive min-plus service curve.

We provide a proof of the mapping of an adaptive service curve from the time domain to the space domain. The proof for the other direction is analogous.

Proof. For an adaptive min-plus service curve S as given by Definition 7.3, we pick two values κ and ν with $\kappa < \nu$ and set

$$t_\nu = T_D(\nu) = D^{\uparrow}(\nu),$$
$$t_\kappa = T_D(\kappa) = D^{\uparrow}(\kappa).$$

Consider t_ν and $t_\kappa + \varepsilon$, where $\varepsilon > 0$. With property *(P4)* we have

$$\begin{aligned} D(t_\nu) &= D(D^{\uparrow}(\nu)) \leq \nu, \\ D(t_\kappa+\varepsilon) &= D(D^{\uparrow}(\kappa)+\varepsilon) > \kappa. \end{aligned} \tag{11.1}$$

According to (9.3), for $t = t_\nu$ and $s = t_\kappa + \varepsilon$, a min-plus adaptive service curve satisfies one of the two inequalities

$$D(t_\nu) \geq D(t_\kappa+\varepsilon) + S(t_\nu - t_\kappa - \varepsilon),$$
$$D(t_\nu) \geq A \underset{t_\kappa+\varepsilon}{\otimes} S(t_\nu).$$

Suppose the first inequality is satisfied. Then, with (11.1) we get

$$\nu - \kappa \geq S(t_\nu - t_\kappa - \varepsilon) \xrightarrow[\varepsilon\to 0]{} S(t_\nu - t_\kappa),$$

where, for the limit $\varepsilon \to 0$, we use that S is left-continuous. Applying $S^{\uparrow}$ to both sides of the inequality, we derive

$$\begin{aligned} S^{\uparrow}(\nu - \kappa) &\geq S^{\uparrow}(S(t_\nu - t_\kappa)) \\ &\geq t_\nu - t_\kappa \\ &= T_D(\nu) - T_D(\kappa)\,, \end{aligned}$$

where we arrived at the second line with ***(P2)***. Hence, the first condition of an adaptive max-plus service curve from (7.6) is satisfied.

Now suppose that the second inequality of (9.3) holds, that is, $D(t_\nu) \geq A \underset{t_\kappa+\varepsilon}{\otimes} S(t_\nu)$. With property ***(P6)*** we then have

$$(A \underset{t_\kappa+\varepsilon}{\otimes} S)^{\uparrow}(D(t_\nu)) \geq t_\nu\,. \tag{11.2}$$

With (11.1) and Lemma 11.4, and setting $A^{\uparrow} = T_A$, we get

$$T_A \underset{A(t_\kappa+\varepsilon)}{\overline{\otimes}} S^{\uparrow}(\nu) \geq T_D(\nu)\,,$$

where we used the fact that $\big(A \underset{t_\kappa+\varepsilon}{\otimes} S\big)^{\uparrow}$ is a non-decreasing function. Next we derive

$$\begin{aligned} T_A \underset{A(t_\kappa+\varepsilon)}{\overline{\otimes}} S^{\uparrow}(\nu) &\leq T_A \underset{D(t_\kappa+\varepsilon)}{\overline{\otimes}} S^{\uparrow}(\nu) \\ &\leq T_A \underset{\kappa}{\overline{\otimes}} S^{\uparrow}(\nu)\,, \end{aligned}$$

where the first inequality follows from $A \geq D$, and the second inequality uses (11.1). Since $D(t_\kappa + \varepsilon) = D(D^{\uparrow}(\kappa) + \varepsilon) > \kappa$ and $t_\nu = T_D(v)$, we obtain from (11.2) that

$$T_D(\nu) \leq T_A \underset{\kappa}{\overline{\otimes}} S^{\uparrow}(\nu)\,.$$

Summarizing, we have shown that, for an adaptive min-plus service S, the upper pseudo-inverse $S^{\uparrow}$ satisfies one of the inequalities in (7.6) for arbitrary ν and κ with $\kappa < \mu$. Therefore, $S^{\uparrow}$ meets the requirements for an adaptive max-plus service curve. We have skipped the case $\kappa = \nu$, which is trivial since $T_D(\nu) \leq T_A \underset{\kappa}{\overline{\otimes}} S^{\uparrow}(\nu)$ holds. $\square$

$$\underbrace{\begin{array}{cc} \gamma_{S_1} \text{ adaptive max-plus} & \gamma_{S_2} \text{ adaptive max-plus} \\ \Updownarrow & \Updownarrow \\ \gamma_{S_1}^{\downarrow} \text{ adaptive min-plus} & \gamma_{S_2}^{\downarrow} \text{ adaptive min-plus} \end{array}}$$

$$\begin{array}{c} \Downarrow \\ \gamma_{S_1}^{\downarrow} \otimes \gamma_{S_2}^{\downarrow} \text{ adaptive min-plus} \\ \Updownarrow \\ (\gamma_{S_1} \overline{\otimes} \gamma_{S_2})^{\downarrow} \text{ adaptive min-plus} \\ \Updownarrow \\ ((\gamma_{S_1} \overline{\otimes} \gamma_{S_2})^{\downarrow})^{\uparrow} \text{ adaptive max-plus} \\ \Updownarrow \\ \gamma_{S_1} \overline{\otimes} \gamma_{S_2} \text{ adaptive max-plus} \end{array}$$

Figure 11.1: Convolution of two adaptive max-plus service curves.

With the mapping of min-plus to adaptive max-plus service curves at hand, we can offer a quick proof that adaptive max-plus service curves can be concatenated. Recall that, in §7.3, we deferred this proof when we discussed the properties of adaptive max-plus service curves. In the min-plus network calculus, it is established that if S_1 and S_2 are adaptive min-plus service curves, then $S_1 \otimes S_2$ is also adaptive min-plus [26, Theorem 13]. This allows us to draw the diagram in Figure 11.1, which presents a proof sketch that the max-plus convolution $\gamma_1 \overline{\otimes} \gamma_2$ of two adaptive max-plus service curves γ_1 and γ_2 is also adaptive max-plus.

We skip the mapping of strict service curves between the time and space domains. Since strict service curves are deficient in delivering suitable delay guarantees, we find them to be less useful than adaptive service curves. Note that a mapping of strict service curves must consider that the definition of strictness in the time domain uses busy periods (where the time-domain backlog satisfies $B(t) > 0$), whereas strictness in the space domain involves busy sequences (where the space-domain delay satisfies $W(\nu) > 0$). As we will see in §11.5, busy periods are not always well aligned with busy sequences in the space domain. Hence, a mapping would be confined to network elements where busy sequences and busy periods coincide. An example where this happens is a work-conserving buffered link with fixed rate C with a right-continuous arrival time function T_A. In this case, as seen in the proof of Lemma 5.1, for every busy sequence $[\kappa, \mu]$ we have $B(t) > 0$ in the time interval $[T_A(\kappa), T_D(\mu))$.

11.3 Mapping of performance bounds

The mapping between the min-plus and max-plus network calculus allows us to compare bounds for departures, backlog, and delay at network elements that can be derived in the two frameworks. The bounds for the max-plus network calculus are given in Theorem 8.1, and those of the min-plus network calculus are found at the end of §9. The max-plus bounds are expressed for units of traffic, *i.e.*, the backlog and delay of a bit, and min-plus bounds are expressed as functions of time. Nonetheless, we find that the bounds under a time-domain and a space-domain analysis are in agreement.

We consider a network element with a lower max-plus service curve γ_S, and arrivals that are bounded by a max-plus traffic envelope λ_E, or by a min-plus lower service curve S and a min-plus traffic envelope E. The min-plus and max-plus characterizations are related by the lower and upper pseudo-inverses, that is, $E = \lambda_E^{\downarrow}, \lambda_E = E^{\uparrow}, S = \gamma_S^{\downarrow}$, and $\gamma_S = S^{\uparrow}$.

For the envelope of departures at the network element, we use the fact that traffic envelopes can be expressed in terms of a min-plus or max-plus convolution, as shown at the beginning of §11.1. For the min-plus traffic envelope of the departures, we have that $D \sim E \oslash S$ if and only if $D = D \otimes (E \oslash S)$. Taking the upper pseudo-inverse and using the fact that $T_D = D^{\uparrow}$, Theorem 10.3 allows us to derive

$$\begin{aligned} T_D &= \Big(D \otimes (E \oslash S)\Big)^{\uparrow} \\ &= T_D \,\overline{\otimes}\, (E \oslash S)^{\uparrow} \\ &= T_D \,\overline{\otimes}\, (E^{\uparrow} \,\overline{\oslash}\, S^{\uparrow}) \\ &= T_D \,\overline{\otimes}\, (\lambda_E \,\overline{\oslash}\, \gamma_S)\,. \end{aligned}$$

Since $T_D \sim \lambda_E \,\overline{\oslash}\, \gamma_S$ if and only if $T_D = T_D \,\overline{\otimes}\, (E \oslash S)$, we conclude that $D \sim E \oslash S$ if and only if $T_D \sim \lambda_E \,\overline{\oslash}\, \gamma_S$. Since max-plus traffic envelopes do not take negative values, we can add a non-negativity constraint and write $T_D \sim [\lambda_E \,\overline{\oslash}\, \gamma_S]^+$.

The backlog bound in the min-plus algebra is given by the min-plus deconvolution $E \oslash S(0)$, see (9.4). Using the mapping of operations in the min-plus and max-plus algebras, we can write

$$\begin{aligned} E \oslash S(0) &= \Big((E \oslash S)^{\uparrow} \Big)^{\downarrow}(0) \\ &= (E^{\uparrow} \,\overline{\oslash}\, S^{\uparrow})^{\downarrow}(0) \\ &= (\lambda_E \,\overline{\oslash}\, \gamma_S)^{\downarrow}(0) \\ &= \inf\{x \geq 0 \mid \lambda_E \,\overline{\oslash}\, \gamma_S(x) \geq 0\} \,. \end{aligned}$$

The last line, which inserts the definition of the lower pseudo-inverse, is identical to the bound on the arrival backlog B^a from (8.2). The restriction to values $x \geq 0$ is permitted since for $x < 0$, we have that $\lambda_E(x) = -\infty$, and, therefore, $\lambda_E \,\overline{\oslash}\, \gamma_S(x) = -\infty$.

For the delay bounds, we take the max-plus delay bound from (8.1), given by $-\lambda_E \,\overline{\oslash}\, \gamma_S(0)$ and derive

$$\begin{aligned} -\lambda_E \,\overline{\oslash}\, \gamma_S(0) &= -\Big((\lambda_E \,\overline{\oslash}\, \gamma_S)^{\downarrow} \Big)^{\uparrow}(0) \\ &= -(\lambda_E^{\downarrow} \oslash \gamma_S^{\downarrow})^{\uparrow}(0) \\ &= -(E \oslash S)^{\uparrow}(0) \\ &= -\sup\{x \mid E \oslash S(x) \leq 0\} \\ &= \inf\{x \geq 0 \mid E \oslash S(-x) \leq 0\} \,. \end{aligned}$$

The fourth line uses the definition of the upper pseudo-inverse. In the last line, we can add the constraint $x \geq 0$ since $E \oslash S(-x) > 0$ for $x < 0$. The last line is of course the delay bound from the min-plus algebra in (9.5).

11.4 Mapping of backlog and delay

We now relate the performance metrics of backlog and delays in the space and time domains. For the space domain, we defined the waiting time $W(\nu)$ in (5.1), and the arrival backlog $B^a(\nu)$ and departure backlog $B^d(\nu)$ in (5.2).

For the time domain with arrival and departure functions A and D at a network element, the backlog at time t, $B(t)$, is defined in (9.1) and the (virtual) delay $W(t)$ is defined in (9.2). To emphasize that the virtual delay is measured at an actual or imagined (virtual) arrival instant, we denote the virtual delay by $W^a(t)$, and refer to it as *virtual arrival delay*. Analogous to the departure backlog $B^d(\nu)$, we can also define the delay for departing traffic. Let

$$W^d(t) = \inf\{\tau > 0 \mid D(t) \geq A(t-\tau)\}$$

denote the *virtual departure delay* at time t. The delay $W^d(t)$ expresses the delay that an actual or imagined departure at time t has incurred at a network element.

Different from arrivals, departures, envelopes, service curves, and the performance bounds, the backlog in the time and space domains are not easily mapped to each other using one of the pseudo-inverses. In fact, an exact mapping is generally not feasible, and we are left with providing upper and lower bounds. We first discuss bounds for the backlog, and then address bounds for the delay.

Theorem 11.6.

(a) The time domain backlog $B(t)$ is bounded by

$$B^a(A(t)) \leq B(t) \leq B^d(D(t))\,.$$

(b) The space domain backlog $B^a(\nu)$ is bounded by

$$\nu - D(T_A(\nu)^+) \leq B^a(\nu) \leq \nu - D(T_A(\nu))\,.$$

In Figure 11.2 we illustrate that the bounds in Theorem 11.6 are not necessarily sharp. Figure 11.2(a) captures a scenario where $B(t) > 0$, yet $B^a(A(t)) = 0$. Figure 11.2(b) depicts a case where $B^d(D(t))$ is significantly larger than $B(t)$. Incidentally, note that $B(t) = B^d(D(t))$ holds in Figure 11.2(a), and $B(t) = B^a(A(t))$ in Figure 11.2(b).

The backlog in the space domain can be bounded by backlog expressions from the time domain when the arrival and departure functions satisfy additional continuity properties. This will be stated in a corollary that follows the proof.

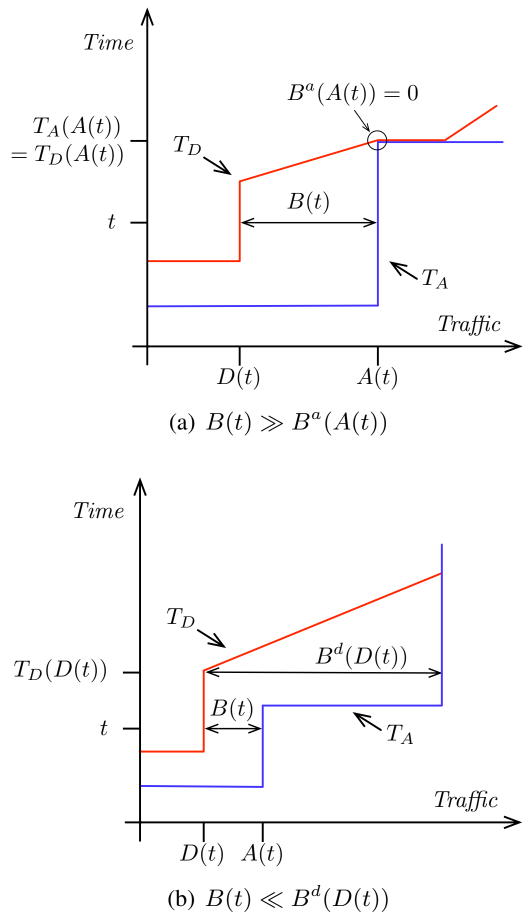


Figure 11.2: Relationship between the time-domain backlog and the space-domain backlog.

Proof.
(a) To get the lower bound for $B(t)$ we derive

$$\begin{aligned} B^a(A(t)) &= \sup\{\kappa \geq 0 \mid T_D(A(t)-\kappa) > T_A(A(t))\} \\ &\leq \sup\{\kappa \geq 0 \mid T_D(A(t)-\kappa) > t\} \\ &\leq \sup\{\kappa \geq 0 \mid A(t)-\kappa \geq D(t)\} \\ &= A(t) - D(t) \\ &= B(t)\,, \end{aligned}$$

where we used property ***(P2)*** in the second line and property ***(P8)*** in the third line. The upper bound follows from

$$\begin{aligned} B^d(D(t)) &= \inf\{\kappa \geq 0 \mid T_A(D(t)+\kappa) \geq T_D(D(t))\} \\ &\geq \inf\{\kappa \geq 0 \mid T_A(D(t)+\kappa) \geq t\} \\ &\geq \inf\{\kappa \geq 0 \mid D(t)+\kappa \geq A(t)\} \\ &= A(t) - D(t) \\ &= B(t)\,, \end{aligned}$$

where we again used properties ***(P2)*** and ***(P8)*** in the second and third line, respectively.
(b) We derive the lower bound on $B^a(\nu)$ as

$$\begin{aligned} B^a(\nu) &= \inf\{\kappa \geq 0 \mid T_D(\nu-\kappa)+\varepsilon \leq T_A(\nu)+\varepsilon\} \\ &\geq \inf\{\kappa \geq 0 \mid D(T_D(\nu-\kappa)+\varepsilon) \leq D(T_A(\nu)+\varepsilon)\} \\ &\geq \inf\{\kappa \geq 0 \mid \nu-\kappa \leq D(T_A(\nu)+\varepsilon)\} \\ &= \nu - D(T_A(\nu)+\varepsilon)\,. \end{aligned}$$

In the first line, we added a constant $\varepsilon > 0$ and applied the departure function D on both sides. Then we applied property ***(P4)***, and used that D is non-decreasing. Since the inequality also holds for the limit $\varepsilon \to 0$ for positive ε, we obtain

$$B^a(\nu) \geq \nu - D(T_A(\nu)^+)\,.$$

For the upper bound of $B^a(\nu)$ we derive

$$
\begin{aligned}
B^a(\nu) &= \sup\{\kappa \geq 0 \mid T_D(\nu-\kappa) > T_A(\nu)\} \\
&\leq \sup\{\kappa \geq 0 \mid D(T_D(\nu-\kappa)) \geq D(T_A(\nu))\} \\
&\leq \sup\{\kappa \geq 0 \mid \nu-\kappa \geq D(T_A(\nu))\} \\
&= \nu - D(T_A(\nu)),
\end{aligned}
$$

where we used property *(P4)* in the third line.

□

Observe that the bounds of Theorem 11.6(b) differ by the instantaneous departures that occur at time $T_A(\nu)$. Hence, the bounds are improved when the burstiness of departures is reduced. If the arrival and departure functions are continuous and do not allow any bursts, we can bound the backlog $B^a(\nu)$ by the time-domain backlog function.

Corollary 11.7.

(a) If A is continuous at $T_A(\nu)$, then $B(T_A(\nu)^+) \leq B^a(\nu) \leq B(T_A(\nu))$.

(b) If D is continuous at $T_A(\nu)$, then $B(T_A(\nu)) \leq B^a(\nu) \leq B(T_A(\nu)^+)$.

(c) If A and D are both continuous at $T_A(\nu)$, then $B(T_A(\nu)) = B^a(\nu)$.

Proof. By *(P4)*, we have $A(T_A(\nu)) \leq \nu \leq A(T_A(\nu)^+)$. This allows us to write the expression in Theorem 11.6(b) as

$$
A(T_A(\nu)) - D(T_A(\nu)^+) \leq B^a(\nu) \leq A(T_A(\nu)^+) - D(T_A(\nu)) .
$$

The first statement in the corollary follows from Theorem 11.6(b) since $A(T_A(\nu)) = A(T_A(\nu)^+)$ if A is continuous at $T_A(\nu)$. The second statement holds since $D(T_A(\nu)) = D(T_A(\nu)^+)$ if D is continuous at $T_A(\nu)$. The third statement follows since $B(T_A(\nu)) = B(T_A(\nu)^+)$ if both A and B are continuous at $T_A(\nu)$. □

The relationships between delays in the time and space domains are similar to those for the backlog, as stated in the next theorem.

Theorem 11.8.

(a) The space domain delay $W(\nu)$ is bounded by

$$W^a(T_A(\nu)) \leq W(\nu) \leq W^d(T_D(\nu)) \,.$$

(b) The arrival delay in the time domain is bounded by

$$T_D(A(t)^-) - t \leq W^a(t) \leq T_D(A(t)) - t \,.$$

Proof.
(a) The lower bound is obtained from

$$\begin{aligned} W^a(T_A(\nu)) &= \sup\{\tau \geq 0 \mid D(T_A(\nu)+\tau) < A(T_A(\nu))\} \\ &\leq \sup\{\tau \geq 0 \mid D(T_A(\nu)+\tau) \leq \nu\} \\ &\leq \sup\{\tau \geq 0 \mid T_A(\nu)+\tau \leq T_D(\nu)\} \\ &= T_D(\nu) - T_A(\nu) \\ &= W(\nu) \,. \end{aligned}$$

Here, we have taken advantage of *(P4)* in the second line and of *(P6)* in the third line. The derivation of the upper bound also uses properties *(P4)* and *(P6)*, and yields

$$\begin{aligned} W^d(T_D(\nu)) &= \inf\{\tau \geq 0 \mid D(T_D(\nu)) \geq A(T_D(\nu)-\tau)\} \\ &\geq \inf\{\tau \geq 0 \mid \nu \geq A(T_D(\nu)-\tau)\} \\ &\geq \inf\{\tau \geq 0 \mid T_A(\nu) \geq T_D(\nu)-\tau\} \\ &= T_D(\nu) - T_A(\nu) \\ &= W(\nu) \,. \end{aligned}$$

(b) We obtain the lower bound by using a constant $\varepsilon > 0$ and writing

$$\begin{aligned} W^a(t) &= \inf\{\tau \geq 0 \mid D(t+\tau) - \varepsilon \geq A(t) - \varepsilon\} \\ &\geq \inf\{\tau \geq 0 \mid T_D(D(t+\tau)-\varepsilon) \geq T_D(A(t)-\varepsilon)\} \\ &\geq \inf\{\tau \geq 0 \mid t+\tau \geq T_D(A(t)-\varepsilon)\} \\ &= T_D(A(t)-\varepsilon) - t, \end{aligned}$$

where the third line follows from *(P2)*. For $\varepsilon \to 0$ we obtain

$$W^a(t) \geq T_D(A(t)^-) - t \,.$$

The upper bound is also derived with *(P2)* with

$$\begin{aligned}
W^a(t) &= \sup\{\tau \geq 0 \mid D(t+\tau) < A(t)\} \\
&\leq \sup\{\tau \geq 0 \mid T_D(D(t+\tau)) \leq T_D(A(t))\} \\
&\leq \sup\{\tau \geq 0 \mid t+\tau \leq T_D(A(t))\} \\
&= T_D(A(t)) - t\,.
\end{aligned}$$

Since, by *(P2)*,

$$T_A(A(t)^-) \leq t \leq T_A(A(t))\,,$$

the bounds on the arrival delay can be written as

$$T_D(A(t)^-) - T_A(A(t)) \leq W^a(t) \leq T_D(A(t)) - T_A(A(t)^-)\,.$$

□

Corollary 11.9.

(a) If T_A is continuous at $A(t)$, then

$$W(A(t)^-) \leq W^a(t) \leq W(A(t))\,.$$

(b) If T_D is continuous at $A(t)$, then

$$W(A(t)) \leq W^a(t) \leq W(A(t)^-)\,.$$

(c) If both T_A and T_D are continuous at $A(t)$, then $W^a(t) = W(A(t))$.

Proof. Since, by *(P2)*, it holds that

$$T_A(A(t)^-) \leq t \leq T_A(A(t))\,,$$

the inequalities in Theorem 11.8 on the arrival delay can be written as

$$T_D(A(t)^-) - T_A(A(t)) \leq W^a(t) \leq T_D(A(t)) - T_A(A(t)^-)\,.$$

Then, if T_A is continuous at $A(t)$, we get $T_A(A(t)) = T_A(A(t)^-)$, which gives the first claim. If T_D is continuous at $A(t)$, we have $T_D(A(t)) = T_D(A(t)^-)$, which gives the second claim. If both T_A and T_D are continuous, we have $W(A(t)) = W(A(t)^-))$ and obtain the third claim. □

Similar derivations can be made for the departure backlog B^d in the space domain, and the departure delay W^d in the time domain.

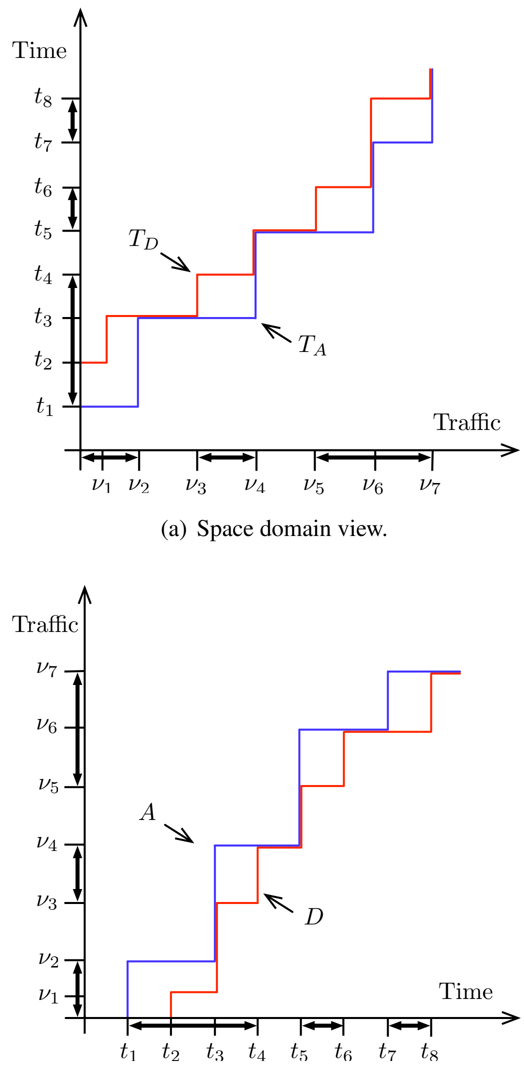
11.5 Mapping of busy periods and busy sequences

We saw that the space-domain and time-domain notions of backlog and delays are not linked by a one-to-one correspondence, at least in the presence of discontinuous arrivals and departures. The relationship between busy periods in the time domain and busy sequences in the space domain is even more complicated, and may occasionally even appear counterintuitive.

Let us quickly recall that a busy period is defined as a maximal time interval with a non-zero time-domain backlog, and that a busy sequence is a maximal continuous interval of bits with non-zero space-domain delay. That is, we have $B(t) > 0$ if t lies in a busy period, and $W(\nu) > 0$ if ν lies in a busy sequence. For an interval where the time-domain backlog or space-domain delay are zero, we speak, respectively, of an idle period or an idle sequence.

We want to explore how to map busy periods to busy sequences, and vice versa. More broadly, we seek a description of busy periods using concepts from the space domain, and of busy sequences using concepts from the time domain. We have collected a few clues so far. For the work-conserving buffered link we were able to relate a non-zero space-domain delay $W(\nu)$ to the time-domain backlog $B(t)$ via (5.4). This relationship, however, does not extend to general service curves. Another relationship can be obtained from the comparison of the space-domain and time-domain backlog in §11.4. Since $B(t) \geq B^a(A(t))$, we obtain that during an idle period where $B(t) = 0$, we also have $B^a(A(t)) = 0$. Note that the opposite is not true, that is, we can have $B^a(A(t)) = 0$ even if t lies in a busy period.

In general, busy periods are not easily characterized from the perspective of the space domain, and the same holds for busy sequences from the time domain. The arrival and departure scenario in Figure 11.3 illustrates some of the issues that may arise. The figure has two plots that each show the same scenario. Figure 11.3(a) presents a space domain view, with arrival and departure time functions T_A and T_D, and Figure 11.3(b) depicts the same scenario from the perspective of the time domain, with arrival and departure functions A and D. T_A and A are related by the upper and lower pseudo-inverses, and so are T_D and D. In both figures, double-headed arrows at the axes indicate busy periods and busy sequences. For example, the double-headed arrow connecting ν_3 and ν_4 indicates that $[\nu_3, \nu_4)$ is a busy sequence, and the



(a) Space domain view.

(b) Time domain view.

Figure 11.3: Busy periods and busy sequences. Both figures show an identical arrival and departure scenario. Double-headed arrows on the axes indicate busy sequences in the space domain and busy periods in the time domain.

double-headed arrow between t_5 and t_6 indicates a busy period.

The figures show two busy sequences and one idle sequence between 0 and ν_4. When we project arrivals and departures of these bits to the time domain, we see that the interval $(t_1, t_4]$ is covered by a single busy period. The arrivals and departures in the busy sequence $[\nu_3, \nu_4)$ map to a single time instant t_3, which is located in the busy period. Since A and D are left-continuous, we have $A(t_3) = \nu_2$, $D(t_3) = \nu_1$, yielding $B(t_3) = \nu_2 - \nu_1 > 0$. Since $B(t_3) > 0$, time t_3 lies in a busy period, but there are a lot of other things happening at time t_3. First of all, the entire backlog $\nu_2 - \nu_1$ departs at t_3. Then, we have an instantaneous arrival of $\nu_4 - \nu_2$ bits. Of these, $\nu_3 - \nu_2$ bits depart immediately without experiencing delay. In summary, at time t_3, the network element at which the arrivals and departures are observed, is busy (having nonzero backlog), idle (when the backlog has been completely cleared), and then busy again (since some, but not all arrivals occurring at time t_3 are released). With our left-continuous interpretation of traffic we have that $B(t_3) = \nu_2 - \nu_1$ and $B(t_3^+) = \nu_4 - \nu_3$, so the time-domain depiction does not capture that the bits in the interval $[\nu_2, \nu_3)$ do not experience a delay or a backlog.

A similar observation can be made for the busy sequence $[\nu_5, \nu_7)$, which maps in the time domain to a busy period $(t_5, t_6]$, an idle period $(t_6, t_7]$, and another busy period $(t_7, t_8]$. Here, all bits in $[\nu_5, \nu_7)$ experience a non-zero delay. The bits in this range fall into two groups. For the first group we have $W(\nu) = t_6 - t_5$ for $\nu \in [\nu_5, \nu_6)$, and for the second group we obtain $W(\nu) = t_8 - t_7$ for $\nu \in [\nu_6, \nu_7)$. All bits in the first group depart (at t_6), before the bits in the second group arrive (at t_7). In between, in the time interval $(t_6, t_7]$, the system has no backlog and, therefore, the time domain indicates an idle period. If we compute the time-domain delay $W(t)$ (which we referred to as $W^a(t)$ in 11.4) in the interval $(t_6, t_7]$, we obtain a delay of zero. Note, however, that this delay is a *virtual delay* which indicates the delay that an arrival at that time would experience. However, since there are no arrival events in $(t_6, t_7]$, the delay in this interval is never realized.

The above examples illustrate that the mapping between busy periods and busy sequences is not entirely straightforward when arrivals and departures are not continuous. While it is generally not feasible to express a busy period in terms of space-domain backlog and delay, or to express a busy sequence in terms of time-domain backlog and delay, it is possible to specify a busy period using space-domain functions, and to specify a busy sequence with functions from the time domain. The next lemma expresses a busy period using arrival and departure time functions from the space domain.

Lemma 11.10. The time-domain backlog satisfies $B(t) > 0$ if and only if there exist ν, μ with $\nu > \mu$ such that $T_A(\nu) < t \leq T_D(\mu)$.

Proof. We write the time-domain backlog in terms of pseudo-inverses as

$$\begin{aligned} B(t) &= A(t) - D(t) \\ &= \sup \{\nu \mid T_A(\nu) < t\} - \inf \{\mu \mid T_D(\mu) \geq t\} \\ &= \sup \{\nu - \mu \mid T_A(\nu) < t \leq T_D(\mu)\} \ . \end{aligned}$$

Then the lemma follows by definition of the supremum. $\square$

Verifying the condition in the lemma is quite cumbersome. Unfortunately, there are no obviously better alternatives. If $T_D(\nu^-) < T_A(\nu)$ for some ν, *i.e.*, the arrival time of ν is later than the departure time of the traffic immediately before it, we can conclude that the interval $[T_D(\nu^-), T_A(\nu)]$ is an idle period. However, this is only a sufficient condition for an idle period, since it is possible to have $B(t) = 0$ with $v = A(t)$, but $T_D(\nu^-) = T_A(\nu)$.

12

Scheduling for Rate and Delay Guarantees

A strength of the max-plus network calculus is its compatibility with traffic control algorithms that compute timestamps, *e.g.*, the earliest time when a packet is permitted to depart a greedy shaper or the latest time by which a packet must be transmitted. In the max-plus network calculus, these timestamps can be obtained directly from the value of a computed departure time function at a network element. In the min-plus network calculus, departure functions determine the amount of departing traffic for a given time value. Extracting timestamps for a departure time at a network element from such a function requires the solution of an inverse problem, *e.g.*, finding the earliest time t such the departure function $D(t)$ meets a given requirement.

We will use the max-plus network calculus to formulate scheduling algorithms that ensure rate and delay guarantees to flows. We assume a work-conserving link with rate C with traffic from several flows, each of which receives guarantees on the minimum rate and the maximum delay. The guarantees are lower bounds in that flows may receive additional service if surplus link capacity is available. This is different from the service discussed in §2, where a single flow acquired the entire link rate C without the possibility of obtaining additional capacity. The scheduling algorithms we construct are max-plus versions of the Service Curve Earliest Deadline First (SCED) scheduling algorithm [30]. The idea behind SCED is to assign to each arrival a target departure time, referred to as *deadline*, which is computed from a convolution of the arrivals with a reference service curve, and to transmit traffic in the order of deadlines. As long as the flows at the link satisfy a schedulability condition, SCED can guarantee that all traffic departs by the assigned deadlines. SCED scheduling offers a solution to the difficult problem of simultaneously offering rate and delay guarantees to a flow, where rate and delay bounds for each flow can be chosen arbitrarily.

12.1 Earliest Deadline First in the max-plus algebra

A scheduling algorithm, in short, *scheduler*, at a buffered link is a traffic control algorithm that selects backlogged traffic for transmission. In the Earliest Deadline First (EDF) scheduling algorithm, each traffic arrival ν from a flow j is tagged with a deadline $D\ell_j(\nu) \geq 0$, and the link transmits backlogged traffic in increasing order of deadlines. The deadline $D\ell_j(\nu)$ indicates the latest permitted departure time of ν. Traffic that departs after its assigned deadline, that is, $T_{D_j}(\nu) > D\ell_j(\nu)$, is said to experience a *deadline violation*. EDF scheduling for packet-level traffic can be implemented by maintaining a sorted transmission queue. In the most common version of EDF [19, 23], each traffic flow j is associated with a delay bound d_j. When a packet from flow j arrives at time t, it is assigned the deadline $t + d_j$. In the following, we consider a more general interpretation of EDF, where packet deadlines can be assigned arbitrarily as long as the deadline function $D\ell_j$ of flow j is non-decreasing. Note that a non-decreasing function $D\ell_j$ is compatible with a locally FIFO transmission order. Since there is no deadline violation if traffic departs exactly at the assigned deadlines, that is, $T_{D_j} = D\ell_j$, the deadline function $D\ell_j$ is right-continuous.

With packetized traffic, when a packet is assigned a deadline that is less than that of the packet currently in transmission, the packet with the lower deadline must wait until the transmission of the currently served packet is completed. This is referred to as *non-preemptive scheduling*. In our analysis of EDF, we first assume that scheduling is preemptive. With preemptive scheduling, when a packet arrives with a shorter deadline than that of the packet currently in transmission, the untransmitted portion of the packet in transmission is returned to the buffer, and the new arrival starts its transmission immediately. Preemptive and non-preemptive scheduling are identical for fluid-flow traffic without burst arrivals, since such traffic can be viewed as consisting of infinitesimally small packets. The difference between preemptive and non-preemptive scheduling in EDF and other scheduling algorithms is generally bounded by the transmission time of a packet with maximum size.

For a given scheduling algorithm, an arrival scenario with arrival time functions $T_{A_1}, \ldots, T_{A_N}$ and deadline assignments $D\ell_1, \ldots, D\ell_N$ is said to be *schedulable* if the transmission schedule does not result in a deadline violation, that is, $T_{D_j} \le D\ell_j$ for all j. With regard to schedulability, EDF is an optimal scheduling algorithm, as stated in the next lemma.

Lemma 12.1. Given N flows with arrival time functions $T_{A_1}, \ldots, T_{A_N}$ and deadline functions $D\ell_1, \ldots, D\ell_N$ at a work-conserving link with rate C. If there exists a scheduling algorithm for which the arrival scenario is schedulable, then the arrival scenario is also schedulable under EDF.

Proof. The proof is quite intuitive for packetized arrivals. Consider an arbitrary scheduling algorithm and an arrival scenario that does not result in a deadline violation for any packet. Let us consider the backlog at a time when the scheduler selects the next packet for transmission. We denote by p^* the packet in the backlog with the smallest current deadline and by p' the packet that the scheduler selects next for transmission. By the assumption of feasibility, there is no deadline violation. Now consider the selection according to EDF, *i.e.*, $p' = p^*$. In this case, the departure time of p^* is reduced, and it meets its deadline. The departure time of other packets is possibly increased by the transmission time of p^*. However, since their deadlines are larger than that of p^*, and p^* did not experience a deadline violation under the old schedule ($p' \neq p^*$), they will not experience a deadline violation with $p' = p^*$. Since a selection of $p' = p^*$ for each packet selection is equal to the EDF algorithm, the claim follows. The argument can be extended to fluid-flow arrivals, where, instead of packets, we consider sequences of backlogged traffic with equal deadlines. $\square$

Next we derive conditions for deadline violations under EDF scheduling at a work-conserving link with rate C for a set of flows with arrival time functions $T_{A_1}, \ldots, T_{A_N}$ and deadline functions $D\ell_1, \ldots, D\ell_N$. A deadline violation occurs if, for some flow i and some value μ, we have $D\ell_i(\mu) < T_{D_i}(\mu)$. For bit μ of flow i, we define the bit value of each flow j with a lower or equal deadline by

$$\nu_j^{i,\mu} = \sup\{\kappa \geq 0 \mid D\ell_j(\kappa) \leq D\ell_i(\mu)\}\,,$$

with $\nu_i^{i,\mu} = \mu$, and we define $\nu^{i,\mu} = \nu_1^{i,\mu} + \ldots + \nu_N^{i,\mu}$. Also, we define modified arrival time functions $T_{A_j}^{i,\mu}$ for each flow j as

$$T_{A_j}^{i,\mu}(\kappa) = \begin{cases} T_{A_j}(\kappa)\,, & \kappa \leq \nu_j^{i,\mu}\,, \\ \infty\,, & \kappa > \nu_j^{i,\mu}\,. \end{cases} \tag{12.1}$$

Let $T_A^{i,\mu}$ denote the modified arrival time function of the aggregate, which is given with (3.2) by

$$T_A^{i,\mu}(\kappa) = \inf_{\substack{\kappa_1,\ldots,\kappa_N \\ \kappa=\kappa_1+\ldots+\kappa_N}} \max_{j=1,\ldots,N} T_{A_j}^{i,\mu}(\kappa_j)\,.$$

We define modified departure time functions $T_{D_j}^{i,\mu}$ for each flow j and $T_D^{i,\mu}(\kappa)$ in the same way. The modified arrival and departure time functions eliminate all traffic with a deadline after $D\ell_i(\mu)$. Since, under EDF scheduling, a deadline violation of μ from flow i only depends on arrivals with deadlines no later than $D\ell_i(\mu)$, μ from flow i experiences a deadline violation under the arrival scenario $T_{A_1}^{i,\mu}, \ldots, T_{A_N}^{i,\mu}$ if and only if there is a deadline violation under $T_{A_1}, \ldots, T_{A_N}$. For a given $\nu \geq 0$ we define $\underline{\nu}^{i,\mu}$ as the beginning of the busy sequence under the aggregate modified arrival time function, which, in accordance with (5.3), is given by

$$\underline{\nu}_{i,\mu} = \sup\{\kappa \mid 0 \leq \kappa \leq \nu\,, T_A^{i,\mu}(\kappa) = T_D^{i,\mu}(\nu)\}\,.$$

With these definitions, we can give a condition for a deadline violation.

Lemma 12.2. Given arrival time functions $T_{A_1}, \ldots, T_{A_N}$ and deadline assignments $D\ell_1, \ldots, D\ell_N$ at a work-conserving link with rate C. A bit value μ of flow i experiences a deadline violation ($D\ell_i(\mu) < T_{D_i}(\mu)$) under preemptive EDF scheduling if and only if

$$\max_{j=1,\ldots,N} D\ell_j(\nu_j^{i,\mu}) < T_A^{i,\mu}(\underline{\nu}^{i,\mu}) + \frac{\nu^{i,\mu} - \underline{\nu}^{i,\mu}}{C}\,. \tag{12.2}$$

Proof. Suppose that $D\ell_i(\mu) < T_{D_i}(\mu)$. Consider the modified arrival scenario $T_A^{1,\mu}, \ldots, T_A^{N,\mu}$ with the corresponding departure time functions $T_D^{1,\mu}, \ldots, T_D^{N,\mu}$. Since $T_{D_i}(\mu) = T_D^{i,\mu}(\mu)$, we have $D\ell_i(\mu) < T_{D_i}^{i,\mu}(\mu)$. The value μ from flow i resides in a busy sequence of the modified arrival scenario that starts at $\underline{\nu}_{i,\mu}$. Therefore, in the time interval $I = [T_A^{i,\mu}(\underline{\nu}^{i,\mu}), T_{D_i}^{i,\mu}(\mu)]$, the link with EDF scheduling only transmits traffic with a deadline earlier than $D\ell_i(\mu)$. The amount of bits that must be transmitted in this time interval is $\nu^{i,\mu} - \underline{\nu}^{i,\mu}$, and the required transmission time of these bits is $\frac{\nu^{i,\mu} - \underline{\nu}^{i,\mu}}{C}$. If μ of flow i experiences a deadline violation, the required transmission time exceeds the length of interval I. Since, by construction, $D\ell_i(\mu) = \max_j D\ell_j(\nu_j^{i,\mu})$, (12.2) holds.

For the reverse direction, assume that (12.2) holds for a given μ from flow i. Then, between times $T_A^{i,\mu}(\underline{\nu}^{i,\mu})$ and $D\ell_i(\mu) = \max_j D\ell_j(\nu_j^{i,\mu})$, the amount of traffic in the arrival scenario $T_{A_1}, \ldots, T_{A_N}$ with a deadline less than or equal to $D\ell_i(\mu)$ is $\nu^{i,\mu} - \underline{\nu}^{i,\mu}$. With (12.2), these bits cannot be transmitted by time $D\ell_i(\mu)$. Therefore, there is a deadline violation at time $D\ell_i(\mu)$. □

Since the second part of the proof did not use properties of the EDF scheduling algorithm, any scheduling algorithm will result in a deadline violation at time $D\ell_i(\mu)$ if (12.2) is satisfied, in accordance with Lemma 12.1.

We next consider the non-preemptive EDF scheduling algorithm. The difference between preemptive and non-preemptive scheduling is that, immediately after time $T_A^{i,\mu}(\underline{\nu}^{i,\mu})$, the link may not be able to commence the transmission of traffic with an assigned deadline at or before $D\ell_i(\mu)$ until the packet that is currently in transmission has been completely transmitted. We denote the untransmitted portion of the packet in transmission at time $T_A^{i,\mu}(\underline{\nu}^{i,\mu})$ by $\ell^{i,\mu}$.

Lemma 12.3. Under the assumptions of Lemma 12.2, a bit value μ of flow i experiences a deadline violation ($D\ell_i(\mu) < T_{D_i}(\mu)$) under non-preemptive EDF scheduling if and only if

$$\max_{j=1,\ldots,N} D\ell_j(\nu_j^{i,\mu}) < T_A^{i,\mu}(\underline{\nu}^{i,\mu}) + \frac{\ell^{i,\mu}}{C} + \frac{\nu^{i,\mu} - \underline{\nu}^{i,\mu}}{C}\,. \tag{12.3}$$

Inserting an upper bound for the maximum packet size of any flow, denoted by $\ell^*_{\max}$, for $\ell^{i,\mu}$ in (12.3) yields a sufficient, but not necessary condition for a deadline violation at a non-preemptive EDF scheduler.

Proof. The proof of the non-preemptive condition follows that of Lemma 12.2. The difference with non-preemptive scheduling is that, at time $T_A^{i,\mu}(\underline{\nu}^{i,\mu})$, the link cannot immediately start the transmission of traffic with a deadline at or before $D\ell_i(\mu)$, if there is a packet currently in transmission, which has a deadline greater than $D\ell_i(\mu)$. Once the link completes the transmission of this packet at time $T_A^{i,\mu}(\underline{\nu}^{i,\mu}) + \frac{\ell^{i,\mu}}{C}$, it can attend to packets with an earlier deadline. This reduces the time interval I where the traffic transmits traffic with a deadline at or before $D\ell_i(\mu)$ to $I = [T_A^{i,\mu}(\underline{\nu}^{i,\mu}) + \frac{\ell^{i,\mu}}{C}, T_{D_i}(\mu)]$, while the number of bits that must be served in this interval remains unchanged at $\nu^{i,\mu} - \underline{\nu}^{i,\mu}$. $\square$

12.2 Max-plus Service Curve Earliest Deadline First

Suppose that we want to offer a flow a minimum guarantee in terms of a lower max-plus service curve γ_S. If T_A and T_D denote the arrival and departure time functions of the flow, then the guarantee is fulfilled if the inequality

$$T_D(\nu) \le T_A \,\overline{\otimes}\, \gamma_S(\nu)$$

is satisfied for all $\nu \ge 0$. The objective to satisfy this inequality can be expressed by associating with each value ν a deadline $D\ell(\nu)$, set to

$$D\ell(\nu) = T_A \,\overline{\otimes}\, \gamma_S(\nu)\,, \tag{12.4}$$

and transmitting traffic in increasing order of deadlines. A scheduling algorithm that operates in this fashion is referred to as *max-plus SCED algorithm* or *max-plus SCED scheduler*.

With the deadline assignment of (12.4), the max-plus SCED algorithm ensures that γ_S is a lower service curve if and only if there is no deadline violation, that is,

$$T_D \le T_A \,\overline{\otimes}\, \gamma_S \quad \Longleftrightarrow \quad D\ell \ge T_D\,. \tag{12.5}$$

As a simple example, consider a guarantee to a flow j for a maximum delay $d_j > 0$, which can be achieved by a lower max-plus service curve

$\gamma_{S_j}(\nu) = d_j$. Here, the computation of deadlines for max-plus SCED is simply the sum of the arrival time and the delay guarantee, given by

$$D\ell_j(\nu) = T_{A_j} \overline{\otimes} \gamma_{S_j}(\nu) = T_{A_j}(\nu) + d_j \,, \tag{12.6}$$

which is equivalent to EDF scheduling with fixed delay bounds [19, 23].

Now consider a set of flows with arrival time functions T_{A_j}, traffic envelopes λ_{E_j}, and deadline functions $D\ell_j$ for $j = 1, \ldots, N$ at a work-conserving link with rate C. For a given scheduling algorithm, a condition, which ensures that all compliant arrival scenarios, *i.e.*, $T_{A_j} \sim \lambda_{E_j}$ for each j, are schedulable, is referred to *schedulability condition*. The next result states a schedulability condition for max-plus SCED scheduling, where each flow j is assigned a service curve γ_{S_j}. With (12.5), satisfying the schedulability condition of SCED guarantees that $\gamma_{S_1}, \ldots, \gamma_{S_N}$ are lower max-plus service curves. We first present the schedulability condition for a link with preemption, and then discuss the non-preemptive case. With preemption, the link always transmits traffic with the earliest deadline, even if it requires to interrupt the transmission of a packet. Without preemption, the link always completes an ongoing packet transmission.

Theorem 12.4 (SCHEDULABILITY CONDITION OF MAX-PLUS SCED WITH PREEMPTION)**.** Given a set of N flows at a work-conserving link with a fixed rate C with preemption that performs max-plus SCED scheduling. The deadlines of flow j ($j = 1, \ldots, N$) are computed with a function $\gamma_{S_j} \in \mathcal{T}_o$. The arrivals of each flow comply to a max-plus traffic envelope, that is $T_{A_j} \sim \lambda_{E_j}$ for each j. Then, the functions $\gamma_{S_1}, \ldots, \gamma_{S_N}$ are lower max-plus service curves for the flows, if

$$\inf_{\substack{\nu_1,\ldots,\nu_N \\ \nu=\nu_1+\ldots+\nu_N}} \max_{j=1,\ldots,N} \lambda_{E_j} \overline{\otimes} \gamma_{S_j}(\nu_j) \geq \frac{\nu}{C} \,, \quad \forall \nu \geq 0 \tag{12.7}$$

The condition is necessary if each envelope λ_{E_j} can be saturated, in the sense that there exist arrival time functions $T_{A_1}, \ldots, T_{A_N}$ such that $T_{A_j} = \lambda_{E_j}$ for each flow j.

Proof. Suppose that (12.7) holds and a bit value μ from flow j experiences a deadline violation. Then, according to Lemma 12.2, the condition in (12.2) is satisfied. Simplifying notation by dropping the super- and subscripts 'i, μ', the lemma provides us with tuples $(\nu_1, \ldots, \nu_N)$ and $(\underline{\nu}_1, \ldots, \underline{\nu}_N)$ with $\nu = \nu_1 + \ldots + \nu_N$ and $\underline{\nu} = \underline{\nu}_1 + \ldots + \underline{\nu}_N$, such that

$$\max_{j=1,\ldots,N} T_{A_j} \overline{\otimes} \gamma_{S_j}(\nu_j) < T_A(\underline{\nu}) + \frac{\nu - \underline{\nu}}{C}, \tag{12.8}$$

where we inserted the deadline function of the SCED scheduler from (12.4). Note that, due to dropping 'i, μ', T_{A_j} is in fact the modified arrival time function from (12.1). For flow j we now derive

$$\begin{aligned}
&T_{A_j} \overline{\otimes} \gamma_{S_j}(\nu_j) - T_A(\underline{\nu}) \\
&\quad \geq T_{A_j} \overline{\otimes} \gamma_{S_j}(\nu_j) - T_{A_j}(\underline{\nu}_j) \\
&\quad = \sup_{0 \leq \kappa \leq \nu_j} \{T_{A_j}(\kappa) + \gamma_{S_j}(\nu_j - \kappa) - T_{A_j}(\underline{\nu}_j)\} \\
&\quad \geq \sup_{\underline{\nu}_j \leq \kappa \leq \nu_j} \{T_{A_j}(\kappa) - T_{A_j}(\underline{\nu}_j) + \gamma_{S_j}(\nu_j - \kappa)\} \\
&\quad = \sup_{0 \leq \kappa \leq \nu_j - \underline{\nu}_j} \{T_{A_j}(\kappa + \underline{\nu}_j) - T_{A_j}(\underline{\nu}_j) + \gamma_{S_j}(\nu_j - \underline{\nu}_j - \kappa)\} \\
&\quad \geq \lambda_{E_j} \overline{\otimes} \gamma_{S_j}(\nu_j - \underline{\nu}_j).
\end{aligned} \tag{12.9}$$

The first inequality follows from (3.3). The second inequality restricts the range of the supremum. The third inequality follows since $T_{A_j} \sim \lambda_{E_j}$. With this derivation, we obtain from (12.8) that

$$\max_{j=1,\ldots,N} \lambda_{E_j} \overline{\otimes} \gamma_{S_j}(\nu_j - \underline{\nu}_j) < \frac{\nu - \underline{\nu}}{C}.$$

Hence, we have found a tuple $(\nu_1 - \underline{\nu}_1, \ldots, \nu_N - \underline{\nu}_N)$ with $\sum_{j=1}^{N}(\nu_j - \underline{\nu}_j) = \nu - \underline{\nu}$ that violates (12.7). Since this contradicts that the condition in (12.7) holds for all $\nu \geq 0$, the assumption of a deadline violation was false. With (12.5), we conclude that $\gamma_{S_1}, \ldots, \gamma_{S_N}$ are max-plus lower service curves.

To show necessity, assume that $T_{A_j} = \lambda_{E_j}$ for each flow j, and suppose that (12.7) is violated for a tuple $(\nu_1, \ldots, \nu_N)$ with $\nu = \nu_1 + \ldots + \nu_N$, that is,

$$\max_{j=1,\ldots,N} \lambda_{E_j} \overline{\otimes} \gamma_{S_j}(\nu_j) < \frac{\nu}{C}.$$

All bits of the tuple $(\nu_1, \ldots, \nu_N)$ have a deadline at or before $\max_{j=1,\ldots,N} \lambda_{E_j} \overline{\otimes} \gamma_{S_j}(\nu_j)$. Since the time required to transmit the bits is $\frac{\nu}{C}$, the violation of the condition indicates that some traffic has not been transmitted by the time of the latest deadline. Hence, there is a deadline violation. $\square$

Verifying the schedulability condition in Theorem 12.4 is not easy due to the need to compute the infimum subject to the condition $\nu = \nu_1 + \ldots + \nu_N$. The condition becomes more convenient once we map it to the min-plus algebra. Using Lemma 10.1(e), and Theorems 10.4(b) and 10.4(d), we can write (12.7) as

$$\sum_{j=1}^{N} \lambda_{E_j}^{\downarrow} \otimes \gamma_{S_j}^{\downarrow}(t) \leq Ct\,, \quad \forall t \geq 0\,. \tag{12.10}$$

This results in an equivalent, more straightforward, schedulability condition. The rephrased schedulability condition in (12.10) is in fact the schedulability condition of the min-plus SCED algorithm from [30]. In min-plus SCED, deadlines for arrivals are computed from the min-plus convolution $D \geq A \otimes S$. In the time domain, the deadlines of SCED for an arrival at time t, denoted by $D\ell(t)$, are computed by solving the inverse problem

$$D\ell(t) = \inf\{x > t \mid A \otimes S(x) \geq A(t)\}\,.$$

In [30], it is shown that this assignment does not result in a deadline violation if and only if $D \geq A \otimes S$. The schedulability condition for min-plus SCED, given in [30, Theorem III.2], states that a SCED link with rate C guarantees min-plus service curves $S_1, \ldots, S_N$ to a set of flows with min-plus traffic envelopes E_j if

$$\sum_{j=1}^{N} E_j \otimes S_j(t) \leq Ct\,, \quad \forall t \geq 0\,. \tag{12.11}$$

With $\lambda_{E_j}^{\downarrow} = E_j$ and $\gamma_j^{\downarrow} = S_j$, we see that (12.10) and (12.11) are identical.

With Lemma 12.3, the schedulability condition for a non-preemptive link with SCED scheduling is quickly obtained.

Corollary 12.5 (Schedulability condition of max-plus SCED without preemption)**.** Consider a set of N flows with max-plus traffic envelope λ_{E_j} for each flow $j = 1, \ldots, N$, which arrive to a work-conserving link with rate C without preemption that performs max-plus SCED scheduling. The functions $\gamma_{S_1}, \ldots, \gamma_{S_N}$ are lower max-plus service curves if

$$\inf_{\substack{\nu_1,\ldots,\nu_N \\ \nu=\nu_1+\ldots+\nu_N}} \max_{j=1,\ldots,N} \lambda_{E_j} \,\overline{\otimes}\, \gamma_{S_j}(\nu_j) \geq \frac{\nu + \ell^*_{\max}}{C}\,, \quad \forall \nu \geq 0\,, \tag{12.12}$$

where $\ell^*_{\max}$ is an upper bound on the packet size for all flows. The condition is necessary if the arrival time functions can saturate their traffic envelopes, and if there exists an additional flow k with $k \neq 1, \ldots, N$ with packet arrivals up to size $\ell^*_{\max}$ and without deadlines.

Observe that the difference between the schedulability conditions of preemptive and non-preemptive SCED is the transmission time of a packet with maximum size. The necessity of the schedulability condition assumes the existence of 'best effort' traffic, which is represented by an additional flow k whose arrivals have no deadline ($D\ell_k(\nu) = \infty$).

Proof. The proof proceeds in the same way as the proof of Theorem 12.4. We show sufficiency of the condition, by assuming that (12.12) holds and that a bit value μ from flow j experiences a deadline violation. Then, according to Lemma 12.3, the condition in (12.3) is satisfied. Simplifying notation by dropping the super- and subscripts 'i, μ', the lemma provides us with tuples $(\nu_1, \ldots, \nu_N)$ and $(\underline{\nu}_1, \ldots, \underline{\nu}_N)$ with $\nu = \nu_1 + \ldots + \nu_N$ and $\underline{\nu} = \underline{\nu}_1 + \ldots + \underline{\nu}_N$, such that

$$\max_{j=1,\ldots,N} T_{A_j} \,\overline{\otimes}\, \gamma_{S_j}(\nu_j) < T_A(\underline{\nu}) + \frac{\nu - \underline{\nu}}{C} + \frac{\ell^{i,\mu}}{C} \,. \tag{12.13}$$

Using (12.9) and (12.13) we obtain the inequality

$$\max_{j=1,\ldots,N} \lambda_{E_j} \,\overline{\otimes}\, \gamma_{S_j}(\nu_j - \underline{\nu}_j) < \frac{\nu - \underline{\nu}}{C} + \frac{\ell^{i,\mu}}{C} \,.$$

Since $\ell^{i,\mu} \leq \ell^*_{\max}$, we have found a tuple $(\nu_1 - \underline{\nu}_1, \ldots, \nu_N - \underline{\nu}_N)$ with $\sum_{j=1}^{N} (\nu_j - \underline{\nu}_j) = \nu - \underline{\nu}$ that contradicts (12.12), and, therefore, the assumption of a deadline violation is false. With (12.5), we then have that $\gamma_{S_1}, \ldots, \gamma_{S_N}$ are max-plus lower service curves.

Suppose that (12.12) is violated for a tuple $(\nu_1, \ldots, \nu_N)$ with $\nu = \nu_1 + \ldots + \nu_N$, that is,

$$\max_{j=1,\ldots,N} \lambda_{E_j} \,\overline{\otimes}\, \gamma_{S_j}(\nu_j) < \frac{\nu + \ell^*_{\max}}{C} \,.$$

We construct an arrival scenario, where $T_{A_j} = \lambda_{E_j} + \varepsilon$ for each flow j, where ε is less than the difference between the right hand side and the left hand side in the inequality. Furthermore, there is an arrival at time $t = 0$ from a packet with size $\ell^*_{\max}$ with no deadline. This packet will be in transmission at time ε and will complete transmission at time $\frac{\ell^*_{\max}}{C}$. All bits of the tuple $(\nu_1, \ldots, \nu_N)$ have a deadline at or before $\max_{j=1,\ldots,N} \lambda_{E_j} \overline{\otimes} \gamma_{S_j}(\nu_j) + \varepsilon$. On the other hand, the time to transmit the bits from the tuple as well as the packet of size $\ell^*_{\max}$ is $\frac{\nu+\ell^*_{\max}}{C}$. Hence, some bits of the tuple must incur a deadline violation.

□

We can provide a sufficient schedulability condition for max-plus SCED at a link with rate C, even if upper bounds on the arrivals in terms of traffic envelopes are not available or arrivals are not bounded.

Corollary 12.6 (SCHEDULABILITY CONDITION OF MAX-PLUS SCED WITHOUT INFORMATION ON ARRIVALS)**.** Given a set of N flows at a work-conserving link with a fixed rate C that performs max-plus SCED scheduling, with deadlines of flow j computed with $\gamma_{S_j} \in \mathcal{T}_o$. Then, $\gamma_{S_1}, \ldots, \gamma_{S_N}$ are lower max-plus service curves for the flows at a link with preemption, if

$$\inf_{\substack{\nu_1,\ldots,\nu_N \\ \nu=\nu_1+\ldots+\nu_N}} \max_{j=1,\ldots,N} \gamma_{S_j}(\nu_j) \geq \frac{\nu}{C}, \quad \forall \nu \geq 0 .$$

At a link without preemption, $\gamma_{S_1}, \ldots, \gamma_{S_N}$ are lower max-plus service curves, if

$$\inf_{\substack{\nu_1,\ldots,\nu_N \\ \nu=\nu_1+\ldots+\nu_N}} \max_{j=1,\ldots,N} \gamma_{S_j}(\nu_j) \geq \frac{\nu + \ell^*_{\max}}{C}, \quad \forall \nu \geq 0 .$$

Proof. The corollary follows from Theorem 12.4 and Corollary 12.5, since $\gamma_{S_j} \leq \lambda_j \overline{\otimes} \gamma_{S_j}$ for every $\lambda_j \in \mathcal{T}_o$. Alternatively, if there is no information on the arrivals, we can use $\lambda_j = \overline{\delta}$ as a worst-case envelope, which allows arbitrarily large arrivals. Then, the corollary follows from Theorem 12.4 and Corollary 12.5 with $\gamma_{S_j} = \overline{\delta} \overline{\otimes} \gamma_{S_j}$. □

Next we discuss how SCED provides rate guarantees to a flow. A rate guarantee can be realized with a lower max-plus service curve $\gamma_S(\nu) = \frac{\nu}{R}$, where $R > 0$ is the guaranteed rate. In general, SCED requires the computation of $T_A \overline{\otimes} \gamma_S(\nu)$ for each value of ν. When arrivals occur as discrete-sized packets, there is an easier method to compute the deadlines. The insight to this method can be found in §2, where we saw that the recursive expression for the packet departure times at a work-conserving buffered link with rate C given in (2.1) can be phrased in terms of a max-plus convolution of packet arrival times and the max-plus service curve of a fixed rate link, given by $\gamma_S(\nu) = \frac{\nu}{C}$.

We first recall some notation from §2. We denote by ℓ_k the size of the k-th packet and by $L_n = \sum_{k=1}^{n} \ell_k$ the total size of the first n packets, where we set $L_o = 0$. The data of the k-th packet is given by the interval $[L_{k-1}, L_k)$. When all bits of a packet arrive at once, we use $T_A^p(k)$ to denote the arrival time of the k-th packet, with $T_A(\nu) = T_A^p(k)$ if $L_{k-1} \le \nu < L_k$. The departure time of the k-th packet is denoted by $T_D^p(k)$. Since a packet departs when all its bits have departed, we have $T_D^p(k) = T_D(L_k^-)$. Also, we use $\ell_{\max}$ to denote the maximum packet size of the flow. (Note the difference between $\ell_{\max}$ and $\ell^*_{\max}$, which we earlier defined as an upper bound for the packet sizes from all flows. Clearly, $\ell_{\max} \le \ell^*_{\max}$).

For the service curve $\gamma_S(\nu) = \frac{\nu}{R}$, for any $\nu \ge 0$ belonging to the k-th packet ($L_{k-1} \le \nu < L_k$) we have for $k > 1$ that

$$\begin{aligned}
T_A \,\overline{\otimes}\, \gamma_S(\nu) &= \sup_{0 \le \kappa \le \nu} \left\{ T_A(\kappa) + \frac{\nu - \kappa}{R} \right\} \\
&= \max \Bigg[\sup_{0 \le \kappa < L_{k-1}} \left\{ T_A(\kappa) + \frac{\nu - \kappa}{R} \right\}, \\
&\qquad\qquad \sup_{L_{k-1} \le \kappa \le \nu} \left\{ T_A(\kappa) + \frac{\nu - \kappa}{R} \right\} \Bigg] \\
&= \max \Bigg[\sup_{0 \le \kappa < L_{k-1}} \left\{ T_A(\kappa) + \frac{L_{k-1} - \kappa}{R}) \right\} + \frac{\nu - L_{k-1}}{R}, \\
&\qquad\qquad T_A^p(k) + \frac{\nu - L_{k-1}^-}{R} \Bigg] \\
&= \max\{ T_A \,\overline{\otimes}\, \gamma_S(L_{k-1}^-), T_A^p(k) \} + \frac{\nu - L_{k-1}}{R} \,. \qquad (12.14)
\end{aligned}$$

For $k = 1$ we get

$$T_A \,\overline{\otimes}\, \gamma_S(\nu) = T_A^p(1) + \frac{\nu}{r} \,.$$

In the second line of (12.14), we divide the supremum into two parts. In the third line, we use the fact that $\gamma_S(\nu) = \frac{\nu}{R}$ is additive ($\gamma_S(\nu - \kappa) = \gamma_S(\nu -$

$L_{k-1}) + \gamma_S(L_{k-1} - \kappa))$ and continuous $(\gamma_S(\nu) = \gamma_S(\nu^-) = \gamma_S(\nu^+))$. Also, we have $T_A(\kappa) = T_A^p(k)$ for $L_{k-1} \le \kappa \le \nu$, because we have packetized arrivals. We therefore have

$$\sup_{L_{k-1} \le \kappa \le \nu} \{T_A^p(k) + \gamma_S(\nu - \kappa)\} = T_A(\nu) + \gamma_S(\nu - L_{k-1}) \,.$$

Hence, for a ν that belongs to the k-th packet, (12.14) expresses the max-plus convolution $T_A \overline{\otimes} \gamma_S(\nu)$ in terms of the max-plus convolution computed at the end of the previous packet $(T_A \overline{\otimes} \gamma_S(L_{k-1}^-))$, the arrival time of the current packet $(T_A^p(k))$, and the time to serve the bits of the current packet $(\gamma_S(\nu - L_{k-1}^-) = \frac{\nu - L_{k-1}}{R})$. If ν is part of the first packet, the convolution is the sum of the packet arrival time and the transmission time $\frac{\nu}{R}$.

Equation (12.14) gives rise to a deadline assignment for a max-plus SCED scheduler that guarantees a rate R, where deadlines for $\nu \in [L_{k-1}, L_k)$ are computed with

$$D\ell(\nu) = \max\{D\ell(L_{k-1}^-), T_A^p(k)\} + \frac{\nu - L_{k-1}}{R} \,, \tag{12.15}$$

and $D\ell(L_o^-) = -\infty$. A drawback of this deadline assignment is that each value of ν is assigned a different deadline, which is not practical. A simpler approach is to assign all bits of a packet the same deadline. Let $D\ell^p(k)$ denote the deadline assigned to all bits of packet k, and define $D\ell^p(k) = D\ell(L_k^-)$, that is, the deadline of a packet is set to the deadline computed from (12.15) for the end of the packet. Then we get

$$D\ell^p(k) = \max\{D\ell^p(k-1), T_A^p(k)\} + \frac{\ell_k}{R} \,, \tag{12.16}$$

with $D\ell^p(0) = -\infty$ Now consider that each $\nu \in [L_{k-1}, L_k)$ is assigned the packet deadline $D\ell^p(k)$. Clearly, the deadline assignment $D\ell(L_k^-)$ satisfies (12.4). For values $\nu < L_k$, we have $D\ell(\nu) < D\ell^p(k)$. Then, it is possible that $D\ell(\nu) < T_D(\nu) \le D\ell^p(k)$, that is, the departure time $T_D(\nu)$ meets the packet-level deadline even though it violates the deadline imposed by the service curve $\gamma_S(\nu) = \frac{\nu}{R}$. The maximal deviation of the departure time from the packet-level deadline is bounded by the difference between (12.16) and (12.15). Since $D\ell^p(k) - D\ell(\nu) < \frac{\ell_{\max}}{R}$, where $\ell_{\max}$ is the maximum packet size, we can account for the violation by adding a latency $\frac{\ell_{\max}}{R}$ to the service curve. Defining

$$\gamma_{S'}(\nu) = \gamma_S \overline{\otimes} \overline{\delta}_{\ell_{\max}/R}(\nu) = \frac{\nu}{R} + \frac{\ell_{\max}}{R} \,,$$

the convolution of (12.14) becomes

$$\begin{aligned} T_A \,\overline{\otimes}\, \gamma_{S'}(\nu) &= T_A \,\overline{\otimes}\, \gamma_S \,\overline{\otimes}\, \overline{\delta}_{\ell_{\max}/R}(\nu) \\ &= T_A \,\overline{\otimes}\, \gamma_S(\nu) + \frac{\ell_{\max}}{R} \\ &= \max\{T_A \,\overline{\otimes}\, \gamma_S(L^-_{k-1}), T^p_A(k)\} + \frac{\nu - L_{k-1}}{R} + \frac{\ell_{\max}}{R} \,. \end{aligned}$$

The deadline assignment for a $\nu \in [L_{k-1}, L_k)$ with $\gamma_{S'}$, denoted by $D\ell'(\nu)$, is then given by

$$D\ell'(\nu) = \max\left\{D\ell(L^-_{k-1}), T^p_A(k)\right\} + \frac{\nu - L_{k-1}}{R} + \frac{\ell_{\max}}{R} \,,$$

where the deadline $D\ell(L^-_{k-1})$ inside the maximum is computed with (12.14). The modified deadline gives us

$$D\ell^p(k) \le D\ell'(\nu) \,, \qquad \text{if } \; L_{k-1} \le \nu < L^-_k \,,$$

that is, satisfying the packet-level deadline also satisfies the deadline $D\ell'(\nu)$ of all bits belonging to the k-th packet. Hence, the deadline assignment from (12.16) realizes SCED with a max-plus lower service curve $\gamma_{S'}(\nu) = \frac{\nu}{R} + \frac{\ell_{\max}}{R}$.

The deadline assignment from (12.16) is known as the *VirtualClock* algorithm [34]. It provides an elegant recursion that substitutes an otherwise cumbersome computation of max-plus convolutions. Note, however, that the recursive formula is limited to service curves of the form $\gamma_S(\nu) = \frac{\nu}{R}$. By combining (12.6) and (12.16) we can devise a deadline scheme with packet-level arrivals for a latency-rate max-plus service curve $\gamma_S = \frac{\nu+\ell_{\max}}{R} + d$. This can be done by separately keeping track of the deadlines needed for the rate guarantee, and then adding the delay d to obtain the final deadline of the packet. We first compute the deadline for the rate guarantees $D\ell^p(k)$ with (12.16), and then compute the final deadline, denoted by $\overline{D\ell^p}(k)$, as

$$\overline{D\ell^p}(k) = D\ell^p(k) + d \,. \tag{12.17}$$

The computation of the deadline in two phases is justified since $T_A \,\overline{\otimes}\, (\gamma_S + d)(\nu) = T_A \,\overline{\otimes}\, \gamma_S(\nu) + d$ for any max-plus service curve γ_S.

We summarize the above derivations in the following theorem.

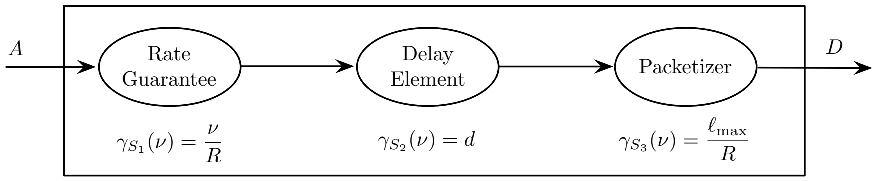


Figure 12.1: Components of a VirtualClock scheduler with a delay guarantee.

Theorem 12.7. Consider a flow with maximum packet size $\ell_{\max}$. A max-plus SCED scheduler that computes the deadlines with (12.16) and (12.17) realizes a lower max-plus service curve

$$\gamma_S(\nu) = \frac{\nu + \ell_{\max}}{R} + d\,, \qquad \forall \nu \geq 0\,,$$

as long as no arrival experiences a deadline violation.

The theorem describes a VirtualClock scheduler with a rate as well as a delay guarantee. The corresponding service curve in the min-plus network calculus is the latency-rate service curve $S(t) = [R(t-d) - \ell_{\max}]^+$. As shown in Figure 12.1 we can think about this scheduler as having three components: one for the rate guarantee ($\gamma_{S_1}(\nu) = \frac{\nu}{R}$), one for the delay guarantee ($\gamma_{S_2}(\nu) = d$), and one for the packetization ($\gamma_{S_3}(\nu) = \frac{\ell_{\max}}{R}$), with $\gamma_S = \gamma_{S_1} \overline{\otimes}\, \gamma_{S_2} \overline{\otimes}\, \gamma_{S_3}$.

Note that the formulation of Theorem 12.7 is only concerned with the computation of SCED deadlines. Whether these deadlines are met can be determined with the appropriate schedulability condition from Theorem 12.4, or Corollaries 12.5 and 12.6.

Remark: The implementation of VirtualClock with a rate guarantee R and a delay guarantee d requires a counter for each traffic flow to keep track of the value of $D\ell_r^p(k)$ in (12.16). The counter of a flow is interpreted as its virtual clock. Denoting the virtual clock of flow j by $\mathtt{VC}_j$, which is initialized to $\mathtt{VC}_j = 0$, the counter is updated upon the arrival of a packet with size ℓ bytes at time t to

$$\mathtt{VC}_j \leftarrow \max\{\mathtt{VC}_j, t\} + \frac{\ell}{R}\,,$$

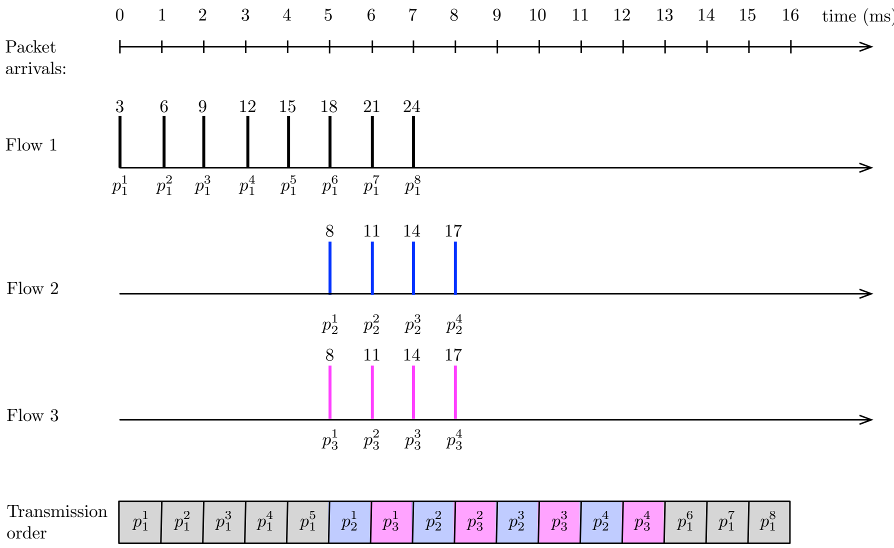


Figure 12.2: Transmission scenario with VirtualClock. There are packet arrivals from three flows that each obtain a rate guarantee of $R = \frac{1}{3}$ Mbps. The label p_j^k denotes the k-th packet from flow j. All packets have a size of 1000 bits. Deadlines, computed with (12.16), are shown for each packet. The transmission order shows the sequence of packet transmissions on a link with $C = 1$ Mbps.

and the packet is assigned the timestamp $\texttt{VC}_j + d$. VirtualClock transmits all packets in increasing order of timestamps.

In §7.3 we discussed a drawback when expressing rate guarantees with a max-plus lower service curve of the form $\gamma_S = \frac{\nu}{R}$. Since max-plus SCED enforces lower max-plus service curves, we expect that the VirtualClock algorithm inherits this drawback. We illustrate this in an example. Consider the packet-level arrival scenario in Figure 12.2. The scenario features three flows, where each flow receives a rate guarantee of $R = \frac{1}{3}$ Mbps. In the figure, packet arrivals are indicated by a vertical bar. We use the label p_j^k for the k-th packet from flow j. The deadlines of the packets, computed with (12.16), are plotted at the top of the vertical bars. In the scenario, flow 1 is active starting at time $t = 0$ ms, and flows 2 and 3 become active at time $t = 5$ ms. We assume that all packets have a size of 1000 bits. Let us compare the deadlines of the packets. Since flow 1 is active before the other flows, its deadlines increase by $\frac{1000 \text{ bits}}{1/3 \text{ Mbps}} = 3$ ms for each packet. By the time the first packets arrive from flow 2 and flow 3 at time $t = 5$ ms, the deadlines of flow 1 have reached 18 ms, whereas packets p_2^1 and p_3^1 receive a deadline of 8 ms. The bottom of Figure 12.2 depicts the transmission order of SCED at a link with rate 1 Mbps. With this rate, each packet requires a transmission time of 1 ms. All packets depart well before their deadlines. Packets $p_1^1, p_1^2, p_1^3, p_1^4$ are transmitted as soon as they arrive since there is no backlog from other traffic. When the other flows become active, a backlog begins to build up. Then, because transmissions occur in the order of deadlines, flow 2 and flow 3 take turns transmitting packets, while flow 1 is no longer selected. Flow 1 resumes transmission only after flow 2 and flow 3 packets have completed their transmissions.

Obviously, from the perspective of flow 1, this is not a desirable realization of a rate guarantee, since, in the interval $t \in [5, 13]$ ms, the output rate of flow 1 is zero. It appears that VirtualClock is unfair to flow 1, in that it punishes flow 1 for being active before the other flows. However, the issue at hand is not related to fairness, but to a weakness of the implementation of rate guarantees. When the SCED scheduler guarantees a lower max-plus service curve $\gamma_S(\nu) = \frac{\nu}{R}$, it guarantees the rate for the space interval $[0, \nu]$. When flow 1 has transmitted five packets at time $t = 5$ ms, the link has satisfied the guarantees for the time interval $[0, 15]$ ms. Flow 1 can be left out until $t = 15$ ms without a violation of its guarantee. The problem experienced by flow 1 in Figure 12.2 can be avoided by a scheduler that enforces an adaptive service curve, as we will see next.

12.3 Max-plus SCED with adaptive service guarantees

We have seen that rate guarantees offered by a max-plus lower service curve of the form $\gamma_S(\nu) = \frac{\nu}{R}$ are not satisfactory since the rate guarantees are amortized over the interval $[0, \nu]$. Thus, if the first μ bits ($\mu < \nu$) receive a higher service rate, and the remaining $\nu - \mu$ bits receive a lower rate, the guarantee $T_D(\kappa) \leq T_A \overline{\otimes} \gamma_S(\kappa)$ may still be satisfied for each $\kappa \in [0, \nu]$. A rate guarantee that holds for any sequence of bits can be expressed with an adaptive max-plus service curve. We can realize an adaptive service curves at a SCED scheduler by replacing the deadline formula from (12.4) with the right hand side of (7.5), yielding

$$D\ell(\nu) = \inf_{\mu \le \nu} \Big\{ \max \Big[T_D(\mu) + \gamma_S(\nu - \mu) \, , T_A \mathop{\overline{\otimes}}_{\mu} \gamma_S(\nu) \Big] \Big\} \, . \tag{12.18}$$

Note that $D\ell \in \mathcal{T}_o$, and, therefore, $D\ell(\nu) = -\infty$ if $\nu < 0$. With this deadline assignment, $D\ell \le T_D$ holds if and only if γ_S is an adaptive max-plus service curve that satisfies Definition 7.3. We refer to a SCED scheduler that computes deadlines with (12.18) as an *adaptive max-plus SCED scheduler*.

The computation of the deadline for an adaptive max-plus SCED scheduler appears cumbersome, but that is not always the case. For example, as shown in (7.8), computing (12.18) for a delay service curve $\gamma_S(\nu) = d$ results in the deadline $D\ell(\nu) = T_A(\nu) + d$, which is identical to the deadline computation for a lower service curve.

For an adaptive service curve $\gamma_S(\nu) = \frac{\nu}{R}$ offering a guaranteed rate R, the deadline computation can be simplified when traffic arrives in packets. This is similar to the deadline computation for rate guarantees in SCED discussed in §12.2. We first convey the main idea of the deadline computation using a packet-level description. Consider the departure time $T_D^p(n)$ of the n-th packet relative to the arrival time of the m-th packet ($1 \le m \le n$). If packets $m, m+1, \ldots, n$ arrive at or before their previous packet departs ($T_A^p(k) \le T_D^p(k-1)$ for all $m \le k \le n$), we want that a rate guarantee R satisfies

$$T_D^p(n) \le T_D^p(m-1) + \frac{\ell_m + \ldots + \ell_n}{R} \, .$$

If $T_A^p(m) > T_D^p(m-1)$, that is, if there is a gap between the departure of packet $m-1$ and the arrival of packet m, and no such gap for packets after packet m, a rate guarantee R should satisfy

$$T_D^p(n) \le T_A^p(m) + \frac{\ell_m + \ldots + \ell_n}{R} \, .$$

If $T_A^p(m+1) > T_D^p(m)$, meaning that there is a gap between the departure of the m-th packet and the arrival of the $(m+1)$-th packet, we expect that

$$T_D^p(n) \le T_A^p(m+1) + \frac{\ell_{m+1} + \ldots + \ell_n}{R} \, .$$

Lastly, if $T_A^p(n) > T_D^p(n-1)$, then a rate guarantee R should satisfy

$$T_D^p(n) \le T_A^p(n) + \frac{\ell_n}{R} \, .$$

At any time, exactly one of the inequalities above is satisfied. This allows us to summarize all inequalities above by

$$T_D^p(n) \le \max\Big[T_D^p(m-1) + \frac{1}{R}\sum_{j=m}^{n} \ell_j\,,\ \max_{m\le k\le n}\{T_A^p(k) + \frac{1}{R}\sum_{j=k}^{n}\ell_j\}\Big]\,.$$

Since this bound can be constructed for any $m \le n$, we have that

$$\begin{aligned} T_D^p(n) \le \min_{1\le m\le n}\Big\{&\max\Big[T_D^p(m-1) + \frac{1}{R}\sum_{j=m}^{n}\ell_j\,, \\ &\max_{m\le k\le n}\{T_A^p(k) + \frac{1}{R}\sum_{j=k}^{n}\ell_j\}\Big]\Big\} \qquad (12.19) \\ =:\ & F^p(n)\,, \end{aligned}$$

where we set $T_D^p(0) = -\infty$. (Setting $T_D^p(0) = 0$ also works since timestamps can safely be assumed to be positive.) Note that the above bound is structurally similar to the definition of an adaptive service curve in (7.5). We denote the right hand side of the inequality in (12.19) by $F^p(n)$, with $F^p(0) = 0$. Then we can express the bound on $T_D^p(n)$ in (12.19) as a recursion by deriving

$$\begin{aligned} T_D^p(n) &\le F^p(n) \\ &= \min\Big\{\max\Big[T_D^p(n-1), T_A^p(n)\Big] + \frac{\ell_n}{R}\,, \\ &\qquad \min_{1\le m\le n-1}\Big(\max\Big[T_D^p(m-1) + \frac{1}{R}\sum_{j=m}^{n-1}\ell_j, \\ &\qquad\qquad \max_{m\le k\le n-1}\{T_A^p(k) + \frac{1}{R}\sum_{j=k}^{n-1}\ell_j\}\,, T_A^p(n)\Big]\Big) + \frac{\ell_n}{R}\Big\} \\ &= \min\Big\{\max\Big[T_D^p(n-1), T_A^p(n)\Big]\,, \max\Big[T_A^p(n), F^p(n-1)\Big]\Big\} + \frac{\ell_n}{R} \\ &= \max\Big\{T_A^p(n), \min\Big[T_D^p(n-1), F^p(n-1)\Big]\Big\} + \frac{\ell_n}{R}\,. \end{aligned}$$

We can use this bound for the deadline assignment of a SCED scheduler. Denoting the deadline of the k-th packet by $D\ell^p(k)$ and setting $D\ell^p(k) = F^p(n)$ we obtain

$$D\ell^p(k) = \max\Big\{T_A^p(k), \min\{D\ell^p(k-1), T_D^p(k-1)\}\Big\} + \frac{\ell_k}{R}\,, \qquad (12.20)$$

with $D\ell^p(0) = T_D^p(0) = -\infty$. Compared to the deadline expression for a rate guarantee with a lower service curve from (12.16), there is an additional minimum involving the departure time $T_D^p(k-1)$ and the deadline of the previous packet. The additional minimum slows down the increase of the deadline $D\ell^p(k)$ in situations when a flow receives a service rate exceeding R, so that there is no penalty when new flows become active.

We can combine the rate guarantee with a delay guarantee. This is done in the same fashion as discussed in §12.2 by separately keeping track of the deadlines for the rate guarantee and adding the delay d. The deadline for the rate guarantee of packet k, $D\ell^p(k)$, is computed with (12.20). Then we determine the final deadline as

$$\overline{D\ell^p}(k) = D\ell^p(k) + d\,. \tag{12.21}$$

A SCED scheduling algorithms that computes deadlines with (12.20) and (12.21) is known as *Packet Scale Rate Guarantees* (*PSRG*) scheduler [4]. A difference between the computation of the deadlines in VirtualClock and PSRG is that the latter requires the departure time of the previous packet. Therefore, the determination of the deadline of a packet must sometimes be deferred until the departure time of the previous packet from the same flow.

The above packet-level derivation is not sufficient for arguing that the deadline assignment results in an adaptive max-plus service curve. For this, we have to show that the assignment satisfies (7.5) for every $\nu \geq 0$. The next theorem states that, as long as the service curve is increased by $\frac{\ell_{\max}}{R}$, the deadline assignment in (12.20) and (12.21) satisfies the requirement of an adaptive service curve. In §12.4, we present the proof of the theorem.

Theorem 12.8. Consider a flow with maximum packet size $\ell_{\max}$. An adaptive max-plus SCED scheduler that computes deadlines with (12.20) and (12.21) realizes an adaptive max-plus service curve

$$\gamma_S(\nu) = \frac{\nu + \ell_{\max}}{R} + d\,, \qquad \forall \nu \geq 0\,,$$

if no arrival experiences a deadline violation.

As with the VirtualClock scheduler, we can think about the scheduler as a concatenation of three network elements as shown in Figure 12.1.

We do not offer schedulability conditions to verify if a set of flows at a work-conserving link with rate C with an adaptive max-plus SCED scheduler results in a violation of the assigned deadlines. It is an open question whether the schedulability conditions in Theorem 12.4 and Corollaries 12.5 and 12.6 extend to adaptive max-plus SCED.

Remark: Similar to VirtualClock, the implementation of PSRG with a rate guarantee R and a delay guarantee d requires a counter $\texttt{VC}_j$ for each flow j that keeps track of the value $D\ell_j^p(k)$ in (12.20). The scheduler has a transmission queue of packets with assigned deadlines and transmits packets from that queue in increasing order of deadlines. For packets that are not assigned a deadline upon their arrival, PSRG additionally maintains a FIFO queue, $\texttt{FIFO}_j$, for packets from flow j without deadlines. Packets in this queue are timestamped with their arrival time. If the k-th packet from flow j with size ℓ arrives at time t after the previous packet has departed ($t > T^p_{D_j}(k-1)$), the counter of flow j is updated to

$$\texttt{VC}_j \leftarrow t + \frac{\ell}{R}\,,$$

where ℓ is the size of the k-th packet. Denoting by $\texttt{TS}_j^k$ the timestamp of the k-th packet from flow j, the packet is timestamped with its deadline by setting $\texttt{TS}_j^k = \texttt{VC}_j + d$ and added to the transmission queue. Otherwise, if $t \leq T^p_{D_j}(k-1)$, the packet is timestamped with its arrival time by setting $\texttt{TS}_j^k = t$ and added to $\texttt{FIFO}_j$. If the k-th packet from flow j departs at time t, and this happens after the arrival time of the next packet ($t > T^p_{A_j}(k+1)$), the $(k+1)$-th packet from flow j resides at the head of $\texttt{FIFO}_j$ and has been timestamped (upon its arrival) with its arrival time. In this case, the counter of flow j is updated to

$$\texttt{VC}_j \leftarrow \max\{\texttt{TS}_j^{k+1}, \min\{\texttt{VC}_j, t\}\} + \frac{\ell}{R}\,,$$

where ℓ is the size of the $(k+1)$-th packet . Then, packet $k+1$ is removed from $\texttt{FIFO}_j$, its timestamp is set to its deadline by setting $\texttt{TS}_j^k = \texttt{VC}_j + d$, and it is added to the transmission queue. Note that, with this implementation, the transmission queue contains at most one packet from each flow.

In Figure 12.3, we present an arrival and departure scenario that illustrates the operation of PSRG scheduling. The arrival scenario and all parameters are identical to the VirtualClock example in Figure 12.2. All packet arrivals are labeled with the deadline assigned according to (12.20). Except for the first

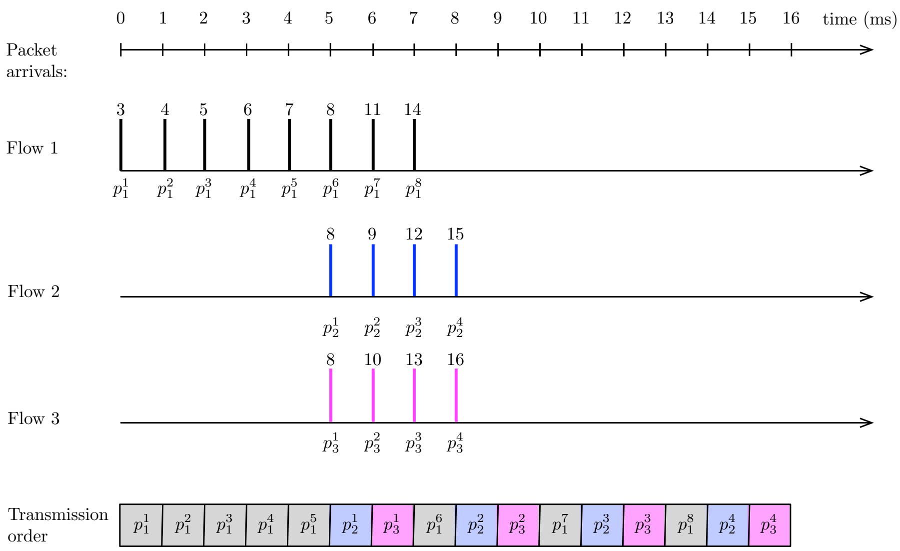


Figure 12.3: Transmission scenario with PSRG. The arrival scenario and all parameters are identical as in Figure 12.2.

arrival from a flow, the deadlines can be computed only when the previous packet from the same flow has departed. At time $t = 5$ ms, when flows 2 and 3 become active, the deadlines of packets p_1^6, p_2^1 and p_3^1 are identical. Hence, flow 1 is not penalized for having been active before the other flows. The transmission schedule at the bottom of the figure shows that flow 1 does not experience any disruption of its service. The transmission of packets with equal deadlines occurs in an arbitrary order. In the depicted transmission order, packets from flow 1 are selected last.

12.4 Proof of Theorem 12.8

We begin with (7.5), which bounds the departure time at a network element with an adaptive service curve. For an arbitrary value $\nu \geq 0$, we let $F(\nu)$ denote the right hand side of (7.5) for the service curve $\gamma_S(\nu) = \frac{\nu}{R}$, that is

$$F(\nu) = \inf_{\mu \leq \nu} \Big\{ \max \Big[T_D(\mu) + \frac{\nu - \mu}{R} \, , \sup_{\mu \leq \kappa \leq \nu} \{ T_A(\kappa) + \frac{\nu - \kappa}{R} \} \Big] \Big\} \, . \quad (12.22)$$

Assuming that ν belongs to packet k, that is, $\nu \in [L_{k-1}, L_k)$, we will express $F(\nu)$ in terms of $F(L_{k-1}^-)$, with the goal to create a recursive expression similar to the formula for the deadline in (12.20). We first consider values $\nu \geq L_1$, that is, ν belongs to the second or a later packet. We proceed by splitting the range of the infimum into three partially overlapping intervals: $L_{k-1} \leq \mu \leq \nu$, $L_{k-2} \leq \mu < L_{k-1}$, and $\mu < L_{k-1}$, where $k \geq 2$. Note that the second and third intervals overlap. We define three functions F_1, F_2, and F_3, each evaluating F for one of the intervals, and then take their minimum. The derivations take advantage of the next lemma.

Lemma 12.9. Consider a network element that offers an adaptive service curve $\gamma_S(\nu) = \frac{\nu}{R}$ and experiences packetized arrivals. Then, for each value ν that belongs to the k-th packet, that is, $L_{k-1} \leq \nu < L_k$, we have

$$T_D(\nu) = \inf_{L_{k-1} \leq \mu \leq \nu} \{T_D(\mu) + \frac{\nu - \mu}{R}\}\,.$$

Proof. Two values μ and ν that belong to the same packet k, with $L_{k-1} \leq \mu \leq \nu < L_k$, satisfy

$$\begin{aligned} T_D(\nu) &\leq \max\Big[T_D(\mu) + \frac{\nu - \mu}{R}\,, \sup_{\mu \leq \kappa \leq \nu} \{T_A(\kappa) + \frac{\nu - \kappa}{R}\}\Big] \\ &= T_D(\mu) + \frac{\nu - \mu}{R}\,. \end{aligned}$$

This follows since $T_A(\kappa) = T_A(\mu)$ for $\kappa \in [L_{k-1}, L_k)$ and $T_D(\mu) \geq T_A(\mu)$. Considering all values $\mu \in [L_{k-1}, \nu]$, we obtain

$$T_D(\nu) \leq \inf_{L_{k-1} \leq \mu \leq \nu} \{T_D(\mu) + \frac{\nu - \mu}{R}\} \leq T_D(\nu)\,,$$

where the second equality follows by setting $\mu = \nu$. $\square$

Now we evaluate the functions F_1, F_2, and F_3 for the three intervals. For the first interval we get

$$\begin{aligned} F_1(\nu) &= \inf_{L_{k-1} \leq \mu \leq \nu} \Big\{\max\Big[T_D(\mu) + \frac{\nu - \mu}{R}\,, \sup_{\mu \leq \kappa \leq \nu} \{T_A(\kappa) + \frac{\nu - \kappa}{R}\}\Big]\Big\} \\ &= \inf_{L_{k-1} \leq \mu \leq \nu} \Big\{T_D(\mu) + \frac{\nu - \mu}{R}\Big\} \\ &= T_D(\nu)\,. \end{aligned}$$

The second line uses that $T_A(\kappa) = T_A(\mu)$ for $\kappa \in [L_{k-1}, L_k)$, as well as $T_D(\mu) \geq T_A(\mu)$. We arrive at the third line with Lemma 12.9. For the second interval of the infimum, denoted by $F_2(\nu)$, we can derive

$$
\begin{aligned}
F_2(\nu) &= \inf_{L_{k-2} \leq \mu < L_{k-1}} \Big\{ \max\Big[T_D(\mu) + \frac{\nu-\mu}{R} \, , \sup_{\mu \leq \kappa \leq \nu} \{ T_A(\kappa) + \frac{\nu-\kappa}{R} \} \Big] \Big\} \\
&= \inf_{L_{k-2} \leq \mu < L_{k-1}} \Big\{ \max\Big[T_D(\mu) + \frac{\nu-\mu}{R} \, , \sup_{\mu \leq \kappa < L_{k-1}} \{ T_A(\kappa) + \frac{\nu-\mu}{R} \} \, , \\
&\qquad\qquad \sup_{L_{k-1} \leq \kappa \leq \nu} \{ T_A(\kappa) + \frac{\nu-\kappa}{R} \} \Big] \Big\} \\
&= \inf_{L_{k-2} \leq \mu < L_{k-1}} \Big\{ \max\Big[T_D(\mu) + \frac{L_{k-1}-\mu}{R} \, , \\
&\qquad\qquad T_A(\mu) + \frac{L_{k-1}-\mu}{R} \, , T_A^p(k) \Big] \Big\} + \frac{\nu - L_{k-1}}{R} \\
&= \inf_{L_{k-2} \leq \mu < L_{k-1}} \Big\{ \max\Big[T_D(\mu) + \frac{L_{k-1}-\mu}{R} \, , T_A^p(k) \Big\} + \frac{\nu - L_{k-1}}{R} \\
&= \max\Big[T_A^p(k), \inf_{L_{k-2} \leq \mu < L_{k-1}} \Big\{ T_D(\mu) + \frac{L_{k-1}-\mu}{R} \Big\} \Big] + \frac{\nu - L_{k-1}}{R} \\
&= \max\{ T_A^p(k), T_D^p(k-1) \} + \frac{\nu - L_{k-1}}{R} \, .
\end{aligned}
$$

In the second step, we extract the term $\frac{\nu - L_{k-1}}{R}$, and split the supremum into two parts. Since $\gamma_S(\nu) = \frac{\nu}{R}$ is continuous, we have $\gamma_S(\nu) = \gamma_S(\nu^-)$. The third step evaluates the two suprema. The next step uses $T_D \geq T_A$. Then we rearrange the infimum and the maximum. The last step takes advantage of Lemma 12.9 and uses $T_D^p(k-1) = T_D(L_{k-1}^-)$.

We now proceed with the derivation of F_3, the last part of the infimum.

$$
\begin{aligned}
F_3(\nu) &= \inf_{\mu < L_{k-1}} \Big\{ \max\Big[T_D(\mu) + \frac{\nu-\mu}{R} \, , \sup_{\mu \leq \kappa \leq \nu} \{ T_A(\kappa) + \frac{\nu-\kappa}{R} \} \Big] \Big\} \\
&= \inf_{\mu < L_{k-1}} \Big\{ \max\Big[T_D(\mu) + \frac{\nu-\mu}{R} \, , \sup_{\mu \leq \kappa < L_{k-1}} \{ T_A(\kappa) + \frac{\nu-\kappa}{R} \} \, , \\
&\qquad\qquad \sup_{L_{k-1} \leq \kappa \leq \nu} \{ T_A(\kappa) + \frac{\nu-\kappa}{R} \} \Big] \Big\}
\end{aligned}
$$

$$
\begin{aligned}
&= \inf_{\mu < L_{k-1}} \Big\{ \max\Big[T_D(\mu) + \frac{L_{k-1}-\mu}{R}\,, \\
&\qquad \sup_{\mu \le \kappa < L_{k-1}} \{ T_A(\kappa) + \frac{L_{k-1}-\kappa}{R} \}\,, T_A^p(k) \} \Big] \Big\} + \frac{\nu - L_{k-1}}{R} \\
&= \max\Big[T_A^p(k), \inf_{\mu < L_{k-1}} \Big\{ \max\Big(T_D(\mu) + \frac{L_{k-1}-\mu}{R}\,, \\
&\qquad \sup_{\mu \le \kappa < L_{k-1}} \{ T_A(\kappa) + \frac{L_{k-1}-\kappa}{R} \} \Big) \Big\} \Big] + \frac{\nu - L_{k-1}}{R} \\
&= \max(T_A^p(k), F(L_{k-1}^-)) + \frac{\nu - L_{k-1}}{R}\,.
\end{aligned}
$$

Here, we split the supremum into two parts, and use that one part evaluates to $T_A^p(n) + \frac{\nu - L_{k-1}}{R}$. Since, $\inf_{\mu < L_{k-1}} f(\mu) = \inf_{\mu \le L_{k-1}^-} f(\mu)$ and $\sup_{\mu \le \kappa < L_{k-1}} f(\kappa) = \sup_{\mu \le \kappa \le L_{k-1}^-} f(\kappa)$, we can replace the second term in the maximum (the infimum) by $F(L_{k-1}^-)$. We now summarize the three terms, and write (12.22) as

$$
\begin{aligned}
F(\nu) &= \min\Big(F_1(\nu)\,, F_2(\nu)\,, F_3(\nu) \Big) \\
&= \min\Big(T_D(\nu)\,, \max\{T_A^p(k), T_D^p(k-1)\} + \frac{\nu - L_{k-1}}{R}\,, \\
&\qquad\qquad \max(T_A^p(k), F(L_{k-1}^-)) + \frac{\nu - L_{k-1}}{R} \Big) \\
&\le \max\Big(T_A^p(k)\,, \min\{T_D^p(k-1), F(L_{k-1}^-)\} \Big) + \frac{\nu - L_{k-1}}{R}\,, \quad (12.23)
\end{aligned}
$$

where dropping the term $T_D(\nu)$ creates the inequality.

Now we consider $\nu \in [0, L_1)$, that is, ν belongs to the first packet. To compute (12.22) in this range, we split the infimum into two intervals $0 \le \mu \le \nu$ and $\mu < 0$. The infimum for the first interval, denoted by F_1, is

$$
\begin{aligned}
F_1(\nu) &= \inf_{0 \le \mu \le \nu} \Big\{ \max\Big[T_D(\mu) + \frac{\nu - \mu}{R}\,, \sup_{0 \le \kappa \le \nu} \{ T_A(\kappa) + \frac{\nu - \kappa}{R} \} \Big] \Big\} \\
&= \inf_{L_{k-1} \le \mu \le \nu} \{ T_D(\mu) + \frac{\nu - \mu}{R} \} \\
&= T_D(\nu)\,,
\end{aligned}
$$

where we used Lemma 12.9 for the second line with $L_o = 0$.

The infimum of the second interval, denoted by F_2, gives

$$\begin{aligned} F_2(\nu) &= \inf_{\mu<0}\Big\{\max\Big[T_D(\mu) + \frac{\nu-\mu}{R}\,,\ \sup_{\mu\le\kappa\le\nu}\{T_A(\kappa) + \frac{\nu-\kappa}{R}\}\Big]\Big\} \\ &= \inf_{\mu<0}\Big\{\sup_{\mu\le\kappa\le\nu}\{T_A(\kappa) + \frac{\nu-\kappa}{R}\}\Big\} \\ &= T_A^p(1) + \frac{\nu}{R}\,, \end{aligned}$$

where we used $T_D(\nu) = T_A(\nu) = -\infty$ for $\nu < 0$. Summarizing the results we obtain

$$F(\nu) = \min\big(F_1(\nu)\,, F_2(\nu)\big) \le T_A^p(1) + \frac{\nu}{R}\,.$$

With $L_o = 0$ and $T_D^p(0) = 0$, and by setting $F(L_o^-) = 0$, we have shown that (12.23) holds for all values $\nu \ge 0$.

The expression for $F(\nu)$ in (12.23) can be used for a deadline assignment of an adaptive max-plus SCED algorithm, by defining the deadline

$$D\ell(\nu) = \max\Big(T_A^p(k)\,, \min\{T_D^p(k-1), D\ell^p(k-1)\}\Big) + \frac{\nu - L_{k-1}}{R}\,, \tag{12.24}$$

for $\nu \in [L_{k-1}, L_k)$, where we use $D\ell^p(k) = D\ell(L_k^-)$ to denote the deadline of packet k, and with $D\ell^p(0) = T_D^p(0) = -\infty$.

If deadlines are computed for the end of the k-th packet ($D\ell(L_k^-)$) then the deadline in (12.24) is that of (12.20). On the other hand, if $\nu < L_k$, we have $D\ell(\nu) < D\ell^p(k)$. In this case it is possible, that the departure time meets the packet-level deadline ($T_D(\nu) \le D\ell^p(k)$), yet the deadline imposed by the adaptive service curve $\gamma_S(\nu) = \frac{\nu}{R}$ is violated ($D\ell(\nu) < T_D(\nu)$). The violation can be addressed by an adjustment of the service curve analogous to the derivation of Theorem 12.7. The difference between the continuous-space deadline $D\ell(\nu)$ in (12.24) and the packet-level deadline $D\ell^p(k)$ in (12.20) is bounded by $D\ell^p(k) - D\ell(\nu) < \frac{\ell_{\max}}{R}$, with $\ell_{\max}$ denoting the maximum packet size of the flow. Let us consider a modified service curve $\gamma_{S'}(\nu) = \frac{\nu}{R} + \frac{\ell_{\max}}{R}$ and recompute (7.5). For this, we let $F'(\nu)$ denote the right hand side of (7.5) with service curve $\gamma_{S'}(\nu)$ to obtain

$$F'(\nu) = \inf_{\mu \le \nu} \Big\{ \max \Big[T_D(\mu) + \frac{\nu - \mu}{R} + \frac{\ell_{\max}}{R} \, , \sup_{\mu \le \kappa \le \nu} \{ T_A(\kappa) + \frac{\nu - \kappa}{R} + \frac{\ell_{\max}}{R} \} \Big] \Big\}$$
$$= F(\nu) + \frac{\ell_{\max}}{R} \, ,$$

where $F(\nu)$ is as given in (12.22). With (12.23), we get for $\nu \in [L_{k-1}, L_k)$ that

$$F'(\nu) \le \max \Big(T_A^p(k) \, , \min\{ T_D^p(k-1), F(L_{k-1}^-) \} \Big) + \frac{\nu - L_{k-1}}{R} + \frac{\ell_{\max}}{R} \, ,$$

which suggests the deadline assignment

$$D\ell'(\nu) = \max \Big(T_A^p(k) \, , \min\{ T_D^p(k-1), D\ell^p(k-1) \} \Big) + \frac{\nu - L_{k-1}}{R} + \frac{\ell_{\max}}{R} \, ,$$

where we again use $D\ell^p(k) = D\ell(L_k^-)$. Comparing $D\ell'(\nu)$ with (12.20) yields

$$D\ell^p(k) \le D\ell'(\nu) \, , \qquad \text{if } \; L_{k-1} \le \nu < L_k^- \, .$$

Hence, a deadline assignment from (12.20) realizes an adaptive max-plus SCED algorithm with service curve $\gamma_{S'}(\nu) = \frac{\nu}{R} + \frac{\ell_{\max}}{R}$, when all deadlines are satisfied.

To account for the delay guarantee d, we use the adaptive service curve $\gamma_{S''}(\nu) = \frac{\nu}{R} + \frac{\ell_{\max}}{R} + d$ This is verified by recomputing (7.5) with $\gamma_{S''}$. If $F''(\nu)$ denotes the right hand side of (7.5) with $\gamma_{S''}$, we get $F''(\nu) = F'(\nu) + d$. Therefore, the deadline assignment

$$D\ell''(\nu) = D\ell'(\nu) + d$$

ensures that $D\ell^p(k)$ as given in (12.21) satisfies

$$D\ell^p(k) \le D\ell''(\nu) \, , \qquad \text{if } \; L_{k-1} \le \nu < L_k^- \, .$$

In summary, the deadline assignment of (12.20) and (12.21) ensures an adaptive max-plus SCED algorithm with service curve $\gamma_{S''}(\nu) = \frac{\nu}{R} + \frac{\ell_{\max}}{R} + d$, as long as no deadline violations occur.

12.5 Traffic shaping

The results on SCED scheduling with rate and delay guarantees can be applied to create efficient implementations of greedy shapers with a rate component. In §6, we presented a greedy shaper as a network element that enforces a max-plus traffic envelope λ_E on the departing traffic. Since a greedy shaper offers an exact max-plus service curve λ_E (Theorem 6.2), greedy shaping can be accomplished by a max-plus convolution, where $T_A \,\overline{\otimes}\, \lambda_E(\nu)$ determines the departure time of ν. For traffic envelopes that express a traffic rate, the departure times can be computed in the same fashion as deadlines at a SCED scheduler with rate guarantees. In particular, for packetized traffic arrivals, we can take advantage of the recursive expression for SCED deadlines with rate guarantees from §12.2.

A key difference between a SCED scheduler and a greedy shaper is that SCED realizes a lower service curve, whereas a greedy shaper implements an exact service curve. Therefore, the need for an adaptive guarantee does not arise in the context of greedy shaping. In SCED, the convolution $T_A \,\overline{\otimes}\, \gamma(\nu)$ is used to compute the deadline $D\ell(\nu)$. With a greedy shaper, we use the convolution to compute the earliest departure time of ν, which we refer to as the *release time* $R\ell(\nu)$, that is, we set

$$R\ell(\nu) = T_A \,\overline{\otimes}\, \lambda_E(\nu)\,. \tag{12.25}$$

In a perfect implementation of a greedy shaper we have $T_D(\nu) = R\ell(\nu)$ for every ν, that is, each bit departs at its release time.

Let us consider a greedy shaper for the rate-based max-plus traffic envelope $\lambda_E(\nu) = \frac{\nu}{r}$ with rate $r > 0$. If all arrivals are packetized, we can, as in (12.14), express the convolution by the recursive expression

$$T_A \,\overline{\otimes}\, \lambda_E(\nu) = \max\{T_A \,\overline{\otimes}\, \lambda_E(L_{k-1}^-), T_A^p(k)\} + \frac{\nu - L_{k-1}}{r}\,, \tag{12.26}$$

for every $\nu \in [L_{k-1}, L_k)$, with $L_o = 0$. With packetized traffic, all bits of a packet depart the shaper at the same time. Denoting by $R\ell^p(k)$ the release time of the k-th packet, we set $R\ell^p(k) = R\ell(L_k^-)$, that is, packet k departs at the release time assigned to the end of the packet. This yields the recursive expression

$$R\ell^p(k) = \max\{R\ell^p(k-1)), T_A^p(k)\} + \frac{\ell_k}{r}\,, \tag{12.27}$$

with $R\ell^p(0) = -\infty$. Collecting (12.25), (12.26), and (12.27), for $\nu \in [L_{k-1}, L_k)$ it holds that

$$R\ell^p(k) = R\ell(\nu) + \frac{L_k - \nu}{r}\,. \tag{12.28}$$

With packetized departures, all bits of the k-th packet are set to depart at the release time of the packet, that is, $T_D(\nu) = R\ell^p(k)$ for $\nu \in [L_{k-1}, L_k)$. With (12.28), the departure time of $\nu \in [L_{k-1}, L_k)$ is bounded by

$$R\ell(\nu) + \frac{L_k - \nu}{r} = T_D(\nu) \geq R\ell(\nu)\,.$$

Since $L_k - \nu \leq \ell_k \leq \ell_{\max}$, where $\ell_{\max}$ is the maximum packet size of the flow, we have with (12.25) that

$$T_A \,\overline{\otimes}\, \lambda_E \,\overline{\otimes}\, \overline{\delta}_{\ell_{\max}/r} \geq T_D \geq T_A \,\overline{\otimes}\, \lambda_E\,.$$

In other words, the release time assignment for a greedy shaper in (12.27) ensures an upper service curve $\frac{\nu}{r}$ and a lower service curve $\frac{\nu}{r} \,\overline{\otimes}\, \overline{\delta}_{\ell_{\max}/r} = \frac{\nu}{r} + \frac{\ell_{\max}}{r}$. While the exact service curve property $T_D = T_A \,\overline{\otimes}\, \lambda_E$ is lost by the packetization, the deviation from the exact service curve is less than $\frac{\ell_{\max}}{r}$.

If a max-plus traffic envelope of a greedy shaper has an earliness allowance e in addition to a rate r, the envelope is $\lambda_E(\nu) = [\frac{\nu}{r} - e]^+$. In this case, for computing $T_A \,\overline{\otimes}\, \lambda_E$, we maintain the recursion (12.27) for each packet, and subtract from it the earliness allowance. Using $\overline{R\ell^p}(k)$ to denote the release time of packet k after taking into account the earliness allowance, we have

$$\overline{R\ell^p}(k) = [R\ell^p(k-1) - e]^+\ , \tag{12.29}$$

with $R\ell^p(k)$ as given in (12.27). Recall that this greedy shaper is equivalent to a token bucket with rate r and burstiness er. For traffic envelopes with multiple tuples $(r_i, e_i)_{i=1,\ldots,N}$, *e.g.*, as given in (6.4) for $N = 2$, we compute release times separately for each traffic envelope $\lambda_{E_i}(\nu) = \left[\frac{\nu}{r_i} - e_i\right]^+$, and then take their maximum.

The above realization of a greedy shaper for max-plus traffic envelopes suggests an alternative take on a token bucket implementation. Recall from the Example in §11.1, that $\lambda_E(\nu) = \left[\frac{\nu - b}{r}\right]^+$ is the upper pseudo-inverse of the min-plus traffic envelope $E(t) = b + rt$. A greedy shaper that realizes the min-plus envelope E is referred to as token bucket. It enforces a long-term rate $r > 0$ and a maximum burst size of $b > 0$. The implementation of a token bucket often draws on an analogy to a bucket which is filled with tokens at rate r. Once the bucket contains b tokens, it is considered full, and no more tokens are added. A packet can depart only if there are sufficient tokens in the bucket. For a packet of size ℓ bits or bytes, the bucket must contain at least ℓ tokens. If the number of available tokens is insufficient, a packet must wait until the required number of tokens are available. The computation of the release times for a max-plus traffic envelope with a rate and an earliness allowance, as given by (12.29), suggests an analogy that is closer to the VirtualClock implementation described at the end of §12.2. When a packet of size ℓ bytes arrives at time t, the virtual clock `VC` is set to

$$\mathtt{VC} = \max\{\mathtt{VC}, t\} + \frac{\ell}{r},$$

with initial value $\mathtt{VC} = 0$. If $\mathtt{VC} - \frac{b}{r} \leq t$, then the packet departs immediately. Otherwise, the departure time of the packet is set $\mathtt{VC} - \frac{b}{r}$.

13

Related Literature

The textbooks by LeBoudec and Thiran [7] and Chang [9] have a comprehensive coverage of the deterministic network calculus. Both books discuss the min-plus and the max-plus versions of the network calculus, but use the min-plus network calculus as their main point of reference. Many of the concepts and results presented in §3–9 are covered in these textbooks, as well as in a survey paper [20]. Results on a probabilistic extension of the max-plus network calculus are presented in [31, 32, 33].

The earliest and most thorough development of a max-plus network calculus is presented by Chang [9, Chp. 6] and Chang and Lin [10]. The analysis takes a packet-level approach. Instead of using a single arrival time function T_A, arrivals to a network element are characterized by two functions, one describing the arrival time instants and the other the packet sizes. An example of such a characterization of arrivals is given in Figure 3.2. Then, a packet-level max-plus convolution operation is defined for $n \in \mathbb{N}_o$ as

$$F \odot_H G(n) = \max_{0 \leq k \leq n} \{F(k) + G(H(n) - H(k))\} ,$$

where F and G are non-decreasing real-valued functions, and $H(n)$ is a non-decreasing integer-valued function. The output at a work-conserving link with rate C is expressed in [9, Example 6.3.7] as

$$\tau^2(n) = \tau \odot_{L'} \gamma_S(n) + \frac{\ell'_n}{C}\,, \tag{13.1}$$

with $\gamma_S(\nu) = \frac{\nu}{C}$, where the index n indicates the $(n+1)$-th packet, $\tau^2(n)$ is the departure time of the $(n+1)$-th packet, $\tau(n)$ is the arrival time of the $(n+1)$-th packet, ℓ'_n denotes the size of the $(n+1)$-th packet, and $L'(n) = \sum_{k=0}^{n-1} \ell'_k$ is the cumulative size of the first n packets. Evaluating the convolution yields

$$\tau^2(n) = \max_{0\le k\le n} \left\{ \tau(k) + \frac{L'(n) - L'(k)}{C} \right\} + \frac{\ell'_n}{C}\,.$$

Note that packets are numbered starting with index zero for the first packet, with ℓ'_0 the size of the first packet, ℓ'_1 the size of the second packet, and so on. Relating this notation to our notation in §2 by setting

$$\tau^2(n) = T_D^p(n+1)\,, \quad \tau(n) = T_A^p(n+1)\,, \quad \ell'_{n-1} = \ell_n\,,$$

and evaluating (2.2) for the departure time of the $(n+1)$-th packet, we obtain

$$\begin{aligned} T_D^p(n+1) &= \max_{0\le k\le n} \Big\{ T_A^p(n+1-k) + \frac{\ell_{n+1-k} + \ldots + \ell_{n+1}}{C} \Big\} \\ &= \max_{0\le k\le n} \Big\{ T_A^p(k+1) + \frac{\ell_{k+1} + \ldots + \ell_n}{C} \Big\} + \frac{\ell_{n+1}}{C} \\ &= \max_{0\le k\le n} \Big\{ \tau(k) + \frac{1}{C}(L'(n) - L'(k)) \Big\} + \frac{\ell_{n+1}}{C} \\ &= \tau \odot_{L'} \gamma_S(n) + \frac{\ell'_n}{C}\,, \end{aligned}$$

with $\gamma_S(\kappa) = \frac{\kappa}{C}$. Hence, the departure function given in (2.2) is compatible with (13.1).

The $\odot_{L'}$-convolution expresses the waiting time of a packet in the buffer, before it begins transmission. In the special case of unit-sized packets, where $\ell_n = 1$ for all $n \ge 0$, the departure time is

$$\begin{aligned} T_D^p(n+1) &= \tau \odot_{L'} \gamma_S(n) + \frac{1}{C} \\ &= \max_{0\le k\le n} \Big\{ \tau(k) + \frac{n-k}{C} \Big\} + \frac{1}{C}\,, \end{aligned}$$

which is equivalent to the bit-level description of the departure time in (2.6). Note that the recovery of (2.6) necessitates that packet indices start with zero. With general packet sizes, the $\odot_{L'}$-operation is neither associative nor commutative, therefore, the resulting algebra is limited. On the other hand, if $\ell_n = 1$, the algebra of increasing functions endowed with operations $\vee$ and $\odot_{L'}$ is a dioid. The argument can be extended to equally sized packets, by adjusting the service function γ_S. The fact that dioid properties are recovered only in the special case of equal packet sizes is the main reason that works applying or extending the max-plus network calculus from [9, Chp. 6][10] assume that packet sizes are identical or unit-sized.

The application of upper and lower pseudo-inverses for relating the max-plus network calculus to the min-plus network calculus also originates in [9, Chp. 6][10]. Due to the packet-level characterization of the max-plus network calculus in [9, Chp. 6], the mapping of max-plus results into the min-plus algebra does not perfectly match those derived within the min-plus network calculus. As an example, the mapping of a packet-level max-plus traffic regulator of the form $\lambda(\nu) = \frac{1}{r}[\nu - b]^+$ maps to the min-plus traffic envelope $E(\tau) = b + r\tau + \ell_{\max}$, whereas an application of the lower pseudo-inverse results in $E(\tau) = b + r\tau$. The difference between these envelopes is a consequence of the packet-level interpretation of traffic algorithms, and not due to a rounding error or an inaccuracy of the analysis. The packet-level max-plus network calculus can be viewed as integrating each network element with a packetizer. An alternative explanation is that all bits belonging to the same packet receive the same timestamp, which, in a continuous-space view, is the timestamp assigned to the end of the packet. As discussed in the context of the VirtualClock and PSRG scheduling algorithms (in §12), ensuring that all bits are compliant with a given timestamp algorithm requires adding the transmission time of the longest packet.

The mapping between envelope functions and lower service curves in the min-plus and max-plus algebras are presented in [32]. The max-plus algebra adopts the packet-level description from [10]. The mapping deviates from the mapping of traffic envelopes and service curves in [9, Lemma 6.2.8] and [9, Lemma 6.2.8], respectively, which may be due to a different use of pseudo-inverses.

Obviously, the packet-level network calculus with the $\odot_{L'}$-convolution does not lead to an isomorphism, or even a homomorphism between max-plus and min-plus algebras. Even for the special case of unit-sized packets, the need to add terms $\ell_{\max} = 1$ or $\frac{\ell_{\max}}{R} = \frac{1}{R}$ to pseudo-inverses leaves the mapping between algebras imperfect. This may have led to conclusions about a lacking correspondence between the min-plus and max-plus network calculus, as cited in §1.

A special case of a mapping between backlog in the time domain and delay in the space domain is considered in [25]. Using a packet-level description as in [10] with unit size packets, and under the assumption of strictly increasing packet-level arrival and departure time functions T_A^p and T_D^p, the relationship for the n-th packet is

$$\begin{aligned} B(T_D^p(n)) &= A(T_D^p(n)) - A(T_D^p(n) - W(n)) \\ &= A(T_A^p(n) + W(n)) - A(T_A^p(n)) \,, \end{aligned}$$

where $W(n) = T_D^p(n) - T_A^p(n)$ is the delay associated with packet n. If T_A^p and T_D^p are strictly increasing, the functions are injective, and $A(T_A^p(n)) = D(T_D^p(n)) = n$ holds. In a continuous space domain, the requirement that T_A and T_D be strictly increasing is limiting since it does not allow burst arrivals or departures.

An extensive discussion of mappings between dioid algebras can be found in [3, §4.4]. The mappings exploit the lattice structure of complete dioids, and apply results from the residuation theory for lattices [5]. The properties of pseudo-inverse functions, summarized in Theorems 4.50 and 4.51 in [3, §4.4], correspond to some of the inequalities in properties *(P1)–(P4)* in §10. The mapping applies to all complete dioids, and is not limited to the dioids $(\mathcal{T}, \vee, \overline{\otimes})$ and $(\mathcal{F}, \wedge, \otimes)$. When applying the change of notation:

$$\Pi \longrightarrow F \,, \qquad \Pi^\sharp \longrightarrow F^\uparrow \,, \qquad \Pi^\flat \longrightarrow F^\downarrow \,,$$

and the change of terminology:

residual	$\longrightarrow$	upper pseudo-inverse,
dual residual	$\longrightarrow$	lower pseudo-inverse,
isotone mapping	$\longrightarrow$	non-decreasing function,
isotone and upper semi-continuous	$\longrightarrow$	right-continuous,
isotone and lower semi-continuous	$\longrightarrow$	left-continuous,

residuated mapping	$\longrightarrow$	left-continuous and non-decreasing function,
dual residuated mapping	$\longrightarrow$	right-continuous and non-decreasing function,

and using $\mathcal{I}$ to denote the identity of a dioid, the expressions can be matched as follows:

Equation in [3]:				Property:
(4.19)	$\Pi \circ \Pi^\sharp \leq \mathcal{I}$	$\longleftrightarrow$	$F(F^\uparrow(y)) \leq y$	*(P4)*
(4.20)	$\Pi^\sharp \circ \Pi \geq \mathcal{I}$	$\longleftrightarrow$	$x \leq F^\uparrow(F(x))$	*(P2)*
(4.23)	$\Pi \circ \Pi^\flat \geq \mathcal{I}$	$\longleftrightarrow$	$y \leq F(F^\downarrow(y))$	*(P3)*
(4.23)	$\Pi^\flat \circ \Pi \leq \mathcal{I}$	$\longleftrightarrow$	$F^\downarrow(F(x)) \leq x$	*(P1)*

The second inequality appearing in properties *(P1)–(P4)* does not follow from general residuation theory.

A set of papers [16, 17, 18] has applied residuation theory with the goal of deriving max-plus network calculus results from corresponding expressions in the min-plus network calculus. The papers describe several relationships of operations that involve minima, maxima, min-plus and max-plus convolutions, and min-plus and max-plus deconvolutions, but do not perform a mapping of the entire algebra.

The SCED algorithm was originally developed within the framework of the min-plus algebra [15, 30]. The max-plus version of SCED is discussed by Chang in [9, §6.4.2]. An advantage of max-plus SCED is that deadlines for traffic arrivals are directly obtained from a max-plus convolution, whereas min-plus SCED computes deadlines from a pseudo-inverse of a min-plus convolution. Chang also points out the similarity of max-plus SCED for rate guarantees and the VirtualClock algorithm.

Adaptive service curves are proposed and investigated in [2, 26] using the min-plus network calculus. The definition in (9.3) is a simplified version of the adaptive service curve in [2, 26]. The complete definition, given by

$$D(t) \geq \sup_{s \leq t} \Big\{ \min \Big[D(s) + S_1(t-s)\,, A \underset{s}{\otimes} S_2(t) \Big] \Big\}\,,$$

has two functions S_1 and S_2 to describe the service times. We have not used the more general definition previously, since it is not needed when expressing rate guarantees. Note that, with the original definition, the convolution of multiple adaptive service curves is more complex, but it retains the adaptive property.

The PSRG algorithm is proposed in [4] as a solution to a scheduling problem that ensures bounds on the service rate as well as delay variations (*delay jitter*). PSRG, as presented in [4], has a rate and a latency component. For didactic reasons, the exposition in §12.2 separates the two components, and first introduce PSRG as a scheduler for rate guarantees only. The relationship between PSRG and adaptive service curves is discussed in [7, Chp. 7], where PSRG is presented as the max-plus analogue to an adaptive min-plus service curve for rate guarantees. By presenting a definition of adaptive max-plus service and establishing the one-to-one relationship to the adaptive service curve in [2], we have made the analogy rigorous.

A number of studies, *e.g.*, [11, 12, 22, 27], have addressed similarities of and differences between network calculus and queueing theory as methods for performance evaluation of networked systems. In [22] the min-plus and max-plus convolutions are juxtaposed with the *Lindley equation* [24] as distinct methods for characterizing the behavior of a queueing system. We want to point out that the *Lindley equation* at a buffered link is in fact equivalent to the max-plus convolution from (2.8). The Lindley equation is a recursive expression that expresses the time that a packet resides in the buffer at a work-conserving link. It states that for $n \geq 2$,

$$\overline{W}_n = \max\{0, \overline{W}_{n-1} + S_{n-1} - A_{n-1}\}\,,$$

where

$\overline{W}_n$	is the waiting time (or queueing time) of the n-th packet arrival in the buffer before transmission,
S_{n-1}	is the service time (or transmission time) of the $(n-1)$-th packet, and
A_{n-1}	is the time elapsed between the arrival of the $(n-1)$-th and the n-th packet.

Since the first packet arrival does not have to wait in the buffer, we have $\overline{W}_1 = 0$. Assuming a buffered link with rate C, we have $S_n = \frac{\ell_n}{C}$. With the notation from §2, we have $A_n = T_A^p(n) - T_A^p(n-1)$. Denoting the delay of the n-th packet by W_n, we have

$$W_n = \overline{W}_n + S_n\,,$$

so we can rewrite the Lindley equation as

$$W_n = \max\{0, W_{n-1} - A_{n-1}\} + S_n \,.$$

With $W_n = T_D^p(n) - T_A^p(n)$ we can derive

$$\begin{aligned}
T_D^p(n) &= W_n + T_A^p(n) \\
&= \max\{0, W_{n-1} - (T_A^p(n) - T_A^p(n-1))\} + S_n + T_A^p(n) \\
&= \max\{T_A^p(n), (T_D^p(n-1) - T_A^p(n-1)) \\
&\qquad\qquad - (T_A^p(n) - T_A^p(n-1) + T_A^p(n))\} + S_n \\
&= \max\{T_A^p(n), T_D^p(n-1))\} + \frac{\ell_n}{C} \,.
\end{aligned}$$

This equation is equal to (2.1), which, in turn, is equal to the max-plus convolutions in (2.6) and (2.8).

14

Conclusions

We presented a continuous-space version of the max-plus network calculus and showed that it has an isomorphic relationship with a continuous-time min-plus network calculus. This clarifies a previously held belief that the two versions of the network calculus are closely related, but not equivalent. Being able to switch between the min-plus and max-plus points of view without a loss of accuracy provides flexibility in the characterization and analysis of traffic control algorithms. For example, the implementation of traffic control algorithms with timestamps is more efficient in a max-plus algebra, since it avoids the computation of inverse function values required in a min-plus setting. On the other hand, computing capacity requirements, which demands the consideration of aggregated traffic flows, is more straightforward using a min-plus algebra.

Our formulation of the max-plus network calculus dispenses with the assumption of equal packet sizes, which is found in most recent applications of the max-plus network calculus, *e.g.*, [25, 31, 32, 33]. Different from the min-plus network calculus, where results derived for continuous-time functions seem to hold in a discrete-time setting, we found differences between the continuous-space and discrete-space versions of the max-plus network calculus, *e.g.*, for the service curve of a work-conserving link (§2) and the definition of busy sequences (§5). It could be useful to quantify these differences.

Even though descriptions for arrivals, departures, and service are isomorphic, backlog and delay in the time domain and space domain could not be precisely mapped to each other. This is noteworthy since the worst-case bounds in the two domains are identical. It is an open question if this still holds when arrivals and service are governed by random processes.

The schedulability conditions for adaptive max-plus SCED scheduling at a work-conserving link with rate C is left as an open question for future research.

Appendices

A

Proofs of Lemma 4.1 and Lemma 4.2

We present proofs of the properties in Lemmas 4.1 and 4.2 for a continuous-space domain. We assume that F, G, and H are right-continuous and non-decreasing (in $\mathcal{T}$). When a property requires one-sided functions (in $\mathcal{T}_o$), we state so. In this way, we prove the properties in their broadest generality.

Closure of $\overline{\otimes}$. If the ranges of F and G are $\mathbb{R}_o^+ \cup \{\infty\} \cup \{-\infty\}$, then the sum $F(\kappa) + G(\nu - \kappa)$ cannot be a finite negative value for any choice of $\kappa \in \mathbb{R}$. Therefore, the range $F \overline{\otimes} G$ must be $\mathbb{R}_o^+ \cup \{\infty\} \cup \{-\infty\}$. For continuous-time processes, we show that the max-plus convolution of right-continuous functions is again right-continuous. We only need to assume that one of the functions, say G, is right-continuous. Consider the max-plus convolution

$$F \overline{\otimes} G(t + \varepsilon) = \sup_{s \in \mathbb{R}} \{F(s) + G(t - s + \varepsilon)\}.$$

G is right-continuous if and only if for $\varepsilon \to 0$ we have $G(t + \varepsilon) \to G(t)$. It follows that $F \overline{\otimes} G(t + \varepsilon) \to F \overline{\otimes} G(t)$ for $\varepsilon \to 0$. Hence, $F \overline{\otimes} G$ is right-continuous.

Now consider that F and G are non-decreasing. Select values $\nu \in \mathbb{R}$ and $\mu \geq 0$ and assume there exists a $\kappa^* \in \mathbb{R}$ such that

$$F \overline{\otimes} G(\nu + \mu) = F(\kappa^*) + G(\nu + \mu - \kappa^*).$$

We show that $F \otimes G$ is non-decreasing by exploiting that

$$F(\kappa^*) + G(\nu + \mu - \kappa^*) \geq F(\kappa^*) + G(\nu - \kappa^*) \geq F \mathbin{\overline{\otimes}} G(\nu)\,.$$

In general, if the supremum is not attained for a $\kappa^* \in \mathbb{R}$, we can use that for any $\varepsilon > 0$, we can find a κ^ε, such that $F \mathbin{\overline{\otimes}} G(\nu + \mu) + \varepsilon > F(\kappa^\varepsilon) + G(\nu + \mu - \kappa^\varepsilon) \geq F \mathbin{\overline{\otimes}} G(\nu)$. Letting $\varepsilon \to 0$, proves that $F \mathbin{\overline{\otimes}} G$ is non-decreasing.

The above establishes that $F \mathbin{\overline{\otimes}} G \in \mathcal{T}$. Now suppose that $F, G \in \mathcal{T}_o$. Then, for all $\nu < 0$,

$$\begin{aligned}
F \mathbin{\overline{\otimes}} G(\nu) &= \sup_{\kappa \in \mathbb{R}} \{F(\kappa) + G(\nu - \kappa)\} \\
&= \max\{\sup_{\kappa > \nu} \{F(\kappa) + G(\nu - \kappa)\}, \sup_{\kappa \leq \nu} \{F(\kappa) + G(\nu - \kappa)\}\} \\
&= \max\{\sup_{\kappa > \nu} \{F(\kappa) + (-\infty)\}, \sup_{\kappa \leq \nu} \{-\infty + G(\nu - \kappa)\}\} \\
&= -\infty\,.
\end{aligned}$$

Hence, $F \mathbin{\overline{\otimes}} G \in \mathcal{T}_o$.

Associativity. The following holds for any three F, G and H in $\mathcal{T}$. We first expand the expressions and rearrange the applications of the 'sup' operators. Then we perform a substitution $\eta = \kappa - \mu$.

$$\begin{aligned}
(F \mathbin{\overline{\otimes}} G) \mathbin{\overline{\otimes}} H(\nu) &= \sup_{\kappa \in \mathbb{R}} \{\sup_{\mu \in \mathbb{R}} \{F(\mu) + G(\kappa - \mu)\} + H(\nu - \kappa)\} \\
&= \sup_{\kappa \in \mathbb{R}} \{\sup_{\mu \in \mathbb{R}} \{F(\mu) + G(\kappa - \mu) + H(\nu - \kappa)\}\} \\
&= \sup_{\mu \in \mathbb{R}} \{F(\mu) + \sup_{\kappa \in \mathbb{R}} \{G(\kappa - \mu) + H(\nu - \mu - (\kappa - \mu))\}\} \\
&= \sup_{\mu \in \mathbb{R}} \{F(\mu) + \sup_{\eta \in \mathbb{R}} \{G(\eta) + H(\nu - \mu - \eta)\}\} \\
&= \sup_{\mu \in \mathbb{R}} \{F(\mu) + G \mathbin{\overline{\otimes}} H(\nu - \mu)\} \\
&= F \mathbin{\overline{\otimes}} (G \mathbin{\overline{\otimes}} H)(\nu)\,.
\end{aligned}$$

Commutativity. Let $F, G \in \mathcal{T}$. We show the property with a substitution $\kappa = \nu - \mu$.

$$\begin{aligned} F \,\overline{\otimes}\, G(\nu) &= \sup_{\mu \in \mathbb{R}} \{F(\mu) + G(\nu - \mu)\} \\ &= \sup_{\nu - \kappa \in \mathbb{R}} \{F(\nu - \kappa) + G(\kappa)\} \\ &= \sup_{\kappa \in \mathbb{R}} \{G(\kappa) + F(\nu - \kappa)\} \\ &= G \,\overline{\otimes}\, F(\nu)\,, \end{aligned}$$

where we have used that the supremum over $\nu - x \in \mathbb{R}$ is equivalent to the supremum over $x \in \mathbb{R}$.

Distributivity. Let $F, G, H \in \mathcal{T}$. For any choice $\nu \in \mathbb{R}$, we have

$$\begin{aligned} ((F \vee G) \,\overline{\otimes}\, H)(\nu) &= \sup_{\mu \in \mathbb{R}} \{\max\{F(\mu), G(\mu)\} + H(\nu - \mu)\} \\ &= \sup_{\mu \in \mathbb{R}} \{\max\{F(\mu) + H(\nu - \mu), G(\mu) + H(\nu - \mu)\}\} \\ &= \max\{\sup_{\mu \in \mathbb{R}} \{F(\mu) + H(\nu - \mu)\}, \sup_{\mu \in \mathbb{R}} \{G(\mu) + H(\nu - \mu)\}\} \\ &= ((F \,\overline{\otimes}\, H) \vee (G \,\overline{\otimes}\, H))(\nu)\,. \end{aligned}$$

The crucial step is in the third line, where we exploit that the order of maximization can be exchanged.

Neutral element $\overline{\delta}$.

$$\begin{aligned} F \,\overline{\otimes}\, \overline{\delta}(\nu) &= \sup_{\kappa \in \mathbb{R}} \{F(\kappa) + \delta(\nu - \kappa)\} \\ &= \sup_{\kappa \leq \nu} \{F(\kappa)\} \\ &= F(\nu)\,. \end{aligned}$$

The second line follows since $\overline{\delta}(\nu - \kappa) = -\infty$ for $\kappa \leq \nu$. The third line follows since F is non-decreasing.

Time shift.

$$\begin{aligned} F \overline{\otimes} \overline{\delta}_T(\nu) &= \sup_{\kappa \in \mathbb{R}} \{F(\kappa) + \overline{\delta}_T(\nu - \kappa)\} \\ &= \sup_{\kappa \in \mathbb{R}} \{F(\kappa) + \overline{\delta}(\nu - \kappa)\} + T \\ &= F(\nu) + T\,. \end{aligned}$$

Order preserving. Select values ν and $\nu + \mu$ ($\mu \geq 0$) and assume there exists an $\kappa^* \in \mathbb{R}$ such that

$$\begin{aligned} F \overline{\otimes} H(\nu) &= F(\kappa^*) + H(\nu - \kappa^*) \\ &\leq G(\kappa^*) + H(\nu - \kappa^*) \\ &\leq \sup_{\kappa \in \mathbb{R}} \{G(\kappa) + H(\nu - \kappa)\} \\ &= G \overline{\otimes} H(\nu)\,. \end{aligned}$$

When the supremum of the max-plus convolution is not attained, we can make a construction such that for any $\varepsilon > 0$, there exists a κ^ε, with $F \overline{\otimes} H(\nu) + \varepsilon \leq F(\kappa^\varepsilon) + H(t - \kappa^\varepsilon)$. The property follows by taking $\varepsilon \to 0$.

Boundedness of $\overline{\otimes}$. Since $G \in \mathcal{T}_o$, we have $G \geq \overline{\delta}$. With the order preserving property, we obtain for each $F \in \mathcal{T}$ that $F \overline{\otimes} G \geq F \overline{\otimes} \overline{\delta} = F$. If $F \in \mathcal{T}_o$, we have $F \overline{\otimes} F \geq F \overline{\otimes} \overline{\delta} = F$.

Existence of maximum. Recall that for functions $F, G \in \mathcal{T}_o$, the computation of the convolution $F \overline{\otimes} G(\nu)$ must only consider the interval $[0, \nu]$. In a discrete-space domain, the maximum exists since the elements in the interval is finite. With a continuous-space domain, the existence of the maximum can be quickly argued if we borrow a result from real analysis about upper semi-continuous functions. An *upper semi-continuous function* f is defined by the property that for all values of x,

$$\limsup f(x) \leq f(x)\,,$$

where '$\limsup$' denotes the limit superior.[1] This means that the function value at a point is never below the function values close to that point. The

[1] The limit superior for a sequence $\{x_n\}_{n=1,2,\ldots}$ with $\lim_{n\to\infty} x_n = x$ is defined as $\limsup f(x) = \lim_{n\to\infty} \sup_{m \geq n} f(x_m)$.

definition does not assume that f is non-decreasing. On the other hand, if a function is right-continuous and non-decreasing, it is upper semi-continuous. Likewise, a function is upper semi-continuous, if it is left-continuous and non-increasing. Now, a property of upper semi-continuous functions is that the supremum of the function over a finite interval is always attained. That is, if f is an upper semi-continuous function, then for any supremum over a finite interval $[a, b]$, there exists a value y^* with $a \leq y^* \leq b$ such that $f(y^*) = \sup_{a \leq y \leq b} f(y)$. If we consider the max-plus convolution

$$F \,\overline{\otimes}\, G(\nu) = \sup_{0 \leq \kappa \leq \nu} \{F(\kappa) + G(\nu - \kappa)\} ,$$

we have that $F(\kappa)$ is non-decreasing and right-continuous, $G(\nu - \kappa)$ (as a function of κ) is non-increasing and left-continuous. Therefore, both $F(\kappa)$ and $G(\nu - \kappa)$ are upper semi-continuous, and so is the sum $F(\kappa) + G(\nu - \kappa)$. Applying the property for upper semi-continuous functions gives that there exists an κ^* with $0 \leq \kappa^* \leq \nu$, such that $F \,\overline{\otimes}\, G(\nu) = F(\kappa^*) + G(\nu - \kappa^*)$.

The existence property can be used to simplify the proofs of some other properties above (*e.g.*, monotonicity), as long as $F, G \in \mathcal{T}_o$.

Operation $\oslash$ not closed. We provide a proof with a counter example. Consider the delay functions $\overline{\delta}_1, \overline{\delta}_2, \overline{\delta}_4 \in \mathcal{T}_o$ and $\nu \geq 0$. Then, we can write

$$\begin{aligned} \overline{\delta}_2 \,\overline{\oslash}\, \overline{\delta}_4(\nu) &= \inf_{\kappa \geq 0} \{\overline{\delta}_2(\nu + \kappa)\} - \overline{\delta}_4(\kappa)\} \\ &= \inf_{\kappa \geq 0} \{2 - 4\} \\ &= -2 . \end{aligned}$$

So, $\overline{\delta}_2 \,\overline{\oslash}\, \overline{\delta}_4 \notin \mathcal{T}_o$ since it violates non-negativity for $\nu \geq 0$.

We can show that $\overline{\oslash}$ is not even closed in $\mathcal{T}$. Define

$$F(\nu) = \begin{cases} -\infty , & \nu < 0 , \\ \overline{\delta}_1 , & 0 \leq \nu < 2 , \\ \overline{\delta}_2 , & \nu \geq 2 . \end{cases}$$

Clearly, $F \in \mathcal{T}_o$. If we compute $\overline{\delta}_4 \,\overline{\oslash}\, F$ at $\nu = 1$ and $\nu = 2$, we get

$$\begin{aligned} \overline{\delta}_4 \,\overline{\oslash}\, F(1) &= \overline{\delta}_4 \,\overline{\oslash}\, \overline{\delta}_1 = 3 , \\ \overline{\delta}_4 \,\overline{\oslash}\, F(2) &= \overline{\delta}_4 \,\overline{\oslash}\, \overline{\delta}_2 = 2 . \end{aligned}$$

This shows that $\overline{\delta}_4 \,\overline{\oslash}\, F \notin \mathcal{T}$ since it is not non-decreasing.

Operation $\oslash$ not associative. This is also shown using a counter example.

$$\begin{aligned}(\overline{\delta}_4 \,\overline{\oslash}\, \overline{\delta}_2) \,\overline{\oslash}\, \overline{\delta}_1 &= (4-1)-2 = 1\,,\\ \overline{\delta}_4 \,\overline{\oslash}\, (\overline{\delta}_2 \,\overline{\oslash}\, \overline{\delta})_1 &= 4-(1-2) = 5\,.\end{aligned}$$

Operation $\oslash$ not commutative. This follows from the following counter example.

$$\begin{aligned}\overline{\delta}_4 \,\overline{\oslash}\, \overline{\delta}_2 &= 4-2 = 2\,,\\ \overline{\delta}_2 \,\overline{\oslash}\, \overline{\delta}_4 &= 2-4 = -2\,.\end{aligned}$$

Composition of $\overline{\otimes}$ and $\overline{\oslash}$. We first expand the operators, and then substitute $\eta = \mu + \kappa$.

$$\begin{aligned}(F \,\overline{\oslash}\, G) \,\overline{\oslash}\, H(\nu) &= \inf_{\mu \geq 0}\{F \,\overline{\oslash}\, G(\nu+\mu) - H(\mu)\}\\ &= \inf_{\mu \geq 0}\{\inf_{\kappa \geq 0}\{F(\nu+\mu+\kappa) - G(\kappa)\} - H(\mu)\}\\ &= \inf_{\mu \geq 0}\inf_{\eta \geq \mu}\{F(\nu+\eta) - G(\eta-\mu) - H(\mu)\}\\ &= \inf_{\mu \geq 0}\{F(\nu+\eta) - \sup_{\mu \leq \eta}\{G(\eta-\mu) + H(\mu)\}\}\\ &= F \,\overline{\oslash}\, (G \,\overline{\otimes}\, H)(\nu)\,.\end{aligned}$$

Duality of $\overline{\oslash}$. We first show that if $F \geq G \,\overline{\otimes}\, H$ then $F \,\overline{\oslash}\, H \geq G$. We rewrite $F \geq G \,\overline{\otimes}\, H$ as

$$\forall \nu \in \mathbb{R}, \forall \kappa \in \mathbb{R} \,:\, F(\nu) \geq G(\kappa) + H(\nu-\kappa)\,.$$

Substituting $\mu = \nu - \kappa$, the above is equivalent to

$$\forall \kappa \in \mathbb{R}, \forall \mu \in \mathbb{R} \,:\, G(\kappa) \leq F(\mu+\kappa) - H(\mu)\,.$$

This is equivalent to

$$\forall \mu \in \mathbb{R} \,:\, G(\mu) \leq \inf_{\kappa \in \mathbb{R}}\{F(u+s) - H(u)\}\,,$$

which implies

$$\forall \mu \in \mathbb{R} \,:\, G(\mu) \leq \inf_{\kappa \geq 0} \{F(u+s) - H(u)\} = F \overline{\oslash} H(\mu)\,.$$

Now we show that $F \overline{\oslash} H \geq G$ implies $F \geq G \overline{\otimes} H$. We write $F \overline{\oslash} H \geq G$ as

$$\forall \nu \in \mathbb{R}\,, \forall \kappa \geq 0 \,:\, F(\nu + \kappa) \geq G(\nu) + H(\kappa)\,.$$

With a change of variables $\mu = \nu + \kappa$, we have

$$\forall \nu \in \mathbb{R}\,, \forall \mu \leq \nu \,:\, F(\mu) \geq G(\nu) + H(\mu - \nu)\,,$$

which is equivalent to

$$\forall \mu \in \mathbb{R}\,, \forall \nu \leq \mu \,:\, F(\mu) \geq G(\nu) + H(\mu - \nu)\,,$$

and which we can rewrite as

$$\forall \mu \in \mathbb{R} \,:\, F(\nu) \geq \sup_{\nu \leq \mu} \{G(\nu) + H(\mu - \nu)\}\,.$$

This implies

$$\forall \mu \in \mathbb{R} \,:\, F(\nu) \geq \sup_{0 \leq \nu \leq \mu} \{G(\nu) + H(\mu - \nu)\} = G \overline{\otimes} H(\mu)\,.$$

Acknowledgements

The author would like to thank Almut Burchard for numerous discussions and suggestions, and Markus Fidler, George Kesidis, Sami Baroudi, and Natchanon Luangsomboon for their valuable comments on the manuscript. The partial support of the Natural Sciences and Engineering Research Council of Canada (NSERC) is gratefully acknowledged.

References

[1] R. Agrawal, R. L. Cruz, C. Okino, and R. Rajan. Performance bounds for flow control protocols. *IEEE/ACM Transactions on Networking*, 7(3):310–323, June 1999.

[2] R. Agrawal, R. L. Cruz, C. M. Okino, and R. Rajan. A framework for adaptive service guarantees. In *Proc. of the Annual Allerton Conference on Communication and Computing*, volume 36, pages 693–702, 1998.

[3] F. Baccelli, G. Cohen, G. J. Olsder, and J.-P. Quadrat. *Synchronization and Linearity: An Algebra for Discrete Event Systems*. Wiley New York, 1992.

[4] J. C. R. Bennett, K. Benson, A. Charny, W. F. Courtney, and J.-Y. Le Boudec. Delay jitter bounds and packet scale rate guarantee for expedited forwarding. *IEEE/ACM Transactions on Networking*, 10(4):529–540, 2002.

[5] T. S. Blyth and M. F. Janowitz. *Residuation Theory*. International Series of Monographs in Pure and Applied Mathematics, Vol. 102, Pergamon Press, 1972.

[6] J. Y. Le Boudec. Application of network calculus to guaranteed service networks. *IEEE/ACM Transactions on Information Theory*, 44(3):1087–1097, May 1998.

[7] J. Y. Le Boudec and P. Thiran. *Network Calculus*. Springer Verlag, Lecture Notes in Computer Science, LNCS 2050, 2001.

[8] C. S. Chang. On deterministic traffic regulation and service guarantees: a systematic approach by filtering. *IEEE Transactions on Information Theory*, 44(3):1097–1110, 1998.

[9] C. S. Chang. *Performance Guarantees in Communication Networks*. Springer Verlag, 2000.

[10] C. S. Chang and Y. H. Lin. A general framework for deterministic service guarantees in telecommunication networks with variable length packets. *IEEE Transactions on Automatic Control*, 46(2):210–221, 2001.

[11] Y. Chen. Exact solution for one type of Lindley's equation for queueing theory and network calculus. Technical report, arXiv:1501.06845v2, 2015.

[12] F. Ciucu. Network calculus delay bounds in queueing networks with exact solutions. In *Proc. 20th International Teletraffic Congress (ITC)*, pages 495–506. June 2007.

[13] R. L. Cruz. A calculus for network delay, Part I : Network elements in isolation. *IEEE Transactions on Information Theory*, 37(1):114–121, January 1991.

[14] R. L. Cruz. A calculus for network delay, Part II : Network analysis. *IEEE Transactions on Information Theory*, 37(1):121–141, January 1991.

[15] R. L. Cruz. SCED+: efficient management of quality of service guarantees. In *Proc. IEEE Infocom*, pages 625–634, April 1998.

[16] B. Fan, H. Zhang, and W. Dou. Application of residuation theory in network calculus. In *Proc. IEEE International Conference on Networking, Architecture, and Storage (NAS)*, pages 180–183, July 2009.

[17] B. Fan, H. Zhang, and W. Dou. A max-plus network calculus. In *Proc. Eighth IEEE/ACIS International Conference on Computer and Information Science (ICIS)*, pages 149–154, June 2009.

[18] B. Fan, H. Zhang, and W. Dou. Residuated pairs in network analysis. In *Proc. IEEE 15th Asia-Pacific Conference on Communications (APCC)*, pages 892–895, October 2009.

[19] D. Ferrari and D. Verma. A scheme for real-time channel establishment in wide-area networks. *IEEE Journal on Selected Areas in Communications*, 8(3):368–379, April 1990.

[20] M. Fidler. Survey of deterministic and stochastic service curve models in the network calculus. *IEEE Communications Surveys & Tutorials*, 12(1):59–86, 2010.

[21] M. Gondran and M. Minoux. *Graphs, dioids and semirings: New models and algorithms*, volume 41. Springer Publishing Company, 2008.

[22] Y. Jiang. Network calculus and queueing theory: two sides of one coin: invited paper. In *Proc. Valuetools (4th International ICST Conference on Performance Evaluation Methodologies and Tools), Article No. 37*, 2009.

[23] J. Liebeherr, D. Wrege, and D. Ferrari. Exact admission control for networks with bounded delay services. *IEEE/ACM Transactions on Networking*, 4(6):885–901, December 1996.

[24] D. V. Lindley. The theory of queues with a single server. *Mathematical Proceedings of the Cambridge Philosophical Society*, 48(2):277–289, 1952.

[25] R. Lübben, M. Fidler, and J. Liebeherr. Stochastic bandwidth estimation in networks with random service. *IEEE/ACM Transactions on Networking*, 22(2):484–497, 2014.

[26] C. M. Okino. *A framework for performance guarantees in communication networks*. PhD thesis, Ph. D. Dissertation, University of California San Diego, 1998.

[27] K. Pandit, J. Schmitt, and R. Steinmetz. Network calculus meets queueing theory-a simulation based approach to bounded queues. In *Proc. 12th IEEE International Workshop on Quality of Service (IWQOS)*, pages 114–120, June 2004.

[28] A. Parekh and R. Gallager. A generalized processor sharing approach to flow control in integrated services networks: the single-node case. *IEEE/ACM Transactions on Networking*, 1(3):344–357, June 1993.

[29] A. Parekh and R. Gallager. A generalized processor sharing approach to flow control in integrated services networks: the multiple node case. *IEEE/ACM Transactions on Networking*, 2(2):137–150, April 1994.

[30] H. Sariowan, R. L. Cruz, and G. C. Polyzos. SCED: A generalized scheduling policy for guaranteeing quality-of-service. *IEEE/ACM Transactions on Networking*, 7(5):669–684, 1999.

[31] J. Xie and Y. Jiang. Stochastic network calculus models under max-plus algebra. In *Proc. IEEE Globecom*, pages 1–6, November/December 2009.

[32] J. Xie and Y. Jiang. Stochastic service guarantee analysis based on time-domain models. In *Proc. IEEE International Symposium on Modeling, Analysis & Simulation of Computer and Telecommunication Systems (MASCOTS)*, pages 1–12, 2009.

[33] J. Xie and Y. Jiang. A temporal network calculus approach to service guarantee analysis of stochastic networks. In *Proc. Valuetools (5th International ICST Conference on Performance Evaluation Methodologies and Tools)*, pages 408–417, May 2011.

[34] L. Zhang. VirtualClock: A New Traffic Control Algorithm for Packet Switching Networks. In *Proc. ACM Sigcomm*, pages 19–29, September 1990.