Rheinland-Pfälzische Technische Universität Kaiserslautern Landau

Master Thesis

Getting Out of Your Comfort Zone: Extending FIFO Network Calculus Results to Derive Bounds for General Piecewise Linear Curves

by

Lukas Wildberger

July 14, 2024

University of Kaiserslautern-Landau
Department of Computer Science
Distributed Systems Lab

Supervisor: Prof. Dr.-Ing. Jens B. Schmitt
Examiner: M. Sc. Anja Hamscher

Abstract

This thesis investigates the application of piecewise linear concave and convex functions as arrival and service curves within the deterministic network calculus framework. We introduce performance bound calculations for concave piecewise linear arrival curves with rate-latency service curves. Additionally, we utilize both concave and convex piecewise linear arrival and service curves to derive performance bounds. This approach allows for a more flexible and precise modeling of network traffic and service capabilities, which is essential for accurately predicting network performance under various conditions. We provide a comprehensive analysis of performance bounds associated with these curves and develop efficient methods for calculating them.

These theoretical insights are applied to the analysis of FIFO multiplexed networks, particularly focusing on the selection of the parameter θ. Our results demonstrate how θ can be chosen to minimize the latency of the leftover service curve and to achieve the smallest possible backlog or delay bound for a flow of interest (foi). Furthermore, we present a toolbox implementation that facilitates the visualization of theoretical results and enhances understanding through numerical examples. In addition to our primary focus, we extend the work of Hamscher et al. "Extending Network Calculus To Deal With Partially Negative And Decreasing Service Curves" regarding finite shared buffers by addressing the scenario including service curves with nonzero latencies.

Acknowledgement

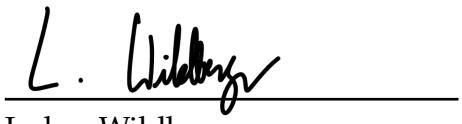
First and foremost, I extend my gratitude to Prof. Dr.-Ing. Jens B. Schmitt for the opportunity to write this master thesis under his supervision. My sincere thanks also go to M. Sc. Anja Hamscher, whose dedication and assistance were instrumental in addressing my questions and solving problems while writing this master thesis. Lastly, I am deeply grateful to Vlad-Cristian Constantin and Andreas Tomasini for their proofreading, which significantly enhanced the readability of this thesis.

Lukas Wildberger

Eidesstattliche Erklärung

Hiermit versichere ich, die vorliegende Masterarbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet zu haben. Alle wörtlich oder sinngemäß übernommenen Zitate sind als solche kenntlich gemacht worden. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Kaiserslautern, den July 14, 2024



Lukas Wildberger

Contents

List of Figures

1 Introduction

In modern network technology, efficient and reliable data transmission presents a central challenge. Networks must support a variety of services, from real-time video streaming to critical communication in autonomous systems, each with distinct requirements for bandwidth, latency, and reliability. To meet these diverse demands, it is crucial to develop robust mechanisms for analyzing and predicting network behavior. One promising approach in this context is Network Calculus, initially introduced by Cruz in 1991 [Cru91a, Cru91b].

Network Calculus encompasses two main branches: Deterministic Network Calculus (DNC) and Stochastic Network Calculus (SNC). DNC is a mathematical framework based on queuing theory, providing worst-case guarantees for performance metrics such as delay and backlog. Unlike stochastic methods, which rely on probabilistic models, DNC offers deterministic guarantees, making it particularly suited for safety-critical applications where reliability and predictability are paramount.

To apply the DNC framework effectively to real-world scenarios, it is essential to develop accurate yet tractable models. This requires balancing the complexity of the model with its simplicity to ensure manageability. Highly accurate models may necessitate complex mathematical treatments, which can complicate or even prevent practical application. One way to achieve this balance is by using piecewise linear concave and convex functions, as discussed by Bouillard et al. [BBLC18].

The use of piecewise linear concave and convex functions in first-in first-out (FIFO) multiplexed networks has been explored in prior research [BS12, BS14], where linear programming problems were employed to derive performance bounds. This thesis aims to extend these results by developing methods to utilize piecewise linear concave and convex functions more directly, without relying on linear programming problems. We will demonstrate how to calculate performance bounds and apply these techniques to understand the behavior of FIFO multiplexed networks.

This thesis is organized into seven chapters. Chapter 2 lays the groundwork by introducing the fundamental mathematical tools and providing a comprehensive overview of deterministic network calculus. Building on this foundation, Chapter 3 presents new findings for piecewise linear arrival curves with rate-latency service. The analysis is further expanded in Chapter 4 to encompass piecewise linear service curves. In Chapter 5, we delve into the calculation of FIFO leftover service curves, specifically using piecewise linear arrival and service curves.

Chapter 6 showcases the functionality of the visualization tool created during this research. Finally, finite shared buffer systems and the computation of performance bounds in such systems are considered in Chapter 7.

2 Deterministic Network Calculus Background

In the 1990s, Network Calculus was developed as a deterministic theory for evaluating the quality of service in packet-switched data networks. This theory models traffic arrivals using upper envelope functions [FR14]. Service curves are utilized to describe the minimum service guarantees for systems like routers, schedulers, or links. Network Calculus provides convolution forms that allow for the derivation of worst-case performance bounds, including backlog and delay [FR14].

In this chapter, we will first introduce the fundamental mathematical definitions (Section 2.1) that are essential for the analysis in the field of deterministic network calculus. After this we will begin presenting the key concepts of arrival and service curves (Section 2.2 and Section 2.3), which are crucial for modeling and analyzing network behavior. These curves enable a formal description and examination of data flows and service capacities within a network.

Building upon these definitions, we will explore performance bounds (Section 2.4) such as delay and backlog bound. These bounds allow us to determine upper limits for delays and buffer requirements, which are critical for ensuring the quality of service (QoS) in networks.

Towards the end of the chapter, we will briefly discuss scheduling algorithms (Section 2.5) and piecewise linear functions (Section 2.6). Scheduling algorithms are important for the efficient management of network resources, while piecewise linear functions are frequently used to model and analyze complex network characteristics.

This chapter provides the necessary foundation for a comprehensive understanding of the subsequent chapters.

2.1 Basics

Before we take a closer look at concepts of Deterministic Network Calculus, it is essential to establish a clear understanding of the mathematical foundations that will underpin our discussions throughout this thesis. Specifically, we will concentrate on the principles and applications of min-plus algebra, a key mathematical framework that plays a crucial role in our analyses. The analysis of min-plus algebra has been studied in depth in [BCOQ92]. This section aims to provide a comprehensive overview of the fundamental concepts and definitions necessary for an understanding of DNC.

Definition 2.1. (Set of Increasing Functions [LBT01]). We define $\mathcal{F}$ to be the set of real-valued, non-negative, increasing functions with $f(t) = 0$ for $t < 0$:

$$\mathcal{F} := \{f : \mathbb{R} \to \mathbb{R}^+ \cup \{+\infty\} \mid \forall s \leq t : 0 \leq f(s) \leq f(t),\ \forall t < 0 : f(t) = 0\},$$

where $\mathbb{R}^+ = [0, \infty)$. The set of functions $\mathcal{F}$ such that $f(0) = 0$ is defined as $\mathcal{F}_0$

Definition 2.2. (Min-Plus Convolution [LBT01]). Let f and g be two functions of $\mathcal{F}$. The *min-plus convolution* of f and g is the function

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

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

Definition 2.3. (Min-Plus Deconvolution [LBT01]). Let f and g be two functions of $\mathcal{F}$. The *min-plus deconvolution* of f by g is the function

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

Definition 2.4. (Vertical Deviation [LBT01]). Let $f, g \in \mathcal{F}$. The vertical deviation between f and g is defined as

$$v(f,g) := \sup_{t \geq 0} \{f(t) - g(t)\} = f \oslash g(0).$$

Definition 2.5. (Horizontal Deviation [LBT01]). Let $f, g \in \mathcal{F}$. The horizontal deviation between f and g is defined as

$$\begin{aligned} h(f,g) &:= \sup_{t \geq 0} \{\inf\{d \geq 0 \mid f(t) \leq g(t+d)\}\} \\ &= \sup_{t \geq 0} \{\inf\{d \geq 0 \mid f(t-d) \leq g(t)\}\} \\ &= \inf\left\{d \geq 0 \mid \sup_{t \geq 0} \{f(t-d) - g(t)\} \leq 0\right\} \\ &= \inf\{d \geq 0 \mid f \oslash g(-d) \leq 0\}. \end{aligned}$$

Definition 2.4 and Definition 2.5 can be effectively illustrated using Figure 2.1. For two given functions f and g in the set $\mathcal{F}$, the vertical and horizontal deviation are marked by purple dotted lines.

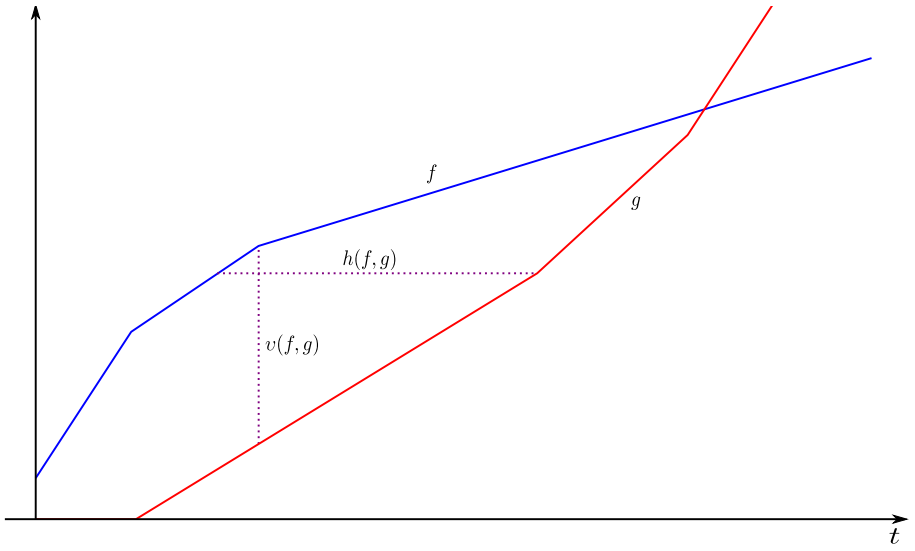


Figure 2.1: Vertical and horizontal deviation of functions f and g.

Definition 2.6. (Impulse Function [SN21]). The *impulse function* is defined by

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

Remark 2.7. (Shift Property of the Impulse Function [SN21]). The convolution of the impulse function δ_T with a function is similar to shifting to the right by T. Mathematically speaking, for any $f \in \mathcal{F}$, we have for $t \geq T$

$$\begin{aligned} f \otimes \delta_T(t) &= \inf_{0 \leq s \leq t} \{f(t-s) + \delta_T(s)\} \\ &\overset{(\delta_T(s)=+\infty, \text{ if } s>T)}{=} \inf_{0 \leq s \leq T} \{f(t-s) + \delta_T(s)\} \\ &\overset{(\delta_T(s)=0)}{=} f(t-T), \end{aligned}$$

where we used that $f(t-s)$ is decreasing in s.

A shift of a function $f(t)$ by T is illustrated in Figure 2.2.

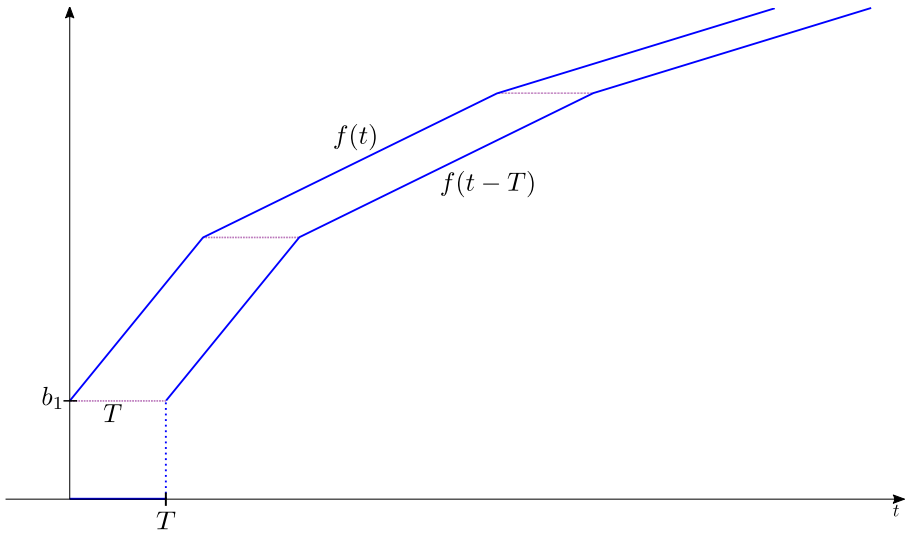


Figure 2.2: $f(t)$ shifted to the right by T.

Remark 2.8. (Visual Interpretation of the Deconvolution [SN21]). Let $f, g \in \mathcal{F}$. Then it holds that

$$\begin{aligned} f \oslash g(t) &= \sup_{u \geq 0}\{f(t+u) - g(u)\} \\ &= \sup_{u \geq 0}\{f \otimes \delta_{-t}(u) - g(u)\} \\ &= v(f \otimes \delta_{-t}, g). \end{aligned}$$

Definition 2.9. (Pseudo-inverse [BBLC18]). Let $f \in \mathcal{F}$ be a non-negative and non-decreasing function. The pseudo-inverse f^{-1} is for all $x \in \mathbb{R}^+$ given by

$$f^{-1}(x) = \inf\{t | f(t) \geq x\}.$$

2.2 Arrival

To derive performance bounds for a particular network or server, it is crucial to have precise and analyzable definitions of traffic arrivals. The following definitions are provided for this purpose.

Definition 2.10. (Arrival Process [LBT01]). Consider a system $\mathcal{S}$. The *arrival process* $A(t)$ cumulatively counts the number of work units that arrive at $\mathcal{S}$ in the interval $[0, t)$. Thus, it holds that $A \in \mathcal{F}_0$, or in other words:

$$\begin{aligned} &A(0) = 0, \\ &A(s) \leq A(t) \quad \forall s \leq t. \end{aligned}$$

Definition 2.11. (Arrival Curve [LBT01]). Given an increasing function $\alpha \in \mathcal{F}_0$. We say that α is an *arrival curve* for an arrival process A if for all $s \leq t$

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

We also say that $A(t)$ is α-smooth or $A(t)$ is constrained by α.

Example 2.12. (Token Bucket Arrival Curve [LBT01]). The token bucket (or leaky bucket) arrival curve is defined by

$$\alpha(t) = \gamma_{r,b}(t) := \begin{cases} 0, & \text{if } t = 0, \\ b + r \cdot t, & \text{otherwise.} \end{cases}$$

An illustration of a token bucket arrival curve is given in Figure 2.3.

2.3 Departure / Service

The final component necessary to calculate the performance bounds is the definition of departures or service curves. For this purpose, the following definitions are required.

Definition 2.13. (Departure Process [BBLC18]). Consider a system $\mathcal{S}$. $D(t)$ is the cumulative *departure process* (or output function) of $\mathcal{S}$ at time t.

Let us examine a system $\mathcal{S}$ in which both the input $A(t)$ and the departure $D(t)$ are observed simultaneously. It is required that for all $t \in \mathbb{R} \cup \{\infty\}$, the following condition holds:

$$A(t) \geq D(t).$$

This property, known as *causality*, essentially means that the system does not generate any bits on its own. So only the bits that have previously entered the system can exit it. [SN21]

Definition 2.14. (Strict Service Curve [LBT01]). A system is said to offer a *strict service curve* β to a flow if, during any (continuously) backlogged period (s,t), the output of the system is at least equal to $\beta(t-s)$, that is,

$$D(t) - D(s) \geq \beta(t-s).$$

Definition 2.15. (Service Curve [LBT01]). Consider a system $\mathcal{S}$ and a flow through $\mathcal{S}$ with arrival process A and according departure process D. $\mathcal{S}$ offers a (*minimum*) *service curve* β to A if $\beta \in \mathcal{F}_0$ and for all $t \in \mathbb{R}$

$$D(t) \geq A \otimes \beta(t) = \inf_{0 \leq s \leq t} \{A(t-s) + \beta(s)\}.$$

Example 2.16. (Rate-Latency Service Curve [LBT01]). The rate-latency service curve is defined by

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

where $[x]^+ := \max\{x, 0\}$ denotes the positive part.

The service curve for the special case where $T = 0$, thus

$$\beta_{C,0}(t) = C \cdot t,$$

is called constant rate service curve.

An illustration of a rate-latency service curve is given in Figure 2.3.

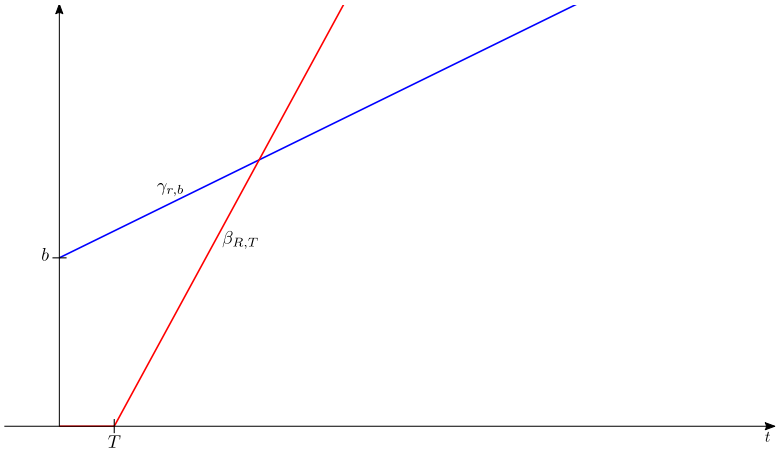


Figure 2.3: Token bucket arrival curve and rate-latency service curve.

Proposition 2.17. (Pseudo-Inverse of Token Bucket Arrival Curve and Rate-Latency Service Curve [BBLC18]) *Let $\gamma_{r,b}(t)$ be a token bucket arrival curve and $\beta_{R,T}(t)$ be a rate-latency service curve. Then, the following holds for the pseudo-inverse (see Defintion 2.9) of $\gamma_{r,b}(t)$ and $\beta_{R,T}(t)$:*

$$\gamma_{r,b}^{-1}(t) = \beta_{\frac{1}{r},b}(t),$$
$$\beta_{R,T}^{-1}(t) = \gamma_{\frac{1}{R},T}(t),$$

with $r > 0$ and $R > 0$.

2.4 Performance Bounds

The subsequent section presents fundamental network calculus results. These bounds pertain to the single flow, single server system (refer to Figure 3.4). Nonetheless, it is important to note that these findings suffice for all subsequent and more intricate topologies. This is because we can break them down into the single flow, single server system.

Theorem 2.18. (Backlog Bound [LBT01, p.22]). *Assume an arrival process A, that is constrained by arrival curve α, traverses a system $\mathcal{S}$ that offers a service curve β. For all t the backlog $q(t)$ is bounded by*

$$q(t) \leq \sup_{s \geq 0} \{\alpha(s) - \beta(s)\} = \alpha \oslash \beta(0) = v(\alpha, \beta).$$

Theorem 2.19. (Delay Bound [LBT01, p.23]). *Assume an arrival process A, that is constrained by arrival curve α, traverses a system $\mathcal{S}$ that offers a service curve β. For all t the delay $d(t)$ is bounded by*

$$d(t) \leq \inf\{\tau \geq 0 \mid \alpha \oslash \beta(-\tau) \leq 0\} = h(\alpha, \beta).$$

Theorem 2.20. (Output Bound [LBT01, p.23]). *Assume an arrival process A, that is constrained by arrival curve α, traverses a system $\mathcal{S}$ that offers a service curve β. The departures of the flow, D are constrained for all t by the arrival curve*

$$\alpha'(t) := \begin{cases} \alpha \oslash \beta(t), & \text{if } t > 0 \\ 0, & \text{otherwise.} \end{cases}$$

Theorem 2.21. (Backlogged Period [BBLC18, p.122]). *Assume an arrival process A, that is constrained by arrival curve α, traverses a system $\mathcal{S}$ that offers a strict service curve β. The duration of the backlogged period of the system is upper bounded by*

$$\inf\{t > 0 \mid \alpha(t) < \beta(t)\}.$$

2.5 Scheduling

Having looked at the performance bounds for single flow, single server systems, we are now focusing on the challenge of managing multiple flows. Specifically, we are exploring how to distribute resources when several participants compete for them. To address this, let us consider scheduling for such systems. Scheduling essentially involves fairly distributing limited resources amid competition. We will focus on a scenario with two flows and one server system (as shown in Figure 5.1). This scheduling follows predefined rules that determine resource allocation clearly. In this thesis, we will primarily consider FIFO (First In First Out) scheduling servers and discuss the service we can expect for a designated "flow of interest" (foi).

Theorem 2.22. (Leftover Service Curve for Earliest Deadline First (EDF) [LBL07]). *Let $t \geq 0$. Consider a system $\mathcal{S}$ that multiplexes two flows f_1 and f_2 using EDF scheduling. The arrivals of f_2, A_2, are constrained by α_2. Further, assume that system $\mathcal{S}$ guarantees a strict service curve β to the aggregate of the flows. Let the relative deadlines of flow f_1 and f_2 be d_1 and d_2, respectively. Then, the leftover service*

$$\beta^1_{EDF}(t) := \beta^1(t) := \left[\beta(t) - \alpha_2(t - [d_2 - d_1]^+)\right]^+$$

is a service curve for flow f_1 if $\beta^1 \in \mathcal{F}_0$.

Theorem 2.23. (Leftover Service Curve for FIFO [Cru98]). *Let $t \geq 0$. Consider a system $\mathcal{S}$ that multiplexes two flows f_1 and f_2 using FIFO scheduling. The arrivals of f_2, A_2, are constrained by α_2. Further, assume that $\mathcal{S}$ guarantees a strict service curve β to the aggregate of the flows. Then, for any $\theta \geq 0$, the leftover service*

$$\begin{aligned} \beta^1_{\theta,FIFO}(t) = \beta^1_\theta(t) &= \begin{cases} [\beta(t) - \alpha_2(t-\theta)]^+, & \text{if } t > \theta, \\ 0, & \text{otherwise.} \end{cases} \\ &= [\beta(t) - \delta_\theta \otimes \alpha_2(t)]^+ \wedge \delta_\theta(t) \end{aligned} \tag{2.1}$$

is a service curve for flow f_1 if $\beta^1_\theta \in \mathcal{F}_0$.

Remark 2.24. (Traffic Light Interpretation [SN21]) Considering the proof of Theorem 2.23 given in [SN21], we get insight into the parameter θ within the context of earliest deadline first (see Theorem 2.22). Instead of strictly applying FIFO multiplexing, we use a delay of θ to the foi's traffic and adjust the delay index to eventually achieve the same deadline. This can be conceptualized as a traffic light at a crossing. Instead of implementing FIFO for each car, we allow multiple cars to cross in a batch before giving priority to cars from a different direction (representing a different flow). In our worst-case analysis, we make sure that the foi encounters the initial "red phase" (delay by θ).

2.6 Piecewise Linear Functions

The final section of this chapter introduces the category of piecewise linear functions. Specifically, we will present the normal forms for piecewise linear concave and convex functions, as these are the central focus of the research presented in this thesis.

Definition 2.25. (Piecewise Linear Concave Function). Let $r_i, b_i \in \mathbb{R}^+$ and $\gamma_i(t) \in \mathcal{F}$ be linear functions of the form $\gamma_i = b_i + r_i \cdot t$ with $i \in \{1, ...n\}$. Then the function f with

$$f = \min\{\gamma_1(t), ..., \gamma_n(t)\} = \min\{\gamma_i(t)\}$$

is called piecewise linear concave.

Definition 2.26. (Piecewise Linear Convex Function). Let $R_i, T_i \in \mathbb{R}^+$ and $\beta_i(t) \in \mathcal{F}$ be linear functions of the form $\beta_i = R_i \cdot [t - T_i]^+$ with $i \in \{1, ...n\}$. Then the function f with

$$f = \max\{\beta_1(t), ..., \beta_n(t)\} = \max\{\beta_i(t)\}$$

is called piecewise linear convex.

Definition 2.27. (Concave Piecewise Linear Normal Form [BBLC18]).
Let $r_1, ..., r_n, b_1, ..., b_n$ be non-negative numbers and set $\gamma_i = \gamma_{r_i,b_i}$. The piecewise linear concave function

$$f = \min\{\gamma_i\}$$

is said to be in normal form, if γ_i are sorted by a decreasing rate and no γ_i can be removed without modifying the minimum:

$$\begin{aligned} &i < j \Rightarrow r_i > r_j, \\ &\forall i, \exists t > 0, \forall j \neq i, \gamma_i(t) < \gamma_j(t). \end{aligned}$$

If $f = \min\{\gamma_i\}, i \in [1, n]$ is in normal form, then there is a sequence of t_i of respective intersections of the linear functions γ_i and γ_{i+1}. These intersection are defined as follows:

$$t_i := \begin{cases} t_1 = 0, & \text{if } i = 1, \\ t_i = \frac{b_{i+1} - b_i}{r_i - r_{i+1}}, & \text{if } 2 \leq i \leq n, \\ t_{n+1} = \infty, & \text{otherwise.} \end{cases}$$

Definition 2.28. (Convex Piecewise Linear Normal Form [BBLC18]).
Let $R_1, ..., R_n, T_1, ..., T_n$ be non-negative numbers and set $\beta_i = \beta_{R_i,T_i}$. The piecewise linear convex function

$$f = \max\{\beta_i\}$$

is said to be in normal form, if β_i are sorted by a increasing rate and no β_i can be removed without modifying the maximum:

$$i < j \Rightarrow R_i < R_j,$$
$$\forall i, \exists t > 0, \forall j \neq i, \beta_i(t) > \beta_j(t).$$

If $f = \max\{\beta_i\}, i \in [1, n]$ is in normal form (see Definition 2.28), then there is a sequence of u_i of respective intersections of the linear functions β_i and β_{i+1}. These intersection are defined as follows:

$$u_i := \begin{cases} u_1 = T_1, & \text{if } i = 1, \\ u_i = \frac{R_i T_i - R_{i+1} T_{i+1}}{R_i - R_{i+1}}, & \text{if } 2 \leq i \leq n, \\ u_{n+1} = \infty, & \text{otherwise.} \end{cases}$$

Remark 2.29. It can be easily seen that the piecewise linear function f from Definition 2.25 and Definition 2.26 is also element of $\mathcal{F}$. Furthermore, we can clearly see in Figure 2.4 why f (or g) is called a piecewise linear function, as we have linear functions for the respective intervals, e.g., for $t \in [t_1, t_2]$ or $t \in [u_2, u_3]$.

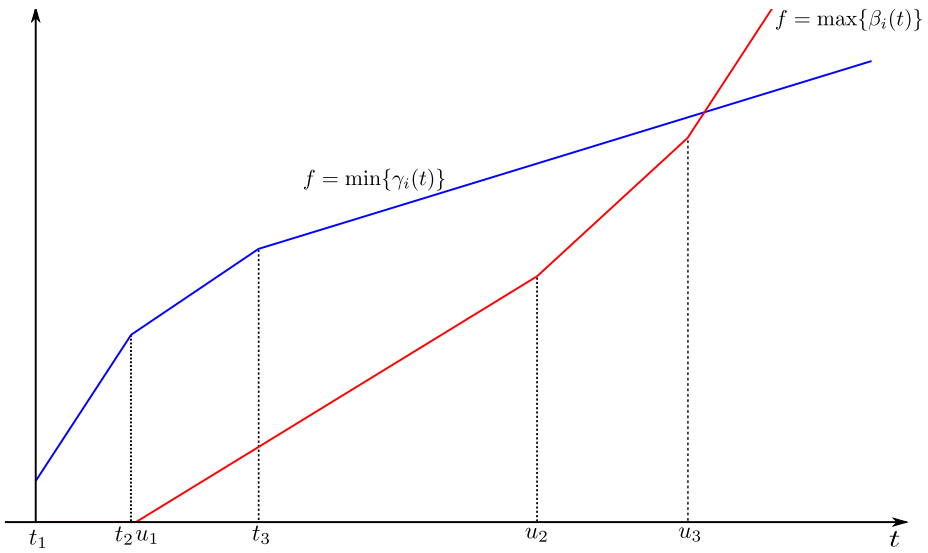


Figure 2.4: Piecewise linear functions in concave and convex normal form, respectively.

The convexity and concavity of piecewise linear functions afford us the advantage of stability in addition and minimum (or maximum) operations. This means that the minimum (or maximum) or sum of two concave or convex piecewise linear functions results in another concave or convex piecewise linear function, respectively. Additionally, this ensures that we cannot have constant sections within the function, thus avoiding a staircase function.

Proposition 2.30. (Concave Piecewise Linear Function Properties [BBLC18]). *Let $f = \min\{\gamma_i\} = \gamma_{r_i,b_i}, i \in [1,n]$ be a piecewise linear concave function in normal form. It holds that the sequence $(t_i)_{i=1}^n$ (as defined in Definition 2.27) is increasing $(t_i < t_{i+1})$.*

Proposition 2.31. (Convex Piecewise Linear Function Properties [BBLC18]). *Let $f = \max\{\beta_i\} = \beta_{R_i,T_i}, i \in [1,n]$ be a piecewise linear convex function in normal form. It holds that the sequence $(u_i)_{i=1}^n$ (as defined in Definition 2.28) is increasing $(u_i < u_{i+1})$.*

Lemma 2.32. (Pseudo-Inverse of Piecewise Linear Arrival and Service Curve [BV16]) *Let $f = \min\{\gamma_i\} = \gamma_{r_i,b_i}, i \in [1,m]$ be a piecewise linear concave function in normal form and let $g = \max\{\beta_j\} = \beta_{R_j,T_j}, j \in [1,n]$ be a piecewise linear convex function in normal form. The pseudo-inverse of f and g, respectively, is given by:*

$$f^{-1} = \max\{\beta_i\} = \max\{\beta_{\frac{1}{r_i},b_i}\},$$
$$g^{-1} = \min\{\gamma_j\} = \min\{\gamma_{\frac{1}{R_j},T_j}\},$$

with β_i and γ_j being the pseudo-inverse of γ_i and β_j, according to Proposition 2.17.

Remark 2.33. Towards the end of this chapter, we introduce a fundamental assumption that extends throughout the entire thesis, called "stability condition".

Consider a given arrival curve $\alpha(t)$ (which can be either a token bucket or a concave piecewise linear arrival curve in normal form) and a service curve $\beta(t)$ (which can be either a rate-latency or a convex piecewise linear service curve in normal form). Let $\{r_i | i \in \mathbb{N}\}$, be the set of rate(s) of $\alpha(t)$, and $\{R_j | j \in \mathbb{N}\}$, be the set of rate(s) of $\beta(t)$. The stability condition is defined as follows:

$$\exists r \in \{r_i | i \in \mathbb{N}\} \wedge \exists R \in \{R_j | j \in \mathbb{N}\} : r < R.$$

We always assume the stability condition to be fulfilled.

3 Single Flow Crosses One Server: Piecewise Linear Arrival Curve with Rate-Latency Service Curve

Following the introduction to the background of Deterministic Network Calculus, we now transition to the core focus of this thesis. The primary objective is to extend the current Network Calculus results by incorporating FIFO-scheduled servers in conjunction with piecewise linear curves. Before delving into the detailed analysis of piecewise linear arrival and service curves (as discussed in Chapter 4), this chapter will address the simpler scenario of a piecewise linear arrival curve paired with a rate-latency service curve.

The primary aim of this chapter is to establish the methodology for calculating performance bounds for systems characterized by piecewise linear arrival curves and rate-latency service curves. To achieve this, we will first introduce the foundational concepts of min-plus algebra in Section 3.1, demonstrating how to compute the convolution and deconvolution specific to our context. Building on these concepts, Section 3.2 will utilize these mathematical tools to investigate and derive the performance bounds.

3.1 Min-Plus Convolution and Deconvolution

In this section, we focus on the essential operations of min-plus algebra, specifically the convolution (Subsection 3.1.1) and deconvolution (Subsection 3.1.2), tailored to our setting of piecewise linear arrival curves and rate-latency service curves. These operations are fundamental for our subsequent performance bound calculations, providing the mathematical tools needed.

3.1.1 Convolution

Lemma 3.1. *Let $\alpha(t) = \min\{\gamma_i\}(t)$, $i \in [1, n]$ be a piecewise linear concave function in normal form, with $\gamma_i(t) = \gamma_{r_i,b_i}(t) = r_i \cdot t + b_i$ being token bucket arrival curves. Let $\beta(t) = \beta_{R,T}(t) = R \cdot [t - T]^+$ be a rate-latency service curve.*

Then it holds for $t \geq 0$

$$\alpha \otimes \beta(t) = \begin{cases} 0, & \text{if } t \leq T, \\ \min\{\beta(t), \alpha(t-T)\}, & \text{otherwise.} \end{cases}$$

Proof. Let t_i be the sequence of intersections as defined in Definition 2.27.

Let us first define t_a as follows:

- If $i \geq 2 \wedge (\exists t_i : r_{i-1} \geq R \wedge r_i < R)$:
  We define t_a, $a \in [2, n]$, as the intersection of γ_{a-1} and γ_a for which holds that $r_{a-1} \geq R$ and $r_a < R$.
- Otherwise (if $i < 2 \vee (\forall r_i : r_i < R)$:
  We set $t_a = t_1$.

Calculate the convolution $\alpha \otimes \beta(t)$ by calculating the result for each different case:

$$\alpha \otimes \beta(t) = \inf_{0 \leq s \leq t} \{\alpha(t-s) + \beta(s)\}$$

Case I: $t \leq T$

$$\alpha \otimes \beta(t) = \inf_{0 \leq s \leq t} \{\alpha(t-s) + 0\} \overset{(s=t)}{=} \alpha(0) = \underline{\underline{0}}$$

Case II: $t > T$

$$\begin{aligned} \alpha \otimes \beta(t) &= \inf_{0 \leq s \leq T} \{\alpha(t-s) + \beta(s)\} \wedge \inf_{T < s < t} \{\alpha(t-s) + \beta(s)\} \wedge \inf_{s=t} \{\alpha(t-s) + \beta(s)\} \\ &\overset{(s=T, s=t)}{=} \{\alpha(t-T) + 0\} \wedge \inf_{T < s < t} \{\alpha(t-s) + \beta(s)\} \wedge \{\alpha(0) + \beta(t)\} \\ &= \{\alpha(t-T)\} \wedge \{\beta(t)\} \wedge \inf_{T < s < t} \{\alpha(t-s) + \beta(s)\} \end{aligned}$$

Case IIa: $t_a \leq T$

$$\begin{aligned} \alpha \otimes \beta(t) &\overset{(s=T)}{=} \{\alpha(t-T)\} \wedge \{\beta(t)\} \wedge \{\alpha(t-T)\} \\ &= \underline{\underline{\min\{\beta(t), \alpha(t-T)\}}} \end{aligned}$$

Case IIb: $t_a > T$
Case IIba: $t_a < t$

$$\begin{aligned} \alpha \otimes \beta(t) &\overset{(s=T)}{=} \{\alpha(t-T)\} \wedge \{\beta(t)\} \wedge \{\alpha(t-T)\} \\ &= \underline{\underline{\min\{\beta(t), \alpha(t-T)\}}} \end{aligned}$$

Case IIbb: $t_a \geq t$

$$\begin{aligned} \alpha \otimes \beta(t) &\overset{(s=t)}{=} \{\alpha(t-T)\} \wedge \{\beta(t)\} \wedge \{\beta(t)\} \\ &= \underline{\underline{\min\{\beta(t), \alpha(t-T)\}}} \end{aligned}$$

□

Remark 3.2. In the proof of Lemma 3.1, we first defined t_a, which was used for case differentiation throughout the proof. We will see that we will use this t_a in almost every Lemma or Theorem in this Chapter. It will always be defined as follows:

- If $i \geq 2 \wedge (\exists t_i : r_{i-1} \geq R \wedge r_i < R)$:
  We define t_a, $a \in [2, n]$, as the intersection of γ_{a-1} and γ_a for which holds that $r_{a-1} \geq R$ and $r_a < R$.
- Otherwise (if $i < 2 \vee (\forall r_i : r_i < R)$:
  We set $t_a = t_1$.

In other words, we choose the intersection where the rate before the intersection is greater than or equal to the service rate R, and the rate after the intersection is actually lower than R.

Why is it important for us to identify this intersection? It helps us determine from which intersection the service curve can "catch up" with the arrival curve. This enables us to select the respective linear sections that are crucial for our calculations.

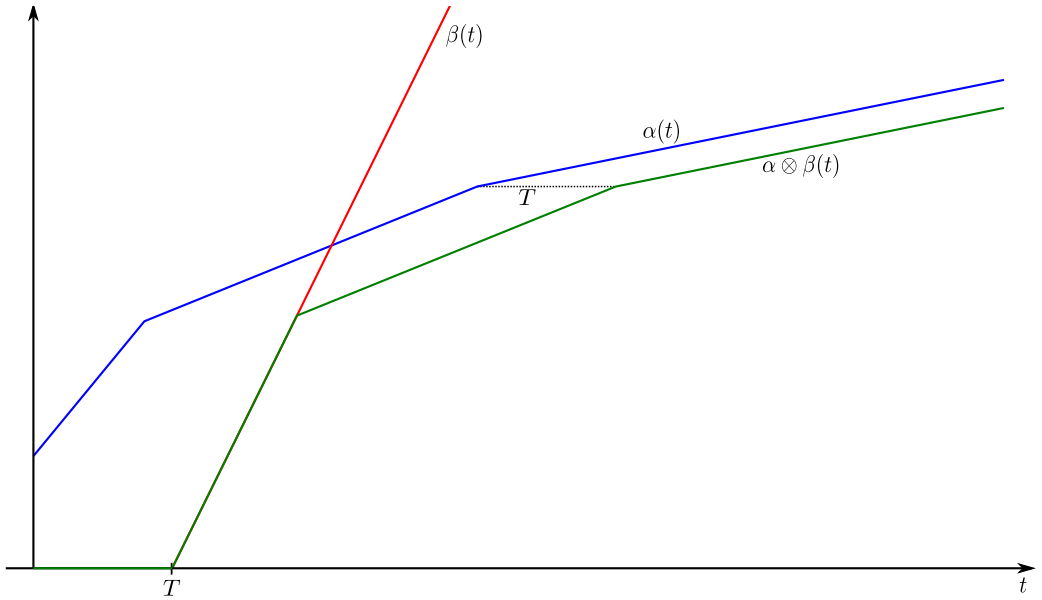


Figure 3.1: Min-plus convolution of $\alpha(t)$ and $\beta(t)$.

Consider the calculation of the convolution from Lemma 3.1 illustrated by the example in Figure 3.1. For a given piecewise linear arrival curve $\alpha(t)$ (blue curve) and rate-latency service curve $\beta(t)$ (red curve), the convolution $\alpha \otimes \beta(t)$ is given by the green curve (see Figure 3.1). In this example, we observe that $t_a = t_1$ because $\forall r_i : r_i < R$. Relating this to the cases in the proof of Lemma 3.1, we see that Cases I, II, and IIa are applied in this instance.

3.1.2 Deconvolution

Having examined the convolution, we now turn our attention to the deconvolution. We begin by considering a special case, detailed in Lemma 3.3, where the arrival curve consists of only two linear segments. Subsequently, in Lemma 3.4, we extend this analysis to address the deconvolution for arrival curves with any number of linear segments.

Lemma 3.3. *Let $\alpha(t) = \min\{\gamma_i\}(t)$, $i \in [1,2]$ be a piecewise linear concave function in normal form, with $\gamma_i(t) = \gamma_{r_i,b_i}(t) = r_i \cdot t + b_i$ being token bucket arrival curves. Let $\beta(t) = \beta_{R,T}(t) = R \cdot [t-T]^+$ be a rate-latency service curve. Let t_i be the sequence of intersections as defined in Definition 2.27.*

It holds for all $t \in \mathbb{R}$

$$\alpha \oslash \beta(t) = \begin{cases} R(t+T) + b_1, & \text{if } t \leq t_2 - T \wedge r_1 \leq R \wedge t \leq -T, \\ r_1(t+T) + b_1, & \text{if } t \leq t_2 - T \wedge r_1 \leq R \wedge t > -T, \\ R(t+T-t_2) + b_2 + r_2 t_2, & \text{if } t \leq t_2 - T \wedge r_1 > R, \\ r_2(t+T) + b_2, & \text{otherwise.} \end{cases}$$

Proof. Calculate the convolution $\alpha \oslash \beta(t)$ by calculating the result for each different case:

$$\alpha \oslash \beta(t) = \sup_{s \geq 0}\{\alpha(t+s) - \beta(s)\}$$

Case I: $t_2 \leq T$

$$\begin{aligned} \alpha \oslash \beta(t) &= \sup_{0 \leq s \leq t_2}\{\alpha(t+s) - \beta(s)\} \vee \sup_{s > t_2}\{\alpha(t+s) - \beta(s)\} \\ &\overset{(s=t_2)}{=} \alpha(t+t_2) \vee \sup_{s > t_2}\{\alpha(t+s) - \beta(s)\} \end{aligned}$$

Case Ia: $t \leq t_2 - T$
Case Iaa: $r_1 \leq R$
Case Iaaa: $t \leq -T$

$$\begin{aligned} \alpha \oslash \beta(t) &= \alpha(t+t_2) \vee \sup_{t_2 < s \leq -t}\{\alpha(t+s) - \beta(s)\} \vee \sup_{s > -t}\{\alpha(t+s) - \beta(s)\} \\ &\overset{(t+t_2 \leq 0)}{=} 0 \vee \sup_{t_2 < s \leq -t}\{r_1(t+s) + b_1 - \beta(s)\} \vee \sup_{s > -t}\{\alpha(t+s) - \beta(s)\} \\ &\overset{(s=-t)}{=} r_1(t-t) + b_1 - R(-t-T)\} \vee 0 \\ &= b_1 - R(-t-T) = \underline{\underline{R(t+T) + b_1}} \end{aligned}$$

Case Iaab: $t > -T$

$$\begin{aligned}\alpha \oslash \beta(t) &= \alpha(t+t_2) \vee \sup_{t_2 < s \le T} \{\alpha(t+s) - \beta(s)\} \vee \sup_{s>T} \{\alpha(t+s) - \beta(s)\} \\ &\overset{(s=T)}{=} \alpha(t+t_2) \vee \alpha(t+T) - 0 \vee \alpha(t+T) - 0 \\ &= \alpha(t+t_2) \vee \alpha(t+T) \\ &\overset{(t_2 \le T)}{=} \alpha(t+T) \overset{(t+T \le t_2)}{=} \gamma_1(t+T) = \underline{\underline{r_1(t+T) + b_1}}\end{aligned}$$

Case Iab: $r_1 > R$

$$\begin{aligned}\alpha \oslash \beta(t) &= \alpha(t+t_2) \vee \sup_{t_2 < s \le T} \{\alpha(t+s) - \beta(s)\} \vee \sup_{s>T} \{\alpha(t+s) - \beta(s)\} \\ &\overset{(s=T)}{=} \alpha(t+t_2) \vee \alpha(t+T) - 0 \vee \sup_{s>T} \{\alpha(t+s) - \beta(s)\} \\ &\overset{(t_2 \le T)}{=} \alpha(t+T) \vee \sup_{T < s \le t_2 - t} \{\alpha(t+s) - \beta(s)\} \vee \sup_{s > t_2 - t} \{\alpha(t+s) - \beta(s)\} \\ &= \alpha(t+T) \vee \sup_{T < s \le t_2 - t} \{\alpha(t+s) - \beta(s)\} \vee \sup_{s > t_2 - t} \{b_2 + r_2 t_2 + RT + (r_2 - R)s\} \\ &\overset{(s=t_2-t)}{=} \alpha(t+T) \vee b_2 + r_2 t_2 + RT + (r_2 - R)(t_2 - t) \\ &\quad \vee b_2 + r_2 t_2 + RT + (r_2 - R)(t_2 - t) \\ &= \alpha(t+T) \vee b_2 + r_2 t_2 + R(t + T - t_2) \\ &= \underline{\underline{R(t+T-t_2) + b_2 + r_2 t_2}}\end{aligned}$$

Case Ib: $t > t_2 - T$

$$\begin{aligned}\alpha \oslash \beta(t) &= \alpha(t+t_2) \vee \sup_{t_2 < s \le T} \{\alpha(t+s) - \beta(s)\} \vee \sup_{s>T} \{\alpha(t+s) - \beta(s)\} \\ &\overset{(s=T)}{=} \alpha(t+t_2) \vee \alpha(t+T) - 0 \vee \alpha(t+T) - 0 \\ &= \alpha(t+t_2) \vee \alpha(t+T) \\ &\overset{(t_2 \le T)}{=} \alpha(t+T) \overset{(t+T > t_2)}{=} \gamma_2(t+T) = \underline{\underline{r_2(t+T) + b_2}}\end{aligned}$$

Case II: $t_2 > T$
Case IIa: $t \le t_2 - T$
Case IIaa: $r_1 \le R$
Case IIaaa: $t \le -T$

$$\begin{aligned}\alpha \oslash \beta(t) &= \sup_{0 \le s \le -t} \{\alpha(t+s) - \beta(s)\} \vee \sup_{s>-t} \{\alpha(t+s) - \beta(s)\} \\ &= \sup_{0 \le s \le -t} \{r_1(t+s) + b_1 - \beta(s)\} \vee \sup_{s>-t} \{\alpha(t+s) - \beta(s)\} \\ &\overset{(s=-t)}{=} r_1(t-t) + b_1 - R(-t-T) \vee \sup_{s>-t} \{\alpha(t+s) - \beta(s)\} \\ &\overset{(s=-t)}{=} b_1 + R(t+T) \vee 0 = \underline{\underline{R(t+T) + b_1}}\end{aligned}$$

Case IIaab: $t > -T$

$$\begin{aligned}\alpha \oslash \beta(t) &= \sup_{0\le s\le T}\{\alpha(t+s)-\beta(s)\} \vee \sup_{s>T}\{\alpha(t+s)-\beta(s)\}\\ &\overset{(s=T,t+T\le t_2)}{=} \gamma_1(t+T) - 0 \vee \gamma_1(t+T) - 0\\ &= \underline{\underline{r_1(t+T)+b_1}}\end{aligned}$$

Case IIab: $r_1 > R$

$$\begin{aligned}\alpha \oslash \beta(t) &= \sup_{0\le s\le T}\{\alpha(t+s)-\beta(s)\} \vee \sup_{s>T}\{\alpha(t+s)-\beta(s)\}\\ &\overset{(s=T)}{=} \alpha(t+T) \vee \sup_{T<s\le t_2-t}\{\alpha(t+s)-\beta(s)\} \vee \sup_{s>t_2-t}\{\alpha(t+s)-\beta(s)\}\\ &= \alpha(t+T) \vee \sup_{T<s\le t_2-t}\{b_2+r_2t+RT+(r_2-R)s\}\\ &\quad \vee \sup_{s>t_2-t}\{b_2+r_2t+RT+(r_2-R)s\}\\ &\overset{(s=t_2-t)}{=} \alpha(t+T) \vee b_2+r_2t+RT+(r_2-R)(t_2-t)\\ &\quad \vee b_2+r_2t+RT+(r_2-R)(t_2-t)\\ &\overset{(t+T\le t_2)}{=} \gamma_1(t+T) \vee R(t+T-t_2)+b_2+r_2t_2\\ &= \underline{\underline{R(t+T-t_2)+b_2+r_2t_2}}\end{aligned}$$

Case IIb: $t > t_2 - T$

$$\begin{aligned}\alpha \oslash \beta(t) &= \sup_{0\le s\le T}\{\alpha(t+s)-\beta(s)\} \vee \sup_{s>T}\{\alpha(t+s)-\beta(s)\}\\ &= \sup_{0\le s\le T}\{\alpha(t+s)-\beta(s)\} \vee \sup_{T<s\le t_2}\{\alpha(t+s)-\beta(s)\} \vee \sup_{s>t_2}\{\alpha(t+s)-\beta(s)\}\\ &\overset{(s=T,s=t_2)}{=} \alpha(t+T) - 0 \vee \alpha(t+T) - 0 \vee \alpha(t+t1) - \beta(t_2)\\ &= \alpha(t+T) \vee \alpha(t+t1) - \beta(t_2)\\ &\overset{(t+T\ge t_2,t+t_2\ge t_2)}{=} \gamma_2(t+T) \vee \gamma_2(t+t_2) - R(t_2-T)\\ &\overset{(r_2<R,t_2>T)}{=} \gamma_2(t+T) = \underline{\underline{r_2(t+T)+b_2}}\end{aligned}$$

Since we get the same results for some cases, they can be merged and we get the following simplification:

$$\alpha \oslash \beta(t) = \begin{cases} R(t+T)+b_1, & \text{if } t \le t_2 - T \wedge r_1 \le R \wedge t \le -T,\\ r_1(t+T)+b_1, & \text{if } t \le t_2 - T \wedge r_1 \le R \wedge t > -T,\\ R(t+T-t_2)+b_2+r_2t_2, & \text{if } t \le t_2 - T \wedge r_1 > R,\\ r_2(t+T)+b_2, & \text{otherwise.}\end{cases}$$

□

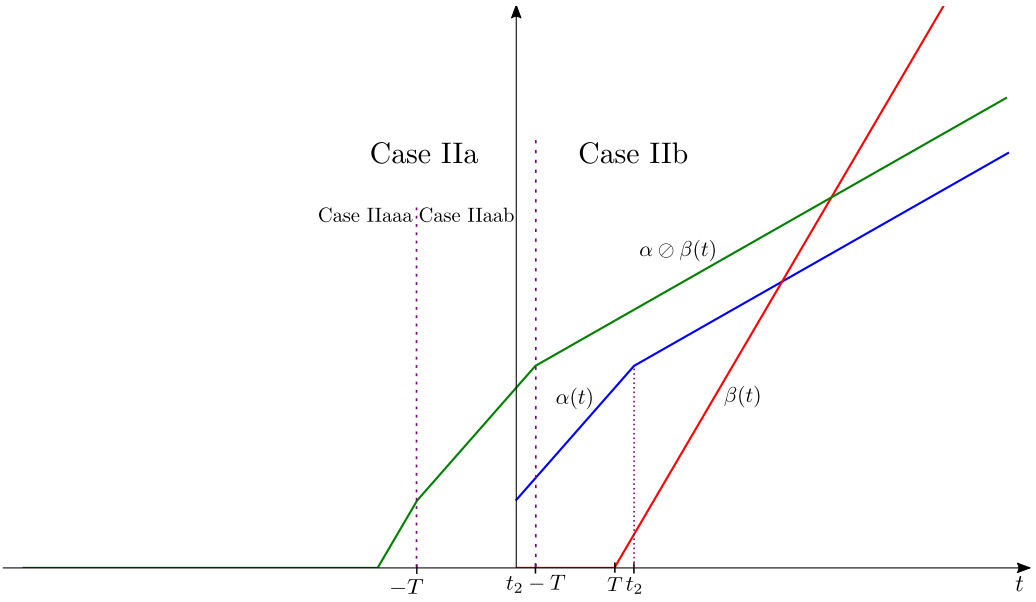


Figure 3.2: Case distinction Min-plus deconvolution of $\alpha(t)$ and $\beta(t)$.

Given the potential complexity of the case distinctions within the proof of Lemma 3.3, let us turn our attention to Figure 3.2. Since $t_2 > T$ holds, we examine the sub-case differentiations of Case II (the structure would be similar for Case I). For Case IIa, i.e., $t \leq t_2 - T$, we get Case IIaa, since $r_1 \leq R$ holds. The sub-Cases IIaaa and IIaab are differentiated at $-T$. Case IIb applies for $t > t_2 - T$.

Lemma 3.4. *Let $\alpha(t) = \min\{\gamma_i\}(t)$, $i \in [1, n]$ be a piecewise linear concave function in normal form, with $\gamma_i(t) = \gamma_{r_i,b_i}(t) = r_i \cdot t + b_i$ being token bucket arrival curves. Let $\beta(t) = \beta_{R,T}(t) = R \cdot [t - T]^+$ be a rate-latency service curve.*

Let t_i be the sequence of intersections as defined in Definition 2.27 and let t_a be defined as follows:

- *If $i \geq 2 \wedge (\exists t_i : r_{i-1} \geq R \wedge r_i < R)$:*
  We define t_a, $a \in [2, n]$, as the intersection of γ_{a-1} and γ_a for which holds that $r_{a-1} \geq R$ and $r_a < R$.
- *Otherwise (if $i < 2 \vee (\forall r_i : r_i < R)$):*
  We set $t_a = t_1$.

Then it holds that

$$\alpha \oslash \beta(t) = \begin{cases} R(t + T - t_a) + b_a + r_a t_a, & \text{if } t \leq t_a - T, \\ \alpha(t + T), & \text{otherwise.} \end{cases}$$

Proof. Calculate the deconvolution $\alpha \oslash \beta(t)$ by calculating the result for each different case:

$$\alpha \oslash \beta(t) = \sup_{s \geq 0}\{\alpha(t+s) - \beta(s)\}$$

Case I: $t \leq t_a - T$

$$\begin{aligned}
\alpha \oslash \beta(t) &= \sup_{0 \leq s \leq T}\{\alpha(t+s) - \beta(s)\} \vee \sup_{s > T}\{\alpha(t+s) - \beta(s)\} \\
&\overset{(s=T)}{=} \alpha(t+T) \vee \sup_{T < s \leq t_a - t}\{\alpha(t+s) - \beta(s)\} \vee \sup_{s > t_a - t}\{\alpha(t+s) - \beta(s)\} \\
&= \alpha(t+T) \vee \sup_{T < s \leq t_a - t}\{r_a(t+s) + b_a - R(s-T)\} \\
&\quad \vee \sup_{s > t_a - t}\{r_a(t+s) + b_a - R(s-T)\} \\
&\overset{(s=t_a - t)}{=} \alpha(t+T) \vee r_a(t+t_a-t) + b_a - R(t_a - t - T) \\
&\quad \vee r_a(t+t_a-t) + b_a - R(t_a - t - T) \\
&= \alpha(t+T) \vee r_a(t+t_a-t) + b_a - R(t_a - t - T) \\
&= \underline{\underline{R(t+T-t_a) + b_a + r_a t_a}}
\end{aligned}$$

Case II: $t > t_a - T$
Case IIa: $t_a \leq T$

$$\begin{aligned}
\alpha \oslash \beta(t) &= \sup_{0 \leq s \leq t_1}\{\alpha(t+s) - \beta(s)\} \vee \sup_{t_1 < s \leq t_2}\{\alpha(t+s) - \beta(s)\} \vee \\
&\quad \cdots \vee \sup_{t_{a-1} < s \leq t_a}\{\alpha(t+s) - \beta(s)\} \vee \sup_{s > t_a}\{\alpha(t+s) - \beta(s)\} \\
&\overset{(s=t_1, \cdots, s=t_a)}{=} \alpha(t+t_1) - 0 \vee \cdots \vee \alpha(t+t_a) - 0 \\
&\quad \vee \sup_{t_a < s \leq T}\{\alpha(t+s) - \beta(s)\} \vee \sup_{s > T}\{\alpha(t+s) - \beta(s)\} \\
&\overset{(s=T)}{=} \alpha(t+t_1) \vee \cdots \vee \alpha(t+t_a) \vee \alpha(t+T) - 0 \vee \alpha(t+T) - 0 \\
&\overset{(t_1, \cdots, t_a \leq T)}{=} \underline{\underline{\alpha(t+T)}}
\end{aligned}$$

Case IIb: $t_a > T$

$$\begin{aligned}
\alpha \oslash \beta(t) &= \sup_{0 \leq s \leq T}\{\alpha(t+s) - \beta(s)\} \vee \sup_{s > T}\{\alpha(t+s) - \beta(s)\} \\
&= \sup_{0 \leq s \leq T}\{\alpha(t+s) - \beta(s)\} \vee \sup_{T < s \leq t_a}\{\alpha(t+s) - \beta(s)\} \vee \sup_{s > t_a}\{\alpha(t+s) - \beta(s)\} \\
&\overset{(s=T, s=t_a)}{=} \alpha(t+T) - 0 \vee \alpha(t+T) - 0 \vee \alpha(t+t_a) - \beta(t_a) \\
&= \underline{\underline{\alpha(t+T)}}
\end{aligned}$$

□

Let us now briefly compare Lemma 3.3 and Lemma 3.4. In the special case with only two linear segments, as discussed in Lemma 3.3, we had to consider and analyze more cases than in the more general setting presented in Lemma 3.4.

By defining t_a as described in Remark 3.2, we can directly identify the section relevant for our case differentiation. In the scenario with only two segments in the arrival curve from Lemma 3.3, we must also determine through case distinctions whether the relevant section occurs before the intersection at t_2 or not. Therefore, it is more straightforward to deal directly with the section of interest (from the intersection t_a) than to do multiple case distinctions.

Nevertheless, we observe that for an arrival curve with two segments, both lemmas yield the same results.

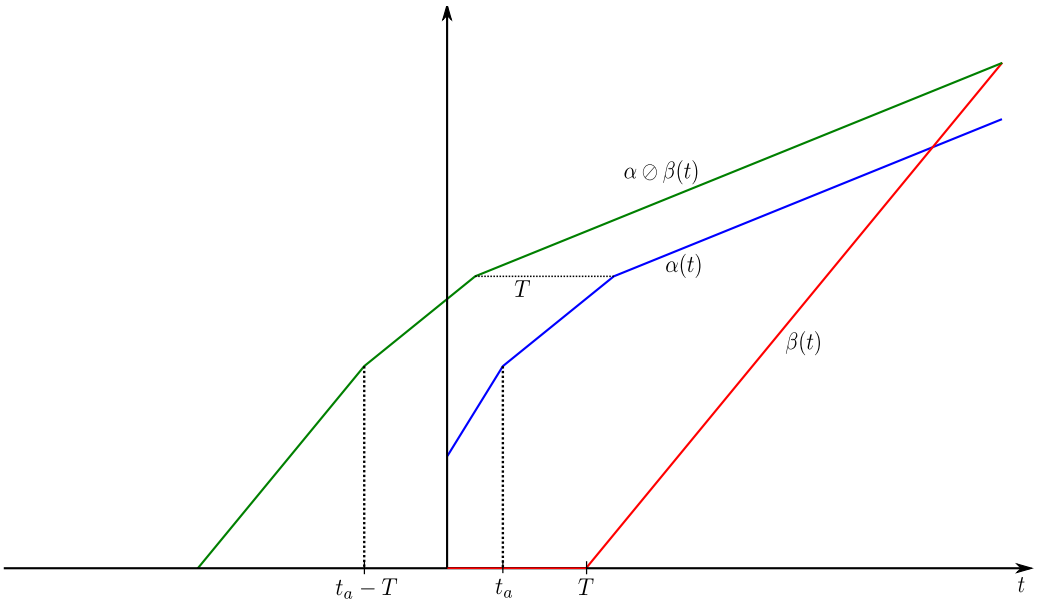


Figure 3.3: Min-plus deconvolution of $\alpha(t)$ and $\beta(t)$.

Finally, let us examine an example of deconvolution illustrated in Figure 3.3, as per Lemma 3.4.

Here, we have a blue arrival curve divided into three linear sections and a red rate-latency service curve. We observe that t_a lies between the first and second sections of the arrival curve. Thus, from the second section onward, the service rate exceeds that of the arrival, considering our assumption of piecewise linear curves in normal form.

Examining the green deconvolution curve, we encounter the first case of Lemma 3.4 for $t \leq t_a - T$ and the second case for $t > t_a - T$. Particularly noteworthy is the second case, which demonstrates a clear shift of the arrival curve to the left by T.

3.2 Performance Bounds

This section represents a significant step forward as we delve into addressing the performance bounds. We consider a scenario characterized by a single flow and a single server, as illustrated in Figure 3.4. It is worth noting that this simplified setting is adequate for our purposes, as it allows us to break down more complex network topologies and settings to this elementary case. To derive the performance bounds, we leverage the insights obtained in Section 3.1. Here, we adeptly utilize our understanding of min-plus algebra to calculate the convolution and deconvolution operations tailored specifically for piecewise linear arrival curves and rate-latency service curves.

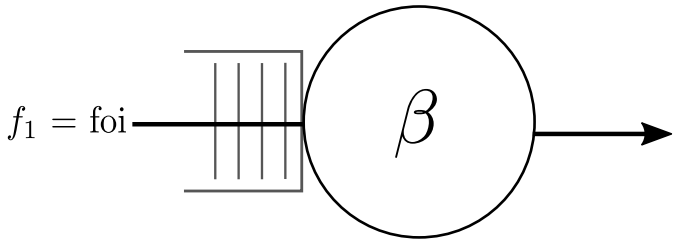


Figure 3.4: Single arrival flow at one node with service curve β.

3.2.1 Backlog Bound

Theorem 3.5. *Let $\alpha(t) = \min\{\gamma_i\}(t)$, $i \in [1, n]$ be a piecewise linear concave function in normal form, with $\gamma_i(t) = \gamma_{r_i,b_i}(t) = r_i \cdot t + b_i$ being token bucket arrival curves. Let $\beta(t) = \beta_{R,T}(t) = R \cdot [t - T]^+$ be a rate-latency service curve.*

Let t_i be the sequence of intersections as defined in Definition 2.27 and let t_a be defined as follows:

- *If $i \geq 2 \wedge (\exists t_i : r_{i-1} \geq R \wedge r_i < R)$:*
  We define t_a, $a \in [2, n]$, as the intersection of γ_{a-1} and γ_a for which holds that $r_{a-1} \geq R$ and $r_a < R$.
- *Otherwise (if $i < 2 \vee (\forall r_i : r_i < R)$):*
  We set $t_a = t_1$.

Then it holds for all $t \geq 0$

$$q(t) \leq \begin{cases} \alpha(T), & \text{if } t_a \leq T, \\ \alpha(t_a) - \beta(t_a), & \text{otherwise.} \end{cases}$$

Proof. Calculate the backlog bound $q(t)$ by calculating the result for each different case:

$$q(t) \leq \sup_{s>0}\{\alpha(s) - \beta(s)\}$$

Case I: $t_a \leq T$

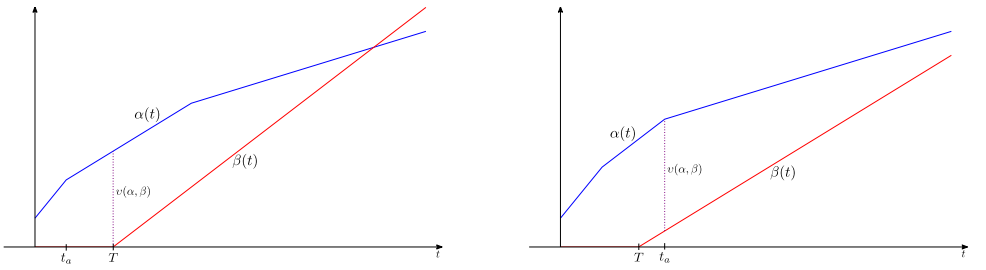
$$\begin{aligned} q(t) &\leq \sup_{0<s\leq t_1}\{\alpha(s)-\beta(s)\} \vee \cdots \vee \sup_{t_{a-1}<s\leq t_a}\{\alpha(s)-\beta(s)\} \vee \sup_{s>t_a}\{\alpha(s)-\beta(s)\} \\ &= \sup_{0<s\leq t_1}\{\alpha(s)\} \vee \cdots \vee \sup_{t_{a-1}<s\leq t_a}\{\alpha(s)\} \vee \sup_{t_a<s\leq T}\{\alpha(s)-\beta(s)\} \\ &\quad \vee \sup_{s>T}\{\alpha(s)-\beta(s)\} \\ &\overset{(s=t_1,\cdots,s=t_a,s=T)}{=} \{\alpha(t_1)\} \vee \cdots \vee \{\alpha(t_a)\} \vee \{\alpha(T)\} \vee \{\alpha(T)\} \\ &\overset{(t_1,\cdots,t_a\leq T)}{=} \alpha(T) \end{aligned}$$

Case II: $t_a > T$

$$\begin{aligned} q(t) &\leq \sup_{0<s\leq t_a}\{\alpha(s)-\beta(s)\} \vee \sup_{s>t_a}\{\alpha(s)-\beta(s)\} \\ &\overset{(s=t_a)}{=} \{\alpha(t_a)-\beta(t_a)\} \vee \{\alpha(t_a)-\beta(t_a)\} \\ &= \alpha(t_a)-\beta(t_a) \end{aligned}$$

□

So let us take a closer look at the result of calculating the backlog bound from Theorem 3.5, illustrated in Figure 3.5. As described in Remark 3.2, we again select the intersection t_a which is of interest to us. If $t_a \leq T$ applies, as can be seen in Figure 3.5a, then the vertical deviation increases up to time $t = T$, as the service rate is greater than the arrival rate(s) from then on. If $t_a > T$ applies, as can be seen in Figure 3.5b, the t_a indicates when the service rate becomes greater than the arrival rate and we know that the vertical deviation must be greatest at this point.



(a) Backlog bound for $t_a \leq T$ (Case I). (b) Backlog bound for $t_a > T$ (Case II).

Figure 3.5: Backlog bound: case distinction.

3.2.2 Delay Bound

Theorem 3.6. *Let $\alpha(t) = \min\{\gamma_i\}(t)$, $i \in [1, n]$ be a piecewise linear concave function in normal form, with $\gamma_i(t) = \gamma_{r_i,b_i}(t) = r_i \cdot t + b_i$ being token bucket arrival curves. Let $\beta(t) = \beta_{R,T}(t) = R \cdot [t - T]^+$ be a rate-latency service curve.*

Let t_i be the sequence of intersections as defined in Definition 2.27 and let t_a be defined as follows:

- *If $i \geq 2 \wedge (\exists t_i : r_{i-1} \geq R \wedge r_i < R)$:*
  We define t_a, $a \in [2, n]$, as the intersection of γ_{a-1} and γ_a for which holds that $r_{a-1} \geq R$ and $r_a < R$.
- *Otherwise (if $i < 2 \vee (\forall r_i : r_i < R)$):*
  We set $t_a = t_1$.

The delay is then bounded by:

$$d(t) \leq T - t_a + \frac{b_a + r_a t_a}{R}.$$

Proof. The proof follows the definitions of delay bound and deconvolution. According to Theorem 2.19 we know that

$$d(t) \leq \inf\{\tau \geq 0 \mid \alpha \oslash \beta(-\tau) \leq 0\}.$$

Using this in combination with the deconvolution given in Lemma 3.4 we have to consider two cases.

Case I: $t_a \leq T$

$$\begin{aligned}
d(t) &\leq \inf\{\tau \geq 0 \mid \alpha \oslash \beta(-\tau) \leq 0\} \\
&= \inf\{0 \leq \tau < T - t_a \mid \alpha \oslash \beta(-\tau) \leq 0\} \wedge \inf\{\tau \geq T - t_a \mid \alpha \oslash \beta(-\tau) \leq 0\} \\
&= \inf\{0 \leq \tau < T - t_a \mid \alpha(-\tau + T) \leq 0\} \\
&\quad \wedge \inf\{\tau \geq 0 \mid R(T - t_a - \tau) + b_a + r_a t_a \leq 0\} \\
&= \{\infty\} \wedge \inf\{\tau \geq 0 \mid T - t_a + \frac{b_a + r_a t_a}{R} - \tau \leq 0\} \qquad (3.0) \\
&= \inf\{\tau \geq 0 \mid T - t_a + \frac{b_a + r_a t_a}{R} \leq \tau\} \\
&= T - t_a + \frac{b_a + r_a t_a}{R}
\end{aligned}$$

In line 5 (Equation (3.0)) we used that $0 \leq \tau < T - t_a$ and therefore $(-\tau + T) \in (t_a, T]$ and $\alpha(-\tau + T) > 0$ holds.

Case II: $t_a > T$

$$\begin{aligned} d(t) &\leq \inf\{\tau \geq 0 \mid \alpha \oslash \beta(-\tau) \leq 0\} \\ &= \inf\{\tau \geq 0 \mid R(T - t_a - \tau) + b_a + r_a t_a \leq 0\} \\ &= \inf\{\tau \geq 0 \mid T - t_a + \frac{b_a + r_a t_a}{R} - \tau \leq 0\} \\ &= \inf\{\tau \geq 0 \mid T - t_a + \frac{b_a + r_a t_a}{R} \leq \tau\} \\ &= T - t_a + \frac{b_a + r_a t_a}{R} \end{aligned}$$

It follows that

$$d(t) \leq T - t_a + \frac{b_a + r_a t_a}{R}.$$

□

In Case II of the proof of Theorem 3.6, we do not need to consider both cases of the calculation of the deconvolution, given in Lemma 3.4, since $t_a - T > 0$ holds.

3.2.3 Output Bound

Theorem 3.7. *Let $\alpha(t) = \min\{\gamma_i\}(t)$, $i \in [1, n]$ be a piecewise linear concave function in normal form, with $\gamma_i(t) = \gamma_{r_i,b_i}(t) = r_i \cdot t + b_i$ being token bucket arrival curves. Let $\beta(t) = \beta_{R,T}(t) = R \cdot [t - T]^+$ be a rate-latency service curve.*

Let t_i be the sequence of intersections as defined in Definition 2.27 and let t_a be defined as follows:

- *If $i \geq 2 \wedge (\exists t_i : r_{i-1} \geq R \wedge r_i < R)$:*
  We define t_a, $a \in [2, n]$, as the intersection of γ_{a-1} and γ_a for which holds that $r_{a-1} \geq R$ and $r_a < R$.
- *Otherwise (if $i < 2 \vee (\forall r_i : r_i < R)$):*
  We set $t_a = t_1$.

The output is bounded by:

$$\alpha'(t) = \begin{cases} R(t + T - t_a) + b_a + r_a t_a, & \text{if } 0 < t \leq t_a - T, \\ \alpha(t + T), & \text{if } 0 < t_a - T < t, \\ 0, & \text{otherwise.} \end{cases}$$

Proof. The proof follows the definitions of output bound and deconvolution. According to Theorem 2.20 we know that

$$\alpha'(t) = \begin{cases} \alpha \oslash \beta(t), & \text{if } t > 0, \\ 0, & \text{otherwise.} \end{cases}$$

If we use the results of the deconvolution (see Lemma 3.4) we get the following

$$\alpha'(t) = \begin{cases} R(t + T - t_a) + b_a + r_a t_a, & \text{if } 0 < t \leq t_a - T, \\ \alpha(t + T), & \text{if } 0 < t_a - T < t, \\ 0, & \text{otherwise.} \end{cases}$$

□

3.2.4 Maximum Length of a Backlogged Period

Theorem 3.8. *Let $\alpha(t) = \min\{\gamma_i\}(t)$, $i \in [1,n]$ be a piecewise linear concave function in normal form, with $\gamma_i(t) = \gamma_{r_i,b_i}(t) = r_i \cdot t + b_i$ being token bucket arrival curves. Let $\beta(t) = \beta_{R,T}(t) = R \cdot [t-T]^+$ be a rate-latency service curve. Let t_i be the sequence of intersections as defined in Definition 2.27.*
If $i \geq 2$, let A and B be sets of intersections for which the following holds:

$$A = \{t_i \mid \forall t_i, i \in [1,n] : \alpha(t_i) \geq \beta(t_i)\} \text{ and}$$
$$B = \{t_i \mid \forall t_i, i \in [1,n] : \alpha(t_i) < \beta(t_i)\}.$$

Further let $t_a = \max A$ and $t_b = \min B$. Let $\gamma_x(t) = r_x \cdot t + b_x$ be the function that is used for the interval $[t_a, t_b]$, so $\alpha(t) = \gamma_x(t)$, with $t \in [t_a, t_b]$. If $i = 1$, $\gamma_x(t) = \gamma_1(t)$.

The maximum length of a backlogged period is upper bounded by:

$$\inf\{t > 0 \mid \alpha(t) < \beta(t)\} = \frac{b_x + RT}{R - r_x}.$$

Proof. There are two cases to consider: $t \leq T$ and $t > T$. Since for $t \leq T$ the infimum is infinite, it is enough to consider $t > T$.

$$\inf\{t > 0 \mid \alpha(t) < \beta(t)\} = \inf\{t > 0 \mid \min\{\gamma_1(t), \cdots, \gamma_n(t)\} < R \cdot (t-T)\}$$

For the calculation of the infimum we are interested in the intersection of α and β. We know that this intersection has to be in the interval $[t_a, t_b]$ and therefore $\gamma_x(t)$ is the used function for this interval. For the infimum this is the only function γ_i that is important for the calculation.

$$\begin{aligned}
\inf\{t > 0 \mid \alpha(t) < \beta(t)\} &= \inf\{t > 0 \mid \gamma_x < R \cdot (t-T)\} \\
&= \inf\{t > 0 \mid r_x \cdot t + b_x < R \cdot (t-T)\} \\
&= \inf\{t > 0 \mid b_i + RT < (R - r_x) \cdot t\} \\
&= \inf\{t > 0 \mid \frac{b_x + RT}{R - r_x} < t\} \\
&= \frac{b_x + RT}{R - r_x}.
\end{aligned}$$

□

3.2.5 Numerical Example

Now that we can calculate the performance bounds, let us consider a concrete example with concrete numerical values. The example was created using our toolbox (see Chapter 6) and can be found under *numerical_example_chapter_3.py*.

Let us first look at the arrival curve. The following token bucket arrival curves are given:

- $\gamma_1(t) = \gamma_{r_1,b_1}(t) = \gamma_{2,3}(t)$,
- $\gamma_2(t) = \gamma_{r_2,b_2}(t) = \gamma_{1,5}(t)$,
- $\gamma_3(t) = \gamma_{r_3,b_3}(t) = \gamma_{\frac{1}{2},10}(t)$.

Let the concave piecewise linear arrival curve be given by:

$$\alpha(t) = \min\{\gamma_1, \gamma_2, \gamma_3\}(t).$$

The service curve is given by:

$$\beta(t) = \beta_{R,T}(t) = \beta_{\frac{5}{4},5}(t).$$

We see that $r_1 > R$ and $r_2 < R$ holds, so therefore $t_a = t_2 = 2$.

If we now apply Theorem 3.5, 3.6 and 3.8, we receive the following performance bounds:

- backlog bound: $v(\alpha, \beta) = 10.0$,
- delay bound: $h(\alpha, \beta) = 8.6$,
- maximum length of backlogged period: maxBP = $21\frac{2}{3}$.

The output bound, $\alpha \oslash \beta(t)$ with $t > 0$, is calculated by applying Theorem 3.7.

To illustrate everything even better, let us take a look at Figure 3.6. The performance bounds are shown there respectively.

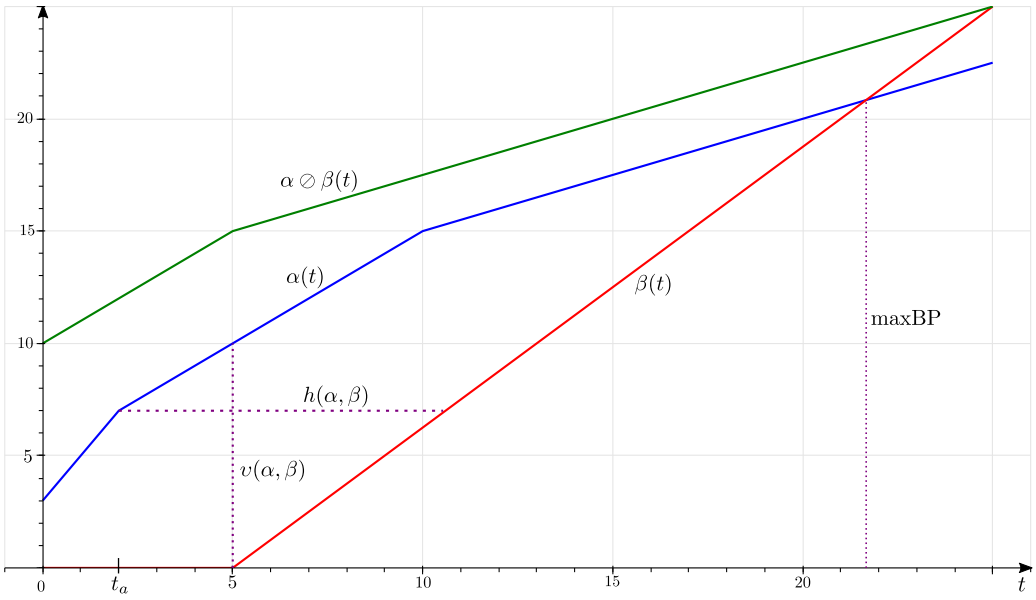


Figure 3.6: Numerical example: performance bounds.

4 Single Flow Crosses One Server: Piecewise Linear Arrival Curve with Piecewise Linear Service Curve

Having thoroughly explored the case of piecewise linear arrival curves and rate-latency service curves, we now advance to the study of piecewise linear arrival and service curves. Firstly, in Section 4.1, we will derive the min-plus convolution and deconvolution. Subsequently, in Section 4.2, we will employ these tools to calculate the performance bounds for our new setting. By following a methodology analogous to that presented in Chapter 3, we ensure a consistent approach to our analysis.

4.1 Min-Plus Convolution and Deconvolution

In this section, we lay the groundwork essential for the calculation of performance bounds. In Subsection 4.1.1, we delve into the computation of the min-plus convolution. Following this, in Subsection 4.1.2, we turn our attention to the computation of the min-plus deconvolution. Together, these subsections provide a detailed foundation for the subsequent performance bound computation.

4.1.1 Convolution

Lemma 4.1. *Let $\alpha(t) = \min\{\gamma_i\}(t)$, $i \in [1, m]$ be a concave piecewise linear arrival curve in normal form, with $\gamma_i(t) = \gamma_{r_i,b_i}(t) = r_i \cdot t + b_i$ being token bucket arrival curves. Let $\beta(t) = \max\{\beta_j\}(t)$, $j \in [1, n]$ be a convex piecewise linear service curve in normal form, with $\beta_j(t) = \beta_{R_j,T_j}(t) = R_j \cdot [t - T_j]^+$ being rate-latency service curves.*

Then it holds that

$$\alpha \otimes \beta(t) = \begin{cases} 0, & \text{if } t \leq T, \\ \min\{\beta(t), \alpha(t-T)\}, & \text{otherwise.} \end{cases}$$

Proof. Let t_i and u_j be the sequence of intersections as defined in Definition 2.27 and Definition 2.28 and let $I = \{t_i \mid i \in [1, m]\} \cup \{u_j | j \in [1, n]\}$ be the set of all intersections. We define

$$a = \min\{x \in I : (\alpha(x) = \gamma_y(x) \wedge \beta(x) = \beta_z(x)) \wedge r_y \leq R_z\}$$

as the first intersection for which the rate of the arrival curve is less or equal to the rate of the service curve.

Calculate the convolution $\alpha \otimes \beta(t)$ by calculating the result for each different case:

$$\alpha \otimes \beta(t) = \inf_{0 \le s \le t} \{\alpha(t-s) + \beta(s)\}.$$

Case I: $t \le T$

$$\alpha \otimes \beta(t) = \inf_{0 \le s \le t} \{\alpha(t-s) + 0\} \overset{(s=t)}{=} \alpha(0) = \underline{0}$$

Case II: $t > T$

$$\begin{aligned}\alpha \otimes \beta(t) &= \inf_{0 \le s \le T} \{\alpha(t-s) + \beta(s)\} \wedge \inf_{T < s < t} \{\alpha(t-s) + \beta(s)\} \wedge \inf_{s=t} \{\alpha(t-s) + \beta(s)\} \\ &\overset{(s=T, s=t)}{=} \{\alpha(t-T) + 0\} \wedge \inf_{T < s < t} \{\alpha(t-s) + \beta(s)\} \wedge \{\alpha(0) + \beta(t)\} \\ &= \{\alpha(t-T)\} \wedge \{\beta(t)\} \wedge \inf_{T < s < t} \{\alpha(t-s) + \beta(s)\}\end{aligned}$$

Case IIa: $a \le T$

$$\begin{aligned}\alpha \otimes \beta(t) &\overset{(s=T)}{=} \{\alpha(t-T)\} \wedge \{\beta(t)\} \wedge \{\alpha(t-T)\} \\ &= \underline{\underline{\min\{\beta(t), \alpha(t-T)\}}}\end{aligned}$$

Case IIb: $a > T$
Case IIba: $a < t$

$$\begin{aligned}\alpha \otimes \beta(t) &\overset{(s=T)}{=} \{\alpha(t-T)\} \wedge \{\beta(t)\} \wedge \{\alpha(t-T)\} \\ &= \underline{\underline{\min\{\beta(t), \alpha(t-T)\}}}\end{aligned}$$

Case IIbb: $a \ge t$

$$\begin{aligned}\alpha \otimes \beta(t) &\overset{(s=t)}{=} \{\alpha(t-T)\} \wedge \{\beta(t)\} \wedge \{\beta(t)\} \\ &= \underline{\underline{\min\{\beta(t), \alpha(t-T)\}}}\end{aligned}$$

□

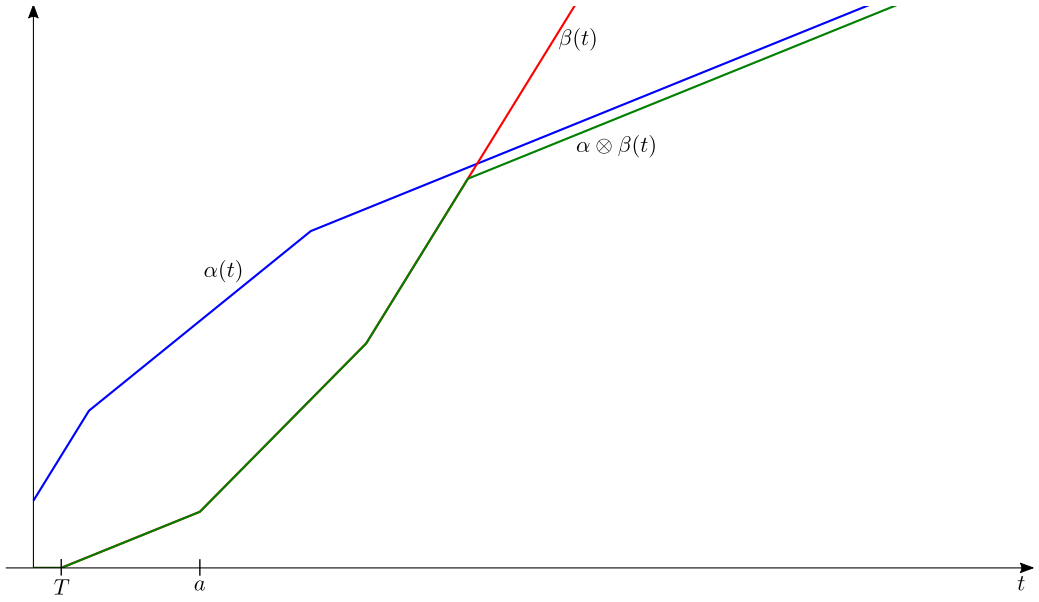


Figure 4.1: Min-plus convolution of $\alpha(t)$ and $\beta(t)$.

Let us now closely examine the convolution as described in Lemma 4.1. An example illustrating this concept is provided in Figure 4.1. In this example, we focus on Case II, where the condition $a > T$ holds, leading us specifically to Case IIb. In the figure, the arrival curve $\alpha(t)$ is depicted in blue and the service curve $\beta(t)$ is shown in red. The convolution of these two curves is depicted in green.

4.1.2 Deconvolution

Before we turn our attention to the deconvolution, we first need to introduce and define a crucial set, denoted as A. This set A consists of the arrival curves $\alpha(t)$, each shifted by an amount s. Introducing this set A allows us to streamline the notation and formulation for the deconvolution.

Definition 4.2. For a given arrival curve $\alpha(t)$ and a given $s \in \mathbb{R}$, we define

$$\alpha'_s(t) = \alpha \otimes \delta_{-s}(t)$$

as the arrival curve shifted to the left by s (see Remark 2.7). So for $s \in \mathbb{R}$ we get a set of arrival curves A, with

$$A = \{\alpha'_s(t) | s \in \mathbb{R}\}.$$

With this groundwork laid, we are now prepared to proceed with the deconvolution, leveraging the simplified notation provided by set A.

Lemma 4.3. *Let $\alpha(t) = \min\{\gamma_i\}(t)$, $i \in [1, m]$ be a piecewise linear concave function in normal form, with $\gamma_i(t) = \gamma_{r_i,b_i}(t) = r_i \cdot t + b_i$ being token bucket arrival curves.*

Let $\beta(t) = \max\{\beta_j\}(t)$, $j \in [1, n]$ be a piecewise linear convex function in normal form, with $\beta_j(t) = \beta_{R_j,T_j}(t) = R_j \cdot [t - T_j]^+$ being rate-latency service curves.

Let t_i and u_j be the sequence of intersections as defined in Definition 2.27 and Definition 2.28.

Let $I = \{t_i | i \in [1, m]\} \cup \{u_j | j \in [1, n]\}$ be the set of all intersections.

Let $\alpha'_t \in A$, with A being defined as in Definition 4.2.

Further, we define

$$a(t) = \min\{x \in I : (\alpha'_t(x) = \gamma'_y(x) \wedge \beta(x) = \beta_z(x)) \wedge r_y \leq R_z\} \tag{4.1}$$

as the first intersection for which the rate of the respective arrival curve α'_t is less or equal to the rate of the service curve.

Then it holds that

$$\alpha \oslash \beta(t) = v(\alpha'_t, \beta) = \alpha(a(t)) - \beta(a(t)).$$

Proof. According to Remark 2.8 it holds that

$$\alpha \oslash \beta(t) = v(\alpha \otimes \delta_{-t}, \beta) = v(\alpha'_t, \beta).$$

We know that the backlog bound can be calculated for each shifted arrival curve of the set of arrival curves α'_t and β.

For the calculation of the backlog bound we get for each α'_t a new value a, with a being the first intersection for which the rate of the arrival curve is less or equal to the rate of the service curve.

The respective a for the calculation of the backlog bound of $\alpha'_t(t)$ and $\beta(t)$ can be described by

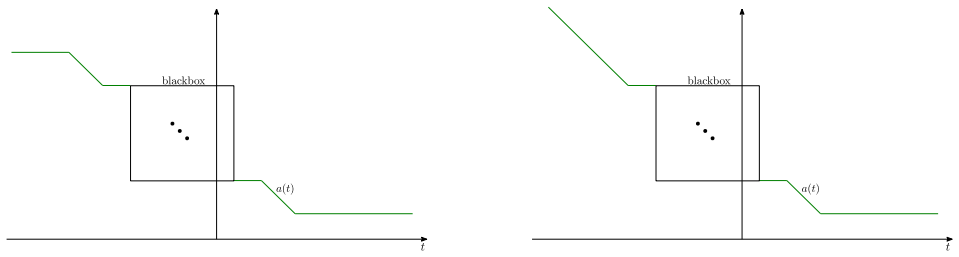
$$a(t) = \min\{x \in I : (\alpha'_t(x) = \gamma'_y(x) \wedge \beta(x) = \beta_z(x)) \wedge r_y \leq R_z\}.$$

Then it holds, according to Theorem 4.4, that

$$\alpha \oslash \beta(t) = v(\alpha'_t, \beta) = \alpha(a(t)) - \beta(a(t)).$$

□

In Lemma 4.3, Equation (4.1) define $a(t)$, which is essentially crucial for the calculation of the deconvolution. Let us take a brief look at how $a(t)$ essentially appears. Figure 4.2 illustrate how we can conceptualize $a(t)$.



(a) Constant section. (b) Decreasing section.

Figure 4.2: Illustration of the function $a(t)$.

We observe a staircase function, with a constant section on the right hand side and either a constant section (Figure 4.2a) or a linearly decreasing section on the left hand side (Figure 4.2b). Depending on the definition of the arrival and service curves, additional steps are added within the black box. Let us first consider the constant section on the right side hand side ($t > 0$), as depicted in Figure 4.2. As t increases, we shift the arrival curve to the left, ultimately retaining only the last linear section of the original arrival curve. Therefore, once we reach the last intersection where the final section begins, $a(t)$ also stops changing since we remain constant at an intersection where the arrival curve rate is less than or equal to the service rate.

We can proceed analogously for the shift to the right, where eventually, with the shifted arrival curve, we start at the last intersection point of the service curve, thus remaining constant with the intersections crucial for $a(t)$ (Figure 4.2a), unless as shown in Figure 4.2b, where the intersection used for $a(t)$ is one of the shifted arrival curves, in which case, it will be shifted indefinitely.

The staircases within the black box can also be easily explained. For any given t, the value of $a(t)$ assumes a unique intersection where the shifted arrival curve

rate at that point is less than or equal to the service curve rate. This intersection can be from the shifted arrival curve or the service curve. Therefore, if initially, for a given t, the intersection for $a(t)$ is from the service curve, and then for increasing t, suddenly the intersection for $a(t)$ switches to that of the shifted arrival curve, we obtain a constant stair-step for the duration of this linear section of the arrival or service curve until a new intersection is used.

The interactive plots provided by the Toolbox:DNC-PWL (refer to Chapter 6) can vividly illustrate all of this.

Now, let us delve into an illustrative example showcasing the deconvolution process described in Lemma 4.3. For this demonstration, we turn our attention to Figure 4.3. In this figure, we are presented with the deconvolution of the arrival curve $\alpha(t)$ depicted in blue and the service curve $\beta(t)$ shown in red. The resulting deconvolution curve is represented in green.

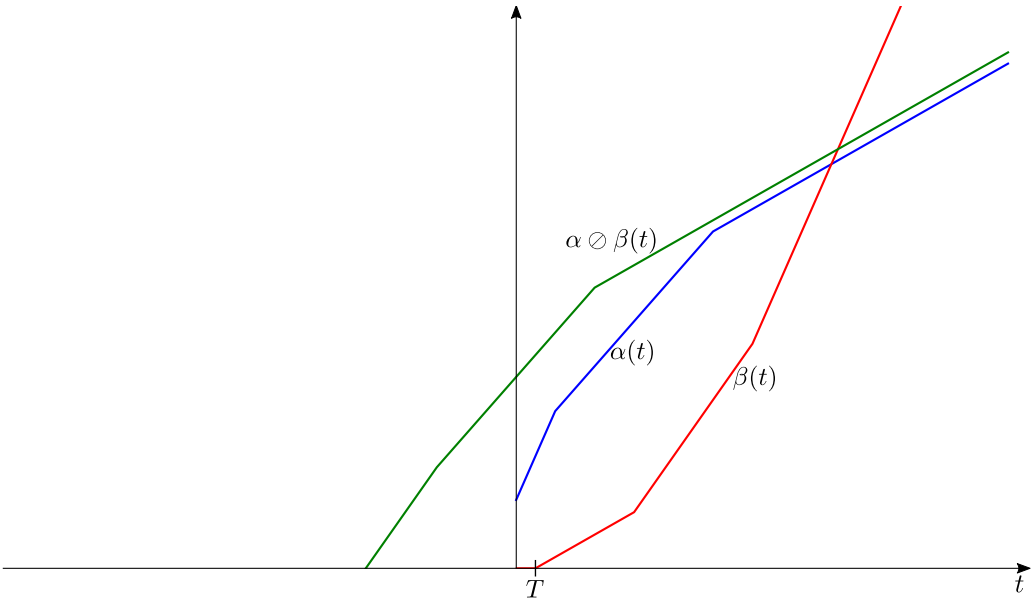


Figure 4.3: Min-plus deconvolution of $\alpha(t)$ and $\beta(t)$.

Additionally, Figure 4.4 complements our understanding by showcasing the corresponding function $a(t)$, which plays a crucial role in the deconvolution calculation depicted in Figure 4.3.

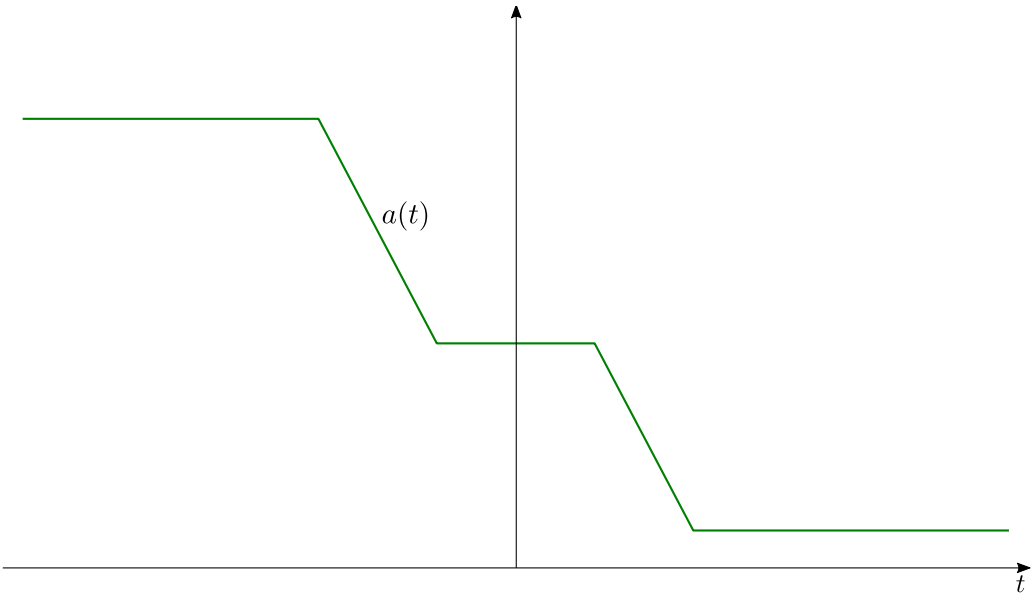


Figure 4.4: Function $a(t)$.

4.2 Performance Bounds

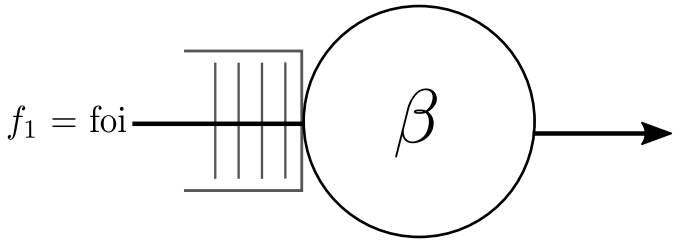


Figure 4.5: Single arrival flow at one node with service curve β.

4.2.1 Backlog Bound

Theorem 4.4. *Let $\alpha(t) = \min\{\gamma_i\}(t)$, $i \in [1, m]$ be a piecewise linear concave function in normal form, with $\gamma_i(t) = \gamma_{r_i,b_i}(t) = r_i \cdot t + b_i$ being token bucket arrival curves.*

Let $\beta(t) = \max\{\beta_j\}(t)$, $j \in [1, n]$ be a piecewise linear convex function in normal form, with $\beta_j(t) = \beta_{R_j,T_j}(t) = R_j \cdot [t - T_j]^+$ being rate-latency service curves.

Let t_i and u_j be the sequence of intersections as defined in Definition 2.27 and Definition 2.28 and let $I = \{t_i | i \in [1, m]\} \cup \{u_j | j \in [1, n]\}$ be the set of all intersections.

We define

$$a = \min\{x \in I : (\alpha(x) = \gamma_y(x) \wedge \beta(x) = \beta_z(x)) \wedge r_y \leq R_z\}$$

as the first intersection for which the rate of the arrival curve is less or equal to the rate of the service curve.

Then it holds for all $t \geq 0$:

$$q(t) \leq \begin{cases} \alpha(T_1), & \text{if } a \leq T_1, \\ \alpha(a) - \beta(a), & \text{otherwise.} \end{cases}$$

Proof. Calculate the backlog bound $q(t)$ by calculation the result for each different case:

$$q(t) \leq \sup_{s>0}\{\alpha(s) - \beta(s)\}$$

Case I: $a \leq T_1$

$$\begin{aligned} q(t) &= \sup_{0<s\leq T_1}\{\alpha(s) - \beta(s)\} \vee \sup_{s>T_1}\{\alpha(s) - \beta(s)\} \\ &\overset{(s=T_1)}{=} \{\alpha(T_1) - 0\} \vee \{\alpha(T_1) - 0\} \\ &= \alpha(T_1) \end{aligned}$$

Case II: $a > T_1$

$$\begin{aligned} q(t) &= \sup_{0<s\leq a} \{\alpha(s) - \beta(s)\} \vee \sup_{s>a} \{\alpha(s) - \beta(s)\} \\ &\overset{(s=a)}{=} \{\alpha(a) - \beta(a)\} \vee \{\alpha(a) - \beta(a)\} \\ &= \alpha(a) - \beta(a) \end{aligned}$$

□

Remark 4.5. It is easy to see that there exists an unique minimal intersection a for which the rate of the arrival curve is less or equal to the rate of the service curve, since the stability condition has to be fulfilled, the arrival curve is concave and the service curve is convex.

4.2.2 Delay Bound

Theorem 4.6. *Let $\alpha(t) = \min\{\gamma_i\}(t)$, $i \in [1, m]$ be a piecewise linear concave function in normal form, with $\gamma_i(t) = \gamma_{r_i,b_i}(t) = r_i \cdot t + b_i$ being token bucket arrival curves.*

Let $\beta(t) = \max\{\beta_j\}(t)$, $j \in [1, n]$ be a piecewise linear convex function in normal form, with $\beta_j(t) = \beta_{R_j,T_j}(t) = R_j \cdot [t - T_j]^+$ being rate-latency service curves.

Let t_i and u_j be the sequence of intersections as defined in Definition 2.27 and Definition 2.28

Let $T = \{t_i | i \in [1, m]\}$ be the set of intersections of $\alpha(t)$ and let $U = \{u_j | j \in [1, n]\}$ be the set of intersections of $\beta(t)$.

For t_{min} and u_{min} the following holds:

$$\begin{aligned} t_{min} &= \min\{t \in T : \alpha(t) = \alpha_x(t) \wedge \beta(b) = \beta_y(b) = \alpha(t) \wedge r_x \leq R_y\}, \\ u_{min} &= \min\{u \in U : \beta(u) = \beta_x(u) \wedge \alpha(a) = \alpha_y(a) = \beta(u) \wedge R_x \geq r_y\}, \end{aligned} \tag{4.2}$$

with $b = \beta^{-1}(\alpha(t))$ and $a = \alpha^{-1}(\beta(u))$.

Let $a_{min} = \alpha^{-1}(\beta(u_{min}))$ and $b_{min} = \beta^{-1}(\alpha(t_{min}))$

Let $d_\alpha = b_{min} - t_{min}$ and $d_\beta = u_{min} - a_{min}$.

Then the delay is bounded by:

$$d(t) \leq \max\{d_\alpha, d_\beta\}$$

Proof. To begin this proof, let us address why we can assume the delay bound at an intersection of the arrival curve $\alpha(t)$ or the service curve $\beta(t)$ and why only the intersections need to be considered.

To demonstrate this aspect, let us proceed with a proof by contradiction.

Suppose the delay bound, denoted by the horizontal deviation, is assumed or calculated at the point $\alpha(x_\alpha) = \beta(x_\beta)$, where $x_\alpha \notin T$ and $x_\beta \notin U$, implying $d(t) \leq x_\beta - x_\alpha$.

Now, let us scrutinize the linear segments of $\alpha(t)$ and $\beta(t)$ at $t_\alpha = x_\alpha$ and $t_\beta = x_\beta$ respectively. Specifically, we will focus on the arrival rate r_{x_α} and the service rate R_{x_β}.

It becomes apparent that $r_{x_\alpha} \leq R_{x_\beta}$ must hold. If $r_{x_\alpha} > R_{x_\beta}$, then $\alpha(t)$ and $\beta(t)$ would diverge further apart, as the arrival curve $\alpha(t)$ would outpace the service curve $\beta(t)$) This contradicts the assumed convergence at $\alpha(x_\alpha) = \beta(x_\beta)$.

Hence, we establish that $r_{x_\alpha} \leq R_{x_\beta}$. Consequently, as t increases, $\alpha(t)$ and $\beta(t)$ converge, thus diminishing the horizontal separation. Conversely, as t decreases, $\alpha(t)$ and $\beta(t)$ diverge, increasing the horizontal distance. By gradually reducing t at the delay bound calculation point until reaching the first intersection of $\alpha(t)$ or

$\beta(t)$ where $r \leq R$, we clearly obtain a larger distance than the initially assumed point from above. This contradiction refutes the initial assumption. $\lightning$

Therefore, the point yielding the maximum distance, and thus the delay bound, must reside at an intersection point of the arrival or service curve.

To complete the proof, we now need to efficiently determine this intersection of $\alpha(t)$ or $\beta(t)$.

For this purpose, we only need to consider the intersections where $r < R$. Additionally, only the first intersection point of the arrival and service curve where $r < R$ holds is of interest. Let us denote these points as t_{min} and u_{min}, and we obtain them as follows:

$$t_{min} = \min\{t \in T : \alpha(t) = \alpha_x(t) \wedge \beta(b) = \beta_y(b) = \alpha(t) \wedge r_x \leq R_y\},$$
$$u_{min} = \min\{u \in U : \beta(u) = \beta_x(u) \wedge \alpha(a) = \alpha_y(a) = \beta(u) \wedge R_x \geq r_y\},$$

with $b = \beta^{-1}(\alpha(t))$ and $a = \alpha^{-1}(\beta(u))$. Here, $\beta^{-1}(t)$ and $\alpha^{-1}(t)$ represent the pseudo-inverse of $\beta(t)$ and $\alpha(t)$ respectively (see Lemma 2.32).

Now, to determine the horizontal distance between $\alpha(t)$ and $\beta(t)$ at the points t_{min} and u_{min}, we must calculate the time at which the service curve reaches the value $\alpha(t_{min})$ or the time at which the arrival curve reaches the value $\beta(u_{min})$. Thus, we calculate:

$$a_{min} = \alpha^{-1}(\beta(u_{min})),$$
$$b_{min} = \beta^{-1}(\alpha(t_{min})).$$

The distances are then obtained as:

$$d_\alpha = b_{min} - t_{min},$$
$$d_\beta = u_{min} - a_{min}.$$

Consequently, the delay bound is the larger of the two distances:

$$d(t) \leq \max\{d_\alpha, d_\beta\}.$$

□

4.2.3 Output Bound

Corollary 4.7. Let $\alpha(t) = \min\{\gamma_i\}(t)$, $i \in [1, m]$ be a piecewise linear concave function in normal form, with $\gamma_i(t) = \gamma_{r_i,b_i}(t) = r_i \cdot t + b_i$ being token bucket arrival curves. Let $\beta(t) = \max\{\beta_j\}(t)$, $j \in [1, n]$ be a piecewise linear convex function in normal form, with $\beta_j(t) = \beta_{R_j,T_j}(t) = R_j \cdot [t - T_j]^+$ being rate-latency service curves.

The output is bounded by:

$$\alpha'(t) = \begin{cases} \alpha \oslash \beta(t), & \text{if } t > 0, \\ 0, & \text{otherwise,} \end{cases}$$

with $\alpha \oslash \beta(t)$ using the deconvolution from Lemma 4.3.

4.2.4 Maximum Length of a Backlogged Period

Theorem 4.8. *Let $\alpha(t) = \min\{\gamma_i\}(t)$, $i \in [1,n]$ be a piecewise linear concave function in normal form, with $\gamma_i(t) = \gamma_{r_i,b_i}(t) = r_i \cdot t + b_i$ being token bucket arrival curves. Let $\beta(t) = \max\{\beta_j\}(t)$, $j \in [1,n]$ be a piecewise linear convex function in normal form, with $\beta_j(t) = \beta_{R_j,T_j}(t) = R_j \cdot [t - T_j]^+$ being rate-latency service curves. Let t_i and u_j be the sequence of intersections as defined in Definition 2.27 and Definition 2.28.*
If $i \geq 2$ resp. $j \geq 2$, let A, B, C and D be sets of intersections for which the following holds:

$$\begin{aligned} A &= \{t_i \mid \forall t_i, i \in [1,n] : \alpha(t_i) \geq \beta(t_i)\}, \\ B &= \{t_i \mid \forall t_i, i \in [1,n] : \alpha(t_i) < \beta(t_i)\}, \\ C &= \{u_j \mid \forall u_j, j \in [1,n] : \alpha(u_j) \geq \beta(u_j)\}, \\ D &= \{u_j \mid \forall u_j, j \in [1,n] : \alpha(u_j) < \beta(u_j)\}. \end{aligned}$$

Further let $t_a = \max A$, $t_b = \min B$, $u_c = \max C$ and $u_d = \min D$.
Let $\gamma_x(t) = r_x \cdot t + b_x$ be the function that is used for the interval $[t_a, t_b]$, so $\alpha(t) = \gamma_x(t)$, with $t \in [t_a, t_b]$. Let $\beta_y(t) = R_x \cdot [t - T_x]^+$ be the function that is used for the interval $[u_c, u_d]$, so $\beta(t) = \beta_y(t)$, with $t \in [u_c, u_d]$. If $i = 1$ resp. $j = 1$, $\gamma_x(t) = \gamma_1(t)$ and $\beta_y(t) = \beta_1(t)$.

The maximum length of a backlogged period is upper bounded by:

$$\inf\{t > 0 \mid \alpha(t) < \beta(t)\} = \frac{b_x + R_y T_y}{R_y - r_x}.$$

Proof. There are two cases to consider: $t \leq T_1$ and $t > T_1$. Since for $t \leq T_1$ the infimum is infinite, it is enough to consider $t > T_1$. For the calculation of the infimum we are interested in the intersection of α and β. For the arrival curve we know that this intersection has to be in the interval $[t_a, t_b]$ and therefore $\gamma_x(t)$ is the used function for this interval. For the service curve we know that this intersection has to be in the interval $[u_c, u_d]$ and therefore $\beta_y(t)$ is the used function for this interval. For the infimum these are the only functions of γ_i and β_j that are important for the calculation.

$$\begin{aligned} \inf\{t > 0 \mid \alpha(t) < \beta(t)\} &= \inf\{t > 0 \mid \gamma_x < \beta_y\} \\ &= \inf\{t > 0 \mid r_x \cdot t + b_x < R_y \cdot (t - T_y)\} \\ &= \inf\{t > 0 \mid b_i + R_y T_y < (R_y - r_x) \cdot t\} \\ &= \inf\{t > 0 \mid \frac{b_x + R_y T_y}{R_y - r_x} < t\} \\ &= \frac{b_x + R_y T_y}{R_y - r_x}. \end{aligned}$$

□

4.2.5 Numerical Example

Now that we can calculate the performance bounds, let us consider a concrete example with concrete numerical values. The example was created using our toolbox (see Chapter 6) and can be found under *numerical_example_chapter_4.py*.

Let us first look at the arrival curve. The following token bucket arrival curves are given:

- $\gamma_1(t) = \gamma_{r_1,b_1}(t) = \gamma_{2,3}(t)$,
- $\gamma_2(t) = \gamma_{r_2,b_2}(t) = \gamma_{1,5}(t)$,
- $\gamma_3(t) = \gamma_{r_3,b_3}(t) = \gamma_{\frac{1}{2},10}(t)$.

Let the concave piecewise linear arrival curve be given by:

$$\alpha(t) = \min\{\gamma_1, \gamma_2, \gamma_3\}(t).$$

The following rate-latency service curves are given:

- $\beta_1(t) = \beta_{R_1,T_1}(t) = \beta_{\frac{1}{2},1}(t)$,
- $\beta_2(t) = \beta_{R_2,T_2}(t) = \beta_{\frac{5}{4},4}(t)$,
- $\beta_3(t) = \beta_{R_3,T_3}(t) = \beta_{2,7}(t)$.

Let the convex piecewise linear service curve be given by:

$$\beta(t) = \max\{\beta_1, \beta_2, \beta_3\}(t).$$

If we now apply Theorem 4.4, 4.6 and 4.8, we receive the following performance bounds:

- backlog bound: $v(\alpha, \beta) = 8.5$,
- delay bound: $h(\alpha, \beta) = 7.6$,
- maximum length of backlogged period: maxBP = 16.

The output bound, $\alpha \oslash \beta(t)$ with $t > 0$, is calculated by applying Corollary 4.7.

To illustrate everything even better, let us take a look at Figure 4.6. The performance bounds are shown there respectively.

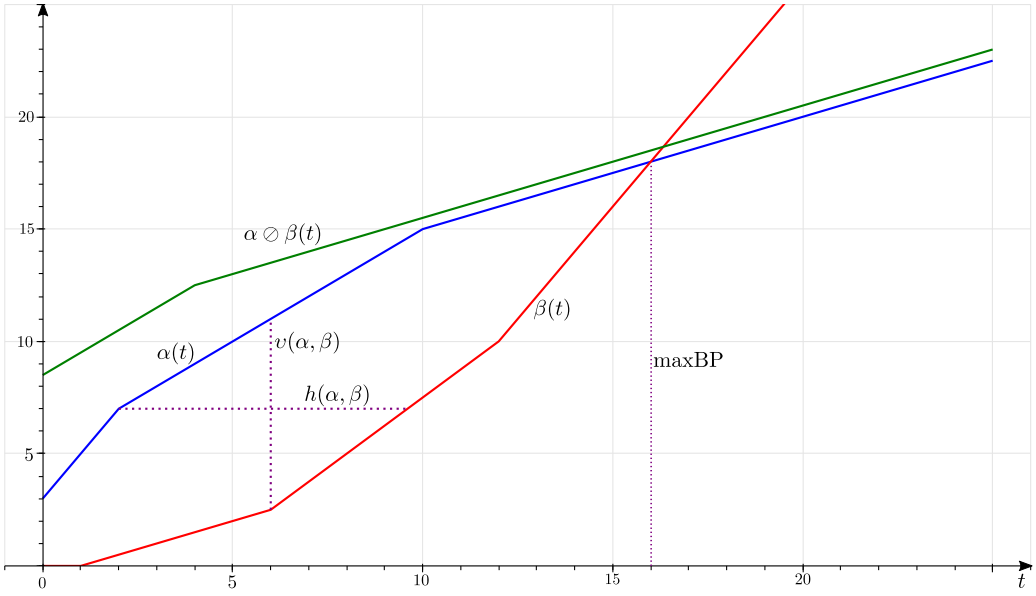


Figure 4.6: Numerical example: performance bounds.

5 FIFO Leftover Service Curve

After extensively delving into the simple scenario of the single flow, single server case and deriving the corresponding performance bounds in the preceding chapters, we now turn our attention to a slightly more complex scenario. In this new chapter, our focus shifts to the FIFO leftover service curve, which becomes relevant when considering two flows and one server, as illustrated in Figure 5.1.

Our objective is to initially gain a basic insight into the behavior of the FIFO leftover service curve. For this purpose, we examine in Section 5.1 a simple case of token bucket arrivals and a rate-latency server. Through the analysis of this scenario, we aim to develop a rough understanding that will help us dealing with more complex scenarios involving concave and convex piecewise linear arrival and service curves (see Section 5.2).

Subsequently, in the course of this chapter, we will delve deeply into this specific scenario. Our primary focus will be on finding an appropriate value for θ, as described in Theorem 2.23. Here, our focus will be specifically on investigating the optimal θ for achieving minimal latency (Subsection 5.2.1), as well as its role in attaining the optimal scenario for both backlog and delay bound (Subsection 5.2.2 and Subsection 5.2.3).

This investigation will enable us to gain important insights into the behavior of the FIFO leftover service curve and lay the groundwork for analyzing more complex network models.

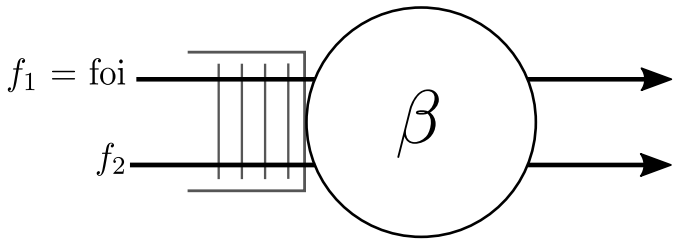


Figure 5.1: Two arrival flows at one node with service curve β.

Remark 5.1. Before we dive into the details of token bucket arrivals and rate-latency service, we need to discuss the stability condition for a system with multiple flows. This requires adjusting the stability condition from Remark 2.33.

We assume having a system $\mathcal{S}$ that multiplexes multiple flows $f_1, \cdots, f_n, n \in \mathbb{N}$, according to FIFO.

Let the arrivals of $f_1, \cdots, f_n$ be constrained by piecewise linear concave function in normal form $\alpha_n(t) = \min\{\gamma_i\}(t)$, $i \in [1, m]$, with $\gamma_i(t) = \gamma_{r_i,b_i}(t) = r_i \cdot t + b_i$ being token bucket arrival curves. Let the rates of the linear sections of α_n be in the set A_n.

Assume that $\mathcal{S}$ guarantees a convex piecewise linear service curve in normal form $\beta(t) = \max\{\beta_j\}(t)$, $j \in [1, o]$, with $\beta_j(t) = \beta_{R_j,T_j}(t) = R_j \cdot [t - T_j]^+$ being rate-latency service curves. Let B be the set of the rates of the linear sections of β.

The stability condition is given by:

$$\exists r_1 \in A_1 \wedge \cdots \wedge \exists r_n \in A_n \wedge \exists R \in B : \sum_{i=1}^{n} r_i < R.$$

We always assume the stability condition to be fulfilled.

5.1 Basic Case: Token Bucket Arrival Curve and Rate-Latency Service Curve

In Theorem 2.23 and Remark 2.24, we have already explored how to calculate the leftover service curve and the interpretation of θ. Now, let us delve into how the choice of θ impacts the leftover service curve, leveraging the results from Lenzini et al. (see [LMS05]).

Consider a system $\mathcal{S}$ that multiplexes two flows, f_1 and f_2, according to the FIFO principle. The arrivals of f_2 are constrained by the piecewise linear function $\alpha_2(t) = \gamma_2(t) = b_2 + r_2 \cdot t$, and the system $\mathcal{S}$ guarantees a rate-latency service curve $\beta(t)$. For flow f_1, the leftover service curve is expressed as [LMS05]:

$$\beta_\theta^1(t) = \begin{cases} \beta_{R-r_2,\left(T+\frac{b_2+r_2(T-\theta)}{R-r_2}\right)}(t), & \text{if } \theta \leq T + \frac{b_2}{R}, \\ \gamma_{R-r_2,R\left(\theta-\left(T+\frac{b_2}{R}\right)\right)}(t-\theta), & \text{otherwise .} \end{cases} \tag{5.1}$$

Equation (5.1) shows that if $\theta \leq T + \frac{b_2}{R}$, we obtain a classic rate-latency service curve. Conversely, if $\theta > T + \frac{b_2}{R}$, the leftover service curve is in a token bucket form. In other words, we have a "service burst", i.e. a jump that occurs in the service curve with a height of $R(\theta - (T + \frac{b_2}{R}))$ at the point $t = \theta$. This behavior of the leftover service curve and this service burst jump can be seen in Figure 5.2.

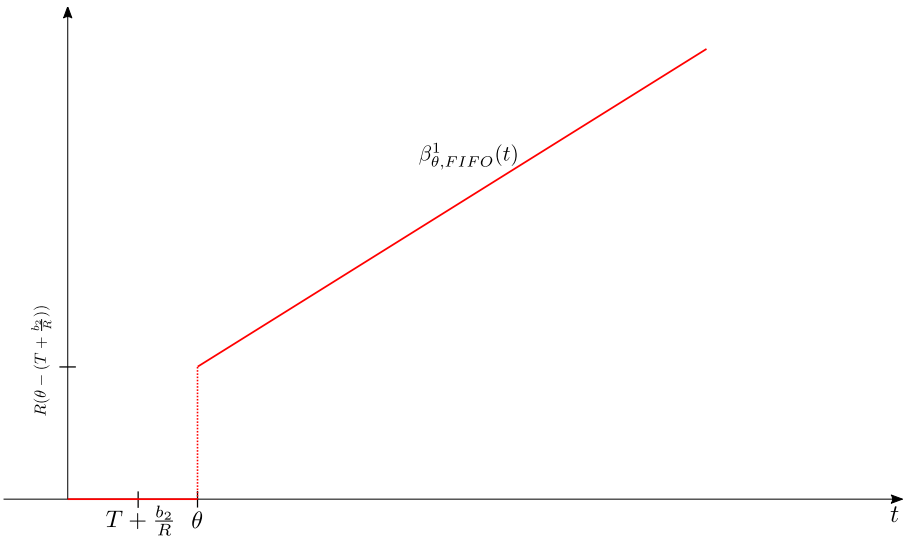


Figure 5.2: FIFO lefover service curve (with jump) for token bucket arrivals und rate-latency service.

Let us delve deeper into the cause of this burst. We will revisit how the FIFO leftover service curve is calculated, focusing on Theorem 2.23 and Equation (2.1):

$$\beta^1_{\theta,FIFO}(t) = \beta^1_\theta(t) = \begin{cases} [\beta(t) - \alpha_2(t-\theta)]^+, & \text{if } t > \theta, \\ 0, & \text{otherwise.} \end{cases}$$

Suppose $\beta(t)$ and $\alpha_2(t-\theta)$ intersect at time $t = x$. Thus, for $t > x$, we have:

$$\beta(t) > \alpha_2(t-\theta).$$

If $\theta > x$, it becomes apparent that the FIFO leftover service for $t \leq \theta$ remains 0. This indicates that while there could already be service available for the flow of interest, because $\beta(t) > \alpha_2(t-\theta)$, we are still waiting for the delay θ caused by the cross flow.

Thus, this "service burst" arises due to the delay introduced by θ, as illustrated by the Traffic Light interpretation in Remark 2.24. This phenomenon underscores the importance of understanding how different values of θ influence the behavior of the FIFO leftover service curve.

5.2 Piecewise Linear Arrival and Service Curve

We assume having a system $\mathcal{S}$ that multiplexes two flows f_1 and f_2 according to FIFO.

Let the arrivals of f_1 be constrained by a piecewise linear concave function in normal form $\alpha_1(t) = \min\{\gamma_i\}(t)$, $i \in [1, m]$, with $\gamma_i(t) = \gamma_{r_i,b_i}(t) = r_i \cdot t + b_i$ being token bucket arrival curves.

Further let the arrivals of f_2 be constrained by a piecewise linear concave function in normal form $\alpha_2(t) = \min\{\gamma_j\}(t)$, $j \in [1, n]$, with $\gamma_j(t) = \gamma_{r_j,b_j}(t) = r_j \cdot t + b_j$ being token bucket arrival curves.

Assume that $\mathcal{S}$ guarantees a convex piecewise linear service curve in normal form $\beta(t) = \max\{\beta_j\}(t)$, $j \in [1, o]$, with $\beta_j(t) = \beta_{R_j,T_j}(t) = R_j \cdot [t - T_j]^+$ being rate-latency service curves.

Let t_i and u_j be the sequence of intersections as defined in Definition 2.27 and Definition 2.28. Let T be the set of all t_i and U the set of all u_j. Further, $I = T \cup U$.

Let d be the delay bound of $\beta(t)$ and $\alpha_2(t)$, where we used either t_{min} or u_{min} for the calculation, given in Equation (4.2). For the cross flow $\alpha_2(t)$, shifted by a given θ, we define $x_{DelayBound}$ as follows:

$$x_{DelayBound} = \begin{cases} u_{min} & \text{if } u_{min} \text{ was used for delay bound calculation,} \\ t^{\theta}_{min} & \text{if } t_{min} \text{ was used for delay bound calculation,} \end{cases} \tag{5.2}$$

with t^{θ}_{min} being the intersection t_{min} shifted by θ.

At What Value of θ Does the Jump Occur?

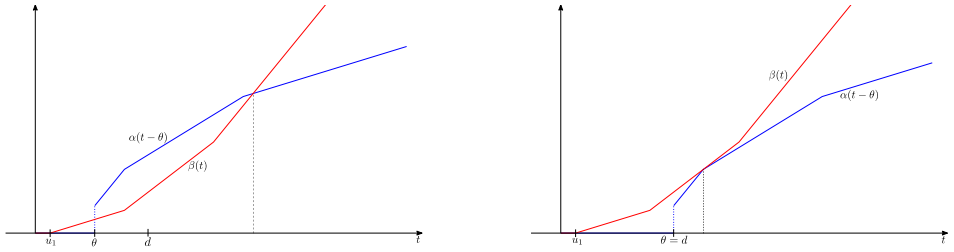
In Section 5.1, we thoroughly examined the behavior of the FIFO leftover service curve under the constraints of token bucket arrivals and rate-latency services. Our objective here is to identify the critical value of θ that triggers a jump in the FIFO leftover service curve.

To pinpoint this critical θ, we analyze the maximum horizontal distance between the service curve $\beta(t)$ and the arrival curve $\alpha(t)$. This distance, referred to as d, can be calculated using Theorem 4.6 as delay bound.

If the arrival curve $\alpha(t - \theta)$ is horizontally shifted by a distance such that $\theta \leq d$, the service curve $\beta(t)$ and the shifted arrival curve $\alpha(t - \theta)$ will intersect at least once, satisfying the condition:

$$\beta(t) = \alpha(t - \theta) > 0.$$

This horizontal shift of the arrival curve is effectively illustrated in Figure 5.3. We can observe that for $\theta \leq d$, there is at least one intersection point between the arrival and service curves, which is marked by the black dotted line.



(a) Arrival Curve shift by $\theta < d$. (b) Arrival Curve shift by $\theta = d$.

Figure 5.3: Arrival curve shift by θ for FIFO leftover service curve without jump.

To be precise, if we shift the arrival curve by $\theta = d$, as shown in Figure 5.3b, then arrival and service obviously intersect at the exact intersection we used to calculate the delay bound. So we know that the last intersection is given by $x_{DelayBound}$, see Equation (5.2).

On the other hand, if $\theta > d$, the curves $\beta(t)$ and $\alpha(t-\theta)$ only intersect at the height 0, specifically:

$$\beta(t) = \alpha(t-\theta) = 0.$$

As depicted in Figure 5.4, for $\theta > d$, there are no intersection points between the arrival and service curves beyond height of 0. The final intersection point is at $(u_1, 0)$.

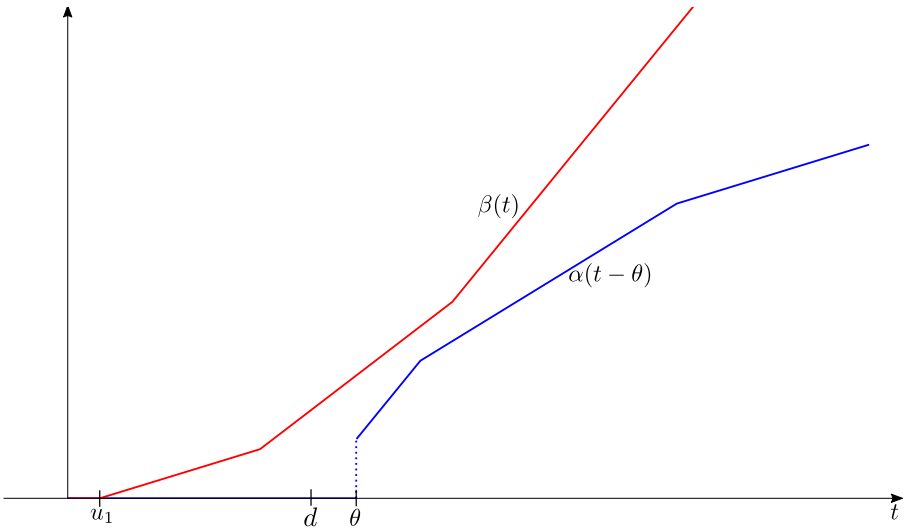


Figure 5.4: Arrival curve shift by θ for FIFO leftover service curve with jump.

Given that $\theta > u_1$, and referring to the insights from Section 5.1, it becomes evident that this results in a jump in the leftover service curve.

To summarize our findings: Given a cross flow $\alpha_2(t)$ and a service curve $\beta(t)$, let d be the maximum horizontal distance between $\alpha(t)$ and $\beta(t)$. For the leftover service curve:

- if $\theta \leq d$, no jump occurs,
- if $\theta > d$, a jump occurs.

In the subsequent part of this section, we will delve into the precise timing t at which this jump manifests.

At What Value of t Does the Jump Occur? How High Is the Jump?

Referring back to the FIFO leftover service curve as outlined in Theorem 2.23, we observe that when the cross flow is shifted by $\theta > d$, no service is provided until $t > \theta$.

Therefore, the jump must occur precisely at this point because for $t \leq \theta$, it holds that

$$\beta^1_{\theta,FIFO} = 0.$$

For $t > \theta$, the following applies:

$$\beta^1_{\theta,FIFO} = \beta(t) - \alpha(t - \theta).$$

Given that, as previously noted, service could have commenced from $t = u_1$, the height of the jump at $t = \theta$ is calculated as:

$$h = \beta(\theta) - t_1. \tag{5.3}$$

This indicates that up to the point $t = \theta$, the FIFO leftover service for the flow of interest could have been provided to the extent of $\beta(\theta)$, minus the burst of the cross flow that needed to be processed first.

Now that we know where the jump in the FIFO leftover service curve theoretically takes place and how high the jump is, we need to consider what the FIFO leftover service curve looks like. Especially for the case in which a jump occurs, we also need to make a case distinction to ensure that we have a leftover service curve that is non-decreasing.

Leftover Service Curve

Case I: $\theta \leq d$

Let us analyze the scenario where the leftover service curve does not exhibit a jump. In the absence of a jump, it is evident that the service curve and the arrival curve, shifted by θ, intersect at least once. Assuming the last intersection point occurs at $t = x$, it follows that for all $t > x$, $\beta(t) > \alpha_2(t)$ holds true. Therefore, the initial latency of the leftover service curve is given by x.

Next, we must determine the structure of the linear segments of the leftover service curve. To achieve this, we traverse the linear segments of the service curve

and subtract the rate(s) of the arrival curve segments within the respective interval of each service curve segment from the service rate of that segment.

Case II: $\theta > d$

In this case we consider the scenario where the leftover service curve exhibits a jump. As previously discussed, after the jump, service is provided immediately, whereas before the jump, it is not.

As mentioned above, the last intersection of the shifted arrival and service curve for $\theta = d$ is at the intersection $x_{DelayBound}$, see Equation (5.2). The t_{min} or u_{min} (see Equation (4.2)) are defined in just the way that at the height of this intersection, the rate of the arrival curve is less than or equal to the rate of the service curve at this height. Therefore, up to this height of $\alpha(t_{min})$ or $\beta(u_{min})$, the arrival curve increases faster than the serivce curve.

However, this means that if the intersection $x_{DelayBound}$ is greater than θ after the shift of the arrival curve by $\theta = d$, i.e.

$$x_{DelayBound} > \theta,$$

it holds that we get a decreasing leftover service curve up to $x_{DelayBound}$.

To calculate the FIFO leftover service curve, θ cannot be defined in such a way that we obtain a decreasing leftover service curve. It therefore holds for θ:

$$\theta \in \mathbb{N}_0 \setminus (d, x_{DelayBound}). \tag{5.4}$$

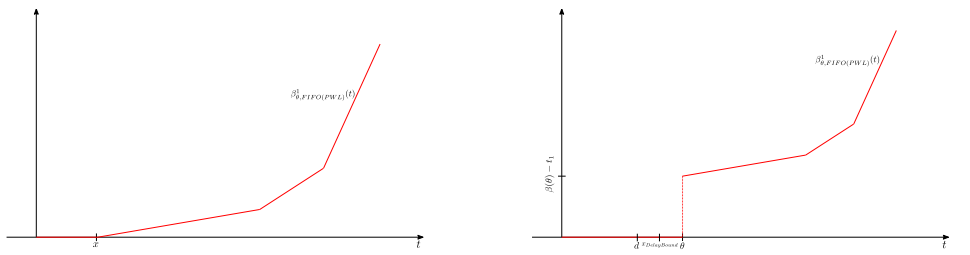
However, if $x_{DelayBound} \leq \theta$ holds, then θ can be chosen without restrictions and it holds for θ:

$$\theta \in \mathbb{N}_0.$$

Since the jump occurs at $t = \theta$, the initial latency of the leftover service curve is equal to θ.

From $t = \theta$ onwards, we can analyze the linear segments of the leftover service curve in the same manner as described in Case I.

An illustration of the FIFO leftover service curve for piecewise linear arrival and service curve $\beta^1_{\theta,FIFO(PWL)}$ is given in Figure 5.5. Case I is given in Figure 5.5a and Case II can be seen in Figure 5.5b.



(a) FIFO lefover service curve (no jump).

(b) FIFO lefover service curve (with jump).

Figure 5.5: FIFO lefover service curve for piecewise linear arrivals and service.

5.2.1 Optimal θ for Minimal Latency

Minimal Latency without Jump

Lemma 5.2. *Let $\alpha(t) = \min\{\gamma_i\}(t)$, $i \in [1, m]$ be a piecewise linear concave function in normal form, with $\gamma_i(t) = \gamma_{r_i,b_i}(t) = r_i \cdot t + b_i$ being token bucket arrival curves.*

Let $\beta(t) = \max\{\beta_j\}(t)$, $j \in [1, o]$ be a piecewise linear convex function in normal form, with $\beta_j(t) = \beta_{R_j,T_j}(t) = R_j \cdot [t - T_j]^+$ being rate-latency service curves.

If we shift the arrival curve by s to the right, so we have $\alpha(t - s)$ for $s \in \mathbb{R}^+$, then the intersection p_s of $\alpha(t - s)$ and $\beta(t)$ is decreasing in s.

Proof. It holds that $\alpha(t) \in \mathcal{F}$. So for a fixed t it holds that $\alpha(t - s)$ with $s \in \mathbb{R}^+$ is decreasing for increasing s. Due to the face that $\alpha(t - s)$ is decreasing for increasing s the intersection p_s of $\alpha(t - s)$ and $\beta(t)$ has to at an earlier time t. So p_s is decreasing for increasing s. □

The result of Lemma 5.2 is also illustrated in Figure 5.3.

Theorem 5.3. *Let $\alpha(t) = \min\{\gamma_i\}(t)$, $i \in [1, m]$ be a piecewise linear concave function in normal form, with $\gamma_i(t) = \gamma_{r_i,b_i}(t) = r_i \cdot t + b_i$ being token bucket arrival curves. Let $\alpha(t)$ be a cross flow.*

Let $\beta(t) = \max\{\beta_j\}(t)$, $j \in [1, o]$ be a piecewise linear convex function in normal form, with $\beta_j(t) = \beta_{R_j,T_j}(t) = R_j \cdot [t - T_j]^+$ being rate-latency service curves.

Let d be the delay bound of $\alpha(t)$ and $\beta(t)$ according to Theorem 4.6.

The optimal θ for the FIFO leftover service curve that minimizes the latency, without having a jump, is given by:

$$\theta_{min\_latency} = d.$$

Proof. In this case, where we want to avoid a jump, we know from Section 5.2 that, in terms of optimizing the minimal latency, the values of θ must satisfy:

$$0 \leq \theta \leq d. \tag{5.5}$$

Additionally, since we do not have a jump, the latency of the leftover service curve must correspond to the point of the last intersection between $\beta(t)$ and $\alpha_2(t - \theta)$ (as opposed to optimizing minimal latency with a jump).

Let $t = x$ be the point of the last intersection between $\beta(t)$ and $\alpha_2(t - \theta)$. According to Lemma 5.2, the time t of the last intersection decreases as θ increases. Consequently, the smallest value of t for the last intersection occurs at the maximum possible value of θ. The smallest value for t is, per definition, given by $x_{DelayBound}$ (see Equation (5.2)), which is also the minimal latency.

The maximum value for θ is d (according to Equation (5.5)), thus we achieve the minimal latency of the leftover service curve, without a jump, for $\theta = d$. □

Minimal Latency with Jump

Theorem 5.4. *Let $\alpha(t) = \min\{\gamma_i\}(t)$, $i \in [1, m]$ be a piecewise linear concave function in normal form, with $\gamma_i(t) = \gamma_{r_i,b_i}(t) = r_i \cdot t + b_i$ being token bucket arrival curves. Let $\alpha(t)$ be a cross flow.*

Let $\beta(t) = \max\{\beta_j\}(t)$, $j \in [1, o]$ be a piecewise linear convex function in normal form, with $\beta_j(t) = \beta_{R_j,T_j}(t) = R_j \cdot [t - T_j]^+$ being rate-latency service curves.

Let d be the delay bound of $\alpha(t)$ and $\beta(t)$ according to Theorem 4.6 and let $x_{DelayBound}$ be defined as in Equation (5.2).

The optimal θ for the FIFO leftover service curve that minimizes the latency, with having a jump, is given by:

$$\theta_{min\_latency} = \begin{cases} d + \epsilon, & \text{if } x_{DelayBound} \leq d, \\ x_{DelayBound}, & \text{otherwise.} \end{cases}$$

with $0 < \epsilon \ll 1$.

Proof. In this case, where we want to induce a jump, we know from Section 5.2 that, for the purpose of optimizing minimal latency, the values of θ must satisfy:

$$\theta > d.$$

Case I: $x_{DelayBound} \leq d$

Additionally, given that a jump is present, we know that $\beta(t)$ and $\alpha_2(t - \theta)$ do not intersect at any point (> 0). Therefore, the latency of the leftover service curve is precisely θ, as we wait for the duration of θ for the delay of the cross flows to pass.

To minimize the latency, we need to choose the smallest possible θ. Since $\theta > d$, we add a minimally small ϵ, where $0 < \epsilon \ll 1$.

Thus, we achieve the minimal latency of the leftover service curve, with a jump, for $\theta = d + \epsilon$.

Case II: $x_{DelayBound} > d$

If $x_{DelayBound} \geq d$ holds, we cannot of course choose the same θ analogous to Case I, since $\theta \notin (d, x_{DelayBound})$ applies. We have to consider the edge of the excluded interval for the optimal θ.

In this case, we can choose both $\theta = d$ or $\theta = x_{DelayBound}$ for θ with regard to the latency and we obtain a latency of $x_{DelayBound}$ in each case.

For $\theta = d$ we proceed in the same way as in Theorem 5.3 and obtain a latency of $x_{DelayBound}$ but no jump in the FIFO leftover service curve. However, since we want a jump, we choose $\theta = x_{DelayBound}$ and get a jump at the point $t = x_{DelayBound}$ and also a latency of $x_{DelayBound}$. □

5.2.2 Optimal θ for Backlog Bound

Having determined the optimal choice of θ for minimizing the latency of the leftover service curve, we now examine how th choice of θ affects the backlog bound of the leftover service curve and $\alpha_1(t)$.

First, we revisit the calculation of the backlog bound for piecewise linear arrival and service curves as outlined in Theorem 4.4.

We define a as follows:

$$a = \min\{x \in I : (\alpha_1(x) = \gamma_y(x) \wedge \beta^1_{\theta,FIFO(PWL)}(x) = \beta_z(x)) \wedge r_y \leq R_z\},$$

as the first intersection for which the rate of the arrival curve is less or equal to the rate of the service curve.

It is evident that we aim to minimize a to achieve a smaller backlog bound. Up to $t = a$, the vertical deviation between the arrival and service curves increases, so a smaller a results in a smaller backlog bound.

Thus, we can argue that the θ minimizing the latency of the leftover service curve also yields an a that is at least as good as any a for larger values of θ. This is because the rates of the linear sections of the leftover service curve are derived by subtracting the rates of the linear sections of the cross flows from the rates of the original service curve at the intersections. By minimizing the latency, we ensure that service is provided as early as possible, thus obtaining the minimal a that minimizes the backlog bound. In summary, by offering service as early as possible, we ensure the smallest a, preventing any artificial enlargement due to excessive delays (from a too-large θ).

5.2.3 Optimal θ for Delay Bound

Next, we determine the optimal choice of θ for minimizing the delay bound for our flow of interest.

We begin with the results from Lenzini et al. (see [LMS05]). Consider two token bucket arrivals $\gamma_1 = \gamma_{r_1,b_1}$ and $\gamma_2 = \gamma_{r_2,b_2}$, where γ_1 represents our foi. These arrivals receive a rate-latency service $\beta_{R,T}$ under FIFO multiplexing.

The tightest delay bound for our foi is given by:

$$\theta = T + \frac{b_1 + b_2}{R}. \tag{5.6}$$

For this θ, we obtain a leftover service curve with a jump height of $R \cdot (\theta - T) - b_2$

at $t = T + \frac{b_1+b_2}{R}$. This jump height is precisely b_1, as shown:

$$\begin{aligned} R \cdot (\theta - T) - b_2 &\overset{(Eq.(5.6))}{=} R \cdot (T + \frac{b_1 + b_2}{R} - T) - b_2 \\ &= R \cdot (\frac{b_1 + b_2}{R}) - b_2 \\ &= b_1 + b_2 - b_2 \\ &= b_1. \end{aligned}$$

Thus, we choose θ to ensure a jump in the leftover service curve at the height where we calculate the delay bound (in the token bucket and rate-latency case, always from the initial burst height). [LMS05]

We now apply this understanding to the piecewise linear arrival and service curve scenario. Analogous to the token bucket and rate-latency case, we also want to jump to the height of the initial burst b_1 of the arrival curve to get the tightest delay bound. So, let us take a look at why we jump to this height if possible. The delay bound is defined as the largest horizontal distance between arrival and service curve. If there is a jump in the FIFO leftover service that is higher than b_1, then we trivially get a horizontal distance at the point t_1 of infinity, i.e. $h(\alpha, \beta) = \infty$. This is unlike the tightest possible delay bound we can get, so we know for now that we do not want a jump higher than the initial burst.

A lower jump, down to no jump at all, provides us with a larger delay bound, since with a lower jump we reduce the service burst resulting from the undivided rate of the server, i.e. we reach the height b_1 earlier compared to a lower jump or no jump at all.

In other words, we choose the θ for the tightest delay bound as follows:

$$\theta_{opt} = \{\theta > 0 : t_1 = \beta(\theta) - t_1\}.$$

However, if $x_{DelayBound} > \theta$ applies to our optimal θ_{opt} and $\theta_{opt} \in (d, x_{DelayBound})$, then of course we are unable to use it (see Equation (5.4)). But we also know that we cannot choose the θ_{opt} larger, as we would make the jump height greater than b_1 and thus obtain a delay bound of infinity. Accordingly, the optimal θ_{opt} would have to be chosen at the lower limit of the excluded interval $(d, x_{DelayBound})$. We would therefore choose $\theta_{opt} = d$ and, although we would no longer obtain a jump for our FIFO leftover service curve, we would still obtain the tightest possible delay bound for these conditions.

6 Toolbox: Deterministic Network Calculus with Piecewise Linear Curves

As part of this master thesis, we developed a specialized toolbox that allows for the illustrative representation and verification of theoretical results in deterministic network calculus using real-world examples. The toolbox, written in the Python programming language [VRD09], utilizes the "bokeh" library for creating plots [Bok18].

The name of the toolbox is: "Toolbox: Deterministic Network Calculus With Piecewise Linear Curves" (toolbox:dnc-pwl) and can be found for usage here [Wil24].

The motivation behind developing this toolbox was to make the complex theoretical concepts of deterministic network calculus more tangible and understandable. By visually representing the results, we could not only verify the correctness of our theoretical derivations but also develop an intuitive understanding of the underlying theory. Additionally, we integrated brute-force approaches into the toolbox, which supported us in generating theoretical results. These approaches served as reference points to validate the theoretical models, ensuring that our derivations were not only theoretically sound but also practically applicable.

In this chapter, we will take a brief look at the code structure of the toolbox to demonstrate its easy and intuitive usage (Section 6.1). We will start with an overview of the key modules and functions that form the foundation of the toolbox. Subsequently, we will explain how the individual components work together.

At the end of this chapter, in Section 6.2, we will introduce an interactive plot that greatly simplifies the understanding of deconvolution for piecewise linear arrival and service curves. This interactive plot has proven to be extremely useful as it provides a dynamic way to comprehend this more complex mathematical process.

6.1 Code Structure

Before we briefly address the performance bounds and FIFO leftover service curve, we first need to discuss our foundational building blocks: arrival and service curves.

6.1.1 Arrival and Service Curves

The structure is straightforward and easy to understand. We begin with the abstract class *ArrivalCurve*, which contains several abstract methods. These methods are then implemented by the concrete classes *TokenBucketArrivalCurve* and *PiecewiseLinearArrivalCurve*. The abstract methods cover all the essential requirements for an arrival curve necessary for further calculations.
For instance, there might be a method to calculate the value of the arrival curve at a specific point in time t (*calculate_function_value(t)*) or a method to return the initial burst (*get_initial_burst()*). This design allows us to implement performance bounds or the FIFO leftover service curve for general arrival curves without needing to differentiate whether it is a token bucket or piecewise linear arrival curve.

We took a similar approach with service curves. Here, we also have an abstract class *ServiceCurve* with several abstract methods, which are then implemented by the concrete classes *RateLatencyServiceCurve* and *PiecewiseLinearServiceCurve*.

6.1.2 Performance Bounds and FIFO Leftover Service Curve

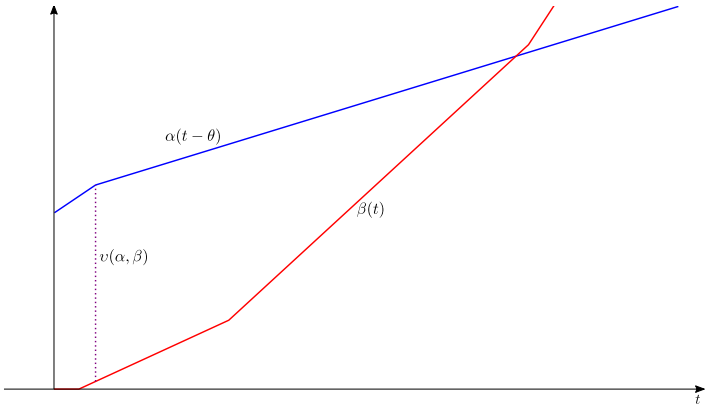
The performance bounds, FIFO leftover service curve, and the necessary min-plus algebra operations were each implemented in their own methods. These methods can be used seamlessly for different concrete classes of the abstract classes *ArrivalCurve* and *ServiceCurve*, as mentioned above. These methods are located under *dnc_operations* and *dnc_leftover_service*.

6.1.3 Solution Checker

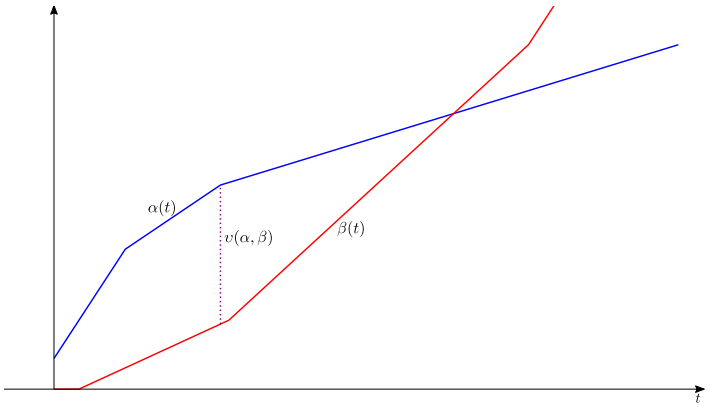
The aforementioned brute force implementations of the methods for deconvolution and convolution for piecewise linear arrival and service curves can be found under *solution_checker*.

6.2 Interactive Plot

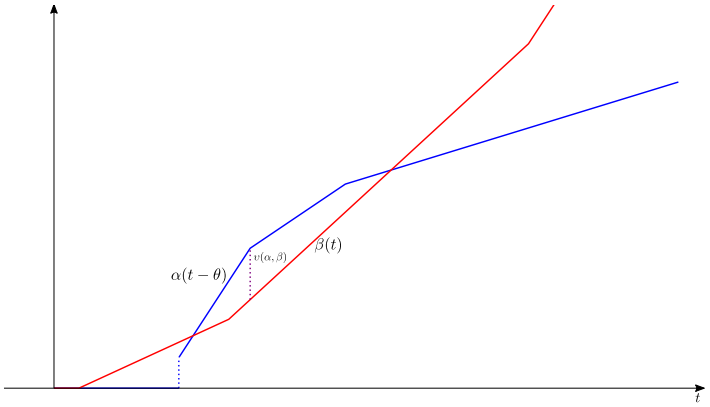
Finally, we come to an interactive plot that we used to understand and illustrate the deconvolution of piecewise linear arrival and service curves (see Lemma 4.3). This plot is particularly useful as it shows the intersection needed for calculating the backlog bounds $v(\alpha, \beta)$, which derives $a(t)$ from Equation (4.1). This plot is displayed in Figure 6.1. The arrival curve $\alpha(t - \theta)$, which can be shifted by θ using a slider, is shown in blue. The service curve is depicted in red, and the backlog bound $v(\alpha, \beta)$ is shown in purple. The subplots display different values of θ, thus showing the arrival curve shifted accordingly. In Figure 6.1a we see $\theta = 5$, in Figure 6.1b we have $\theta = 0$ and finally, in Figure 6.1c, $\theta = -5$.



(a) Arrival curve shifted to the left by 5.



(b) Arrival cruve not shifted.



(c) Arrival curve shifted to the right by 5.

Figure 6.1: Example of interactive plot with 3 different values of the shifting parameter θ.

7 Finite Shared Buffers

In all previous analyses of systems, such as those illustrated in Figure 3.4 and Figure 5.1 we have consistently assumed the presence of sufficiently large buffers that could accommodate the traffic we were examining. However, in practical scenarios, this assumption often does not hold, as systems may not always possess buffers of adequate size at all times. To address the risk of buffer overflow, a mechanism known as "flow control", which is employed in the transmission control protocol (TCP), is utilized [Pos81]. Within the network calculus framework, one approach to model this mechanism is through the window flow control concept, initially introduced by Agrawal et al. [ACOR98].

In this chapter, our focus shifts to the examination of finite shared buffers. A finite shared buffer system (see Figure 7.2) is characterized by multiple flows sharing a buffer of limited size. Specifically, we aim to expand upon the subsection of [HCS24] on finite shared buffers by incorporating the use of rate-latency service curves (Section 7.2). To set the stage, in Section 7.1 we will first give a very brief overview of the results of Hamscher et al. that we will use. Finally, we will present a numerical example that applies our new findings with rate-latency service curves (Section 7.3).

7.1 Theoretical Background

The report by Hamscher et al. [HCS24] delves into the extension of the Network Calculus framework through the introduction of minimal arrival curves. This innovative approach ultimately allowed them to derive tight performance bounds for systems with negative service curves. In this section, we will provide a brief overview of the key results from their work that are essential for understanding and contextualizing the subsequent sections of this chapter.

Definition 7.1. (Sub-additive Closure [LBT01]). Let $f \in \mathcal{F}$. The sub-additive closure of f is defined as:

$$f^* := \inf_{n \geq 0} \left\{ f^{(n)} \right\},$$

with $f^{(n)}$ being the n-fold self convolution of f. That is, $f^{(0)} = \delta_0$, $f^{(1)} = f$ and for $n \geq 2$ $f^{(n)} = \bigotimes_{i=0}^{n} f^{(i)}$.

Definition 7.2. (Maximal and Minimal Arrival Curve [CKT03]). Let $\overline{\alpha}, \underline{\alpha} \in \mathcal{F}_0$. For an arrival process A the function $\overline{\alpha}$ is called maximal arrival curve and $\underline{\alpha}$ is called minimal arrival curve, if for all $0 \leq s \leq t$ holds that

$$\underline{\alpha}(t-s) \leq A(t) - A(s) \leq \overline{\alpha}(t-s).$$

Remark 7.3. The maximal arrival curve $\overline{\alpha}$ is equal to the arrival curve α, as defined in Definition 2.11. One function used for minimum arrival curves is the rate-latency curve $\beta_{R,T}$ (see Example 2.16).

In [HCS24] the authors present a negative service curve, which they use for their further results. To do this, they replace the original service curve "safely" by

$$\xi = \beta_{\downarrow} := \beta \overline{\oslash} 0,$$

with $\beta \overline{\oslash} 0 = \inf_{s \geq 0}\{\beta(t+s) - 0\}$ being the (max,plus) deconvolution. [BBLC18] Since $\beta_{\downarrow}$ is the largest non-decreasing function for which $\beta_{\downarrow} \leq \beta$ holds, it is called "lower non-decreasing closure". [HCS24]

In Figure 7.1 an example of $\xi_{n,R,T}$ can be seen.

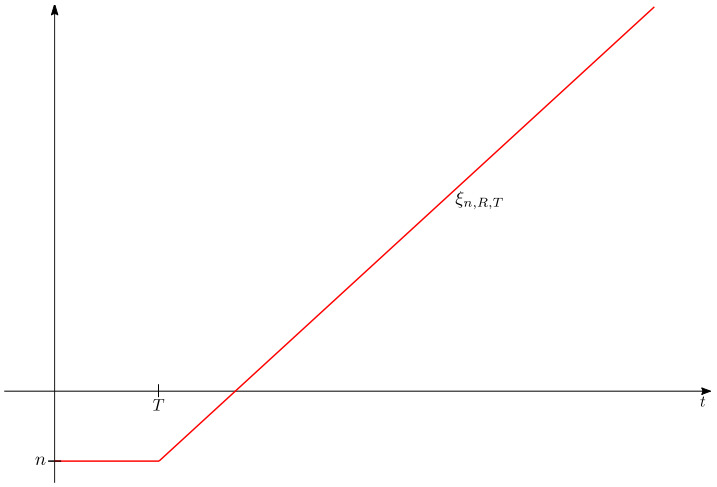


Figure 7.1: Example of $\xi_{n,R,T}$.

Theorem 7.4. (Generalized Delay Bound [HCS24]). *Let an arrival process A traverse a system $\mathcal{S}$. Further, let the arrivals be constrained by maximal arrival curve $\overline{\alpha} \in \mathcal{F}_0$ and minimal arrival curve $\underline{\alpha} \in \mathcal{F}_0$, and let the system offer a service curve ξ. The virtual delay $d(t)$ satisfies for all $t \geq 0$*

$$d(t) \leq z(\underline{\alpha}, \xi) \vee h(\overline{\alpha}, \xi),$$

with $z(\underline{\alpha}, \xi) := \inf\{\tau \geq 0 \mid \underline{\alpha} \otimes \xi(\tau) \geq 0\}$.

7.2 Finite Shared Buffers with Rate-Latency Service Curves

This Section focuses on finite shared buffers, in particular the system shown in Figure 7.2. This system equals the one utilized in the study in [HCS24]. Within this context, we assume that β_1 and β_2 are rate-latency service curves. Our objective is to derive performance bounds for this system. To achieve this, we will employ both the conventional analysis method and the approach introduced in [HCS24]. By comparing these methodologies, we aim to elucidate the advantages and potential improvements offered by the new method.

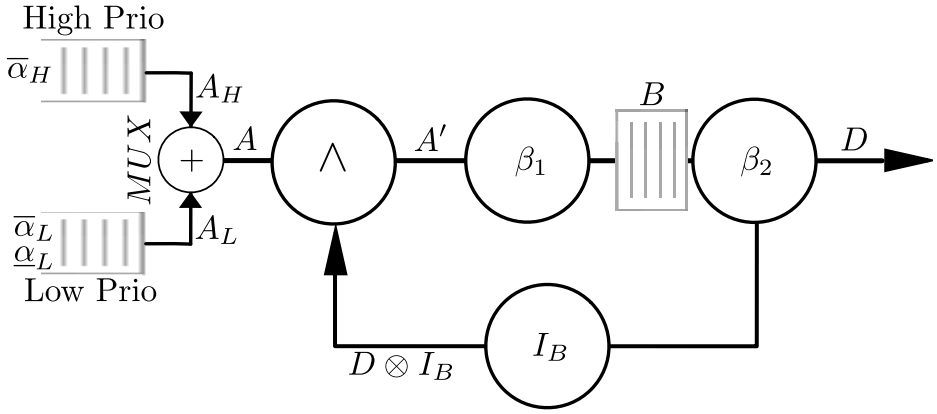


Figure 7.2: Finite shared buffer system. [HCS24]

Let us examine the system depicted in Figure 7.2. This system features two priority queues: one designated for a high-priority flow and the other for a low-priority flow. Each flow is upper-bounded by token bucket arrival curves, denoted as $\overline{\alpha}_H = \gamma_{r_H,b_H}$ for the high-priority flow and $\overline{\alpha}_L = \gamma_{r_L,b_L}$ for the low-priority flow.

Let I_B be the service curve of the feedback control for arrivals that exceed the finite buffer capacity B at β_2. For $t \geq 0$, $I_B(t)$ is defined as follows (as in [ACOR99] and [BBLC18]):

$$I_B(t) = \begin{cases} +\infty, & \text{if } t > 0, \\ B, & \text{otherwise.} \end{cases}$$

It applies that

$$D \otimes I_B(t) = D(t) + B.$$

Further, let $\beta_1 = \beta_{R_1,T_1}$ and $\beta_2 = \beta_{R_2,T_2}$ and it holds that $R = R_1 \wedge R_2$ and $T = T_1 + T_2$.

For the closed-loop feedback system holds that

$$A' \geq A \wedge (D \otimes I_B), D \geq A' \otimes \beta_1 \otimes \beta_2,$$

with $A = A_H + A_L$. This leads to

$$D \geq A \otimes (\beta_1 \otimes \beta_2) \wedge D \otimes (I_B \otimes \beta_1 \otimes \beta_2),$$

and can be transformed into an open-loop system [Cha12]

$$D \geq A \otimes (\beta_1 \otimes \beta_2) \otimes (I_B \otimes \beta_1 \otimes \beta_2)^*,$$

with $(I_B \otimes \beta_1 \otimes \beta_2)^*$ being the sub-additive closure as defined in Definition 7.1.

According to [HCS24] we know if $RT > B$ holds, the system offers a service curve

$$\beta^{FB} = (\beta_1 \otimes \beta_2) \otimes (I_B \otimes \beta_1 \otimes \beta_2)^*. \tag{7.1}$$

From [LBT01, p. 119] we know that if $RT > B$ applies, $(I_B \otimes \beta_1 \otimes \beta_2)^*$ is a staircase function. Therefore β^{FB} is also a staircase function and is illustrated as the red curve in Figure 7.3.

For the case that $RT \leq B$ holds the service curve equals $\beta^{FB} = \beta_1 \otimes \beta_2 = \beta_{R,T}$.

Let us now consider these two cases. We start with the case $RT > B$ and afterwards we take a look at the $RT \leq B$ case.

Case I: $RT > B$

Let us first consider the service curve used for this case, given in Equation (7.1).

As previously mentioned, we know that β^{FB} is a staircase function as shown in Figure 7.3. We know from Bouillard et al. in [BBLC18, p.41] that there is no easy-to-use analytic expression for it yet. Despite this limitation, we aim to examine this case by utilizing a lower bound approximation of the staircase function. However, before proceeding with this approach, it is essential to take a look at what the staircase function looks like.

Since it holds that $\beta_1 \otimes \beta_2 = \beta_{R,T}$ (see [BBLC18]), we can rewrite Equation (7.1) and we get $\beta^{FB} = \beta_{R,T} \otimes (I_B \otimes \beta_{R,T})^*$.

First, let us focus on the sub-additive closure $(I_B \otimes \beta_{R,T})^*$.

For that, the convolution of $I_B \otimes \beta_{R,T}$ by itself can be calculated as follows:

$$\begin{aligned}(I_B \otimes \beta_{R,T}) \otimes (I_B \otimes \beta_{R,T}) &= (B + \beta_{R,T}) \otimes (B + \beta_{R,T}) \\ &= 2B + (\beta_{R,T} \otimes \beta_{R,T}) \\ &= 2B + \beta_{R,2T},\end{aligned}$$

where we used that $I_B \otimes \beta_{R,T} = \inf_{0 \leq s \leq t}\{I_B(t-s) + R \cdot [s-T]^+\} \overset{(s=t)}{=} B + \beta_{R,T}$.

The sub-additive closure is given by

$$\begin{aligned}(I_B \otimes \beta_{R,T})^* &= \inf_{n \geq 0}\{(I_B \otimes \beta_{R,T})^{(n)}\} \\ &= \delta_0 \wedge \inf_{n \geq 1}\{nB + \beta_{R,nT}\}.\end{aligned}$$

For the service curve β^{FB} we get:

$$\begin{aligned}
\beta^{FB}(t) &= \beta_{R,T} \otimes (I_B \otimes \beta_{R,T})^*(t) \\
&= \inf_{0 \le s \le t} \{(I_B \otimes \beta_{R,T})^*(t-s) + \beta_{R,T}(s)\} \\
&= \inf_{0 \le s \le t} \{\inf_{n \ge 1} \{nB + \beta_{R,nT}(t-s)\} + \beta_{R,T}(s)\} \wedge \{\beta_{R,T}(t)\} \\
&= \inf_{n \ge 1} \{\, nB + \inf_{0 \le s \le t} \{\beta_{R,nT}(t-s) + \beta_{R,T}(s)\}\} \wedge \{\beta_{R,T}(t)\} \\
&= \inf_{n \ge 1} \{\, nB + \beta_{R,(n+1)T}(t)\} \wedge \{\beta_{R,T}(t)\} \\
&= \inf_{n \ge 0} \{\, nB + \beta_{R,(n+1)T}(t)\}. \qquad (7.1)
\end{aligned}$$

So according to [LBT01, p. 119], the service curve β^{FB} looks like the red curve in Figure 7.3.

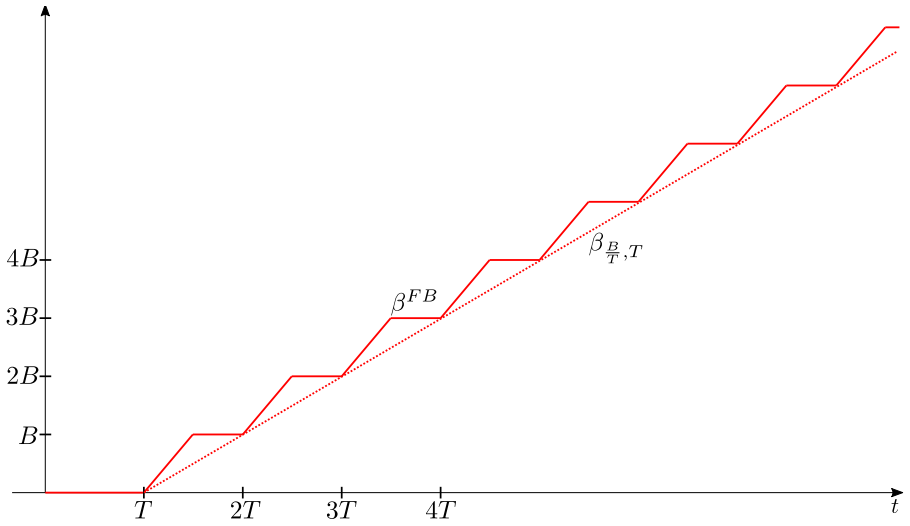


Figure 7.3: Lower bound of the staircase function.

In order analyze this case, we lower bound the staircase function β^{FB} by $\beta^{FB} \ge \beta_{\frac{B}{T},T}$ (see the red dotted curve in Figure 7.3).

Now that we have obtained the service curve β^{FB}, we can begin deriving the performance bounds in a manner analogous to the work presented in [HCS24]. Specifically, in addition to employing the conventional analysis (*ca*), we will also explore the newly introduced method using the minimal arrival curve (*mac*).

Let us define $v_H := \gamma_{r_H,b_H}(T)$ and $v_L := \gamma_{r_L,b_L}(T)$.

At first, we take a look at the buffer requirement for the high priority queue, which is the same for both analysis types (*ca* and *mac*):

$$v(\overline{\alpha}_H, \beta^{FB}) = v(\gamma_{r_H,b_H}, \beta_{\frac{B}{T},T}) = \gamma_{r_H,b_H}(T) = v_H$$

For the buffer size required by the low priority flow, we proceed differently for the respective analysis methods. Although we first have to calculate the residual service curve for both, we can simply do this as follows for the *mac* analysis:

$$\beta_{res}^{mac} = (\beta^{FB} - \overline{\alpha}_H)_{\downarrow} = (\beta_{\frac{B}{T},T} - \overline{\alpha}_H)_{\downarrow} = \xi_{v_H,\frac{B}{T}-r_H,T}.$$

For the required buffer size, we therefore obtain

$$v(\overline{\alpha}_L, \beta_{res}^{mac}) = v_H + v_L.$$

The residual service curve of the *ca* is calculated by

$$\begin{aligned}\beta_{res}^{ca} &= [\beta^{FB} - \overline{\alpha}_H]^+ \otimes ([\beta^{FB} - \overline{\alpha}_H]^+ \otimes I_{B-v(\overline{\alpha}_H,\beta^{FB})})^* \\ &= \beta_{\frac{B}{T}-r_H,\frac{v_H}{\frac{B}{T}-r_H}} \otimes (\beta_{\frac{B}{T}-r_H,\frac{v_H}{\frac{B}{T}-r_H}} \otimes I_{B-v_H})^*,\end{aligned}$$

where we used that $[\beta^{FB} - \overline{\alpha}_H]^+ = \beta_{\frac{B}{T}-r_H,\frac{v_H}{\frac{B}{T}-r_H}}$. [HCS24]

In contrast to the *mac* analysis, we need to consider the relationship between the bandwidth-delay product of the residual feedback system and the buffer size available for the low priority flow. The bandwidth delay product is given by

$$R^{res}T^{res} = (\frac{B}{T} - r_H)\frac{v_H}{\frac{B}{T} - r_H} = v_H$$

and the buffer size is given by $B^{res} = B - v_H$.

For the relation of $R^{res}T^{res}$ and B^{res} we have two cases.

Case A: $R^{res}T^{res} \leq B^{res}$ $(v_H \leq B - v_H)$

In this case it holds that $\beta_{res}^{ca} = \beta_{\frac{B}{T}-r_H,\frac{v_H}{\frac{B}{T}-r_H}}$ (see [LBT01, p. 119]). For the required buffer size we receive the following [HCS24]:

$$v(\overline{\alpha}_L, \beta_{res}^{ca}) = v(\gamma_{r_L,b_L}, \beta_{\frac{B}{T}-r_H,\frac{v_H}{\frac{B}{T}-r_H}}) = b_L + r_L \cdot \frac{v_H}{\frac{B}{T} - r_H}.$$

Case B: $R^{res}T^{res} > B^{res}$ $(v_H > B - v_H)$

For this case, we are not able to calculate a backlog bound using conventional analysis (at least not for arbitrary high and low priority flows). The argumentation for this can be made in the same way as in [HCS24], where further studies on this case are presented.

After we have analyzed the backlog bound for the different approaches (*ca* and *mac*), we now consider to the delay bound. For the high priority flow both analyses provide the same result, since the residual curves are equal. The delay bound is given by

$$h(\overline{\alpha}_H, \beta^{FB}) = h(\gamma_{r_H,b_H}, \beta_{\frac{B}{T},T}) = \frac{b_H}{\frac{B}{T}} + T.$$

Due to the fact that the residual curves are not equal for the low priority flow, we differentiate between the two approaches and first consider the *mac* analysis. For this, we have to use the minimum arrival curve, given by $\underline{\alpha}_L = \beta_{r_L,T_{\underline{\alpha}_L}}$ and we obtain the delay bound [HCS24]:

$$\begin{aligned} d_{e2e}^{mac} &= h(\overline{\alpha}_L, \xi_{v_H,\frac{B}{T}-r_H,T}) \vee z(\underline{\alpha}_L, \xi_{v_H,\frac{B}{T}-r_H,T}) \\ &= \left(\frac{v_H + v_L}{\frac{B}{T} - r_H}\right) \vee \left(T_{\underline{\alpha}_L} + \frac{v_H}{\underline{r}_L}\right). \end{aligned}$$

In order to use the *ca*, we first have to define the number of stairs (for the case $R^{res}T^{res} > B^{res}$) that we need for the delay bound calculation. It holds that the number of stairs is given by $i^* := \left\lceil \frac{v_L}{B-v_H} \right\rceil$. [HCS24]

The delay bound is given by [HCS24]

$$d_{e2e}^{ca} = \begin{cases} \frac{v_H+v_L}{\frac{B}{T}-r_H}, & \text{if } R^{res}T^{res} \leq B^{res}, \\ \frac{v_L-(i^*-1)(B-v_H)}{\frac{B}{T}-r_H} + i^*T^{res}, & \text{otherwise.} \end{cases} \tag{7.2}$$

Case II: $RT \leq B$

We can treat this case in exactly the same way as Case I. Although we do not obtain a staircase function as in Figure 7.3, we directly have a rate-latency service curve, just as we have assumed one as the lower bound of the staircase function. Accordingly, we only replace the rate $\frac{B}{T}$ by the rate R and get the results for this case.

7.3 Numerical Example

In this section, we want to provide a numerical example for the two approaches from Section 7.2. The example was created using our toolbox (see Chapter 6) and can be found under *numerical_example_chapter_7.py*.

But first we have to define all the necessary parameters for the system shown in Figure 7.2.

Let $\overline{\alpha}_H = \gamma_{r_H,b_H}$, with $r_H = 2.5\frac{Mbit}{s}$ and $b_H = 1\ Mbit$. The minimal and maximal arrival curve for the low priority flow is given by $\overline{\alpha}_H = \gamma_{r_L,b_L}$ and $\underline{\alpha}_L = \beta_{\underline{r}_L,T_{\underline{\alpha}_L}}$, with $r_L = 2.5\frac{Mbit}{s}, b_L = 2\ Mbit, \underline{r}_L = 1.0\frac{Mbit}{s}$ and $T_{\underline{\alpha}_L} = \frac{v_L}{\frac{B}{T}}$. Each server offers a service with service curve β_{R_i,T_i} with $R_i = 15\ \frac{Mbit}{s}$ and $T_i = 0.5\ s$.

In Figure 7.4 and Figure 7.5 we present the delay bound for the two different methods as a function of the buffer size.

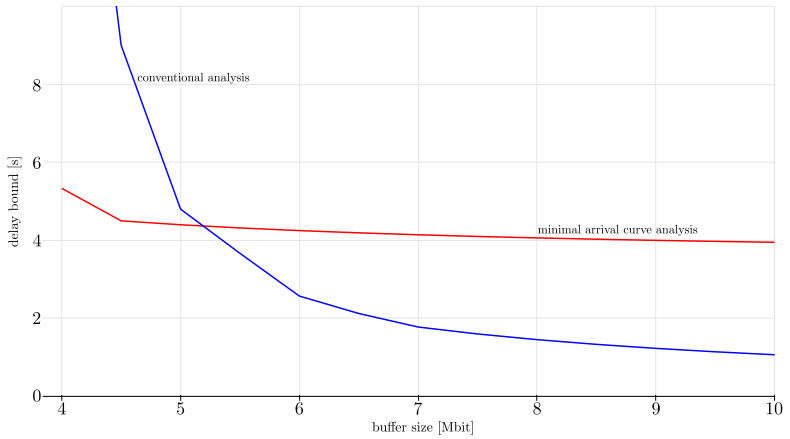


Figure 7.4: Delay bounds for $\underline{r}_L = 1.0\ \frac{Mbit}{s}$.

If $\underline{r}_L$ is changed to $\underline{r}_L = 2.0\ \frac{Mbit}{s}$ we get the following result:

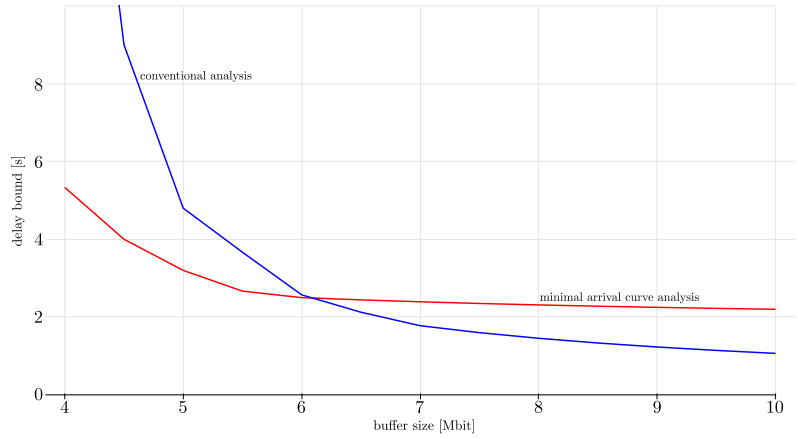


Figure 7.5: Delay bounds for $\underline{r}_L = 2.0\ \frac{Mbit}{s}$.

In Figure 7.4 and 7.5 the red curve represents d_{e2e}^{mac}, which denotes the end-to-end delay obtained using the minimal arrival curve method. The blue curve represents d_{e2e}^{ca}, indicating the end-to-end delay derived from the conventional analysis.

In both figures (for $\underline{r}_L = 1.0\ \frac{Mbit}{s}$ and $\underline{r}_L = 2.0\ \frac{Mbit}{s}$), we observe that the *mac* analysis offers more accurate delay bounds for smaller buffer sizes compared to the *ca* method. However, as the buffer sizes increase, this relationship reverses, and the *ca* method provides more accurate delay bounds. Mathematically, this phenomenon can be explained by considering the shift in cases within Equation (7.2) as the buffer sizes increase. Specifically, for smaller buffer sizes, we fall into the second case of the equation, while for larger buffer sizes, we transition to the first case. This indicates that when the buffer of the residual system is sufficiently large, the *ca* method yields more precise results.

We can also observe that as $\underline{r}_L$ increases, the gap between the delay bounds derived from the *ca* and *mac* analyses narrows. This indicates that the greater the minimum rate of arrivals, the more accurate the delay bounds provided by the *mac* analysis become.

8 Conclusion And Future Work

In this thesis, we have explored the use of piecewise linear concave and convex functions as arrival and service curves in network calculus. We introduced new results for combining concave piecewise linear arrival curves with rate-latency service curves, as well as for using both concave and convex piecewise linear arrival and service curves. We analyzed performance bounds for these curves and developed efficient methods for calculating them. This theoretical framework was then applied to analyze FIFO multiplexed networks, with a particular focus on the selection of the parameter θ. We demonstrated how to choose θ to minimize the latency of the leftover service curve and to achieve the smallest possible backlog or delay bound for our flows of interest. Additionally, we implemented a toolbox to visualize the theoretical results and enhance understanding through numerical examples. Finally, we also extended the work of Hamscher et al. in [HCS24] regarding finite shared buffers to address the scenario including service curves with nonzero latencies.

The results presented in this thesis can be extended in several ways. One area of further research is the FIFO leftover service curve. Our current framework excludes an interval for the choice of θ due to the inability to handle negative or decreasing service curves. However, the work of Hamscher et al. in [HCS24] provides a methodology for dealing with partially negative and decreasing service curves, which could be explored to extend our analysis. Especially since these are new findings that extend the network calculus, this also offers interesting possibilities in this case. Another potential extension involves applying the results of this thesis to other network topologies and settings. For instance, by using the finite shared buffers setting from Chapter 7, we can investigate the behavior of concave piecewise linear arrivals instead of token bucket arrivals. This could yield new insights and broader applicability of our findings. Finally, the visualization tool presented in Chapter 6 offers opportunities for enhancement. Improving the efficiency of the implementations and expanding the tool's capabilities would improve its utility for theoretical research.

Bibliography

[ACOR98] Rajeev Agrawal, RL Cruz, Clayton Okino, and Rajendran Rajan. Performance bounds for flow control protocols. IEEE/ACM Transactions on Networking, preprint available from http://www. ece. wisc. edu/agrawal, 1998.

[ACOR99] Rajeev Agrawal, Rene L Cruz, Clayton Okino, and Rajendran Rajan. Performance bounds for flow control protocols. IEEE/ACM transactions on networking, 7(3):310–323, 1999.

[BBLC18] Anne Bouillard, Marc Boyer, and Euriell Le Corronc. Deterministic network calculus: From theory to practical implementation. John Wiley & Sons, 2018.

[BCOQ92] François Baccelli, Guy Cohen, Geert Jan Olsder, and Jean-Pierre Quadrat. Synchronization and linearity, volume 115. Wiley New York, 1992.

[Bok18] Bokeh Development Team. Bokeh: Python library for interactive visualization, 2018.

[BS12] Anne Bouillard and Giovanni Stea. Exact worst-case delay for fifo-multiplexing tandems. In 6th International ICST Conference on Performance Evaluation Methodologies and Tools, pages 158–167. IEEE, 2012.

[BS14] Anne Bouillard and Giovanni Stea. Exact worst-case delay in fifo-multiplexing feed-forward networks. IEEE/ACM Transactions on Networking, 23(5):1387–1400, 2014.

[BV16] Peter Buchholz and Sebastian Vastag. An introduction to sla calculus for the analytical validation of slas. Technical report, Tech. rep., TU Dortmnund, Informatik IV, 2016.

[Cha12] Cheng-Shang Chang. Performance guarantees in communication networks. Springer Science & Business Media, 2012.

[CKT03] Samarjit Chakraborty, Simon Künzli, and Lothar Thiele. A general framework for analysing system properties in platform-based embedded system designs. In Date, volume 3, page 10190. Citeseer, 2003.

[Cru91a] Rene L Cruz. A calculus for network delay. i. network elements in isolation. IEEE Transactions on information theory, 37(1):114–131, 1991.

[Cru91b] Rene L Cruz. A calculus for network delay. ii. network analysis. IEEE Transactions on information theory, 37(1):132–141, 1991.

[Cru98] Rene L Cruz. Sced+: Efficient management of quality of service guarantees. In Proceedings. IEEE INFOCOM'98, the Conference on Computer Communications. Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies. Gateway to the 21st Century (Cat. No. 98, volume 2, pages 625–634. IEEE, 1998.

[FR14] Markus Fidler and Amr Rizk. A guide to the stochastic network calculus. IEEE Communications Surveys & Tutorials, 17(1):92–105, 2014.

[HCS24] Anja Hamscher, Vlad-Cristian Constantin, and Jens B Schmitt. Extending network calculus to deal with partially negative and decreasing service curves. arXiv preprint arXiv:2403.18042, 2024.

[LBL07] Chengzhi Li, Almut Burchard, and Jörg Liebeherr. A network calculus with effective bandwidth. IEEE/ACM Transactions on Networking, 15(6):1442–1453, 2007.

[LBT01] Jean-Yves Le Boudec and Patrick Thiran. Network calculus: a theory of deterministic queuing systems for the internet. Springer, 2001.

[LMS05] Luciano Lenzini, Enzo Mingozzi, and Giovanni Stea. Delay bounds for fifo aggregates: a case study. Computer Communications, 28(3):287–299, 2005.

[Pos81] Jon Postel. Transmission control protocol. Technical report, Information Sciences Institute, University of Southern California, 1981.

[SN21] Jens Schmitt and Paul Nikolaus. Lecture notes in worst-case analysis of distributed systems, 2021.

[VRD09] Guido Van Rossum and Fred L. Drake. Python 3 Reference Manual. CreateSpace, Scotts Valley, CA, 2009.

[Wil24] Lukas Wildberger. toolbox:dnc-pwl. Website, 2024. Online available at `https://github.com/LuWild/MasterThesis`; accessed July 14, 2024.