Advances in Theory and Applicability of Stochastic Network Calculus

Thesis approved
by the Department of Computer Science
of the University of Kaiserslautern (TU Kaiserslautern)
for the award of the Doctoral Degree
Doctor of Natural Sciences (Dr. rer. nat.)
to

Michael Beck

Date of the viva	**:**	24 October 2016
Dean:	**:**	Prof. Dr. Klaus Schneider
PhD committee		
Chairperson	**:**	Prof. Dr. Stefan Deßloch
Reviewers	**:**	Prof. Dr. Jens B. Schmitt
		Prof. Dr. Markus Fidler
		Prof. Dr. Anne Bouillard

D 386

Advances in Theory and Applicability of Stochastic Network Calculus

Dipl. Math. Michael Alexander Beck

Abstract

Stochastic Network Calculus (SNC) emerged from two branches in the late 90s: the theory of effective bandwidths and its predecessor the Deterministic Network Calculus (DNC). As such SNC's goal is to analyze queueing networks and support their design and control.

In contrast to queueing theory, which strives for similar goals, SNC uses inequalities to circumvent complex situations, such as stochastic dependencies or non-Poisson arrivals. Leaving the objective to compute exact distributions behind, SNC derives stochastic performance bounds. Such a bound would, for example, guarantee a system's maximal queue length that is violated by a known small probability only.

This work includes several contributions towards the theory of SNC. They are sorted into four main contributions:

(1) The first chapters give a self-contained introduction to deterministic network calculus and its two branches of stochastic extensions. The focus lies on the notion of network operations. They allow to derive the performance bounds and simplifying complex scenarios.

(2) The author created the first open-source tool to automate the steps of calculating and optimizing MGF-based performance bounds. The tool automatically calculates end-to-end performance bounds, via a symbolic approach. In a second step, this solution is numerically optimized. A modular design allows the user to implement their own functions, like traffic models or analysis methods.

(3) The problem of the initial modeling step is addressed with the development of a statistical network calculus. In many applications the properties of included elements are mostly unknown. To that end, assumptions about the underlying processes are made and backed by measurement-based statistical methods. This thesis presents a way to integrate possible modeling errors into the bounds of SNC. As a byproduct a dynamic view on the system is obtained that allows SNC to adapt to non-stationarities.

(4) Probabilistic bounds are fundamentally different from deterministic bounds: While deterministic bounds hold for all times of the analyzed system, this is not true for probabilistic bounds. Stochastic bounds, although still valid for every time t, only hold for one time instance at once. Sample path bounds are only achieved by using Boole's inequality. This thesis presents an alternative method, by adapting the theory of extreme values.

(5) A long standing problem of SNC is the construction of stochastic bounds for a window flow controller. The corresponding problem for DNC had been solved over a decade ago, but remained an open problem for SNC. This thesis presents two methods for a successful application of SNC to the window flow controller.

Danksagung

Eine Dissertation ist mehr als nur eine Abschlussarbeit. Oft markiert sie das Ende und den Beginn eines neuen Lebensabschnitts. So ist es auch in diesem Fall. Als Schriftstück im Buchformat bietet sie (die Dissertation) daher eine hervorragende Möglichkeit, sich bei denjenigen Menschen zu bedanken, die einem wichtig sind. Als Schlussstein einer mehrjährigen Ausbildung verleiht sie dieser Gelegenheit zudem die angemessene Würde. Jede Auflistung führt unvermeidlich dazu, dass viele Menschen unerwähnt bleiben, denen ich ebenfalls dankbar bin. Trotzdem möchte ich einigen Personen besonderen Dank aussprechen:

Ich danke Prof. Jens Schmitt für seine hervorragende Betreuung, welche weit über diese Doktorarbeit hinausreichte. Er lehrte mich die Thematik dieser Arbeit, das akademische Leben im Allgemeinen, Skifahren und sehr vieles mehr.

Ich danke meinem Vater, der mir zeigte, was Arbeit bedeutet und meiner Mutter, die mir zeigte was Spielen bedeutet.

Ich danke meiner Schwester Marina, die mich immer motiviert ähnliches zu erreichen [**11**].

Ich danke Tanja für ihre Liebe und unsere gemeinsamen Erlebnisse, Pläne und Träume.

Ich danke Raul und Jessica für ihre jahrelange Freundschaft und ich danke Helene und Mathilde, welche mir zeigen wie alles begann.

Ich danke Tilman, Claire und Daniel, die mich aus meinem Schneckenhaus holten.

Ich danke unserer jahrelangen Spielrunde für unzählige Pokale, Wünsche und Züge.

Ich danke auch meiner anderen Spielrunde, welche täglich beweist, dass Kommunikationsnetze Freundschaften ermöglichen.

Ich danke meinen Arbeitskollegen und Mitforschern; viele Konferenzbeiträge wurden erst durch ihr Leid für andere erträglich.

Contents

Index of Notations

$i, \ldots, n$	Integer indeces
r, s, t	Time indeces
p, q	Hölder indeces
$\delta, \epsilon, \varepsilon$	Violation probabilities and errors
$\mathbb{N}_0$	Set of natural numbers, including zero
$\mathbb{Z}$	Set of integers
$\mathbb{R}$	Real numbers
$\mathbb{R}_0^+$	Their non-negative part
$\Lambda(\mathbb{K})$	Triangular array on totally ordered set $\mathbb{K}$
$\mathcal{F}$	Space of univariate, non-negative, wide-sense increasing functions
$\mathcal{F}_0$	Its subspace of functions vanishing on $(-\infty, 0]$
$\tilde{\mathcal{F}}$	Space of bivariate functions
$\tilde{\mathcal{F}}_+$	Its subspace of functions that are wide-sense increasing in their second variable
$\tilde{\mathcal{F}}_0$	Its subspace of functions vanishing on all times (t, t)
$A, \ldots, E$	Flows
$a(t), \ldots, e(t)$	Their increments at time t (slotted time only)
$a_c(t), \ldots, e_c(t)$	Their "derivatives" at time t (continuous time only)
$a_j(t), \ldots, e_j(t)$	Their instantanous jumps at time t (continuous time only)
$\mathcal{A}, \ldots, \mathcal{E}$	Their arrival curves and envelopes
$A_x, \ldots, E_x$	Crossflow interfering with $A, \ldots, H$
η, ζ	Error functions
U, V, W	Service functions
$u_c(t), v_c(t), w_c(t)$	Their "derivatives" at time t (continuous time only)
$u(t), v(t), w(t)$	Their increments at time t (slotted time only)
U_l, V_l, W_l	Leftover service of U
Σ	Window element
$\mathfrak{b}(t)$	Backlog at time t
$\mathfrak{d}(t)$	Delay at time t
β	Beta distribution
χ^2	chi-square distribution
Φ	Fréchet distribution
Λ	Gumbel distribution
$\phi_X(\theta)$	MGF of X at θ
(σ, ρ)	Their Sigma-Rho-Bound at θ
X, Y, Z	Random variables.
f_X, f_Y, f_Z	Their density functions
F_X, F_Y, F_Z	Their distribution functions

Π	Space of σ-additive operators on $\tilde{\mathcal{F}}$
$\otimes$	min-plus-convolution
$\mathbf{1}$	Its neutral element
$\bigotimes_{k=0}^{l}$	min-plus-convolution from k to l
$A^{(k)}$	k-fold min-plus-convolution of A with itself
$\oslash$	min-plus-deconvolution
$\wedge$	minimum-operator (for real numbers, or interpret pointwisely on spaces $\mathcal{F}$, $\tilde{\mathcal{F}}$)
$\bigwedge_{k=0}^{l}$	minimum-operator from k to l
$\star$	Convolution on field $\mathbb{R}$
$\circ$	Deconvolution on field $\mathbb{R}$
$\bar{\cdot}$	Subadditive closure

Introduction

"THIS IS MOST IRREGULAR.
We're sorry. It's not our fault.
HOW MANY OF YOU ARE THERE?
More than 1300, I'm afraid.
VERY WELL, THEN. PLEASE FORM AN ORDERLY QUEUE."

- Pyramids, Terry Pratchet.

Queues accompany us all life long. We wait for the traffic light to change or just for the elevator to arrive. Other times we realize their existence only when systems behave sluggish – when we have a "bad connection" or our computer's performance drops suddenly. In these cases the queueing systems cannot handle their jobs timely and consequently the queue lengths become too large or processing times too long. But also short or empty queues can be unwanted, like the order book of a craftsman or the charge level of an energy storage.

What defines a queueing system? In this work a queueing system consists of arrivals, a buffer, and a service element. These objects work together as shown in Figure (a): Arrivals from some source are buffered in front of a service element. The service element, depicted by U, works on these arrivals to produce an output. In the figure arrivals are illustrated as a series of distinguishable objects or packets. In models, as well as in real-world scenarios, this might or might not be true: Instead of being discretized the arrivals can also be a continuous, infinitely divisible medium. The buffer, in which the arrivals are stored, might be modeled as finite or infinite. Many more generalizations are possible: There might be multiple sources of arrivals or multiple service elements. Arrivals might be processed in a first-in-first-out manner or any other. We can continue this list of possible differences between two queueing systems much longer and yet come to no end. As long as there are arrivals being stored in a buffer and a service element to process them, we can speak of a queueing system.

Queues not only accompany, but also influence our daily lives. When we encounter a queueing system, we almost immediately ask, "how long will this take?" or, "how many objects are waiting?" The corresponding answers (or guesses) often serve as basis for the decisions we make. The more important these decisions are, the larger is our need for a good analysis of the system.

Together with telephone exchanges also the first mathematical theory for queueing systems emerged: In 1909 A.K. Erlang lays the foundation for what is known today as *queueing theory* [**58, 59**]. Since then queueing systems propagated into our daily routines. Likewise queueing theory evolved [**82, 94, 91, 9, 148, 151, 119**] and propagated to many applications, for example traffic engineering [**98**] and the design of production facilities [**123, 75**]. Queueing theory helps in planning and analyzing a system's performance. Its results help to maximize performance and minimize risks.

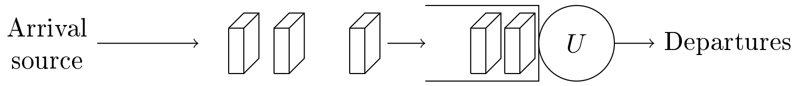


Figure (a): A general queueing system.

Communication networks have continuously changed since the early days of telephony and they will continue to do so. The way we use them and underlying technologies are constantly evolving. And still throughout all these changes the interest in their behavior never ceased. Quite the opposite: As the presence of communication networks increases, the wish for reliability on their performance grows also. With the notion of *quality of service* the interest for performance guarantees in communication networks emerged [**27, 140, 18, 120**]. Next to the *typical* performance of a queueing system, we also want to know how it acts when things go wrong. These are indeed two different problems, which also require different modeling. The following example outside of communication networks illustrates this.

Imagine a football stadium: Before the game starts, the visitors stream to the stadium. They line up in front of security and ticket inspectors and a non-negligible portion also queues to buy refreshments. The stadium's operators are of course interested to provide enough service, that is security personnel, inspectors, and refreshment booths. Although many visitors come to the stadium, there is no need to serve all customers simultaneously. First, the visitors tolerate a certain amount of waiting time and second, all visitors arrive on their own schedule and hence can be served sequentially. If the operators would drop any of these two assumptions, they have to massively (and unviably) increase their service.

Things change dramatically, if in the middle of the football match a fire starts inside the stadium. Now all people need to leave the stadium at the same time. Further there are strict rules how long the evacuation is allowed to take. In total this situation is fundamentally different in its conditions and the desired results.

For the first part of this example queueing theory fits well. It is mainly concerned with the equilibrium of a system and its usual metric is the expected length of queues and delays. However, for an evacuation plan the usual behavior must not be a satisfying answer. The statement "your average escape time is x minutes" should be rather unconvincing. Instead people naturally want to know, "how bad can it get?"

Instead of queueing theory we are inclined to use another model for the second scenario. It should work under worst-case assumptions. For example, visitors might move and react rather slowly. Furthermore, this model is interested in how long it takes to evacuate areas that are far away from exits, instead of taking an average distance. It will also try to produce the highest possible congestions at the thinnest bottlenecks. In short: It takes the role of the devil's advocate and goes with Murphy's law: Anything that can go wrong, will go wrong.

In that sense this model suits naturally the spirit of formal verification and certification.

The seminal works of R. Cruz start a new analysis that implements this worst-case view. In 1991 he lays the foundation for the method known today as *network calculus* [**47, 48**]. Its basic idea is to use much simpler descriptions of the involved processes: Strict bounds on (maximal) arrivals and (minimal) service replace assumptions on their distributions. This step is of additional importance given a new insight of that time: Extensive measurements on internet traffic revealed long range dependencies and self-similarities [**104, 124, 62, 112**]. Two properties that lie in strong contrast to the usual assumptions of queueing theory This leaves the question in how far results from queueing theory apply to internet traffic.

In network calculus, however, the description of arrivals and service is not given explicitly. Instead, they adhere to given specifications called *arrival curves* and *service curves*. Elementary manipulation of the arrival and service curve lead to backlog or delay guarantees. With further effort also networks of systems (a tandem of servers, feedforward networks, feedback loops) and different scheduling policies (FIFO, EDF, GPS) can be analyzed.

One of network calculus' biggest strengths – the analysis of the worst case – is at the same time its biggest drawback. This is illustrated again for the evacuation example: Assume a detailed analysis concludes that the stadium can be timely evacuated for most scenarios. However, if *all* visitors of the western tribune are on their way to buy refreshments the evacuation cannot be achieved on time. Even under very generous assumptions on the visitors' needs, this scenario appears more than unlikely. Situations like these, where sheer possibility meets residual probability, can render network calculus useless.

Apparently one would like to exclude events that only happen with minuscule probability. The result can be a vast improvement on the performance bounds, which must now be understood as soft guarantees: They only hold with a certain, yet computable, probability. In that, these probabilities resemble confidence levels of mathematical statistics. However, the probabilities used in network calculus typically differ from the ones used in mathematical statistics by several orders of magnitude.

The first stochastic extensions appeared quickly after network calculus itself [**32, 96**]. In these a stochastic description replaces the deterministic arrival and service curves. Still there is no distribution assumed, as one would do in queueing theory. Instead, these descriptions are very similar to deterministic curves, but now only hold "most of the time". Manipulations of these curves resemble the ones in pure network calculus and lead to improved bounds. Since these new curves only hold "most of the time", the same restriction must hold for the performance bound too – leading to *stochastic* performance bounds.

The field of *stochastic network calculus* (SNC) is quite young and as such draws research questions from different sources:

- Over the last years two separate, yet related, stochastic extensions evolved: tail-based and MGF-based SNC. They differ in how they bound the involved stochastic processes. This in turn leads to differences, like the scaling of performance bounds in tandem networks [**65, 39**] or how both approaches handle stochastic dependencies. Comparing these two branches to each other will eventually lead to a better foundation and understanding of SNC overall.
- To check SNC results against the insights of queueing theory is an ongoing challenge. Instead of reproducing known results, SNC offers a new approach to hard problems of queueing theory. In scenarios where the

memoryless assumption is dropped or complex topologies appear SNC offers alternative and elegant solutions.

- SNC appears in a simple formulation and the algebraic steps are quickly performed. In contrast to that a numerical evaluation is surprisingly complex: Good performance bounds are sensitive to the numerical optimization of several parameters. Furthermore, exist different ways to derive performance bounds for networks of systems. Hence, useful bounds demand the right approach and an optimal choice of parameters. Both steps are only partially understood by now.
- Some results from deterministic network calculus miss their stochastic counterparts. For example the analysis of a window flow controller has been solved under deterministic bounds over a decade ago. Yet, a corresponding formulation in stochastic network calculus is missing.
- Network calculus almost entirely focuses on communication networks. Applying network calculus and its stochastic extensions in new environments will lead to a better theoretical and practical understanding.

The landscape of possible applications for network calculus' is by far not charted, yet. A hint of what network calculus could be in the future is given by a look on queueing theory. Originating from modeling a concrete application, namely telephone exchanges, queueing theory evolved into a branch of operational research. As such it not only adapted to the ever-changing environment of its main application, but also plays a relevant role in many other fields today. Network calculus has the potential to undergo a similar journey.

Thesis Contributions

This thesis contributes to the theory of stochastic network calculus (SNC) in several ways. For references and related works to each topic see the beginning of the corresponding chapters.

The thesis' first part is a self-contained introduction to deterministic and stochastic network calculus. It differs from other introductive texts by putting forward four operators: the multiplexing and subtraction of flows, as well as the min-plus convolution and deconvolution. These *network operations* form the core of network calculus in two ways: First, the derivation of performance bounds follows by them; and second, they enable the reduction of complex topologies to simpler ones. As such any (stochastic) extension or conversion of network calculus must replicate these four operations.

Chapter 1 presents the network operations mathematically and gives their interpretations. Afterwards follows a discussion on how networks of service elements are simplified by these operations. Chapter 1 concludes by proving the deterministic performance bounds on backlog and delay of a queueing system.

Chapter 2 motivates the two main branches of stochastic extensions to network calculus. How to stochastically bound arrivals and service is introduced and stochastic versions of the network operations are derived. All of these results are given for both branches of SNC. Each description concludes with the derivation of stochastic performance bounds.

The second part of this thesis covers several topics from the area of stochastic network calculus. All of them share the goal of improving its applicability. A more detailed description of these topics is given now:

- There are only a few tools available to automatically derive deterministic performance bounds with network calculus. For SNC even no tool has been existent at all. Chapter 3 presents the *DISCO Stochastic Network Calculator*, the first open source tool for deriving stochastic performance bounds. Due to the variety of possible arrival models and different strategies to accomplish end-to-end analyses, the software follows a modular composition. To this end, it allows its users to implement their own arrival models and end-to-end analyses. The network calculator achieves this by computing the performance bounds on a symbolic level first. In a second step the resulting function is numerically optimized.
- When stochastic network calculus is applied one inevitably makes assumptions about the system. Often these assumptions stem from measurements and a statistical model. However, even extensive measurements cannot rule out modeling errors. While in many fields the probability of erring in the modeling step is of little importance, it becomes non-negligible for SNC. Typical values for violation probabilities in SNC are of order 10^{-3}, 10^{-6} or even 10^{-9}, which exceed easily the usual confidence levels used within statistical methods. This leads to the problem that any statements made by SNC become void, if the probability of having used a wrong parameter is so much higher. However, pushing the confidence level to higher values must result in a more pessimistic view on the system. Chapter 4 presents a method to combine possible modeling errors with SNC. The resulting performance bounds take both aspects into account: First, the possibility of an extreme behavior of the model, and second, the possibility of the model itself being wrong. By this a new level of certainty is reached, when SNC is applied.
- The backlog bounds of deterministic network calculus are, by definition, *never* broken. However, stochastic performance bounds are of a different quality. A stochastic backlog bound holds with a certain probability only. But it also only holds for a specific time t. A simple way to achieve sample path bounds – meaning bounds that are valid on whole intervals of time – is by applying the subadditivity of probabilities. Since the size of backlog of subsequent times are however not stochastically independent, this method is disputable. Chapter 5 introduces a second method based on the theory of extreme values. The backlog bounds of both methods are eventually compared to each other.

The third part of this thesis is entirely dedicated to window flow controlled systems. These kind of systems ensure a deterministic bound on the maximal number of arrivals present in the inner part of the system. Exceeding arrivals are kept outside until the inner part has processed its current workload. Such systems appear for example in the form of transport protocols in communication systems. There the core of the communication channel is protected against a large amount of data, by keeping queueing at the back- and front-ends of the communication. Window flow controlled systems basically trade a higher overall delay for a deterministic guarantee on the backlog inside the inner part.

Chapter 6 collects and presents deterministic results for window flow controlled systems. Special focus lies on the role of subadditivity in the feedback loop. Analyzing window flow controlled systems with stochastic network calculus had been a long standing problem.

Chapter 7 presents two approaches to achieve stochastic performance bounds in this scenario. Both take advantage of the bivariate formulation of MGF-calculus. The first approach is based on preserving subadditivity. The other approach utilizes the stochastic behavior of the system by describing the probability of subadditivity not being available. As the former method is restricted to a class of subadditive service elements it is called the subadditive case. The latter is denoted the general case, as it works without assumptions on the involved service elements.

Part 1

Introduction to Stochastic Network Calculus

CHAPTER 1

Deterministic Network Calculus

This chapter gives a self-contained introduction to network calculus. It focuses on results needed for the second and third part of this thesis. For introductions in a different style see the textbooks of Chang [**35**], Le Boudec and Thiran [**102**], or Jiang [**88**]. The survey by Fidler [**66**] also provides a comprehensive overview of deterministic network calculus.

1.1. Notations

Throughout this thesis time indices often appear as an ordered pair $s \leq t$, where s and t either belong to $\mathbb{N}_0$ or $\mathbb{R}_0^+$; therefore, the following definition is made.

DEFINITION 1.1. Let $\mathbb{K}$ be a totally ordered set. Define the index set $\Lambda(\mathbb{K})$ as

$$\Lambda(\mathbb{K}) := \{(s,t) \in \mathbb{K} \times \mathbb{K} : \, s \leq t\}.$$

Elements of $\Lambda(\mathbb{K})$ are denoted by (s,t).

The usual choice for $\mathbb{K}$ is $\mathbb{N}_0$ or $\mathbb{R}_0^+$. The former represents a slotted time model. The latter describes a continuous time model. In Chapter 4 also negative time indices are allowed and there $\mathbb{K} = \mathbb{Z}$. In the following it always holds $\mathbb{K} \in \{\mathbb{N}_0, \mathbb{R}_0^+\}$, if not mentioned otherwise. If there is no risk of confusion, the index set is just written Λ.

The space domain uses an abstract concept of *arrivals*. Arrivals represent a processable quantity like data, packets, or jobs. In other context they might also be materials, customers, or energy. Arrivals follow a fluid model, i.e., they can be infinitely subdivided. This helps to derive subsequent results. The transformation of results to packetized arrivals is shown for example in Chapter 2 of [**35**].

Flows are the most usual representation of arrivals. They describe the amount of arrivals up to a time $t \in \mathbb{K}$.

DEFINITION 1.2. A *flow* A is a cumulative function

$$\begin{aligned} A : \mathbb{K} &\to \mathbb{R}_0^+ \cup \{\infty\} \\ t &\mapsto A(t) \end{aligned}$$

with $A(0) = 0$.

If time is continuous ($\mathbb{K} = \mathbb{R}_0^+$) flows are considered right-continuous, i.e., $\lim_{t \searrow s} A(t) = A(s)$. If time is slotted ($\mathbb{K} = \mathbb{N}_0$), the *increments* $a(t)$ describe the arrivals during one time-slot. The relation between the increments $(a(t))_{t \in \mathbb{N}_0}$ and the flow A is given by $A(t) = \sum_{s=0}^{t} a(s)$. The bivariate extension of A is defined as

$$\begin{aligned} A : \Lambda &\to \mathbb{R}_0^+ \\ A(s,t) &\mapsto A(t) - A(s). \end{aligned}$$

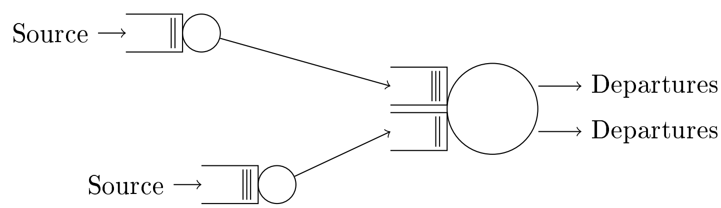


FIGURE 1.1. A simple example of a queueing network. The departures of a queue can form the arrivals at another service element. Further a service element can have several sources of arrivals.

Due to its interpretation of cumulative arrivals, the function A is increasing, i.e., $A(s) \leq A(t)$ for all $(s,t) \in \Lambda$. The following sets fix some of the above properties.

DEFINITION 1.3. Let $\mathcal{F}(\mathbb{K})$ denote the space of increasing functions on $\mathbb{K}$ with codomain $\mathbb{R} \cup \{-\infty, \infty\}$. The subset $\mathcal{F}_0(\mathbb{K}) \subset \mathcal{F}(\mathbb{K})$ describes all $A \in \mathcal{F}(\mathbb{K})$ with $A(0) = 0$; further, let $\tilde{\mathcal{F}}(\mathbb{K})$ denote the space of bivariate functions with codomain $\mathbb{R} \cup \{-\infty, \infty\}$ and $\tilde{\mathcal{F}}_+(\mathbb{K}) \subset \tilde{\mathcal{F}}(\mathbb{K})$ be the subset of bivariate functions that are increasing in their second variable. The subset $\tilde{\mathcal{F}}_0(\mathbb{K}) \subset \tilde{\mathcal{F}}_+(\mathbb{K})$ describes all $A \in \tilde{\mathcal{F}}_+(\mathbb{K})$ with $A(t,t) = 0$ for all $t \in \mathbb{K}$.

If there is no risk of confusion, the notations $\mathcal{F}, \mathcal{F}_0, \tilde{\mathcal{F}}, \tilde{\mathcal{F}}_+$, and $\tilde{\mathcal{F}}_0$ are used. By definition, the set of all possible flows is a subset of $\mathcal{F}_0$. Their bivariate extensions form a subset of $\tilde{\mathcal{F}}_0$.

1.2. Network Operations

While arrivals represent some kind of processable quantity, service elements describe a location at which this quantity is processed. They abstract for example routing elements (serving data), processors (jobs), fabricators (materials), or a counter (customers). Arrivals and service elements might form complex networks in which

- a service element handles several arrivals and
- processed arrivals serve as input to other service elements.

See also Figure 1.1.

Network calculus' goal is to reduce the complexity of such networks. To this end, four operators are defined. Each of them takes as input two functions that represent either an arrival or a service element. The result is a new arrival or a new service element. This simplifies the network of flows and service elements; therefore, they are called *network operations*. The mathematical introduction is given below. An interpretation in the context of arrival and service elements follows afterwards.

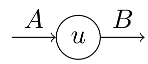


FIGURE 1.2. A single flow A enters a single service element with constant rate u.

DEFINITION 1.4. The *convolution operator* $\otimes$, *multiplexing operator* $\oplus$, *deconvolution operator* $\oslash$, and *subtraction operator* $\ominus$, are defined as

$$\begin{aligned}\otimes, \oplus, \oslash, \ominus : \mathcal{F} \times \mathcal{F} &\to \mathcal{F}\\ A \otimes B(t) &\mapsto \inf_{0\le s\le t}\{A(s) + B(t-s)\}\\ A \oplus B(t) &\mapsto A(t) + B(t)\\ A \oslash B(t) &\mapsto \sup_{0\le s}\{A(t+s) - B(s)\}\\ A \ominus B(t) &\mapsto A(t) - B(t).\end{aligned}$$

The second and last operator represent the pointwise addition and subtraction of the involved functions, respectively. The remaining two operators are analogs of the usual convolution of real functions in the context of min-plus-algebra. In this algebra minima take the role of the addition and the multiplication is replaced by the usual addition (see for instance [**2**] for details on the min-plus algebra).

The following analysis motivates the convolution operator: Assume that a service element has constant service rate u and a flow of arrivals A enters this service element. Denote the departures as flow B. Figure 1.2 illustrates this scenario. If no data is lost or produced inside the service element, the causality-condition $B \le A$ holds; thus, the difference between arrivals and departures must exist as backlog inside the system and is written as $\mathfrak{b}(t) := A(t) - B(t)$. Lindley's equation [**111**] describes the evolution of $\mathfrak{b}$ by

$$\mathfrak{b}(t) = [\mathfrak{b}(t-1) + a(t) - u]^+.$$

This equation gives the backlog at time t as the backlog of the previous slot plus the difference of new arrivals and processed data. As no negative backlog can occur ($B \le A$), the result must be at least zero. By an induction argument, the above resolves to the explicit form

$$\mathfrak{b}(t) = \max_{0\le s\le t}\{A(t) - A(s) - u\cdot(t-s)\}.$$

The convolution operation appears by rewriting the above as

$$B(t) = A(t) - \mathfrak{b}(t) = \min_{0\le s\le t}\{A(s) + u\cdot(t-s)\} = A \otimes U(t), \tag{1.1}$$

where $U(t) := u \cdot t$.

Generalizing Equation (1.1) leads to the definition of service curves [**49, 51**].

DEFINITION 1.5. Assume that a service element has the input-output pair A and B. The service element provides a *service curve* $U \in \mathcal{F}_0$, if

$$B(t) \ge A \otimes U(t). \tag{1.2}$$

The service curve is different from (1.1) in providing more generality and modeling power. It uses a general function U instead of a constant rate service and an

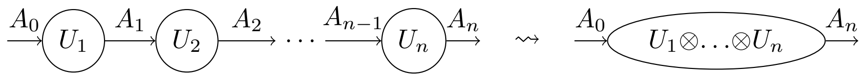


FIGURE 1.3. A tandem of n service elements.

inequality instead of an exact description. This allows the construction of service curves for a variety of service elements.

EXAMPLE 1.6. The most used service curve is the rate-latency-curve [**147**]

$$U_{u,T}(t) = [ut - T]^+$$

with rate u and latency T/u. A server with a rate-latency service curve can be inactive (while being backlogged at the same time) for a period being less than or equal to T/u. After this latency it works at least with rate u.

However, the definition of service curves makes no statement about the maximum service. In this example the service element could be dormant during its latency T/u and then process all accumulated arrivals in one burst. In this case the burst sizes of the output B would exceed those of the input A.

The convolution operator used in Definition 1.5 is commutative; indeed, with the substitution $r = t - s$ it holds

$$\inf_{0 \leq s \leq t} \{A(s) + U(t-s)\} = \inf_{0 \leq r \leq t} \{A(t-r) + U(r)\}. \tag{1.3}$$

In more general scenarios the commutativity is lost, however, for example if $A \notin \mathcal{F}_0$ or in the bivariate setting (see Section 2.3). The service curve description is therefore asymmetric. This hints at a fundamental difference between arrivals and service.

A wider interpretation of the system's elements reveals this difference: The service element converts the quantities A and U into a product B, whereat A and U are handled equally. They differ, however, in the possibility to save or buffer them. Exceeding arrivals are stored in a buffer to be processed in the future; on the other hand, exceeding service cannot be stored. Lindley's equation reflects this in reducing the backlog only by u, excluding any surplus service of the past. Section 9.1 of the appendix discusses how Equation (1.1) changes when both commodities can be stored or when the service element processes more than two commodities.

Although the convolution operator appears in Definition 1.5, its use as a network operation is different. For this let a flow traverse a tandem of multiple service elements as in Figure 1.3. This network reduces, with the help of the convolution operator, to one with only a single service element.

THEOREM 1.7 (Concatenation-Theorem). *Consider a topology as in Figure 1.3. The initial arrival is denoted by A_0 and each departure A_i of service element i is fed directly into the next service element $i+1$. If each service element has a service curve U_i, then the whole system has the service curve $U_1 \otimes \ldots \otimes U_n$, i.e.,*

$$A_n \geq A_0 \otimes U_1 \otimes \ldots \otimes U_n. \tag{1.4}$$

PROOF. The proof is a direct application of the convolution operator's associativity and monotonicity. It is easy to check that

$$(A \otimes B) \otimes C(t) = A \otimes (B \otimes C)(t) \qquad \text{for all } A, B, C \in \mathcal{F},\ t \in \mathbb{K},$$

$$A \otimes U(t) \leq B \otimes U(t) \leq B(t) \qquad \text{for all } A \leq B,\ A, B \in \mathcal{F},\ t \in \mathbb{K}.$$

Applying both properties to the definition of service curves yields

$$A_n(t) \geq A_{n-1} \otimes U_n(t) \geq (A_{n-2} \otimes U_{n-1}) \otimes U_n(t) \geq A_{n-2} \otimes (U_{n-1} \otimes U_n)(t)$$

for all $t \in \mathbb{K}$. Equation (1.4) follows by induction over n. $\square$

The service curve definition connects the convolution operator with a minimum guarantee on the service. In a worst-case analysis such a bound is needed, as the trivial null-service leads to no performance of the system at all; similarly, the deconvolution operator is connected with a maximal bound on the arrivals. If arrivals were unbounded, one could immediately (and infinitely) overload the system. In this case, as with unbounded service elements, no performance guarantees would be possible.

DEFINITION 1.8. A flow A has *arrival curve* $\mathcal{A} \in \mathcal{F}_0$, if it holds

$$A \oslash A(t) \leq \mathcal{A}(t) \tag{1.5}$$

for all $t \in \mathbb{K}$. This is denoted by $A \preceq \mathcal{A}$.

Since $A \oslash A(t) = \sup_{s \geq 0}\{A(t+s) - A(s)\}$, Equation (1.5) restricts the amount of arrivals on each *interval* of length t. Hereby it does not matter at which time s this interval is placed.

EXAMPLE 1.9. The token bucket model can be expressed in an arrival curve [**47, 4, 100**]. In such a model one imagines a bucket of size K is being filled with tokens on a rate a. If the bucket is full, no new tokens are added to it. If a source adheres to the token bucket regulation, it takes a token from the bucket whenever it wants to send an arrival. If there are no tokens in the bucket, the next arrival is delayed until the bucket is refilled. For any interval of length t at most $K + at$ tokens can be taken from the bucket: These are at most K tokens at the beginning of the interval and at most at tokens that are refilling the bucket during the considered interval; hence, it holds

$$A(t+s) - A(s) \leq K + at =: \mathcal{A}(t)$$

for all $s, t \in \mathbb{K}$. This model restricts the arrivals' burst size by K and their average rate by a.

In connection with service elements the deconvolution operator becomes a network operation. It delivers an arrival curve for a service element's output.

THEOREM 1.10 (Output Bound). *Assume that a service element offers a service curve U for a flow $A \preceq \mathcal{A}$. Denote the output flow by B as in Figure 1.4. It holds*

$$B \preceq \mathcal{A} \oslash U. \tag{1.6}$$

PROOF. The proof follows directly by the definitions of service and arrival curves. Let $t \in \mathbb{K}$ be arbitrary, then

$$\begin{aligned}
\sup_{s \geq 0}\{B(t+s) - B(s)\} &\leq \sup_{s \geq 0}\{A(t+s) - A \otimes U(s)\} \\
&= \sup_{s \geq 0}\{\sup_{0 \leq r \leq s}\{A(t+s) - A(r) - U(s-r)\}\} \\
&\leq \sup_{s \geq 0}\{\sup_{0 \leq r \leq s}\{\mathcal{A}(t+s-r) - U(s-r)\}\} \\
&= \sup_{s' \geq 0}\{\mathcal{A}(t+s') - U(s')\} = \mathcal{A} \oslash U(t).
\end{aligned}$$

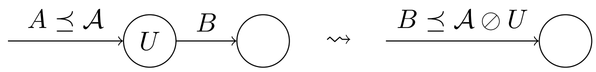


FIGURE 1.4. The output of a system is bounded with the help of the deconvolution operator.

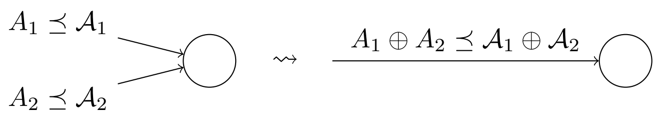


FIGURE 1.5. Two flows with arrival curves are multiplexed into a single one.

□

For the multiplexing operation $\oplus$ let two flows A and B traverse the same part of a network. For simplicity, assume that they both enter a service element denoted by U. Instead of viewing them as two flows, they can be aggregated into the flow $A \oplus B$. The multiplexing operation easily extends to their increments (in the case $\mathbb{K} = \mathbb{N}_0$) and also to their bivariate extensions $\tilde{\mathcal{F}}_0$. This is unsurprising, yet its extension to arrival curves is more interesting.

THEOREM 1.11 (Multiplexing). *Let $A_1 \preceq \mathcal{A}_1$ and $A_2 \preceq \mathcal{A}_2$. The multiplexed flow $A_1 \oplus A_2$ has the arrival curve $\mathcal{A}_1 \oplus \mathcal{A}_2$.*

PROOF. It holds

$$\begin{aligned}(A_1 \oplus A_2) \oslash (A_1 \oplus A_2)(t) &\leq \sup_{s \geq 0}\{A_1(t+s) + A_2(t+s) - A_1(s) - A_2(s)\} \\ &\leq \sup_{s \geq 0}\{A_1(t+s) - A_1(s)\} + \sup_{s \geq 0}\{A_2(t+s) - A_2(s)\} \\ &\leq \mathcal{A}_1(t) + \mathcal{A}_2(t).\end{aligned}$$

for all $t \in \mathbb{K}$. □

So far, the subtraction operator has no interpretation as network operation. To this end, consider a service element that serves two flows under strict priority. This means that the lower priority flow only receives service, if the higher priority flow is not backlogged. The next theorem connects the subtraction operator with strict priority scheduling. To do so, a stronger version of service curves is needed. See the survey of Bouillard et al. [**24**] for a detailed discussion on service curve definitions.

DEFINITION 1.12. A service element offers flow A a *strict* service curve U, if for all times $r \in \mathbb{K}$ inside a backlogged period $[s, t]$ it holds

$$B(s, r) \geq U(r - s).$$

It is easily seen that a strict service curve is also a service curve. The converse is not true in general.

THEOREM 1.13 (Leftover Service Curve). *Consider the scenario as in Figure 1.6. Consider a strict priority scheduling for the incoming flows such that their*

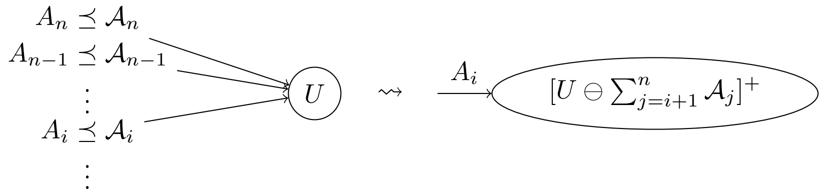


FIGURE 1.6. Subtraction of high priority flows from a service element.

priorities are sorted by their index i. If $A_i \preceq \mathcal{A}_i$ and the service element provides a strict service curve U for the aggregate, the service element offers the service curve $U_i = [U \ominus \sum_{j=i+1}^{n} \mathcal{A}_j]^+$ for flow A_i, i.e.,

$$B_i \geq A_i \otimes U_i. \tag{1.7}$$

PROOF. Fix an arbitrary $r \in \mathbb{K}$ and an index $i \in \{1, \dots, n\}$. Let$s \in \mathbb{K}$ be maximal with

$$\sum_{j=i}^{n} A_j(s) = \sum_{j=i}^{n} B_j(s).$$

Such a time $s \in \mathbb{K}$ always exists since the equation is fulfilled for $s = 0$. For all $t \in \mathbb{K}$ in the period $[s, r]$ it holds by definition of the strict service curve

$$\sum_{j=i+1}^{n} A_j(t) + B_i(t) \geq \sum_{j=i}^{n} B_j(t) \geq \sum_{j=i}^{n} A_j(s) + U(t-s).$$

With

$$\begin{aligned} B_i(t) - A_i(s) &\geq \sum_{j=i+1} A_j(s) - A_j(t) + U(t-s) \\ &\geq U(t-s) - \sum_{j=i+1} A_j(s,t) \geq U(t-s) - \sum_{j=i+1} \mathcal{A}_j(t-s) \end{aligned}$$

and $B_i(r) - A_i(s) = B_i(s, r) \geq 0$ it follows

$$\begin{aligned} B_i(r) &\geq A_i(s) + [U(t-s) - \sum_{j=i+1} \mathcal{A}_j(t-s)]^+ \\ &\geq \inf_{0 \leq s \leq t} \{A_i(s) + U_i(t-s)\} = A_i \otimes U_i(t). \end{aligned}$$

Since the time r was chosen arbitrary the above holds for all $t \in \mathbb{K}$ and Equation (1.7) follows. □

REMARK 1.14. There are leftover service descriptions for other schedulers, such as FIFO [**51**], generalized processor sharing [**35**], and Earliest Deadline First (EDF) [**134**], as well. Of particular interest is the EDF scheduler. A classic result of scheduling theory is Jackson's rule [**81**]. It states that the EDF policy is optimal for a single server with respect to minimizing the total latency (for a proof see for example [**99**]). Jackson's rule is recoverd by Georgiadis et al. [**71**] in the context of communication networks for non-preemptive servers.

The rest of this thesis considers strict priority schedulers only.

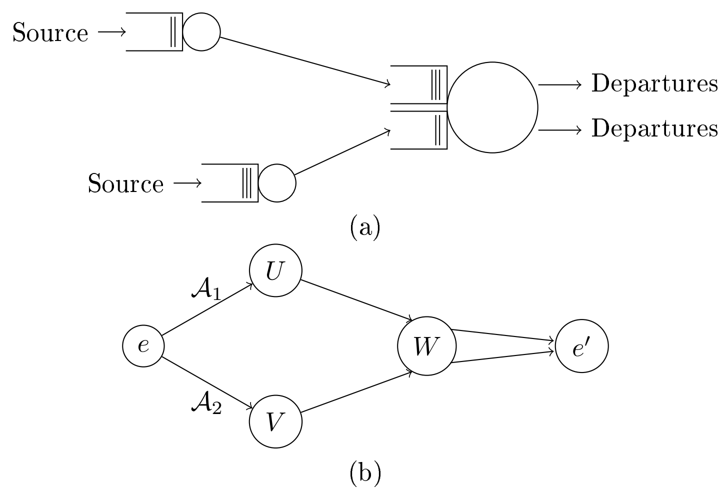


FIGURE 1.7. The original queueing system (a) and its graph representation with initial labels (b).

1.2.1. Networks as Labeled Graphs. The complexity of many networks can be reduced into much simpler ones with the help of the four network operations. To this end, a network is represented by a graph with directed, labeled edges $\mathcal{E}$ and labeled nodes $\mathcal{N}$. Each arrival represents an edge $(i, j) \in \mathcal{N} \times \mathcal{N}$ and each service element represents a node. Multiple edges between the same pair of nodes are allowed. The head of an edge is equivalent to the service element that serves the arrival. The tail of the edge is the node the arrivals originate from. If an arrival enters the network from outside, a dummy node e serves as the corresponding edge's tail. If an arrival leaves the network, a dummy node e' serves as the corresponding edge's head. The labels on the edges are arrival curves $\mathcal{A}$ and the nodes' labels are service curves for the aggregate of incoming flows. Usually the graph is only partially labeled. To obtain a fully labeled graph, the four network operations allow to either

- merge two labeled nodes into a new labeled node, if the aggregated output of the first node equals the aggregated input of the second (convolution operation),
- merge two labeled edges into a new labeled edge, if they have the same head (multiplexing operation),
- merge an unlabeled edge and its labeled tail into a new labeled edge, if the corresponding ingress-edge is labeled (deconvolution operation),
- update a labeled node by canceling a labeled edge, if the head of the edge is equal to the node (subtraction operation).

Table 1 gives an overview of the four network operations and their interpretation in networks.

The goal is to transform the graph to a single labeled node with a single labeled ingress edge. This corresponds to the knowledge of the arrival curve and the (possibly end-to-end) service curve needed to compute performance bounds. To achieve this goal a minimal initial labeling is required. It must cover at least the edges with

Elements merged	Flow $A_2 \preceq \mathcal{A}_2$	Service element with service curve U'
Flow $A_1 \preceq \mathcal{A}_1$	**Operator:** $\oplus$ **Interpretation:** Aggregation of two flows **Result:** Flow $A_1 \oplus A_2 \preceq \mathcal{A}_1 \oplus \mathcal{A}_2$	**Operator:** $\oslash$ **Interpretation:** Computation of an arrival curve for the output B **Result:** Flow $B \preceq \mathcal{A}_1 \oslash U'$
Service element with service curve U	**Operator:** $\ominus$ **Interpretation:** Calculation of leftover service after serving arrival A_2 **Result:** Service $[U \ominus \mathcal{A}_2]^+$	**Operator:** $\otimes$ **Interpretation:** Convolution of two service elements into a single one. **Result:** Service $U \otimes U'$

TABLE 1. Overview of the network operations.

tail e and all nodes. Otherwise the network could be overloaded by infinite arrivals or a service element could offer no service at all.

Today network calculus is capable of reducing a great variety of graphs to the single-node-single-edge case (see for example [**37, 87, 64, 143, 105, 138, 137**] and the references in [**66**]). The next examples illustrate different approaches in performing such a reduction.

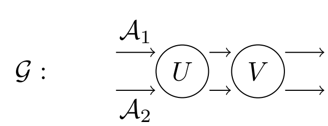


FIGURE 1.8. Network of Example 1.15.

EXAMPLE 1.15. Consider the graph $\mathcal{G}$ given in Figure 1.8. Both service elements are assumed to provide a strict service curve. The graph can be simplified by first merging both arrivals and then either convolute the two service elements ($\mathcal{G}_1$ in Figure 1.9) or calculate an output bound for the departures of the first node ($\mathcal{G}_1'$ in Figure 1.9). The graphs $\mathcal{G}_1$ and $\mathcal{G}_1'$ describe the system for both arrivals aggregated and as such, can also be used to calculate performance bounds for only one of the flows. The graph $\mathcal{G}_1$ describes the end-to-end behavior of the system, whereas $\mathcal{G}_1'$ describes the behavior at the service element V.

$$\mathcal{G}_1 : \quad \mathcal{A}_1 \oplus \mathcal{A}_2 \longrightarrow U \otimes V$$

(a) Convolution after multiplexing.

$$\mathcal{G}_1' : \quad (\mathcal{A}_1 \oplus \mathcal{A}_2) \oslash U \longrightarrow V$$

(b) Deconvolution after multiplexing.

FIGURE 1.9. Resulting graphs for aggregating first.

EXAMPLE 1.16. Another method to reduce $\mathcal{G}$ is to subtract one of the flows – say A_2 – first. Afterwards either the convolution or the deconvolution operator can be applied, leading to the graphs $\mathcal{G}_2$ and $\mathcal{G}_2'$, respectively. Figures 1.10(a) and 1.10(b) show the resulting labels for the networks. The graph $\mathcal{G}_2$ describes an end-to-end behavior, whereas $\mathcal{G}_2'$ is more suitable for the local analysis at the second node. In contrast to the previous example, the flows are considered separately throughout the whole analysis. This approach proves to be better in general topologies, in which flows interfere only locally.

$$\mathcal{G}_2 : \quad \mathcal{A}_1 \longrightarrow [U \ominus \mathcal{A}_2]^+ \otimes [V \ominus (\mathcal{A}_2 \oslash U)]^+$$

(a) Convolution after subtraction.

$$\mathcal{G}_2' : \quad \mathcal{A}_1 \oslash [U \ominus \mathcal{A}_2]^+ \longrightarrow [V \ominus (\mathcal{A}_2 \oslash U)]^+$$

(b) Deconvolution after subtraction.

FIGURE 1.10. Resulting graphs for subtracting first.

EXAMPLE 1.17. Instead of merging one of the edges first, one can also start by a convolution of the two service elements. The resulting node is labeled by $U \otimes V$. The graph $\mathcal{G}_3$ in Figure 1.11(a) equals $\mathcal{G}_1$; indeed, just the order of aggregation and convolution was switched. Subtracting a crossflow from the convoluted service element, instead, would lead to Figure 1.11(b). This step, however, requires more caution: Theorem 1.13 only holds for strict service curves but the convolution of U and V does generally not preserve strictness. Hence, the reduction $\mathcal{G}_3'$ is not achievable without further results. The work of Schmitt et al. [**138**] describes how $\mathcal{G}$ can still be reduced to $\mathcal{G}_3'$.

$$\mathcal{G}_3 : \quad \mathcal{A}_1 \oplus \mathcal{A}_2 \longrightarrow U \otimes V$$

(a) Multiplexing after convolution.

$$\mathcal{G}_3' : \quad \mathcal{A}_1 \longrightarrow [(U \otimes V) \ominus \mathcal{A}_2]^+$$

(b) Subtraction after convolution.

FIGURE 1.11. Resulting graphs for convolving first.

The previous examples make it clear that there exist different strategies to analyze general feedforward networks. The question of how to analyze a (possibly large) network optimally – or just computationally efficient – is a topic of current research [**137, 25, 20, 19**].

1.3. Performance Bounds

After a labeled graph is successfully reduced, the goal of network calculus is to derive performance bounds of the system. Two metrics are of main concern here: the backlog $\mathfrak{b}$ and the time to traverse the system. The latter is captured by the definition of virtual delay:

$$\mathfrak{d}(t) := \inf\{s \geq 0 \,:\, A(t) \leq B(t+s)\}.$$

The notions of arrival curves and service curves lie at the core of the following central result.

THEOREM 1.18 (Fundamental Theorem of Network Calculus). *Let $A \preceq \mathcal{A}$ be a flow entering a service element with service curve U. Then the following performance bounds hold*

$$\mathfrak{b}(t) \leq \sup_{0\leq r}\{\mathcal{A}(r) - U(r)\} \leq \mathcal{A} \oslash U(0) \qquad \textit{for all } t \in \mathbb{K} \tag{1.8}$$

$$\mathfrak{d}(t) \leq \inf\{s \geq 0 \,:\, \mathcal{A} \oslash U(-s) \leq 0\} \qquad \textit{for all } t \in \mathbb{K}. \tag{1.9}$$

REMARK 1.19. Note that in Equation (1.9) the definition of $\oslash$ is implicitly extended to $\mathbb{K} = \mathbb{R}$ and $\mathbb{K} = \mathbb{Z}$, respectively.

PROOF. The proofs for both bounds follow the same three steps. Firstly a connection between the performance measure and the flows A and B is established. In a second step the service curve replaces B. This results in an expression of U and A only. In a last step the arrival curve replaces A.

For arbitrary $t \in \mathbb{K}$ and the backlog it holds

$$\begin{aligned}
\mathfrak{b}(t) &= A(t) - B(t) \\
&\leq A(t) - \inf_{0\leq r\leq t}\{A(t-r) + U(r)\} = \sup_{0\leq r\leq t}\{A(t) - A(t-r) - U(r)\} \\
&\leq \sup_{0\leq r\leq t}\{\mathcal{A}(r) - U(r)\} \leq \mathcal{A} \oslash U(0).
\end{aligned}$$

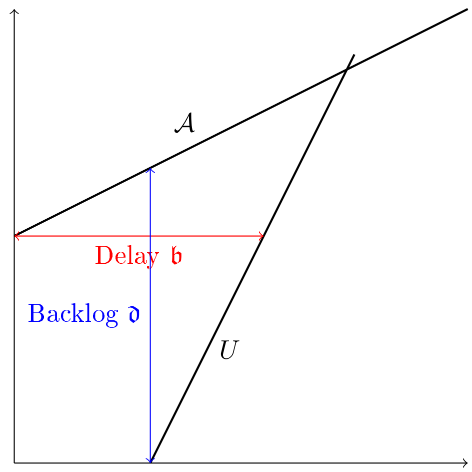


FIGURE 1.12. The backlog bound equals the vertical deviation between $\mathcal{A}$ and U and the delay bound corresponds to the horizontal deviation.

To prove the delay bound fix some $s \in \mathbb{K}$ and assume that there exists a $t \in \mathbb{K}$ with $\mathfrak{d}(t) > s$. From the definition of virtual delay it follows that

$$\begin{aligned}
0 &< A(t) - B(t+s) \\
&\leq A(t) - \inf_{0 \leq r \leq t+s} \{A(t+s-r) + U(r)\} = \sup_{0 \leq r \leq t+s} \{A(t) - A(t+s-r) - U(r)\} \\
&\leq \sup_{0 \leq r \leq t+s} \{\mathcal{A}(r-s) - U(r)\} \leq \mathcal{A} \oslash U(-s).
\end{aligned}$$

The inversion of the achieved implication reads as

$$\mathcal{A} \oslash U(-s) \leq 0 \quad \Rightarrow \quad \mathfrak{d}(t) \leq s \quad \text{for all } t \in \mathbb{K}.$$

Since U lies in $\mathcal{F}_0$, the function $\mathcal{A} \oslash U(-s)$ is decreasing in s. Choosing s minimal results in (1.9). $\square$

The expressions on the right-handed side of Equations (1.8) and (1.9) have a geometric interpretation as well. They describe the maximal vertical and horizontal distance between $\mathcal{A}$ and U, respectively (see Figure 1.12).

The proof for the delay bound appears a bit harder. This is due to flows and their curves being defined in the space domain, whereas delay is a metric of the time domain. In fact, one can adapt the above definitions such that arrival and service curves are expressed in the time domain. In that case the delay bound follows analogously to the above backlog bound and vice versa, see Appendix 9.2 for details.

CHAPTER 2

Stochastic Network Calculus

When systems possess stochastic elements deterministic network calculus can run into heavy problems. This happens whenever the worst case is defined by an event that is possible, yet not very likely. The next example illustrates this.

EXAMPLE 2.1. Assume a flow A has positive, stochastically independent, and identically distributed (i.i.d.) increments $(a(t))_{t\in\mathbb{N}_0}$ with distribution F and define $x_+ := \inf\{x : F(x) = 1\} \in [0,\infty]$. Then for every $x < x_+$ and $t \in \mathbb{N}$ it holds with positive probability

$$A(t) \geq t \cdot x.$$

Hence, a worst case deterministic arrival curve on such a flow must fulfill $\mathcal{A}(t) \geq tx_+$ for all $t > 0$.

The situation of the above example can be in strong contrast to the arrivals' actual behavior. If the increments are, for example, exponentially distributed with some parameter λ, it holds $x_+ = \infty$ and the only possible arrival curve is

$$\mathbf{1}(t) := \begin{cases} 0 & \text{if } t \leq 0 \\ \infty & \text{if } t > 0 \end{cases}.$$

This "worst case" behavior, however, is never observed. Quite contrary $A(t)$ is finite for any $t \in \mathbb{N}$ and the central limit theorem states $A(t) \sim {}^{t}\!/\!_{\lambda}$. Moreover, for any $t \in \mathbb{N}$ and $\varepsilon > 0$ exists a value M such that $\mathbb{P}(A(t) > M) < \varepsilon$; i.e., there is always an upper bound on $A(t)$ that is valid for probabilities arbitrarily close to one. This clearly shows the gap between the "typical" behavior and the worst case scenario when stochastic arrivals (or services) are involved.

Another example considers an aggregate of deterministically bounded flows. Assume a large number N of flows that are all bounded separately by a token bucket arrival curve $\mathcal{A}(t) \leq K + at$. Assume further that all of these flows arrive at the same service element. In a worst case scenario the flows would "gang up" and use their entire bucket simultaneously to create a burst of size $N \cdot K$. Yet, this kind of behavior might miss reality. If the traffic-sources are independent, there is no reason to assume that they conspire against the system in such a way. In fact the larger N is the more unlikely this becomes and the results of deterministic network calculus are less useful than paranoid.

Due to these shortcomings stochastic extensions of network calculus are used. Instead of obtaining hard bounds, which are never violated, one is satisfied now with softer statements. "Softness" means here that bounds can be broken, but only with (very) small probabilities. However, to achieve full modeling power the network operations of network calculus must be carried over: multiplexing, subtraction, convolution and deconvolution. If this is achieved, the result is a *stochastic* network calculus.

Today one can roughly distinguish between two stochastic network calculi. They differ in how stochastic processes are described: Tailbounds cut off unwanted (and unlikely) outcomes, whereas Moment Generating Functions (MGF) are a compact way to describe a distribution as a whole.

This chapter introduces both branches, starting with the tailbounded approach. For both methods the stochastic verisons of the network operations are proven, followed by the respective performance bounds on backlog and delay. Further, the Appendix 10.1 discusses how to transform both calculi into each other.

A considerable portion of network calculus' literature is considered with the development of stochastic extensions. The beginnings of stochastic network calculus are marked by the works of Kurose [**96**], Yaron and Sidi [**161**], and Chang [**33**]. Here Yaron and Sidi lay down the basis for the tailbounded approach, whereas Chang leads the direction towards the MGF-bounded method. The former is further developed by Cruz [**50**], Boorstyn et al. [**21**], Starobinski and Sidi [**144**], and Yin et al. [**162**] to name only a few. The theory of MGF-bounded network calculus is pushed forward by Kesidis et al. [**93**], Chang et al. [**35, 36**], and Fidler [**65**]. The textbooks of Chang [**35**] and Jiang [**88**] are focusing on MGF-calculus and tailbounded network calculus, respectively. Further [**68**] gives a good and extensive overview of the corpus of works in stochastic network calculus.

2.1. Network Operations in Tailbounded Network Calculus

Tailbounds are a straightforward way of generalizing arrival and service curves. The main idea is to define an arrival curve that can be broken by a given probability. For this two types of functions are needed: envelope functions (or just envelopes)

$$\begin{aligned}\mathcal{A} :\ \mathbb{K} \times \mathbb{R}^+ &\to \mathbb{R}_0^+ \cup \{\infty\}\\ (t, \varepsilon) &\mapsto \mathcal{A}(t, \varepsilon),\end{aligned}$$

and error functions (or just errors)

$$\begin{aligned}\eta :\ \mathbb{K} \times \mathbb{R}^+ &\to [0, 1]\\ (t, \varepsilon) &\mapsto \eta(t, \varepsilon).\end{aligned}$$

Envelopes and errors bound arrivals in the following way: The envelope $\mathcal{A}$ is in its first variable an arrival curve and describes the "typical" behavior of A. The second variable of $\mathcal{A}$ describes a deviation from this behavior, e.g., by an additive constant ε. This deviation appears in the error function again: Larger deviations ε should happen with a smaller probability; hence, η is generally decreasing in ε. The time t in the error function gives additional flexibility in its definition: A deviation by ε might occur more or less likely depending on the considered interval's length t.

These relations are condensed in the definition of tailbounds.

DEFINITION 2.2. A flow A is *tailbounded with envelope function $\mathcal{A}$ and error function η*, written shortly $A \preceq (\mathcal{A}, \eta)$, if for all $(s, t) \in \Lambda$ and $\varepsilon > 0$ it holds

$$\mathbb{P}(A(s, t) > \mathcal{A}(t - s, \varepsilon)) \leq \eta(t - s, \varepsilon). \tag{2.1}$$

EXAMPLE 2.3. Assume that a flow fulfills $A \preceq \mathcal{A}_{det}$. Then $A \preceq (\mathcal{A}_{tb}, \eta)$ with $\mathcal{A}_{tb}(t, \varepsilon) = \mathcal{A}_{det}(t)$ and $\eta(t, \varepsilon) = 0$ for all $t \in \mathbb{K}$, $\varepsilon > 0$. Hence, Definition 2.2 generalizes arrival curves.

Several variations of tailbounded arrivals have been developed for stochastic network calculus [**117, 66, 43**]. The most important are the following three examples.

Example 2.4. The *exponentially bounded burstiness* (EBB) model was established by Yaron and Sidi [**161**]. It uses the envelope $\mathcal{A}(t-s,\varepsilon)=\rho(t-s)+\varepsilon$ for some fixed rate ρ and the eponymous error $\eta(t-s,\varepsilon)=Me^{-\theta\varepsilon}$ with some prefactor $M>0$ and decay $\theta>0$. The EBB model connects to effective bandwidths via Chernoff's inequality (see also Section 2.3) [**92, 106**]. For this define the effective bandwidth of A as the normalized *cumulant generating function*

$$E(s,\theta):=\sup_{t\in\mathbb{K}}\left\{\frac{1}{\theta s}\log\mathbb{E}(e^{\theta A(t,t+s)})\right\}\qquad\text{for all } s\in\mathbb{K},\ \theta\in\mathbb{R}_0^+.$$

By L'Hôpital's rule and the properties of cumulant and moment generating functions (see Section 2.3 for a definition) it holds $\lim_{\theta\to 0}E(s,\theta)=\mathbb{E}(A(t,t+s))$ and $\lim_{\theta\to\infty}E(s,\theta)=x_+$. Here x_+ denotes the right endpoint of the distribution of $A(t,t+s)$ and the latter equality requires A to be stationary. This means that $E(s,\theta)$ lies between the average rate and peak rate of A. The effective bandwidth can still exist even if $x_+=\infty$. In this case there is no defined peak rate.

For the connection to tailbounds assume that $E^*(\theta):=\limsup_{s\to\infty}E(s,\theta)$ is finite. Then it holds

$$\mathbb{P}(A(s,t)>(t-s)E^*(\theta)-\tfrac{1}{\theta}\log M+\varepsilon)\le e^{-\theta(t-s)E^*(\theta)}Me^{-\theta\varepsilon}\mathbb{E}(e^{\theta A(s,t)})\le Me^{-\theta\varepsilon}$$

for all $(s,t)\in\Lambda$ and $\varepsilon>0$. The first step uses Chernoff's inequality, which states $\mathbb{P}(X>x)\le e^{-\theta x}\mathbb{E}(e^{\theta X})$. It follows that flows with existent effective bandwidth have for each $\theta>0$, with $\mathbb{E}(e^{\theta A(t,t+s)})<\infty$ an EBB envelope $\mathcal{A}(t,\varepsilon)=tE^*(\theta)-\frac{1}{\theta}\log M+\varepsilon$.

Example 2.5. The *stochastically bounded burstiness* (SBB) model [**144, 40**] uses the envelope $\mathcal{A}(t-s,\varepsilon)=\rho(t-s)+\varepsilon$ and allows the error to be n-fold integrable, meaning

$$\underbrace{\int_\varepsilon^\infty\cdots\int_\varepsilon^\infty}_{n\text{-times}}\eta(t-s,x)(\mathrm{d}x)^n<\infty.$$

The error in the EBB model fulfills

$$\int_\varepsilon^\infty\cdots\int_\varepsilon^\infty\eta(t-s,x)(\mathrm{d}x)^n=\frac{1}{\theta^n}Me^{-\theta\varepsilon}$$

and hence is a special case of the SBB model. The SBB model has more modeling power. For example, it can bound fractional Brownian motions (see Example 2.7). Further, Markov-modulated arrivals can be better fitted by the SBB model compared to the EBB model [**144**].

Example 2.6. This example is not a tailbound in the sense of (2.1), but is strongly related to it. A has a *generally stochastically bounded burstiness* [**162**], if there exists an envelope $\mathcal{A}$ and error η such that for all $t\in\mathbb{K}$ it holds

$$\mathbb{P}(\sup_{s\le t}\{A(s,t)-\mathcal{A}(t-s,\varepsilon)\}>0)\le\eta(t,\varepsilon).\tag{2.2}$$

The difference between the above relation and Definition 2.2 is the supremum inside the probability. Instead of a pointwise violation probability for each point $(s,t)\in\Lambda$, a pathwise violation probability for each $t\in\mathbb{K}$ is considered. Although, appearing as a stronger statement, the bound (2.2) can be constructed from a tailbound of the form (2.1) (see for example [**162**]). The kind of tailbounds in this example have even further modeling power. They can capture for example heavy-tailed arrivals [**86, 70, 162, 107**]. Further they achieve backlog bounds easier, as discussed later.

The next two examples present tailbounds with non-linear envelopes.

EXAMPLE 2.7. Assume that the cumulative arrivals A consist of a constant rate λ and a fractional Brownian motion W_t^H, i.e., $A(t) = \lambda t + W_t^H$ [**116, 104, 121, 122**]. Here $H \in [0, 1]$ is the Hurst-parameter of the fractional Brownian motion. A tailbound can be constructed via its effective bandwidth [**92, 106**] (see also the construction in [**132**]) and is an example for a non-linear envelope $\mathcal{A}$. The tailbound has the form

$$\mathbb{P}(A(s,t) > \lambda(t-s) + \sqrt{\log \varepsilon^{-2}}\sigma(t-s)^H) \leq \varepsilon \tag{2.3}$$

with σ being the variance of the fBm. In [**132**] Equation (2.3) leads to a tailbound of the gSBB type with envelope and error

$$\mathcal{A}(t,\varepsilon) = \lambda t + \sqrt{\log \varepsilon^{-2}}\sigma t^{H+\beta}, \qquad \eta(t,\varepsilon) = \frac{\Gamma(\frac{1}{2\beta})}{2\beta(-\log \varepsilon)^{1/2\beta}}.$$

Here $\varepsilon \in (0, 1)$, Γ is the gamma-function, and $\beta \in (0, 1-H)$ is a free parameter.

In this example $\lim_{t\to\infty} \frac{\mathcal{A}(t,\varepsilon)}{t} = \lambda$ for each ε, i.e., the arrival's long-term rate is λ. This actually allows to approximate $\mathcal{A}$ by a linear envelope again, which in turn allows to derive performance bounds (see Theorem 2.22).

EXAMPLE 2.8. Assume that M flows have arrival rates ρ. Further, each flow i ceases to exist after a random time X_i, where the X_i are independent, exponentially distributed with parameter λ. A single flow's cumulatives are

$$A_i(t) = \begin{cases} \rho t & \text{if } t < X_i \\ \rho X_i & \text{if } t \geq X_i \end{cases}.$$

Thus, the expectation of a flow A_i at time t is

$$\mathbb{E}(A_i(t)) = \frac{\rho}{\lambda}(1 - e^{-\lambda t}).$$

For arbitrary $(s,t) \in \Lambda$, the aggregate of all M flows, denoted by A, fulfills

$$\mathbb{E}(A(s,t)) = \sum_i \mathbb{E}(A_i(t)) - \mathbb{E}(A_i(s)) = \frac{M\rho}{\lambda}(e^{-\lambda s} - e^{-\lambda t}) \leq \frac{M\rho}{\lambda}(1 - e^{-\lambda(t-s)}).$$

The last step is a multiplication by $e^{\lambda s} \geq 1$.

Applying Markov's inequality results in the tailbound

$$\mathbb{P}(A(s,t) > M(1 - e^{-\lambda(t-s)})\varepsilon) \leq \frac{\mathbb{E}(A(s,t))}{M(1 - e^{-\lambda(t-s)})\varepsilon} = \frac{\rho}{\lambda\varepsilon}.$$

As a last step replace ε with $e^{\theta\epsilon}$ for the EBB-like description

$$\mathbb{P}(A(s,t) > M(1 - e^{-\lambda(t-s)})e^{\theta\epsilon}) \leq \frac{\rho}{\lambda}e^{-\theta\epsilon}$$

with the non-linear envelope $\mathcal{A}(t,\epsilon) = M(1 - e^{-\lambda t})e^{\theta\epsilon}$. In contrast to the previous example this envelope has no long-term rate. Instead the convergence $\lim_{t\to\infty} \mathcal{A}(t,\epsilon) = Me^{\theta\epsilon}$ holds. This upper bounded envelope factors in the almost sure finiteness of the arrivals.

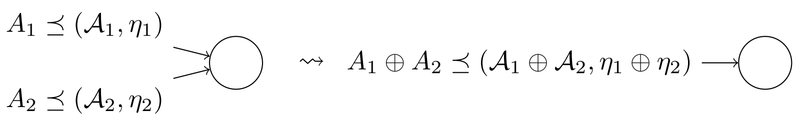


FIGURE 2.1. Tailbounded version of the multiplexing operator.

The network operation of multiplexing carries over to tailbounds (compare Theorem 1.11).

THEOREM 2.9 (Tailbounded Multiplexing). *Let $A_i \preceq (\mathcal{A}_i, \eta_i)$ be two tailbounded flows ($i = 1, 2$). Then $A_1 \oplus A_2 \preceq (\mathcal{A}_1 \oplus \mathcal{A}_2, \eta_1 \oplus \eta_2)$, where $\oplus$ denotes the pointwise addition of $\mathcal{A}_1$ and $\mathcal{A}_2$, η_1 and η_2, respectively.*

PROOF. Choose an arbitrary $(t - s) \in \Lambda$ and $\varepsilon > 0$. If $A_1(t-s) \leq \mathcal{A}_1(t-s,\varepsilon)$ and $A_2(t-s) \leq \mathcal{A}_1(t-s,\varepsilon)$, then $A_1(t-s) + A_2(t-s) \leq \mathcal{A}_1(t-s,\varepsilon) + \mathcal{A}_2(t-s,\varepsilon)$. Reading the previous implication as probabilities results in

$$\begin{aligned}
&\mathbb{P}(A_1 \oplus A_2(t-s) > \mathcal{A}_1 \oplus \mathcal{A}_2(t-s,\varepsilon)) \\
\leq &\mathbb{P}(A_1(t-s) > \mathcal{A}_1(t-s,\varepsilon)) + \mathbb{P}(A_2(t-s) > \mathcal{A}_2(t-s,\varepsilon)) \\
\leq &\eta_1(t-s,\varepsilon) + \eta_2(t-s,\varepsilon).
\end{aligned}$$

□

REMARK 2.10. If the envelopes take a linear form in ε, i.e., $\mathcal{A}_i(t,\varepsilon) = \tilde{\mathcal{A}}_i(t) + \varepsilon$ ($i \in \{1, 2\}$), the above can be improved to $A_1 \oplus A_2 \preceq (\mathcal{A}_1 \oplus \mathcal{A}_2, \eta_1 \otimes \eta_2)$. Here the convolution is taken with respect to the second variables of η_1 and η_2. If further A_1 and A_2 are stochastically independent, the bound on their aggregate can be improved to $A_1 \oplus A_2 \preceq (\mathcal{A}_1 \oplus \mathcal{A}_2, \eta)$. Here $1 - \eta(t,\varepsilon) = (1 - \eta_1(t)) \star (1 - \eta_2(t))$, where the ordinary convolution $\star$ is taken with respect to the second variables. This statement is proven in [**162**] by Yin et al. (there for the gSBB model).

Similar to stochastic arrival curves there are several stochastic service curve definitions in the literature (for an overview see again [**66, 68**]). The one used in this thesis is the most similar to Definition 2.2.

DEFINITION 2.11. Consider an input-output pair A and B of a service element and an envelope U. The service element provides a *tailbounded service curve U with error ζ*, written shortly (U, ζ), if for all $t \in \mathbb{K}$ it holds

$$\mathbb{P}(B(t) < (A \otimes U(\varepsilon))(t)) < \zeta(t,\varepsilon). \tag{2.4}$$

The above convolution reads as $\inf_{0 \leq s \leq t}\{A(s) + U(t-s,\varepsilon)\}$.

A service element often shows stochastic behavior, due to some random cross-flow interfering with an otherwise deterministic service guarantee.

EXAMPLE 2.12. Let time be slotted, i.e., $\mathbb{K} = \mathbb{N}_0$. Assume that a strict service element offers a constant service rate ρ to two inputs A_1 and $A_2 \preceq (\mathcal{A}, \eta)$. Fix some $t \in \mathbb{N}_0$ and $\varepsilon > 0$ and assume it holds

$$A_2(s,t) \leq \mathcal{A}(t-s,\varepsilon) \tag{2.5}$$

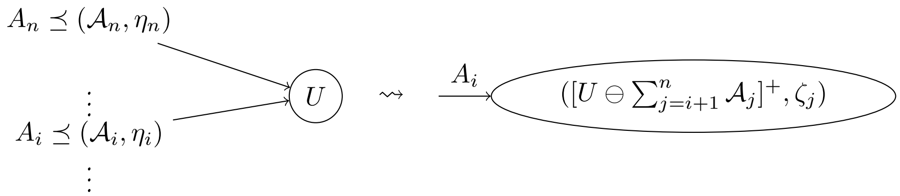


FIGURE 2.2. Tailbounded version of the subtraction operator.

for all $s \leq t$. Under Condition (2.5) Theorem 1.13 yields $B_1(t) \geq (A_1 \otimes U_1(\varepsilon))(t)$. Transferring this implication to probabilities leads to

$$\mathbb{P}(B_1(t) < (A \otimes U(\varepsilon))(t)) \leq \mathbb{P}\Big(\bigcup_{s \leq t} A_2(s,t) > \mathcal{A}(t-s,\varepsilon)\Big),$$

which continues by applying Boole's inequality to

$$\begin{aligned}\mathbb{P}(B_1(t) < (A \otimes U(\varepsilon))(t)) &\leq \sum_{s=0}^{t} \mathbb{P}(A_2(s,t) > \mathcal{A}(t-s,\varepsilon)) \\ &\leq \sum_{s=0}^{t} \eta(t-s,\varepsilon) = \sum_{s=0}^{t} \eta(s,\varepsilon).\end{aligned}$$

Hence the service element offers the service curve (U_1, ζ) with $U_1(t,\varepsilon) = [\rho t - \mathcal{A}(t,\varepsilon)]^+$ and $\zeta(t,\varepsilon) = \sum_{s=0}^{t} \eta(s,\varepsilon)$. The error ζ converges in t, if $A_2 \preceq (\mathcal{A}, \eta)$ has SBB as in Example 2.5. Furthermore the step of applying Boole's inequality is skipped, if one has a gSBB envelope as in Example 2.6.

The network operation $\ominus$ is a straightforward generalization of the above example.

THEOREM 2.13 (Tailbounded Leftover Service). *Let $\mathbb{K} = \mathbb{N}_0$. Assume that a service element offers a strict service curve U to the arrivals $A_i \preceq (\mathcal{A}_i, \eta_i)$ ($i = 1, \ldots, n$) and the incoming flows' priorities are sorted by their index i. Then the service element offers to flow A_j a service curve (U_j, ζ_j), with*

$$U_j(t,\varepsilon) = [U(t) - \sum_{i=j+1}^{n} \mathcal{A}_i(t,\varepsilon)]^+, \qquad \zeta_j(t,\varepsilon) = \sum_{i=j+1}^{n} \sum_{s=0}^{t} \eta_i(s,\varepsilon).$$

REMARK 2.14. This theorem uses a deterministic strict service curve. A similar theorem can also be formulated for a tailbounded version of strict service curves.

REMARK 2.15. In Example 2.12 and the above theorem appears the summation of error functions: $\sum_{s=0}^{t} \eta(s,\varepsilon)$. In general one wants error functions to be summable, i.e.,

$$\sum_{s=0}^{t} \eta(s,\varepsilon) \leq \sum_{s=0}^{\infty} \eta(s,\varepsilon) < \infty.$$

A summable error function is important, as it allows to analyze systems independent of time t. This means it does not matter how far (into the future) a system is analyzed, as one still achieves finite bounds. Such a property is favored, especially for stationary systems. Remark 2.19 after the next theorem provides a method to

$$A \preceq (\mathcal{A}, \eta) \longrightarrow (U, \zeta) \xrightarrow{B} \bigcirc \qquad \rightsquigarrow \qquad B \preceq (\mathcal{A} \oslash U, \bar{\eta} \oplus \zeta) \longrightarrow \bigcirc$$

FIGURE 2.3. Tailbounded version of the deconvolution operator.

achieve summable error functions. It relies on the structure of linear arrival curves $\mathcal{A}$. For the error of tailbounded service curves the situation is more complex – yet still tractable. Eventually, Chapter 7 presents a scenario for which no known method exists to achieve summable error functions.

THEOREM 2.16 (Tailbounded Output Bound). *Let $A \preceq (\mathcal{A}, \eta)$ with $\mathcal{A}(t, \varepsilon) = \rho t + \varepsilon$ arrive at a service element with service (U, ζ). Denote the output by B. Then $B \preceq (\mathcal{A} \oslash U, \bar{\eta} \oplus \zeta)$, where $\bar{\eta}$ depends on η and is defined in the proof.*

PROOF. Fix some $(s, t) \in \Lambda$, $\varepsilon > 0$ and assume for a while that

$$A(r, t) \leq \mathcal{A}(t - r, \varepsilon) \qquad \text{for all } r \leq s \tag{2.6}$$

$$B(s) \geq A \otimes U(\varepsilon)(s). \tag{2.7}$$

By these assumptions A has arrival curve $\mathcal{A}(\varepsilon)$ on $[0, s]$ and the service element offers the service curve $U(\varepsilon)$ at s in the deterministic sense. Theorem 1.10 yields $B(s, t) \leq \mathcal{A}(\varepsilon) \oslash U(\varepsilon)(t - s)$.

Now, similar to Example 2.12, the probability that assumption (2.6) or (2.7) does not hold is bounded by

$$\begin{aligned}
&\mathbb{P}(B(s, t) > \mathcal{A}(\varepsilon) \oslash U(\varepsilon)(t - s)) \leq \mathbb{P}((2.6) \text{ or } (2.7) \text{ does not hold}) \\
\leq\, &\mathbb{P}\Big(\bigcup_{r \leq s} A(r, t) > \mathcal{A}(t - r, \varepsilon)\Big) + \mathbb{P}(B(s) > A \otimes U(\varepsilon)(s)) \\
\leq\, &\mathbb{P}\Big(\bigcup_{r \leq s} A(r, t) > \mathcal{A}(t - r, \varepsilon)\Big) + \zeta(t, \varepsilon).
\end{aligned}$$

For $\mathbb{K} = \mathbb{N}_0$ the above probability is treated as in Example 2.12 and $\bar{\eta}(t, \varepsilon) = \sum_{r=0}^{t} \eta(r, \varepsilon)$. For $\mathbb{K} = \mathbb{R}_0^+$, however, a direct application of Boole's inequality is not possible. Instead the interval $[0, s]$ must be discretized first.

To this end, denote by $(T_i)_{i \in I}$ a countable partition of the interval $[0, s]$, i.e., for all $i \neq j$ it holds $T_i \cap T_j = \emptyset$ and $\bigcup_{i \in I} T_i = [0, s]$. Further let

$$s_{i,u} := \sup\{r \mid r \in T_i\}, \qquad s_{i,l} := \inf\{r \mid r \in T_i\}$$

be the right and left endpoints of T_i, respectively.

Now the event $\{\bigcup_{r \leq s} A(r, t) > \mathcal{A}(t - r, \varepsilon)\}$ can be written as

$$\begin{aligned}
A(t) > \inf_{0 \leq r \leq s} \{A(r) + \mathcal{A}(t - r, \varepsilon)\} &= \min_{i \in I} \inf_{r \in T_i} \{A(r) + \mathcal{A}(t - r, \varepsilon)\} \\
&\geq \min_{i \in I} \{A(s_{i,l}) + \mathcal{A}(t - s_{i,u}, \varepsilon)\},
\end{aligned}$$

where A and $\mathcal{A}$ are increasing in the time variable. Eventually, Boole's inequality applies and finishes the proof:

$$\begin{aligned}\mathbb{P}(\bigcup_{r\leq s} A(r,t) > \mathcal{A}(t-r,\varepsilon)) &\leq \mathbb{P}(\bigcup_{i\in I} \mathcal{A}(t-s_{i,u},\varepsilon) - A(s_{i,l},t) < 0)\\ &\leq \sum_{i\in I} \mathbb{P}(A(s_{i,l},t) > \mathcal{A}(t-s_{i,u},\varepsilon))\\ &= \sum_{i\in I} \mathbb{P}(A(s_{i,l},t) > \mathcal{A}(t-s_{i,l},\varepsilon-\rho(s_{i,u}-s_{i,l})))\\ &\leq \sum_{i\in I} \eta(t-s_{i,l},\varepsilon-\rho(s_{i,u}-s_{i,l})) =: \bar{\eta}(t,\bar{\varepsilon}).\end{aligned}$$

□

Remark 2.17. A particular useful discretization is a partition into intervals of equal lengths, say τ. In this work such an equidistant partitioning is called *canonical*.

Remark 2.18. The discretization-step of the above proof relies on the envelope's linearity to perform the transformation

$$\mathcal{A}(t-s_{i,u},\varepsilon) = \mathcal{A}(t-s_{i,l},\varepsilon-\rho(s_{i,u}-s_{i,l})) = \mathcal{A}(t-s_{i,l},\varepsilon').$$

Yet, Example 2.7 and 2.8 present non-linear envelopes.

Rizk and Fidler [**132**] approximate the original envelope by an envelope of the form $\rho t + \varepsilon$ and the transformation works as before.

For the envelope $\mathcal{A}(t,\varepsilon) = M(1-e^{-\lambda t})e^{\theta\epsilon}$ of Example 2.8 it holds

$$\mathcal{A}(t-s_{i,u},\epsilon) = \mathcal{A}(t-s_{i,l},\epsilon)\frac{1-e^{-\lambda(t-s_{i,u})}}{1-e^{-\lambda(t-s_{i,l})}} = \mathcal{A}(t-s_{i,l},\epsilon'),$$

where $\epsilon' = \epsilon + {}^{1}/_{\lambda}(\log(1-e^{-\lambda(t-s_{i,u})}) - \log(1-e^{-\lambda(t-s_{i,l})}))$.

Crucial for the above arguments is that $\varepsilon' < \varepsilon$ and $\epsilon' < \epsilon$ holds. As the allowed deviation from the arrival curve decreases, the corresponding violation probabilities increase. In both cases a larger difference of $s_{i,u}$ to $s_{i,l}$ leads to larger violation probabilities of the respective terms. This reveals a tradeoff between a partition's cardinality and the lengths of its sets. An exhaustive study of how to find a good or optimal partition has not been addressed, yet.

Remark 2.19. As stated in Remark 2.15, it is beneficial to have $\sup_{t\geq 0}\bar{\eta}(t,\bar{\varepsilon}) < \infty$. However, if one uses the tailbounds as constructed by effective bandwidths (see Example 2.4), the resulting error function $\bar{\eta}$ is unbounded in t. To avoid this, the envelope $\mathcal{A}(t,\varepsilon) = \rho t + \varepsilon$ (for some $\rho > 0$) is relaxed by a slack rate $\delta > 0$. This results in a slightly worse arrival curve $\mathcal{A}$, but also in smaller violation probabilities. To accomplish finite error probabilities one replaces the original $\mathcal{A}(t,\varepsilon)$ in the above theorem by $\mathcal{A}(t,\varepsilon+\delta t)$. Indeed, if $A \preceq (\mathcal{A},\eta)$, then it holds also

$$\mathbb{P}(A(s,t) > \rho(t-s) + \varepsilon + \delta(t-s)) \leq \eta(t-s,\varepsilon+\delta(t-s)).$$

Repeating the proof of Theorem 2.16 leads to a violation probability

$$\bar{\eta}(t,\bar{\varepsilon}) = \sum_{i\in I} \eta(t-s_{i,l},\varepsilon-(\rho+\delta)(s_{i,u}-s_{i,l})+\delta(t-s_{i,l})).$$

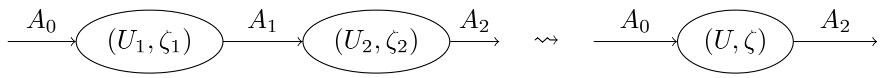


FIGURE 2.4. The concatenation theorem for two tailbounded service elements.

Since the errors decrease in their second variable, the sum is convergent. For example in the EBB model with canonical discretization of size τ the error $\bar{\eta}$ fulfills

$$\bar{\eta}(t, \bar{\varepsilon}) \xrightarrow{t \to \infty} M e^{-\theta\varepsilon}(e^{-\theta\rho\tau} - e^{-\theta\tau(\rho+\delta)})^{-1}.$$

This construction uses again the linear form of the envelope $\mathcal{A}$. For the non-linear envelope of Example 2.8 the envelope $\mathcal{A}(t, \epsilon) = M(1 - e^{-\lambda t})e^{\theta\epsilon}$ is modified to

$$\mathbb{P}(A(s,t) > M(1 - e^{-\lambda t})e^{\theta(\epsilon + \delta t)}) \leq \frac{\rho}{\lambda} e^{-\theta\epsilon + \delta t}$$

and $\bar{\eta}(t, \bar{\epsilon})$ converges for $t \to \infty$.

The last of the network operations to be shown is the concatenation property realized by the operator $\otimes$. A concatenation property for tailbounds has been subject of intense research [**6, 31, 106**]. The search for service descriptions that fulfill the concatenation and produce finite errors ended with the work of Ciucu et al. [**40**]. The main idea is presented now for the special case $\mathbb{K} = \mathbb{N}_0$. For continuous time a partitioning step is needed similar to the one in the theorem before.

Like for arrivals' tailbounds the main idea is to introduce a slack rate δ. To this end, the service curve must have a linear structure in ε. Hence, consider the tailbounded service curve

$$\mathbb{P}(B(t) < A \otimes [\mathcal{U} - \varepsilon]^{+}(t)) < \zeta(t, \varepsilon),$$

where the convolution reads

$$A \otimes [\mathcal{U} - \varepsilon]^{+}(t) = \min_{0 \leq s \leq t} \{A(s) + [\mathcal{U}(t-s) - \varepsilon]^{+}\}$$

for some $\mathcal{U} \in \mathcal{F}$. This is a special case of Definition 2.11 with $U(t, \varepsilon) := [\mathcal{U}(t) - \varepsilon]^{+}$. The relaxation by δ is defined by $\mathcal{U}_{-\delta}(t) := \mathcal{U}(t) - \delta t$.

THEOREM 2.20 (Tailbounded Convolution). *Assume that two service elements in tandem have tailbounded service curves (U_1, ζ_1) and (U_2, ζ_2), respectively (see Figure 2.4). Then the whole system offers a service curve (U, ζ) with*

$$U(t, \varepsilon) = [\mathcal{U}_1 \otimes \mathcal{U}_{2,-\delta}(t) - \varepsilon]^{+}, \qquad \zeta(t, \varepsilon) = \zeta_2(t, \cdot) \otimes \bar{\zeta}_1(t, \cdot)(\varepsilon)$$

and

$$\bar{\zeta}_1(t, \varepsilon) = \sum_{s=0}^{t} \zeta_1(s, \varepsilon + \delta(t-s)).$$

PROOF. Fix some $t \in \mathbb{N}_0$, $\delta > 0$, and $\varepsilon_1, \varepsilon_2, \varepsilon$ such that $\varepsilon_1 + \varepsilon_2 = \varepsilon$. Assume that

$$A_1(s) \geq A \otimes [\mathcal{U}_1 - \varepsilon_1 - \delta(t-s)]^{+}(s) \qquad \text{for all } s \leq t \tag{2.8}$$

$$A_2(t) \geq A_1 \otimes [\mathcal{U}_2 - \varepsilon_2]^{+}(t), \tag{2.9}$$

holds. Then it follows

$$\begin{aligned}A_2(t) &\geq \min_{0\leq s\leq t}\{A_1(s) + [\mathcal{U}_2(t-s) - \varepsilon_2]^+\}\\ &\geq \min_{0\leq s\leq t}\{\min_{0\leq r\leq s}\{A(r) + [\mathcal{U}_1(s-r) - \varepsilon_1 - \delta(t-s)]^+ + [\mathcal{U}_2(t-s) - \varepsilon_2]^+\}\\ &\geq \min_{0\leq r\leq s\leq t}\{A(r) + [\mathcal{U}_1(s-r) + \mathcal{U}_2(t-s) - \varepsilon - \delta(t-s)]^+\}\\ &= A \otimes [\mathcal{U}_1 \otimes \mathcal{U}_{2,-\delta} - \varepsilon]^+(t).\end{aligned}$$

Moving this implication to probabilities yields

$$\begin{aligned}&\mathbb{P}(A(t) < A_2 \otimes U(t,\varepsilon))\\ \leq\, &\mathbb{P}(\text{Condition (2.9) does not hold}) + \mathbb{P}(\text{Condition (2.8) does not hold})\\ \leq\, &\zeta_2(t,\varepsilon_2) + \sum_{s=0}^{t}\zeta_1(s,\varepsilon_1 + \delta(t-s)).\end{aligned}$$

Since this bound holds for all $\varepsilon_1 + \varepsilon_2 = \varepsilon$ one has

$$\mathbb{P}(A(t) < A_2 \otimes U(t,\varepsilon)) \leq \inf_{0\leq\varepsilon'\leq\varepsilon}\{\zeta_2(t,\varepsilon - \varepsilon') + \bar{\zeta}_1(t,\varepsilon')\} = \zeta_2(t,\cdot) \otimes \bar{\zeta}_1(t,\cdot)(\varepsilon).$$

□

EXAMPLE 2.21. Assume that the two service elements are tailbounded for some $U_i(t,\varepsilon) = [\mathcal{U}_1(t) - \varepsilon]^+$ with EBB errors:

$$\zeta_i(t,\varepsilon) = M_i e^{-\theta_i\varepsilon}.$$

Then

$$\begin{aligned}\sup_{t\geq 0}\zeta(t,\varepsilon) = \sup_{t\geq 0}\sum_{s=0}^{t}\zeta_1(s,\varepsilon + \delta(t-s)) &= M_1 e^{-\theta_1\varepsilon}\sum_{s'=0}^{\infty} e^{-\theta_1\delta s'}\\ &= M_1 e^{-\theta_1\varepsilon}(1 - e^{-\theta_1\delta})^{-1} < \infty\end{aligned}$$

as wanted.

With Theorem 2.20 the construction of a tailbounded network calculus is finished. In order to apply this calculus to a queueing system, all left to do is to prove the respective performance bounds.

2.2. Performance Bounds in Tailbounded Network Calculus

This section derives stochastic performance bounds and is hence analog to Section 1.3. The main idea of the next theorem – a tailbounded version of Theorem 1.18 – is the following: After formulating a suitable set of assumptions the results of deterministic network calculus are applied. Afterwards one considers the probability that the assumptions are not fulfilled.

THEOREM 2.22 (Performance Guarantees with Tailbounds). *Consider the arrivals $A \preceq (\mathcal{A},\eta)$ with $\mathcal{A}(t,\varepsilon) = \rho t + \varepsilon$ and assume that A enters a service element with service curve (U,ζ). Denote the output of the service element by B. Then for all $t \geq 0$ and $\varepsilon > 0$ the performance bounds*

$$\mathbb{P}(\mathfrak{b}(t) > \mathcal{A}(\varepsilon) \oslash U(\varepsilon)(0)) \leq \bar{\eta}(t,\bar{\varepsilon}) + \zeta(t,\varepsilon) \tag{2.10}$$

$$\mathbb{P}(\mathfrak{d}(t) > \inf\{s \geq 0 : \mathcal{A}(\varepsilon) \oslash U(\varepsilon)(-s) \leq 0\}) \leq \bar{\eta}(t,\bar{\varepsilon}) + \zeta(t,\varepsilon) \tag{2.11}$$

hold, where $\bar{\eta}$ depends on η.

PROOF. Under the two assumptions

$$A(s,t) \leq \mathcal{A}(t-s,\varepsilon) \qquad \text{for all } s \leq t \tag{2.12}$$

$$B(t) \geq A \otimes U(\varepsilon)(t) \tag{2.13}$$

the arrivals and the service element are deterministically bounded. Theorem 1.18 gives

$$\begin{aligned} \mathfrak{b}(t) &\leq \mathcal{A}(\varepsilon) \oslash U(\varepsilon)(0) \\ \mathfrak{d}(t) &\leq \inf\{s \geq 0 \,:\, \mathcal{A}(\varepsilon) \oslash U(\varepsilon)(-s) \leq 0\}. \end{aligned}$$

Moving to the complementary events yields for the backlog bound

$$\begin{aligned} \mathbb{P}(\mathfrak{b}(t) > & \mathcal{A}(\varepsilon) \oslash U(\varepsilon)(0)) \leq \mathbb{P}((2.12) \text{ or } (2.13) \text{ does not hold}) \\ &\leq \mathbb{P}\Big(\bigcup_{s\leq t} A(s,t) > \mathcal{A}(t-s,\varepsilon)\Big) + \mathbb{P}(B(t) > A \otimes U(\varepsilon)(t)) \\ &\leq \mathbb{P}\Big(\bigcup_{s\leq t} A(s,t) > \mathcal{A}(t-s,\varepsilon)\Big) + \zeta(t,\varepsilon). \end{aligned}$$

The same discretization technique as in Theorem 2.16 delivers eventually

$$\mathbb{P}\Big(\bigcup_{s\leq t} A(s,t) > \mathcal{A}(t-s,\varepsilon)\Big) \leq \sum_{i\in I} \eta(t-s_{i,l}, \varepsilon - \rho(s_{i,u}-s_{i,l})) =: \bar{\eta}(t,\bar{\varepsilon}).$$

The proof for the delay bounds works the same way. □

REMARK 2.23. The parameter ε in the assumptions (2.12) and (2.13) does not need to be the same. By choosing different values one can optimize the performance bounds. An optimization along this way had not been performed in the literature, yet.

This closes the introduction to tailbounded network calculus.

2.3. Network Operations in MGF-Based Network Calculus

The most common way to achieve tailbounds on some random variable X are moment-bounds. Markov's inequality is an example for such a bound. It states

$$\mathbb{P}(X > x) \leq \frac{\mathbb{E}(X)}{x}$$

for a positive random variable and all $x > 0$. The idea of MGF-based network calculus is to work with such moment-bounds directly – without introducing error functions. A particular useful bound is Chernoff's inequality. It results from Markov's inequality by taking exponents:

$$\mathbb{P}(e^{\theta X} > e^{\theta x}) \leq e^{-\theta x}\mathbb{E}(e^{\theta X}) = e^{-\theta x}\phi_X(\theta),$$

where $\phi_X(\theta)$ is the moment generating function (MGF) of X at θ.

In the previous section the arrivals and service descriptions are – as in deterministic calculus – interval-valid. This is best seen in the arrival curve $A(s,t) \leq \mathcal{A}(t-s)$ and its tailbounded version

$$\mathbb{P}(A(s,t) > \mathcal{A}(t-s,\varepsilon)) \leq \eta(t-s,\varepsilon).$$

From an arrival curve's perspective, there is no difference in considering $A(s,t)$ or arrivals in some other interval, like $A(s+r,t+r)$. Possible differences in their distribution are hence not captured. With its bivariate formulation MGF-based calculus can (theoretically) also take non-stationarities into account. This possibility was recently exploited in the work of Becker and Fidler [**17**] (see also the remarks to Definition 2.28). As seen later this potential bivariate formulation is discarded when bounding the MGFs of an arrival or service.

Nevertheless, the MGF-based network calculus is based upon a bivariate formulation [**35**] and with respect to that differs to tailbounded approaches already. This is best seen in Chapter 6. This bivariate formulation demands a redefinition of a service element as well as the operators $\otimes$ and $\oslash$.

DEFINITION 2.24. A service element is a *dynamic U-server*, if for any input-output pair A and B it holds

$$B(t) \geq A \otimes U(0,t) \qquad \text{for all } t \in \mathbb{K}.$$

The bivariate convolution used here is defined as follows.

DEFINITION 2.25. The *bivariate convolution operator* $\otimes$ and *bivariate deconvolution operator* $\oslash$ are defined by

$$\begin{aligned}
\otimes &: \tilde{\mathcal{F}} \times \tilde{\mathcal{F}} \to \tilde{\mathcal{F}} \\
&A \otimes B(s,t) \mapsto \inf_{s \leq r \leq t}\{A(s,r) + B(r,t)\}, \\
\oslash &: \tilde{\mathcal{F}} \times \tilde{\mathcal{F}} \to \tilde{\mathcal{F}} \\
&A \oslash B(s,t) \mapsto \sup_{0 \leq r \leq s}\{A(r,t) - B(r,s)\}.
\end{aligned}$$

The bivariate convolution is – in contrast to the univariate convolution – not commutative. The univariate convolution recurs, if $A(s,t) = A(t-s)$ and $B(s,t) = B(t-s)$ hold for all $(s,t) \in \Lambda$. Indeed, in this case $A \otimes B(s,t) = A \otimes B(t-s)$ and the univariate convolution is the restriction of the bivariate convolution to a subset of $\tilde{\mathcal{F}}$.

However, the univariate deconvolution does not recur in such a way from the bivariate deconvolution. Its form is rather motivated by the bivariate version of the fundamental theorem of network calculus. Indeed, applying the same steps as in Theorem 1.18 leads to

$$\mathfrak{b}(t) \leq \sup_{0 \leq r \leq t}\{A(r,t) - U(r,t)\} = A \oslash U(t,t). \tag{2.14}$$

The choice of Chernoff's inequality under the possible moment-bounds is justified by its properties when combined with the network operations. The following result [**33, 35, 65**] is central for MGF-based calculus.

THEOREM 2.26 (MGFs of Min-Plus Operators). *Let X, Y be two stochastic processes indexed by Λ; further, assume that X and Y are non-increasing in their first variable and stochastically independent. Then*

$$\phi_{X \oslash Y(s,t)}(\theta) \leq \sum_{i \in I} \phi_{X(s_{i,l},t)}(\theta)\phi_{Y(s_{i,u},s)}(-\theta), \tag{2.15}$$

$$\phi_{X \otimes Y(s,t)}(-\theta) \leq \sum_{i \in I} \phi_{X(s,s_{i,l})}(-\theta)\phi_{Y(s_{i,u},t)}(-\theta) \tag{2.16}$$

hold for all $(s,t) \in \Lambda$ and $\theta > 0$ such that the above MGFs exist. Here $s_{i,l}$ and $s_{i,u}$ are the left and right endpoints of a partition $(T_i)_{i \in I}$ of $[0,s]$ (see the proof of Theorem 2.22).

PROOF. Fix some $\theta > 0$ and $(s,t) \in \Lambda$ such that the MGFs in (2.15) exist. Consider some partition of $[0,s]$ as in the proof of Theorem 2.22. Then it holds

$$\begin{aligned} \phi_{X \oslash Y(s,t)}(\theta) &= \mathbb{E}(e^{\theta \sup_{0 \leq r \leq s}\{X(r,t)-Y(r,s)\}}) = \mathbb{E}(\sup_{0 \leq r \leq s}\{e^{\theta X(r,t)-\theta Y(r,s)}\}) \\ &= \mathbb{E}(\max_{i \in I} \sup_{r \in T_i}\{e^{\theta X(r,t)-\theta Y(r,s)}\}) \leq \mathbb{E}(\max_{i \in I}\{e^{\theta X(s_{i,l},t)-\theta Y(s_{i,u},s)}\}) \qquad (2.17) \\ &\leq \sum_{i \in I} \mathbb{E}(e^{\theta X(s_{i,l},t)} e^{-\theta Y(s_{i,u},s)}). \end{aligned}$$

The expectations appearing in the sum can be split due to X and Y being independent. This concludes the proof of (2.15). Inequality (2.16) is proven in the same way. □

REMARK 2.27. The above theorem works for expressions of the form $A \oslash U(t,s)$ with $(s,t) \in \Lambda$ also. Indeed the relation of the variables s and t to each other is of no importance for the proof.

Instead of dealing with the MGFs of A and U directly, they are bounded to simplify further calculations.

DEFINITION 2.28. A flow A is *MGF-bounded by a function f* for some $\theta > 0$, if

$$\phi_{A(s,t)}(\theta) = \mathbb{E}(e^{\theta A(s,t)}) \leq e^{\theta f(t-s,\theta)}$$

holds for all $(s,t) \in \Lambda$.

DEFINITION 2.29. A dynamic U-server is *MGF-bounded by a function f* for some $\theta > 0$, if

$$\phi_{U(s,t)}(-\theta) = \mathbb{E}(e^{-\theta U(s,t)}) \leq e^{\theta f(t-s,\theta)}$$

holds for all $(s,t) \in \Lambda$.

REMARK 2.30. As stated before, the possibility to bound $\mathbb{E}(e^{\theta A(s,t)})$ and $\mathbb{E}(e^{\theta A(s+r,t+r)})$ differently is not utilized here.

EXAMPLE 2.31. The server with constant rate c is MGF-bounded; indeed, for any interval $(s,t]$ it holds $\mathbb{E}(e^{-\theta U(s,t)}) = e^{-\theta c \cdot (t-s)}$.

EXAMPLE 2.32. Let $\mathbb{K} = \mathbb{N}_0$ and assume that the increments are described by i.i.d. random variables X_t. Then $\phi_{A(s,t)}(\theta) = \prod_{r=s+1}^{t} \phi_{X_r}(\theta) = (\phi_{X_r}(\theta))^{t-s}$. As a special case let X_t be exponentially distributed with parameter λ. Then its MGF is $\frac{\lambda}{\lambda-\theta}$ for all $\theta < \lambda$ and $\phi_{A(s,t)}(\theta) = (\frac{\lambda}{\lambda-\theta})^{t-s}$.

Example 2.4 presents a way to construct tailbounded arrival curves from effective bandwidths, via moment generating functions. This reveals a connection between both methods for bounding arrivals (or service). See Appendix 10.1 for how to transform tailbounds into MGF-bounds and vice versa.

REMARK 2.33. This thesis focuses on linear $f(t,\theta) = \theta\rho(\theta)t + \theta\sigma(\theta)$ that define the $\sigma(\theta), \rho(\theta)$-calculus and use the notations

$$\begin{aligned} A \preceq (\sigma,\rho) \quad &:\Leftrightarrow \quad \phi_{A(s,t)}(\theta) \leq e^{\theta\sigma(\theta)+\theta\rho(\theta)(t-s)} \\ U \succeq (\sigma,\rho) \quad &:\Leftrightarrow \quad \phi_{U(s,t)}(-\theta) \leq e^{\theta\sigma(\theta)+\theta\rho(\theta)(t-s)}. \end{aligned}$$

The θ in this notation is omitted. This should not be confounded with the (σ,ρ)-calculus of Chang [**35**], which is a formulation of deterministic network calculus.

The following lemma reformulates Theorem 2.26 by adding MGF-bounds. It is also central in the derivation of performance bounds in the following section.

LEMMA 2.34. *Fix some $\theta > 0$. Let $A \preceq (\sigma_A, \rho_A)$ and $U \succeq (\sigma_U, \rho_U)$ be stochastically independent. For all $(s,t) \in \Lambda$ and canonical discretizations of $[0,s]$ with length τ it holds*

$$\phi_{A\oslash U(s,t)}(\theta) \leq e^{\theta(\sigma_A(\theta)+\sigma_U(\theta))+\theta\rho_A(\theta)\tau} e^{\theta\rho_A(\theta)(t-s)} \sum_{k=0}^{\frac{s}{\tau}-1} e^{\theta(\rho_A(\theta)+\rho_U(\theta))k\tau} \tag{2.18}$$

$$= e^{\theta(\sigma_A(\theta)+\sigma_U(\theta))-\theta\rho_U(\theta)\tau} e^{\theta\rho_U(\theta)(s-t)} \sum_{k=0}^{\frac{s}{\tau}-1} e^{\theta(\rho_A(\theta)+\rho_U(\theta))(t-k\tau)}, \tag{2.19}$$

if the above MGFs exist.

Under the same assumptions it holds for slotted time

$$\begin{aligned}
\phi_{A\oslash U(s,t)}(\theta) &\leq e^{\theta(\sigma_A(\theta)+\sigma_U(\theta))} e^{\theta\rho_A(\theta)(t-s)} \sum_{r=0}^{s} e^{\theta(\rho_A(\theta)+\rho_U(\theta))r} \\
&= e^{\theta(\sigma_A(\theta)+\sigma_U(\theta))} e^{\theta\rho_U(\theta)(s-t)} \sum_{r=0}^{s} e^{\theta(\rho_A(\theta)+\rho_U(\theta))(t-r)}.
\end{aligned}$$

PROOF. From Theorem 2.26 it follows

$$\begin{aligned}
\phi_{A\oslash U(s,t)}(\theta) &\leq \sum_{i\in I} \mathbb{E}(e^{\theta A(s_{i,l},t)})\mathbb{E}(e^{-\theta U(s_{i,u},s)}) \\
&\leq e^{\theta(\sigma_A(\theta)+\sigma_U(\theta))} \sum_{i\in I} e^{\theta\rho_A(\theta)(t-s_{i,l})} e^{\theta\rho_U(\theta)(s-s_{i,u})} \\
&= e^{\theta(\sigma_A(\theta)+\sigma_U(\theta))} e^{\theta\rho_A(\theta)(t-s)} \sum_{i\in I} e^{\theta(\rho_A(\theta)+\rho_U(\theta))(s-s_{i,u})+\theta\rho_A(\theta)(s_{i,u}-s_{i,l})} \\
&= e^{\theta(\sigma_A(\theta)+\sigma_U(\theta)+\rho_A(\theta)\tau)} e^{\theta\rho_A(\theta)(t-s)} \sum_{k=0}^{\frac{s}{\tau}-1} e^{\theta(\rho_A(\theta)+\rho_U(\theta))k\tau}.
\end{aligned}$$

The equality to (2.19) is achieved by factoring out $e^{\theta(\rho_A(\theta)+\rho_U(\theta))(s-t-\tau)}$ and reindexing the sum.

For the special case of slotted time set $\tau = 1$ in the canonical discretization and with $A(s_{i,l}) = \lim_{r\nearrow s_{i,u}} A(r)$ the supremum in (2.17) is discretized to a maximum over the integers. Hence, Theorem 2.26 simplifies to

$$\phi_{A\oslash U(s,t)}(\theta) \leq \sum_{k=0}^{s} \mathbb{E}(e^{\theta A(k,t)})\mathbb{E}(e^{-\theta U(k,s)}).$$

The rest of the proof follows as before. □

Under the assumption of stochastic independence Chernoff's inequality and MGF-bounds lead to a full network calculus. To show this the network operations $\oplus, \otimes, \ominus$, and $\oslash$ must be transferred.

THEOREM 2.35 (MGF-Multiplexing). *Fix some $\theta > 0$. Consider two flows $A_1 \preceq (\sigma_1, \rho_1)$ and $A_2 \preceq (\sigma_2, \rho_2)$. If A_1 and A_2 are stochastically independent,*

$$A_1 \oplus A_2 \preceq (\sigma_1 + \sigma_2, \rho_1 + \rho_2).$$

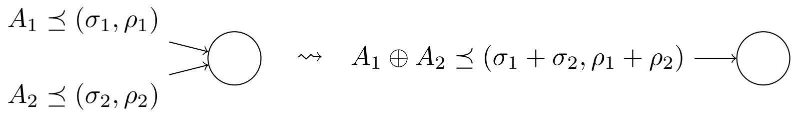


FIGURE 2.5. MGF-bounded version of the multiplexing operator.

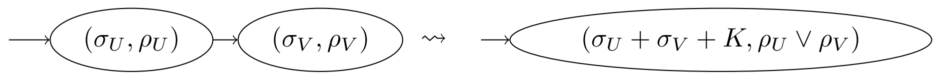


FIGURE 2.6. MGF-bounded version of the convolution operation.

PROOF. The moment generating function of a sum of independent random variables is the product of their moment generating functions. Hence, it holds

$$\phi_{A_1+A_2(s,t)}(\theta) = \phi_{A_1(s,t)}(\theta)\phi_{A_2(s,t)}(\theta) \leq e^{\theta(\sigma_1(\theta)+\rho_1(\theta)(t-s))}e^{\theta(\sigma_2(\theta)+\rho_2(\theta)(t-s))}$$

for all $(s,t) \in \Lambda$. $\square$

A flow's MGF-bound holds for all θ with $\theta < \theta^* \in [0, \infty]$. Hence, if flows A_1 and A_2 provide different $\theta^*_{A_1}$ and $\theta^*_{A_2}$, the bound for the aggregate holds only for $\theta < \theta^*_{A_1} \wedge \theta^*_{A_2}$. Similar considerations hold for the following three network operations.

THEOREM 2.36 (MGF-Convolution). *Fix some $\theta > 0$. Assume that two service elements $U \succeq (\sigma_U, \rho_U)$ and $V \succeq (\sigma_V, \rho_V)$ are stochastically independent. If U and V are decreasing in their first variable, it holds*

$$U \otimes V \succeq (\sigma_U + \sigma_V + K, \rho_U \vee \rho_V)$$

for some constant K dependent on θ, ρ_U, ρ_V.

PROOF. The proof uses a canonical discretization of the interval $[s,t]$ with length τ. By the same steps as in Lemma 2.34 it follows that

$$\begin{aligned} \phi_{U\otimes V(s,t)}(-\theta) \leq & \sum_{i\in I} e^{\theta(\sigma_U(\theta)+\rho_U(\theta)(s_{i,l}-s))}e^{\theta(\sigma_V(\theta)+\rho_V(\theta)(t-s_{i,u}))} \\ =& e^{\theta(\sigma_U(\theta)+\sigma_V(\theta))}e^{\theta\rho_U(\theta)(t-s)}e^{-\theta\rho_U\tau}\sum_{i\in I} e^{\theta(\rho_V(\theta)-\rho_U(\theta))(t-s_{i,u})}. \end{aligned} \tag{2.20}$$

If $\rho_U(\theta) \geq \rho_V(\theta)$, the constant K is given by

$$K := e^{-\theta\rho_U(\theta)\tau}\sum_{i\in I} e^{\theta(\rho_V(\theta)-\rho_U(\theta))(t-s_{i,u})}.$$

If $\rho_V(\theta) \geq \rho_U(\theta)$, one factors out $e^{\theta\rho_V(\theta)(t-s)}e^{-\theta\rho_V(\theta)\tau}$ in (2.20) instead and the constant becomes $K := e^{-\theta\rho_V(\theta)\tau}\sum_{i\in I} e^{\theta(\rho_V(\theta)-\rho_U(\theta))(s-s_{i,l})}$. $\square$

THEOREM 2.37 (MGF-Leftover-Serivce). *Fix some $\theta > 0$. Assume that $A_i \preceq (\sigma_i, \rho_i)$ for all $i = 1, \ldots, n$ and a service element $U \succeq (\sigma_U, \rho_U)$ for the aggregate. Let the flows be indexed by their priority at U such that a higher index corresponds to a higher priority. If $A_1, \ldots, A_n, U$ are stochastically independent, the system is a dynamic U_i-server for a particular flow A_i with*

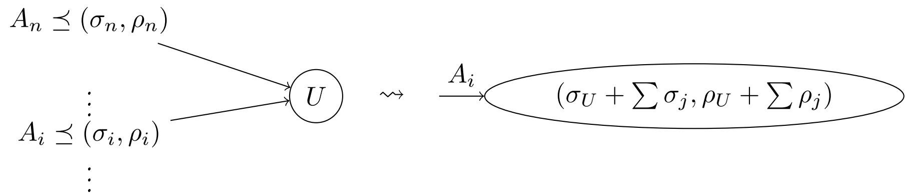


FIGURE 2.7. MGF-bounded version of the subtraction operator.

$$U_i(s,t) := U(s,t) - \sum_{j=i+1}^{n} A_j(s,t)$$

and

$$U_i \succeq (\sigma_U + \sum_{j=i+1}^{n} \sigma_j, \rho_U + \sum_{j=i+1}^{n} \rho_j). \tag{2.21}$$

PROOF. Fix some arbitrary $t \in \mathbb{K}$ and index $i \in \{1, \ldots, n\}$. As the system is a dynamic U-server for the aggregate, it holds

$$\sum_{j=i}^{n} B_j(t) \geq \sum_{j=i}^{n} A_j \otimes U(0,t) = \inf_{0 \leq s \leq t} \{\sum_{j=i}^{n} A_j(0,s) + U(s,t)\}.$$

Following

$$\begin{aligned} B_i(t) &\geq \inf_{0 \leq s \leq t} \{A_i(0,s) + U(s,t) + \sum_{j=i+1}^{n} A_j(0,s) - B_j(t)\} \\ &\geq \inf_{0 \leq s \leq t} \{A_i(0,s) + U(s,t) + \sum_{j=i+1}^{n} A_j(s) - A_j(t)\} = A_i \otimes U_i(0,t). \end{aligned}$$

It is left to show that U_i fulfills Equation (2.21). Let $(s,t) \in \Lambda$ be arbitrary and θ such that the following MGF exist. Due to the independence of the involved random variables it holds

$$\begin{aligned} \phi_{U_i(s,t)}(-\theta) &= \phi_{U(s,t)}(-\theta) \prod_{j=i+1}^{n} \phi_{-A_j(s,t)}(-\theta) = \phi_{U(s,t)}(-\theta) \prod_{j=i+1}^{n} \phi_{A_j(s,t)}(\theta) \\ &\leq e^{\theta\sigma_U(\theta) + \theta\rho_U(\theta)(t-s)} \prod_{j=i+1}^{n} e^{\theta\sigma_j(\theta) + \theta\rho_j(\theta)(t-s)} \\ &= e^{\theta(\sigma_U(\theta) + \sum_{j=i+1}^{n} \sigma_j) + \theta(\rho_U(\theta) + \sum_{j=i+1}^{n} \rho_j(\theta))(t-s)}. \end{aligned}$$

□

THEOREM 2.38 (MGF-Output-Bound). *Fix some $\theta > 0$. Assume that a flow $A \preceq (\sigma_A, \rho_A)$ enters service element $U \succeq (\sigma_U, \rho_U)$ and U is decreasing in its first variable. If A and U are stochastically independent and $\rho_A(\theta) \leq -\rho_U(\theta)$, it holds*

$$B \preceq (\sigma_A + \sigma_U + K, \rho_A), \tag{2.22}$$

where K is a constant depending on $\rho_A(\theta)$, $\rho_U(\theta)$, θ.

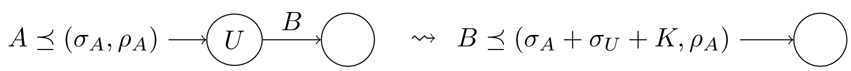


FIGURE 2.8. MGF-bounded version of the deconvolution operator.

PROOF. From the definition of the dynamic U-server it follows

$$\begin{aligned} B(s,t) = B(t) - B(s) \leq A(t) - A \otimes U(0,s) &= \sup_{0 \leq r \leq s} \{A(r,t) - U(r,s)\} \\ &= A \oslash U(s,t) \end{aligned}$$

Lemma 2.34 applies and leads to

$$\begin{aligned} \phi_{B(s,t)}(\theta) &\leq \phi_{A \oslash U(s,t)}(\theta) \\ &= e^{\theta(\sigma_A(\theta)+\sigma_U(\theta)+\rho_A(\theta)\tau)} e^{\theta\rho_A(\theta)(t-s)} \sum_{k=0}^{\frac{s}{\tau}-1} e^{\theta(\rho_A(\theta)+\rho_U(\theta))k\tau} \\ &\leq e^{\theta(\sigma_A(\theta)+\sigma_U(\theta)+\rho_A(\theta)\tau)} e^{\theta\rho_A(\theta)(t-s)} (1 - e^{\theta\tau(\rho_A(\theta)+\rho_U(\theta))})^{-1} \end{aligned}$$

for some canonical discretization with length τ. Defining

$$K = e^{\theta\rho_A(\theta)\tau} (1 - e^{\theta\tau(\rho_A(\theta)+\rho_U(\theta))})^{-1}$$

finishes the proof. $\square$

2.4. Performance Bounds in MGF-Based Network Calculus

Combining MGF-bounds with Lemma 2.34 yields stochastic performance bounds.

THEOREM 2.39 (Performance Guarantees with MGF-Bounds). *Fix some $\theta > 0$. Assume that $A \preceq (\sigma_A, \rho_A)$ and a service $U \succeq (\sigma_U, \rho_U)$ that is decreasing in its first variable with $U \geq 0$. If A and U are stochastically independent, then the performance bounds*

$$\mathbb{P}(\mathfrak{b}(t) > N) \leq e^{-\theta N} e^{\theta(\sigma_A(\theta)+\sigma_U(\theta)+\rho_A(\theta)\tau)} \sum_{k=0}^{\frac{t}{\tau}-1} e^{\theta(\rho_A(\theta)+\rho_U(\theta))k\tau} \tag{2.23}$$

$$\mathbb{P}(\mathfrak{d}(t) > T) \leq e^{\theta\rho_U(\theta)T} e^{\theta(\sigma_A(\theta)+\sigma_U(\theta)+\rho_U(\theta)\tau)} \sum_{k=0}^{\frac{t}{\tau}-1} e^{\theta(\rho_A(\theta)+\rho_U(\theta))(t-k\tau)} \tag{2.24}$$

hold for all $t \geq 0$. The free parameter $\tau > 0$ is the width of the canonical discretization of t (of $t + T$ in (2.24)).

PROOF. The proof is similar to the previous one. From Equation (2.14) follows the implication

$$\mathfrak{b}(t) > N \quad \Rightarrow \quad A \oslash U(t,t) > N.$$

Moving to probabilities and applying Chernoff's inequality leads to

$$\mathbb{P}(\mathfrak{b}(t) > N) \leq \mathbb{P}(A \oslash U(t,t) > N) \leq e^{-\theta N} \mathbb{E}(e^{\theta A \oslash U(t,t)}).$$

Lemma 2.34 applies to this expression and leads to (2.23).

To achieve (2.24) the steps of deriving delay bounds must be transferred to the bivariate setting (compare Theorem 1.18). To that end, fix some $T \in \mathbb{K}$ and assume that there is a $t \in \mathbb{K}$ such that $\mathfrak{d}(t) > T$. Then it follows

$$\begin{aligned}0 &< A(0,t) - B(t+T) \leq A(0,t) - A \otimes U(0,t+T) \\ &= \sup_{0 \leq r \leq t} \{A(r,t) - U(r,T+t)\} \vee \sup_{t < r \leq t+T} \{A(r,t) - U(r,T+t)\}.\end{aligned}$$

As the latter supremum is less than or equal to zero, it can be discarded. This leads to the implication

$$\mathfrak{d}(t) > T \quad \Rightarrow \quad (-U) \oslash (-A)(t,t+T) > 0.$$

Moving this statement to probabilities and applying Chernoff's inequality leads to

$$\mathbb{P}(\mathfrak{d}(t) > T) \leq \mathbb{P}((-U) \oslash (-A)(t,t+T) < 0) \leq \mathbb{E}(e^{\theta(-U)\oslash(-A)(t,t+T)}).$$

Note that $U \succeq (\sigma_U, \rho_U)$ if and only if $-U \preceq (\sigma_U, \rho_U)$ and $A \preceq (\sigma_A, \rho_A)$ if and only if $-A \succeq (\sigma_A, \rho_A)$. Hence, Lemma 2.34 applies and leads to Equation (2.24). $\square$

REMARK 2.40. For slotted time the second part of Lemma 2.34 applies and yields

$$\mathbb{P}(\mathfrak{b}(t) > N) \leq e^{-\theta N} e^{\theta(\sigma_A(\theta)+\sigma_U(\theta))} \sum_{s=0}^{t} e^{\theta(\rho_A(\theta)+\rho_U(\theta))s}, \tag{2.25}$$

$$\mathbb{P}(\mathfrak{d}(t) > T) \leq e^{\theta\rho_U(\theta)T} e^{\theta(\sigma_A(\theta)+\sigma_U(\theta))} \sum_{s=0}^{t+T} e^{\theta(\rho_A(\theta)+\rho_U(\theta))s}. \tag{2.26}$$

In slotted time U does not need to decrease in its first variable, as a discretization is not needed. This is of particular importance, as there are cases in which U is *not* decreasing in its first variable. An example is the leftover service description resulting from Theorem 2.37.

This concludes the construction of an MGF-based network calculus. Results for the cases in which the involved processes are not stochastically independent is deliberately left out at this place. They are given in the next chapter, where end-to-end performance bounds are considered.

Part 2

Advances in Stochastic Network Calculus

CHAPTER 3

The Problem of End-to-End Analysis

Subsection 1.2.1 shows how network operations reduce general feedforward networks. The previous chapter reconstructed these network operations in a stochastic setting; hence, a reduction of networks also becomes possible in SNC. Doing so in practice requires some care. Further, the results of this analysis are involved functions of several variables. Only after a subsequent, non-trivial optimization, one can obtain useful results.

The special case of tandems, as presented in Figure 3.1, was analyzed in the SNC-literature: Fidler considers such tandems in [**65**] under the assumption of stochastic independence between the flows. The MGF-based approach delivers stochastic delay bounds that scale with $\mathcal{O}(n)$. Here n is the number of nodes traversed. The same scaling behavior appears in deterministic network calculus [**102**]. Ciucu et al. use a tailbounded approach for tandem networks in [**39**]. The tailbounded approach has the advantage that the assumption of stochastic independence can be dropped. The bounds in [**39**] scale with $\mathcal{O}(n \log n)$. Further, [**30**] provides a lower delay bound for tandem networks that scales with $\Theta(n \log n)$. This is not in contrast with the work of Fidler [**65**], as the results in [**30**] do not presume stochastic independence of the flows.

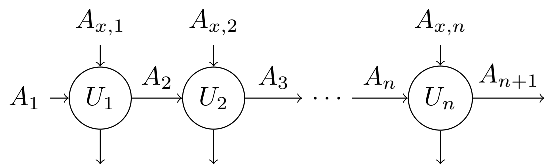
This chapter ties in with the above results and aims at performance bounds in general feedforward networks. To this end, the method used in [**65**], together with a way to perform MGF-calculus with stochastically dependent processes, is presented. Afterwards an algorithm is derived that either finds service descriptions in feedforward networks or identifies the underlying graph as cyclic. This algorithm uses the network operations $\oslash$ and $\ominus$ only. Indeed, these are sufficient for a complete reduction of acyclic graphs; however, the convolution operation $\otimes$ usually improves global bounds significantly. A simple modification of the introduced algorithm allows to take advantage of the convolution operation.

Section 3.3 describes the *DISCO Stochastic Network Calculator* [**14**] – an open-source tool that implements the aforementioned algorithm. None of the other currently existing tools for deterministic network calculus [**135, 78, 152, 136, 26**] extends to the stochastic domain. The *Stochastic Network Calculator* is so far the only available tool that implements SNC.

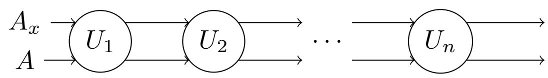
3.1. MGF-bounded Analysis of Tandems

Assume for the rest of this chapter that $\mathbb{K} = \mathbb{N}_0$.

The simplest feedforward network is a tandem that is traversed by the flow of interest (see Figure 3.1). Theorem 1.7 solves this scenario within deterministic network calculus. For the analysis of stochastically independent, MGF-bounded service elements, Theorem 2.36 could be applied; however, in such an iterative application a series of constants K would arise. These constants lead to loose



(a) Tandem with per-server crossflows.



(b) Tandem with end-to-end crossflow.

FIGURE 3.1. Two tandems with crossflows.

bounds, especially when $\rho_{U_1}(\theta) \approx \rho_{U_2}(\theta)$. One can avoid the constants completely, if the arrivals bounds are distributed over the service bounds. To that end, the MGF is bounded *after* deconvolving the arrivals A with the system's service $U_1 \otimes \ldots \otimes U_n$. This is an immediate consequence of Theorem 3 in [**65**].

THEOREM 3.1 (End-to-End Performance in MGF-Calculus). *Fix some $\theta > 0$ and consider a sequence of two service elements as in Theorem 2.36 and Figure 2.6; further, let $A \preceq (\sigma_A, \rho_A)$ be an arrival to this tandem. Let A, U, and V be stochastically independent. Under the stability condition $\rho_A(\theta) < -\rho_U(\theta) \wedge -\rho_V(\theta)$, it holds*

$$\phi_{A \oslash (U \otimes V)(s,t)}(\theta) \leq e^{\rho_A(\theta)(t-s)} \frac{e^{\theta(\sigma_A(\theta)+\sigma_U(\theta)+\sigma_V(\theta))}}{(1-e^{\theta(\rho_U(\theta)+\rho_A(\theta))})(1-e^{\theta(\rho_V(\theta)+\rho_A(\theta))})} \tag{3.1}$$

for all $(s,t) \in \Lambda$. If $U \otimes V(s,t) \geq 0$ for all $(s,t) \in \Lambda$, Equation (3.1) extends to all pairs $s, t \in \mathbb{N}_0 \times \mathbb{N}_0$.

REMARK 3.2. The expression $\phi_{A \oslash U(s,t)}(\theta)$ is the key ingredient in any MGF-based performance bound with arrivals A and service U. See also the proof of Theorem 2.39.

PROOF. First, fix some $(s,t) \in \Lambda$. Expanding the deconvolution leads to

$$A \oslash (U \otimes V)(s,t) = \max_{0 \leq r_1 \leq r_2 \leq s} \{A(r_1,t) - U(r_1,r_2) - V(r_2,t)\}. \tag{3.2}$$

Together with $\mathbb{E}(X \vee Y) \leq \mathbb{E}(X) + \mathbb{E}(Y)$ this gives

$$\begin{aligned}
&\phi_{A\oslash(U\otimes V)(s,t)}(\theta) \\
&\leq \sum_{r_1=0}^{s} \phi_{A(r_1,t)}(\theta) \sum_{r_2=r_1}^{s} \phi_{U(r_1,,r_2)}(-\theta)\phi_{V(r_2,s)}(-\theta) \\
&\leq e^{\theta(\sigma_A(\theta)+\sigma_U(\theta)+\sigma_V(\theta))} \sum_{r_1=0}^{s} e^{\theta\rho_A(\theta)(t-r_1)} \sum_{r_2=r_1}^{s} e^{\theta\rho_U(\theta)(r_2-r_1)+\theta\rho_V(\theta)(s-r_2)} \\
&= e^{\theta(\sigma_A(\theta)+\sigma_U(\theta)+\sigma_V(\theta))} e^{\theta\rho_A(\theta)(t-s)} \\
&\quad \cdot \sum_{r_1=0}^{s} \sum_{r_2=r_1}^{s} e^{\theta(\rho_U(\theta)+\rho_A(\theta))(r_2-r_1)+\theta(\rho_V(\theta)+\rho_A(\theta))(s-r_2)} \\
&\leq e^{\theta(\sigma_A(\theta)+\sigma_U(\theta)+\sigma_V(\theta))} e^{\theta\rho_A(\theta)(t-s)} \\
&\quad \cdot \sum_{r_1'=0}^{\infty} \sum_{r_2'=0}^{\infty} e^{\theta(\rho_U(\theta)+\rho_A(\theta))r_1'+\theta(\rho_V(\theta)+\rho_A(\theta))r_2'} \\
&= \frac{e^{\theta(\sigma_A(\theta)+\sigma_U(\theta)+\sigma_V(\theta))} e^{\theta\rho_A(\theta)(t-s)}}{(1-e^{\theta(\rho_U(\theta)+\rho_A(\theta))})(1-e^{\theta(\rho_V+\rho_A(\theta))})}.
\end{aligned}$$

For the case $s > t$ and the assumption $U \otimes V \geq 0$, the maximum in (3.2) can be limited to

$$A \oslash (U \otimes V)(s,t) = \max_{0\leq r_1 \leq t} \{A(r_1,t) - U \otimes V(r_1,s)\}.$$

Similar steps as before lead to

$$\begin{aligned}
&\phi_{A\oslash(U\otimes V)(s,t)}(\theta) \\
&\leq \sum_{r_1=0}^{t} \sum_{r_2=r_1}^{s} \phi_{A(r_1,t)}(\theta)\phi_{U(r_1,r_2)}(-\theta)\phi_{V(r_2,s)}(-\theta) \\
&\leq e^{\theta(\sigma_A(\theta)+\sigma_U(\theta)+\sigma_V(\theta))} \\
&\quad \cdot \sum_{r_1=0}^{t} \sum_{r_2=r_1}^{s} e^{\theta(\rho_U(\theta)+\rho_A(\theta)(r_2-r_1)+\theta(\rho_V(\theta)+\rho_A(\theta))(s-r_2)+\rho_A(\theta)(t-s)} \\
&\leq \frac{e^{\theta(\sigma_A(\theta)+\sigma_U(\theta)+\sigma_V(\theta))} e^{\theta\rho_A(\theta)(t-s)}}{(1-e^{\theta(\rho_U(\theta)+\rho_A(\theta))})^{-1}(1-e^{\theta(\rho_V(\theta)+\rho_A(\theta))})^{-1}}.
\end{aligned}$$

□

REMARK 3.3. The MGF-bounds of $U_1 \otimes U_2$ and $A \oslash (U_1 \otimes U_2)$ do not depend on the ordering of the two service elements; hence, from a performance bound perspective it is irrelevant in which sequence they are traversed.

Hölder's inequality is central for the analysis of stochastically dependent processes.

THEOREM 3.4 (Hölder's Inequality). *Let X and Y be two positive, real-valued random variables. Then it holds*

$$\mathbb{E}(XY) \leq \mathbb{E}(X^p)^{1/p}\mathbb{E}(X^q)^{1/q}$$

for all Hölder parameters $\frac{1}{p} + \frac{1}{q} = 1$ $(p, q > 1)$.

PROOF. A proof can, for example, be found in [**63**]. □

The network operations (Theorems 2.35-2.38) and the MGF performance bounds (Theorem 2.39 and 3.1) can be generalized with the help of Hölder's inequality. Instead of proving each theorem again, only the convolution result is given. The other network operations follow in the same way.

THEOREM 3.5 (General MGF-Convolution). *Fix some $\theta > 0$ and $\frac{1}{p} + \frac{1}{q} = 1$. If $U \succeq (\sigma_U(p\theta), \rho_U(p\theta))$ and $V \succeq (\sigma_V(q\theta), \rho_V(q\theta))$ with $\rho_U(p\theta) \neq \rho_V(q\theta)$, then*

$$U \otimes V \succeq (\sigma_U(p\theta) - \sigma_V(q\theta) + K, \rho_U(p\theta) \vee \rho_V(q\theta)),$$

where K depends on θ,ρ_U,ρ_V, and the Hölder parameters.

PROOF. Hölder's inequality replaces the assumption of stochastic independence; hence, the MGF of the convolution $U \otimes V$ is bounded by

$$\phi_{U\otimes V(s,t)}(-\theta) \leq \sum_{r=0}^{t} \phi_{U(s,r)}(-p\theta)^{1/p} \phi_{V(r,t)}(-q\theta)^{1/q}.$$

Under the assumption $\rho_U(p\theta) > \rho_V(q\theta)$, the MGF-bounds lead to

$$\begin{aligned}
\phi_{U\otimes V(s,t)}(-\theta) &\leq \sum_{r=0}^{t} (e^{p\theta(\sigma_U(p\theta)+\rho_U(p\theta)(r-s))})^{1/p} (e^{q\theta(\sigma_V(q\theta)+\rho_V(q\theta)(t-r))})^{1/q} \\
&= e^{\theta(\sigma_U(p\theta)+\sigma_V(q\theta))} e^{\theta\rho_U(p\theta)(t-s)} \sum_{r=0}^{t} e^{\theta(\rho_V(q\theta)-\rho_U(p\theta))(t-r)} \\
&\leq e^{\theta(\sigma_U(p\theta)+\sigma_V(q\theta))} e^{\theta\rho_U(p\theta)(t-s)} (1 - e^{\theta\rho_V(q\theta)-\theta\rho_U(p\theta)})^{-1} \\
&= e^{\theta(\sigma_U(p\theta)+\sigma_V(q\theta))} e^{\theta\rho_U(p\theta)(t-s)} K.
\end{aligned}$$

If $\rho_V(q\theta) > \rho_U(p\theta)$, the constant reads $K = (1 - e^{\theta\rho_U(p\theta)-\theta\rho_V(q\theta)})^{-1}$ instead. □

3.2. An Algorithmic Approach to Feedforward Networks

This section presents an algorithm to analyze feedforward networks with the help of MGF-calculus. As in Section 1.2, networks are considered as a graph with labeled edges and nodes. The labels are (σ, ρ)-bounds on the arrivals and service elements, respectively. The goal is to reduce the graph to a single labeled node and a single labeled flow. The initial information on the network consists of bounds on the ingress arrivals and the service elements; further, the flows' priorities at each node are given.

The following notations are introduced to describe the graph and its labels: The graph is defined by a set of nodes $\mathcal{N}$ and I routes $R_i = (r_{i,1}, \dots r_{i,l_i}) \subset \mathcal{N}^{l_i}$ together with priorities $P_i \in (\mathbb{R}_0^+)^{l_i}$ for each $i = 1, \dots, I$. Here $p_{i,k}$ denotes the priority of the i-th route at its k-th hop. At each node the priorities are well-defined, i.e., can be totally ordered. The initial arrival into the network is labeled and denoted by $A_{i,r_{i,1}} \preceq (\sigma_{A_i}, \rho_{A_i})$. The arrivals on subsequent hops of a route i are denoted by $A_{i,r_{i,2}}, A_{i,r_{i,3}}, \dots$ and are initially unlabeled. The set of edges is constructed from the routes via $\mathcal{E} = \bigcup_{i=1}^{I} \bigcup_{k=1}^{l_i} (r_{i,k-1}, r_{i,k})$. Each service element $j \in \mathcal{N}$ is a dynamic U_j-server for the aggregate $\{(i,k) \in \mathcal{E} : k = j\}$ of its respective arrivals and is initially labeled with $U_j \succeq (\sigma_{U_j}, \rho_{U_j})$. As each node offers a strict priority scheduling, the incoming arrival with the highest priority sees a dynamic U_j-server;

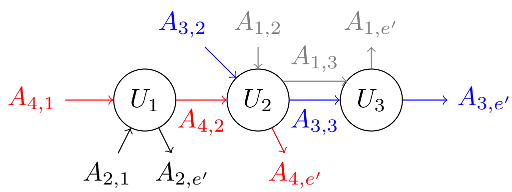


FIGURE 3.2. An example network with routes $R_1 = (2,3)$, $R_2 = (1)$, $R_3 = (2,3)$, $R_4 = (1,2)$, $P_1 = (10,5)$, $P_2 = (5)$, $P_3 = (8,4)$, $P_4 = (7,6)$. The graph is drawn in the following way: When two flows enter the same node, the incoming edge of the flow with the higher priority is drawn above the other flow's edge.

further, the arrivals of route i see a dynamic $U_{i,j}$-server at service element j. As the graph is considered to be acyclic, it follows that each route visits any node at most once. Hence, the notation $U_{i,j}$ is well defined.

Figure 3.2 gives an example of a network's graph $(\mathcal{N}, \mathcal{E})$, as well as its routes R_i with their priority vectors P_i.

The algorithm for computing performance bounds uses a stack $\mathcal{S}$ containing nodes from $\mathcal{N}$. A service element enters the stack, when all its incoming edges are labeled, i.e., an MGF-bound is known for each arriving flow.

EXAMPLE 3.6. In Figure 3.2 the algorithm initializes $\mathcal{S}$ by pushing node U_1. After the stack is updated, the first service element of $\mathcal{S}$ is popped to compute leftover and output descriptions. In the example element U_1 is popped and the leftover operation (Theorem 2.37) gives

$$\begin{aligned} U_{4,1} &= U_1 \succeq (\sigma_{U_1}, \rho_{U_1}) \\ U_{2,1} &= U_1 \ominus A_{4,1} \succeq (\sigma_{U_1} + \sigma_{A_4}, \rho_{U_1} + \rho_{A_4}). \end{aligned}$$

Inserting these into the output operation (Theorem 2.38) gives the new labels

$$\begin{aligned} A_{4,2} &\preceq (\sigma_{U_1} + \sigma_{A_4} + K_{1,4}, \rho_{A_4}) \\ A_{2,e'} &\preceq (\sigma_{U_1} + \sigma_{A_4} + \sigma_{A_2} + K_{1,2}, \rho_{A_2}). \end{aligned}$$

After computing these bounds, all ingress edges of node U_2 are labeled and U_2 is pushed into $\mathcal{S}$. In the next step the algorithm pops U_2 and calculates labels for $A_{1,3}$, $A_{3,3}$, and $A_{4,e'}$. This, in turn, allows U_3 to be pushed. In a last step (by popping U_3 immediately again), the last service descriptions and output bounds are calculated. The result is a fully labeled network.

In general one is interested in a particular flow as it traverses a sequence of one or several service elements. This flow is called the flow of interest. The sequence of service elements it traverses is called the service of interest. Assume for a while that the service of interest consists of a single service element. The above algorithm always terminates by emptying its stack $\mathcal{S}$. At this point either of two results is achieved: 1) the flow of interest is labeled and a service description for the flow of interest is given or 2) the labeling is incomplete. It is easy to see that the algorithm terminates in the former way, if the network is acyclic.

Consider the network operations $\ominus$ and $\oslash$ as one elementary operation. Then the runtime of the algorithm is bounded by $|\mathcal{N}| \cdot I$. This runtime is attained by a tandem network of $|\mathcal{N}|$ nodes and I, where each flow traverses the full tandem (see Figure 3.1 (b) for an example).

The algorithm needs to track any arising stochastic dependencies. In Example 3.6 flows $A_{4,1}$ and $A_{2,1}$ are stochastically independent, flows $A_{4,2}$ and $A_{2,e'}$, however, are not. Both of them have been served by the same service element. In this way stochastic dependencies grow deeper into the network by each hop. The presented algorithm discovers and stores these dependencies for each flow or service object (see next section). Any step involving an $A_{i,j}$ and a $U_{k,l}$ such that the intersection of their corresponding dependency-vectors is nonempty introduces a new pair of Hölder parameters. Hence, the result of this algorithm leads to functions of multiple Hölder parameters.

When the service of interest consists of several service elements, a simple (yet loose) end-to-end delay bound is

$$\mathbb{P}(\mathfrak{d}_i(t) > T) \leq \sum_{k=1}^{l_i} \mathbb{P}\left(\mathfrak{d}_{i,r_{i,k}} > \frac{T}{l_i}\right).$$

Here $\mathfrak{d}_{i,r_{i,k}}$ denotes the delay at the k-th node of flow i. A better end-to-end delay bound is achieved by using the convolution theorem, as presented in the previous section.

3.3. The DISCO Stochastic Network Calculator

This section presents the DISCO Stochastic Network Calculator [**14**]. It is the first – and so far only – publicly available tool that automatizes the steps needed to calculate a stochastic performance bound. It uses the above algorithm in its first step. In a second step the program searches for a near optimal choice of Hölder parameters and θ.

To serve the needs of an evolving theory, the DISCO SNC aims to be easily extendable by its user. The DISCO SNC allows researchers to implement their own methods, e.g., a strategy to compute end-to-end delays or a heuristic for a numerical optimization of free parameters. To this end, the DISCO SNC follows a modular approach, in which the core (the SNC object) relates the modules to each other. Figure 3.3 shows these modules.

In its first step the DISCO SNC works on a symbolic level. This allows to separate analysis techniques (as discussed in Section 1.2) from the step of numerical optimization. As a result, the user can switch strategies for the analysis or the numerical optimization independently and does not need to concern about the core's code.

The different modules and their objects are presented in detail now. The entire code can be found online [**10**].

3.3.1. The SNC Class, the Core of DISCO SNC. ($\rightarrow$ Code Snippet 1) The class `SNC` contains the main-method. It starts and prepares the GUI; further, it serves as an interface to the classes `Network`, `Analysis`, and `Optimizer`. Alternatively, the implemented methods can be used directly in the main-method to construct a network and perform calculations on it.

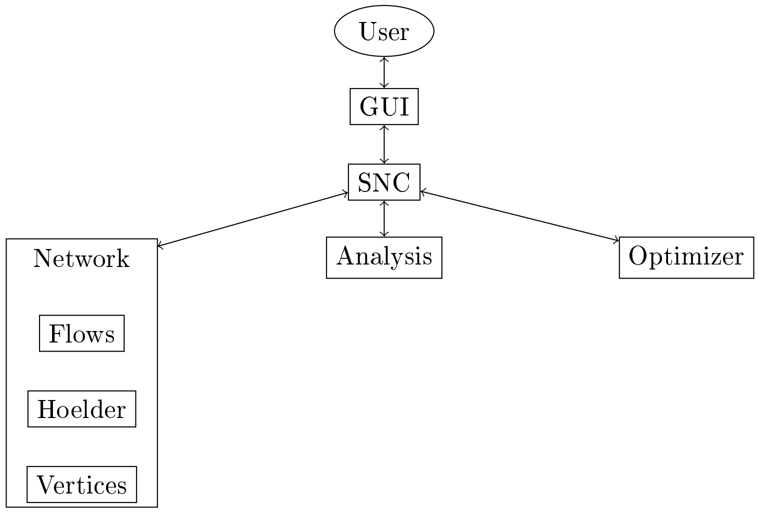


FIGURE 3.3. relation of the DISCO SNC components.

The class `SNC`, further, contains a list of flows and vertices (the building blocks of the network) and a GUI element. It provides methods to load, save, manipulate, and analyze networks. The latter is performed by an `AnalysisType`-object. For the numerical optimization of the resulting functions, an `OptimizationType`-object is used.

3.3.2. The Network-class. ($\rightarrow$ Code Snippet 2) All information of a given scenario is stored in service elements and flows with their routing through and priorities at the service elements. To save and load a scenario, this information is bundled in the network-class. In addition to topological information the network-class contains a `HashMap` of `Hoelders`. These represent Hölder-parameters that occur in the analysis of a scenario.

`Network.java` also contains `HashMap`-objects of `flows` and `vertices` (see the next two subsections); further, it contains methods, such as `addFlow`, to manipulate the network.

3.3.3. Flows and Arrivals. ($\rightarrow$ Code Snippet 5) Each `flow` object represents and contains the information of a flow through the entire network. It contains three `ArrayList`-objects: `vertices`, `priorities`, and the `arrivals`. The former two give the route of a flow by the sequence of crossed vertices together with the priority at each service element. The latter is a description of arrival bounds for each hop of a flow, i.e., the labels in the corresponding graph.

Objects of the `Arrival`-class ($\rightarrow$ Code Snippet 4) are closely related to `flow`-objects. An `arrival`-object contains two `Set<Integer>`-objects to store the Hölder-parameters that are of relevance for evaluating the corresponding arrival-bound. It contains, further, methods to perform the two network operations $\oplus$ and $\oslash$. The former takes two `arrival`-objects as input, the latter an `arrival`- and a `service`-object.

Two other members of an `arrival`-object are `sigma` and `rho` of type `FunctionIF`. One can think of these objects as functions in the mathematical

Code Snippet 1 Members and methods of SNC.java

```
public class SNC {
  //Members

  private static HashMap<Integer, Flow> flows;
  private static HashMap<Integer, Vertex> vertices;
  private static GUI gui;
  ...

  //Main-Method

  public static void main(String[] args) throws Exception{
    SNC snc = new SNC();
    vertices = Network.getVertices();
    flows = Network.getFlows();
    gui = new GUI(snc);
    SwingUtilities.invokeLater(gui);
  }

  //Methods

  //Loading, saving, and manipulations of networks
  public void loadNetwork(File file){...}
  public void saveNetwork(File file){...}
  public boolean removeFlow(Flow flow) {...}
  public int addFlow(Flow flow){...}

  //Analysis of Network (symbolic)
  public Arrival analyzeNetwork(Flow flow, Vertex vertex, SNC.AnalysisType
      analyzer, AbstractAnalysis.Boundtype boundtype){...}

  //Calculation of Performance Bounds (numeric)
  public double calculateBound(Flow flow, Vertex vertex,...,
      SNC.AnalysisType analyzer, SNC.OptimizationType optimizer,...){...}
  ...
```

sense. Indeed, `FunctionIF`'s most important method is to take one or more parameters and evaluate a mathematical function at these parameters. For example `AddedFunctions.java` (→ Code Snippet 3) has two `FunctionIF`-objects – called atom-functions. When the `getValue`-method is called, it evaluates both atom-functions and returns the sum of it. To this end, `AddedFunctions.java` basically serves the purpose of $+$ in the expression $f(\theta) + g(\theta)$, where f and g are the atom-functions. The `FunctionIF`-interface makes the symbolic computation of performance bounds possible. In the context of arrivals, the `FunctionIF sigma` and `rho` serve the corresponding roles in the MGF-bounds (σ, ρ). For details see the `evaluate`-method in Code Snippet 4.

3.3.4. Vertex and Service. (→ Code Snippet 6) The relation between `vertex` and `service` is similar to the one between `flow` and `arrival`. A `vertex`

Code Snippet 2 Members and methods of Network.java

```
public class Network {
  //Members
  ...
  private static HashMap<Integer,Flow> flows = new HashMap<Integer,
      Flow>(0);
  private static HashMap<Integer,Vertex> vertices = new HashMap<Integer,
      Vertex>(0);
  private static HashMap<Integer,Hoelder> hoelders = new HashMap<Integer,
      Hoelder>(0);

  //Methods

  public static void addVertex(Service service, String alias){...}
  public static void addFlow(ArrayList<Integer> route, int priority){...}
  ...
```

Code Snippet 3 AddedFunctions.java.

```
public class AddedFunctions implements FunctionIF {
  //Members

  private FunctionIF first;
  private FunctionIF second;
  ...

  //Methods

  @Override
  public double getValue(double theta, HashMap<Integer, Hoelder>
      parameters)... {
    ...
    //Constructs the parameter-arrays that serve as input for the
        atom-functions
    HashMap<Integer, Hoelder> given1 = new HashMap<Integer, Hoelder>();
    HashMap<Integer, Hoelder> given2 = new HashMap<Integer, Hoelder>();

    //Multiplies the Hölder-coefficients to theta, if needed
    double theta1 = (hoelder == null) ? theta : theta*hoelder.getPValue();
    double theta2 = (hoelder == null) ? theta : theta*hoelder.getQValue();
    return first.getValue(theta1, given1) + second.getValue(theta2,
        given2);
  }
```

contains information about the original service of a service element before calculating any leftover service guarantees. A `vertex` has members `priorities` and `incoming`. Both are `HashMap`-objects. They contain information about the incoming flows at the `vertex` and their priorities. The incoming arrivals are either

Code Snippet 4 Members and methods of Arrival.java

```
public class Arrival ... {
  //Members
  ...
  private FunctionIF rho;
  private FunctionIF sigma;
  private Set<Integer> Arrivaldependencies;
  private Set<Integer> Servicedependencies;
  //Methods
  public double evaluate(double theta, HashMap<Integer, Hoelder>
      sigmaparameters, HashMap<Integer, Hoelder> rhoparameters, int n, int
      m) ... {
    double value;
    try{
      value = Math.exp(theta*sigma.getValue(theta, sigmaparameters) +
          theta*rho.getValue(theta, rhoparameters)*(n-m));
    }
    ...
    return value;
  }
  public Arrival multiplex(Arrival arrival1, Arrival arrival2){
    Arrival arrival;
    //Independent case
    ...
    FunctionIF givensigma = new
        AddedFunctions(arrival1.getSigma(),arrival2.getSigma(),true);
    FunctionIF givenrho = new AddedFunctions(arrival1.getRho(),
        arrival2.getRho(), true);
    arrival = new Arrival(givensigma, givenrho);
    //Keeps track of stochastic dependencies
    arrival.addArrivalDependency(arrival1.getArrivaldependencies());
    arrival.addArrivalDependency(arrival2.getArrivaldependencies());
    arrival.addServiceDependency(arrival2.getServicedependencies());
    arrival.addServiceDependency(arrival1.getServicedependencies());
    return arrival;
  }
  public Arrival output(Arrival arrival, Service service){
    Arrival output;
    FunctionIF givensigma = new AddedFunctions(new
        AddedFunctions(arrival.getSigma(),service.getSigma(),true),new
        BFunction(...),true);
    FunctionIF givenrho = arrival.getRho();
    output = new Arrival(givensigma, givenrho);
    //Keeps track of stochastic dependencies
    ...
    return output;
  }
  ...
```

Code Snippet 5 Members and methods of Flow.java

```
public class Flow ... {
   //Members

   private ArrayList<Integer> vertices;
   private ArrayList<Arrival> arrivals;
   private ArrayList<Integer> priorities;
   ...

   //Methods
   public void learnArrival(Arrival arrival){
      arrivals.set(established_arrivals, arrival);
      established_arrivals++;
   }
   public void setInitialArrival(Arrival arrival){
      arrivals.add(0, arrival);
      arrivals.get(0).addArrivalDependency(flow_ID);
      established_arrivals = 1;
   }
   ...
```

established or *unknown*. In this context being established means that the `vertex` knows an MGF bound of an arrival or, stated differently, the corresponding edge is labeled.

The member `service` contains the MGF-bound for the service element in form of `FunctionIF`-objects. The `service`-class is very similar to the `arrival`-class in its structure. It provides methods to perform the network operations $\otimes$ and $\ominus$. Each of them has a new `service`-object as output.

The `serve`-method is the most important one of a `vertex`-object. It executes the two network operations $\ominus$ and $\oslash$ at the service node. Afterwards, the served flow is removed from the list of incoming flows. Iterating this method results in a successive computation of left-over service for each incoming flow and an MGF-bound for each outgoing flow.

3.3.5. Analysis. ($\rightarrow$ Code Snippet 7) The `analysis`-class performs symbolic operations exclusively. It is an abstract class with the task to derive a stochastic performance bound at a node of interest for the flow of interest. This bound is either a backlog, delay, or an output bound and is given algebraically. The abstract class is presented in Code Snippet 7 and consists mainly of the `analyze`-method.

The result of `analyze` is an object of type `Arrival`. This is because a performance bound as in Theorem 2.39 can be expressed via two functions ρ and σ again.

The algorithm of the previous section is implemented in the `SimpleAnalysis`-class. It extends the `AbstractAnalysis`-class. Code Snippet 8 presents how the stack $\mathcal{S}$ of the previous section is implemented. The analysis of the network is – in principle – just a manipulation of functions on a symbolic level. Hence, this step requires no significant computational effort compared to the numerical optimization of free parameters.

Code Snippet 6 Members and methods of Vertex.java

```
public class Vertex ... {
   //Members
   ...
   private Service service;
   private HashMap<Integer, Integer> priorities;
   private int highest_priority;
   private HashMap<Integer, Arrival> incoming;

   //Methods
   public void learnArrival(int flow_id, Arrival arrival) throws Exception{
      ...
      incoming.put(flow_id, arrival);
      ...
   }
   public Arrival serve() throws Exception{
      ...
      //Calculates the output bound
      Arrival arrival = incoming.get(prioritized_flow_id);
      Arrival output = arrival.output(arrival, service);
      //Calculates the leftover service
      service = service.leftover(arrival, service);
      //Removes the served flow from the arrival-list
      priorities.remove(prioritized_flow_id);
      incoming.remove(prioritized_flow_id);
      ...
      return output;
   }
   ...
```

Code Snippet 7 Members and methods of AbstractAnalysis.java

```
public abstract class AbstractAnalysis {
   //Members
   protected HashMap<Integer, Vertex> vertices;
   protected HashMap<Integer, Flow> flows;
   protected int flow_of_interest;
   protected int vertex_of_interest;
   public enum Boundtype{
      BACKLOG, DELAY, OUTPUT, END_TO_END_DELAY
   };
   protected Boundtype boundtype;

   //Methods
   public abstract Arrival analyze() throws Exceptions;
```

Code Snippet 8 analyze-method of SimpleAnalysis.java

```
public class SimpleAnalysis extends AbstractAnalysis {
   //Members
   private Stack<Vertex> can_serve;
   //Methods

   public Arrival analyze() throws Exceptions{
      //Initializes the stack of vertices, for which all arrivals are known
      for(Map.Entry<Integer, Vertex> entry : vertices.entrySet()){
         if(entry.getValue().canServe()) can_serve.push(entry.getValue());
      }  ...
      Vertex current_vertex;
      Arrival bound = new Arrival();
      boolean successful = false;
      //Successively serves the flows until the FoI and SoI is characterized
      while(!can_serve.isEmpty()){
         //Setup of service and flow
         current_vertex = can_serve.pop();
         int flowID = current_vertex.whoHasPriority();
         Vertex next_vertex =
             vertices.get(flows.get(flowID).getNextVertexID());
         //Checks for the current vertex and flow being the SoI and the FoI,
             respectively
         if(current_vertex.getVertexID() == vertex_of_interest && flowID ==
             flow_of_interest){
            bound = calculateBound(flows.get(flowID).getLastArrival(),
                current_vertex.getService());
            successful = true;
            break;
         }
         //Calculates the output and sets the service in the vertex to the
             next leftover service
         Arrival output = current_vertex.serve(); ...
         flows.get(flowID).learnArrival(output); ...
         next_vertex.learnArrival(flowID, output);
         //will push the next vertex, if it knows all its arrivals
         if(next_vertex.canServe()) {
            can_serve.push(next_vertex);
         }  ...
         //will push the current vertex, if it has more flows to serve
         if(current_vertex.canServe()) can_serve.push(current_vertex);
      }
      //Checks for the FoI and the SoI being calculated
      if(successful == false) throw new DeadlockException("Flow of Interest
          or Arrival of Interest can't be calculated.
          Non-feedforward-Network?");
      return bound;
   }
}
```

Code Snippet 9 Members and methods of AbstractOptimizer.java

```
public abstract class AbstractOptimizer {
   //Members
   protected Arrival input;
   protected AbstractAnalysis.Boundtype boundtype;
   protected HashMap<Integer, Hoelder> sigma_parameters;
   protected HashMap<Integer, Hoelder> rho_parameters;
   protected double max_theta;
   //Methods
   ...
   public abstract double Bound(Arrival input, AbstractAnalysis.Boundtype
       boundtype, double bound, double thetagranularity, double
       hoeldergranularity) throws Exceptions;
   public abstract double ReverseBound(Arrival input,
       AbstractAnalysis.Boundtype boundtype, double violation_probability,
       double thetagranularity, double hoeldergranularity) throws
       Exceptions;
```

3.3.6. Optimization. (→Code Snippet 9) To achieve a numerical value for a performance bound, the method `evaluate` of the `Arrival`-object of the previous step must be called. This requires a numerical value for θ and all additional Hölder-parameters. The resulting performance bound depends heavily on a good choice of these parameters. It is the goal of the optimizer to find a good assignment of parameters.

Two methods were implemented: `Bound` and `ReverseBound`. The former gives the violation probability of a backlog or delay bound controlled by the user. The latter takes as input a fixed violation probability ε. Its output is the best found backlog- or delay bound that is violated by at most ε.

The DISCO SNC provides two methods for optimizing the set of parameters. The first one is a systematic search through the parameter-space. To this end, the interval of possible θ-assignments is divided into steps of length δ_θ. The range of a Hölder-parameter is unlimited in principle. For a finite set of possible values for p, the proposed method discretizes the interval $[1, 2]$ into steps of length δ_p. Now p takes values in this set and afterwards q iterates through the same set of values; since $\frac{1}{p} + \frac{1}{q} = 1$ this method covers a wide range of possible values for the Hölder-pair. The proposed discretization is coarse for p- or q-values close to 1. Yet, practice has shown that optimal Hölder-parameters are rather close to 2; hence, this discretization is still of good value.

The number of Hölder-parameters depends on the scenario and thus must be a variable. The `SimpleOptimization`-class uses a tally counter to systematically search through a parameter-space of unknown dimension. The tally counter returns `FALSE` the first time it has run through all possible choices for the Hölder-parameters.

3.3.7. Improving the Optimization Step. The amount of Hölder-parameters can increase very quickly for a given scenario, even for few nodes and flows. Thus, systematically searching through all combinations of parameters becomes a restricting factor quickly.

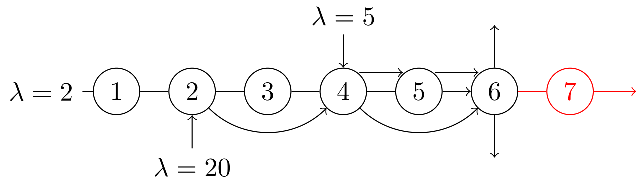


FIGURE 3.4. Network corresponding to Table 1. All service elements have a constant rate of 3. The increments of the flows are stochastically independent and exponentially distributed with parameter λ. The backlog bound is calculated for the red part of the network.

δ_θ, δ_p	r_{HJ}	r_S	x_{HJ}	x_S
0.05	≈1 sec.	≈3 min.	60.3	60.285
0.04	≈1 sec.	≈7 min.	53.013	52.155
0.03	≈1 sec.	≈37 min.	52.035	52.035
0.001	≈1 sec.	N/A	51.092	N/A

TABLE 1. Runtime and delay bounds of HJ heuristic and systematic (S) search at different granularities. A systematic search for $\delta_\theta = \delta_p = 10^{-3}$ exceeded the author's patience (8 hours).

The DISCO SNC implements a second method that uses the heuristic of Hooke and Jeeves [**79**] to find good sets of parameters much faster. The HJ-heuristic is a discrete version of gradient based optimization: From some starting point in the discretized parameter space, the performance bounds of all 1-hop-neighbors are compared to the performance bound of the starting point. A neighbor is defined as a set of parameters that differs from the original set of parameters in exactly one variable (θ or an arbitrary Hölder-parameter) by exactly one δ_θ or δ_p, respectively. If a better performance bound is found amongst the neighbours, the heuristic moves to the best performing neighbor. This procedure continues until a set of parameters is found that cannot be improved further by moving to a neighboring set of parameters (local minimum). Code Snippet 10 presents the loop in which this method improves the performance bounds in pseudocode. As this heuristic tests a small portions of the parameter-space, a much faster runtime is expected.

To evaluate this effect, a toy example (Figure 3.4) consisting of three flows and seven nodes is tested. Solving such a scenario by hand, although looking simple, is already out of scope. The resulting parameter space encompasses four pairs of Hölder parameters. The runtime r of the program and the resulting quality of the bound $\mathfrak{b}$ depends on δ_θ and δ_p, which are chosen equal for this evaluation. Table 1 shows that the runtime of the HJ-heuristic easily outperforms the runtime of a systematic search. At the same time the resulting bounds are of almost equal quality (x_{HJ} and x_S are the best found value for x in the bound $\mathbb{P}(\mathfrak{b} > x) \leq 10^{-4}$).

Code Snippet 10 The core loop of SimpleGradient.java

```
//Computes initial value for Backlog Bound with all Hoelder-Parameter equal
    to 2 and theta maximally chosen ...
double backlogprob = input.evaluate(theta, 0, 0);
while(improving the bound){
   //Checks for a better result, if theta is decreased ...
   theta = theta - thetagranularity;
   new_backlogprob = input.evaluate(theta, 0, 0);
   if(backlogprob > new_backlogprob){
      backlogprob = new_backlogprob;
      change = SimpleGradient.Change.THETA_DEC;
   }
   theta = theta + thetagranularity;
   //Checks for a better result, if theta is increased ...
   //Checks all neighbors resulting from decreasing the P-value of one
       Hoelder parameter
   for(all considered neighbors){
      updateHoelderParameters;
      new_backlogprob = input.evaluate(theta, 0, 0);
      if(backlogprob > new_backlogprob){
         backlogprob = new_backlogprob;
         change = SimpleGradient.Change.HOELDER_P;
      } ...
   }
   //Check each neighbor resulting from decreasing the Q-Value of one
       Hoelder parameter ...
   switch(change){
      case THETA_INC:
         theta = theta + thetagranularity;
         improved = true;
         break;
      case THETA_DEC: ...
      case HOELDER_P:
         updateHoelderParameters;
         improved = true;
         break;
      case HOELDER_Q: ...
      default:
         improved = false;
         break;
      }
   }
result = backlogprob;
```

CHAPTER 4

Statistical Network Calculus

The results of this chapter are joint work with S. Henningsen, S. Birnbach, and J. Schmitt [**12**].

Stochastic network calculus works with a variety of different traffic models. Independent and identically distributed increments are bounded in Example 2.32, MGF-bounds for Markov-modulated arrivals are constructed in Lemma 10.6, and a model using fractional Brownian motions is discussed in [**132**], to name three examples.

As all of these are models only, any real arrivals must be analyzed first before stochastic network calculus can be applied. In this modeling step assumptions must be made. These assumptions lie in the chosen traffic model and its parameters. But also in their relations between arrivals and service elements, i.e., the stochastic dependency structures. Usually these assumptions stem from measurements in combination with statistical methods. Consider, as example, independent, exponentially distributed increments with rate parameter λ; then, the parameter λ can be estimated by the arithmetic mean of several observed increments. Doing so bears the risk that the seen sample deviates so strongly from the usual behavior, that the resulting λ is too large. Consequently, the real size of future increments would be underestimated.

In many fields this risk – the confidence level α – is of little or no importance. In stochastic network calculus, however, one specifically asks for events that happen with almost vanishing probabilities. Usually these violation probabilities are much smaller (of order 10^{-3}, 10^{-6} or even 10^{-9}) than the usual level of confidence used in statistics ($\alpha = 0.95$, 0.99 or 0.999). Hence, the endeavors put into SNC to achieve a violation probability of 10^{-6} can quickly become vain, if the corresponding model is wrong with a probability of 1%. Simply pushing α to a value extremely close to 1 cannot be a sufficient solution either, as the resulting model becomes ever more conservative. This chapter presents a third way: The confidence level α is integrated into the performance bounds directly. This in turn integrates the modeling step into the analysis of the system. In the above example this works as follows: Instead of modeling the increments for some fixed value of λ, only their exponential distribution is assumed. This effectively leads to a reduction of assumptions.

Such *statistical* performance bounds are the goal of this chapter. To construct these, a sufficient condition on the used statistics is given in Theorem 4.2. This condition allows to integrate the confidence level into the performance bounds. This chapter gives three estimators as examples. Each of them differs in the amount of presumed knowledge. Eventually a numerical evaluation is concerned with the costs, but also the opportunities that arise from the usage of statistical network calculus (StatNC).

Ciucu and Schmitt construct stochastic arrival curves from measurements for weighted hyperexponential traffic in [**43**]. Similarly, Liebeherr et al. construct arrival curves for heavy-tailed, self-similar traffic in [**107**]. However, in these works no confidence levels are involved or used inside the resulting performance bounds.

The estimation of service curves has been object of recent research [**108, 114, 113, 115**]. The authors perform active measurements of a system's available service with the help of probing traffic. The so obtained stochastic service descriptions are similar to the one in Section 2.1. An approach more similar to the one in this chapter is followed by Jiang et al. [**149, 29**]. There internet routers are described by parametrized service curves. These parameters are estimated by performance-measurements like the length of backlog periods. However, these works do not calculate or integrate the confidence level α either. Further, the objective is different from this chapter that focuses on estimating the arrivals.

Measuring arrivals is also a topic in the earlier works of admission control [**85, 74, 76, 130**]. Originating in a time in which stochastic network calculus was still under heavy construction, they center around the effect of statistical multiplexing. Further, these works assume a given arrival description and measure its validity. In contrast to that, the following sections are concerned with constructing these arrival descriptions in the first place.

An interesting direction is opened by the work of Dong, Wu, and Srinivasan [**56**]. Following the here presented idea, they apply copulas for an analysis of stochastic dependencies between arrivals. A full integration of copula-based statistics into stochastic network calculus promises to be beneficial: It might not only reduce the number of assumptions made, but also improve the resulting performance bounds overall.

4.1. The Framework of Statistical Network Calculus

In the derivation of backlog bounds (see Equation (2.25)) the MGF-bound $\phi_{A(s,t)}(\theta) \leq e^{\theta\sigma_A(\theta)+\theta\rho_A(\theta)(t-s)}$ plays a key role. Here problems arise, when the model of A is uncertain. If the exact distribution of the increments is unknown, the expression $\phi_{A(s,t)}(\theta)$ cannot be calculated. This in turn prohibits calculation of further results. Tools of mathematical statistics can bound $\phi_{A(s,t)}(\theta)$ and effectively replace the MGF in the proof of 2.39. The so used statistic is a function of the sample $\bar{a} = (a(t_0), \dots, a(-1))$. This section's goal is to derive sufficient conditions such that the uncertainty of the used estimator is included in the performance bounds.

This chapter uses $\mathbb{K} = \mathbb{Z}$ and assumes that a single flow A enters a dynamic U-server. For $\mathbb{K} = \mathbb{Z}$ the bivariates $A(s,t)$ must be generalized to the negative axis: $A(s,t) := \sum_{r=s+1}^{t} a(r)$. Since the basic idea of StatNC is to apply statistical methods on past observations, time indices $t < 0$ are thought of as lying in the past; further, this chapter assumes a value $t_0 \leq 0$ such that $a(t) = 0$ for all $t < t_0$. The time t_0 represents the beginning of the observations.

The following lemma prepares the main result of this chapter. It shows that the backlog bound is monotone in the flow's MGF.

LEMMA 4.1. *Fix some $\theta > 0$. If A and U are stochastically independent and $\hat{\phi}_{s,t}(\theta) \geq \phi_{A(s,t)}(\theta)$ for all $(s,t) \in \Lambda$, then*

$$\mathbb{P}(q(t) > N) \leq e^{-\theta N} \sum_{s=0}^{t} \hat{\phi}_{s,t}(\theta)\phi_{U(s,t)}(-\theta)$$

for all $t \in \mathbb{N}_0$.

Proof. From the proof of Theorem 2.39 it follows

$$\mathbb{P}(q(t) > N) \leq e^{-\theta N} \sum_{s=0}^{t} \phi_{A(s,t)}(\theta)\phi_{U(s,t)}(-\theta) \leq e^{-\theta N} \sum_{s=0}^{t} \hat{\phi}_{k,n}(\theta)\phi_{U(s,t)}(-\theta).$$

□

Now define the space $\mathcal{S}$ by

$$f \in \mathcal{S} \qquad \Leftrightarrow \qquad f : \mathbb{N}_0 \times \mathbb{N}_0 \times \mathbb{R}^+ \to \bar{\mathbb{R}}_0^+.$$

The MGF's continuation

$$\bar{\phi}_{A(s,t)}(\theta) := \begin{cases} \phi_{A(s,t)}(\theta) & \text{if } \phi_{A(s,t)}(\theta) \text{ is defined} \\ \infty & \text{if } \phi_{A(s,t)}(\theta) \text{ is undefined} \end{cases}$$

is an example for a mapping belonging to $\mathcal{S}$.

The following theorem constructs the framework of statistical network calculus.

Theorem 4.2. *Fix some* $t \in \mathbb{N}_0$. *Let* $\theta^* = \sup\{\theta : \phi_{A(s,t)}(\theta) < \infty\}$ *and* $P : \mathbb{R}^{|t_0|} \to \mathcal{S}$ *be a statistic such that*

$$\sup_{\theta \in (0,\theta^*)} \mathbb{P}\Big(\bigcup_{(s,t)\in\Lambda} P(\bar{a})(s,t,\theta) < \phi_{A(s,t)}(\theta) \Big) \leq \alpha. \tag{4.1}$$

If A and U are stochastically independent, it holds

$$\mathbb{P}(\mathfrak{b}(t) > x) \leq \alpha + e^{-\theta N} \sum_{s=0}^{t} P(\bar{a})(s,t,\theta)\phi_{U(s,t)}(-\theta)$$

for all $t \in \mathbb{N}_0$ *and* $0 < \theta < \theta^*$.

Proof. Fix some $0 < \theta < \theta^*$. Then

$$\begin{aligned} \mathbb{P}(\mathfrak{b}(t) > x) &= \mathbb{P}\Big(\mathfrak{b}(t) > x \cap \bigcup_{(s,t)\in\Lambda} P(\bar{a})(s,t,\theta) < \phi_{A(s,t)}(\theta) \Big) \\ &\quad + \mathbb{P}\Big(\mathfrak{b}(t) > x \cap \bigcap_{(s,t)\in\Lambda} P(\bar{a})(s,t,\theta) \geq \phi_{A(s,t)}(\theta) \Big) \\ &\leq \alpha + \mathbb{P}\Big(\mathfrak{b}(t) > x \,\Big|\, \bigcap_{(s,t)\in\Lambda} P(\bar{a})(s,t,\theta) \geq \phi_{A(s,t)}(\theta) \Big) \\ &\leq \alpha + e^{-\theta x} \sum_{s=0}^{t} P(\bar{a})(s,t,\theta)\phi_{U(s,t)}(-\theta). \end{aligned}$$

The last step uses Lemma 4.1. □

Remark 4.3. By applying Hölder's inequality results for the stochastically dependent case are obtained as well.

The statistic P is defined here in the most general way. There is, however, no need to use the complete past $\bar{a}$ to construct a P. Consider a subsample $\bar{a}' \subset \bar{a}$ and a statistic $P' : \mathbb{R}^{|\bar{a}'|} \to \mathcal{F}$ on $\bar{a}'$. If P' meets the assumption of Theorem 4.2 for $\bar{a}'$, it extends canonically to a statistic P, by setting $P(\bar{a}) = P'(\bar{a}')$ for all $\bar{a}$ such that $\bar{a} \supset \bar{a}'$ holds.

Another method of subsampling would be to use a sliding window of length l on the observations. Such an approach is particularly interesting for an online estimation that dynamically adapts to changes in the arrival's characteristics. In Section 4.3.2, the versatility of this dynamic view is leveraged to achieve bounds that improve upon a pure SNC approach.

4.2. Examples of Estimators

Estimating the quantity $\phi_{A(s,t)}(\theta)$ for an arbitrary $\theta \in (0, \theta^*)$ and $(s,t) \in \Lambda$ is key for StatNC. The following scenarios and corresponding estimators showcase how such statistics can be constructed. A fairly simple example illustrates the core idea of parametric estimators; more complex scenarios follow.

4.2.1. Exponential Traffic. Assume the increments to be independend, exponentially distributed for some unknown parameter λ. As λ completely determines the distribution of $a(t)$, the construction of P relies on estimating that parameter. A lower bound for λ with confidence level α is given by

$$\underline{\lambda} := \frac{\chi^2_\alpha(2|t_0|)}{2 \cdot A(t_0 - 1, -1)},$$

where $\chi^2_\alpha(2|t_0|)$ is the one-sided α-quantile of a Chi-squared distribution with $2|t_0|$ degrees of freedom; indeed, scaling $A(t_0 - 1, -1)$ by 2λ results in a random variable that is $\chi^2(2|t_0|)$-distributed. The MGF of the exponential distribution is $\phi_E(\lambda, \theta) = (\frac{\lambda}{\lambda - \theta})$, for all $\theta < \lambda$. From this follows the implication

$$\underline{\lambda} \le \lambda \Rightarrow \frac{\underline{\lambda}}{\underline{\lambda} - \theta} \ge \frac{\lambda}{\lambda - \theta}.$$

Hence, the statistic

$$P(\bar{a})(s,t,\theta) := \begin{cases} \left(\frac{\underline{\lambda}}{\underline{\lambda} - \theta}\right)^{n-m} & \text{for } \theta < \underline{\lambda} \\ \infty & \text{for } \underline{\lambda} \le \theta \end{cases}$$

fulfills

$$P(\bar{a})(s,t,\theta) \ge \phi_{A(s,t)}(\theta)$$

for all $\theta < \lambda$ and $(s,t) \in \Lambda$. By the choice of $\underline{\lambda}$ it follows

$$1 - \alpha = \mathbb{P}(\underline{\lambda} \le \lambda) \le \inf_{\theta \in (0,\lambda)} \mathbb{P}\Big(\bigcap_{s \le t} P(\bar{a})(s,t,\theta) \ge \phi_{A(s,t)}(\theta) \Big),$$

which is Condition (4.1).

4.2.2. Bandwidth-Limited i.i.d. Traffic. Let $a(t)$ be i.i.d. and adhere to a bandwidth limitation M, i.e., $a(t) \leq M$ for all $t \in \mathbb{Z}$. In contrast to the previous subsection no further knowledge about the distribution of the increments $a(t)$ is assumed. Yet, with the help of the Dvoretzky-Kiefer-Wolfowitz inequality, a P fulfilling Condition (4.1) can be constructed.

LEMMA 4.4 (Dvoretzky-Kiefer-Wolfowitz Inequality). *Denote by F the distribution of $a(t)$ and by $F_{t_0}(x) := {}^1\!/\!{}_{|t_0|} \sum_{s=t_0}^{-1} \mathbf{1}_{\{a(s)\leq x\}}$ the empirical distribution function of the sample $\bar{a}$. For all $\varepsilon > 0$ it holds*

$$\mathbb{P}\Big(\sup_{x\in[0,M]} |F_{t_0}(x) - F(x)| \leq \varepsilon\Big) \geq 1 - 2e^{-2|t_0|\varepsilon^2}.$$

PROOF. See for example [**118**]. □

THEOREM 4.5. *Let $\varepsilon > 0$. The statistic P defined by*

$$P(\bar{a})(s,t,\theta) := (\bar{A} + \varepsilon(e^{\theta M} - 1))^{t-s}$$

satisfies condition (4.1), where $\bar{A} := \frac{1}{|t_0|}\sum_{s=t_0}^{-1} e^{\theta a_s}$.

PROOF. From the event in the Dvoretzky-Kiefer-Wolfowitz inequality it follows successively

$$\begin{aligned}
& F(x) \geq F_{t_0}(x) - \varepsilon \qquad \text{for all } x \in [0,M],\\
\Rightarrow \qquad & 1 - F(x) \leq 1 - F_{t_0}(x) + \varepsilon \qquad \text{for all } x \in [0,M],\\
\Rightarrow \qquad & \mathbb{P}\left(e^{\theta a(s)} > x\right) \leq 1 - F_{t_0}\left({}^1\!/\!{}_\theta \log(x)\right) + \varepsilon \qquad \text{for all } x \in [1, e^{\theta M}],
\end{aligned}$$

$$\begin{aligned}
\Rightarrow \qquad \phi_{a(s)}(\theta) &= 1 + \int_1^{e^{\theta M}} \mathbb{P}\left(e^{\theta a(s)} > x\right) \mathrm{d}x\\
&\leq 1 + \int_1^{e^{\theta M}} 1 - F_{t_0}\left({}^1\!/\!{}_\theta \log(x)\right) + \varepsilon \mathrm{d}x.
\end{aligned}$$

Hence, it holds

$$\begin{aligned}
\mathbb{P}\Big(\phi_{a(s)}(\theta) \leq 1 + \int_1^{e^{\theta M}} 1 - F_{t_0}\Big(\frac{1}{\theta}\log(x)\Big) + \varepsilon \mathrm{d}x\Big) &\geq \mathbb{P}\Big(\sup_{x\in[0,M]} |F_{t_0}(x) - F(x)| \leq \varepsilon\Big)\\
&\geq 1 - 2e^{-2|t_0|\varepsilon^2}
\end{aligned}$$

for all $\theta > 0$. This means

$$\left[0, 1 + \int_1^{e^{\theta M}} 1 - F_{t_0}\left({}^1\!/\!{}_\theta \log(x)\right) + \varepsilon \mathrm{d}x\right]$$

is a one-sided confidence interval for $\phi_{a(s)}(\theta)$ with significance level $\alpha = 2e^{-2|t_0|\varepsilon^2}$. The integral simplifies to:

$$\begin{aligned}
\int_1^{e^{\theta M}} 1 - F_{t_0}\left(1/\theta \log(x)\right) + \varepsilon \mathrm{d}x &= (1+\varepsilon)(e^{\theta M}-1) - 1/|t_0| \int_1^{e^{\theta M}} \sum_{i=t_0}^{-1} \mathbf{1}_{\{e^{\theta a(s)} \leq x\}} \mathrm{d}x \\
&= (1+\varepsilon)(e^{\theta M}-1) - 1/|t_0| \sum_{s=t_0}^{-1} \int_1^{e^{\theta M}} \mathbf{1}_{\{e^{\theta a(s)} \leq x\}} \mathrm{d}x \\
&= (1+\varepsilon)(e^{\theta M}-1) - 1/|t_0| \sum_{s=t_0}^{-1} e^{\theta M} - e^{\theta a(s)} \\
&= (1+\varepsilon)(e^{\theta M}-1) - e^{\theta M} + 1/|t_0| \sum_{s=t_0}^{-1} e^{\theta a(s)} \\
&= \bar{A} - 1 + \varepsilon(e^{\theta M}-1).
\end{aligned}$$

Inserting the corresponding ε for a significance level α, the confidence interval becomes

$$\left[0, \bar{A} + \sqrt{\frac{-\log(\frac{\alpha}{2})}{2|t_0|}}(e^{\theta M}-1)\right].$$

The statistic P fulfills

$$\begin{aligned}
&\inf_{\theta>0} \mathbb{P}\Big(\bigcap_{(s,t)\in\Lambda} P(\bar{a})(s,t,\theta) \geq \phi_{A(s,t)}(\theta)\Big) \\
&= \inf_{\theta>0} \mathbb{P}\Big(\bigcap_{(s,t)\in\Lambda} \prod_{r=s+1}^{t} \bar{A} + \varepsilon(e^{\theta M}-1) \geq \prod_{r=s+1}^{t} \phi_{a(r)}(\theta)\Big) \\
&\geq \inf_{\theta>0} \mathbb{P}(\bar{A} + \varepsilon(e^{\theta M}-1) \geq \phi_{a(r)}(\theta)) \geq 1-\alpha
\end{aligned}$$

or equivalently

$$\sup_{\theta>0} \mathbb{P}\Big(\bigcup_{(s,t)\in\Lambda} P(\bar{a})(s,t,\theta) < \phi_{A(s,t)}(\theta)\Big) \leq \alpha.$$

□

4.2.3. Markov-Modulated Arrivals. This section considers a traffic class that no longer has i.i.d. increments. For this, consider a Markov chain $(Y_t)_{t\in\{t_0,\dots\}} \in \{0,1\}^{\mathbb{N}_0}$, also called the *signal*, with transition matrix

$$T = \begin{pmatrix} 1-\mu & \mu \\ \nu & 1-\nu \end{pmatrix}.$$

Denote the first state as *Off*-state, in which $Y_t = 0$ and the second state the *On*-state with $Y_t = 1$. The flow's increments are defined by

$$a(t) = X_t Y_t,$$

where $(X_t)_{t\in\{t_0,\dots\}}$ is a sequence of i.i.d. random variables. Further, assume there is a bandwidth limitation M on the X_t, as in the previous section. Although the X_t are i.i.d. the increments $a(t)$ are neither identically distributed nor stochastically independent, due to the Markov-modulation. This model generalizes the well-known and popular Markov-modulated on-off traffic model [**5**]. Choose for Y_{t_0} the chain's stationary distribution:

$$\pi_0 = \frac{\nu}{\mu+\nu}, \qquad \pi_1 = \frac{\mu}{\mu+\nu}.$$

Indeed, it is easy to see that $\pi = (\pi_0, \pi_1)$ is a left eigenvector of T.

The next lemma shows the monotonicity of $\phi_{A(s,t)}(\theta)$ with respect to parameters μ and ν.

LEMMA 4.6. *For all $(s,t) \in \Lambda$ and $\theta > 0$ the implications*

$$\tilde{\mu} \geq \mu \quad \Rightarrow \quad \mathbb{E}(e^{\theta A_{\tilde{\mu},\nu}(s,t)}) \geq \mathbb{E}(e^{\theta A_{\mu,\nu}(s,t)})$$
$$\tilde{\nu} \leq \nu \quad \Rightarrow \quad \mathbb{E}(e^{\theta A_{\mu,\tilde{\nu}}(s,t)}) \geq \mathbb{E}(e^{\theta A_{\mu,\nu}(s,t)})$$

hold, where $A_{\mu,\nu}$ denotes a flow following the above model with parameters μ and ν.

PROOF. Since the X_t are i.i.d. the expression $\phi_{A_{\tilde{\mu},\nu}(s,t)}(\theta)$ depends on the Markov chain only via the number of visits in the *On*-state. Denoting by $O_{\mu,\nu}(t)$ the number of visits in the *On*-state for a Markov chain with parameters μ and ν up to time t, it suffices to show $\mathbb{E}(O_{\tilde{\mu},\nu}(t)) \geq \mathbb{E}(O_{\mu,\nu}(t))$. This is achieved by

$$\mathbb{E}(O_{\tilde{\mu},\nu}(t)) = \mathbb{E}\Big(\sum_{s=0}^{t} \tilde{Y}_s\Big) = \sum_{s=0}^{t} \mathbb{E}(\tilde{Y}_s) = (t+1)\frac{\tilde{\mu}}{\tilde{\mu}+\nu} \geq (t+1)\frac{\mu}{\mu+\nu} = \mathbb{E}(O_{\mu,\nu}(t)).$$

The inequality $\mathbb{E}(O_{\mu,\tilde{\nu}}(t)) \geq \mathbb{E}(O_{\mu,\nu}(t))$ follows in the same way. $\square$

The next theorem constructs a P, fulfilling Condition (4.1), for unknown μ, ν, and distribution X_t. For this, multiple statistics are combined into P. Denote by $T_{i,j}$ the observed number of transitions from state i to state j in an arrival sample $\bar{a} = (a_{t_0}, \ldots, a_{-1})$; for example, $T_{0,1}$ denotes the number of transitions from the *Off*- to the *On*-state in $\bar{a}$. Denote further the observed number of visits in the *Off*-state in $\bar{a}$ by O^c.

THEOREM 4.7. *Define for $\mathbb{P}(X_s = 0) = 0$, a confidence level $\alpha = \alpha_\mu + \alpha_\nu + \alpha_d$, and a sample $\bar{a} = (a_{t_0}, \ldots, a_{-1})$ with $O \neq 0$ the statistics*

$$\mu_u := \beta^{-1}(1-\alpha_\mu; O^c - T_{0,0} + 1, T_{0,0}),$$
$$\nu_l := \beta^{-1}(\alpha_\nu; O - T_{1,1}, T_{1,1} + 1),$$

where β^{-1} is the inverse of the beta distribution; further, define the transition matrix

$$T^* = \begin{pmatrix} 1-\mu_u & \mu_u \\ \nu_l & 1-\nu_l \end{pmatrix}$$

and

$$A^* = \frac{1}{O} \sum_{s:Y_s=1} e^{\theta X_s} + \left(\frac{-\log\left(\alpha_d/2\right)}{2|O|}\right)^{1/2} \left(e^{\theta M} - 1\right).$$

Then the statistic $P : \mathbb{R}^{|t_0|} \to \mathcal{F}$, defined by

$$P(\bar{a})(s,t,\theta) := A^* \frac{\bar{x}_{On} \vee \bar{x}_{Off}}{\bar{x}_{On} \wedge \bar{x}_{Off}} \rho(\bar{E}T^*)^{t-s-1},$$

satisfies the condition of Theorem 4.2. The quantities $\bar{E}$, $\bar{x}_{On}$, and $\bar{x}_{Off}$ are defined in the proof.

PROOF. Fix some arbitrary $t \in \mathbb{N}_0$.

Step 1: It is to show that $[0, \mu_u]$ and $[\nu_l, 1]$ are confidence intervals for μ and ν at the confidence levels α_μ and α_ν. These are the Clopper-Pearson intervals [**44**]. They are constructed as follows: Interpret $T_{0,0}$ as the outcome of a $Bin(1-\mu, O^c)$-distributed random variable, denoted by X. Then it holds

$$\mathbb{P}(X \leq k) = \sum_{i=0}^{k} \binom{O^c}{i} (1-\mu)^i \mu^{O^c - i} = \beta(\mu;\, O^c - k,\, k+1).$$

One needs to find μ_u such that a random variable $X \sim Bin(1-\mu_u, O^c)$ fulfills $\alpha_\mu = \mathbb{P}_{\mu_u}(X \geq T_{0,0})$, or equivalently

$$1 - \alpha_\mu \stackrel{!}{=} \mathbb{P}_{\mu_u}(X \leq T_{0,0} - 1) = \beta(\mu_u;\, O^c - T_{0,0} + 1, T_{0,0}).$$

Solving for μ_u leads to

$$\mu_u = \beta^{-1}(1 - \alpha_\mu;\, O^c - T_{0,0} + 1, T_{0,0}).$$

This results in $\mathbb{P}(\mu_u < \mu) \leq \mathbb{P}(X \geq T_{0,0}) = \alpha_\mu$.

Similarly one proceeds to find ν_l: The number $T_{1,1}$ is interpreted as successes in a $Bin(1-\nu, O)$-distributed random variable, denoted by Z. Now ν_l must be found such that for $Z \sim Bin(1-\nu_l, O)$ it holds

$$\alpha_\nu = \mathbb{P}_{\nu_l}(Z \leq T_{1,1}) = \beta(\nu;\, O - T_{1,1}, T_{1,1} + 1).$$

This is solved by

$$\nu_l = \beta^{-1}(\alpha_\nu;\, O - T_{1,1}, T_{1,1} + 1),$$

resulting in $\mathbb{P}(\nu_l > \nu) \leq \mathbb{P}(Z \leq T_{1,1}) = \alpha_\nu$.

Step 2: Assume for a moment that

$$\begin{aligned} \mu_u &> \mu, \\ \nu_l &< \nu, \\ A^* &> \phi_{X_s}(\theta) \end{aligned}$$

holds and define by Y^* a Markov chain with transition matrix T^*. For all $(s,t) \in \Lambda$ it holds by the previous lemma

$$\mathbb{E}(e^{\theta A(s,t)}) \leq \mathbb{E}(e^{\theta \sum_{r=s+1}^{t} X_r Y_r}) \leq \mathbb{E}(e^{\theta \sum_{r=s+1}^{t} X_r Y_r^*}).$$

If the last expression can be bounded by $P(\bar{a})(s,t,\theta)$, the bounding of the MGF $\phi_{A(s,t)}(\theta)$ will be successful. With such a bound it will follow that

$$\begin{aligned} \alpha = \alpha_\mu + \alpha_\nu + \alpha_d &\geq \mathbb{P}(\mu_u < \mu \cup \nu_l > \nu \cup A^* > \phi_{X_s}(\theta)) \\ &\geq \mathbb{P}\Big(\bigcup_{s \leq t} \mathbb{E}(e^{\theta A(s,t)}) > P(\bar{a})(s,t,\theta) \Big) \end{aligned}$$

for any $t \in \mathbb{N}_0$ and the theorem be proven.

Step 3: The missing inequality builds upon the estimator for bandwidth limited traffic. Define a new Markov-modulated arrival with constant rate arrivals $X_s^* = {}^1\!/\!_\theta \log(A^*)$ in the *On*-state and T^* as transition matrix. Such a Markov-modulated arrival has a (σ, ρ)-bound given by $P(\bar{a})(s,t,\theta)$, as shown in Appendix

10.2; further,

$$\mathbb{E}(e^{\theta \sum_{r=s+1}^{t} X_r Y_r^*}) = \sum_{k=0}^{t-s} \mathbb{P}(O^*(s,t)=k)\phi_{X_s}(\theta)^k \leq \sum_{k=0}^{t-s} \mathbb{P}(O^*(s,t)=k)\phi_{X_s^*}(\theta)^k$$
$$= \mathbb{E}(e^{\theta \sum_{r=s+1}^{t} X_r^* Y_r^*}) \leq P(\bar{a})(s,t,\theta).$$

Here O^* denotes the number of *On*-states for the new Markov chain. □

The above examples show that the technically hard part of applying StatNC is the construction of the estimators. Taking care of other – potentially more complex – arrival processes is just a question of finding the corresponding P.

4.3. Numerical Evaluation

This section compares the statistical network calculus with its stochastic counterpart. For this, the costs of involving statistics are investigated (in terms of looser bounds). Furthermore, properties of StatNC that the SNC lacks are studied. These are its dynamic view on the measurements and its robustness against false assumptions.

4.3.1. The Price of StatNC. The first scenario asks if the additional uncertainty resulting from the statistical part of the performance bounds is acceptable. To that end, the smallest x is calculated such that the guarantee $\mathbb{P}(\mathfrak{b}(t) > x) \leq \varepsilon$ still holds. For a perfect bound, one would encounter after a large number of simulations, say K, that roughly $K \cdot \varepsilon$ of them produce a backlog greater than N at time t. In this evaluation the backlog process is simulated K times and the empirical distribution of the observed backlogs is compared with the x found by SNC and StatNC. The bounds are the better, the closer they lie to the $(1-\varepsilon)$-quantile of the empirical backlog distribution.

Here a Markov-modulated arrival process, as described in Section 4.2.3, is considered. The X_t are exponentially distributed, but capped by a bandwidth limitation M. The parameter λ of the exponential distributions is chosen to be 0.2, while the bandwidth limitation is set to $M = 20$. This means a hypothetical access link is maximally utilized at 25%. The transition probabilities of the Markov chain are set to $\mu = 0.1$ and $\nu = 0.1$. The service element offers a constant rate of $u = 5$; hence, during the *On*-state the utilization, considering the bandwidth limitation, peaks to $\frac{1}{\lambda u}(1 - e^{-\lambda M}) \approx 98\%$. Taking into account the Markov-modulation the average utilization is $\approx 49\%$. Backlog bounds for SNC are computed by Theorem 2.39. The StatNC bounds are computed according to Section 4.2.3. For illustration $K = 10^6$ runs of this system's backlog are evaluated at time slot 100 (at which time the initial distribution of the Markov chain fades out and steady state is reached). Figure 4.1 shows the empirical distribution function of the backlog. The bounds are calculated for a violation probability of $\varepsilon = 10^{-4}$. Both bounds are reasonably close to the $(1-\varepsilon)$-quantile. But, even more importantly, the bounds are close to each other. This demonstrates that the price for using StatNC is not too high.

4.3.2. Exploiting the Dynamic Behavior of StatNC. This scenario focuses on StatNC's dynamic point of view. The statistic uses a sliding window over the last t_0 observations (as discussed in Section 4.1). This leads to a subsampling that eventually forgets old measurements and learns from new arrivals. In this way the observation window tracks changes in the arrival process on longer timescales.

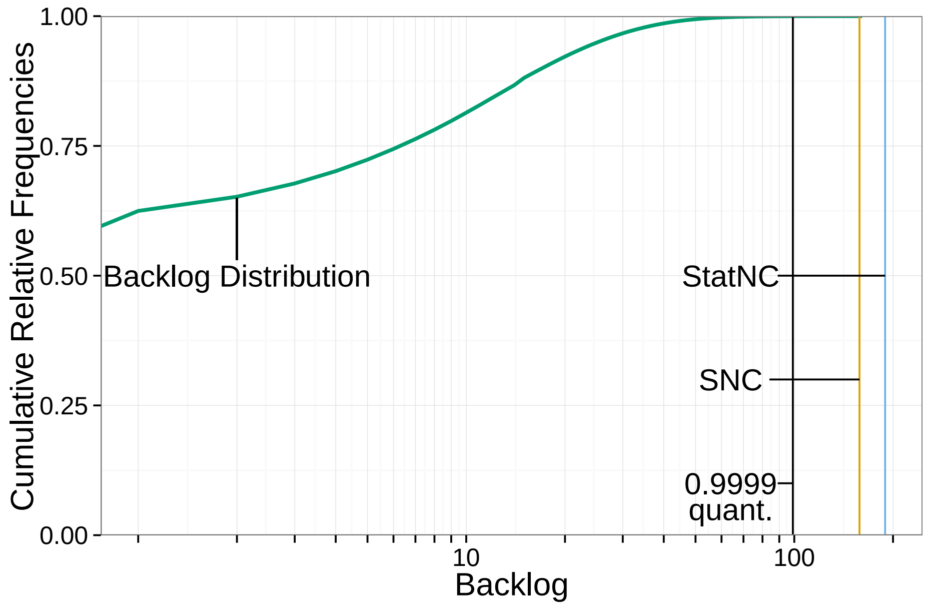


FIGURE 4.1. StatNC and SNC backlog bounds as well as the empirical backlog distribution of the backlog measured at time $t = 100$ for $K = 10^6$ simulation runs.

Such changes can stem from seasonal effects such as the time of day. One example is a non-stationary flow, whose increments diminish over time. See also Example 2.8 and the work by Becker et al. [**17**], which applies SNC to transient phases of queueing systems. Standard SNC has problems in capturing the non-stationary behavior, as it lacks the corresponding adaptability. Consequently the bounds become looser, when t increases. StatNC, on the other hand, readjusts by discarding old increments as time passes.

To investigate this effect a Markov-modulated arrival process, similar to the previous one, is used. The signal space consists of the states *High* and *Low*. For both of these, arrivals are drawn from an exponential distribution with a parameter λ_{Y_t} (and then capped by M); here, the parameter λ_{Y_t} depends on the state of the Markov chain (*high* or *low*).

This scenario uses the estimators from Subsections 4.2.1 and 4.2.2 and *not* the estimator presented in Subsection 4.2.3. Instead of "learning" the Markov chain itself, StatNC uses the observation window to track changes of states. Transition probabilities of $\mu = 0.001$ and $\nu = 0.001$ create a non-stationary behavior of the arrival process. The arrival rates are $\lambda_{Low} = 5$ and $\lambda_{High} = 0.2$ and the bandwidth limitation is $M = 10$. With a service rate of $u = 5$ this results in a utilization of 4% in the *low*-state and 86% in the *high*-state. The simulations start the arrival process at time $t_0 = -1000$ to provide an initial observation window for StatNC. A typical run of this scenario is plotted in Figures 4.2 (for the exponential traffic estimator) and 4.3 (for the i.i.d. bandwidth-limited estimator). In addition to the bounds the plots also show the simulated backlog process over time. Due to their

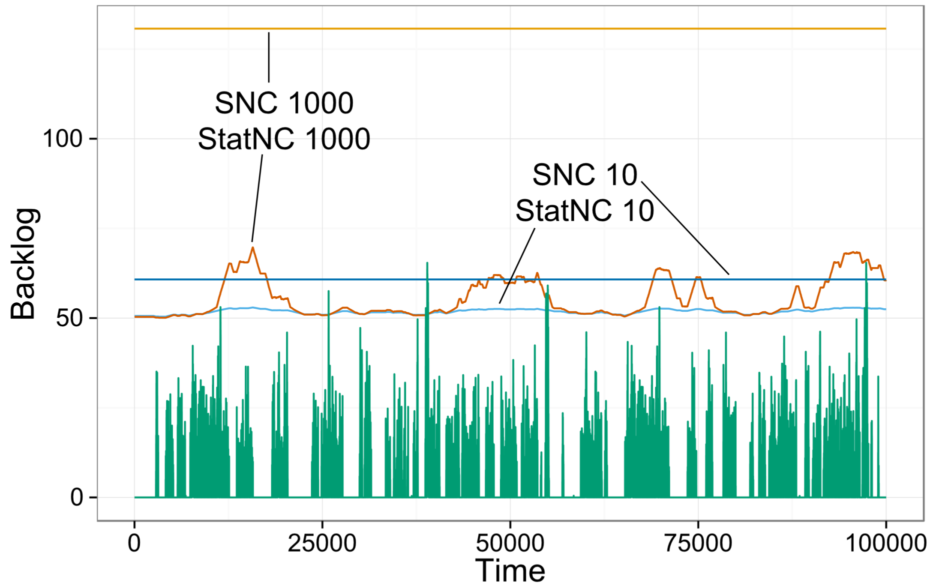


FIGURE 4.2. The backlog process for a typical simulation run as well as StatNC and SNC-bounds for t = 10, 1000. Here, the parametric estimator of Subsection 4.2.1 was used for the StatNC bounds.

dynamic nature, the StatNC bounds also evolve over time. They are computed for a violation probability of $\varepsilon = 10^{-4}$ and for a time that lies t time slots after the point they have been computed. The results correspond to $t = 10$ and $t = 1000$ time slots, representing a short and long prediction horizon, respectively.

The StatNC bounds react and ultimately adapt to the observed arrivals: If the arrival intensity is high (indicated by larger backlogs), the statistical bounds also increase, whereas they decrease, when the Markov chain changes to the *low*-state. The StatNC bounds track the changes of states with some delay, since old measurements need to be discarded first. The prediction horizon results in higher bounds for larger t irrespective of the bounding method. For comparison the SNC-bounds are calculated for an exact model of the Markov chain (Lemma 10.6). Although the SNC-bounds use complete information and perfect modeling they lie far above the StatNC bounds. The StatNC bounds perform better and are not violated too often (i.e., they are in accordance with the violation probability $\varepsilon = 10^{-4}$).

Another effect observed when comparing the two plots with each other, is the value of additional modeling information. In Figure 4.2 the estimator uses knowledge about the type of distribution (i.e., that they are exponentially distributed), whereas the estimator in Figure 4.3 does not. For the same run the StatNC bounds of Figure 4.2 perform moderately better. Taking more assumptions about the arrivals into account, however, bears the risk of making *false* assumptions. These, in turn, can lead to false bounds, as seen in the next subsection.

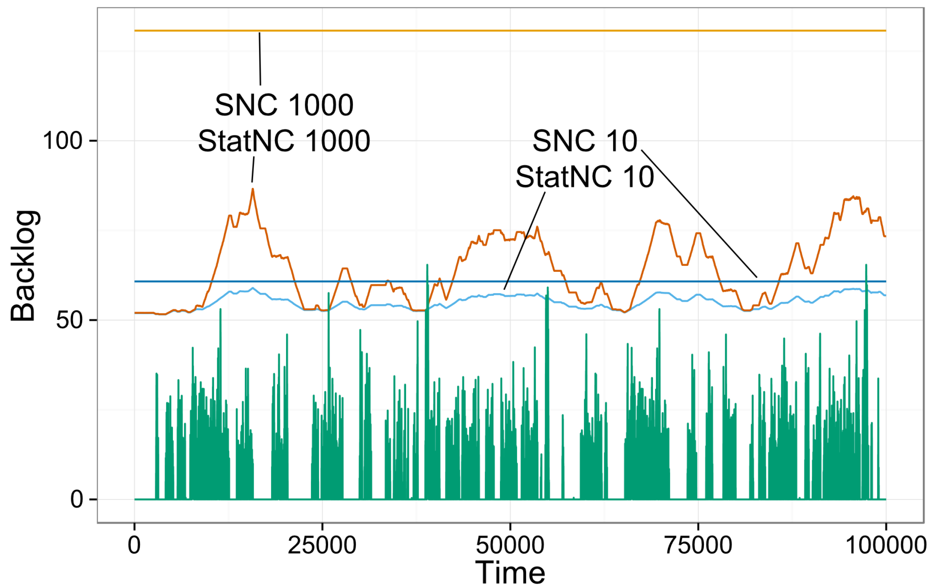


FIGURE 4.3. The backlog process for a typical simulation run as well as StatNC and SNC-bounds for $t = 10, 1000$. Here, the non-parametric estimator of Subsection 4.2.2 was used for the StatNC bounds.

4.3.3. Robustness of StatNC. This scenario focuses on the robustness of StatNC and SNC-bounds against false assumptions. This reveals another feature of StatNC when using the estimator of Subsection 4.2.2: StatNC is sufficient with rather few assumptions about the arrivals and consequently achieves a flexibility that SNC is missing.

To illustrate this, SNC deliberately makes a false assumption about the distribution of the arrivals' increments. The increments are independent, Pareto-distributed with parameters x_{min} and m, again capped by the bandwidth limitation M; yet, for the calculation of the SNC-bound the increments are assumed to be exponentially distributed with parameter λ (also capped by M). The rate parameter λ is set such that the expectations of the Pareto-distributed arrivals and the assumed exponentially distributed arrivals coincide. The expectation of a truncated, exponentially distributed random variable with parameter λ is $1/\lambda(1-e^{-\lambda M})$. Further the expectation of a truncated Pareto distribution with parameters x_{min} is

$$\begin{cases} x_{min} + \frac{x_{min}}{1-m} \cdot \left(\left(\frac{x_{min}}{M} \right)^{m-1} - 1 \right) & \text{if } m \neq 1 \\ x_{min} + x_{min} \log \left(\frac{M}{x_{min}} \right) & \text{if } m = 1. \end{cases}$$

The fitting λ is found by numerically solving

$$\begin{cases} \frac{1}{\lambda} \left(1 - e^{-\lambda M}\right) \overset{!}{=} x_{min} + \frac{x_{min}}{1-m} \cdot \left(\left(\frac{x_{min}}{M} \right)^{m-1} - 1 \right) & \text{if } m \neq 1 \\ \frac{1}{\lambda} \left(1 - e^{-\lambda M}\right) \overset{!}{=} x_{min} + x_{min} \log \left(\frac{M}{x_{min}} \right) & \text{if } m = 1. \end{cases}$$

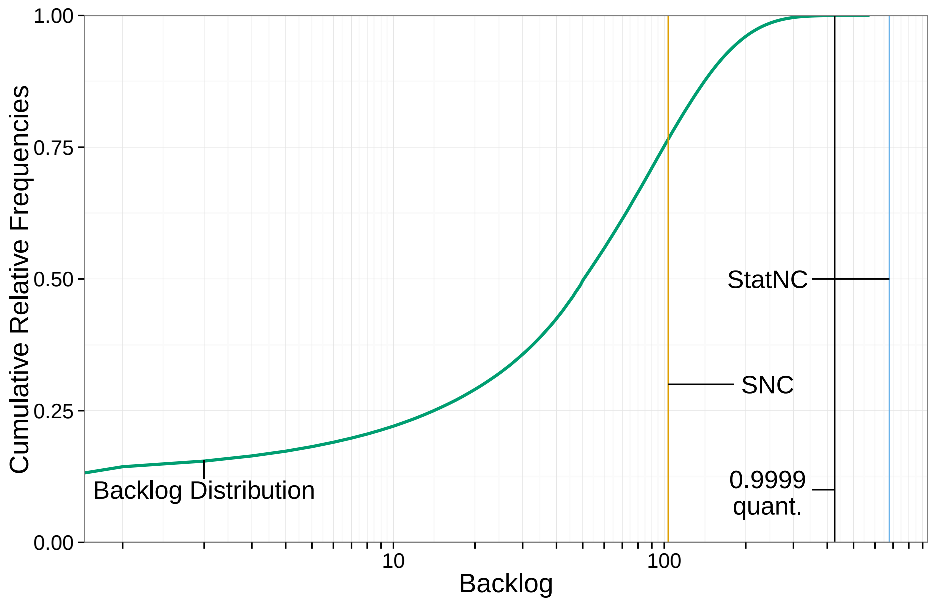


FIGURE 4.4. Under a false distribution assumption (exponential instead of Pareto-distributed increments), SNC delivers a grossly invalid bound whereas StatNC remains correct.

Figure 4.4 shows how false assumptions on the arrival process lead to disastrous result: As in the first scenario, the empirical distribution of the backlog is displayed and compared with the bounds calculated by StatNC and SNC. The parameters are $x_{min} = 1$, $m = 1$, $M = 55$ with a violation probability of $\varepsilon = 10^{-4}$ at time $t = 100$. The plot shows the empirical backlog distribution for 10^6 simulation runs, which in turn results in a tight 10^{-4} bound to be violated 100 times in expectation. The SNC-bound however is broken by 234,526 runs, i.e., in approximately 23% of the simulations. This lies far below the empirical $(1 - \varepsilon)$-quantile, the location of a “sharp” bound. SNC is way too optimistic and hence rendered useless. In contrast, the StatNC-bound remains valid and stays reasonably close to the empirical quantile.

4.4. Confidence Level α

This section discusses the choice of α in the used statistics. In Theorem 4.2 the confidence level α appears as a linear term as well as inside the estimator. In all considered estimators the linear influence of α exceeds that of the estimator by far. Following this insight α should be rather small to achieve good performance bounds.

Instead of a rigorous optimization this section gives a rule of thumb on how to pick α. The focus lies on the nonparametric estimator of Subsection 4.2.2, while the other estimators presented in Section 4.2 show a very similar behavior.

α	10^{-10}	10^{-100}	10^{-1000}	10^{-10000}
$q(\alpha)$	$\approx 1.5 \cdot 10^{-5}$	$\approx 4 \cdot 10^{-5}$	$\approx 1.5 \cdot 10^{-4}$	$\approx 4 \cdot 10^{-4}$

TABLE 1. Values for $q(\alpha)$ with $\varepsilon_{viol} = 10^{-4}$ and $|t_0| = 10^3$.

For a service element with constant rate u Theorem 4.2 gives

$$\begin{aligned}\mathbb{P}(\mathfrak{b}(t) > x) &\leq \alpha + e^{-\theta x} \sum_{s=0}^{t} P(\bar{a})(s,t,\theta)\phi_{U(s,t)}(-\theta) \\ &= \alpha + e^{-\theta x} \sum_{s=0}^{t} (\bar{A} + \varepsilon_\alpha(e^{\theta M} - 1))^{t-s} e^{-\theta u(t-s)} \qquad (4.2)\end{aligned}$$

and if the summands are less than 1, the bound becomes

$$\mathbb{P}(\mathfrak{b}(t) > x) \leq \alpha + e^{-\theta x}(1 - e^{-\theta u}(\bar{A} + \varepsilon_\alpha(e^{\theta M} - 1))^{-1}.$$

The condition for bounding the sum by its series translates to

$$(\bar{A} + \varepsilon_\alpha(e^{\theta M} - 1))e^{-\theta u} \overset{!}{<} 1.$$

This is solved for α_c by

$$\alpha \overset{!}{>} 2e^{-(\frac{e^{\theta u} - \bar{A}}{e^{\theta M} - 1})^2 2|t_0|} =: \alpha_c.$$

Taking into account the possible values for α the performance bound becomes:

$$\mathbb{P}(\mathfrak{b}(t) > x) \leq \inf_{\alpha > \alpha_c} \alpha + e^{-\theta x}\big(1 - e^{-\theta u}(\bar{A} + \varepsilon_\alpha(e^{\theta M} - 1))\big)^{-1} =: \varepsilon_{viol}.$$

If the violation probability is fixed to ε_{viol}, the smallest x fulfilling the above inequality is of interest. This x in turn depends on the choice of α. Solving the above for x_α results in

$$x_\alpha = -\frac{1}{\theta} \log\left(\varepsilon_{viol}(1 - e^{-\theta u}\bar{A}) - \alpha(1 - e^{-\theta u}\bar{A}) - q(\alpha)(e^{\theta M} - 1)e^{-\theta u}\right)$$

with $q(\alpha) := (\varepsilon_{viol} - \alpha)\sqrt{\frac{-\log(\alpha/2)}{|t_0|}}$ for all $\alpha > \alpha_c$. To minimize x the expressions inside the logarithm must become large. Under the stability condition $e^{\theta u} > \bar{A}$ the contribution of $\alpha(1 - e^{-\theta u}\bar{A})$ is negative. Hence, for this term α should be chosen as small as possible. On the other side the function q increases for small α. Overall the range of the function q lies in $(0, \infty)$ and it diverges to infinity for $\alpha \to 0$, while it converges to zero for $\alpha \to \varepsilon_{viol}$. Due to the logarithm under the square-root, $q(\alpha)$ remains relatively unchanged for most choices of α. Table 1 presents the differences in q for small α and $\varepsilon_{viol} = 10^{-4}$. These values of α lie far beyond the numerical precision of usual computers.

This essentially allows to insert the value of $\alpha_c = \alpha$ for the above infimum and bound $q(\alpha)$ from above. In this example $Q = 4 \cdot 10^{-4}$ could be such a bound. This reduces the contribution of the summand $\alpha(1 - e^{-\theta c}\bar{A})$ and leads to the bound

$$N_Q = -\tfrac{1}{\theta} \log\left((\varepsilon_{viol} - \alpha_c)(1 - e^{-\theta u}\bar{A}) - Q(e^{\theta M} - 1)e^{-\theta u}\right).$$

Often the parameter α can be reduced beyond α_c to achieve even better bounds for a fixed time t. However, by doing so the summands in Equation (4.2) do not converge anymore. As a consequence the optimal choice of α depends on t.

CHAPTER 5

Sample Path Backlog Bounds

This chapter presents joint work with J. Schmitt [**13**].

When moving from deterministic bounds of the form

$$\mathfrak{b}(t) \leq x \qquad \text{for all } t \in \mathbb{K} \tag{5.1}$$

to stochastic bounds

$$\mathbb{P}(\mathfrak{b}(t) > x) \leq \varepsilon \qquad \text{for all } t \in \mathbb{K}, \tag{5.2}$$

the bound's quality changes in two ways. First, probabilities are introduced. This allows the backlog bound to be invalid with a residual error probability. This is justified by a dramatic improvement on the bound x (e.g., [**106**]). The second change of quality is the bound's validity with respect to t. In the deterministic statement (5.1) the bound is valid for all $t \in \mathbb{K}$ simultaneously. The stochastic bounds (5.2), however, provide a set of inequalities of which each depends on t. The difference becomes clear, if one asks "Will the buffer exceed x in the next 23 time steps?" The deterministic bound (5.1) gives a direct answer. In the stochastic scenario the question translates into the expression

$$\mathbb{P}(\sup_{0 \leq t \leq 23} \mathfrak{b}(t) > x) =: p_{23}, \tag{5.3}$$

which is not covered by Equation (5.2). One might even go a step further and ask, "Will the buffer *ever* exceed x?" The deterministic answer remains unchanged, whereas the stochastic scenario tries to bound $\mathbb{P}(\sup_{t \geq 0} \mathfrak{b}(t) > x) =: p_\infty$. In a stationary system the lemma of Borel-Cantelli states $p_\infty \in \{0, 1\}$; hence, any efforts to obtain a differentiated answer for the above question must be in vain.

This chapter discusses sample path backlog bounds as in Equation (5.3). The easiest approach to construct a sample path backlog bound is to combine pointwise performance bounds via Boole's inequality:

$$\mathbb{P}(\max_{0 \leq t \leq T} \mathfrak{b}(t) > x) \leq \sum_{t=0}^{T} \mathbb{P}(\mathfrak{b}(t) > x). \tag{5.4}$$

Here T is the length of the interval for which the backlog bound x has to be fulfilled.

Boole's inequality, however, becomes imprecise the more the considered events overlap. The events $\{\mathfrak{b}(t) > x\}$ in Equation (5.4) clearly overlap, as $\mathbb{P}(\mathfrak{b}(t) > x)$ depends on $\mathfrak{b}(t-1)$. For example, when $\mathfrak{b}(t-1) > x$ and the service element offers a maximal service of u per time step, it follows $\mathfrak{b}(t) > x - u$. Boole's inequality neglects these dependencies. This observation is supported by a linear grow of the backlog bounds for T, as shown in the numerical evaluation of this chaper (Section 5.4).

This chapter obtains finite sample path bounds that do not build upon Boole's inequality. It combines the system dynamics with extreme value theory (EVT), a tool mainly used in financial and actuarial mathematics [**131, 57, 55**]. This method naturally lends itself to the calculation of finite sample path backlog bounds; further it can deliver bounds that lie below the ones derived by an application of Boole's inequality. These results motivate the development of an alternative SNC,i.e., to recreate the network operations presented in Chapter 1 and 2. This is of particular interest, as the needed conditions differ from the ones needed to apply SNC.

Classical queueing theory is able to compute the steady state backlog distribution for simple source models [**109**]. Applying further techniques to this asymptotic domain such as deviations [**141**], local limit theorems [**126**], or extreme value theory [**57**], leads to approximations similar to Equation (5.3). This chapter, however, focuses on non-asymptotic bounds rather than asymptotic approximations for the backlog process. Furthermore, the aforementioned methods are typically tailored to certain assumptions on the arrival processes. In contrast to that, the results presented here follow the framework-oriented approach of SNC and keep the analysis as generic as possible.

In the works of Poloczek et al. [**42, 128, 127, 129**] the union bound is replaced with Doob's inequality. This requires the construction of appropriate martingales for the arrivals or the service process. However, the union bound replaced in these works is different from the one in Equation (5.4). Instead of constructing sample path backlog bounds Poloczek et al. are concerned with bounding the min-plus deconvolution (i.e., the last step in the proofs of Theorems 2.22 and 2.26, respectively).

Alternative solutions to improve the union bound are given in [**95, 159, 160**]. These results are, however, not directly applicable as they consider a finite underlying probability space. Yet, this is not the case for the union bounds considered in Equation (5.4).

5.1. Alternative bound

To avoid Boole's inequality the queueing system is analyzed in a way different from Chapter 2. There the backlog is derived from Lindley's equation and eventually cast into bounds on the arrivals moment generating function. The intuition in this chapter is the following: If (a constant rate) system experiences an extraordinary large backlog, there must have been an extraordinary large number of arrivals. If this was not the case, assumptions about the system's stability would ensure that the arrivals are processed quickly enough and no significant backlog would have accumulated. These considerations can also be extended to the case of varying service rates. This intuition is similar to the tailbounded approach of Sections 2.1-2.2. To experience a large backlog the participating random processes must deviate largely from their typical behavior. This view on the system is given by Theorem 5.1.

Assume for the rest of this chapter that $\mathbb{K} = \mathbb{N}_0$. Denote by E_T^τ the number of arrivals $a(t)$ up to time T exceeding a threshold τ, i.e.,

$$E_T^\tau := \sum_{t=1}^{T} \mathbf{1}_{\{a_t > \tau\}} \in \{0, \dots, T\}.$$

The arrivals exceeding τ form a subsequence of $(a(t))_{t \in \mathbb{N}_0}$, denoted by $(a(t_i))_{i \in \{0,\dots,E_T^\tau\}}$.

THEOREM 5.1. *Assume that a dynamic U-server has an incoming flow with increments $a(t)$. Then holds the following sample path backlog bound*

$$\begin{aligned}&\mathbb{P}(\max_{1\le t\le T}\mathfrak{b}(t)>x)\\ &\le 1-\sum_{n=0}^{T}\mathbb{P}(E_T^\tau=n)\mathbb{P}(\max_{1\le i\le n}a(t_i)\le\tau+\tfrac{x}{n}\cap\bigcap_{0\le s\le t\le T}U(s,t)\ge(t-s)\tau\mid E_T^\tau=n).\end{aligned}$$

for all $\tau\in[0,\infty)$; further, if U is stochastically independent of A, it holds

$$\begin{aligned}&\mathbb{P}(\max_{1\le t\le T}\mathfrak{b}(t)>x)\\ &\le 1-\mathbb{P}(\bigcap_{0\le s\le t\le T}U(s,t)\ge(t-s)\tau)\sum_{n=0}^{T}\mathbb{P}(E_T^\tau=n)\mathbb{P}(\max_{1\le i\le n}a(t_i)\le\tau+\tfrac{x}{n}\mid E_T^\tau=n).\end{aligned}$$

PROOF. Assume for a while that $E_T^\tau=n$ together with

$$\max_{1\le i\le n}a(t_i)\le\tau+\frac{x}{n}$$

and

$$U(s,t)\ge(t-s)\tau\qquad\text{for all }\ 0\le s\le t\le T$$

holds.

From the first assumption follows that

$$A(t)-A(s)\le\tau(t-s)+\frac{x}{n}(t-s\wedge n),$$

as $A(t)-A(s)$ contains $t-s$ many increments. For the backlog this results in

$$\begin{aligned}\mathfrak{b}(t)&\le\max_{0\le s\le t}\{A(t)-A(s)-U(s,t)\}\le\max_{0\le s\le t}\{A(t)-A(s)-(t-s)\tau\}\\ &\le\max_{0\le s\le t}\{\tfrac{x}{n}(t-s\wedge n)\}\le x\end{aligned}$$

for every $t\in\{1,\dots,T\}$. Hence, by the law of total probability it follows

$$\begin{aligned}&\mathbb{P}(\max_{1\le t\le T}\mathfrak{b}(t)>x)\\ &=1-\mathbb{P}(\max_{1\le t\le T}\mathfrak{b}(t)\le x)=1-\sum_{n=0}^{T}\mathbb{P}(E_T^\tau=n)\mathbb{P}(\max_{1\le t\le T}\mathfrak{b}(t)\le x\mid E_T^\tau=n)\\ &\le 1-\sum_{n=0}^{T}\mathbb{P}(E_T^\tau=n)\mathbb{P}(\max_{1\le i\le n}a(t_i)\le\tau+\tfrac{x}{n}\cap\bigcap_{0\le s\le t\le T}U(s,t)\ge(t-s)\tau\mid E_T^\tau=n).\end{aligned}$$

For the stochastic independent case $\mathbb{P}(A\cap B\mid C)=\mathbb{P}(B\mid C)\mathbb{P}(A\mid B\cap C)$ leads to

$$\begin{aligned}&\mathbb{P}(\max_{1\le t\le T}\mathfrak{b}(t)>x)\\ &=1-\mathbb{P}(\bigcap_{0\le s\le t\le T}U(s,t)\ge(t-s)\tau)\sum_{n=0}^{T}\mathbb{P}(E_T^\tau=n)\mathbb{P}(\max_{1\le i\le n}a(t_i)\le\tau+\tfrac{x}{n}\mid E_T^\tau=n).\end{aligned}$$

□

REMARK 5.2. The parameter τ, in the above bound, is left open as a free parameter. The above theorem needs no assumptions about the service or the arrivals having corresponding MGFs or being i.i.d. sequences. Further, the above bound is always less than 1, as expected of a violation probability.

REMARK 5.3. For the special case of $U(s,t) = U(t) - U(s)$ the probabilities simplify to

$$\begin{aligned}&\mathbb{P}(\max_{1\le t\le T} \mathfrak{b}(t) > x)\\ &\le 1 - \mathbb{P}(\min_{1\le t\le T} u(t) \ge \tau) \sum_{n=0}^{T} \mathbb{P}(E_T^\tau = n)\mathbb{P}(\max_{1\le i\le n} a(t_i) \le \tau + \tfrac{x}{n} \mid E_T^\tau = n),\end{aligned}$$

where $u(t)$ describes the increments $U(t) - U(t-1)$.

5.2. A Brief Introduction to EVT

The bound in Theorem 5.1 relies on the analysis of $\mathbb{P}(\max_{1\le t\le T} X_t \le x)$, where $(X_t)_{t\in\mathbb{N}}$ is a sequence of random variables and $x \in \mathbb{R}$. This probability is well studied in Extreme Value Theory (EVT) under different assumptions on $(X_t)_{t\in\mathbb{N}}$ (see for example [**131, 57, 55**]). The following small selection of results from EVT holds for nonnegative and i.i.d. $(X_t)_{t\in\mathbb{N}}$.

Denote by F_X the distribution of X_t, then $\mathbb{P}(\max_{1\le t\le T} X_t \le x) = (F_X(x))^T$. For simple distributions F_X the result in the previous theorem leads directly to finite sample path backlog bounds. However, taking the T-th power of F_X might be computationally unstable. The question arises if this expression can be approximated instead. By the Borel-Cantelli lemma the above probability converges to either zero or one; hence, some kind of scaling is needed to give proper information about the behavior of $\max X_t$. This leads to the next definition.

DEFINITION 5.4. A random variable X lies in the *domain of attraction* of some non-degenerate random variable G, written $X \in \mathcal{D}(G)$, if there exist sequences α_t, β_t such that

$$\mathbb{P}(\max_{1\le t\le T} X_t \le \alpha_T x + \beta_T) \xrightarrow{T\to\infty} G(x), \tag{5.5}$$

where the X_t are a sequence of i.i.d. variables with distribution F_X.

The next definition can be found in [**131, 57**]. Denote the right endpoint of some distribution F by $x_+ := \inf\{x : F(x) = 1\} \in [0,\infty]$.

DEFINITION 5.5. A distribution F is a *von Mises function*, if there exist $c > 0$, $z < x_+$, and some function f such that for all $z < x < x_+$ it holds

$$1 - F(x) = c\exp\left(-\int_z^x \frac{1}{f(u)}du\right).$$

Further, the *auxiliary function* f must fulfill

- $f(u) > 0$ for all $z < u < x_+$,
- f is absolutely continuous on (z, x_+),
- $\lim_{u\nearrow x_+} f'(u) = 0$.

Three examples of von Mises functions that appear in the context of queueing systems follow now.

EXAMPLE 5.6. The exponential distribution is a von Mises function with the constant auxiliary function $f(x) = \frac{1}{\lambda}$.

EXAMPLE 5.7. The Weibull distribution $\Psi_\alpha(x) = 1 - e^{-\lambda x^\alpha}$ is a von Mises function with the auxiliary function $f(x) = \frac{1}{\lambda\alpha} x^{1-\alpha}$ for $x > 0$. Special cases of the Weibull distribution are the exponential distribution ($\alpha = 1$) and the Rayleigh distribution ($\alpha = 2$). The Weibull distribution itself plays – together with the Gumbel- and the Fréchet-distribution – a central role in the theory of extreme values (see below). Further, the Weibull distribution does not have a closed form for its MGF. This makes its usage in SNC difficult.

EXAMPLE 5.8. The Erlang distribution $E(x) = 1 - e^{-\lambda x} \sum_{k=0}^{n-1} \frac{(\lambda x)^k}{k!}$ is a von Mises function with auxiliary function $f(x) = \sum_{k=0}^{n-1} \frac{(n-1)!}{(n-k-1)!} \lambda^{-(k+1)} x^{-k}$ for $x > 0$.

Further examples of von Mises functions include the gamma-, normal-, and lognormal-distributions. See [**57**] for a list of distributions and corresponding auxiliary functions.

Von Mises functions lie in $\mathcal{D}(\Lambda)$, where $\Lambda(x) := \exp(-e^{-x})$ is the Gumbel distribution (see [**131**]). The von Mises *condition* is another characterization of functions that lie in $\mathcal{D}(\Lambda)$.

DEFINITION 5.9. Define the function φ by

$$\varphi := -\log(-\log(F)).$$

A distribution F fulfills the *von Mises condition*, if

$$h(x) := \left(\frac{1}{\varphi'(x)}\right)' = -\log F(x) + \frac{F(x)F''(x)\log F(x)}{(F'(x))^2} \xrightarrow{x \to x_+} 1.$$

When some distribution F fulfills the von Mises Condition, define $g(x) := \sup_{y \geq x} |h(y)|$.

As Resnick shows in [**131**], the von Mises condition is fulfilled if and only if F is a twice differentiable von Mises function. Furthermore, if the von Mises condition is fulfilled, a uniform speed of convergence to Λ can be given. This result is essential for obtaining performance bounds from Theorem 5.1.

LEMMA 5.10. *Let the X_t be i.i.d. and their corresponding distribution F_X fulfill the von Mises Condition. Then it holds*

$$\mathbb{P}(\max_{1 \leq t \leq T} X_t \leq x\alpha_T + \beta_T) \leq \Lambda(x) - e^{-1} g(\beta_T)$$

for all $T \in \mathbb{N}$ and $x \geq 0$. Here $\varphi(\beta_t) := \log t$ and $\alpha_t := \frac{F(\beta_t)}{tF'(\beta_t)}$. The function g stems from Definition 5.9.

PROOF. See Proposition 2.18. in [**131**]. □

One of EVT's main results is the following: If a distribution lies in any domain of attraction, it lies either in $\mathcal{D}(\Lambda)$, $\mathcal{D}(\Phi_\alpha)$, or $\mathcal{D}(\Psi_\alpha)$, where $\Phi_\alpha(x) := e^{-x^{-\alpha}}$ is the Fréchet distribution [**54**]. The thesis at hand uses only results concerning the Gumbel distribution. Corresponding conditions for the convergence and the uniform speed of convergence towards Φ_α and Ψ_α are hence skipped here. Results about the uniform speed of convergence date back as early as 1970. See for example [**45, 53, 142, 1**], the textbooks [**131, 57**], and the references therein.

The von Mises condition takes for Theorem 5.1 a similar role as the existence of the moment generating function for Theorem 2.39. It is a sufficient condition for deriving performance bounds. Yet, Example 5.7 demonstrates that there are von

Mises functions that would be hard to analyze via MGF-calculus. Further examples include the Pareto distribution and power law distributions, which lie in $\mathcal{D}(\Phi_\alpha)$ and $\mathcal{D}(\Psi_\alpha)$, respectively.

In the context of queueing networks as described in Section 1.2 Theorem 5.1 cannot be applied directly. In SNC the network operations allow to reduce a network to the case of a single node and a single flow. The next section follows this approach and derives network operations in accordance to Theorem 5.1.

5.3. Sample Path Network Operations

LEMMA 5.11 (Output Bound). *Let U be the service of some node with input flow A. Denote by B the node's output. Then it holds*

$$\mathbb{P}(\max_{0\le t\le T} b(t) \le x) \ge \mathbb{P}(\max_{0\le t\le T} a(t) \le \tfrac{x}{T} + \tfrac{T+1}{T}\tau \cap \bigcap_{0\le s<t\le T} U(s,t-1) \ge (t-1-s)\tau)$$

for all $x \in [0,\infty)$ and $\tau \in [0,x]$.

PROOF. By the definition of service it holds

$$\begin{aligned} b(t) &= B(t) - B(t-1) \le B(t) - \min_{0\le s<t}\{A(s) + U(s,t-1)\} \\ &\le \max_{0\le s\le t-1}\{A(t) - A(s) - U(s,t-1)\} = \max_{0\le s<t}\{\sum_{r=s+1}^{t} a(r) - U(s,t-1)\} \\ &= \max_{0\le s<t}\{a(t) - U(s,t-1) + \sum_{r=s+1}^{t-1} a(r)\}. \end{aligned}$$

Fix some $\tau \in [0,x]$ and assume for a while that

$$a(t) \le \frac{x}{T} + \frac{T-1}{T}\tau \quad \text{for all } 0 \le t \le T$$

and

$$U(s,t-1) \ge (t-1-s)\tau \quad \text{for all } 0 \le s < t \le T$$

holds. Then it would follow

$$\begin{aligned} b(t) &\le \max_{0\le s<t\le T}\{a(t) - U(s,t-1) + \sum_{r=s+1}^{t-1} a(r)\} \\ &\le \max_{0\le s<t\le T}\{(\tfrac{x}{T} + \tfrac{T+1}{T}\tau)(t-s) - (t-1-s)\tau\} = \max_{0\le s<t\le T}\{(\tfrac{x}{T} - \tfrac{\tau}{T})(t-s) + \tau\} \\ &= \max_{0\le s<t\le T}\{\tfrac{x-\tau}{T}(t-s) + \tau\} \le x. \end{aligned}$$

Hence, it holds

$$\begin{aligned} \mathbb{P}(\max_{0\le t\le T} b(t) \le x) &\ge \mathbb{P}(\bigcap_{t=1}^{T} \max_{0\le s<t}\{a(t) - U(s,t-1) + \sum_{r=s+1}^{t-1} a(r)\} \le x) \\ &\ge \mathbb{P}(\max_{0\le t\le T} a(t) \le \tfrac{x}{T} + \tfrac{T-1}{T}\tau \cap \bigcap_{0\le s<t\le T} U(s,t-1) \ge (t-1-s)\tau) \end{aligned}$$

for all $\tau \in [0,x]$. □

The parameter $\tau \in [0,x]$ is subject to optimization and there is no gain in τ being greater than x. In the special case of $U(s,t) = u(t-s)$ the optimal choice is $\tau = x$ and it leads to the bound $\mathbb{P}(\max_{1\le t\le T} b(t) \le x) \ge \mathbb{P}(\max_{1\le t\le T} a(t) \le x)$.

LEMMA 5.12 (Multiplexing). *Let A_1 and A_2 be two arrivals. Denote by $a(t)$ the increments of $A_1 \oplus A_2$. It holds*

$$\mathbb{P}(\max_{0 \le t \le T} a(t) \le x) \ge \mathbb{P}(\{\max_{0 \le t \le T} a_1(t) \le x - \tau)\} \cap \{\max_{0 \le t \le T} a_2(t) \le \tau\})$$

for all $x \in [0, \infty)$ and $\tau \in [0, x]$.

PROOF. Choose an arbitrary $x > 0$. Then under the assumption of

$$\max_{0 \le t \le T} a_1(t) \le x - \tau \quad \text{for all } 0 \le t \le T,$$
$$\max_{0 \le t \le T} a_2(t) \le \tau \quad \text{for all } 0 \le t \le T$$

it holds

$$\max_{0 \le t \le T} a(t) \le x \quad \text{for all } 0 \le t \le T$$

for any $\tau \in [0, x]$. Moving to probabilities proves the lemma. □

The following network operation of subtracting crossflows is formulated for two flows only. This is no loss of generality, as any number of crossflows can be subtracted together from the service element via the above lemma.

LEMMA 5.13 (Leftover Service). *Consider the scenario as presented in Theorem 1.13 with two flows A_1 and A_2. It holds*

$$\mathbb{P}(\bigcap_{0 \le s \le t \le T} U_1(s,t) \ge (t-s)x) \ge \mathbb{P}(\max_{0 \le t \le T} a_2(t) \le \tau \cap \bigcap_{0 \le s \le t \le T} U(s,t) \ge (x+\tau)(t-s))$$

for all $x \in [0, \infty)$ and $\tau \in [0, \infty)$.

PROOF. Let $x \in [0, \infty)$. Assume for a while that

$$\max_{0 \le t \le T} a_1(t) \le \tau \quad \text{for all } 0 \le t \le T$$

and

$$U(s,t) \ge (x+\tau)(t-s) \quad \text{for all } 0 \le s \le t \le T$$

holds. Then it follows

$$\begin{aligned} U_1(s,t) &= \max\{0, U(s,t) - A(t) + A(s)\} = \max\{0, U(s,t) - \sum_{r=s+1}^{t} a(r)\} \\ &\ge \max\{0, (x+\tau)(t-s) - (t-s)\tau\} = x(t-s) \end{aligned}$$

for all $0 \le s \le t \le T$. The assertion follows then as in the previous proof. □

The situation simplifies in the special case $U(s,t) = u(t-s)$. If $x \in [0, u]$, the optimal τ is $u - x$. This results in

$$\mathbb{P}(\min_{0 \le t \le T} u_1(t) \ge x) \ge \mathbb{P}(\max_{0 \le t \le T} a(t) \le u - x).$$

The last network operation is the convolution of nodes.

LEMMA 5.14 (Convolution). *Consider two dynamic U_i-service elements ($i = 1, 2$) such that the first node's output is the second node's input. For $U = U_1 \otimes U_2$ and arbitrary $x \ge 0$ it holds*

$$\mathbb{P}(\bigcap_{0 \le s \le t \le T} U(s,t) \ge (t-s)x) \ge (\bigcap_{0 \le s \le t \le T} \max\{U_1(s,t), U_2(s,t)\} \ge (t-s)x).$$

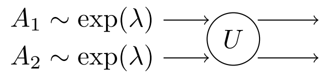


FIGURE 5.1. The example considered for the numerical evaluation. The service element has a constant rate of u and gives priority to A_2.

PROOF. Assume for a while that

$$U_1(s,r) \geq (r-s)x,$$
$$U_2(r,t) \geq (t-r)x$$

holds for all $r = s, \dots, t$. Then it would follow

$$U_1(s,r) + U_2(r,t) \geq (t-s)x$$

and hence

$$U(s,t) = \min_{s \leq r \leq t} \{U_1(s,r) + U_2(r,t)\} \geq (t-s)x.$$

Translating this implication into probabilities completes the proof. □

5.4. Numerical Evaluation

The following scenario compares the result of Theorem 5.1 to the one given by Boole's inequality (Equation (5.4)). Consider the scenario as in Figure 5.1: A constant rate server processes a low and a high priority flow,where A_1 is the flow of interest. For the sake of simplicity, both flows have independent, exponentially distributed increments with parameter λ, i.e.,

$$F_{a_1}(x) = F_{a_2}(x) = \begin{cases} 1 - e^{-\lambda x} & \text{if } x \in [0, \infty) \\ 0 & \text{if } x \in (-\infty, 0) \end{cases}.$$

The node's service rate is denoted by u.

5.4.1. MGF-Calculus Bound.

Denote the leftover service for A_1 at the node by U_1. First Theorem 2.37 is applied. The (σ, ρ)-bound for the arrivals is

$$\mathbb{E}(e^{\theta A_i(s,t)}) = \prod_{r=s+1}^{t} \mathbb{E}(e^{\theta a(r)}) = \left(\frac{\lambda}{\lambda - \theta}\right)^{t-s} = e^{\theta \rho(\theta)(t-s)},$$

where $\rho(\theta) := \frac{1}{\theta} \log(\frac{\lambda}{\lambda - \theta})$ and $\theta \in (0, \lambda)$. Hence, the high and low priority flows are $(0, \rho)$-bounded. Using Theorem 2.37 and that the constant rate service element is $(0, u)$-bounded results in $U_1 \succeq (0, \rho + u)$. The pointwise performance bound is given by Theorem 2.39. Together with Boole's inequality the finite sample path backlog bound reads:

$$\mathbb{P}(\max_{1 \leq t \leq T} \mathfrak{b}(t) \geq x)$$

$$= \inf_{0 \leq \theta < \lambda} \sum_{t=0}^{T} e^{-\theta x} \frac{1 - \left(\frac{\lambda}{\lambda - \theta}\right)^{2(t+1)} e^{-\theta(t+1)u}}{1 - \left(\frac{\lambda}{\lambda - \theta}\right)^{2} e^{-\theta u}} \leq \inf_{0 \leq \theta < \lambda} \sum_{t=0}^{T} e^{-\theta x} \frac{1 - e^{\theta(t+1)(2\rho(\theta)+u)}}{1 - e^{\theta(2\rho(\theta)+u)}}.$$

To compute a competitive backlog bound the above must be optimized by the free parameter θ.

5.4.2. Alternative Bound. Denote by E_T^τ the number of low priority arrivals that exceed the value τ and denote these arrivals by the subsequence $(a_1(t_i))_{i\in\{0,\dots,E_T^\tau\}})$. Then:

$$\begin{aligned}&\mathbb{P}(\max_{1\le t\le T}\mathfrak{b}(t)>x)\\ &\le 1-\mathbb{P}(\min_{1\le t\le T}u_1(t)\ge\tau)\sum_{n=0}^{T}\mathbb{P}(E_T^\tau=n)\mathbb{P}(\max_{1\le i\le n}a_1(t_i)\le\tau+\tfrac{x}{n}\mid E_T^\tau=n)\end{aligned}$$

and with Lemma 5.13 it follows

$$\begin{aligned}&\mathbb{P}(\max_{1\le t\le T}\mathfrak{b}(t)>x)\\ &\le 1-\mathbb{P}(\max_{1\le t\le T}a_2(t)\le u-\tau)\sum_{n=0}^{T}\mathbb{P}(E_T^\tau=n)\mathbb{P}(\max_{1\le i\le n}a_1(t_i)\le\tau+\tfrac{x}{n}\mid E_T^\tau=n)\\ &\le 1-\mathbb{P}(\max_{1\le t\le T}a_2(t)\le u-\tau)\sum_{n=0}^{T}\mathbb{P}(E_T^\tau=n)\cdot\mathbb{P}(\max_{1\le i\le n}a_1(t_i)\le\tfrac{x}{n}).\end{aligned}$$

The last step uses the memoryless property of the exponential distribution and the i.i.d. property of the arrivals.

Due to the arrival's nature one can either use the EVT-approximation or directly compute the above expression by $\mathbb{P}(\max_{1\le t\le T}a_i(t)\le x)=(F_{a_i}(x))^T$. In this evaluation both methods are followed to test the quality of the EVT-approximation. The exponential distribution fulfills the conditions of Lemma 5.10 with the norming sequences

$$\begin{aligned}\alpha_n&=-\frac{\log(1-e^{-1/n})}{\lambda},\\ \beta_n&=\frac{1}{n\lambda(e^{1/n}-1)}\end{aligned}$$

and

$$g(x)=-\frac{\log(1-e^{-\lambda x})}{e^{-\lambda x}}-1.$$

Inserting this into the Theorem 5.1 yields

$$\begin{aligned}\mathbb{P}(\max_{1\le t\le T}\mathfrak{b}(t)>x)\le 1-&(\exp(-e^{-\gamma_T(\lambda(u-\tau)+\log(1-e^{-T^{-1}}))})-\tilde{g}(T))\\ &\cdot\sum_{n=0}^{T}\mathbb{P}(E_T^\tau=n)(\exp(-e^{-\gamma_n(\lambda\frac{x}{n}+\log(1-e^{-n^{-1}}))})-\tilde{g}(n)),\end{aligned}$$

where

$$\tilde{g}(n):=\frac{1}{e\cdot n(1-e^{-n^{-1}})},\qquad \gamma_n:=n(e^{1/n}-1).$$

Similar to the MGF-bound a numerical optimization must be performed to achieve a competitive bound – in this case for the parameter $\tau\in[0,u]$.

5.4.3. Results. For the evaluation u is set to one. By varying λ different utilizations $u=\frac{2}{\lambda}$ of the node are considered. The plots show the minimal x such that $\mathbb{P}(\max_{1\le t\le T}\mathfrak{b}(t)>x)\le\varepsilon$ with $\varepsilon=10^{-6}$ (Figures 5.2 and 5.3) and $\varepsilon=10^{-9}$ (Figures 5.4 and 5.5). The results are dependent on the considered sample path length T. To find reasonable values for T the queueing system was simulated and

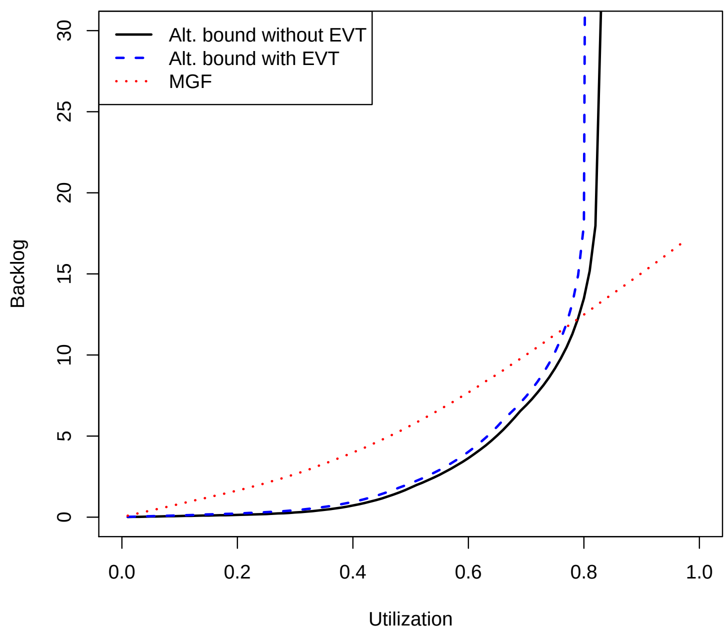
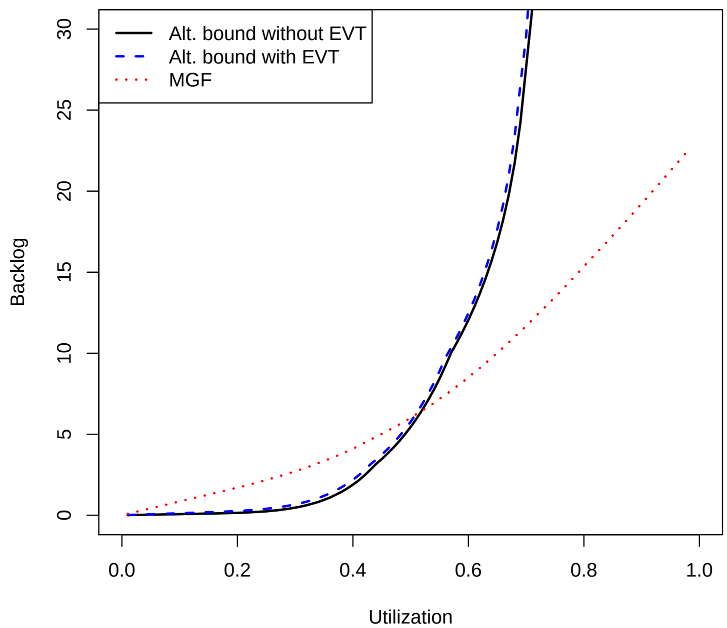


FIGURE 5.2. $T = 10$ and $\varepsilon = 10^{-6}$.

the duration of the backlogged periods observed. The starting point of a backlogged period is defined as the time step, in which the node starts to accumulate backlog and the endpoint is defined as the first time step thereafter, in which no backlog occurs at the node. In the simulations 100,000 backlogged periods were observed under different utilizations. For example, for a utilization of 80% an average period length of 3.2 is observed and 99.9% of the periods have a length shorter than 37. For this reason sample path bounds are plotted for the lengths $T = 10$, $T = 20$ and $T = 40$.

The results for T under different utilizations are displayed in Figures 5.2-5.5. Two observations are made. First, the results of EVT approximations (solid black lines) and direct computations (dashed blue line) are close to each other; second, the alternative method outperforms the MGF-method, given by the dotted red line, in the region of lower utilizations. However, the alternative method has for large T a tipping point after which, only by an immense increase of N the wished violation probability ε is achieved. Comparing the three methods under increasing T the MGF-method loses the least. All three methods are quite robust against the transition from a violation probability of 10^{-6} to 10^{-9}. However, the MGF-method looses a bit more here.



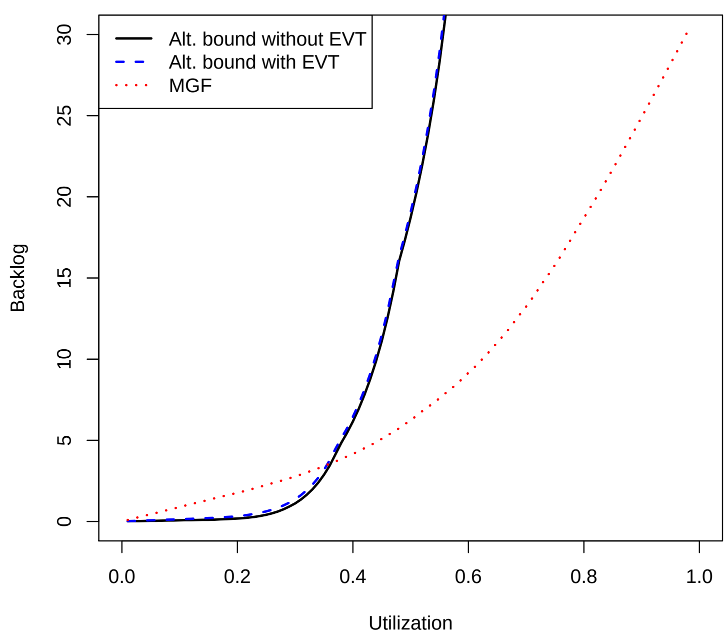


FIGURE 5.3. From top: $T = 20$, $T = 40$. $\varepsilon = 10^{-6}$.

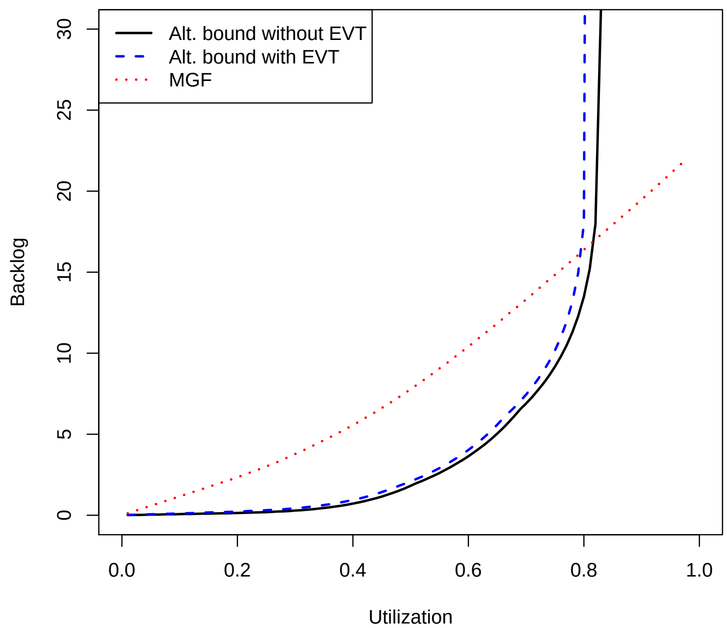


FIGURE 5.4. $T = 10$ and $\varepsilon = 10^{-9}$.

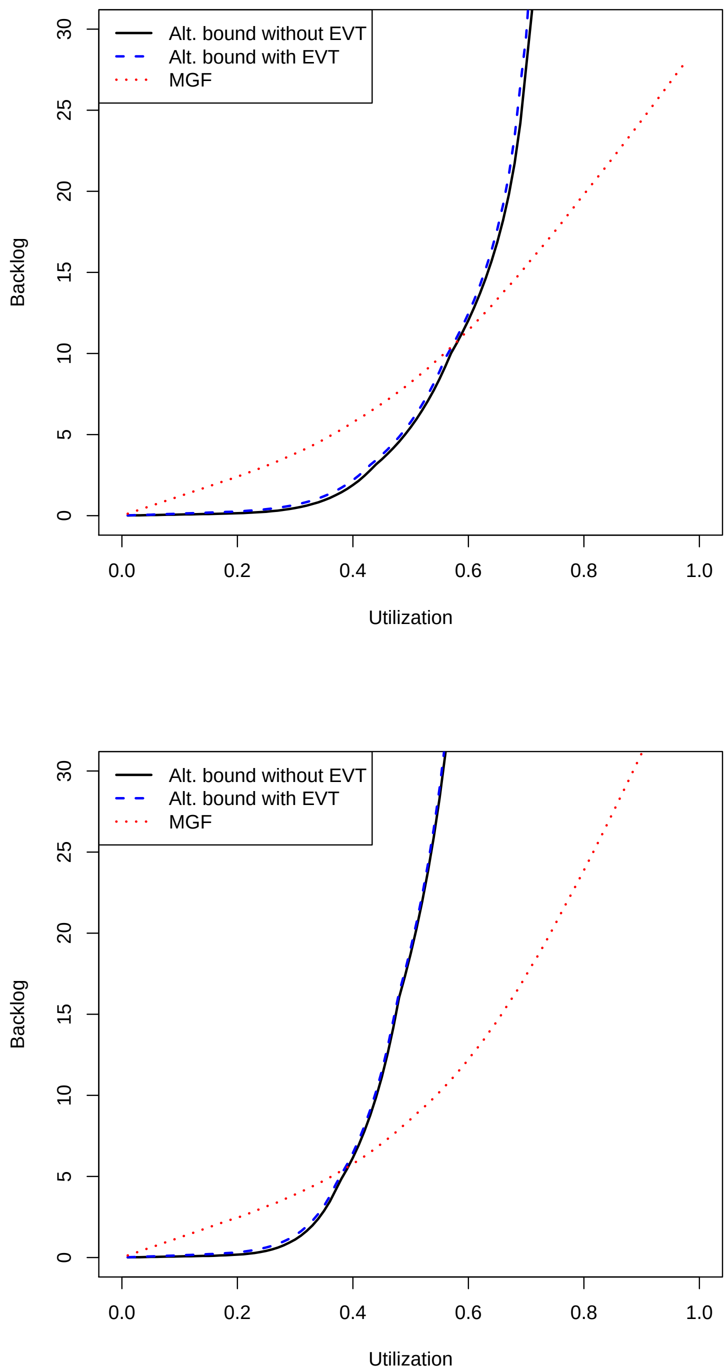


FIGURE 5.5. From top: $T = 20$, $T = 40$. $\varepsilon = 10^{-9}$.

Part 3

Window Flow Controller

Stochastic Network Calculus (SNC) has matured in recent years to provide an alternative method for the performance analysis of stochastic queueing systems (see [**88, 68, 43**]). Many results – foremost the network operations – have been transferred to the stochastic domain. The transfer of some results had been rather immediate, while others required considerable effort [**39**]. One major open challenge is the stochastic analysis of feedback systems. In communication networks such systems model for example window-based transport protocols like TCP [**83, 84, 28, 156, 146, 145, 77**]. While there are very elegant solutions for the window flow controller (WFC) in the deterministic setting [**3, 34, 101**], corresponding results in SNC are identified as a challenging open research question [**88, 66, 68, 41**].

In this part of the thesis window flow controller and their solution in deterministic network calculus is introduced in detail. This includes the well-known, univariate results, as presented first by [**4, 52**], and the solution for bivariate equations with $\mathbb{K} = \mathbb{N}_0$, as found in [**35**]. Section 6.3 further presents the framework of σ-additive operators, which also allows to model feedback systems. This framework in particular is used to analyze the special case of WFC systems with a fixed-delay element (see also [**139**]).

In Chapter 7 two methods are presented to reproduce the deterministic results in a stochastic setting. The first approach works under a subadditivity assumption for the involved service processes. With such a subadditive service description the stochastic analysis simplifies by far; however, this assumption also seems restrictive. To that end, a class of networks is identified that preserves the subadditivity assumption and thus remains analytically tractable.

The second approach aims at a general analysis. The key idea is to stochastically control how far the service deviates from being subadditive. This is cast into the setting of MGF-calculus as a further contribution to the violation probabilities of the performance bounds.

CHAPTER 6

Window Flow Control in Deterministic Network Calculus

Window flow controlled (WFC) systems model situations in which the amount of arrivals to a subsystem is strictly bounded. The idea is to separate the system into a throttle and a feedback loop. The throttle governs how much workload is admitted to the feedback loop. It blocks any traffic that would push the current load beyond a threshold that is called window size. As new arrivals are only allowed to enter the subsystem if there is enough "room", the throttle needs information about the feedback loop's departures. Receiving this information might be delayed, as it has to pass through service elements itself (e.g., the acknowledgements in transport protocols).

In the context of network calculus the most prominent example for WFC systems are window based transport protocols, like TCP [**156, 146, 145**]. The feedback loop is, in this case, the end-to-end channel between sender and receiver. The objective is to avoid congestion in the communication channel. Indeed, WFC systems only allow a certain amount of data – the window size – to be in transit. The metric of interest for this system is the end-to-end delay between sender and receiver, including both subsystems: the throttle and the feedback loop. See also the survey of Kafi et al. about congestion control protocols in wireless sensor networks [**90**].

The WFC model extends beyond the usage in transport protocols. Two examples: Consider an exit of a highway that ends at a traffic signal. On its lengths (in meters) the exit can only hold a certain amount of vehicles. Hence, it can be considered as a buffer to the traffic light, which is the service element. If more cars queue at the traffic light than fit on the exit lane, the congestion reaches the highway itself. In this scenario, the delay of acknowledgments is the time the cars need to cross after the beginning of a green phase. For more details on the management of traffic lights and road traffic in general, see [**97, 72, 155, 38, 61**].

The second example considers the storing of energy that is produced by renewable and typically unreliable sources [**60, 153, 158, 73**]. The methods to store energy differ in capacity, efficiency and costs [**73**]. Efficiency means here the waste of energy, while charging or discharging the energy buffer. Of course one want to use low-efficiency storages only if the high efficiency storages are full. In this scenario the high-efficiency storage represents the feedback system. The notion of "service" corresponds to the energy demand and the incoming arrival flow is the energy produced by renewable sources. Further, the backlog inside the feedback system corresponds to how far the storage is charged. Any energy that does not "fit" into the feedback loop must be buffered at the throttle and results in the usage

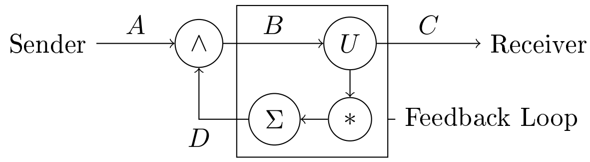


FIGURE 6.1. A window flow controlled system. The element denoted by $*$ is a placeholder.

of low efficiency storages. The goal in this case – quite contrary to the one in communication networks – is to neither let the buffer run dry, nor to fill it completely. In the former case the result would be an outage of power; whereas the latter case enforces either overcharging or the usage of low-efficient storage. A particularly interesting "energy storage" is the consumption of energy in households, see for example [**133, 89**]. Here the energy demand is influenced by dynamic electricity pricing.

Very similar examples can be constructed in the context of production networks (for example [**110, 22**]), in which raw materials must be stored, when not processed. Again the conflicting goals of availability and storage costs must be considered at the same time.

WFC systems and their treatment is closely linked to the development of network calculus itself. The analysis of WFC systems began with the work of Cruz [**46**]. There the results' applicability depends on the number of service elements: Feedback loops with more than two service elements must be recursively reduced to the two server case. Baccelli, Cohen, Olsder and Quadrat [**7**] consider feedback systems in the max-plus algebra. This algebra shares many properties with the framework of the later developed network calculus; indeed, in [**7**] the max-plus version of the subadditive closure appears already. Cruz and Okino [**52**], as well as Agrawal and Rajan [**4**] describe the window flow controller for the first time by the subadditive closure of service curves. Chang [**34**], and LeBoudec and Thiran [**23**] extend and refine the results of [**52, 4**]. In [**34**] the definition of maximal f-regulators is directly connected to WFC systems via the subadditive closure. Baccelli and Hong consider TCP as a max-plus linear system in [**8**]. Eventually the results had been consolidated in the textbooks of Chang [**35**] and LeBoudec and Thieran [**102**].

Despite the extensive treatment of WFC systems in deterministic network calculus, the corresponding analysis with stochastic network calculus remained an open problem ever since [**88, 66, 68, 41**].

This chapter presents the deterministic results of network calculus concerning WFC systems, including the univariate and the bivariate formulations; furthermore, the notion of σ-additive operators is introduced, which allows a third formulation of the feedback system. The notion of σ-additive operators is of particular interest, as it allows the commutation of service elements with fixed-delay elements. This in turn leads to a closed form of the subadditive closure, as shown in Section 6.4.

6.1. Introduction and Notations

The general framework of a WFC system is given in Figure 6.1. Ignoring the element Σ for a while, this system works as follows: The arrivals A reach a throttle before they are allowed to enter the system's feedback loop. The throttle, represented by a $\wedge$-symbol, takes A and a control flow D to produce the output

$$B(t) = A(t) \wedge D(t) \qquad \text{for all } t \in \mathbb{K}.$$

The throttle's output B is called the *effective arrival* to the system. It is processed with service curve U. Its output C plays two roles: 1) it is the data that arrives at the receiver and 2) it is processed into the control flow D. The processing of C is indicated by a placeholder element in Figure 6.1. Eventually, the control flow D is fed back into the throttle element $\wedge$; hence, the information of departures from U is brought back to the throttle element in form of flow D, which in turn allows new arrivals to U.

The element denoted by Σ is a window-element. In principle it just adds a positive constant Σ to its ingress flow. This has two effects on the system: 1) Without a window-element the system could never start running. All initial flows would be equal to zero and hence the throttle would never admit any workload to the feedback loop. 2) The amount of backlog inside the feedback loop never exceeds the window size Σ. This is seen by

$$\mathfrak{b}_U(t) := B(t) - C(t) \leq (A(t) \wedge C(t) + \Sigma) - C(t) \leq \Sigma.$$

This strict guarantee on the amount of backlog inside the feedback loop gives the WFC its modeling strength.

For the analysis of the above system, the notion of subadditive closure is central.

DEFINITION 6.1 (Subadditive Closure). Let U either be in $\mathcal{F}$ or $\tilde{\mathcal{F}}$. Its *n-th self-convolution* is defined by

$$U^{(n)} := \underbrace{U \otimes \ldots \otimes U}_{n \text{ times}}.$$

Its *subadditive closure* $\bar{U}$ is defined by $\bar{U} := \bigwedge_{n=0}^{\infty} U^{(n)}$.

All operations in the above definition hold pointwise and use the univariate or bivariate convolution, respectively. For the empty operation $U^{(0)}$ of the convolution its neutral element must be defined. It is given by

$$\mathbf{1}(t) := \begin{cases} 0 & \text{if } t = 0 \\ \infty & \text{if } t > 0 \end{cases}, \qquad \mathbf{1}(s,t) := \begin{cases} 0 & \text{if } s = t \\ \infty & \text{if } s < t \end{cases}.$$

Indeed $\mathbf{1} \otimes U = U \otimes \mathbf{1} = U$.

DEFINITION 6.2. Let U either be in $\mathcal{F}$ or $\tilde{\mathcal{F}}$. It is *subadditive*, if

$$U(t) \leq U(t-s) + U(s) \qquad \text{for all } (s,t) \in \Lambda$$
$$U(s,t) \leq U(s,r) + U(r,t) \qquad \text{for all } (s,t),\ (s,r),\ (r,t) \in \Lambda$$

holds.

LEMMA 6.3. *Let U be subadditive, then $\bar{U} = \mathbf{1} \wedge U$.*

PROOF. If U is subadditive, then

$$U(s,t) \geq \min_{s \leq r \leq t} \{U(s,r) + U(r,t)\} \geq \min_{s \leq r \leq t} \{U(s,t)\}$$

for all $(s,t) \in \Lambda$ and hence $U^{(n)} = U$ for all $n \geq 1$. The univariate case follows in the same way. $\square$

As the name suggests, the subadditive closure of some U is subadditive again (see for example [**35, 102**]). A well-known property of the min-plus convolution is its distributivity with respect to minima. The following lemma is needed later and is easily checked.

LEMMA 6.4. *Let A, B, C be either in $\mathcal{F}$ or $\tilde{\mathcal{F}}$. Then $A \otimes (B \wedge C) = A \otimes B \wedge A \otimes C$.*

6.2. Univariate Results

This section derives and solves the univariate feedback equation. A short discussion on the role of subadditivity and the window size Σ follows.

Adding a constant Σ can be expressed by the min-plus convolution. To that end, define

$$\mathbf{1}_\Sigma(t) := \begin{cases} \Sigma & \text{if } t = 0 \\ \infty & \text{if } t > 0 \end{cases}.$$

Indeed, it holds

$$\mathbf{1}_\Sigma \otimes C(t) = \inf_{0 \le s \le t} \{\mathbf{1}_\Sigma(s) + C(t-s)\} = \Sigma + C(t)$$

for any flow C and $t \in \mathbb{K}$. This allows to describe the whole feedback loop via Theorem 1.7 and the service curve $U_{fb} = U \otimes U_* \otimes \mathbf{1}_\Sigma$. Here U_* is a service curve for the placeholder in Figure 6.1. The whole system is summarized in

$$B(t) = A(t) \wedge D(t) \ge A(t) \wedge B \otimes U_{fb}(t). \tag{6.1}$$

This is the *feedback equation* for the univariate case.

To analyze the performance of WFC systems a service description for the throttle must be found. This means that the expression $A(t) \wedge B \otimes U_{fb}(t)$ must be recast into some service curve $U_\wedge$ such that

$$B(t) \ge A \otimes U_\wedge.$$

Theorem 6.5 shows how to construct $U_\wedge$. For now assume that such a service curve $U_\wedge$ exists, then Theorem 1.7 gives the end-to-end service curve $U_{sys}(t) = U_\wedge \otimes U(t)$. With U_{sys} the system in Figure 6.1 can be analyzed; for example, an end-to-end delay bound derives from using U_{sys} in Theorem 1.18.

THEOREM 6.5. *Assume that the feedback loop in Figure 6.1 is described by a service curve U_{fb} with $U_{fb}(0) > 0$. Further, assume that $A(t) < \infty$ for all $t \in \mathbb{K}$. Then the throttle element $\wedge$ has service curve $\bar{U}_{fb}$.*

PROOF. The proof is based on Theorem 2.1.6 in [**35**]. It is to show that every flow fulfilling Equation (6.1), also fulfills

$$B(t) \ge A \otimes \bar{U}_{fb}.$$

To that end, let B fulfill Equation (6.1) and fix some $t \in \mathbb{K}$, then

$$B \otimes U_{fb}^{(n+1)}(t) \ge B \otimes U_{fb}^{(n+1)}(0) \ge (n+1) U_{fb}(0).$$

Since $A(t) < \infty$ and $U_{fb}(0) > 0$ there exists an integer $n^* \in \mathbb{N}$ such that $B \otimes U_{fb}^{(n^*+1)}(t) > A(t)$. Iterating the feedback equation leads to

$$\begin{aligned} B(t) &\ge A(t) \wedge B \otimes U_{fb}(t) \ge A(t) \wedge (A(t) \wedge B \otimes U_{fb}) \otimes U_{fb}(t) \\ &= A(t) \wedge A \otimes U_{fb}(t) \wedge B \otimes U_{fb}^{(2)}(t) \ge \ldots \ge \bigwedge_{k=0}^{n^*} A \otimes U_{fb}^{(k)}(t) \wedge B \otimes U_{fb}^{(n^*+1)}(t). \end{aligned}$$

Because $B \otimes U_{fb}^{(n^*+1)}(t) \geq A(t) \geq B(t)$ the last term can be omitted and it follows

$$B(t) \geq \bigwedge_{k=0}^{n^*} A \otimes U_{fb}^{(k)}(t) \wedge B \otimes U_{fb}^{(n^*+1)}(t) = A \otimes (\bigwedge_{k=0}^{n^*} U_{fb}^{(k)})(t) \geq A \otimes \bar{U}_{fb}(t).$$

□

The condition $U_{fb}(0) > 0$ explains the role of the window-element: Without the addition of Σ the resulting service curve would generally not fulfill the positivity condition in zero.

The role of Σ goes even further: If Σ is large enough, the feedback loop's service curve will become subadditive. In this case the calculation of $\bar{U}_{fb}$ simplifies. To that end, denote the service curve of the feedback loop without Σ by U_{res}. It is defined by

$$U_{fb} = U \otimes U_* \otimes \mathbf{1}_\Sigma =: U_{res} \otimes \mathbf{1}_\Sigma.$$

The self-convolution of U_{fb} reads then as

$$\begin{aligned} U_{fb}^{(2)}(t) &= (U_{res} \otimes \mathbf{1}_\Sigma)^{(2)}(t) = \inf_{0 \leq s \leq t}\{U_{res}(s) + \Sigma + U_{res}(t-s) + \Sigma\} \\ &= 2\Sigma + \inf_{0 \leq s \leq t}\{U_{res}(s) + U_{res}(t-s)\}; \end{aligned}$$

and if the infimum can be bounded by

$$\inf_{0 \leq s \leq t}\{U_{res}(s) + U_{res}(t-s)\} \geq U_{res}(t) - K \tag{6.2}$$

for some constant $0 < K \leq \Sigma$, the self-convolution continues to

$$U_{fb}^{(2)}(t) \geq 2\Sigma - K + U_{res}(t) \geq U_{res}(t) + \Sigma = U_{fb}(t).$$

Hence, if the self-convolution of U_{res} is bounded, the service curve U_{fb} becomes subadditive for Σ large enough. This insight is used later in Chapter 7.

Example 6.6. Let $U_{res}(t) = [r(t-T)]^+$ be a rate latency service curve. The self-convolution of U_{res} fulfills

$$\begin{aligned} U_{res}^{(2)}(t) &= \inf_{0 \leq s \leq t}\{[r(s-T)]^+ + [r(t-s-T)]^+\} \geq \inf_{0 \leq s \leq t}\{[rs - rT + rt - rs - rT]^+\} \\ &\geq \inf_{0 \leq s \leq t}\{[rt - rT]^+\} - rT = U_{res}(t) - rT. \end{aligned}$$

If $\Sigma \geq rT$, the feedback system's throttle simplifies to the service curve $\bar{U}_{fb} = U_{res} \otimes \mathbf{1}_\Sigma$. In this case the end-to-end service curve is

$$U_{sys}(t) = U \otimes (U_{res} \otimes \mathbf{1}_\Sigma)(t) \geq U_{res}(t).$$

6.3. Bivariate Results

The results for the bivariate formulation parallel those of the univariate case; still, their value exceeds a simple finger exercise or transcript of the univariate results. The definition of the bivariate window-element makes this clear.

Definition 6.7. A *window-element* is described by a function $\Sigma : \mathbb{K} \to \mathbb{R}_0^+$ such that for any input A it produces an output B by

$$B(t) = A(t) + \Sigma(t)$$

$$\Sigma(t) - \Sigma(s) \geq -A(s,t) \qquad \text{for all } s \leq t. \tag{6.3}$$

As for flows the bivariates of Σ are defined by $\Sigma(s,t) := \Sigma(t) - \Sigma(s)$.

REMARK 6.8. The constraint (6.3) enforces the window-element's output $B(s,t)$ to be nonnegative.

The above definition allows the window-element to vary in time. Still the addition of Σ is described as a bivariate convolution. Define

$$\mathbf{1}_\Sigma(s,t) := \begin{cases} \Sigma(t) & \text{if } s = t \\ \infty & \text{if } s \neq t \end{cases}.$$

Then it holds $\mathbf{1}_\Sigma \otimes C(0,t) = C \otimes \mathbf{1}_\Sigma(0,t) = C(t) + \Sigma(t)$ for any flow C.

The bivariate feedback equation has the form

$$B(t) = A(t) \wedge D(t) \geq A(t) \wedge B \otimes U_{fb}(0,t), \tag{6.4}$$

where $U_{fb} = U \otimes U_* \otimes \mathbf{1}_\Sigma$.

The next theorem is from the textbook of Chang [**35**].

THEOREM 6.9. *Assume that the feedback loop in Figure 6.1 is a dynamic U_{fb}-server with $\inf_{t\in\mathbb{K}}\{U_{fb}(t,t)\} > 0$ and $U_{fb} \in \tilde{\mathcal{F}}_+$. Further assume that $A(t) < \infty$ for all $t \in \mathbb{K}$. Then the throttle element $\wedge$ is a dynamic $\bar{U}_{fb}$-server.*

PROOF. The proof is similar to the one of Theorem 6.5. Fix some time $t \in \mathbb{K}$. It holds

$$\begin{aligned} B \otimes U_{fb}^{(n)}(0,t) &= \inf_{0\leq s_1 \leq \ldots \leq s_{n+1}=t}\{B(s_1) + \sum_{i=1}^{n} U_{fb}(s_i, s_{i+1})\} \\ &\geq \inf_{0\leq s_1 \leq \ldots \leq s_{n+1}=t}\{\sum_{i=1}^{n} U_{fb}(s_i, s_i)\} \geq n \inf_{s\geq 0}\{U_{fb}(s,s)\} \end{aligned}$$

for all $n \geq 1$. Hence, an n^* exists with $B \otimes U_{fb}^{(n^*+1)}(0,t) > A(t)$. The proof concludes as in Theorem 6.5. $\square$

As in the univariate case, large values of Σ can produce a subadditive service description of the feedback loop. Write again

$$U_{fb} = U \otimes U_* \otimes \mathbf{1}_\Sigma =: U_{res} \otimes \mathbf{1}_\Sigma.$$

Assume that $U_{res}^{(2)}$ is bounded by

$$\inf_{s\leq r\leq t}\{U_{res}(s,r) + U_{res}(r,t)\} \geq U_{res}(s,t) - K \tag{6.5}$$

for all $(s,t) \in \Lambda$ and some constant $0 < K \leq \inf_t\{\Sigma(t)\}$. Then the self-convolution of U_{fb} leads to

$$\begin{aligned} U_{fb}^{(2)}(s,t) = (U_{res} \otimes \mathbf{1}_\Sigma)^{(2)}(s,t) &= \inf_{s\leq r\leq t}\{U_{res}(s,r) + \Sigma(r) + U_{res}(r,t) + \Sigma(t)\} \\ &= \Sigma(t) + \inf_{s\leq r\leq t}\{U_{res}(s,r) + \Sigma(r) + U_{res}(r,t)\} \\ &\geq \Sigma(t) + U_{res}(s,t) - K + \inf_{s\leq r\leq t}\{\Sigma(r)\} \geq U_{fb}(s,t) \end{aligned}$$

and U_{fb} is subadditive. As before, the subadditive closure takes the closed form $\bar{U}_{fb} = \mathbf{1} \wedge U_{fb}$.

The next corollary appears trivial at first; however, it relaxes the condition $U_{fb} \in \tilde{\mathcal{F}}_+$ and hence applies to service descriptions that are the result of the leftover operation $\ominus$.

Corollary 6.10. *Assume that the feedback loop in Figure 6.1 is a dynamic U_{fb}-server with $U_{fb} = U_{res} \otimes \mathbf{1}_\Sigma$ for some subadditive U_{res}. If $\inf_{t\in\mathbb{K}}\{\Sigma(t)\} > 0$ and $A(t) < \infty$ for all $t \in \mathbb{K}$, it holds*

$$B(t) \geq A \otimes (\mathbf{1} \wedge U_{fb})(0,t)$$

for all $t \in \mathbb{K}$.

Proof. Extend B recursively for

$$\begin{aligned} B(t) &\geq A(t) \wedge B \otimes U_{fb}(0,t) \geq \dots \\ &\geq \bigwedge_{n=0}^{n^*} A \otimes U_{fb}^{(n)}(0,t) \wedge B \otimes U_{fb}^{(n^*+1)} \end{aligned}$$

The last convolution is expressed as

$$\begin{aligned} &\inf_{0\leq s_1 \leq \ldots \leq s_n \leq s_{n+1}=t} \{B(s_1) + \sum_{i=1}^{n^*+1} U_{res}(s_i, s_{i+1}) + \Sigma(s_i)\} \\ \geq &\inf_{0\leq s_1\leq t}\{B(s_1) + U_{res}(s_1,t)\} + n^* \inf_{0\leq r\leq t} \Sigma(r)\}. \end{aligned}$$

For n^* large enough the above convolution is greater than $A(t)$, from which follows

$$B(t) \geq \bigwedge_{n=0}^{n^*} A \otimes U_{fb}^{(n)}(0,t).$$

The same argument applies to the terms $A \otimes U_{fb}^{(n)}(0,t)$ too and leads to $A \otimes U_{fb}^{(n)}(0,t) \geq A \otimes U_{fb}(0,t)$ for all $n > 1$. Hence, it follows

$$B(t) \geq A \otimes (\mathbf{1} \wedge U_{fb})(0,t).$$

□

6.4. σ-additive Operators

When switching from univariate to bivariate calculus the min-plus convolution looses its commutativity (see also Chapter 1). This denies the step

$$(U \otimes U_* \otimes \mathbf{1}_\Sigma)^{(2)} = U^{(2)} \otimes U_*^{(2)} \otimes \mathbf{1}_\Sigma^{(2)}. \tag{6.6}$$

Subsequently, calculating the subadditive closure becomes much harder.

This section shows how one can still commute a dynamic U-server with a fixed-delay element. This allows to calculate the subadditive closure by self-convolving the single service descriptions, as one would like to do in (6.6). The notion of σ-additive operators **[35]** is helpful in this context.

For the entirety of this section let $\mathbb{K} = \mathbb{N}_0$.

Definition 6.11. An operator $\pi : \tilde{\mathcal{F}}_+ \to \tilde{\mathcal{F}}_+$ is called *σ-additive*, if

$$\pi\Big(\bigwedge_{n=1}^{\infty} F_n\Big) = \bigwedge_{n=1}^{\infty} \pi(F_n), \tag{6.7}$$

where F_n is any sequence in $\tilde{\mathcal{F}}_+$ and the infima are understood pointwisely. The space of σ-additive operators is denoted by Π.

LEMMA 6.12. *The space Π is closed under taking compositions and countable minima. Further σ-additive operators distribute over countable minima, i.e.,*

$$\Big(\bigwedge_{n=1}^{\infty}\pi_n\Big)\circ\pi=\bigwedge_{n=1}^{\infty}(\pi_n\circ\pi),\quad \pi\circ\Big(\bigwedge_{n=1}^{\infty}\pi_n\Big)=\bigwedge_{n=1}^{\infty}(\pi\circ\pi_n).$$

PROOF. Let $\pi_1,\pi_2\in\Pi$. Then it holds

$$\pi_1\Big(\pi_2\Big(\bigwedge_{k=1}^{\infty}F_k\Big)\Big)=\pi_1\Big(\bigwedge_{k=1}^{\infty}\pi_2(F_k)\Big)=\bigwedge_{k=1}^{\infty}\pi_1(\pi_2(F_k))=\bigwedge_{k=1}^{\infty}\pi(F_k)$$

for any sequence $F_k\in\tilde{\mathcal{F}}_+$.

Let $\pi_n\in\Pi$ be a sequence of σ-additive operators. Define $\pi:=\bigwedge_{n=1}^{\infty}\pi_n$. Then it holds

$$\pi\Big(\bigwedge_{k=1}^{\infty}F_k\Big)=\bigwedge_{n=1}^{\infty}\pi_n\Big(\bigwedge_{k=1}^{\infty}F_k\Big)=\bigwedge_{n=1}^{\infty}\bigwedge_{k=1}^{\infty}\pi_n(F_k)=\bigwedge_{k=1}^{\infty}\bigwedge_{n=1}^{\infty}\pi_n(F_k)=\bigwedge_{k=1}^{\infty}\pi(F_k)$$

for all sequences $F_k\in\tilde{\mathcal{F}}_+$.

Eventually for any $F\in\tilde{\mathcal{F}}_+$ it holds

$$\Big(\bigwedge_{n=1}^{\infty}\pi_n\Big)\circ\pi(F)=\Big(\bigwedge_{n=1}^{\infty}\pi_n\Big)(\pi(F))=\bigwedge_{n=1}^{\infty}\pi_n(\pi(F))=\bigwedge_{n=1}^{\infty}(\pi_n\circ\pi)(F),$$

$$\pi\circ\Big(\bigwedge_{n=1}^{\infty}\pi_n\Big)(F)=\pi\circ\Big(\bigwedge_{n=1}^{\infty}\pi_n(F)\Big)=\bigwedge_{n=1}^{\infty}\pi(\pi_n(F))=\bigwedge_{n=1}^{\infty}(\pi\circ\pi_n)(F)$$

because of $\pi(F),\pi_n(F)\in\tilde{\mathcal{F}}_+$. □

EXAMPLE 6.13. The bivariate convolution with some $U\in\tilde{\mathcal{F}}_+$ is an example for a σ-additive operator. Indeed for any sequence $F_n\in\tilde{\mathcal{F}}_+$ and choice $(s,t)\in\Lambda$ it follows

$$\begin{aligned}\min_{s\le r\le t}\{\bigwedge_{n=1}^{\infty}F_n(s,r)+U(r,t)\}&=\min_{s\le r\le t}\{\bigwedge_{n=1}^{\infty}(F_n(s,r)+U(r,t))\}\\&=\bigwedge_{n=1}^{\infty}\min_{s\le r\le t}\{F_n(s,r)+U(r,t)\}=\bigwedge_{n=1}^{\infty}F_n\otimes U(s,t)\end{aligned}$$

and hence

$$\pi_U(\bigwedge_{n=1}^{\infty}F_n):=(\bigwedge_{n=1}^{\infty}F_n)\otimes U=\bigwedge_{n=1}^{\infty}F_n\otimes U=\bigwedge_{n=1}^{\infty}\pi_U(F).$$

Other examples for σ-additive operators in the context of WFC systems are

$$\begin{aligned}\pi_{\tilde{e}}(A)&=A,\\ \pi_{+\Sigma}(A)(s,t)&=\begin{cases}A(s,t) & \text{if } s\neq 0\\ A(0,t)+\Sigma & \text{if } s=0\end{cases},\\ \pi_{\Delta}(A)(s,t)&=\begin{cases}A(s,t-\Delta) & \text{if } s\le t-\Delta\\ 0 & \text{else}\end{cases},\end{aligned}$$

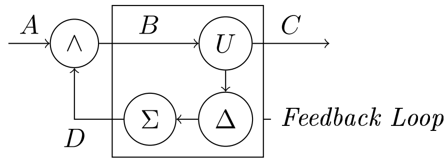


FIGURE 6.2. A window flow controlled system with a fixed-delay element of length Δ. The delay element is described by the operator π_Δ.

where $\Sigma \in \mathbb{R}$. The above operators represent the identity operator, a window-element, and a fixed-delay element, respectively. Their σ-additivity and the following properties are an immediate consequence of their definitions:

- $\pi_{+\Sigma}$ commutes with π_U, i.e., $\pi_U \circ \pi_{+\Sigma} = \pi_{+\Sigma} \circ \pi_U$.
- $\pi_{+\Sigma}$ commutes with π_Δ, i.e., $\pi_\Delta \circ \pi_{+\Sigma} = \pi_{+\Sigma} \circ \pi_\Delta$.
- $\pi_{+\Sigma}^2 = \pi_{+2\Sigma}$, and U is subadditive if and only if π_U is idempotent, i.e., $\pi_U^2 = \pi_U$.

With the above operators the feedback loop of Figure 6.1 is

$$B = A \wedge \pi_{fb}(B) := A \wedge \pi_{+\Sigma} \circ \pi_* \circ \pi_U(B), \tag{6.8}$$

where π_* describes the behavior of the placeholder $*$. The solution of this feedback equation resembles those of the univariate and bivariate formulations.

THEOREM 6.14. *Assume that the feedback loop in Figure 6.1 is described by the σ-additive operator π_{fb}, i.e., $D \geq \pi_{fb}(B)$. If $A(s,t) < \infty$ for all $(s,t) \in \Lambda$ and if there exists a $\delta > 0$ such that $\pi_{fb}^{(n)}(E)(t,t) \geq n\delta$ for all $n, t \in \mathbb{N}_0$ and $E \in \tilde{\mathcal{F}}$, then it holds*

$$B \geq \bigwedge_{k=0}^{\infty} \pi_{fb}^k(A) =: \bar{\pi}_{fb}(A),$$

where $\pi_{fb}^0 = \pi_{\tilde{e}}$.

PROOF. The proof works similar to the ones in Theorem 6.5 and 6.9. Let $(s,t) \in \Lambda$ be arbitrary. Then, since $\pi_{fb}^n(B) \in \tilde{\mathcal{F}}_+$, it holds

$$\pi_{fb}^n(B)(s,t) \geq \pi_{fb}^n(s,s) \geq n\delta$$

and hence an n^* exists such that $\pi_{fb}^{n^*}(B)(s,t) \geq A(s,t)$. The rest follows as for the univariate and bivariate formulations. $\square$

In the next lemma the placeholder $*$ is set to a fixed-delay element; further, the service U is assumed to be subadditive. As a result, the commutation step in (6.6) becomes possible, after introducing an error term.

LEMMA 6.15. *Consider the WFC system of Figure 6.2, where π_Δ is a fixed-delay element, as defined above. If $U \in \tilde{\mathcal{F}}_+$ is subadditive, decreasing in its first variable, and $U(t,t) = 0$ for all $t \in \mathbb{N}_0$, the system has the description*

$$B(t) \geq \pi_{sys}(A)(0,t) = \bigwedge_{n=0}^{n^*} n(\Sigma - U_\Delta) + \min_{n\Delta \leq s \leq t} \{A(0, s - n\Delta) + U(s,t)\}$$

for all $t \in \mathbb{N}_0$. Here n^* is the first $n \in \mathbb{N}$ such that $n\Delta > t$ and $U_\Delta := U(t - \Delta, t)$.

PROOF. The feedback system is described by

$$B(t) \geq A(t) \wedge \pi_{+\Sigma} \circ \pi_\Delta \circ \pi_U(B)(0,t) = \pi_{fb}(B)(0,t).$$

Choose some $n, t \in \mathbb{N}_0$ and $E \in \tilde{\mathcal{F}}_+$, then it is easy to see that $\pi_{fb}^{(n)}(E)(t,t) \geq n\Sigma$ and the conditions of Theorem 6.14 are fulfilled. Thus, it holds

$$B \geq \pi_\wedge(A) := \bigwedge_{n=0}^{\infty} (\pi_{+\Sigma} \circ \pi_\Delta \circ \pi_U)^n(A).$$

The question is: How does π_{fb}^n evolve for $n \to \infty$? By the associativity of σ-additive operators and the commutativity of $\pi_{+\Sigma}$ with the other operators, it holds

$$\pi_{fb} \circ \pi_{fb} = \pi_{+\Sigma} \circ \pi_\Delta \circ \pi_U \circ \pi_{+\Sigma} \circ \pi_\Delta \circ \pi_U = \pi_{+2\Sigma} \circ (\pi_\Delta \circ \pi_U) \circ (\pi_\Delta \circ \pi_U).$$

The goal is to compare $\pi_\Delta \circ \pi_U$ with $\pi_U \circ \pi_\Delta$. For $A \in \tilde{\mathcal{F}}_+$ it holds

$$\pi_\Delta \circ \pi_U(A)(s,t) + U(t-\Delta,t) = \min_{s \leq r \leq t-\Delta} \{A(s,r) + U(r,t-\Delta) + U(t-\Delta,t)\}.$$

Since U is subadditive, it holds $U(r,t-\Delta) + U(t-\Delta,t) \geq U(r,t)$, which results in

$$\begin{aligned}
\pi_\Delta \circ \pi_U(A)(s,t) &\geq \min_{s \leq r \leq t-\Delta} \{A(s,r) + U(r,t)\} - U(t-\Delta,t) \\
&= \min_{s+\Delta \leq r' \leq t} \{A(s,r'-\Delta) + U(r'-\Delta,t)\} - U(t-\Delta,t) \\
&\geq \min_{s+\Delta \leq r' \leq t} \{\pi_\Delta(A)(s,r') + U(r',t)\} - U(t-\Delta,t) \\
&= \min_{s \leq r' \leq t} \{\pi_\Delta(A)(s,r') + U(r',t)\} - U(t-\Delta,t) \\
&= \pi_U \circ \pi_\Delta(A)(s,t) - U(t-\Delta,t).
\end{aligned}$$

The minimum in line three does not decrease when the interval extends to $[s,t]$, due to the definition of the operator π_Δ and U being decreasing in the first variable.

The above relation between $\pi_\Delta \circ \pi_U$ and $\pi_U \circ \pi_\Delta$ introduces the term $U(t-\Delta,t)$, which is independent of s and A. Iterating the above leads to

$$\begin{aligned}
\pi_{fb}^{n+1}(A)(s,t) &= (\pi_{+\Sigma} \circ \pi_\Delta \circ \pi_U)^{n+1}(A)(s,t) \\
&= \pi_{+\Sigma} \circ \pi_\Delta \circ \pi_U \circ \pi_{fb}^n(A)(s,t) \\
&\geq \pi_{+\Sigma} \circ \pi_\Delta \circ \pi_U \circ \pi_{+n\Sigma} \circ \pi_U \circ \pi_{n\Delta}(A)(s,t) - nU(t-\Delta,t) \\
&= \pi_{+(n+1)\Sigma} \circ \pi_\Delta \circ \pi_U \circ \pi_{n\Delta}(A)(s,t) - nU(t-\Delta,t) \\
&\geq \pi_{+(n+1)\Sigma} \circ \pi_U \circ \pi_{(n+1)\Delta}(A)(s,t) - (n+1)U(t-\Delta,t).
\end{aligned}$$

Next it is to show that it is sufficient to take the minimum over a finite set of integers in the closure $\bar{\pi}_{fb}$. To that end, let n^* be the first n such that $n\Delta > t$ and denote for brevity $U(t-\Delta,t) = U_\Delta$. Then

$$\begin{aligned}
\pi_{fb}^{n^*}(A)(0,t) &\geq \pi_{+n^*(\Sigma - U_\Delta)} \circ \pi_U \circ \pi_{n^*\Delta}(A)(0,t) \\
&= n^*(\Sigma - U_\Delta) + \min_{0 \leq s \leq t} \{\pi_{n^*\Delta}(A)(0,s) + U(s,t)\} = n^*(\Sigma - U_\Delta),
\end{aligned}$$

as the last minimum always reduces to zero by setting $s = t$. In particular it holds

$$\begin{aligned}\pi_{fb}^{n^*+1}(A)(0,t) &\geq (n^*+1)\Sigma - n^* U_\Delta + \pi_\Delta \circ \pi_U \circ \pi_{n^*\Delta}(A)(0,t)\\ &= (n^*+1)\Sigma - n^* U_\Delta + \min_{0\leq s\leq t-\Delta}\{\pi_{n^*\Delta}(A)(0,s) + U(s,t-\Delta)\}\\ &= (n^*+1)\Sigma - n^* U_\Delta \geq n^*(\Sigma - U_\Delta).\end{aligned}$$

Hence,

$$\pi_{fb}^n(A)(0,t) \geq \left\lceil \frac{t}{\Delta} \right\rceil (\Sigma - U(t-\Delta,t)) \qquad \text{for all } n > \frac{t}{\Delta}$$

and consequently

$$\pi_\wedge(A)(0,t) = \bigwedge_{n=0}^{\infty} \pi_{fb}^n(A)(0,t) \geq \bigwedge_{n=0}^{n^*} \pi_{fb}^{n^*}(A)(0,t).$$

With a finite representation of $\pi_\wedge$ the end-to-end behavior is captured by π_{sys} with

$$\begin{aligned}\pi_{sys}(A)(0,t) = \pi_U \circ \pi_\wedge(A)(0,t) &= \Big(\bigwedge_{n=0}^{\infty} \pi_U \circ \pi_{fb}^n\Big)(A)(0,t)\\ &\geq \bigwedge_{n=0}^{n^*} \pi_U \circ \pi_{+n(\Sigma+U_\Delta)} \circ \pi_U \circ \pi_{n\Delta}(A)(0,t)\\ &= \bigwedge_{n=0}^{n^*} \pi_{+n(\Sigma+U_\Delta)} \circ \pi_U \circ \pi_{n\Delta}(A)(0,t)\\ &= \bigwedge_{n=0}^{n^*} n(\Sigma - U_\Delta) + \min_{0\leq s\leq t}\{\pi_{n\Delta}(A)(0,s) + U(s,t)\}\\ &= \bigwedge_{n=0}^{n^*} n(\Sigma - U_\Delta) + \min_{n\Delta\leq s\leq t}\{A(0,s-n\Delta) + U(s,t)\}.\end{aligned}$$

□

The above lemma provides a closed form of $\bar{\pi}_{fb}$ for any fixed $t \in \mathbb{N}_0$. As seen in the next chapter, this allows to extend the deterministic result to a stochastic setting, in which any non-closed forms of $\bigwedge_{n=0}^{\infty} \pi_{fb}^n$ would cause severe problems.

CHAPTER 7

Window Flow Control in Stochastic Network Calculus

Parts of this chapter are joint work with J. Schmitt and have been presented first in [**15**].

Window flow control in SNC has been considered a challenging and open problem for years [**88, 66, 68**]. Recently, first successes can be marked [**41, 16, 139**]. This chapter presents the challenge of WFC in SNC in more detail and presents first steps towards a solution. To that end, Section 7.1 investigates why standard methods only give limited results for WFC systems. Afterwards two methods for obtaining stochastic performance bounds in WFC systems are presented. The first method is based on preserving subadditivity inside the feedback loop. The second method utilizes the problem's stochastic nature: It bounds the *probability* of the feedback loop not being subadditive. This bound is derived by SNC-methods again. As a result, both methods consider a subadditive description of the feedback loop, which simplifies the construction of the subadditive closure $\bar{U}_{fb}$. A numerical evaluation that compares the obtained results with the deterministic methods and to each other concludes this chapter.

7.1. Problem Exposition

To create stochastic performance bounds for WFC systems the throttle's service must be stochastically bounded. Here the subadditive closure $\bar{U}_{fb}(t)$, $\bar{U}_{fb}(s,t)$, or $\bar{\pi}_{fb}$ forms a severe obstacle. This section describes, why the straightforward application of already known methods only leads to limited results. For ease of presentation $\mathbb{K} = \mathbb{N}_0$.

Assume that the feedback loop is tailbounded by a service curve U_{fb} with error ζ and fix some $t \in \mathbb{K}$ and $\varepsilon > 0$. To successfully apply Theorem 6.5 the assumption

$$D(s) \geq (B \otimes U_{fb}(\varepsilon))(s) \qquad \text{for all } s \leq t \tag{7.1}$$

is needed, as it leads to Equation (6.1). In this case the throttle has service curve $\bar{U}_{fb}$ and hence

$$\begin{aligned}\mathbb{P}(B(t) < (A \otimes \bar{U}_{fb}(\varepsilon))(t)) &\leq \mathbb{P}\big(\bigcup_{s\leq t} D(s) < (B \otimes U_{fb}(\varepsilon))(s)\big)\\ &\leq \sum_{s=0}^{t} \mathbb{P}(D(s) < (B \otimes U_{fb}(\varepsilon))(s)) \leq \sum_{s=0}^{t} \zeta(s,\varepsilon).\end{aligned}$$

The above forms a tailbounded service curve $\bar{U}_{fb}$ with error $\bar{\zeta}(t,\varepsilon) = \sum_{s=0}^{t} \zeta(s,\varepsilon)$ for the throttle. As the error $\bar{\zeta}$ is not convergent in t, however, this bound is of limited use.

The reason for the divergence of $\bar{\zeta}$ lies in assumption (7.1): The service curve property must hold for the entire interval $[0, t]$. This condition resembles the one encountered in the convolution of two tailbounded service curves (see Theorem 2.20). There the error function ζ is improved by introducing a slack rate δ to the service curve. Hence, the apparent first idea is to reduce the tailbounded service curve U_{fb} in the same way.

To that end, the tailbounded service curve definition of Chapter 3 is repeated here: Assume that for the feedback loop the tailbound

$$\mathbb{P}(B(t) < A \otimes [\mathcal{U} - \varepsilon]^{+}(t)) < \zeta(t, \varepsilon),$$

holds, where the convolution reads

$$A \otimes [\mathcal{U} - \varepsilon]^{+}(t) = \min_{0 \le s \le t} \{A(s) + [\mathcal{U}(t-s) - \varepsilon]^{+}\}.$$

Now fix some $t \in \mathbb{N}_0$. Similar to Theorem 2.20 relax Assumption (7.1) to

$$D(s) \ge (B \otimes [\mathcal{U} - \varepsilon - \delta(t-s))]^{+})(s) \qquad \text{for all } s \le t.$$

If Theorem 6.5 did apply, this would leads to

$$\begin{aligned}
\mathbb{P}(B(t) < (A \otimes \bar{U}_{fb}(\varepsilon))(t)) &\le \mathbb{P}\Big(\bigcup_{s \le t} D(s) < (B \otimes [\mathcal{U} - \varepsilon - \delta(t-s)]^{+})(s)\Big) \\
&\le \sum_{s=0}^{t} \zeta(s, \varepsilon + \delta(t-s)) < \infty. \qquad (7.2)
\end{aligned}$$

However, the condition $[\mathcal{U}(0) - \varepsilon - \delta t]^{+} > 0$ of Theorem 6.5 is only fulfilled for a limited choice of $t \in [0, t^{*})$, where t^{*} solves $[\mathcal{U}(0) - \varepsilon - \delta t]^{+} = 0$. Reducing δ leads to larger t^{*}; hence, to achieve Inequality (7.2) for all $t \in \mathbb{N}_0$ the parameter δ must tend to zero. This in turn leads to a divergence of the error $\bar{\zeta}$ again. Overall a tailbounded service curve with error independent of t cannot be reached by this approach.

Instead of a tailbounded service, assume now that the feedback loop is a (σ_{fb}, ρ_{fb})-bounded dynamic U_{fb}-server. Further, assume that the service element inside the feedback is a (σ_U, ρ_U)-bounded dynamic U-server and $A \preceq (\sigma_A, \rho_A)$. By Theorem 6.9 the throttle is a dynamic $\bar{U}_{fb}$-server, if $\inf_t U_{fb}(t,t) > 0$.

The goal is to find an MGF-bound for the expression $\phi_{A \oslash U_{sys}(s,t)}(\theta)$, as it is central in deriving performance bounds (see Theorem 2.39). In Theorem 3.1 the arrivals A distribute over the minima of U_{sys}, resulting in

$$\begin{aligned}
A \oslash U_{sys}(s,t) &= \sup_{0 \le r \le s} \Big\{ A(r,t) - \bigwedge_{n=0}^{\infty} U \otimes U_{fb}^{(n)}(r,s) \Big\} \\
&= \sup_{0 \le r \le s} \Big\{ \bigvee_{n=0}^{\infty} A(r,t) - U \otimes U_{fb}^{(n)}(r,s) \Big\} = \bigvee_{n=0}^{\infty} A \oslash (U \otimes U_{fb}^{(n)})(s,t).
\end{aligned}$$

This translates into the MGF-bounds

$$\phi_{A \oslash U_{sys}(s,t)}(\theta) \le \sum_{n=0}^{\infty} \phi_{A \oslash (U \otimes U_{fb}^{(n)})(s,t)}(\theta).$$

Using a generalized version of Hölder's inequality and the steps in Theorem 3.1 the MGF corresponding to index n is bounded by

$$
\begin{aligned}
&\phi_{A\oslash(U\otimes U_{fb}^{(n)})(s,t)}(\theta)\\
&\leq \sum_{r_1=0}^{s}\phi_{A(r_1,t)}(\theta)\sum_{r_1\leq r_2\leq\ldots r_{n+1}\leq s}\mathbb{E}(e^{-\theta(U(r_1,r_2)+U_{fb}(r_2,r_3)+\ldots+U_{fb}(r_{n+1},s))})\\
&\leq \sum_{r_1=0}^{s}\phi_{A(r_1,t)}(\theta)\sum_{r_1\leq\ldots\leq r_{n+1}\leq s=r_{n+2}}\phi_{U(r_1,r_2)}^{1/p_1}(-\theta p_1)\prod_{k=2}^{n+1}\phi_{U_{fb}(r_k,r_{k+1})}^{1/p_k}(-\theta p_k)
\end{aligned}
\tag{7.3}
$$

$$
\begin{aligned}
&\leq e^{\theta\sigma_A(\theta)+\theta\sigma_U(p_1\theta)+\sum_{k=2}^{n+1}\sigma_{U_{fb}}(p_k\theta)}\sum_{r_1=0}^{s}e^{\theta\rho_A(\theta)(t-s)}\\
&\quad\cdot\sum_{r_1\leq\ldots\leq r_{n+1}\leq s=r_{n+2}}e^{\theta(\rho_U(p_1\theta)+\rho_A(\theta))(r_2-r_1)}\prod_{k=2}^{n+1}e^{\theta(\rho_{U_{fb}}(p_k\theta)+\rho_A(\theta))(r_{k+1}-r_k)}\\
&=:\Phi(\theta,s,t),
\end{aligned}
$$

where $\sum_{k=1}^{n+1}\frac{1}{p_k}=1$. The next example shows that the above MGF diverges, in general, for $n,s,t\to\infty$.

Example 7.1. Assume that the arrivals and the crossflow have independent exponential increments and that the server U has a constant rate. Assume further that the feedback loop contains, besides U, only the window-element Σ. Thus,

$$
\begin{aligned}
\rho_A(\theta)&=\frac{1}{\theta}\log\Big(\frac{\lambda}{\lambda-\theta}\Big), & \sigma_A(\theta)&=0,\\
\rho_U(\theta)=\rho_{U_{fb}}(\theta)&=-c+\frac{1}{\theta}\log\Big(\frac{\lambda}{\lambda-\theta}\Big), & \sigma_U(\theta)&=0, \qquad \sigma_{U_{fb}}(\theta)=-\Sigma
\end{aligned}
$$

for all $\theta<\lambda$. Denote the prefactor in Line (7.3) by $e^{\theta\sigma(\theta)}$.

Let k_n be a sequence of indices for which $p_{k_n}\xrightarrow{n\to\infty}\infty$. Such a sequence exists, as for each $n\in\mathbb{N}$ it holds $1=\sum_{k=1}^{n}\frac{1}{p_k}\geq n\min\{\frac{1}{p_k}\}$. From this the existence of some $k\in\{1,\ldots,n\}$ with $p_k\geq n$ follows. As the following steps do not depend on the usage of $\rho_{U_{fb}}$ or ρ_U, the index k_n is without loss of generality greater than 1.

The sum in Line (7.3) contains as one possible choice of indices

$$r_2=r_3=\ldots=r_{k_n}=r_1;\quad r_{k_n+1}=\ldots r_{n+1}=s.$$

Considering only this choice results in

$$
\begin{aligned}
\Phi(\theta,s,t)&\geq e^{\theta\sigma(\theta)}\sum_{r_1=0}^{s}e^{\theta\rho_A(\theta)(t-s)}e^{\theta(\rho_{U_{fb}}(p_{k_n}\theta)+\rho_A(\theta))(s-r_1)}\\
&=e^{\theta\sigma(\theta)}\sum_{r_1'=0}^{s}e^{\theta\rho_A(\theta)(t-s)}e^{\theta(\rho_{U_{fb}}(p_{k_n}\theta)+\rho_A(\theta))r_1'}\\
&=e^{\theta\sigma(\theta)+\rho_A(\theta)(t-s)}\frac{1-e^{\theta(\rho_{U_{fb}}(p_{k_n}\theta)+\rho_A(\theta))(s+1)}}{1-e^{\theta(\rho_{U_{fb}}(p_{k_n}\theta)+\rho_A(\theta))}}\\
&\xrightarrow[(t-s)=\text{const.}]{t,s\to\infty}\quad e^{\theta\sigma(\theta)+\rho_A(\theta)(t-s)}\frac{1}{1-e^{\theta(\rho_{U_{fb}}(p_{k_n}\theta)+\rho_A(\theta))}}.
\end{aligned}
$$

Consider now the exponent $\theta(\rho_{U_{fb}}(p_{k_n}\theta) + \rho_A(\theta))$. It holds

$$\theta(\rho_{U_{fb}}(p_{k_n}\theta) + \rho_A(\theta)) = -c\theta + \frac{1}{p_{k_n}} \log\left(\frac{\lambda}{\lambda - p_{k_n}\theta}\right) + \log\left(\frac{\lambda}{\lambda-\theta}\right) \xrightarrow{n\to\infty} 0$$

as $p_{k_n} \xrightarrow{n\to\infty} \infty$ and therefore $\theta \xrightarrow{n\to\infty} 0$ (it must always hold $p_k\theta < \lambda$). Hence, the sum over all r_1 diverges to infinity, for this particular choice of indices. As all summands in Line (7.3) are positive, there cannot be other summands to compensate this divergence and Φ must be divergent too.

For tailbounds and MGF-bounds the difficulty in bounding the throttle's service stems from the infinite convolution in U_{sys}. The next two sections show a way towards a closed description for U_{sys}. By this the infinite convolution is avoided.

7.2. Subadditive Feedback Loops

This section focuses on subadditive service descriptions. An example for a subadditive server is a constant rate server $U_o(s,t) = r_U(t-s)$ with some positive rate r_U serving a crossflow A_U. As it turns out, subadditivity is the central property for an analysis of WFC systems.

LEMMA 7.2. *Fix some $\theta > 0$ and let $\mathbb{K} = \mathbb{N}_0$. Let the feedback loop of Figure 6.1 be a dynamic U_{fb}-server with $U_{fb} = U_{res} \otimes \mathbf{1}_\Sigma$. Further, assume that the MGF-bounds $U \preceq (\sigma,\rho)$ and $U_{res} \preceq (\sigma_{res}, \rho_{res})$ hold. If U_{res} is subadditive, the whole system is a dynamic U_{sys}-server for the input A and departures C with*

$$\phi_{U_{sys}(s,t)}(-\theta) \le e^{\theta\sigma(p\theta)+\theta\sigma_{res}(q\theta)} \sum_{r=s}^{t} e^{\theta\rho(p\theta)(r-s)+\theta\rho_{res}(q\theta)(t-r)},$$

where $\frac{1}{p} + \frac{1}{q} = 1$.

PROOF. Corollary 6.10 applies and characterizes the throttle as a dynamic $U_\wedge$-server with $U_\wedge = \mathbf{1} \wedge U_{fb}$. Concatenating $U_\wedge$ with U leads to the service description

$$\begin{aligned} U_{sys}(s,t) &= U_\wedge \otimes U(s,t) \ge U(s,t) \wedge U_{fb} \otimes U(s,t) \\ &\ge U(s,t) \wedge U_{res} \otimes U(s,t) + \inf_{s \le r \le t} \Sigma(r) \ge U_{res} \otimes U(s,t). \end{aligned}$$

Following the same steps as in Theorem 2.36 leads to the assertion. □

The above Lemma shows that the subadditive case avoids the infinite convolution. The fact that subadditivity is not necessary for a closed form of the subadditive closure is shown by Lemma 6.15. There the feedback loop consists of a subadditive dynamic U-server and a fixed delay element that are in concatenation not subadditive anymore. The feedback loop with a fixed-delay element is picked up again in the next section. The goal for the rest of this section is to investigate the class of subadditive service elements or, to be more precise, under which conditions Lemma 7.2 applies.

As before write $U_{fb} = U_{res} \otimes \mathbf{1}_\Sigma$. Now consider U_{res} as the service description of any kind of network with input B and departures C (see Figure 6.1). The definition of subadditivity extends intuitively to subadditivity of networks.

DEFINITION 7.3. A network with input flow B and output C is *subadditive*, if there exists a subadditive dynamic U_{e2e}-server such that

$$C(t) \ge B \otimes U_{e2e}(0,t) \qquad \text{for all } t \ge 0.$$

To answer the question which kind of networks are subadditive the network operations $\ominus$ and $\otimes$ are checked.

THEOREM 7.4. *Consider a subadditive dynamic U-server and a flow A. The leftover service $U_l = U - A$ is subadditive.*

PROOF. Fix some $s, r, t \in \mathbb{N}_0$ with $s \leq r \leq t$, then

$$\begin{aligned} U_l(s,r) + U_l(r,t) &= U(s,r) + U(r,t) - A(s,r) - A(r,t) \geq U(s,t) - A(s,t) \\ &= U_l(s,t). \end{aligned}$$

□

The concatenation of subadditive service elements is in general not subadditive.

EXAMPLE 7.5. Fix some $s < r_B < r_A < t$ with all times in $\mathbb{N}_0$, and define the bivariates

$$\begin{aligned} A(s,r) &:= \begin{cases} 0 & \text{if } r < r_A \\ A & \text{if } r \geq r_A \end{cases}, \qquad & B(r,t) &:= \begin{cases} 0 & \text{if } r \geq r_B \\ B & \text{if } r < r_B \end{cases}, \\ U(s,t) &:= u(t-s) - A(s,t), \qquad & V(s,t) &:= v(t-s) - B(s,t) \end{aligned}$$

for some constants $u, v, A, B \in \mathbb{R}^+$. The functions U and V correspond to the service description of constant rate servers that serve the high priority crossflows A and B, respectively.

For the convolution of U and V it holds

$$\begin{aligned} U \otimes V(s,t) &= \inf_{s \leq r \leq t} \{u(r-s) - A(s,r) + v(t-r) - B(r,t)\} \\ &\geq u(t-s) + \inf_{s \leq r \leq t} \{-A(s,r) - B(r,t)\} = (u \wedge v)(t-s) - (A \vee B). \end{aligned}$$

Whereas, for some $r_B \leq r \leq r_A$ it holds

$$\begin{aligned} U \otimes V(s,r) + U \otimes V(r,t) &= \inf_{s \leq r_1 \leq r} \{u(r_1 - s) + v(r_1 - r) - B\} \\ &\quad + \inf_{r \leq r_2 \leq t} \{u(r - r_2) - A + v(t - r_2)\} \\ &= (u \wedge v)(r-s) - B + (u \wedge v)(t-r) - A \\ &= (u \wedge v)(t-s) - (A+B) < (u \wedge v)(t-s) - (A \vee B) \\ &\leq U \otimes V(s,t). \end{aligned}$$

Hence $U \otimes V$ is not subadditive.

The above example shows that a straightforward concatenation of service elements does not preserve subadditivity. Still, there exist subadditive descriptions for concatenated service elements. For that purpose a stronger property than subadditivity is needed.

DEFINITION 7.6. A dynamic U-server is called *separable*, if there exists a subadditive U_o and a flow A_U such that

$$U(s,t) = U_o(s,t) - A_U(s,t).$$

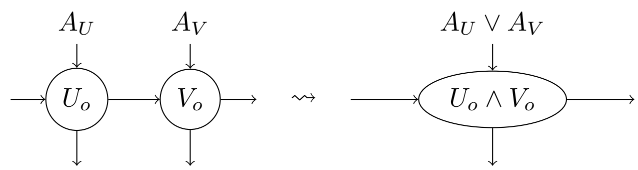


FIGURE 7.1. Theorem 7.8 allows a subadditive end-to-end description of separable service elements.

If $\mathbb{K} = \mathbb{R}_0^+$, the bivariates U_o and A_U must further have the representation

$$U_o(s,t) = \int_s^t u_{o,c}(r)\mathrm{d}r, \tag{7.4}$$

$$A_U(s,t) = \int_s^t a_{U,c}(r)\mathrm{d}r + \sum_{r\in[s,t]} \lim_{r'\searrow r} A_U(r') - A(r) \tag{7.5}$$

for some integrable functions $u_{o,c}$ and $a_{U,c}$. The summation is over all r for which the instantaneous increments $\lim_{r'\to r} A_U(r') - A_U(r)$ are nonzero. For a separable U this set must be countable. The instantaneous bursts are henceforth denoted by $a_{U,j}(r)$.

REMARK 7.7. A separable dynamic U-server is always subadditive by Theorem 7.4, but the converse must not be the case.

THEOREM 7.8. *Assume that U and V are separable. They form a separable dynamic W-server, as in Figure 7.1, with*

$$W_o(s,t) := \sum_{r=s}^{t-1} U_o(r,r+1) \wedge V_o(r,r+1),$$
$$A_W(s,t) := \sum_{r=s}^{t-1} A_U(r,r+1) \vee A_V(r,r+1),$$

if $\mathbb{K} = \mathbb{N}_0$ *and*

$$W_o(s,t) := \int_s^t u_{o,c}(r) \wedge v_{o,c}(r)\mathrm{d}r,$$
$$A_W(s,t) := \int_s^t a_{U,c}(r) \vee a_{V,c}(r)\mathrm{d}r + \sum_{r\in[s,t]} a_{U,j}(r) + a_{V,j}(r),$$

if $\mathbb{K} = \mathbb{R}_0^+$.

REMARK 7.9. Here $r \in [s,t]$ denotes all times for which either $a_{U,j}(r) \neq 0$ or $a_{V,j}(r) \neq 0$.

PROOF. Without loss of generality let $\mathbb{K} = \mathbb{R}_0^+$. The proof for slotted time follows as a special case.

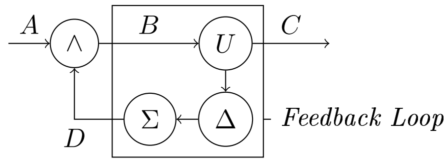


FIGURE 7.2. A window flow controlled system with a fixed-delay element of length Δ.

The tandem is a $U \otimes V$-server with

$$\begin{aligned} U \otimes V(s,t) &\geq \inf_{s \leq r \leq t} \{U_o(s,r) + V_o(r,t)\} - \sup_{s \leq r \leq t} \{A_U(s,r) + A_V(r,t)\} \\ &\geq \inf_{s \leq r \leq t} \Big\{ \int_s^r u_{o,c}(x) \wedge v_{o,c}(x) \mathrm{d}x + \int_r^t u_{o,c}(x) \wedge v_{o,c}(x) \mathrm{d}x \Big\} \\ &\quad - \sup_{s \leq r \leq t} \Big\{ \int_s^t a_{U,c}(x) \vee a_{V,c}(x) \mathrm{d}x + \sum_{r' \in [s,r]} a_{U,j}(r') + \sum_{r' \in [r,t]} a_{V,j}(r') \Big\} \\ &\geq W_o(s,t) - A_W(s,t), \end{aligned}$$

for all $s \leq t$. By the convolution's monotonicity it holds

$$C(t) \geq B \otimes (U \otimes V)(0,t) \geq B \otimes W(0,t)$$

and W is a separable dynamic server for the tandem. $\square$

The above proof can be generalized to tandems with more than two servers and can be applied to several crossflows. Extending each crossflow to the entire path of A_i, however, is surely not optimal and can in fact become arbitrarily loose. Similar to the construction of end-to-end service curves in deterministic network calculus (see Examples 1.15-1.17), better ways for attaining a networks service are available (see for example the numerical evaluation in Section 7.5). Such improvements rely on the considered network's structure and an efficient ordering of the demultiplexing and convolution operations. In contrast to the methods discussed in Chapter 1, here the additional condition of preserving subadditivity is crucial. More sophisticated exploits of topological characteristics are left to future work.

7.3. Fixed-Delay Elements

Before delving into the general case, a stochastic extension to Lemma 6.15 is given. This example considers a subadditive service U and a fixed-delay element in the feedback loop (see Figure 7.2). Such feedback loops are not subadditive due to the fixed-delay element. This scenario is of particular interest as the resulting stochastic bound converges in t.

THEOREM 7.10. *Let $\mathbb{K} = \mathbb{N}_0$. Consider the WFC system of Figure 7.2, where Δ is a fixed-delay element described by $\pi_\Delta \in \Pi$. Here $U \in \tilde{\mathcal{F}}_+$ is subadditive, decreasing in its first variable, and $U(t,t) = 0$ for all $t \in \mathbb{N}_0$. Let $\frac{1}{p} + \frac{1}{q} = 1$ and*

$(\sigma,\rho) \preceq U \preceq (\bar\sigma,\bar\rho)$ for some $\theta>0$. Under the stability conditions

$$(\rho_A(\theta)+\bar\rho_U(q\theta))\Delta \le \Sigma,$$
$$\rho_A(\theta)+\rho_U(p\theta) < 0,$$
$$\bar\rho_U(\theta)\Delta < \Sigma$$

the system has the end-to-end delay bound

$$\mathbb{P}(\mathfrak{d}(t)>T) \le \frac{e^{-\theta\rho_A(\theta)T+\theta\sigma(\theta,p,q)}}{(1-e^{\theta(\rho_A(\theta)\Delta+\bar\rho_U(q\theta)\Delta-\Sigma)})(1-e^{\theta(\rho_A(\theta)+\rho_U(p\theta))})}+\varepsilon(t)$$

with $\varepsilon(t) \xrightarrow{t\to\infty} 0$.

PROOF. Fix some $t,T\in\mathbb{N}_0$, $\theta>0$ and define $n^*=\lceil\frac{t+T}{\Delta}\rceil$. If $\mathfrak{d}(t)>T$, then it holds by the definition of the virtual delay

$$0 < A(t)-B(t+T) \le A(t)-\pi_{sys}(A)(0,t+T),$$

which yields

$$0<\bigvee_{n=0}^{n^*}\max_{n\Delta\le s\le t+T}\{A(t)-A(s-n\Delta)-U(s,t+T)\}-n(\Sigma-U_\Delta)$$

with $U_\Delta=U(t-\Delta,t)$ by Lemma 6.15. Recasting this in Chernoff's inequality and MGF-bounds results in

$$\begin{aligned}
&\mathbb{P}(\mathfrak{d}(t)>T)\\
&\le \sum_{n=0}^{n^*}\mathbb{E}(e^{\theta(\max_{n\Delta\le s\le t+T}\{A(t)-A(s-n\Delta)-U(s,t+T)\}-n(\Sigma-U_\Delta))})
\end{aligned}$$

(7.6)

$$\begin{aligned}
&\le \sum_{n=0}^{n^*-1}\Big(\sum_{s=n\Delta}^{t+T}\mathbb{E}(e^{\theta A(s-n\Delta,t)-\theta U(s,t+T)+\theta nU_\Delta})e^{-\theta n\Sigma}\Big)+\mathbb{E}(e^{\theta n^*U_\Delta})e^{-\theta n^*\Sigma}\\
&\le e^{\theta(\sigma_A(\theta)+\sigma_U(p\theta)+\bar\sigma_U(q\theta))}\sum_{n=0}^{n^*-1}\sum_{s=n\Delta}^{t+T}e^{\theta\rho_A(\theta)(t-s+n\Delta)}e^{\rho_U(p\theta)(t+T-s)}e^{\theta n\bar\rho_U(q\theta)\Delta}e^{-\theta n\Sigma}\\
&\quad+e^{\theta n^*(\bar\rho_U(\theta)\Delta-\Sigma)}e^{\theta\bar\sigma_U(\theta)}\\
&=: e^{-\theta\rho_A(\theta)T+\theta\sigma(\theta,p,q)}\sum_{n=0}^{n^*-1}e^{\theta n(\rho_A(\theta)\Delta+\bar\rho_U(q\theta)\Delta)-\theta n\Sigma}\sum_{s=n\Delta}^{t+T}e^{\theta(\rho_A(\theta)+\rho_U(p\theta))(t+T-s)}\\
&\quad+\varepsilon(t).
\end{aligned}$$

Under the assumed stability conditions the above bound converges in t since

$$\begin{aligned}
&\mathbb{P}(\mathfrak{d}(t)>T)\\
&\le e^{-\theta\rho_A(\theta)T+\theta\sigma(\theta,p,q)}\sum_{n=0}^{n^*-1}\Big(e^{\theta n(\rho_A(\theta)\Delta+\bar\rho_U(q\theta)\Delta-\Sigma)}(1-e^{\theta(\rho_A(\theta)+\rho_U(p\theta))})^{-1}\Big)+\varepsilon(t)\\
&\le \frac{e^{-\theta\rho_A(\theta)T+\theta\sigma(\theta,p,q)}}{(1-e^{\theta(\rho_A(\theta)\Delta+\bar\rho_U(q\theta)\Delta-\Sigma)})(1-e^{\theta(\rho_A(\theta)+\rho_U(p\theta))})}+\varepsilon(t).
\end{aligned}$$

□

REMARK 7.11. This bound is similar to the one in the unthrottled case (using the first part of Lemma 2.34)

$$\mathbb{P}(\mathfrak{d}(t) > T) \leq e^{-\theta\rho_A(\theta)T+\theta(\sigma_A(\theta)+\sigma_U(\theta))}(1 - e^{\theta(\rho_A(\theta)+\rho_U(\theta))})^{-1}.$$

Under stricter conditions on U the above bound can be improved.

COROLLARY 7.12. *Assume the setting from Theorem 7.10. Further assume that U has independent increments and is additive; i.e., for non-overlapping intervals $(s_1, t_1]$ and $(s_2, t_2]$ the random variables $U(s_1, t_1)$ and $U(s_2, t_2)$ are stochastically independent and it holds*

$$U(s, t) = U(s, r) + U(r, t)$$

for all $s \leq r \leq t \in \mathbb{N}_0$. Then

$$\begin{aligned}&\mathbb{P}(\mathfrak{d}(t) > T)\\ &\leq \frac{e^{\theta\rho_U(\theta)T}}{1 - e^{\theta(\rho_A(\theta)\Delta+\bar{\rho}_U(\theta)\Delta-\Sigma)}}\left(\frac{e^{\theta\rho_A(\theta)\Delta-\theta\bar{\rho}_U(\theta)\Delta}}{1 - e^{\theta(\rho_A(\theta)+\rho_U(\theta))}} + (T+\Delta)e^{\theta\rho_A(\theta)(\Delta-1)}\right) + \varepsilon(t)\end{aligned}$$

holds with $\varepsilon(t) \xrightarrow{t\to\infty} 0$.

PROOF. Choose some $n < n^* - 1$, where n^* is defined again as the first integer such that $n^*\Delta > t$. The quantity $nU_\Delta - U(s, t+T)$ in Equation (7.6) simplifies for all $s \leq t - \Delta$ to

$$(n-1)U_\Delta - U(s, t-\Delta) - U(t, t+T)$$

due to the additivity of U and U_Δ being defined as $U(t-\Delta, t)$. For all $s > t - \Delta$ it holds instead

$$nU_\Delta - U(s, t+T) = (n-1)U_\Delta + U(t-\Delta, s) - U(t, t+T) \leq nU_\Delta - U(t, t+T).$$

For $n = n^* - 1$ the summation in Equation (7.6) ranges from $s = (n^*-1)\Delta > t - \Delta$ to $t + T$ and hence the above distinction is of no concern.

Overall Equation (7.6) can be written as

$$\begin{aligned}&\sum_{n=0}^{n^*-1}\Big(\sum_{s=n\Delta}^{t+T}\mathbb{E}(e^{\theta A(s-n\Delta,t)-\theta U(s,t+T)+\theta nU_\Delta})e^{-\theta n\Sigma}\Big) + \mathbb{E}(e^{\theta n^* U_\Delta})e^{-\theta n^*\Sigma}\\ &\leq \sum_{n=0}^{n^*-2} e^{-\theta n\Sigma}\Big(\sum_{s=n\Delta}^{t-\Delta}\mathbb{E}(e^{\theta A(s-n\Delta,t)-\theta U(s,t-\Delta)-\theta U(t,t+T)+\theta(n-1)U_\Delta})\\ &\qquad + \sum_{s=t-\Delta+1}^{t+T}\mathbb{E}(e^{\theta A(s-n\Delta,t)-\theta U(t,t+T)+\theta nU_\Delta})\Big)\\ &\quad + e^{-\theta(n^*-1)\Sigma}\sum_{s=(n^*-1)\Delta}^{t+T}\mathbb{E}(e^{\theta A(s-(n^*-1)\Delta,t)-\theta U(t,t+T)+\theta(n^*-1)U_\Delta})\\ &\quad + \mathbb{E}(e^{\theta n^* U_\Delta})e^{-\theta n^*\Sigma}.\end{aligned}$$

All expectations appearing in the above term can be separated without using Hölder's inequality, as the increments of U are assumed to be stochastically independent. Doing so and applying the MGF-bounds leads to

$$\begin{aligned}
&\mathbb{P}(\mathfrak{d}(t) > T) \\
&\leq \sum_{n=0}^{n^*-2} e^{-\theta n\Sigma}\Big(\sum_{s=n\Delta}^{t-\Delta} e^{\theta(\rho_A(\theta)(t-s+n\Delta)+\rho_U(\theta)(T+t-\Delta-s)+\bar{\rho}_U(\theta)(n-1)\Delta)} \\
&\qquad + \sum_{s=t-\Delta+1}^{t+T} e^{\theta(\rho_A(\theta)(t-s+n\Delta)+\rho_U(\theta)T+\bar{\rho}_U(\theta)n\Delta)}\Big) \\
&\quad + e^{-\theta(n^*-1)\Sigma} \sum_{s=(n^*-1)\Delta}^{t+T} e^{\theta\rho_A(\theta)(t-s+(n^*-1)\Delta)+\theta\rho_U(\theta)T+\theta(n^*-1)\Delta\bar{\rho}_U(\theta)} \\
&\quad + e^{\theta n^*\bar{\rho}_U(\theta)\Delta} e^{-\theta n^*\Sigma}.
\end{aligned}$$

The summation over $s = n\Delta, \ldots, t-\Delta$ can be reindexed as a summation over $s' = t-(n+1)\Delta, \ldots, 0$ resulting in the geometric sum

$$\begin{aligned}
&e^{\theta\rho_A(n+1)\Delta+\bar{\rho}_U(\theta)(n-1)\Delta+\theta\rho_U(\theta)T} \sum_{s'=0}^{t-(n+1)\Delta} e^{\theta(\rho_A(\theta)+\rho_U(\theta)s'} \\
&\leq e^{\theta\rho_U(\theta)T} \frac{e^{\theta\rho_A(n+1)\Delta+\bar{\rho}_U(\theta)(n-1)\Delta}}{1-e^{\theta(\rho_A(\theta)+\rho_U(\theta))}}.
\end{aligned}$$

Further, the summation over $s = t-\Delta+1, \ldots, t+T$ is bounded by

$$\begin{aligned}
&\sum_{s=t-\Delta+1}^{t+T} e^{\theta\rho_A(\theta)(t-s+n\Delta)+\theta\rho_U(\theta)T+\theta\bar{\rho}_U(\theta)n\Delta} \\
&\leq (T+\Delta)e^{\theta(\rho_A(\theta)+\bar{\rho}_U(\theta))n\Delta+\theta\rho_A(\theta)(\Delta-1)+\theta\rho_U(\theta)T}
\end{aligned}$$

and the third summation over s is, since $n^*\Delta - 1 \geq t \geq (n^*-1)\Delta$, similarly bounded by

$$\begin{aligned}
&\sum_{s=(n^*-1)\Delta}^{t+T} e^{\theta\rho_A(\theta)(t-s+(n^*-1)\Delta)+\theta\rho_U(\theta)T+\theta(n^*-1)\Delta\bar{\rho}_U(\theta)} \\
&\leq (T+\Delta)e^{\theta(\rho_A(\theta)+\bar{\rho}_U(\theta))t+\theta\rho_U(\theta)T}.
\end{aligned}$$

Applying all of these inequalities results in

$$
\begin{aligned}
&\mathbb{P}(\mathfrak{d}(t) > T) \\
&\leq e^{\theta\rho_U(\theta)T} \sum_{n=0}^{n^*-2} e^{-\theta n\Sigma} \Big(\frac{e^{\theta\rho_A(n+1)\Delta + \bar{\rho}_U(\theta)(n-1)\Delta}}{1 - e^{\theta(\rho_A(\theta)+\rho_U(\theta))}} \\
&\qquad + (T+\Delta) e^{\theta(\rho_A(\theta)+\bar{\rho}_U(\theta)n\Delta + \rho_A(\theta)(\Delta-1))} \Big) \\
&\quad + e^{-\theta(n^*-1)\Sigma}(T+\Delta) e^{\theta(\rho_A(\theta)+\bar{\rho}_U(\theta))t + \theta\rho_U(\theta)T} \\
&\quad + e^{\theta n^* \bar{\rho}_U(\theta)\Delta} e^{-\theta n^*\Sigma} \\
&\leq e^{\theta\rho_U(\theta)T} \Big(\frac{e^{\theta\rho_A\Delta - \theta\bar{\rho}_U(\theta)\Delta}}{1 - e^{\theta(\rho_A(\theta)+\rho_U(\theta))}} + (T+\Delta) e^{\theta\rho_A(\theta)(\Delta-1)} \Big) \sum_{n=0}^{n^*-2} e^{\theta n(\rho_A(\theta)\Delta + \bar{\rho}_U(\theta)\Delta - \Sigma)} \\
&\quad + e^{\theta\Sigma}(T+\Delta) e^{\theta n^*(\rho_A(\theta)\Delta + \bar{\rho}_U(\theta)\Delta - \Sigma) + \theta\rho_U(\theta)T} \\
&\quad + e^{\theta n^*(\bar{\rho}_U(\theta)\Delta - \Sigma)}.
\end{aligned}
$$

Now under the stability condition $\rho_A(\theta)\Delta + \bar{\rho}_U(\theta)\Delta < \Sigma$ the above geometric sum is convergent and the two last expressions vanish for $t \to \infty$, since $n^* \to \infty$. This eventually leads to

$$
\begin{aligned}
&\mathbb{P}(\mathfrak{d}(t) > T) \\
&\leq \frac{e^{\theta\rho_U(\theta)T}}{1 - e^{\theta(\rho_A(\theta)\Delta + \bar{\rho}_U(\theta)\Delta - \Sigma)}} \left(\frac{e^{\theta\rho_A(\theta)\Delta - \theta\bar{\rho}_U(\theta)\Delta}}{1 - e^{\theta(\rho_A(\theta)+\rho_U(\theta))}} + (T+\Delta) e^{\theta\rho_A(\theta)(\Delta-1)} \right) + \varepsilon(t)
\end{aligned}
$$

□

This concludes the SNC-bound for a feedback loop of the form $\pi_{fb} = \pi_{+\Sigma} \circ \pi_\Delta \circ \pi_U$. Very related to this is the report [**139**] by Shekaramiz, Liebeherr, and Burchard. There feedback-loops consist of an additive service element $U \in \tilde{\mathcal{F}}_+$ and a fixed-delay element. Yet, the used methods are different from the above in the way the fixed-delay element is handled. Corollary 7.12 works on the level of σ-additive operators to achieve a commutation, whereas in [**139**] the system is analyzed on the level of convolutions. There the fixed-delay element in conjunction with the window-element is exploited. Wether the approach presented in [**139**] can be generalized to U resulting from the operations $\ominus$ or $\otimes$ is an open question. The difficulty lies in preserving $U \in \tilde{\mathcal{F}}_+$(to apply Theorem 6.9) and keeping the additivity of U. For example: The subtraction of crossflows leads, in general, to $U \notin \tilde{\mathcal{F}}_+$. This holds even if the leftover service is defined by the increments $u(t) = u_o(t) - a_U(t)$. Here u_o and a_U are the increments of the original server and the crossflows, respectively. This can be avoided by subtracting a crossflow on the level of increments from a *strict* dynamic U-server, i.e., by defining U via $u(t) = [u_o(t) - a_U(t)]^+$.

7.4. The General Case

The two previous sections gave examples for which the stochastic analysis of the WFC systems is successful; however, these results are strongly connected to the subadditivity of U_{res} or U, respectively. This assumption is rather strict and Theorem 7.8 must be employed, if multiple service elements are involved. As a consequence the available service inside the feedback loop is, in general, underestimated. Eventually leading to looser performance bounds. Clearly one would like to drop the assumption of subadditivity and deal with general feedback loops instead.

This section shows a way to separate the assumption of subadditivity from the performance bounds. To that end, two cases are distinguished: 1) the subadditivity of the feedback loop holds and the results of Section 7.2 apply; or 2) the subadditivity is not given and thus no performance bound either. Yet, the probability of the second case happening can be stochastically bounded. Performance bounds follow by combining these two cases into an overall violation probability. In that way, this section's main idea is similar to the one in Chapter 4. The two cases distinguished there are the events that the statistic delivers a valid MGF-bound or not. In this chapter the events to distinguish are that the feedback loop shows a subadditive behavior or not.

Recalling the solution of Theorem 6.5 an MGF-bound is needed for

$$U_{sys} = \Big(\bigwedge_{n=0}^{\infty} (U_{fb})^{(n)} \Big) \otimes U(s,t).$$

Section 7.1 presents the result of a straightforward application of the usual MGF-bound techniques. The generalized Hölder inequality for each self-convolution $U_{fb}^{(n)} \otimes U$ leads to difficulties and to a fast divergence of the resulting bounds in t.

Here another path is opened by Condition (6.5), which is repeated here: Let $U_{fb} = U_{res} \otimes \mathbf{1}_\Sigma = U \otimes U_* \otimes \mathbf{1}_\Sigma$ and fix some $t \in \mathbb{K}$. The service U_{fb} is subadditive, if the condition

$$\inf_{s \le r \le t} \{U_{res}(s,r) + U_{res}(r,t)\} \ge U_{res}(s,t) - K \tag{7.7}$$

holds for some positive constant $K \le \inf_{r \ge 0}\{\Sigma(r)\}$ and all $(s,t) \in \Lambda$. This condition is written differently as

$$\begin{aligned} & U_{res}(s,t) - U_{res}^{(2)}(s,t) \le K \qquad \text{for all } (s,t) \in \Lambda \\ \Leftrightarrow\quad & U_{res} \oslash U_{res}^{(2)}(t,t) \le K \end{aligned} \tag{7.8}$$

The following lemma summarizes the situation.

Lemma 7.13. *Fix some t. Under Condition (7.7) the feedback system is a dynamic U_{res}-server.*

Proof. For all $(s,t) \in \Lambda$ it holds by Corollary 6.10

$$\begin{aligned} U_{sys}(s,t) &= (\mathbf{1} \wedge U_{fb}) \otimes U(s,t) \\ &= U(s,t) \wedge \inf_{s \le r \le t}\{U_{res}(s,r) + \Sigma(r) + U(r,t)\} \\ &\ge U(s,t) \wedge U_{res} \otimes U_{res}(s,t) + \Sigma_{min}(t) \\ &\ge U(s,t) \wedge U_{res}(s,t) - K + \Sigma_{min}(t) \ge U(s,t) \wedge U_{res}(s,t) = U_{res}(s,t). \end{aligned} \tag{7.9}$$

□

As the bivariate U_{res} is a stochastic process, Condition (7.7) is not a deterministic event; thus, the probability of Equation (7.7) being fulfilled comes into focus. To this end, fix some $t \in \mathbb{K}$ and denote by $S(t)$ the event that Condition (7.7) holds.

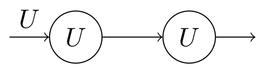


FIGURE 7.3. Equation (7.8) can be interpreted as a backlog bound in the above system.

Then

$$\begin{aligned}\mathbb{P}(\mathfrak{d}_{sys}(t) > T) &= \mathbb{P}(\mathfrak{d}_{sys}(t) > T \mid S(t))\mathbb{P}(S(t)) + \mathbb{P}(\mathfrak{d}_{sys}(t) > T \mid \neg S)\mathbb{P}(\neg S(t)) \\ &\leq \mathbb{P}(\mathfrak{d}_{U \otimes U_*}(t) > T) + \mathbb{P}(\neg S(t)),\end{aligned}$$

where $\mathbb{P}(\mathfrak{d}_{U\otimes U_*}(t) > T)$ is calculated by applying Theorem 1.7 and Theorem 2.39. If $\mathbb{P}(\neg S(t))$ is bounded, the above gives a delay bound for the WFC system.

THEOREM 7.14. *Fix some $t \in \mathbb{K}$ for condition $S(t)$. If $(\sigma, \rho) \preceq U_{res} \preceq (\bar{\sigma}, \bar{\rho})$, then*

$$\mathbb{P}(\neg S(t)) \leq e^{-\theta K + \theta(\bar{\sigma}(p\theta) + \sigma(qp'\theta) + \sigma(qq'\theta))} L \sum_{s'=0}^{t} e^{\theta(\bar{\rho}(p\theta) + \theta\rho(qp'\theta))s'},$$

where $\frac{1}{p} + \frac{1}{q} = \frac{1}{p'} + \frac{1}{q'} = 1$ and L is a constant given in the proof.

PROOF. The alternative representation (7.8) has the interpretation of a backlog bound. The corresponding system consists of a "flow"' U and a dynamic $U \otimes U$-server (see Figure 7.3). As the flow and service are stochastically dependent the Hölder parameters $\frac{1}{p} + \frac{1}{q} = \frac{1}{p'} + \frac{1}{q'} = 1$ are introduced. By Theorem 2.39 it holds

$$\begin{aligned}&\mathbb{P}(\mathfrak{b}(t) > x) \\ \leq& \mathbb{P}(U_{res} \oslash U_{res}^{(2)}(t,t) > x) \leq e^{-\theta x} \sum_{s=0}^{t} e^{\theta\bar{\rho}(p\theta)(t-s) + \theta\bar{\sigma}(p\theta)} \big(\mathbb{E}(e^{-\theta q U \otimes U(s,t)}\big)^{1/q} \\ \leq& e^{-\theta x + \theta\bar{\sigma}(p\theta)} \sum_{s=0}^{t} e^{\theta\bar{\rho}(p\theta)(t-s)} \Big(\sum_{r=s}^{t} e^{\theta q\rho(qp'\theta)(r-s) + \theta q\sigma(qp'\theta)} e^{\theta q\rho(qq'\theta)(t-r) + \theta q\sigma(qq'\theta)} \Big)^{1/q} \\ \leq& e^{-\theta x + \theta\bar{\sigma}(p\theta) + \theta\sigma(qp'\theta) + \theta\sigma(qq'\theta)} \\ &\cdot \sum_{s=0}^{t} e^{\theta\bar{\rho}(p\theta)(t-s)} e^{\theta\rho(qp'\theta)(t-s)} \Big(\sum_{r'=0}^{t-s} e^{\theta q(\rho(qq'\theta) - \rho(qp'\theta))r'} \Big)^{1/q}. \end{aligned} \tag{7.10}$$

Assume that w.l.o.g. $q' > p'$. Then the last sum converges for $t \to \infty$. The corresponding series' limit is denoted by L and the above term can be written as

$$\begin{aligned}\mathbb{P}(\mathfrak{b}(t) > x) &\leq e^{-\theta x + \theta(\bar{\sigma}(p\theta) + \sigma(qp'\theta) + \sigma(qq'\theta))} L \sum_{s=0}^{t} e^{\theta\bar{\rho}(p\theta)(t-s) + \theta\rho(qp'\theta)(t-s)} \\ &\leq e^{-\theta b + \theta(\bar{\sigma}(p\theta) + \sigma(qp'\theta) + \sigma(qq'\theta))} L \sum_{s'=0}^{t} e^{\theta(\bar{\rho}(p\theta) + \theta\rho(qp'\theta))s'}.\end{aligned}$$

□

REMARK 7.15. The above sum does not converge in general. This is due to the three inequalities:

$$\begin{aligned} \bar{\rho}(\theta) &\geq -\rho(\theta) && \text{for all } \theta \geq 0, \\ \bar{\rho}(p\theta) &\geq \bar{\rho}(\theta) && \text{for all } p \geq 1, \\ \rho(q\theta) &\geq \rho(\theta) && \text{for all } q \geq 1. \end{aligned}$$

For the case $q' = p' = 2$, the exponents in the last sum of Line (7.10) vanish and the sum reduces to $t - s + 1 \leq e^{\theta(t-s)\frac{1}{\theta}}$; hence, the bound becomes

$$\mathbb{P}(q(t) > b) \leq e^{-\theta b + \theta(\bar{\sigma}(p\theta) + 2\sigma(2q\theta))} \sum_{s'=0}^{t} e^{\theta(\bar{\rho}(p\theta) + \theta\rho(2q\theta) + \frac{1}{q\theta})s'}.$$

The system in Figure 7.3 is unstable (the concatenation of the two servers with service $U \otimes U$ is at most as large as the arrivals U). By this the bound of Theoerm 7.14 is only valid for finite time t. Still it has advantages over the bound in Equation (7.3) as it remains analytically tractable.

Overall, the stochastic performance bound for the general feedback loop is given in the next theorem.

THEOREM 7.16. *Consider a WFC system as in Figure 6.1 with the placeholder $*$ being a dynamic U_*-server. Let $\frac{1}{p} + \frac{1}{q} = \frac{1}{p'} + \frac{1}{q'} = 1$ and $q' > p'$. Define $\Sigma_{min}(t) = \inf_{0 \leq r \leq t}\{\Sigma(r)\}$. If $(\sigma, \rho) \preceq U_{res} = U \otimes U_* \preceq (\bar{\sigma}, \bar{\rho})$, the whole system fulfills the probabilistic delay bound*

$$\mathbb{P}(\mathfrak{d}_{sys}(t) > T) \leq \mathbb{P}(\mathfrak{d}_{U_{res}}(t) > T) + e^{-\theta\Sigma_{min}(t+T) + \sigma_{tot}(\theta,p,p')} B \sum_{s=0}^{t+T} e^{\theta\rho_{tot}(\theta,p,p')s}$$

with

$$\begin{aligned} \sigma_{tot}(\theta,p,p') &= \overline{\sigma}(p\theta) + \sigma(qp'\theta) + \sigma(qq'\theta), \\ \rho_{tot}(\theta,p,p') &= \begin{cases} \overline{\rho}(p\theta) + \rho(qp'\theta) & \text{if } p' \neq 2 \\ \overline{\rho}(p\theta) + \rho(2q\theta) + \frac{1}{q\theta} & \text{if } p' = 2 \end{cases}, \\ B &= \begin{cases} \left(\frac{1}{1 - e^{\theta q(\rho(qq'\theta) - \rho(qp'\theta))}}\right)^{1/q} & \text{if } p' \neq 2 \\ 1 & \text{if } p' = 2 \end{cases} \end{aligned}$$

and $\mathfrak{d}_{U_{res}}$ being the delay of an unthrottled tandem consisting of a dynamic U_{res}-server.

REMARK 7.17. Step (7.9) in Lemma 7.13 is not a necessary one. One could, for example, choose $K < \Sigma_{min}(t)$ and continue with

$$U_{sys}(s,t) \geq U(s,t) \wedge U_{res}(s,t) + \Sigma_{min}(t) - K. \tag{7.11}$$

Theorem 7.16 applies to this U_{sys} (which is at least as large as U_{res}) also. This procedure has the interpretation of shifting the violation probability towards the subadditive part (event $S(t)$) of the bound. This tradeoff is analyzed in more detail in the numerical evaluation.

The next example is used in the numerical evaluation of Section 7.6.

EXAMPLE 7.18. Let $\mathbb{K} = \mathbb{N}_0$ and assume that U and U_* are two constant rate servers that serve high-priority crossflows A_U and A_*, respectively. The service elements' rates are u and u^*, respectively, with $u > u^*$. If $A_U \preceq (\sigma_{A_U}(\theta), \rho_{A_U}(\theta))$ and $A_{U^*} \succeq (\sigma_{A_*}(\theta), \rho_{A_*}(\theta))$, it holds

$$\mathbb{P}(\neg S(t)) \leq e^{\theta(\sigma_{A_U}(\theta)+\sigma_{A_{U^*}}(\theta))} \sum_{r=0}^{t} \sum_{r'=1}^{t-r} e^{\theta r'(u-u^*+\rho_{A_U}(\theta)+\rho_{A_*}(\theta))} \tag{7.12}$$

for all $t \in \mathbb{N}_0$. This is seen as follows: Fix any triple of times $s \leq r \leq t \in \mathbb{N}_0$. It holds

$$\begin{aligned}
&U \otimes U_*(s,r) \\
&= \min_{s \leq r' \leq r} \{u(r'-s) + u^*(r-r') - A_U(s,r') - A_*(r',r)\} \\
&= \min_{s \leq r' \leq r} \{u(r'-s) + u^*(t-r') - u^*(t-r) - A_U(s,r') - A_*(r',t) + A_*(r,t)\} \\
&\geq U \otimes U_*(s,t) - u^*(t-r) + A_*(r,t).
\end{aligned}$$

Now assume that it held $U \otimes U_*(r,t) \geq u^*(t-r) - A_*(r,t)$, then

$$U \otimes U_*(s,r) + U \otimes U_*(r,t) \geq U \otimes U_*(s,t)$$

would follow.

Hence,

$$\mathbb{P}(\neg S(t)) \leq \mathbb{P}(\exists r \leq t : U \otimes U_*(r,t) < u^*(t-r) - A_*(r,t)).$$

The condition inside the probability is equivalent to the existence of an $r \leq t$ such that

$$\begin{aligned}
0 &> \min_{r \leq r' \leq t} \{u(r'-r) + u^*(t-r') - A_U(r,r') - A_*(r',t)\} - u^*(t-r) + A_*(r,t) \\
&= \min_{r \leq r' \leq t} \{(u-u^*)(r'-r) - A_U(r,r') + A_*(r,r')\}.
\end{aligned}$$

Hence,

$$\begin{aligned}
\mathbb{P}(\neg S(t)) &\leq \mathbb{P}\Big(\bigcup_{r=0}^{t} \bigcup_{r'=r+1}^{t} A_U(r,r') - A_*(r,r') > (u-u^*)(r'-r)\Big) \\
&\leq \sum_{0 \leq r < r' \leq t} \mathbb{P}(A_U(r,r') - A_*(r,r') > (u-u^*)(r'-r)) \\
&\leq \sum_{0 \leq r < r' \leq t} e^{\theta(u-u^*)(r'-r)} \mathbb{E}(e^{\theta A_U(r,r') - \theta A_*(r,r')}) \\
&\leq e^{\theta(\sigma_{A_U}(\theta)+\sigma_{A_*}(\theta))} \sum_{0 \leq r < r' \leq t} e^{\theta(u-u^*)(r'-r)} e^{\theta(\rho_{A_U}(\theta)(r'-r) - \rho_{A_*}(\theta)(r'-r))}.
\end{aligned}$$

From the above Equation (7.12) follows by the substitution $r' - r = r''$.

By changing the roles of U and U_* in the above example, it also holds

$$\mathbb{P}(\neg S(t)) \leq e^{\theta(\sigma_{A_U}(\theta)+\sigma_{A_*}(\theta))} \sum_{r=0}^{t} \sum_{r'=1}^{t-r} e^{\theta r'(u^*-u+\rho_{A_*}(\theta)+\rho_{A_U}(\theta))}$$

for the case $u^* > u$.

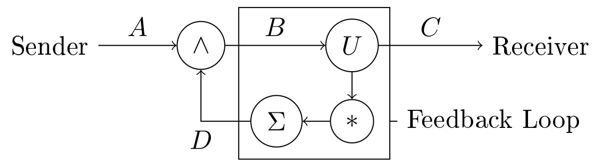


FIGURE 7.4. A window flow controlled system. The element denoted by $*$ is a placeholder.

7.5. Numerical Evaluation of Delay Bounds

This section investigates how the bounds derived in Theorems 7.8 and 7.16 evolve in their parameters. Further, the impact of window flow control on the system's delay is quantified, by comparing it to a similar unthrottled system. In this section U is a constant rate server $U_o(s,t) = u(t-s)$ that serves a high priority crossflow $A_U(s,t)$. Similarly the placeholder in Figure 7.4 is a dynamic U_*-server that also offers a constant rate $U_{*,o}(s,t) = u_*(t-s)$ to a high priority crossflow $A_*(s,t)$, such that $U_*(s,t) = u_*(t-s) - A_*(s,t)$. Both service descriptions are subadditive by themselves but, as Example 7.5 shows, their subadditivity is lost for $U_{res} = U \otimes U_*$. The entire evaluation is performed for slotted time ($\mathbb{K} = \mathbb{N}_0$).

The service element's rate u_* is set to be greater than u to account for the typically smaller size of acknowledgments flowing back to the throttle. The crossflows in this example consist of independent, exponentially distributed increments $a_U(t)$ and $a_*(t)$, respectively. The arrivals at the WFC system, denoted by A also consist of independent, exponentially distributed increments. All flows are stochastically independent of each other. It would be possible to analyze more sophisticated crossflows or to drop the independency assumption. Yet, this is not the focus of this evaluation. Further the window size Σ is held constant, for all times $t \geq 0$.

A corresponding unthrottled system would just consist of the flow A being fed into the service element U.

To achieve reasonable values in the bounds of Theorem 2.39 and Theorem 7.16, the parameter θ and the Hölder pairs p, q, p', q' are numerically optimized. When not specified otherwise, the parameters in all calculations are set as follows: The bound is taken at time $t = 5$ and asks for a delay $T = 10$, i.e., the probability $\mathbb{P}(d_{sys}(5) > 10)$ is searched for. The parameter of the exponential distributions for the arrivals and crossflows is $\lambda = 4$ (three flows have the same rate λ for simplicity), while the server rates are $u = 1$ and $v = 2$. This corresponds to a utilization of 50% and 25%, respectively. Results are presented for a window size of $\Sigma = 15$.

7.5.1. Throttled vs. Unthrottled System. First the system is compared with its unthrottled counterpart. To that end, the arrival rates λ are varied, resulting in utilizations from 30% to 80%. The corresponding violation probabilities are plotted against the performance bounds on a logarithmic scale for the throttled and the unthrottled system. Different window sizes $\Sigma = 10$, 15, 20 are considered. The results are displayed in Figure 7.5 as black and blue lines for the throttled and unthrottled system, respectively. As expected, the throttled system behaves better, the greater Σ is. For $\Sigma = 20$ the throttled system behaves almost identically to the unthrottled one.

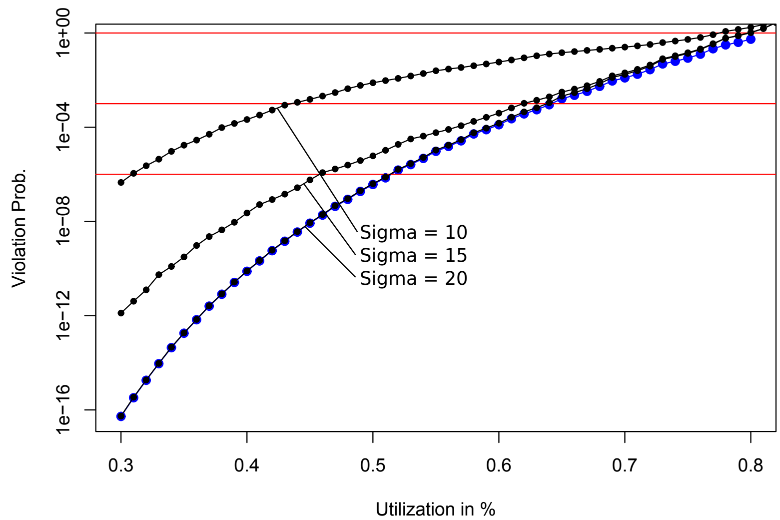


FIGURE 7.5. A graph showing the violation probabilities depending on the utilization of throttled (black) and unthrottled (blue) systems for different window sizes. The red lines are equal to 1, 10^{-3} and 10^{-6}.

7.5.2. Dependence on Delay. A major difference between the unthrottled and the throttled analysis lies in the dependency on the delay T. While for the unthrottled system an increase in T leads to a decrease in the violation probability, the bound of Theorem 7.16 increases in the term $\mathbb{P}(\neg S(t))$ for increasing T. In Figure 7.6, one sees for the black line how the bound evolves for an increasing T. The two red lines show how the bound differs when choosing $b = \frac{\Sigma}{2}, \frac{9\Sigma}{10}$ as suggested in Remark 7.17 (Equation (7.11)). The trend here is that for increasing T the bound becomes worse, the greater the difference between K and Σ is. However, for small values of T there is a very slight improvement for $K = \frac{9\Sigma}{10}$ and even for $K = \frac{\Sigma}{2}$. In this scenario, trading a higher violation probability for the event $S(t)$ is not worthwhile the gain from a better service description U_{sys}.

To investigate the composition of the delay bound further, Figure 7.7 separates the two parts of the bound. The blue circles correspond to the delay part of the violation probability $\mathbb{P}(\mathfrak{d}_{U_{res}}(t) > T)$ and the red circles correspond to the violation probability of event $S(t)$, while the solid black circles are the sum of both. From a certain point onwards the probability $\mathbb{P}(\neg S(t))$ dominates the overall violation probability. The additional lines in the graph show how the different parts of the bound are affected when setting $K = \frac{\Sigma}{2}, \frac{3\Sigma}{4}, \frac{9\Sigma}{10}$ in Equation (7.11). The delay part (blue) of the probability experiences no considerable change, while the probability of violating event $S(t)$ (red) increases significantly, when $K < \Sigma$.

7.5.3. Convergence to Unthrottled System. Figure 7.8 shows the convergence of the throttled system towards the unthrottled one, when increasing the window size. From Theorem 7.16 the violation probability $\mathbb{P}(\neg S(t))$ vanishes for increasing window sizes. However, the throttled system does not fully converge to the unthrottled one. This is due to $\mathbb{P}(\mathfrak{d}_U(t) > T) \leq \mathbb{P}(\mathfrak{d}_{U_{res}}(t) > T)$. The size of the gap, which cannot be closed by increasing the window size further, is completely

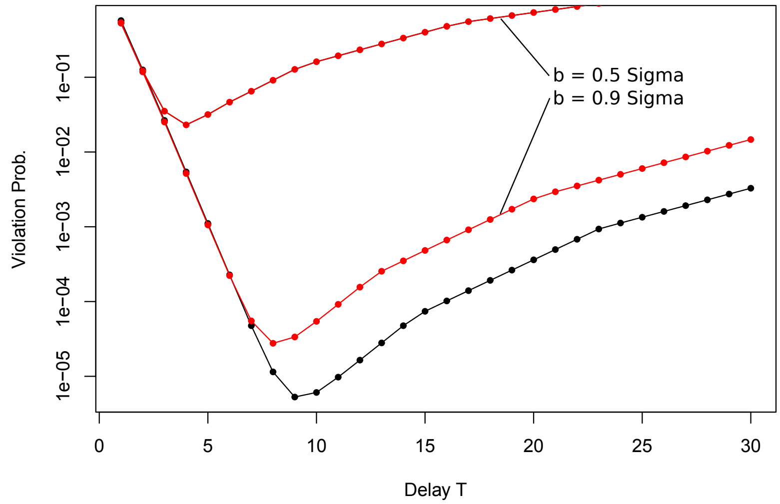


FIGURE 7.6. A graph showing how the bound evolves when increasing the delay T. The red colored lines represent a shift towards the violation probability of event $S(t)$.

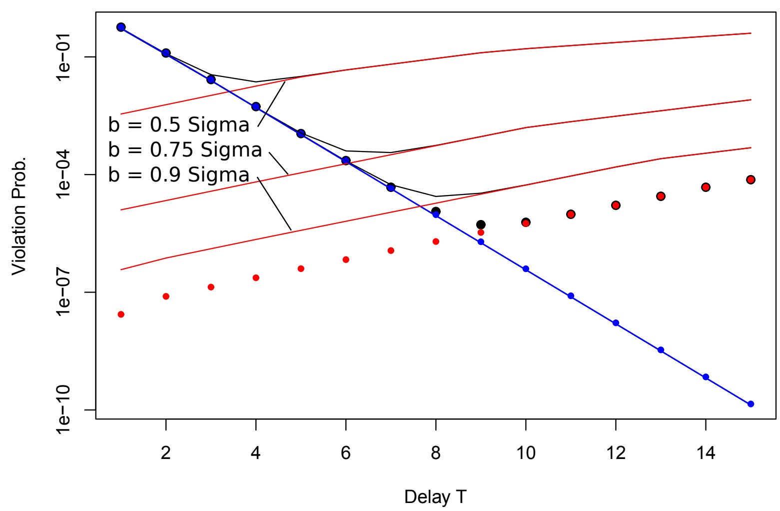


FIGURE 7.7. A graph showing the different components of the violation probability: The blue circles are the delay part, while the red circles represent the subadditivity part. The black circles are the sum of both parts. The lines show the same for different shifts towards the violation probability $\mathbb{P}(\neg S(t))$.

dependent on the service descriptions U and U_*. The graph also shows the results for systems with $u_* = 2$, 1.5, 1.1. The gap to the delay of the unthrottled system (red) increases, when reducing the rate of u_*.

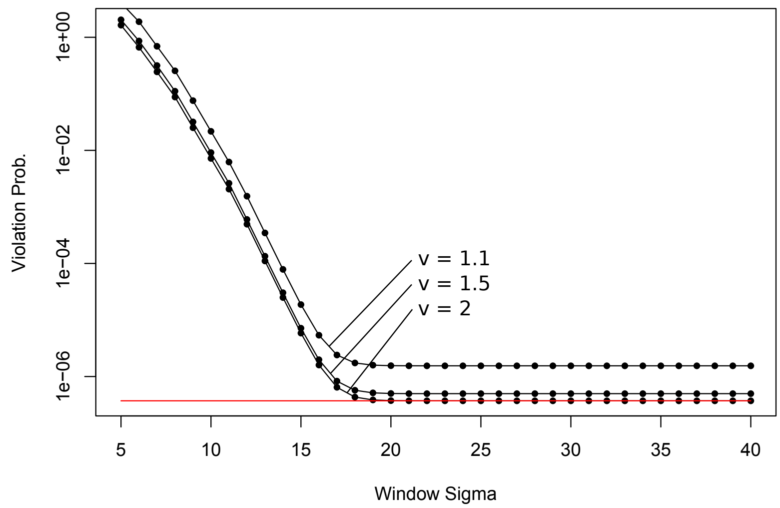


FIGURE 7.8. A graph showing the convergence of the throttled system towards the unthrottled one for increasing Σ. The red line is the bound for the unthrottled system. The black lines show the throttled system for different rates of v.

7.6. Admissible Flows in WFC

This Section compares the two methods derived in Theorems 7.8 and 7.16 to each other and with a deterministic analaysis. This is done in the context of an admission problem. The question therein is, "Given a queueing system, how many flows of the same type can be admitted to it, without exceeding a given performance bound?" In that sense, this evaluation follows the spirit of the one in [**106**]. There it is shown, how stochastic network calculus can, compared to a deterministic analysis, admit a much larger number of flows to an unthrottled system. This is due to the statistical multiplexing gain, which cannot be captured by deterministic network calculus.

The performance bound used in this example is a backlog constraint on the throttle.

7.6.1. Scenario Description. The scenario is similar to the one in the previous section: The feedback loop consists of two constant rate service elements with rates u and u^* and high-priority crossflows A_U and A_*. The crossflow A_U consists of N_U single flows that are multiplexed to form A_U. Each of the crossflows has an average rate λ_U. This leads to a utilization at the first node of $N_U\lambda_U/u$. The crossflow A_* at service element U_* follows the same structure with variables N_{U_*} and λ_{U_*}, respectively.

Also, the aggregate A consists of N_A flows, each with an average rate λ_A. The system is scaled such that $\lambda_A = 1$ and the rate at the first service element equals $u = 100$. By this admitting one subflow of A increases the utilization at the service element U by 1%. Hence, neglecting the window flow controller, the server U would be fully utilized after admitting $100(1 - N_U\lambda_U/u)$ flows.

The performance bound considered here is $\mathfrak{b}_\wedge \leq x$ for some $x \in \mathbb{R}^+$.

Deterministic Analysis. Arrival curves as in Definition 1.8 are needed for a deterministic analysis . A common model are leaky buckets (see Example 1.9), which result in affine arrival curves of the form $\mathcal{A}(t) = K + \lambda(t-1)$. The backlog bound at the throttle is given by the maximal vertical deviation between the throttle's service curve and the arrival curve of the aggregate (as presented in Figure 1.12). In this scenario $\mathfrak{b}_\wedge$ is bounded by

$$\mathfrak{b}_\wedge = N_A \left(K_A + \lambda_A \left(\frac{N_U(K_U - \lambda_U)}{u - \lambda_U N_U} + \frac{N_{U_*}(K_{U_*} - \lambda_{U_*})}{u^* - \lambda_{U_*} N_{U_*}} \right) - 1 \right) - \Sigma$$

and solving for N_A leads to the maximal number of admissible flows.

Stochastic Analysis. For the stochastic analysis assume that all flows are stochastically independent of each other. Further, the flows are assumed to have independent $\exp(\lambda_A^{-1})$-, $\exp(\lambda_U^{-1})$-, and $\exp(\lambda_V^{-1})$-distributed increments, respectively. The moment generating function of A_U is

$$\mathbb{E}(e^{\theta A_U(s,t)}) = \prod_{i=1}^{N_U} \mathbb{E}(e^{\theta A_{U,i}(s,t)}) = \left(\frac{\lambda_U}{\lambda_U - \theta} \right)^{N_U(t-s)}.$$

The MGF of A_{U_*} takes the same form.

The MGF of W as constructed in Theorem 7.8 reads

$$\mathbb{E}(e^{-\theta W(s,t)}) = \left(e^{-\theta |u-u^*|} \left(\frac{\lambda_U}{\lambda_U - \theta} \right)^{N_U} \left(\frac{\lambda_{U_*}}{\lambda_{U_*} - \theta} \right)^{N_{U_*}} \right)^{(t-s)}.$$

By inserting these bounds into Theorem 2.39 the inequality $\mathbb{P}(\mathfrak{b}_\wedge(t) > x) \leq \varepsilon$ can be solved for N_A and given ε, t. This corresponds to the method discussed in Section 7.2.

Alternatively Theorem 7.16 can be used. The scenario used in this evaluation equals the one described in Example 7.18. Since $\mathbb{P}(\neg S(t))$ joins the equation, the resulting number of admissible flows becomes dependent on the time t at which the backlog is evaluated.

7.6.2. Deterministic vs. Stochastic Bounds. For the first evaluation the number of admissible flows is plotted against the number of admitted crossflows A_U. Due to the normalization this corresponds to the utilization at U before any subflows of A are admitted: $\lambda_U N_U/100$. The average arrival rates are fixed to $\lambda_U = \lambda_{U_*} = \lambda_A = \lambda$, further the deterministic arrival curves have a peak rate $K = 10\lambda$.

First an asymmetric scenario is considered. In this the service element U_* has rate $u^* = 200$ and handles $U_* = 25$ crossflows. This reflects the circumstance that the feedback information can usually be easier processed than the system's arrivals. Figure 7.9 presents the number of admissible flows for $x = 50$, $\Sigma = 5$ and violation probabilities of $\varepsilon = 10^{-3}, 10^{-6}, 10^{-9}$. The event $S(t)$ is evaluated at $t = 1000$, but could have been chosen much larger, without changes to the results. Using a stochastic analysis turns out ot be clearly beneficial over the deterministic analysis for all utilizations. Further, in this scenario, the general method of Theorem 7.16 admits more flows to the system than the analysis with Theorem 7.8 does.

The quantity of admitted flows for $\lambda_U N_U = 0.5$ (corresponding to a utilization of 50% before admitting subflows of A) emphasizes this results: The deterministic analysis admits 2 flows, without breaking the backlog guarantee. This would result

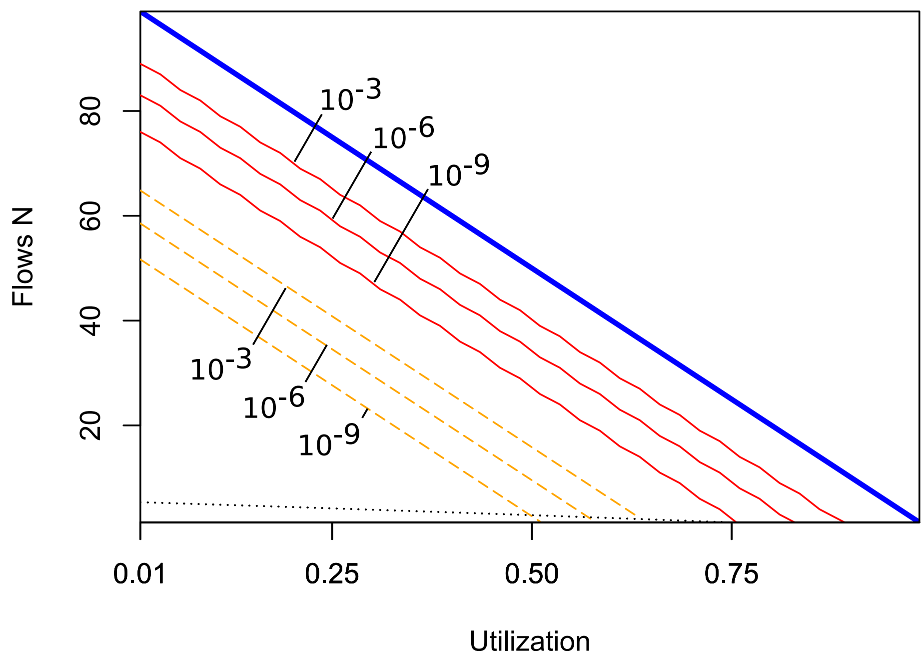


FIGURE 7.9. The number of admitted flows N_A for varying utilizations at U. The thick blue line represents an upper bound (stability condition). The dashed orange lines correspond to using Theorem 7.8 and the solid red lines to Theorem 7.16, respectively. The deterministic bound is given as a dotted black line.

(ignoring the flow control) in an overall utilization of 52% at the dynamic U-server. Using Theorem 7.8 allows up to 15 flows instead, when setting the violation probability to 10^{-3}. This corresponds to an overall utilization of 65%. With Theorem 7.16 the number of admitted flows ranges from 27 ($\varepsilon = 10^{-9}$) up to 40 ($\varepsilon = 10^{-3}$) or an overall utilization of 77-90%.

7.6.3. Dependence on Burst Sizes. The next evaluation considers different burst sizes of the arrivals. To that end, all parameters are chosen as before, but $\lambda = 0.25, 1, 25$. To achieve the same utilization at U_* the number of crossflows is adapted to $N_{U_*} = 100, 25, 10$, respectively. With fewer and burstier flows the statistical multiplexing gain decreases and one should expect the number of admitted flows to decrease. Figure 7.10 illustrates this effect.

7.6.4. Bounding $\mathbb{P}(\neg S(t))$. The above evaluations describe an asymmetric situation in which the service element U_* provides more service than U. This is beneficial for bounding $S(t)$, whereas the bound in Theorem 7.8 is rather loose. This is investigated in more detail now. To this end, the servers' rates are set equal now, i.e., $u^* = u = 100$. Further the number of crossflows at U_* is reduced to $N_{U_*} = 10$. This means that the two service elements are quite similar, when N_U is a small number. As a result, the bounding of $\mathbb{P}(\neg S(t))$ is harder, whereas the bound in Theorem 7.8 is tighter. For a large number of crossflows at U the opposite is the case, instead, as the service elements loose their similarity.

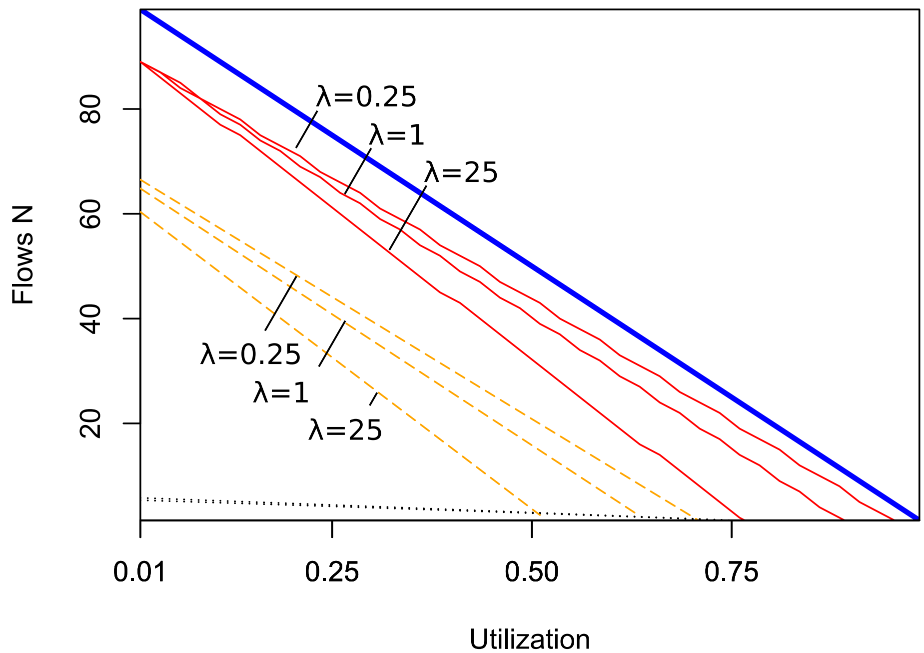


FIGURE 7.10. The number of admitted flows N for varying $\lambda_U = \lambda_{U^*}$. For the stochastic analysis the violation probability ε is set to 10^{-3} .

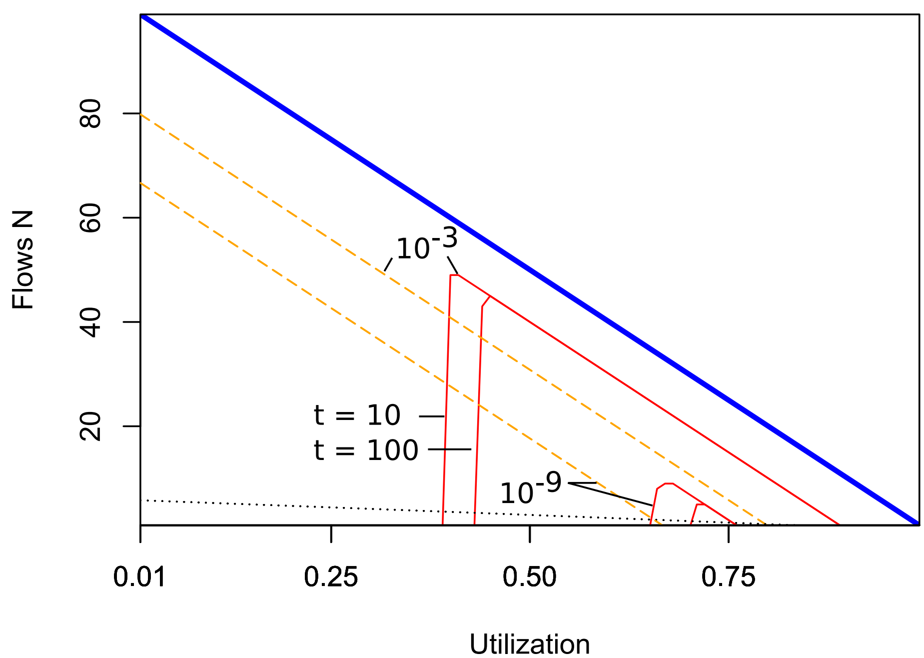


FIGURE 7.11. Number of admissible flows N for the case in which service elements U and U_* are similar in rate.

Figure 7.11 shows this effect: The method from Theorem 7.16 breaks down completey for low utilizations. In these cases the probability $\mathbb{P}(\neg S(t))$ exceeds ε and hence no flows can be admitted via this analysis. As the utilization at U rises, it reaches a point at which $\mathbb{P}(\neg S(t))$ can be bounded. From that point onwards Theorem 7.16 becomes viable again. The second pair of curves shows the same effect for $\varepsilon = 10^{-9}$.

Also, a significant boost of admissible flows for the first method can be noted. This is due to the reduced number of crossflows at U_*. Reducing u^* to the value 100 has no effect for Theorem 7.8 as its result uses the minimum of u and u^* anyways.

CHAPTER 8

Conclusion and Outlook

This chapter gives an overview of the achieved results; further, it discusses future directions with respect to the topics covered in this thesis.

Several contributions to the theory and application of stochastic network calculus were made:

(1) The *DISCO Stochastic Network Calculator* is the first – and so far only – open-source tool for applying stochastic network calculus. It distinguishes itself from tools for deterministic network calculus by separating the analysis and the numerical optimization. The analysis, which is performed on a symbolic level, applies the network operations derived in Chapter 2 to achieve a backlog- or delay-bound for the flow of interest. The resulting bound is given in form of a mathematical function. Afterwards this function is numerically optimized to obtain a near-optimal value for the performance measure. Due to its modular structure the analysis-step or the optimization-step can be completely substituted without changing other parts of the *Network Calculator*. This also holds for other parts of the program, like traffic- or server-models or the graphical user interface. The *DISCO Stochastic Network Calculator* gives the opportunity to investigate large scale and general feedforward networks with stochastic network calculus. Methods of an end-to-end analysis as known from deterministic network calculus (Total Flow Analysis, Separate Flow Analysis, Pay-Bursts-Only-Once,...) can be implemented. But also effects like the usage – or avoidance – of Hölder's inequality should be adressed and taken into account. The implementation of further traffic models, such as fractional Brownian motion, or the extension to feedback-systems are also items of future.

(2) The development of a statistical network calculus is an effective analytical tool to reduce assumptions about the involved stochastic processes. Obtaining knowledge about the arrivals or service via statistical methods introduces another uncertainty: The probability that the used statistic delivered a wrong result. This probability is usually captured by a confidence level α. Chapter 4 integrates this probability into the stochastic performance bound. The result is a methodology that captures both: the fluctuations of stochastic processes and the possibility that the used statistics underestimated the arrival's MGF. Furthermore, statistical network calculus gives a dynamic perspective on the system. This makes it possible to adapt to changes in the system or capture transient phases. Another effect of using statistics is a higher level of robustness against false modeling assumptions.
Future work includes the construction of further statistics. Fractional

Brownian motion as traffic model is one example here. But also advanced subsampling techniques are of interest. Further statistical methods for capturing stochastic dependencies and implementing those into the performance bounds are an interesting direction of future research [**56**].

(3) Stochastic performance bounds hold pointwisely, i.e., for specific values of t. Yet, in practice one is often interested in interval-valid bounds: What is the probability of exceeding a target-delay in the next minutes, hours, or days? So far, questions of this type are seldomly adressed in the SNC-literature. Chapter 5 evaluates the application of Boole's inequality and compares it against a new method that is based on the theory of extreme values. Results show no clear winner: In the regime of low-utilizations the bounds based on extreme value theory outperform stochastic network calculus. Further they are very robust to changes in the violation probability ε. However, in scenarios with high utilizations the bounds obtained by stochastic network calculus scale better with respect to the length of the considered sample-path. One of the major advantages of stochastic network calculus are their network operations. They allow to reduce the complexity of queueing networks. Thus, Chapter 5 also includes a full set of network operations for the EVT-based calculus.
The challenge of how to achieve sample-path performance bounds is not solved, yet. Chapter 5 shows, that invoking Boole's inequality leads to rather loose bounds, especially for low utilizations. Still EVT-based bounds do not scale well for higher utilizations. An overall good way to avhieve sample-path performance bounds is still missing. The usage of extreme value theory also offers a new approach to traffic models, which are hard to handle by stochastic network calculus: The assumptions on the arrivals made in Chapter 5 are different from the ones usually made in SNC (i.e., the existence of a moment generating function). The class of heavy-tailed arrivals might be analyzable via an EVT-approach, whereas they pose hard problems to an SNC-approach [**107**].

(4) The last part of this thesis presents the challenge of window flow controlled systems in detail. Two approaches towards a solution for a stochastic analysis are given. Here the key-property is the subadditivity of the feedback loop.
In the first approach the subadditivity is conserved by performing an appropriate network analysis. This network analysis differs from the usual methods towards an end-to-end description, as it must preserve subadditivity. Eventually this thesis characterizes a class of subadditive networks. It is also shown, that this characterization is not exhaustive: systems that include fixed-delay elements can be analyzed, but do not fall into the given characterization.
The second approach towards an analysis of window flow controlled systems utilizes the probabilistic nature of the problem. Instead of preserving subadditivity one asks for its likelihood instead. This probability is then integrated into the performance bounds. For this the probability that the feedback loop is not subadditive must be bounded. This can again be accomplished with the methods of stochastic network calculus.
Analyzing window flow controlled system with stochastic network calculus

had been an open problem for over a decade.
Yet, the challenge to analyze window flow controlled systems has not come to end. It has rather just begun. The here presented methods are not complete, as they miss desirable properties: Parts of the available service is lost, when one wants to preserve subadditivity directly. Further, the probability of the event $\{\neg S(t)\}$ cannot be bounded for $t \to \infty$, yet. Also the analysis of feedback-loops with fixed-delay elements has its shortcomings, as it is strongly tied to a special case of general feedback systems. Future research in this area is needed to overcome these obstacles.

Part 4

Appendix

CHAPTER 9

Network Calculus

9.1. Multicommodity Queueing Systems

As discussed in Chapter 1, in a service element the two commodities A and U meet to get processed into the commodity B. In this process the commodity A can be stored while any exceeding service U is lost. This is described by Lindley's equation, which states $\mathfrak{b}(t) = [\mathfrak{b}(t-1) + a(t) - u]^+$.

Instead, if both commodities are storable, backlog for A and U can occur. Under an instantaneous processing of the commodities A and U, if both are present at the service element, the backlogs are described by

$$\mathfrak{b}_A(t) = [A(t) - U(t)]^+, \qquad \mathfrak{b}_U(t) = [U(t) - A(t)]^+.$$

If both commodities are stored in the same buffer, the above equations combine to $\mathfrak{b}(t) = |A(t) - U(t)|$.

One might also consider both commodities not to be storable. In this case $\mathfrak{b}_A = \mathfrak{b}_U = 0$ and the output of the system, in slotted time, is $B(t) = \sum_{s=0}^{t} a(s) \wedge u(s)$.

Again: Lindley's equation and service curves describe the case in which one commodity is storable (in the field of communication networks: data) and one commodity is not storable (service). A general framework distinguishes the storable commodities A_i $(i \in I)$ and non-storable commodities U_j $(j \in J)$ that are to be transformed into a product B. Denote by B_i the output of a system that would result from A_i and the non-storable commodities only. In this case a general form of Lindley's equation leads to

$$B_i(t) = \min_{0 \le s \le t} \Big\{ A_i(s) + \sum_{r=0}^{t-s} \bigwedge_{j \in J} u_j(r) \Big\},$$

where $u_j(r)$ is the available non-storable commodity in time-slot r. As the output B is the minimum over all B_i, the system that works all storable commodities simultaneously is described by

$$B(t) = \bigwedge_{i \in I} A_i \otimes U(t),$$

with $U(t) = \sum_{s=0}^{t} \bigwedge_{j \in J} u_j(s)$. For the backlog of commodity A_k one gets in this scenario:

$$\mathfrak{b}_k(t) = \max_{0 \le s \le t} \Big\{ \bigvee_{i \in I} A_k(t) - A_i(s) - U(t-s) \Big\} \tag{9.1}$$

The literature gives some examples of storable commodities being different to data [**110, 22**]. One example is the charging and discharging of accumulators [**154, 153, 158, 103, 73**]. The storable commodity in this case is energy and the non-storable commodity is energy demand. The service element at which both commodities meet and get processed is the rechargeable battery. Still, in this example the service element processes exactly one storable and exactly one non-storable commodity. A network calculus using Equation (9.1) together with the dependencies between the commodities are nicely captured in the form of matrices by Chang et al. [**34, 36**]. Chapter 4 of [**35**] gives a good introduction to this topic and defines the matrix-valued network operations $\oplus, \ominus, \otimes, \oslash$.

9.2. Inverse Calculus

The goal of this section is to derive the deterministic performance bounds in another way. Let $\mathcal{A}$ and U be the arrival and service curve of a flow A and a service element, respectively. While the backlog bound is given as the simple vertical deviation $\mathfrak{b}(t) \leq \sup_{r\geq 0}\{\mathcal{A}(r) - U(r)\}$, the delay bound requires to calculate the horizontal deviation between $\mathcal{A}$ and U, which is harder to perform. The idea presented here is to consider generalized inverse functions of $\mathcal{A}$ and U. Indeed, as derived here, the virtual delay is bounded by the vertical deviation of the inverse functions. This Section is related to [**67**] of Fidler and Recker. There the min-plus operations $\otimes$ and $\oslash$ are expressed via the simple addition and subtraction of their Legendre transforms, respectively. There the performance bounds can also be calculated in the space of Legendre-transforms, although that is not the main goal of [**67**]. For a different derivation of Theorem 9.4 see the paper of Valaee [**150**]. See also Chapter 3 of the textbook [**102**] by LeBoudec and Thiran, which derives the horizontal deviation of two functions by their pseudo-inverses. The resulting operations of Lemma 9.1 and 9.2 are the max-plus deconvolution and max-plus convolution, respectively. For those see [**102**] and the work of Baccelli et al. [**7**].

Let $A \in \mathcal{F}(\mathbb{R})$ be left-continuous. Then its generalized inverse function is

$$\begin{aligned} A^{-1} :& \mathbb{R}_0^+ \to \mathbb{R}_0^+ \\ & x \mapsto \sup\{t \mid A(t) \leq x\}. \end{aligned}$$

LEMMA 9.1. *Let $A, \mathcal{A}$ be left-continuous. Then $A \preceq \mathcal{A}$, if and only if*

$$A^{-1}(x+y) - A^{-1}(y) \geq \mathcal{A}^{-1}(x)$$

for all $x, y \in \mathbb{R}_0^+$.

PROOF. For the *only if* part choose some $x, y \in \mathbb{R}_0^+$ and set $s^* = A^{-1}(y)$. Then

$$A(s^*) = A(A^{-1}(y)) \leq y$$

and

$$\begin{aligned} A^{-1}(x+y) &= \sup\{s' \mid A(s') \leq x+y\} \geq \sup\{s' \mid A(s') - A(s^*) \leq x\} \\ &= s^* + \sup\{s' \mid A(s'+s^*) - A(s^*) \leq x\} \\ &\geq s^* + \sup\{s' \mid \mathcal{A}(s') \leq x\} = A^{-1}(y) + \mathcal{A}^{-1}(x). \end{aligned}$$

For the *if* part choose some $s, t \in \mathbb{R}_0^+$ and set $x = \mathcal{A}(t)$. Then

$$\mathcal{A}^{-1}(x) = \sup\{s' \mid \mathcal{A}(s') \leq x\} = \sup\{s' \mid \mathcal{A}(s') \leq \mathcal{A}(t)\} = t$$

and hence $t \leq \sup\{s' \mid A(s') \leq \mathcal{A}(t) + y\} - A^{-1}(y)$ for all $y \in \mathbb{R}_0^+$. Set $y = A(s)$ to follow

$$\begin{aligned} t &\leq \sup\{s' \mid A(s') \leq \mathcal{A}(t) + A(s)\} - s = \sup\{s'' \mid A(s+s'') - A(s) \leq \mathcal{A}(t)\} \\ &= \sup\{s'' \mid A(s+s'') - A(s) \leq \mathcal{A}(t)\}. \end{aligned}$$

This implies $A(s+t) - A(s) \leq \mathcal{A}(t)$. □

The above lemma shows that arrival curves are equivalent to bounds on arrivals in the time domain. The arrival curve states that the number of arrivals in an interval of length t cannot exceed $\mathcal{A}(t)$. Similarly the inequality $A^{-1}(x+y) - A^{-1}(y) \geq \mathcal{A}^{-1}(x)$ means that x arrivals cannot occur in intervals shorter than$\mathcal{A}^{-1}(x)$. The next lemma shows a similar equivalence for service curves U and the inequality

$$B^{-1}(x) \leq \sup_{0 \leq y \leq x} \{A^{-1}(y) + U^{-1}(x-y)\}. \tag{9.2}$$

Service curves make a statement about the minimal departures of a system. The above inequality bounds the time needed for x departures from above.

LEMMA 9.2. *Consider a service element with arrivals A. Denote the departures by B. The service element offers service curve U if and only if Equation (9.2) holds for all $x \in \mathbb{R}_0^+$.*

PROOF. For the *only if* part fix some $x \in \mathbb{R}_0^+$. It is to show there exists a $y \leq x$ such that $B^{-1}(x) \leq A^{-1}(y) + U^{-1}(x-y)$. Define $t = B^{-1}(x)$. It follows as in the previous lemma $B(t) \leq x$. By definition of the service curve exists a $s \in [0, t]$ with

$$B(t) \geq A(s) + U(t-s).$$

Define $y = A(s)$ from which follows $A^{-1}(y) = s$ as in the previous lemma. Since $U(t-s) \geq 0$ it holds $y = A(s) \leq B(t) \leq x$. Further,

$$\begin{aligned} A^{-1}(y) + U^{-1}(x-y) = s + U^{-1}(x-y) &= s + \sup\{s' \mid U(s') \leq x - A(s)\} \\ &= \sup\{s'' \mid U(s''-s) + A(s) \leq x\} \\ &\geq \sup\{s'' \mid U(s''-s) + A(s) \leq B(t)\} \geq t = B^{-1}(x). \end{aligned}$$

For the *if* part fix some $t \in \mathbb{R}_0^+$ and define $x = B(t)$ following $B^{-1}(x) = t$. Then there exists a $y \leq x$ with

$$t = B^{-1}(x) \leq A^{-1}(y) + U^{-1}(x-y).$$

Define now $s = A^{-1}(y)$ (following $A(s) \leq y$) and it holds

$$s = A^{-1}(y) = \sup\{s' \mid A(s') \leq y\} \leq \sup\{s' \mid B(s') \leq x\} = t.$$

Eventually,

$$\begin{aligned} t = B^{-1}(x) &\leq s + \sup\{s' \mid U(s') \leq x - y\} = \sup\{s'' \mid U(s''-s) + y \leq B(t)\} \\ &\leq \sup\{s'' \mid U(s''-s) + A(s) \leq B(t)\} \end{aligned}$$

from which follows $U(t-s) + A(s) \leq B(t)$. □

The next result shows that the virtual delay $\mathfrak{d}(t)$ is just the vertical distance of the inverse functions A^{-1} and B^{-1}.

LEMMA 9.3. *Let A and B be the arrivals and departures of a system, respectively. If $\mathfrak{d}(t) < T$ holds for all $t \in \mathbb{R}_0^+$, then*

$$B^{-1}(x) - A^{-1}(x) < T \qquad \textit{for all } x \in \mathbb{R}_0^+.$$

PROOF. Choose some $x \in \mathbb{R}_0^+$ and assume the first inequality to hold. Define $t = A^{-1}(x)$ and it follows

$$T > \mathfrak{d}(t) = \inf\{s \mid A(t) < B(t+s)\} = -t + \inf\{s' \mid A(t) < B(s')\}.$$

If $A(t) = x$, the above continues to

$$T > -A^{-1}(x) + B^{-1}(x)$$

If $A(t) < x$, define $y = \lim_{s \searrow t} A(s) \geq x$ instead. Similarly as before it holds

$$T > \lim_{s \searrow t} \mathfrak{d}(s) = \lim_{s \searrow t} -s + \inf\{r' \mid A(s) < B(r')\} = -A^{-1}(x) + B^{-1}(y)$$

and by the monotonicity of B^{-1} it follows $T > B^{-1}(x) - A^{-1}(x)$ again. $\square$

Eventually the delay-bound can be expressed as the vertical distance between the inverse of the arrival curve and the service curve.

THEOREM 9.4. *Let A and B be the arrivals and departures of a system, respectively. If $A \preceq \mathcal{A}$ and the system offers a service curve U, then*

$$\mathfrak{d}(t) \leq \sup_{x \geq 0} \{U^{-1}(x) - \mathcal{A}^{-1}(x)\}.$$

PROOF. By the previous lemma all left to show is that

$$U^{-1}(x) - A^{-1}(x) \leq \sup_{x \geq 0} \{U^{-1}(x) - \mathcal{A}^{-1}(x)\}$$

holds for all $x \in \mathbb{R}_0^+$. Fix some x, by Lemma 9.1 and 9.2 it holds

$$\begin{aligned} U^{-1}(x) - A^{-1}(x) &\leq \sup_{0 \leq y \leq x} \{A^{-1}(y) + U^{-1}(x-y)\} - A^{-1}(x) \\ &= \sup_{0 \leq y \leq x} \{A^{-1}(y) - A^{-1}(x) + U^{-1}(x-y)\} \\ &\leq \sup_{0 \leq y \leq x} \{U^{-1}(x-y) - \mathcal{A}^{-1}(x-y)\} \leq \sup_{0 \leq y} \{U^{-1}(y) - \mathcal{A}^{-1}(y)\}. \end{aligned}$$

$\square$

CHAPTER 10

Stochastic Network Calculus

10.1. Duality of Tail- and MGF-bounded Network Calculus

The two branches of tailbounded and MGF-bounded network calculus are related to each other. To some extend the arrival- and service-descriptions of one calculus can be transferred into the other and vice versa. This section describes how to do this. The main idea is drawn from [**106**]. There it is shown how to obtain MGF-bouds from tailbounds and vice versa for arrivals. Related to this is also the work of Wu et al. concerning the different methods of tailbounds [**157**].

LEMMA 10.1. *If a flow A is MGF-bounded by function f, it is also tailbounded for appropriate error η and envelope $\mathcal{A}$. If $A \preceq (\mathcal{A}, \eta)$ with error $\eta(t, \varepsilon) = \varepsilon$ and envelope $\mathcal{A}$ fulfilling $\int_0^1 e^{\theta \mathcal{A}(t-s,\varepsilon)} \mathrm{d}\varepsilon < \infty$, then it is also MGF-bounded by appropriate f.*

PROOF. Fix some $(s, t) \in \Lambda$. The proof follows the steps in [**106**]. Assume that A is MGF-bounded by f and Choose some envelope $\mathcal{A}$. For every $\theta > 0$ it holds then

$$\begin{aligned}\mathbb{P}(A(s,t) > \mathcal{A}(t-s,\varepsilon)) &\leq \phi_{A(s,t)}(\theta) e^{-\theta \mathcal{A}(t-s,\varepsilon)} \\ &\leq f(t-s,\theta) e^{-\theta \mathcal{A}(t-s,\varepsilon)} =: \eta(t-s,\varepsilon).\end{aligned}$$

For the second part of the lemma assume without loss of generality that the cumulative distribution function $F_{s,t}(x) := \mathbb{P}(A(s,t) \leq x)$ is continuous and strictly increasing (if it does not fulfill these assumptions a fitting approximation can be found). Denote further by $G_{s,t}(\varepsilon)$ the inverse function of

$$1 - F_{s,t}(x) = \mathbb{P}(A(s,t) > x).$$

It holds then

$$\mathbb{P}(A(s,t) > G_{s,t}(\varepsilon)) = \varepsilon \geq \mathbb{P}(A(s,t) > \mathcal{A}(t-s,\varepsilon)),$$

as $A \preceq (\mathcal{A}, \eta)$. From this follows that $G_{s,t}(\varepsilon) \leq \mathcal{A}(t-s,\varepsilon)$. Using the substitution $x = G_{s,t}(\varepsilon)$ it follows

$$\begin{aligned}\phi_{A(s,t)}(\theta) = \int_0^\infty e^{\theta A(s,t)} \mathrm{d}F_{s,t}(x) &= \int_0^1 e^{\theta G_{s,t}(\varepsilon)} \mathrm{d}\varepsilon \\ \leq \int_0^1 e^{\theta \mathcal{A}(t-s,\varepsilon)} \mathrm{d}\varepsilon &=: f(t-s,\theta)\end{aligned}$$

for all $\theta > 0$. □

A corresponding result for service descriptions requires the notion of exact and universal service elements.

DEFINITION 10.2. A dynamic U-server is *exact* if for any input-output pair A and B holds $B(t) = A \otimes U(0,t)$. A tailbounded server is called *universal*, if it fulfills Definition 2.11 independent of the choice of A.

REMARK 10.3. The convolution of exact service elements is exact. The convolution of universal service elements is universal. The leftover calculation $\ominus$ as in Theorem 2.37 does not preserve exactness.

THEOREM 10.4. *If a dynamic U-server is MGF-bounded by a function f, then it is also tailbounded for appropriate error ζ and envelope $\mathcal{U}$. If an exact and universal dynamic U-server is tailbounded with an error $\zeta(t,\varepsilon) = \varepsilon$ and some envelope $\mathcal{U}$ fulfilling $\int_0^1 e^{-\theta\mathcal{U}(t-s,\varepsilon)}\mathrm{d}\varepsilon < \infty$, then it is also MGF-bounded for appropriate f.*

PROOF. Assume first that $\mathbb{K} = \mathbb{N}_0$ and fix some $t \in \mathbb{N}_0$. Choose some envelope $\mathcal{U}$ increasing in its time-variable. and consider the case, that

$$U(s,t) \geq \mathcal{U}(t-s,\varepsilon) \qquad \text{for all } s \leq t.$$

Then would follow

$$\begin{aligned} B(t) \geq A \otimes U(0,t) &= \min_{0\leq s\leq t}\{A(s) + U(s,t)\} \\ &\geq \min_{0\leq s\leq t}\{A(s) + \mathcal{U}(t-s,\varepsilon)\} = A \otimes \mathcal{U}(\varepsilon)(0,t). \end{aligned}$$

Choose some $\theta > 0$ such that $\phi_{U(s,t)}(-\theta) \leq f(t-s,\theta)$. Translating this implication into probabilities results in

$$\begin{aligned} \mathbb{P}(B(t) < A \otimes \mathcal{U}(\varepsilon)(0,t)) &\leq \mathbb{P}\Big(\bigcup_{s=0}^{t} U(s,t) < \mathcal{U}(t-s,\varepsilon)\Big) \\ &\leq \sum_{s=0}^{t} \mathbb{P}(-U(s,t) > -\mathcal{U}(t-s,\varepsilon)) \\ &\leq \sum_{s=0}^{t} \phi_{U(s,t)}(-\theta)e^{-\theta\mathcal{U}(t-s,\varepsilon)} \\ &\leq \sum_{s=0}^{t} f(t-s,\theta)e^{-\theta\mathcal{U}(t-s,\varepsilon)} =: \zeta(t-s,\varepsilon). \end{aligned}$$

For the case $\mathbb{K} = \mathbb{R}_0^+$ Boole's inequality preceeds a discretization step.

For the second part of the theorem fix some $(s,t) \in \Lambda$ and define

$$A_s^\infty(t) = \begin{cases} 0 & \text{if } t \leq s \\ \infty & \text{if } t > s \end{cases}.$$

Denote by B the output of the dynamic U-server when fed with A_s^∞. Then it holds $B(t) = A_s^\infty \otimes U(0,t) = U(s,t)$, as the dynamic U-server is exact. Further, fix some $\varepsilon > 0$, then holds: $A_s^\infty \otimes \mathcal{U}(\varepsilon)(t) = \mathcal{U}(t-s,\varepsilon)$. Assume now for a while $B(t) \geq A_s^\infty \otimes \mathcal{U}(t-s,\varepsilon)$, then follows $U(s,t) \geq \mathcal{U}(t-s,\varepsilon)$; or written as probabilities

$$\mathbb{P}(U(s,t) < \mathcal{U}(t-s,\varepsilon)) \leq \mathbb{P}(B(t) < A_s^\infty \otimes \mathcal{U}(\varepsilon)(t)) \leq \zeta(t,\varepsilon).$$

Assume without loss of generality there exists a continuous and strictly increasing cumulative distribution function $F_{s,t}(x) = \mathbb{P}(U(s,t) < x)$. Denote its inverse by

$G_{s,t}$. Then

$$\begin{aligned}\zeta(t,\varepsilon) &= \mathbb{P}\big(U(s,t) < G_{s,t}(\zeta(t,\varepsilon))\big)\\ &\geq \mathbb{P}(U(s,t) < \mathcal{U}(t-s,\varepsilon)),\end{aligned}$$

from which follows $G_{s,t}(\zeta(t,\varepsilon)) \geq \mathcal{U}(t-s,\varepsilon)$.

Using the substitution $x = G_{s,t}(\varepsilon)$ it follows

$$\begin{aligned}\phi_{U(s,t)}(-\theta) &= \int_0^\infty e^{-\theta x}\mathrm{d}F_{s,t}(x) = \int_0^1 e^{-\theta G_{s,t}(\zeta(t,\varepsilon))}\mathrm{d}\varepsilon \leq \int_0^1 e^{-\theta\mathcal{U}(t,\varepsilon)}\mathrm{d}\varepsilon\\ &=: f(t-s,\theta)\end{aligned}$$

for all $\theta > 0$. □

10.2. MGF-bound for Markov-Modulated Arrivals

Let $\mathbb{K} = \mathbb{N}_0$. The next lemma is very similar to Lemma 7.2.7. in [**35**]. However, the proof given here is constructive. A corollary is needed first.

COROLLARY 10.5. *Assume a finite signal space S. Let $A = [a_{ij}] \in Mat(S)$ be a nonnegative matrix and $A^t = [a_{ij}^{(t)}]$ be its t-th power. If A has a positive eigenvector $\bar{x} = [x_i]$, then for all $t \in \mathbb{N}$ and all $i \in S$ the inequality*

$$\sum_{j\in S} a_{ij}^{(t)} \leq \frac{\max_{k\in S}\bar{x}_k}{\min_{k\in S}\bar{x}_k}\cdot \mathrm{sp}(A)^t$$

holds, where $\mathrm{sp}(\cdot)$ *is the spectral radius of some matrix in* $\mathrm{Mat}(S)$.

PROOF. See (8.1.33) in [**80**]. □

Assume now a Markov-modulated arrival A with finite signal space S, i.e., the distribution of the increments of A depend on the current state of an underlying Markov chain Y. The Markov chain is described by S and transition matrix $T = [t_{ij}]$ such that $t_{ij} > 0$ for all $i, j \in S$. Denote by $E \in \mathrm{Diag}(S)$ the matrix with entries $E_i := E_{ii} := \mathbb{E}(e^{\theta a(t)}|Y_t = i)$ for all states $i \in S$.

LEMMA 10.6. *For the above holds $A \preceq (\sigma,\rho)$ with*

$$\begin{aligned}\sigma(\theta) &= \frac{1}{\theta}\log\left(\max_{i\in S} E_i\cdot\frac{\max_{i\in S}\bar{x}_i}{\min_{i\in S}\bar{x}_i}\cdot\frac{1}{\mathrm{sp}(E\cdot T)}\right),\\ \rho(\theta) &= \frac{1}{\theta}\log\left(\mathrm{sp}(E\cdot T)\right),\end{aligned}$$

where x is a positive eigenvector of ET.

PROOF. Fix $\theta > 0$, for every $i \in S$ holds the backward equation

$$\begin{aligned}\mathbb{E}(e^{\theta A(t)}|\,Y_1 = i) =: E_i(t) &= \sum_{j\in S}\mathbb{E}(e^{\theta A(t)}|\,Y_1 = i,\,Y_2 = j)\mathbb{P}(Y_2 = j|\,Y_1 = i)\\ &= \sum_{j\in S}\mathbb{E}(e^{\theta a(1)}|\,Y_1 = i,\,Y_2 = j)\mathbb{E}(e^{\theta A(t)-a(1)}|\,Y_1 = i,\,Y_2 = j)t_{ij}\\ &= E_i\sum_{j\in S}\mathbb{E}(e^{\theta A(t-1)}|\,Y_1 = j)t_{ij} = E_i\sum_{j\in S}E_j(t-1)t_{ij}.\end{aligned}$$

Hence, the vector $E(t)$ with entries $E_i(t)$ has the form $E(t) = ET\cdot E(t-1)$. Applying this recursion results in $E(t) = (ET)^{t-1}E\cdot\mathbf{1}$, where $\mathbf{1}$ is the unit column vector on S. Assume now the beginning state of the chain is not given but follows a distribution $(\pi_i)_{i\in S}$. Then an application of the law of total probability yields

$$\begin{aligned}\mathbb{E}_\pi(e^{\theta A(t)}) &= \sum_{j\in S}\mathbb{P}(Y_1 = j)E_j(t) = \sum_{j\in S}\pi_j E_j(t) = \sum_{j\in S}\pi_j\left((ET)^{t-1}E\cdot\mathbf{1}\right)_j \\ &= \pi\cdot(ET)^{t-1}E\cdot\mathbf{1}.\end{aligned}$$

Since T is positive and E has positive entries on the diagonal, the matrix ET is also positive (i.e., every entry is larger 0). This allows to apply the Perron-Frobenius theorem [**125, 69**], which guarantees an eigenvector with positive entries. Denote this eigenvector by $\bar{x}\in\mathbb{R}^S$; it fulfills the conditions of Corollary 10.5. Eventually holds

$$\begin{aligned}E_\pi(t) &= \sum_{i\in S}\pi_i\sum_{j\in S}(ET)^{t-1}_{ij}(E\cdot\mathbf{1})_j \le \sum_{i\in S}\pi_i\Big(\max_{k\in S}E_k\Big)\sum_{j\in S}(ET)^{t-1}_{ij} \\ &\le \Big(\max_{k\in S}E_k\Big)\sum_{i\in S}\pi_i\frac{\max_{k\in S}\bar{x}_k}{\min_{k\in S}\bar{x}_k}\mathrm{sp}(ET)^{t-1} = \Big(\max_{k\in S}E_k\Big)\frac{\max_{k\in S}\bar{x}_k}{\min_{k\in S}\bar{x}_k}\mathrm{sp}(ET)^{t-1}\end{aligned}$$

for every starting distribution π and all $t\in\mathbb{N}$. This allows to finish the proof via $\phi_{A(s,t)}(\theta) = \mathbb{E}(e^{\theta(A(t)-A(s))}\,|\,Y_s) = E_{Y_s}(t-s)$. $\square$

Bibliography

[1] L. De Haan A. A. Balkema. A convergence rate in extreme-value theory. *Journal of Applied Probability*, 27(3):577–585, September 1990.

[2] R. Agrawal, F. Baccelli, and R. Rajan. An algebra for queueing networks with time-varying service and its application to the analysis of integrated service networks. *Mathematics of Operations Research*, 29(3):559–591, 2004.

[3] R. Agrawal, R. L. Cruz, C. Okino, and R. Rajan. Performance bounds for flow control protocols. *IEEE/ACM Transactions on Networking*, 7(3):310–323, June 1999.

[4] R. Agrawal and R. Rajan. Performance bounds for guaranteed and adaptive services. Technical Report RC20649, IBM, May 1996.

[5] D. Anick, D. Mitra, and M. M. Sondhi. Stochastic theory of a data-handling system with multiple sources. *The Bell System Technical Journal*, 61(8):1871–1894, October 1982.

[6] S. Ayyorgun and R. L. Cruz. A service-curve model with loss and a multiplexing problem. In *Proc. IEEE ICDCS*, pages 756–765, March 2004.

[7] F. Baccelli, G. Cohen, G. J. Olsder, and J.-P. Quadrat. *Synchronization and Linearity: An Algebra for Discrete Event Systems*. Wiley, 1992.

[8] F. Baccelli and D. Hong. Tcp is max-plus linear and what it tells us on its throughput. In *Proc. ACM SIGCOMM*, pages 219–230, October 2000.

[9] F. Baskett, K. M. Chandy, R. R. Muntz, and F. G. Palacios. Open, closed, and mixed networks of queues with different classes of customers. *Journal of the ACM*, 22(2):248–260, April 1975.

[10] M. Beck. *DiscoSNC - A tool for stochastic network calculus*. http://disco.informatik.uni-kl.de/index.php/projects/disco-snc, 2013.

[11] M. Beck. *Funktion und Zeremoniell im mariatheresianischen Schlossbau (1740-80): Die Wiener Hofburg, Schloss Schönbrunn, Schloss Laxenburg, Schloss Holtisch und Schloss Hof.* PhD thesis, Trier University, 2014.

[12] M. Beck, S. A. Henningsen, S. B. Birnbach, and J. B. Schmitt. Towards a statistical network calculus – dealing with uncertainty in arrivals. In *Proc. IEEE INFOCOM*, pages 2382–2390, April 2014.

[13] M. Beck and J. Schmitt. On the calculation of sample-path backlog bounds in queueing systems over finite time horizons. In *Proc. Valuetools*, pages 148–157, October 2012.

[14] M. Beck and J. Schmitt. The disco stochastic network calculator version 1.0: When waiting comes to an end. In *Proc. Valuetools*, pages 282–285, 2013.

[15] M. Beck and J. Schmitt. Window flow control in stochastic network calculus. Technical Report 391/15, University of Kaiserslautern, September 2015.

[16] M. Beck and J. Schmitt. Window flow control in stochastic network calculus – the general service case. In *Proc. Valuetools*, January 2016.

[17] N. Becker and M. Fidler. A non-stationary service curve model for performance analysis of transient phases. In *Proc. ITC-27*, pages 116–124, September 2015.

[18] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. An architecture for differentiated service. RFC 2475, December 1998.

[19] S. Bondorf and J. Schmitt. Improving cross-traffic bounds in feed-forward networks – there is a job for everyone. In *Proc. GI/ITG MMB*, pages 9–24, April 2016.

[20] S. Bondorf and J. B. Schmitt. The discodnc v2: A comprehensive tool for deterministic network calculus. In *Proc. Valuetools*, pages 44–49, 2014.

[21] R.-R. Boorstyn, A. Burchard, J. Liebeherr, and C. Oottamakorn. Statistical service assurances for traffic scheduling algorithms. *IEEE Journal on Selected Areas in Communications*, 18(12):2651–2664, December 2000.

[22] A. Bose, X. Jiang, B. Liu, and G. Li. Analysis of manufacturing blocking systems with network calculus. *Performance Evaluation*, 63(12):1216 – 1234, December 2006.

[23] J. Y. Le Boudec and P. Thiran. A note on time and space methods in network calculus. In *International Zurich Seminar on Broadband Communications*, pages 267–272, February 1998.

[24] A. Bouillard, L. Jouhet, and E. Thierry. Service curves in network calculus: dos and don'ts. Research Report RR-7094, INRIA, 2009.

[25] A. Bouillard, L. Jouhet, and E. Thierry. Tight performance bounds in the worst-case analysis of feed-forward networks. In *Proc. IEEE INFOCOM*, pages 1–9, March 2010.

[26] A. Bouillard and É. Thierry. An algorithmic toolbox for network calculus. *Discrete Event Dynamic Systems*, 18(1):3–49, March 2008.

[27] R. Braden, D. Clark, and S. Shenker. Integrated services in the internet architecture: an overview. RFC 1633, June 1994.

[28] L. S. Brakmo, S. W. O'Malley, and L. L. Peterson. Tcp vegas: New techniques for congestion detection and avoidance. *Proc. ACM SIGCOMM*, 24(4):24–35, October 1994.

[29] M. Bredel, Z. Bozakov, and Y. Jiang. Analyzing router performance using network calculus with external measurements. In *Proc. IWQoS*, pages 1–9, June 2010.

[30] A. Burchard, J. Liebeherr, and F. Ciucu. On theta(h log h) scaling of network delays. In *Proc. IEEE INFOCOM*, pages 1866–1874, May 2007.

[31] A. Burchard, J. Liebeherr, and S.D. Patek. A min-plus calculus for end-to-end statistical service guarantees. *IEEE Transactions on Information Theory*, 52(9):4105–4114, 2006.

[32] C.-S. Chang. Stability, queue length and delay. ii. stochastic queueing networks. In *Proc. IEEE CDC*, pages 1005–1010 vol.1, 1992.

[33] C.-S. Chang. Stability, queue length and delay of deterministic and stochastic queueing networks. *IEEE Transactions on Automatic Control*, 39(5):913–931, May 1994.

[34] C.-S. Chang. On deterministic traffic regulation and service guarantees: A systematic approach by filtering. *IEEE Transactions on Information Theory*, 44(3):1097–1110, May 1998.

[35] C.-S. Chang. *Performance Guarantees in Communication Networks*. Telecommunication Networks and Computer Systems. Springer-Verlag, 2000.

[36] C.-S. Chang, R. L. Cruz, J.-Y. Le Boudec, and P. Thiran. A min, + system theory for constrained traffic regulation and dynamic service guarantees. *IEEE/ACM Transactions on Networking*, 10(6):805–817, December 2002.

[37] A. Charny and J.-Y. Le Boudec. Delay bounds in a network with aggregate scheduling. In *Proc. QofIS*, pages 1–13, September 2000.

[38] A. Chen, Z. Zhou, P. Chootinan, S. Ryu, C. Yang, and S.C. Wong. Transport network design problem under uncertainty: A review and new developments. *Transport Reviews*, 31(6):743–768, 2011.

[39] F. Ciucu, A. Burchard, and J. Liebeherr. A network service curve approach for the stochastic analysis of networks. In *Proc. ACM SIGMETRICS*, pages 279–290, June 2005.

[40] F. Ciucu, A. Burchard, and J. Liebeherr. Scaling properties of statistical end-to-end bounds in the network calculus. *IEEE Transactions on Information Theory*, 52(6):2300–2312, June 2006.

[41] F. Ciucu, M. Fidler, J. Liebeherr, and J. Schmitt. Network calculus. *Dagstuhl Reports*, pages 63–83, March 2015.

[42] F. Ciucu, F. Poloczek, and J. Schmitt. Sharp bounds in stochastic network calculus. In *Proc. ACM SIGMETRICS*, pages 367–368, June 2013.

[43] F. Ciucu and J. B. Schmitt. Perspectives on network calculus: no free lunch, but still good value. *Proc. of ACM SIGCOMM*, pages 311–322, August 2012.

[44] C. J. Clopper and E. S. Pearson. The use of confidence or fiducial limits illustrated in the case of the binomial. *Biometrika*, 26(4):404–413, 1934.

[45] J. P. Cohen. Convergence rates for the ultimate and pentultimate approximations in extreme-value theory. *Advances in Applied Probability*, 14(4):833–854, 1982.

[46] R. Cruz. Guaranteed throughput for window flow control in a multi-tasking environment. In *Proc. IEEE CISS*, March 1991.

[47] R. L. Cruz. A calculus for network delay, Part I: Network elements in isolation. *IEEE Transactions on Information Theory*, 37(1):114–131, January 1991.

[48] R. L. Cruz. A calculus for network delay, Part II: Network analysis. *IEEE Transactions on Information Theory*, 37(1):132–141, January 1991.

[49] R. L. Cruz. Quality of service guarantees in virtual circuit switched networks. *IEEE Journal on Selected Areas in Communications*, 13(6):1048–1056, August 1995.

[50] R. L. Cruz. Quality of service management in integrated services networks. In *Proc. Semi-Annual Research Review, Center Wireless Communication, UCSD*, 1996.

[51] R. L. Cruz. SCED+: Efficient management of quality of service guarantees. In *Proc. IEEE INFOCOM*, volume 2, pages 625–634, March 1998.

[52] R. L. Cruz and C. M. Okino. Service guarantees for window flow control. In *Proc. Allerton Conference on Communication, Control, and Computing*, October 1996.

[53] R. A. Davis. The rate of convergence in distribution of the maxima. *Statistica Neerlandica*, 36(1):31–35, 1982.

[54] L. De Haan. *On Regular Variations and Its Application to the Weak Convergence of Sample Extremes*. PhD thesis, Math. Centre Tract 32. Math. Centrum Amsterdam, 1970.

[55] L. De Haan and A. Ferreira. *Extreme Value Theory: An Introduction*. Springer, 2006.

[56] F. Dong, K. Wu, and V. Srinivasan. Copula analysis for statistical network calculus. In *Proc. IEEE INFOCOM*, pages 1535–1543, April 2015.

[57] P. Embrechts, C. Klüppelberg, and T. Mikosch. *Modelling Extremal Events for Insurance and Finance*. Springer, 1997.

[58] A.K. Erlang. The theory of probabilities and telephone conversations. *Nyt Tidsskrift for Matematik*, 20(B):33–39, 1909.

[59] A.K. Erlang. Solution of some problems in the theory of probabilities of significance in automatic telephone exchanges. *Elektrotkeknikeren*, 13:138–155, 1917.

[60] X. Fang, S. Misra, G. Xue, and D. Yang. Smart grid – the new and improved power grid: A survey. *IEEE Communications Surveys Tutorials*, 14(4):944–980, April 2012.

[61] R.Z. Farahani, E. Miandoabchi, W.Y. Szeto, and H. Rashidi. A review of urban transportation network design problems. *European Journal of Operational Research*, 229(2):281 – 302, 2013.

[62] A. Feldmann, A. C. Gilbert, P. Huang, and W. Willinger. Dynamics of ip traffic: A study of the role of variability and the impact of control. *SIGCOMM Comput. Commun. Rev.*, 29(4):301–313, August 1999.

[63] W. Feller. *An introduction to probability theory and its applications*, volume 2. Wiley, 2008.

[64] M. Fidler. Extending the network calculus pay bursts only once principle to aggregate scheduling. In *Proc. QoS-IP*, pages 19–34, February 2003.

[65] M. Fidler. An end-to-end probabilistic network calculus with moment generating functions. In *Proc. of IEEE IWQoS*, pages 261–270, June 2006.

[66] M. Fidler. Survey of deterministic and stochastic service curve models in the network calculus. *IEEE Communications Surveys Tutorials*, 12(1):59–86, 2010.

[67] M. Fidler and S. Recker. Conjugate network calculus: A dual approach applying the legendre transform. *Proc. QoS-IP*, 50(8):1026 – 1039, 2006.

[68] M. Fidler and A. Rizk. A guide to the stochastic network calculus. *IEEE Communications Surveys Tutorials*, 17(1):92–105, 2015.

[69] G. F. Frobenius. *Über Matrizen aus nicht negativen Elementen*, volume 1. Sitzungsber. Königl. Preuss. Akad. Wiss., 1912.

[70] J. R. Gallardo, D. Makrakis, and L. Orozco-Barbosa. Use of α-stable self-similar stochastic processes for modeling traffic in broadband networks. *Performance Evaluation*, 40(1-3):71 – 98, 2000.

[71] L. Georgiadis, R. Guerin, and A. Parekh. Optimal multiplexing on a single link: delay and buffer requirements. *IEEE Transactions on Information Theory*, 43(5):1518–1535, September 1997.

[72] N. Geroliminis and A. Skabardonis. Identification and analysis of queue spillovers in city street networks. *IEEE Transactions on Intelligent Transportation Systems*, 12(4):1107–1115, December 2011.

[73] Y. Ghiassi-Farrokhfal, S. Keshav, and C. Rosenberg. Toward a realistic performance analysis of storage systems in smart grids. *IEEE Transactions on Smart Grid*, 6(1):402–410, January 2015.

[74] R. Gibbens and F. Kelly. Measurement-based connection admission control. In *Proc. ITC 15*, pages 781–790, June 1997.

[75] M. K. Govil and M. C. Fu. Queueing theory in manufacturing: A survey. *Journal of Manufacturing Systems*, 18(3):214 – 240, 1999.

[76] M. Grossglauser and D.N.C. Tse. A framework for robust measurement-based admission control. *IEEE/ACM Transactions on Networking*, 7:293–309, June 1999.

[77] S. Ha, I. Rhee, and L. Xu. Cubic: A new tcp-friendly high-speed tcp variant. *ACM SIGOPS Operating Systems Review*, 42(5):64–74, July 2008.

[78] R. Henia, A. Hamann, M. Jersak, R. Racu, K. Richter, and R. Ernst. System level performance analysis - the symta/s approach. *IEE Proceedings - Computers and Digital Techniques*, 152(2):148–166, March 2005.

[79] R. Hooke and T. A. Jeeves. "direct search" solution of numerical and statistical problems. *J. ACM*, 8(2):212–229, April 1961.

[80] R.A. Horn and C.R. Johnson. *Matrix Analysis*. Cambridge University Press, 1990.

[81] J. R. Jackson. Scheduling a production line to minimize maximum tardiness. Research Report 43, Management Science Research Project, University of California, 1955.

[82] J. R. Jackson. Networks of waiting lines. *Operations Research*, 5(4):518–521, 1957.

[83] V. Jacobson. Congestion avoidance and control. *SIGCOMM Comput. Commun. Rev.*, 18(4):314–329, August 1988.

[84] V. Jacobson. Modified TCP congestion avoidance algorithm. end2end-interest mailing list, April 1990.

[85] S. Jamin, P.B. Danzig, S. J. Shenker, and L. Zhang. A measurement-based admission control algorithm for integrated services packet networks. *IEEE/ACM Transactions on Networking*, 5:56–70, February 1997.

[86] P. R. Jelenkovic. Long-tailed loss rates in a single server queue. In *Proc. IEEE INFOCOM*, pages 1462–1469, March 1998.

[87] Y. Jiang. Delay bounds for a network of guaranteed rate servers with fifo aggregation. *Computer Networks*, 40(6):683–694, 2002.

[88] Y. Jiang and Y. Liu. *Stochastic Network Calculus*. Springer, 2008.

[89] C. Joe-Wong, S. Sen, S. Ha, and M. Chiang. Optimized day-ahead pricing for smart grids with device-specific scheduling flexibility. *IEEE Journal on Selected Areas in Communications*, 30(6):1075–1085, July 2012.

[90] M. A. Kafi, D. Djenouri, J. Ben-Othman, and N. Badache. Congestion control protocols in wireless sensor networks: A survey. *IEEE Communications Surveys Tutorials*, 16(3):1369–1390, March 2014.

[91] F. P. Kelly. Networks of queues with customers of different types. *Journal of Applied Probability*, 12(3):542–554, 1975.

[92] F. P. Kelly. Notes on effective bandwidths. In *Stochastic Networks: Theory and Applications*, number 4 in Royal Statistical Society Lecture Notes, pages 141–168. Oxford University Press, 1996.

[93] G. Kesidis and T. Konstantopoulos. Worst-case performance of a buffer with independent shaped arrival processes. *IEEE Communication Letters*, 4(1):26–28, January 2000.

[94] L. Kleinrock. *Communication Nets; Stochastic Message Flow and Delay*. Dover Publications, Incorporated, 1972.

[95] H. Kuai, F. Alajaji, and G. Takahara. A lower bound on the probability of a finite union of events. *Discrete Mathematics*, 215(1-3):147 – 158, 2000.

[96] J. Kurose. On computing per-session performance bounds in high-speed multi-hop computer networks. In *Proc. of ACM SIGMETRICS*, pages 128–139, 1992.

[97] S. Lämmer and D. Helbing. Self-control of traffic lights and vehicle flows in urban road networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(04), 2008.

[98] R. C. Larson and A. R. Odoni. *Urban Operations Research*. Prentice-Hall, 1981.

[99] E. L. Lawler. *Mathematical Programming The State of the Art: Bonn 1982*, chapter Recent Results in the Theory of Machine Scheduling, pages 202–234. Springer, 1983.

[100] J.-Y. Le Boudec. Network calculus made easy. Technical Report EPFL-DI 96/218, December 1996.

[101] J.-Y. Le Boudec and P. Thiran. Network Calculus viewed as a Min-plus System Theory applied to Communication Networks. Technical report, 1998.

[102] J.-Y. Le Boudec and P. Thiran. *Network Calculus A Theory of Deterministic Queuing Systems for the Internet*. Number 2050 in Lecture Notes in Computer Science. Springer-Verlag, Berlin, Germany, 2001.

[103] J.-Y. Le Boudec and D.-C. Tomozei. A demand-response calculus with perfect batteries. *Proc. GI/ITG MMB*, pages 273–287, March 2012.

[104] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson. On the self-similar nature of ethernet traffic (extended version). *IEEE/ACM Transactions on Networking*, 2(1):1–15, February 1994.

[105] L. Lenzini, L. Martorini, E. Mingozzi, and G. Stea. Tight end-to-end per-flow delay bounds in fifo multiplexing sink-tree networks. *Perform. Eval.*, 63(9):956–987, October 2006.

[106] C. Li, A. Burchard, and J. Liebeherr. A network calculus with effective bandwidth. *IEEE/ACM Transactions on Networking*, 15(6):1442–1453, December 2007.

[107] J. Liebeherr, A. Burchard, and F. Ciucu. Delay bounds in communication networks with heavy-tailed and self-similar traffic. *IEEE Transactions on Information Theory*, 58:1010–1024, February 2012.

[108] J. Liebeherr, M. Fidler, and S. Valaee. A min-plus system interpretation of bandwidth estimation. In *Proc. IEEE INFOCOM*, pages 1127–1135, May 2007.

[109] N. Likhanov, R. R. Mazumdar, and O. Ozturk. Large buffer asymptotics for fluid queues with heterogeneous m/g/∞ weibullian inputs. *Queueing Systems*, 45(4):333–356, 2003.

[110] J. T. Lim, S. M. Meerkov, and F. Top. Homogeneous, asymptotically reliable serial production lines: theory and a case study. *IEEE Transactions on Automatic Control*, 35(5):524–534, May 1990.

[111] D. V. Lindley. The theory of queues with a single server. *Mathematical Proceedings of the Cambridge Philosophical Society*, 48:277–289, 4 1952.

[112] P. Loiseau, P. Gonçalves, G. Dewaele, P. Borgnat, P. Abry, and P. V.-B. Primet. Investigating self-similarity and heavy-tailed distributions on a large-scale experimental facility. *IEEE/ACM Transactions on Networking*, 18(4):1261–1274, August 2010.

[113] R. Lübben. *System identification of computer networks with random service*. PhD thesis, Hannover University, 2014.

[114] R. Lübben, M. Fidler, and J. Liebeherr. A foundation for stochastic bandwidth estimation of networks with random service. In *Proc. of IEEE INFOCOM*, pages 1817–1825, April 2011.

[115] R. Lübben, M. Fidler, and J. Liebeherr. Stochastic bandwidth estimation in networks with random service. *IEEE/ACM Transactions on Networking*, 22(2):484–497, April 2014.

[116] B. B. Mandelbrot and J. W. Van Ness. Fractional brownian motions, fractional noises and applications. *SIAM Review*, 10(4):422–437, 1968.

[117] S. Mao and S. S. Panwar. A survey of envelope processes and their applications in quality of service provisioning. *IEEE Communications Surveys Tutorials*, 8(3):2–20, 2006.

[118] P. Massart. The tight constant in the dvoretzky-kiefer-wolfowitz inequality. *Annals of Probability*, 18(3):1269–1283, 1990.

[119] I. Mitrani. *Probabilistic Modelling*. Cambridge University Press, 1998.

[120] K. Nichols, V Jacobson, and L. Zhang. A two-bit differentiated services architecture for the internet. RFC 2638, July 1999.

[121] I. Norros. A storage model with self-similar input. *Queueing Systems*, 16(3):387–396, 1994.

[122] I. Norros. On the use of fractional brownian motion in the theory of connectionless networks. *IEEE Journal on Selected Areas in Communications*, 13(6):953–962, August 1995.

[123] H.T. Papadopoulos and C. Heavey. Queueing theory in manufacturing systems analysis and design: A classification of models for production and transfer lines. *European Journal of Operational Research*, 92(1):1 – 27, 1996.

[124] V. Paxson and S. Floyd. Wide area traffic: The failure of poisson modeling. *IEEE/ACM Transactions on Networking*, 3(3), June 1995.

[125] O. Perron. Zur theorie der matrices. *Mathematische Annalen*, 64:248–263, 1907.

[126] V. V. Petrov. *Sums of Independent Random Variables*. Springer, 1975.

[127] F. Poloczek and F. Ciucu. A martingale-envelope and applications. *SIGMETRICS Perform. Eval. Rev.*, 41(3):43–45, January 2014.

[128] F. Poloczek and F. Ciucu. Scheduling analysis with martingales. *Performance Evaluation*, 79:56 – 72, 2014.

[129] F. Poloczek and F. Ciucu. Service-martingales: Theory and applications to the delay analysis of random access protocols. In *Proc. IEEE INFOCOM*, pages 945–953, April 2015.

[130] J. Qiu and E. W. Knightly. Measurement-based admission control with aggregate traffic envelopes. *IEEE/ACM Transactions on Networking*, 9(2):199–210, April 2001.

[131] S. I. Resnick. *Extreme Values, Regular Variation and Point Processes*. Springer, 1987.

[132] A. Rizk and M. Fidler. Non-asymptotic end-to-end performance bounds for networks with long range dependent fbm cross traffic. *Computer Networks*, 56(1):127–141, January 2012.

[133] M. Roozbehani, M. Dahleh, and S. Mitter. Dynamic pricing and stabilization of supply and demand in modern electric power grids. In *Proc. IEEE SmartGridComm*, pages 543–548, Oct 2010.

[134] H. Sariowan, R. L. Cruz, and G. C. Polyzos. Sced: A generalized scheduling policy for guaranteeing quality-of-service. *IEEE/ACM Transactions on Networking*, 7(5):669–684, October 1999.

[135] H. Schiøler, J. J. Jessen, J. D. Nielsen, and K. G. Larsen. Network calculus for real time analysis of embedded systems with cyclic task dependencies. In *Computers and Their Applications*, pages 326–332, 2005.

[136] H. Schiøler, H. P. Schwefel, and M. B. Hansen. Cync: A matlab/simulink toolbox for network calculus. In *Proc. Valuetools*, pages 60:1–60:10, 2007.

[137] J. B. Schmitt, F. A. Zdarsky, and M. Fidler. Delay bounds under arbitrary multiplexing: When network calculus leaves you in the lurch... In *Proc. IEEE INFOCOM*, pages 1669–1677, April 2008.

[138] J. B. Schmitt, F. A. Zdarsky, and I. Martinovic. Improving performance bounds in feed-forward networks by paying multiplexing only once. In *Proc. GI/ITG MMB*, pages 1–15, March 2008.

[139] A. Shekaramiz, J. Liebeherr, and A. Burchard. Window flow control systems with random service. abs/1507.04631, 2015.

[140] S. Shenker. Specification of guaranteed quality of service. RFC 2212, RFC Editor, September 1997.

[141] A. Shwartz and A. Weiss. *Large Deviations for Performance Analysis: Queues, Communications and Computing.* Chapman and Hall, 1995.

[142] R. L. Smith. Uniform rates of convergence in extreme-value theory. *Advances in Applied Probability*, 14(3):600–622, 1982.

[143] D. Starobinski, M. Karpovsky, and L. A. Zakrevski. Application of network calculus to general topologies using turn-prohibition. *IEEE/ACM Transactions on Networking*, 11(3):411–421, 2003.

[144] D. Starobinski and M. Sidi. Stochastically bounded burstiness for communication networks. In *Proc. IEEE INFOCOM*, volume 1, pages 36–42 vol.1, Mar 1999.

[145] W. Stevens. Tcp slow start, congestion avoidance, fast retransmit, and fast recovery algorithms. RFC 2001, 1997.

[146] W. R. Stevens and G. R. Wright. *TCP/IP Illustrated (Vol. 2): The Implementation.* Addison-Wesley Longman, 1995.

[147] D. Stiliadis and A. Varma. Latency-rate servers: A general model for analysis of traffic scheduling algorithms. *IEEE/ACM Transactions on Networking*, 6(5):611–624, October 1998.

[148] K. S. Trivedi. *Probability and Statistics with Reliability, Queuing and Computer Science Applications.* Prentice Hall PTR, 1982.

[149] A. Undheim, Y. Jiang, and P. J. Emstad. Network calculus approach to router modeling with external measurements. In *Proc. CHINACOM*, pages 276–280, August 2007.

[150] S. Valaee. The duality of maximum delay and maximum backlog. Technical report, 2002.

[151] J. Walrand. *An Introduction to Queueing Networks.* Prentice Hall, 1988.

[152] E. Wandeler. *Modular performance analysis and interface-based design for embedded real-time systems.* PhD thesis, ETH Zurich, 2006.

[153] K. Wang, F. Ciucu, C. Lin, and S. Low. A stochastic power network calculus for integrating renewable energy sources into the power grid. *IEEE Journal on Selected Areas in Communications: Smart Grid Communications Series*, 30:1037–1048, July 2012.

[154] K. Wang, S. Low, and C. Lin. How stochastic network calculus concepts help green the power grid. In *Proc. IEEE SmartGridComm*, pages 55–60, October 2011.

[155] T. Wongpiromsarn, T. Uthaicharoenpong, Y. Wang, E. Frazzoli, and D. Wang. Distributed traffic signal control for maximum network throughput. In *Proc. IEEE Conference on Intelligent Transportation Systems*, pages 588–595, September 2012.

[156] G. R. Wright and W. R. Stevens. *TCP/IP Illustrated*, volume 2. Addison-Wesley Professional, 1995.

[157] K. Wu, Y. Jiang, and J. Li. On the model transform in stochastic network calculus. In *Proc. IWQoS*, pages 1–9, June 2010.

[158] K. Wu, Y. Jiang, and D. Marinakis. A stochastic calculus for network systems with renewable energy sources. In *Proc. IEEE Computer Communications Workshops (INFOCOM)*, pages 109–114, March 2012.

[159] J. Yang, F. Alajaji, and G. Takahara. New bounds on the probability of a finite union of events. In *IEEE International Symposium on Information Theory*, pages 1271–1275, June 2014.

[160] J. Yang, F. Alajaji, and G. Takahara. On Bounding the Union Probability Using Partial Weighted Information. *ArXiv e-prints*, June 2015.

[161] O. Yaron and M. Sidi. Performance and stability of communication networks via robust exponential bounds. *IEEE/ACM Transactions on Networking*, 1(3):372–385, June 1993.

[162] Q. Yin, Y. Jiang, S. Jiang, and P. Y. Kong. Analysis of generalized stochastically bounded bursty traffic for communication networks. In *Proc. IEEE LCN*, pages 141–149, November 2002.

Curriculum Vitae

Name	Michael Alexander Beck
Education	
since 2010	Research Assistant and Ph.D. Candidate – Computer Science Department University of Kaiserslautern
2005 – 2010	Mathematics Diploma – Probability Theory University of Kaiserslautern Thesis – Capacity-Bounds for Average Exit-times of Grids (in German)
Teaching Experience	
since 2010	Teaching Assistant Distributed Computer Systems Lab University of Kaiserslautern
2006 – 2010	Teaching Assistant Mathematics Department University of Kaiserslautern