DISSERTATION

DEALING WITH DEPENDENCE IN THE END-TO-END PERFORMANCE ANALYSIS IN STOCHASTIC NETWORK CALCULUS

Thesis approved by the Department of Computer Science of the
TU Kaiserslautern for the award of the Doctoral Degree

Doctor of Natural Sciences (Dr. rer. nat.)

to

Paul Nikolaus

DATE OF DEFENSE:	:	March 24, 2022
DEAN:	:	Prof. Dr. Jens B. Schmitt
PHD COMMITTEE CHAIR	:	Prof. Dr. Christoph Garth
REVIEWERS:	:	Prof. Dr. Jens B. Schmitt
		Prof. Dr. Florin Ciucu
		Prof. Dr. Amr Rizk

D 386

DEALING WITH DEPENDENCE IN THE END-TO-END PERFORMANCE ANALYSIS IN STOCHASTIC NETWORK CALCULUS

PAUL NIKOLAUS

TECHNISCHE UNIVERSITÄT KAISERSLAUTERN

Department of Computer Science
TU Kaiserslautern

ABSTRACT

Communication networks, in particular the Internet, have become a pivotal part of our life. Since their beginnings, a key aspect of their applicability has been the performance. Safety-critical applications, for example, can sometimes only be implemented in a responsible manner if guarantees about their end-to-end delay can be made. A mathematical modeling and performance evaluation of communication networks requires a powerful set of tools that is able to incorporate their increasing complexity.

The stochastic network calculus (SNC) is a versatile, mathematical framework that allows for a calculation of probabilistic end-to-end performance bounds of distributed systems. Its flexibility enables to incorporate a large class of different schedulers as well as different models of traffic processes beyond the assumption of Poisson arrivals that is predominant in queueing theory-based analyses. It originates in the so-called deterministic network analysis (DNC) in the 90's of the 20th century that was introduced to provide deterministic, "hard" guarantees that are of relevance, e.g., in the context of real-time systems. While the DNC of today can be used to calculate fast and accurate delay bounds of arbitrary feedforward networks, the SNC is still in a significantly earlier stage. In particular, *method-pertinent dependencies*, i.e., a phenomenon that occurs when independent flows become stochastically dependent after sharing resources in the network, can be considered a major challenge in the SNC with moment-generating functions (MGFs).

This thesis argues to contribute to the SNC in several ways. First, we show that the "pay multiplexing only once" (PMOO) analysis known from the DNC is also possible in the SNC. Not only does it significantly improve end-to-end delay bounds, it also needs to consider less method-pertinent dependencies. Therefore, complexity and runtimes of the calculation are greatly reduced. Second, we introduce the concept of negative dependence to the SNC with MGFs and give numerical evidence that this can further lead to better performance bounds. Third, for the larger problem of end-to-end performance bounds of tree networks, we introduce so-called "*h*-mitigators", a modification in the calculation of MGF output bounds. It is minimally invasive, all existing results and procedures are still applicable, and improves performance bounds. As a fourth contribution, we conduct extensive numerical evaluations to substantiate our claims. Moreover, we made the respective code, the "SNC MGF toolbox", publicly available to ensure that the results are reproducible. At last, we conduct different stochastic analyses of a popular fair scheduler, generalized processor sharing (GPS). We give an overview of the state-of-the-art analyses in the SNC and substantiate the comparison through numerical evaluations.

DECLARATION

Parts of this thesis have been previously published by the author in the following joint publications:

CONFERENCE PROCEEDINGS

[NS17] Paul Nikolaus and Jens Schmitt. "On Per-Flow Delay Bounds in Tandem Queues under (In)Dependent Arrivals." In: Proc. IFIP Net- working 2017 Conference (NETWORKING'17). Stockholm, Sweden, 2017.

[NS18] Paul Nikolaus and Jens Schmitt. "Improving Output Bounds in the Stochastic Network Calculus Using Lyapunov's Inequality." In: Proc. IFIP Networking 2018 Conference (NETWORKING'18). Zurich, Switzerland: IEEE, May 2018.

[NS20b] Paul Nikolaus and Jens Schmitt. "On the Stochastic End-to-End Delay Analysis in Sink Trees Under Independent and Dependent Arrivals." In: Proc. Conference on Measurement, Modelling and Evaluation of Computing Systems (MMB'20). Springer. 2020.

[NSC19b] Paul Nikolaus, Jens Schmitt, and Florin Ciucu. "Dealing with Dependence in Stochastic Network Calculus – Using Independence as a Bound." In: Proc. International Conference on Analytical and Stochastic Modeling Techniques and Applications (ASMTA'19). Springer. Moscow, Russia, 2019.

JOURNAL ARTICLE

[BNS22] Anne Bouillard, Paul Nikolaus, and Jens Schmitt. "Unleashing the Power of Paying Multiplexing Only Once in Stochastic Network Calculus." In: Proceedings of the ACM on Measurement and Analysis of Computing Systems (POMACS) 6.2 (2022), p. 34.

[NSS19] Paul Nikolaus, Jens Schmitt, and Malte Schütze. "h-Mitigators: Improving your stochastic network calculus output bounds." In: Computer Communications 144 (2019), pp. 188–197.

TECHNICAL REPORTS

[BNS21] Anne Bouillard, Paul Nikolaus, and Jens B. Schmitt. "Unleashing the Power of Paying Multiplexing Only Once in Stochastic Network Calculus." In: CoRR abs/2104.14215 (2021). arXiv: 2104.14215.

[NSC19a] Paul Nikolaus, Jens B. Schmitt, and Florin Ciucu. Dealing with Dependence in Stochastic Network Calculus – Using Independence as a Bound. Tech. rep. TU Kaiserslautern, Germany, May 2019.

The complete list of publications can be found on page 155.

ACKNOWLEDGMENTS

I would like to express my profound gratitude to my parents, Irina and Roman, for supporting me my whole life. Nothing would have been possible without you. I am thankful to my girlfriend, Yuki, for always being patient with me. You gave me the strength I needed to achieve this goal.

I would like to express my heartfelt gratitude to my supervisor, Prof. Jens Schmitt, for inviting me to the world of academia, for his support beyond measure, and for all the fruitful conversations we had. You really had a formative influence on me. I couldn't imagine a better mentor.

I would like to thank the PhD committee for their time and consideration to review this thesis, namely Prof. Florin Ciucu, Prof. Christoph Garth, Prof. Amr Rizk, and Prof. Jens Schmitt.

I would like to thank all my coauthors for their collaboration, namely Dr. Michael Beck, Prof. Steffen Bondorf, Dr. Anne Bouillard, Prof. Florin Ciucu, Dr. Sebastian Henningsen, Prof. Jens Schmitt, Malte Schütze.

I would like to thank two of my best friends, David and Sebastian. Bärenbande forever.

I would like to thank all the former and current distributed computer systems (disco) members, namely Dr. Michael Beck, Dr. Daniel Berger, Prof. Steffen Bondorf, Vlad-Cristian Constantin, Markus Espen, Anja Gerber, Anja Hamscher, Eric Jedermann, Dr. Kai Lampka, Carolina Nogueira, Steffen Reithermann, Dr. Matthias Schäfer, Alexander Scheffler, Prof. Hao Wang, and Raffaele Zippo, for the great time we had together over the last 5 years.

Thank you,
Paul

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

---

iid	independent and identically distributed
CDF	cumulative distribution function
DNC	deterministic network calculus
DRR	deficit round robin
e2e	end-to-end
EBB	exponentially bounded burstiness
EDF	earliest deadline first
fBm	fractional Brownian motion
fGn	fractional Gaussian noise
FIFO	first-in first-out
foi	flow of interest
GPS	generalized processor sharing
HP	high priority
LP	low priority
LRD	long-range dependence
MGF	moment-generating function
MMOO	Markov-modulated On-Off
MMP	Markov-modulated process
ND	negatively (orthant) dependent
NSC	network service curve
PBOO	pay bursts only once
PMOO	pay multiplexing only once
PSOO	pay segregation only once
RPPS	rate proportional processor sharing
seqPMOO	sequential pay multiplexing only once
seqSFA	sequential separated flow analysis
SDF	shortest-to-destination first
SFA	separated flow analysis
SNC	stochastic network calculus
SP	static priority
TFA	total flow analysis

Part I

INTRODUCTION AND NETWORK CALCULUS BACKGROUND

1 INTRODUCTION

1.1 A NEED FOR A STOCHASTIC END-TO-END ANALYSIS

Communication networks play a vital role in many of our daily activities. We rely heavily on the fact that we can obtain information or interact with our environment through Internet services in a short amount of time. In fact, an increasing number of applications require certain types of performance guarantee of a queueing system, e.g., to ensure safety criticality. By queueing system, we mean "any system in which arrivals place demands upon a finite-capacity resource" [Kle75, p. vii]. See, e.g., Internet at the speed of light [Sin+14], Tactile Internet [Fet14], Internet of Things [WADX15], or the envisioned cyber-physical systems [Raj+10].

Historically, this type of performance analysis was conducted with the help of *queueing theory*. It originated in the analysis of circuit-switched telephone networks at the beginning of the 20th century [Erl09, Erl17] and was later widely applied to packet-switched networks [HB13]. It basically consists of deriving the backlog or delay distribution based on assumptions on the arrival process, service distribution, and on the scheduling. A significant contribution can be found in the field of *product-form networks*, where, e.g., the queue distribution in the entire network can be described by the product of the queue distributions of each single system [Jac57, Jac63, Bas+75, Kel75, Kel76]. Yet, despite the advances made in the longer as well as in the recent past, the analysis of queueing networks is "largely constrained by the technical assumption of Poisson arrivals" [CS12]. Even though this type of queueing system still finds many applications, when it comes to Internet traffic however, it is considered impractical as self-similarity and long-range dependence have been observed to be ubiquitous [Lel+94, PF95, Wil+97, CB97, Fel+99].

1.2 THE STOCHASTIC NETWORK CALCULUS FRAMEWORK

(The) stochastic network calculus (SNC) is an alternative approach that avoids the problem of obtaining exact results by computing *per-flow performance bounds* of the form $\mathrm{P}(\text{backlog} > B) \leq \varepsilon$ or $\mathrm{P}(\text{delay} > T) \leq \varepsilon$ instead, where ε is a small value, e.g., 10^{-6}. It originated in the seminal work of Rene L. Cruz [Cru91a, Cru91b] in the early 1990s and was developed to derive nonprobabilistic bounds assuming worst-case arrival constraints and minimum service guarantees. His work was later supplemented by the *min-plus algebra* [Bac+92] and is called today the deterministic network calculus (DNC) [Cha00, LT01, Lie17, BBC18]. Shortly afterwards, stochastic extensions were introduced that were able to capture the statistical multiplexing gain [Kur92, YS93, Cha94].

The SNC basically consists of two branches: One that is based on moment-generating functions (MGFs) and one that employs envelope functions / tail bounds. The former assumes bounds on the MGF of the involved arrival and service processes [Chaoo, Fido6] and can be seen as a successor of the *effective bandwidth* theory [Kel91, Kel96]. The latter, on the other hand, can be interpreted as a direct stochastic relaxation of deterministic guarantees [Cru96, CBL06, JL08]. While more processes can be modeled with tail bounds, under the assumption of independence of all processes, [RF11, RF12b] comes to the conclusion that the MGF-based SNC leads to tighter delay bounds.

Two features of SNC (both branches) have been identified as key to overcome the "technical barriers of queueing networks" [CS12]:

- *Scheduling abstraction*: Specific properties of schedulers are "abstracted away" by a function called *leftover service*. It allows even for the analysis of *arbitrary multiplexing* [SZF08], i.e., we can give performance guarantees without specifying a specific scheduling algorithm;

- *Convolution-form networks*: The end-to-end (e2e) analysis can be simplified by convolving the service processes across all servers to a single end-to-end service.

Apart from these key aspects, a list of features is responsible for enabling a *uniform framework*:

- Instead of focusing on the Poisson arrival model as in queueing theory, often *arrival classes* such as (σ, ρ)-constrained arrival [Chaoo] or exponentially bounded burstiness (EBB) [YS93] are considered. Since many arrival processes belong to these classes, they can be interchanged effortlessly.

- In combination with the above scheduling abstraction, where we change the scheduling by simply inserting a different leftover service function, SNC (actually network calculus in general) yields a high level of *modularization*. In other words, arrival distributions and scheduling algorithms are reduced to components of a toolbox that we combine to create whole network topologies. This is a central implication of separating arrival and service processes.

- In order to calculate bounds on backlog or delay, we need to make statements about sample-path events. Both branches of SNC share the application of the Union bound to evaluate this type of event:

  $$\mathrm{P}\left(\max_{i=1,\ldots,n} X_i > x\right) \leq \sum_{i=1}^{n} \mathrm{P}(X_i > x)\,.$$

  In contrast to other inequalities on sample-path events (e.g, Doob's martingale inequality [Doo53, Theorem 3.2]) it does not come with any restrictive assumptions.

As a consequence of this uniform framework, SNC comes with a non-negligible price of a gap between exact results (that we try to quantify with simulations) and calculated bounds. In fact, there also exists a tight analysis for the single-node case for some traffic classes based on martingale techniques [CPS14, PC14, PC15, CPS16, CP18]; yet, an end-to-end martingale analysis remains an elusive goal.

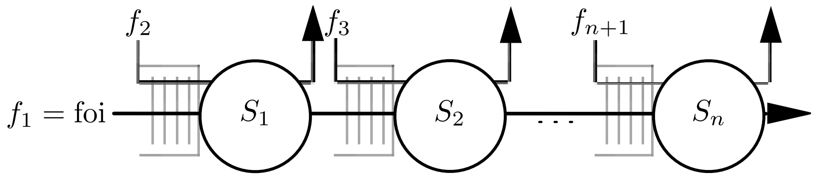


Figure 1.1: Canonical tandem

1.3 STATE-OF-THE-ART: THE CANONICAL TANDEM

At the beginning of Section 1.2, we already noted that SNC provides per-flow performance bounds. We call the flow for which we calculate bounds the flow of interest (foi). State-of-the-art results in SNC mostly focused on topologies where the foi traverses the "canonical tandem" in Figure 1.1, e.g., as in [CBL06, Fid06]. For the MGF-based analysis, this has the important advantage that, if all arrivals are initially assumed to be independent, this independence remains since all cross-flows immediately exit the network. While this makes a network analysis quite tractable, it is a very limiting assumption. Thus, in order to make progress in the end-to-end analysis, the SNC has to go beyond the canonical tandem. On the other hand, for a topology as in Figure 1.2, the outputs of the two flows depend on each other after sharing the resources at server S_1. For the state-of-the-art analysis in SNC (the so-called separated flow analysis), we have to consider dependencies. We call this dependence *method-pertinent*. Typically, one deals with the dependence of two arrival process A_1 and A_2 by the invocation of Hölder's inequality:

$$\mathrm{E}\left[e^{\theta(A_1(s,t)+A_2(s,t))}\right] \leq \mathrm{E}\left[e^{p\theta A_1(s,t)}\right]^{\frac{1}{p}} \mathrm{E}\left[e^{q\theta A_2(s,t)}\right]^{\frac{1}{q}}, \quad \text{for all } 0 \leq s \leq t,$$

where $\frac{1}{p} + \frac{1}{q} = 1$ and $p, q > 1$ (in other words, p and q are *Hölder conjugates* of each other). It often leads to conservative bounds and requires the optimization of an additional parameter for each application. While the runtimes of optimizing parameters have not been a focus of SNC literature, it is obvious that they could possibly explode if all dependencies in a complex network are bounded by Hölder's inequality. However, non-nested interference structures are typical when considering topologies of practical relevance. Therefore, it is imperative to find a stochastic analysis that benefits from stochastic independence but also comes with minimal method-pertinent dependencies. Otherwise, the gap between (simulated) exact results and calculated bounds mentioned in Section 1.2 above becomes larger the more dependencies occur. In the worst case, this means that we cannot provide reasonable performance bounds at all.

In the DNC, different analysis methods have been proposed and evaluated, such as total flow analysis (TFA), separated flow analysis (SFA), or pay multiplexing only one (PMOO) analysis [Fid03, SZ06, SZM06, Bou+08] by considering different combinations of the two key features, leftover service (scheduling abstraction) and convolution. While the TFA consists of an additive result for the end-to-end performance, the latter two, SFA and PMOO, calculate via convolution the *end-to-end service* that contains the information of the whole network topology. These algorithms can be interpreted as an evolution that makes the transition from a local perspective to a global one. Moreover, it has been shown that the PMOO analysis, by convolving first and calculating the

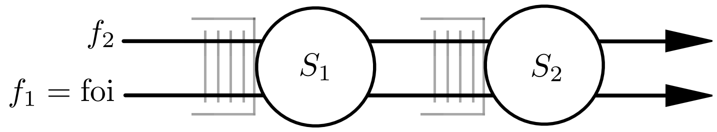


Figure 1.2: Two flows at two nodes in a tandem.

leftover service afterwards, one can further significantly improve the bounds[1] compared to SFA. While some works already started to integrate the PMOO, e.g., [ZBHB16], for the large majority of SNC research, the PMOO has not really been a focus. In particular, it was not really investigated how method-pertinent dependencies of processes impacts the MGF-based SNC in terms of bound accuracy and runtimes.

Since we calculate per-flow performance bounds, as a consequence, any feed-forward network can be transformed into a tandem from the foi's perspective. In DNC, this step has seen some advanced treatment recently [BNS17b] and has been shown to result in bounds with a high degree of accuracy. Yet in SNC, it has been largely neglected in the sense that no work beyond the standard output bound calculation was invested.

The authors of [BS13, Bec16a] released the *DISCO Stochastic Network Calculator*, the "first open-source tool to automate the steps of calculating MGF-based performance bounds". A strong and accessible tool support is of paramount importance for a wide deployment of SNC in practice. However, this tool shares the fate of the SNC with MGFs of struggling with method-pertinent dependencies. Therefore, any progress on the algorithmic aspect of the network analysis can only be of resounding success if accompanied by an according software development.

The authors of [CPS13] show in the single server case, that bounds obtained via martingale-based techniques provide tight performance bounds in comparison to simulations. However, this comes with the notable exception of GPS; here, even the martingale bounds indicate a notable gap compared to simulations. Yet, only a homogeneous case where all arrivals have the same parameters is considered.

1.4 THESIS STATEMENT AND CONTRIBUTIONS

THESIS STATEMENT: Reducing the gap between simulations and bounds in the end-to-end analysis of more complex topologies while maintaining the uniform framework is *the* main challenge of the stochastic network calculus. A key aspect in this reduction is how we deal with method-pertinent dependencies.

This thesis contributes to the SNC in several ways.

- We show that the PMOO analysis known from DNC cannot just significantly improve end-to-end delay bounds, but allows us to consider less method-pertinent dependencies. This is first presented for simple tandems with nested interference structures and then extended to arbitrary tree networks. This bound improvement is accompanied by a

1 However, there are some notable exceptions, see [SZF08].

reduced complexity that also leads a reduced number of parameters that need to be optimized. This, in turn, can reduce runtimes by several orders of magnitude depending on the used heuristic.

- We give a perspective on how these method-pertinent dependencies can further be reduced for flows that directly or indirectly share the capacity of a server. Even though we only give numerical evidence without a rigorous proof, we show how the concept of negative dependence can be used to obtain improved performance bounds.

- For the larger problem of end-to-end performance bounds of tree networks, we present a modification of the MGF output bound computation that mitigates the effect of the Union bound, the so-called h-mitigators. It is minimally invasive, as all existing results and procedures are still applicable. While numerical evaluations indicate that the gain is, on average, rather moderate, it is much stronger in some cases and does not impose any additional assumptions. It comes at the price of additional parameters to optimize, yet can be scaled.

- We conduct extensive numerical evaluations to substantiate the claims about the improvements. Moreover, we made a toolbox publicly available that allows for implementing the calculation of end-to-end performance bounds including the optimization. Subsequently, it can also be used to replicate the above mentioned results.

- Apart from arbitrary multiplexing, we perform different stochastic analyses of a popular fair scheduler, generalized processor sharing (GPS). We provide stochastic delay bounds for the case of independent arrivals as well as the general case. We give an overview of what can be considered as state-of-the-art analysis in the SNC and substantiate this claim through numerical evaluations.

1.5 THESIS OUTLINE

The first part is about the necessary background. In Chapter 2, we start with the DNC including the pay multiplexing only one (PMOO) analysis. Then, we continue with a background on SNC presenting the state-of-the-art analysis in Chapter 3. Since the thesis is almost exclusively based on SNC with MGFs, we only cover the MGF part and present the SNC with tail bound only as we need it. This chapter includes a modular analysis, where all network operations are presented as separated modules that can be applied sequentially, and direct end-to-end bounds that are able to avoid some sequencing penalties.

In the second part, we show how one can deal with dependence in a network. In Chapter 4, we show how the PMOO property can be applied to topologies with nested interference, tandem queues and sink trees. Here, we focus on time-dependent bounds that are numerically evaluated under the fractional Brownian motion (fBm) arrival model that is shown to capture the self-similar property of Internet traffic. Chapter 5 then extends the PMOO property to arbitrary tree networks in the SNC. We use the concept of negative dependence to improve performance bounds in Chapter 6.

The third part is about specific challenges of the end-to-end analysis. Chapter 7 shows how the effect of the Union bound can be mitigated when computing MGF-output bounds. Last but not least, we discuss our toolbox that is used to numerically evaluate our results in Chapter 8.

In the fourth part, we depart from the arbitrary multiplexing and conduct an SNC analysis of generalized processor sharing (GPS) (Chapter 9). Chapter 10 concludes the thesis.

The last part contains all appendices of the single chapters.

2 DETERMINISTIC NETWORK CALCULUS BACKGROUND

The SNC originates in the deterministic network calculus (DNC) and can be interpreted as a stochastic extension. The DNC is a modern framework to derive *hard* end-to-end performance guarantees, e.g., for the maximum backlog or delay. These hard guarantees are necessary for a plethora of time-sensitive applications, see, for example, the work on heterogeneous communication systems (HCS) or Avionics Full-Duplex Switched Ethernet (AFDX) [FFG06, Sup+10, BNF12]. We therefore introduce the typical DNC notation and results as we use them.

We start off with the basic network model we assume throughout this thesis.

2.1 NETWORK MODEL

We consider a basic queueing system as in Figure 2.1. Data arriving at some service element (mostly called *servers*) are abstracted as arrivals processes. If the system contains multiple servers, we call the arrival information together with its path a *flow*.

We consider two types of arrival processes of a flow, discrete-time or continuous-time. A discrete-time (cumulative) *arrival process* $A(t) : \mathbb{N} \to \mathbb{R}^+$ is defined as a sum of nonnegative measurable random variables $a_i \geq 0$:

$$A(t) := \sum_{i=1}^{t} a_i. \tag{2.1}$$

The a_i are called *increments.* On the other hand, a continuous-time (cumulative) arrival process $A(t) : \mathbb{R} \to \mathbb{R}^+$ is defined as the respective integral over a nonnegative stochastic process $a(x) \geq 0$:

$$A(t) := \int_0^t a(x)\,\mathrm{d}x. \tag{2.2}$$

Regardless of whether we consider discrete or continuous time, we assume that $A(t) = 0$ for all $t \leq 0$. If time and space are continuous, it is often called *fluid model*. If not stated otherwise, we assume arrival processes to be discrete in time and continuous in space. Furthermore, the data unit is taken to be as one bit. A queueing model where the arrivals have an index t representing time is also called *time domain model*. We define the (cumulative) arrivals over a time interval as

$$A(s,t) := A(t) - A(s),$$

and the according (cumulative) *departure process* as

$$D(s,t) := D(t) - D(s).$$

Let A be an arrival process and D a departure process of the same sample path. For these processes, *causality* holds:

$$D(t) \leq A(t), \quad \forall t \geq 0. \tag{2.3}$$

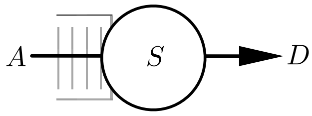


Figure 2.1: Basic queueing system with cumulative arrival process A and cumulative departure process D.

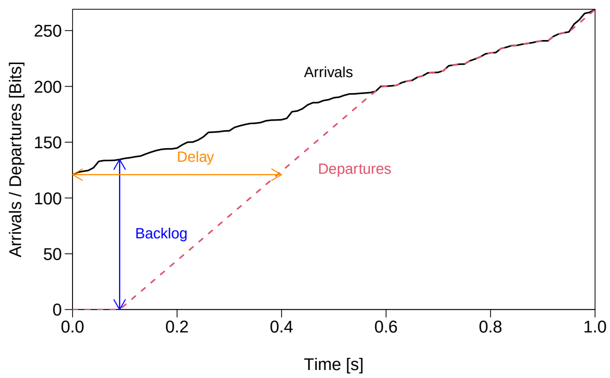


Figure 2.2: Backlog and virtual delay for an arbitrary arrival and departure process.

We define the difference of arrivals and departures to be the *backlog (process)* $q(t)$ at time $t \geq 0$:

$$q(t) := A(t) - D(t). \tag{2.4}$$

By causality (Eqn. (2.3)), it holds that $q(t) \geq 0$. Buffer sizes are always assumed to be infinite[1]. Therefore, throughout this thesis, we assume a *lossless* system and that no data is created by the server itself. If not stated otherwise, time is assumed to be discrete. Anticipating some results of the network calculus framework, one can already establish basic queueing dynamics for a constant rate server with rate $C \geq 0$ based on *Lindley's equation* [Lin52], [Cha00, p. 7]:

$$q(t) = [q(t-1) + a(t) - C]^+ , \tag{2.5}$$

under the initial condition $q(0) = 0$. In simple terms, it intuitively states that the backlog at time t is the backlog of the previous time slot plus new arrivals minus the server rate while maintaining nonnegativity. This recursion can be solved inductively and its explicit solution, also known as *Reich's equation*, is

$$q(t) = \max_{0 \leq s \leq t} \{A(s,t) - C \cdot (t-s)\} . \tag{2.6}$$

For the output, this yields

$$D(t) \overset{(2.4)}{=} A(t) - q(t) \overset{(2.6)}{=} \min_{0 \leq s \leq t} \{A(s) + C(t-s)\} . \tag{2.7}$$

We see later that the network calculus's concept of a service curve can be interpreted as a generalization of Eqn. (2.7).

The *virtual delay* at time $t \geq 0$ is defined as

$$d(t) := \inf \{s \geq 0 : A(t) \leq D(t+s)\} . \tag{2.8}$$

1 Even though most network calculus literature assumes buffers to be infinite, there also exist some results for queueing systems with loss [CL93, Liu93, GFC12, CPR19].

It is the time it takes for the departures to "catch up" with the arrivals. If the arrivals are served in first-in first-out (FIFO) order, the $d(t)$ can be interpreted as the *delay* of a hypothetical bit entering the system at time t. Throughout this thesis, we assume that no reordering inside of a flow occurs (*FIFO per-flow* or *locally-FIFO*). Even though data packet delays and the virtual delay in Eqn. (2.8) are not necessarily equal (packet delays, in contrast to the virtual delay, are always > 0), it has been shown numerically that the difference between the two quantities is negligible [Ciu07, pp. 109].

2.2 MIN-PLUS ALGEBRA

Network calculus is a system-theoretic framework that employs the min-plus algebra [Bac+92]. It facilitates an elegant computation in terms of an end-to-end analysis. In linear system theory, a key operation is the convolution of two functions:

$$f * g(t) := \int_{-\infty}^{\infty} f(t-s)g(s)\,\mathrm{d}s. \tag{2.9}$$

In min-plus algebra the conventional operations $(+, \cdot)$ are replaced by $(\wedge, +)$:

Definition 2.1 (Min-Plus Convolution)**.** Let x and y be nonnegative, increasing functions such that $x(t) = y(t) = 0$ for all $t < 0$. The (univariate) *min-plus convolution* of x and y is defined for $t \geq 0$ as

$$x \otimes y\,(t) := \inf_{0 \leq s \leq t} \{x(t-s) + y(s)\}\,. \tag{2.10}$$

For $t < 0$, we define $x \otimes y\,(t) := 0$.

The class of functions being 0 for negative t and nondecreasing otherwise is sometimes called *wide-sense increasing* [LT01, p. 111].

Basic systems theory characterizes the input-output property of so-called *time-invariant linear systems* [LT01, pp. xiv], [Fid10]. Assuming that input signals A and output signals D are given, time-invariance means that a time-shifted version of an input signal $A(t-s)$ results in an accordingly shifted but otherwise identical output output signal $D(t-s)$.

The system is linear, if any linear combination of input signals $c_1A_1 + c_2A_2$, where c_1, c_2 are some constants, results in a corresponding output signal $c_1D_1 + c_2D_2$.

In systems theory, there exists a fundamental result stating that a system is time-invariant and linear iff (if and only if) the output signal of the system is given as

$$D(t) = \int_{-\infty}^{\infty} A(s) \cdot S(t-s)\,\mathrm{d}s = A * S(t), \tag{2.11}$$

where $*$ is the conventional convolution operator (see Eqn. (2.9)) and $S(t)$ is the system response to the Dirac unity impulse [Fid10].

One can show that the min-plus convolution is a linear operator on the min-plus algebra under a certain condition (see, e.g., [CO96], [LT01, pp. 111]). In fact, we see in the next section (Section 2.3), that this condition is an *exact service curve* (Definition 2.5 below), a property very similar to Eqn. (2.11), but using the min-plus convolution instead.

We see in Section 2.5 that the min-plus algebra also allows for an elegant framework to derive end-to-end performance guarantees. According to [Cha00, pp. 102], the idea of applying min-plus algebra in the network calculus has been developed simultaneously in [Cha97, CO96, AR96, Le 98].

Another key operation in network calculus in the deconvolution.

Definition 2.2 (Min-Plus Deconvolution)**.** Let x and y be nonnegative, increasing functions $\mathbb{R} \to \mathbb{R}^+ \cup \{+\infty\}$ such that $x(t) = y(t) = 0$ for all $t < 0$. The (univariate) *min-plus deconvolution* of x and y is defined as

$$x \oslash y\,(t) := \sup_{u \geq 0} \{x(t+u) - y(u)\}\,. \tag{2.12}$$

Note that the deconvolution is not defined for $x = y = +\infty$.

2.3 ARRIVAL AND SERVICE CURVES

DNC provides worst-case performance bounds under arrival constraints and service guarantees. We start this section with the arrival constraints.

Arrival curves / envelopes

The seminal work by Cruz [Cru91a, Cru91b] marks a paradigm shift in the performance analysis of communication network as it is "nonprobabilistic". Instead, traffic specifications are limited to some regulatory constraints for all time intervals.

Let $\alpha : \mathbb{R} \to \mathbb{R}^+ \cup \{\infty\}$ be a nonnegative, increasing function such that $\alpha(t) = 0$ for all $t < 0$.

Definition 2.3 (Arrival Curve)**.** We say that α is an *arrival curve* (or *envelope*) for an arrival process A if for all $s \leq t$

$$A(s,t) \leq \alpha(t-s). \tag{2.13}$$

Equivalently, we can write for all t

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

If we have $\alpha(0) = 0$, one can even show that $A = A \otimes \alpha$ [Agr+99]. Another equivalent definition of an arrival curve is

$$\begin{aligned} & \sup\nolimits_{u \geq 0} \{A(t+u) - A(u)\} && \leq \alpha(t) \\ \Leftrightarrow \quad & A \oslash A && \leq \alpha. \end{aligned}$$

It is crucial that α is a univariate function. In other words, it only depends on the difference of s and t, but not on the absolute values. An important example of an arrival curve is the *leaky bucket / token bucket arrival curve* [Cru91a, Cru91b], which is defined in discrete time as

$$\alpha(t) = \gamma_{r,b} := \sigma + \rho \cdot t, \tag{2.14}$$

where $\sigma, \rho \in \mathbb{R}^+$. We interpret σ as the *(instantaneous) burst* of the arrivals, whereas ρ is the *long-term average rate*.

For an aggregate of arrivals, e.g., this is necessary when we *multiplex flows*, $A(t) = \sum_{i=1}^{n} A_i$, where each A_i has an according arrival curve α_i. It holds that

$$A(s,t) \leq \sum_{i=1}^{n} \alpha_i(t-s), \quad \text{for all } 0 \leq s \leq t.$$

This means for the special case of a leaky bucket arrival curve that $A(s,t) \leq \sigma + \rho(t-s)$, where $\sigma = \sum_{i=1}^{n} \sigma_i$ and $\rho = \sum_{i=1}^{n} \rho_i$. The aggregate of two arrival curves α_1 and α_2 is sometimes denoted by $\alpha_1 \oplus \alpha_2$.

Service curves

As we already hinted at in Section 2.2, service curves play a major role in the network calculus's interpretation of a min-plus linear system. They enable two key features we discussed in Chapter 1: *Scheduling abstraction* and *convolution-form networks*. The modeling capability is sometimes emphasized in the literature by using the term network element, as it is not limited to servers in the sense of network devices. We start off with an intuitive definition and consider a generalization afterwards. Yet before, we define the *start of a backlogged period* before t as

$$s_0 := \sup \{ s \leq t \mid A(s) = D(s) \} . \tag{2.15}$$

Note that, such an s_0 always exists ($A(0) = D(0) = 0$) and if the system is idle at t, then $s_0 = t$.

Definition 2.4 (Strict Service Curve [CO96])**.** A server is said to offer a *strict service curve* β to a flow if, during any (continuously) backlogged period $[s,t)$, the departures of this flow in the system is at least equal to $\beta(t-s)$, that is,

$$D(s,t) \geq \beta(t-s). \tag{2.16}$$

The notion of a strict service curve plays an important role in the scheduling abstraction, as many mathematical derivations require a strict service curve in order to derive per-flow guarantees.

However, the strict service curve property cannot be preserved when considering the convolution of strict service curves (we discuss this later in this section in more detail). Therefore, a generalization with a stronger modeling capability in the form of *service curves* is necessary. It originates in the work of [PG93, PG94] and is later formalized in [Cru95, SCP95]. For the sake of conciseness, we introduce $\mathcal{F}_0$ to be the set of real-valued, nonnegative, increasing functions with $f(t) = 0$:

$$\mathcal{F}_0 := \left\{ f : \mathbb{R} \to \mathbb{R}^+ \cup \{+\infty\} \mid \forall s \leq t : 0 \leq f(s) \leq f(t),\ f(0) = 0 \right\},$$

where $\mathbb{R}^+ = [0, \infty)$.

Definition 2.5 (Service Curve)**.** Consider a server and a flow with according arrival process A and departure process D. The server offers a *(minimum) service curve* β to A if $\beta \in \mathcal{F}_0$ and for all $t \in \mathbb{R}$

$$D(t) \geq A \otimes \beta\,(t) = \inf_{0 \leq s \leq t} \{ A(t-s) + \beta(s) \} . \tag{2.17}$$

The service curve is called *exact*, if Eqn. (2.17) holds with equality.

One can easily show that, if a server offers a strict service curve β to a flow such that β is a nonnegative, increasing function with $\beta(0) = 0$, then β is also a service curve. Yet, the opposite is not true in general (counterexamples are reported in, e.g., [Fid10] and [LT01, p. 177]).

A typical example of a service curve is the rate-latency server curve [LT01, p. 106]:

$$\beta(t) = \beta_{R,T}(t) := R \cdot [t-T]^+ = \begin{cases} R(t-T), & \text{if } t > T \\ 0, & \text{otherwise.} \end{cases} \tag{2.18}$$

For a detailed discussion on service curves, see [LT01, BJT09, Sch+11, Bou14, BBC18].

2.4 SINGLE-NODE PERFORMANCE BOUNDS

The network calculus framework is able to derive deterministic bounds for backlog, delay, and output. While the focus on single-node bounds appears limiting at first, it is actually the core of the performance analysis. The reason is that the scheduling abstraction as well as the convolution property [CS12] enable us to reduce the performance analysis of any feed-forward network to the single-node case.

Theorem 2.6 (DNC Performance Bounds)**.** *Assume an arrival process A traversing a server $\mathcal{S}$. Further, let the arrivals be constrained by arrival curve α and let the system offer a service curve β.*

1. *The backlog $q(t)$ satisfies for all t*

$$q(t) \leq \sup_{s \geq 0} \{\alpha(s) - \beta(s)\} = \alpha \oslash \beta\,(0). \tag{2.19}$$

2. *The virtual delay $d(t)$ satisfies for all t*

$$\begin{aligned} d(t) &\leq \inf \left\{ \tau \geq 0 \mid \sup_{t \geq 0} \{\alpha(t) - \beta(t+d)\} \leq 0 \right\} \\ &= \inf \{\tau \geq 0 \mid \alpha \oslash \beta\,(-\tau) \leq 0\} \\ &= \sup_{s \geq 0} \{\inf \{\tau \geq 0 \mid \alpha(s) \leq \beta(s+\tau)\}\}. \end{aligned} \tag{2.20}$$

3. *The departures of the flow, D, satisfies for all t*

$$D(s,t) \leq \alpha \oslash \beta\,(t-s) =: \alpha'(t-s). \tag{2.21}$$

Note that in Eqn. (2.20), the deconvolution from the second line does not follow directly from the first line but can be derived separately. Similarly, the third line requires another derivation, too. Proofs of Theorem 2.6 can be found in [Cha00, pp. 42] (for the discrete-time case) or in [LT01, pp. 22] (for left-continuous processes). The bounds are tight meaning that one can create sample paths ("greedy / lazy scenario") such that they hold with equality [LT01, pp. 27]. In the case of token bucket envelopes $\alpha(t) = \sigma + \rho t$ and rate-latency servers $\beta(t) = \beta_{R,T}(t) = R\,[t-T]^+$, Theorem 2.6 yields under the *stability condition*

$$r \leq R \tag{2.22}$$

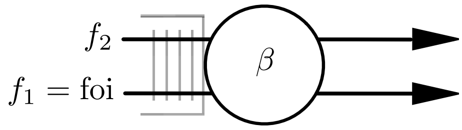


Figure 2.3: Two flows at one node with service curve β.

that

$$q(t) \leq \sigma, \tag{2.23}$$

$$d(t) \leq \frac{\sigma}{R} + T, \tag{2.24}$$

$$D(s,t) \leq \sigma + \rho T + \rho(t-s). \tag{2.25}$$

2.5 END-TO-END ANALYSIS AND PAY BURST ONLY ONCE PRINCIPLE

We start by considering the case of multiple flows arriving at one server (depicted in Figure 2.3). Since network calculus offers per-flow performance bounds, we have to select one that is subject of our analysis, the flow of interest (foi). In addition, the scheduling of the flows has to be considered. A large number of schedulers can be modeled with the help of (strict) service curves. For example, static priority (SP) [Chaoo, pp. 60], [LT01, p. 20], [Scho3], [LBL07], [BBC18, pp. 170], FIFO [Cru98], [LT01, p. 177], generalized processor sharing (GPS) [PG93], [PG94], [Chaoo, p. 67], [LBL07], [BL18], [BBC18, p. 171], or earliest deadline first (EDF) [LBL07], [LGFB11], to name a few.

The usage of bounds enables us to provide service guarantees without assuming a specific order in which flows or traffic is served. This model is referred to as *arbitrary multiplexing* or *blind multiplexing* [LT01, p. 20], [SZF08], or [BBC18, pp. 156].

Theorem 2.7 (Leftover Service Curve for Arbitrary Multiplexing)**.** *Let $t \geq 0$. Consider a server that arbitrarily multiplexes two flows f_1 and f_2, where the arrivals of f_2, A_2, are constrained by α_2. Further, assume that the server guarantees a strict service curve β to the aggregate of the flows. Then, the leftover service*

$$\beta^1_{\text{l.o.}}(t) := \beta^1(t) := [\beta(t) - \alpha_2(t)]^+ \tag{2.26}$$

is a service curve for flow f_1 if $\beta^1 \in \mathcal{F}_0$. This is also denoted by $\beta^1 = \beta \ominus \alpha_2$.

Proof. Let t_0 be the start of the backlogged period before t as in Eqn. (2.15). By assumption, it holds that $D_1(t_0) = A_1(t_0)$ and $D_2(t_0) = A_2(t_0)$. Since β is a strict service curve for the aggregate, we have that

$$D_1(t) + D_2(t) - D_1(t_0) - D_2(t_0) \geq \beta(t - t_0).$$

which yields

$$\begin{aligned} D_1(t) &\geq \underbrace{D_1(t_0)}_{=A_1(t_0)} + \beta(t-t_0) + \underbrace{D_2(t_0)}_{A_2(t_0)} - D_2(t) \\ &= A_1(t_0) + \beta(t-t_0) + A_2(t_0) - D_2(t). \end{aligned}$$

Further, since we assumed that that A_2 has arrival curve α_2, we conclude that

$$D_2(t) - A_2(t_0) \overset{(2.3)}{\leq} A_2(t) - A_2(t_0) \overset{(2.13)}{\leq} \alpha_2(t - t_0).$$

This yields that

$$D_1(t) \geq A_1(t_0) + \beta(t - t_0) - \alpha_2(t - t_0).$$

On the other hand, we know since D_1 is increasing that

$$D_1(t) \geq D_1(t_0) = A_1(t_0).$$

Combining both inequalities gives

$$\begin{aligned} D_1(t) &\geq A_1(t_0) + [\beta(t - t_0) - \alpha_2(t - t_0)]^+ \\ &\geq \inf_{0 \leq s \leq t} \left\{ A_1(s) + [\beta(t - s) - \alpha_2(t - s)]^+ \right\} \\ &= A_1 \otimes [\beta - \alpha_2]^+ \ (t) \end{aligned}$$

which proves the theorem. □

If not stated otherwise, we always derive performance bounds under arbitrary scheduling. Note that β needs to be a strict service curve. For minimum service curves, counterexamples can be constructed (see, e.g., [LT01, p. 177], or [BBC18, p. 158]). In case $[\beta(t) - \alpha_2(t)]^+$ is not a nonnegative, increasing function, leftover service curves can still be obtained with the help of the nonnegative and increasing *closure* [BBC18, p. 45].

If we have arrivals with token bucket arrival curve $\alpha_2(t) = \sigma_2 + \rho_2(t)$ and rate-latency service curve $\beta(t) = \beta_{R,T}(t)$, we obtain for $\rho_2 < R$ that

$$\beta^1(t) = [\beta(t) - \alpha_2(t)]^+ = \beta_{R-\rho_2, T + \frac{\sigma_2 + \rho_2 T}{R - \rho_2}}, \tag{2.27}$$

which is also in $\mathcal{F}_0$.

Theorem 2.7 can easily be generalized to $m - 1$ cross-flows as long as all f_i are constrained by α_i, $i = 2, \ldots, m$:

$$\beta^1(t) = \left[\beta(t) - \sum_{j=2}^{m} \alpha_j(t) \right]^+ .$$

Furthermore, if the leftover service curve needs to be strict again, for example, when considering hierarchical scheduling, one could subtract the output bound (Eqn. (2.21)) of the cross flow instead (see [BBC18, pp. 159]).

Theorem 2.7 shows how to convert a multiple flows - single server scenario to the basic single flow - single server by the use of the leftover service curve.

Next, we discuss the case of a single flow with multiple servers in a tandem (Figure 2.4) and how it can be analyzed with the concatenation theorem. The combination of leftover service curve, *Concatenation Theorem* (Theorem 2.8 below), and output bound then facilitate an end-to-end analysis of feedforward networks.

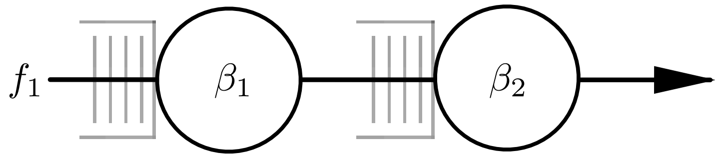


Figure 2.4: One flow at two nodes in a tandem with service curves β_1 and β_2.

Theorem 2.8 (Concatenation Theorem). *Assume a flow with arrival process $A_{\text{sys}} = A_1$ traverses the servers S_1 and S_2 in sequence which offer service curves β_1 and β_2, respectively (see Figure 2.4). Let $D_1 = A_2$ be the output of S_1 / the input to S_2. Then, the concatenation of the two servers offers a service curve of $\beta_{\text{e2e}} = \beta_1 \otimes \beta_2$ to the arrival process.*

The proof is actually a simple consequence of the service curve property combined with the associativity and closure under $\mathcal{F}_0$ of the min-plus convolution [Chaoo, pp. 41], [LT01, pp. 28]:

$$\begin{aligned} D_{\text{sys}}(t) &= D_2(t) \\ &\geq A_2 \otimes \beta_2\,(t) \\ &= D_1 \otimes \beta_2\,(t) \\ &\geq (A_1 \otimes \beta_1) \otimes \beta_2\,(t) \\ &= A_1 \otimes (\beta_1 \otimes \beta_2)\;(t) \\ &= A_{\text{sys}} \otimes \beta_{\text{e2e}}\,(t) \end{aligned}$$

for all $t \geq 0$.

Similar to Theorem 2.7, we can easily extend the result to n servers in a tandem:

$$\beta_{\text{e2e}} = \beta_1 \otimes \beta_2 \otimes \cdots \otimes \beta_n =: \bigotimes_{j=1}^{n} \beta_j. \tag{2.28}$$

Due to the associativity of the convolution operation [LT01, pp. 111], Eqn. (2.28) is well defined. In the case of rate-latency servers, one can easily show that

$$\beta_{R_1,T_1} \otimes \beta_{R_2,T_2}\,(t) = \beta_{\min\{R_1,R_2\},T_1+T_2}. \tag{2.29}$$

In other words, the latencies add up while the rate is reduced to the minimum of all rates.

We remark that only (minimum) service curves are necessary for the Concatenation Theorem. Yet, even if we assume both service curves to be strict, the convolution of both is not necessarily strict anymore. Examples are constructed in [SGM08, Example 2] or [BJT09, Theorem 3].

Note that we could derive performance bounds for a tandem of servers without Theorem 2.8 by simply computing the output bound in Eqn. (2.21). Yet, one can easily show that the resulting delay bound scales significantly worse in the number of servers. Let us assume that we have a flow with arrival constrained by a token bucket envelope $\alpha(t) = \sigma + \rho t$ that traverses a tandem of m servers with equal rate-latency service curve $\beta(t) = \beta_{R,T}(t)$.

1. We start with the "hop-by-hop" analysis, also called the total flow analysis (TFA) [Cru91b, SZ06]. One can easily show that for a token bucket arrival

curve and a rate latency server, the output envelope (which is equal to the input envelope of the next server) is

$$\alpha_2(t) = \alpha_1'(t) \overset{(2.21)}{=} \alpha_1 \oslash \beta_{R,T}(t) = \sigma + \rho T + \rho t, \quad \forall t.$$

In other words, we observe a *burstiness increase* of ρT. Repeating the computation gives us for the envelope at the last server for all $t \geq 0$

$$\begin{aligned} \alpha_n(t) &= \alpha_{n-1}'(t) \\ &= \alpha_{n-1} \oslash \beta_{R,T}(t) \\ &\vdots \\ &= ((\alpha_1 \oslash \beta_{R,T}) \oslash \cdots \oslash \beta_{R,T}) \oslash \beta_{R,T}(t) \\ &= \sigma + (n-1) \cdot \rho T + \rho \cdot t. \end{aligned}$$

At each server j, we obtain a delay bound

$$d_j(t) \overset{(2.20)}{\leq} \inf\left\{\tau \geq 0 \mid \alpha_j \oslash \beta_{R,T}(-\tau) \leq 0\right\} \overset{(2.24)}{=} \frac{\sigma + (j-1)\rho T}{R} + T.$$

This results in the additive delay bound

$$d(t) \leq d^{\mathrm{TFA}} = \sum_{j=1}^{n} \left(\frac{\sigma + (j-1) \cdot \rho T}{R} + T \right) = n \cdot \frac{\sigma}{R} + nT + \frac{\rho T}{2R} n(n-1).$$

In other words, the delay bound increases quadratically in the number of servers $\left(\mathcal{O}(n^2)\right)$.

2. We compare the "hop-by-hop" result with the delay bound based on the Concatenation Theorem (Theorem 2.8). It is also known as the separated flow analysis (SFA) [SZF08]. Here, we have

$$\beta_{\mathrm{e2e}}(t) \overset{(2.28)}{=} \bigotimes_{j=1}^{n} \beta_j(t) \overset{(2.29)}{=} \beta_{R,nT}(t)$$

and this leads to the delay bound

$$d(t) \leq d^{\mathrm{SFA}} \overset{(2.20)}{=} \inf\left\{\tau \geq 0 \mid \alpha_1 \oslash \beta_{R,nT}(-\tau) \leq 0\right\} \overset{(2.24)}{=} \frac{\sigma}{R} + nT.$$

First, we note the delay bound bound using Theorem 2.8 scales linearly in the number of servers $\left(\mathcal{O}(n)\right)$. Second, we observe that the burst σ only has to be "paid" once, in contrast to the additive delay bound where we must do it n times. This phenomenon is often called the pay bursts only once (PBOO) property.

2.6 PAY MULTIPLEXING ONLY ONCE PRINCIPLE AND STATE OF THE ART

In the following, we discuss some caveats of the SFA and possible solutions. Let us consider the topology in Figure 2.5. We assume two flows f_1 and f_2 with respective arrival curves α_1 and α_2 and two servers offering strict service curves to the arriving aggregate. For this tandem, it has been shown that SFA leads to

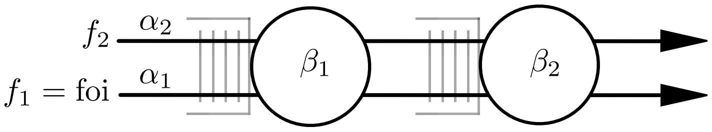


Figure 2.5: Two flows with envelopes α_1 and α_2 at two nodes in a tandem with service curves β_1 and β_2, respectively.

a tighter delay bound than TFA which is the reason why we omit the TFA in the following [SZF08]. Here, the SFA basically follows the "first subtract, then convolve"-rule. Hence, subtracting first yields the topology in Figure 2.6.

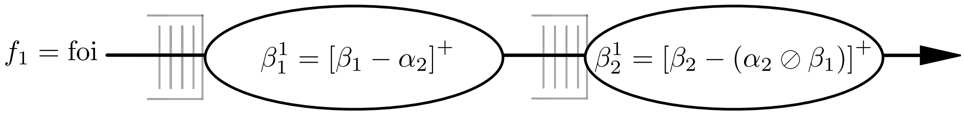


Figure 2.6: Subtract first

Note that we need the output bound (Eqn. (2.21)) of flow f_2 to obtain the leftover service curve β_2^1 (the leftover service curve for flow f_1 at the second server). Now, we can use the Concatenation Theorem (Theorem 2.8) to end up with Figure 2.7.

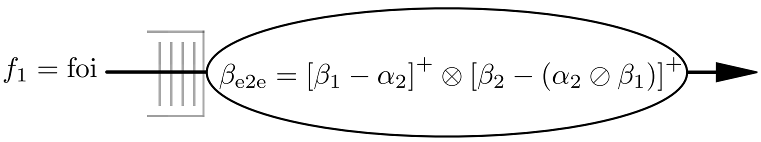


Figure 2.7: Application of SFA to the tandem network.

Let us assume that f_1 and f_2 have token bucket arrival curves $\alpha_i(t) = \sigma_i + \rho_i t, i = 1, 2$ and that both servers provide a strict rate-latency service curve $\beta_{R,T}$. Then, the SFA delay bound is [SZF08]

$$d^{\text{SFA}} = 2T + \frac{\sigma_1 + 2\sigma_2 + 3\rho_2 T}{R - \rho}.$$

We notice that under SFA, even though ensuring the pay burst only principle, we have the burst term σ_2 twice. This additional term is therefore caused by the multiplexing of flows. Yet, we can avoid this by convolving servers first and subtracting cross-flows subsequently. This is known in the literature as the pay multiplexing only once (PMOO) analysis [Fid03, SZ06, SZM08, Bou+08]. Application of the convolution in the first step yields the network in Figure 2.8.

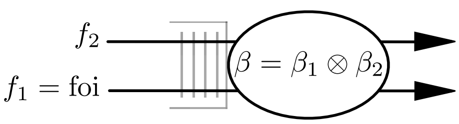


Figure 2.8: Convolve first

Afterwards, we apply Theorem 2.7 to obtain a leftover service curve for flow f_1 (Figure 2.9).

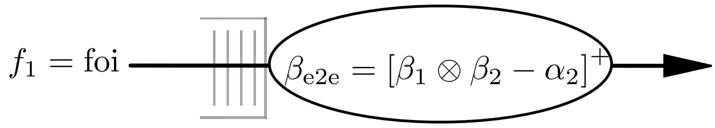


Figure 2.9: Application of PMOO to the tandem network.

If we assume again token bucket arrival curves and the same rate-latency service curve, we receive

$$d^{\text{PMOO}} = 2T + \frac{\sigma_1 + \sigma_2 + 2\rho_2 T}{R - \rho}.$$

Comparing both delay bounds, we find that the PMOO delay is tighter. More importantly, we observe that the burst of flow f_2, σ_2, is paid only once in the PMOO, a phenomenon that is eponymous for the analysis.

Given that the convolution of strict service curves in not strict in general anymore, and that strict service curves are necessary to subtract cross-flows, at first glance, it is surprising that the PMOO analysis is actually rigorous. Yet, it has been shown that the bounds obtained via PMOO are, indeed, valid [SZM08], [BJT09], [BBC18, pp. 236]. There are two variants of PMOO, one for a *nested interference* and one for a *overlapping interference*.

PMOO for nested interference (sequential PMOO)

In simple terms, we say that one flow f_i is *nested* into another flow f_j, if all servers f_i consecutively crosses are also crossed by f_j [Bon16]. We say that two flows are *not related* if they do not share any servers along their paths. A tandem has a *nested interference pattern*, if all flows are either nested or not related [LMS07]. Figure 2.10 depicts a simple nested tandem with three servers. Here, the order of operations of PMOO is uniquely defined: First, we convolve the strict service curves β_1 and β_2, since they share the same cross-flows. Then, we subtract the arrival curve α_3 before convolving the result with the strict service curve β_3. Subtracting the arrival curve of f_2 remains as a last step, resulting in the leftover service curve

$$\beta_{\text{e2e}}^{\text{PMOO}} = \left[\left([(\beta_1 \otimes \beta_2) - \alpha_3]^+ \otimes \beta_3\right) - \alpha_2\right]^+.$$

Since all operators are applied sequentially, we also call this sequential pay multiplexing only once (seqPMOO) and write $\beta_{\text{e2e}}^{\text{seqPMOO}}$.

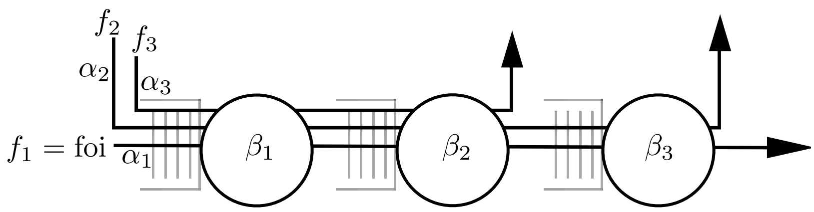


Figure 2.10: Nested tandem

For the sake of comparison, we also derive the SFA leftover service curve, that is, we subtract cross-flows first before convolving the single leftover service curves at each server:

$$\begin{aligned} \beta_{\mathrm{e2e}}^{\mathrm{SFA}} &= \beta_1^1 \otimes \beta_2^1 \otimes \beta_3^1 \\ &= [\beta_1 - \alpha_2 - \alpha_3]^+ \otimes [\beta_2 - \alpha_2' - \alpha_3']^+ \otimes [\beta_3 - \alpha_2'']^+ \\ &\overset{(2.21)}{=} [\beta_1 - (\alpha_2 + \alpha_3)]^+ \otimes [\beta_2 - ((\alpha_2 + \alpha_3) \oslash \beta_1)]^+ \\ &\quad \otimes [\beta_3 - ((\alpha_2 \oslash \beta_1^2) \oslash \beta_2^2)]^+ . \end{aligned}$$

Note that the arbitrary multiplexing leftover service curve is directly connected to a worst-case analysis in that if we do not assume any scheduling, we have to identify the scenario that leads to the worst case. We can see this in the third line: $[\beta_1 - (\alpha_2 + \alpha_3)]^+$ and $[\beta_2 - ((\alpha_2 + \alpha_3) \oslash \beta_1)]^+$ basically indicate that the cross-flows f_2 and f_3 are prioritized at these servers. We denote this by $f_1 \prec \{f_2, f_3\}$. The leftover service curves for flow f_2 remain to be computed. In order to enforce the worst case at the second server, we assume that $f_2 \prec f_3$ giving us

$$\alpha_2 \oslash \beta_1^2 = \alpha_2 \oslash [\beta_1 - \alpha_3]^+ .$$

The reason is that giving α_2 only the leftover service (which is a result of the priority assignment) only increases the output burstiness. This can easily be seen if we assume token bucket arrival curves and rate-latency servers. We see in Eqn. (2.27) that, once we apply the leftover operation, the leftover server rate is decreased and the leftover latency is increased. In Eqn. (2.25), on the other hand, under stability, the rate of the output envelope is equal to the rate of the original envelope, yet, the burstiness in increased by ρT. In other words, if we increase the latency, we increase the output burstiness. Continuing our analysis, we notice that the worst-case priority assignment is the opposite for β_2^2 in $((\alpha_2 \oslash \beta_1^2) \oslash \beta_2^2)$, since, in order to enforce the worst case, we have to give flow f_2 priority ($f_3 \prec f_2$) to see the highest output bound of flow f_3 :

$$\beta_2^2 = \left[\beta_2 - \left(\alpha_3 \oslash [\beta_1 - \alpha_2]^+\right)\right]^+ .$$

Concluding, the *dynamic priority assignment* allows us to construct a valid worst-case leftover service curve, but at the price of making contradicting priority assumptions at different points in the analysis. The problem is called *segregation* in the literature and an analysis that is able to avoid these contradicting priority assignment has the pay segregation only once (PSOO) property [BS16]. Yet, neither the SFA nor the PMOO analysis actually have this property.

PMOO for overlapping interference

Now, we consider a canonical example for the overlapping interference, the "overlapping tandem", in Figure 2.11. In contrast to the nested tandem, we do not have any adjacent servers with equal interference structure. Let the flow f_i with arrival processes A_i have an arrival curve α_i, $i = 1, 2, 3$. We assume that each server j offers a strict service curve β_j, $j = 1, 2, 3$. In order to obtain a leftover service curve, one could subtract one cross-flow first. Since we have two options, this leads to

$$\beta_{\mathrm{e2e}}^{\mathrm{seqPMOO},1} = \left[\left(\beta_1 \otimes [\beta_2 - \alpha_3]^+\right) - \alpha_2\right]^+ \otimes [\beta_3 - \alpha_3 \oslash \beta_2]^+$$

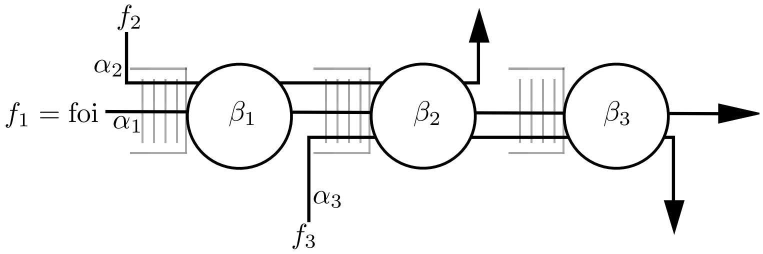


Figure 2.11: Overlapping tandem

and

$$\beta_{\text{e2e}}^{\text{seqPMOO},2} = [\beta_1 - \alpha_2]^+ \otimes \left[\left([\beta_2 - \alpha_2 \oslash \beta_1]^+ \otimes \beta_3 \right) - \alpha_3 \right]^+ .$$

However, in this case, we would have to pay multiplexing twice, so the seqPMOO analysis does not have the PMOO property in general.

Let us now consider an alternative approach in order to preserve this property. It has been reported first in [SZM06, SZM08]. We choose t_1, t_2, t_3 such that $t - t_3$ is the start of the backlogged period (Eqn. (2.15)) of server 3, $t - t_2 - t_3$ is the start of the backlogged period of server 2, and $t - t_1 - t_2 - t_3$ is the start of the backlogged period of server 1. Further, we denote the output of flow f_i at server j by D_i^j (for the output of the last server of a flow, we omit the superscript). Following along the lines of the proof of Theorem 2.7, it holds that

$$\begin{aligned}
D_1(t) + D_3(t) - D_1(t-t_3) - D_3(t-t_3) &\geq \beta_3(t_3), \\
D_1^2(t-t_3) + D_2^2(t-t_3) + D_3^2(t-t_3) & \\
-D_1^2(t-t_2-t_3) - D_2^2(t-t_2-t_3) - A_3(t-t_2-t_3) &\geq \beta_2(t_2), \\
D_1^1(t-t_2-t_3) + D_2^1(t-t_2-t_3) - D_1^1(t-t_1-t_2-t_3) - D_2^1(t-t_1-t_2-t_3) &\geq \beta_1(t_1).
\end{aligned}$$

Since $t - \sum_{k=j}^{1} t_j$ is the start of a backlogged period of server j, we have that $D_i^j\left(t - \sum_{k=j}^{1} t_j\right) = A_i^j\left(t - \sum_{k=j}^{1} t_j\right)$, for $j = 1, 2$. This gives

$$\begin{aligned}
D_1(t) + D_3(t) - D_1^2(t-t_3) - D_3^2(t-t_3) &\geq \beta_3(t_3), \\
D_1^2(t-t_3) + D_2^2(t-t_3) + D_3^2(t-t_3) & \\
-D_1^1(t-t_2-t_3) - D_2^1(t-t_2-t_3) - A_3(t-t_2-t_3) &\geq \beta_2(t_2), \\
D_1^1(t-t_2-t_3) + D_2^1(t-t_2-t_3) - A_1(t-t_1-t_2-t_3) - A_2(t-t_1-t_2-t_3) &\geq \beta_1(t_1).
\end{aligned}$$

Summing all three inequalities and simplifying leads to

$$\begin{aligned}
&D_1(t) + \underbrace{D_2^2(t-t_3)}_{\leq A_2(t-t_3)} + \underbrace{D_3(t)}_{\leq A_3(t)} - A_1(t-t_1-t_2-t_3) - A_2(t-t_1-t_2-t_3) - A_3(t-t_2-t_3) \\
&\geq \sum_{j=1}^{3} \beta_j(t_j).
\end{aligned}$$

Making use of causality (see also Eqn. (2.3)) and isolating $D_1(t)$ on the left-hand side yields

$$D_1(t) \geq A_1(t-t_1-t_2-t_3) - A_2(t-t_1-t_2-t_3, t-t_3) - A_3(t-t_2-t_3, t) + \sum_{j=1}^{3} \beta_j(t_j).$$

Using that $t - t_1 - t_2 - t_3$ is, by assumption, the start of the backlogged period of server 1, we conclude that

$$D_1(t) \geq D_1(t - t_1 - t_2 - t_3) = A_1(t - t_1 - t_2 - t_3).$$

We combine both inequalities and take the infimum to receive

$$\begin{aligned}
& D_1(t) \\
\geq & A_1(t - t_1 - t_2 - t_3) + \left[\sum_{j=1}^{3} \beta_j(t_j) - A_2\left(t - t_1 - t_2 - t_3, t - t_3\right) - A_3\left(t - t_2 - t_3, t\right) \right]^+ \\
\geq & A_1(t - t_1 - t_2 - t_3) + \left[\sum_{j=1}^{3} \beta_j(t_j) - \alpha_2\left(t_1 + t_2\right) - \alpha_3\left(t_2 + t_3\right) \right]^+ \\
\geq & \inf_{0 \leq s \leq t} \left\{ A_1(t-s) + \left[\inf_{\substack{t_1+t_2+t_3=s \\ t_1,t_2,t_3 \geq 0}} \left\{ \sum_{j=1}^{3} \beta_j(t_j) - \alpha_2\left(t_1 + t_2\right) - \alpha_3\left(t_2 + t_3\right) \right\} \right]^+ \right\} \\
= & \inf_{0 \leq s \leq t} \left\{ A_1(t-s) + \beta_{\text{e2e}}^{\text{PMOO}}(s) \right\} \\
= & A_1 \otimes \beta_{\text{e2e}}^{\text{PMOO}}\,(t),
\end{aligned}$$

where

$$\beta_{\text{e2e}}^{\text{PMOO}}(t) := \left[\inf_{\substack{t_1+t_2+t_3=t \\ t_1,t_2,t_3 \geq 0}} \left\{ \sum_{j=1}^{3} \beta_j(t_j) - \alpha_2\left(t_1 + t_2\right) - \alpha_3\left(t_2 + t_3\right) \right\} \right]^+$$

is the resulting leftover service curve. In the following, we call this type of analysis pay multiplexing only once (PMOO). This technique has been formalized with the so-called *multidimensional operator for PMOO* [Bou+08], [BBC18, pp. 236]. For the class of token bucket arrival curves and rate-latency servers, the multidimensional operator yields again rate-latency service curves [SZM06, SZM08].

State-of-the-art analysis in the DNC

One might assume that the PMOO analysis always leads to tighter delay bounds than the SFA. However, it has been shown that even in the simple two flows - two servers topology in Figure 2.5, the SFA can outperform the PMOO when service curves are not assumed to be equal (to be precise, if for the rates of the rate-latency service curves, it holds that $R_1 < R_2$) [SZF08]. Therefore, in [SZF08], an optimization-based algorithm is suggested and it was shown that the optimization-based analysis leads to tighter delay bounds. This has been further advanced to a tight end-to-end delay analysis in [BJT10, BT16]. Yet, both optimization-based analyses suffer from being computationally infeasible for larger networks. An algebraic alternative that maintains fast runtimes while getting close to optimization-based delay bounds has been reported in [Bon16, BNS17b, GB19] and is publicly available in the DISCO network calculator [SZ06, Gol+08, SBS18].

3 STOCHASTIC NETWORK CALCULUS BACKGROUND

Many applications, however, do not require *hard* performance guarantees. Some would still be considered tolerable as long as delayed arrivals are rare. See, for example, the work on Tactile Internet [Fet14], Industrial IoT [Boy+18], or Internet at the speed-of-light [Sin+14]. Therefore, in the stochastic network calculus (SNC), the focus is to study stochastic bounds of the form

$$\mathrm{P}(\text{delay} > T) \leq \varepsilon,$$

where $\varepsilon \geq 0$ is typically a very small quantity, e.g., $\in \left\{10^{-3}, 10^{-6}, 10^{-9}\right\}$. In this chapter, we review prior work on the SNC assuming the network model in Section 2.1.

3.1 STOCHASTIC ARRIVALS AND SERVICE

Stochastic arrival constraints

Based on the deterministic definition of an arrival curve (Definition 2.3), a stochastic version can be derived intuitively.

Definition 3.1 ([Cru96], [CBL06], [JL08, p. 43])**.** An increasing function $\alpha(t-s)$ is said to be a *stochastic envelope* or *stochastic arrival curve* for an arrival process A if, for all $\sigma \in \mathbb{R}$

$$\mathrm{P}(A(s,t) > \alpha(t-s) + \sigma) \leq \varepsilon_a(\sigma), \quad 0 \leq s \leq t, \tag{3.1}$$

where $\varepsilon_a(\sigma) \geq 0$ is a decreasing function, the *error function* or *overflow / deficit profile*.

We want to point out that A is a stochastic process, whereas α is a non-random function that only depends on the interval length $t-s$. If we choose $\sigma = 0$ and $\varepsilon_a(\sigma) = 0$, we recover the deterministic envelope. Probably the most prominent example of a stochastic envelope is exponentially bounded burstiness (EBB) [YS93] with $\alpha(t-s) = \rho \cdot (t-s)$ and $\varepsilon_a(\sigma) = Me^{-\theta\sigma}$, where $\rho, M, \theta \geq 0$. Other approaches to define stochastic arrival constraints can be found in [MS97, SS00, B00+00, FMN00, Yin+02, BLP06, LBL07, JL08, CS12, Riz13]. At this point, we would like to highlight that even heavy-tailed distributions, a traffic class that comes with particular difficulties in a performance analysis, can be modeled [GMOB00, KH01, LGOBM05, LBC12].

A different branch of the SNC is based on moment-generating functions (MGFs). The MGF of an arrival process $A(t)$ is given by

$$\phi_{A(t)}(\theta) := \mathrm{E}\left[e^{\theta A(t)}\right],$$

where $\theta \geq 0$. In accordance to the affine arrival curve Eqn. (2.14), we define the MGF-based class of (σ_A, ρ_A)-constrained arrival processes introduced by Chang ([Cha94], [Cha00, p. 241]).

Definition 3.2 ((σ_A, ρ_A)-constrained Arrivals)**.** An arrival process $A(s,t)$ is (σ_A, ρ_A)-bounded if

$$\mathrm{E}\left[e^{\theta A(s,t)}\right] \leq e^{\theta \rho_A(\theta) \cdot (t-s) + \theta \sigma_A(\theta)}, \quad \text{for all } 0 \leq s \leq t. \tag{3.2}$$

The MGF-based SNC originates in the *effective bandwidth* theory [Kel96, Cha00], where a bound is derived for $\log\left(\mathrm{E}\left[e^{\theta A(t)}\right]\right)/(\theta t)$. The connection of SNC with MGF to Eqn. (3.1) is given by the *Chernoff bound* [Ros96, p. 39]:

$$\mathrm{P}(X \geq a) \leq e^{-\theta a}\,\mathrm{E}\left[e^{\theta X}\right], \tag{3.3}$$

where $\theta \geq 0$. Let us now choose $\alpha(t-s) = \rho_A(\theta) \cdot (t-s)$ in the stochastic envelope (Eqn. (3.1)). Applying Eqn. (3.3) to (σ_A, ρ_A)-constrained arrivals yields

$$\varepsilon_a(\sigma) = e^{\theta \sigma_A(\theta)} e^{-\theta \sigma}. \tag{3.4}$$

Thus, one can state that (σ_A, ρ_A)-constrained arrivals are EBB for $M = e^{\theta \sigma_A(\theta)}$ [RF11, FR15]. One can also show the opposite direction, namely that an EBB process is (σ_A, ρ_A)-constrained. Proofs can be found, e.g., in [ZTK95, Lemma 5] or [BH17].

Let us now visualize the strength of a stochastic envelope in terms of probabilistic guarantees. We consider an aggregate of m flows with an arrival process consisting of stationary independent and identically distributed (iid) Bernoulli distributed increments with parameter p. To be precise, for $0 \leq s \leq t$, we have $A(s,t) = \sum_{i=1}^{m} A_i(s,t)$ with $A_i(s,t) \sim B(t-s,p), i = 1, \ldots, m$, where $B(n,p)$ denotes the Binomial distribution with parameters n and p.

This arrival process is (σ_A, ρ_A)-bounded for $\theta > 0$:

$$\begin{aligned}\mathrm{E}\left[e^{\theta A_i(s,t)}\right] &= \left(1 - p + pe^{\theta}\right)^{t-s} \\ &= e^{\theta \rho_A(\theta) \cdot (t-s) + \theta \sigma_A(\theta)}, \quad i = 1, \ldots, m,\end{aligned}$$

where

$$\begin{aligned}\sigma_A(\theta) &= 0, \\ \rho_A(\theta) &= \frac{\log\left(1 - p + pe^{\theta}\right)}{\theta}.\end{aligned}$$

Different stochastic envelopes are depicted in Figure 3.1. We observe that, in contrast to the peak rate, which gives us a deterministic guarantee, the (σ_A, ρ_A)-envelope is able to exploit statistical multiplexing. Yet, even though we use it to bound small violation probabilities ($\varepsilon_a(b) = 10^{-6}$ in the plot), it still deviates significantly from the peak rate.

A large set of traffic classes can be (σ_A, ρ_A)-bounded. In the following, we list a few examples.

- Discrete-time iid increments [Cha00, p. 243]: Let $A(t) = \sum_{i=1}^{t} a_i$ be a discrete-time process such that all $a_i, i = 1, \ldots, t$ are iid witch existing MGF $\phi_A(\theta) = \mathrm{E}\left[e^{\theta a_1}\right]$. Then, A is (σ_A, ρ_A)-constrained with

$$\begin{aligned}\sigma_A(\theta) &= 0, \\ \rho_A(\theta) &= \frac{1}{\theta} \log\left(\phi_A(\theta)\right).\end{aligned} \tag{3.5}$$

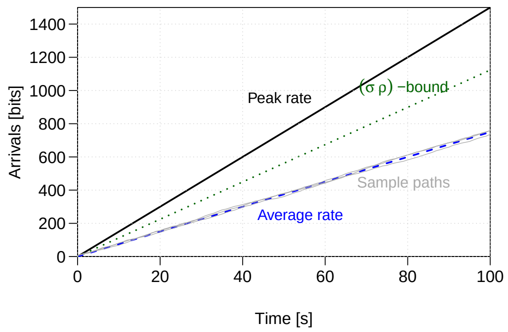


Figure 3.1: Stochastic envelopes for iid Bernoulli increments.

For example, for the case of exponential iid increments [Bec16a] with parameter λ, we have that

$$\rho_A(\theta) = \frac{1}{\theta} \log\left(\frac{\lambda}{\lambda - \theta}\right), \quad \theta < \lambda. \tag{3.6}$$

More generally, for independent, Gamma-distributed increments with shape parameter $\alpha > 0$ and rate parameter $\beta > 0$, it holds that

$$\rho_A(\theta) = \frac{\alpha}{\theta} \log\left(\frac{\beta}{\beta - \theta}\right), \quad \theta < \beta. \tag{3.7}$$

Another example are Weibull distributed increments [CP18] with shape parameter k and scale parameter λ. It does not have an existing MGF for all k, yet, e.g., for the special case of $k = 2$ (also known as the Rayleigh distribution), the arrival process is (σ_A, ρ_A)-constrained with

$$\rho_A(\theta) = \frac{1}{\theta} \log\left(1 + b\theta e^{\frac{(b\theta)^2}{2}} \sqrt{\frac{\pi}{2}} \left(\operatorname{erf}\left(\frac{b\theta}{\sqrt{2}}\right) + 1\right)\right), \tag{3.8}$$

where $\operatorname{erf}(\cdot)$ denotes the Gauss error function and $b := \frac{\lambda}{\sqrt{2}}$. The last iid example we present are Poisson distributed increments with parameter λ. It holds that

$$\rho_A(\theta) = \frac{1}{\theta} \lambda \left(e^{\theta} - 1\right). \tag{3.9}$$

- Discrete-time Markov-modulated process (MMP): Let $x(t)$ be a discrete-time homogeneous Markov chain on the states $\{1, \ldots, M\}$ with the transition matrix P, i.e., p_{ij} is the transition probability from state i to state j. Also let $\{y_i(t), t = 1, 2, \ldots\}, i = 1, \ldots, M$, be M sequences of iid random variables with moment-generating function

$$\phi_i(\theta) = \mathrm{E}\left[e^{\theta y_i(1)}\right].$$

The process $a(t) = y_{x(t)}(t)$ is then an MMP with modulating process $x(t)$. $a(t)$ is stationary if $x(t)$ is stationary. Let π_i be the probability of $x(1)$ being in state i and also let a row vector be

$$\boldsymbol{\pi} = (\pi_1, \ldots, \pi_M).$$

Then, the MGF of $A(t) = \sum_{i=0}^{t} a_i$ is [Cha94], [Cha00, pp. 244]

$$\mathrm{E}\left[e^{\theta A(t)}\right] = \boldsymbol{\pi} \left(\phi(\theta) P\right)^{t-1} \phi(\theta) \mathbf{1}^T$$

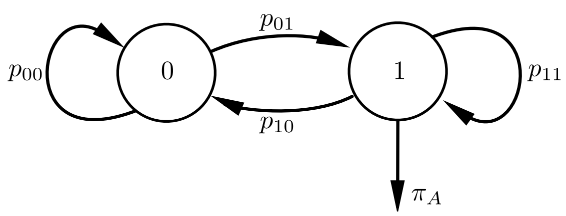


Figure 3.2: Discrete-time MMOO model

where

$$\phi(\theta) := \operatorname{diag}\left(\phi_1(\theta), \dots, \phi_M(\theta)\right) := \begin{pmatrix} \phi_1(\theta) & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \phi_M(\theta) \end{pmatrix}$$

and

$$\mathbf{1} = (1, \dots, 1).$$

For this expression, we can derive a closed-form upper bound [Bec16a, pp. 124]:

$$\boldsymbol{\pi}\left(\phi(\theta)P\right)^{t-1}\phi(\theta)\mathbf{1}^T \leq \left(\max_{k=1,\dots,M}\phi_k(\theta)\right)\sum_{j=1}^{M}\pi_j\frac{\max_{k=1,\dots,M}\{v_k\}}{\min_{k=1,\dots,M}\{v_k\}}\operatorname{sp}(\phi(\theta)P)^{t-1},$$

where $\operatorname{sp}(\cdot)$ denotes the spectral radius of a a matrix (basically the largest eigenvalue) and $v \in \mathbb{R}_+^M$ is a corresponding eigenvector with only positive entries. This yields a (σ_A, ρ_A)-bound

$$\begin{aligned} \sigma_A(\theta) &= \frac{1}{\theta}\log\left(\left(\max_{k=1,\dots,M}\phi_k(\theta)\right)\frac{\max_{k=1,\dots,M}\{v_k\}}{\min_{k=1,\dots,M}\{v_k\}}\cdot\frac{1}{\operatorname{sp}(\phi(\theta)P)}\right), \\ \rho_A(\theta) &= \frac{1}{\theta}\log\left(\operatorname{sp}(\phi(\theta)P)\right). \end{aligned} \tag{3.10}$$

The special case of a two state Markov chain $x(t) \in \{1, 2\}$ with $y_1(t) = 0$ and *peak rate* $y_2(t) = \pi_A$, is called Markov-modulated On-Off (MMOO) arrivals [Soh92] (see Figure 3.2). Then, the spectral radius of $\phi(\theta)P$ is

$$\operatorname{sp}(\phi(\theta)P) = \frac{p_{11} + p_{22}e^{\theta\pi_A} + \sqrt{\left(p_{11} + p_{22}e^{\theta\pi_A}\right)^2 - 4\left(p_{11} + p_{22} - 1\right)e^{\theta\pi_A}}}{2}.$$

For details on the $\sigma_A(\theta)$, we refer the interested reader to Appendix A.1.

- Continuous-time Markov-modulated On-Off Arrivals [AMS82, CW96, CPS14]: Let $x(t)$ be a random process of the form $x(t) = z(t) \cdot b$, the so-called fluid rate at t, where $\pi_A > 0$ is the peak rate and z_t is a continuous-time Markov process. $z(t)$ has two states, 0 and 1 with transition rates μ and λ (see Figure 3.3). Then, we define for all $0 \leq s \leq t$ the according arrival process by

$$A(s, t) \overset{(2.2)}{=} \int_s^t x(u)\,\mathrm{d}u = \int_s^t z(u)\pi_A\,\mathrm{d}u.$$

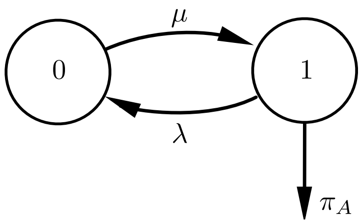


Figure 3.3: Continuous-time MMOO model

Then, $A(t)$ is (σ_A, ρ_A)-constrained with

$$\begin{aligned}\sigma_A(\theta) &= 0,\\ \rho_A(\theta) &= \frac{\theta\pi_A - \mu - \lambda}{2\theta} + \frac{1}{2\theta}\sqrt{(\theta\pi_A - \mu - \lambda)^2 + 4\mu\theta\pi_A}.\end{aligned} \tag{3.11}$$

There are also arrival processes with existing MGF, that are not (σ_A, ρ_A)-constrained. One particularly important process of that kind is the fractional Brownian motion (fBm) arrival model [Nor94, Nor95, Kel96, LBL07, RF12a, Riz13, FR15]. It is defined as

$$A(s,t) := \lambda \cdot (t-s) + \sigma \mathrm{Z}(t-s)\,, \tag{3.12}$$

where λ is the mean arrival rate, σ^2 is the variance of $A(1)$ and $\mathrm{Z}(t)$ is a normalized fBm (see also Definition A.1 in Appendix A.2). Then, its MGF is equal to [Kel96]

$$\mathrm{E}\left[e^{\theta A(s,t)}\right] = e^{\theta\lambda(t-s) + \frac{\theta^2\sigma^2}{2}(t-s)^{2H}}. \tag{3.13}$$

Stochastic service guarantees

In the stochastic analysis, we mostly employ a *bivariate* version to describe the provided service. Mathematically, we define it as follows. A *service process* $S(s,t) \geq 0,\ 0 \leq s \leq t$, is a nonnegative stochastic process such that $S(t,t) = 0$ for all $t \geq 0$. This service process is also known in the literature as *time-varying capacity*. Similar to the DNC, we introduce the (bivariate) (de)convolution to facilitate the expression of operations and performance bounds.

Definition 3.3 (Min-Plus Convolution)**.** Let x and y be nonnegative functions. The (bivariate) *min-plus convolution* of x and y is defined for $0 \leq s \leq t$ as

$$x \otimes y\,(s,t) := \inf_{s \leq \tau \leq t} \{x(s,\tau) + y(\tau,t)\}\,. \tag{3.14}$$

The (bivariate) *min-plus deconvolution* of x and y is defined for $0 \leq s \leq t$ as

$$x \oslash y\,(s,t) := \sup_{0 \leq \tau \leq s} \{x(\tau,t) - y(\tau,s)\}\,. \tag{3.15}$$

Now, we can introduce the bivariate version of service curves. By abuse of notation, we denote the server as well as the offered service by S.

Definition 3.4 (Dynamic Server)**.** A service process S is called a *dynamic server* [CC99], [Cha00, p. 178], [Cha+02] for arrivals $A(t)$, if it satisfies for its respective departures $D(t)$ that

$$D(t) \geq A \otimes S\,(0,t) = \inf_{0 \leq \tau \leq t} \{A(0,\tau) + S(\tau,t)\} \tag{3.16}$$

for any fixed sample path.

As for the deterministic analysis, we need a stronger condition when considering multiple flows at a server.

Definition 3.5 (Work-conserving Server [Fido6])**.** For any $t \geq 0$, let

$$s := \sup\{\tau \in [0,t] : D(\tau) = A(\tau)\}$$

be the start of the backlogged period (Eqn. (2.15)) before t. Let $\tau \in [s,t]$. A *work-conserving server* is a service process $S(\tau,t)$ such that, for any fixed sample path, the server S is non-idling and uses the entire available service $S(\tau,t)$, i.e.,

$$D(t) = D(\tau) + S(\tau,t) \quad \forall s \leq \tau \leq t \tag{3.17}$$

in any continuously backlogged period $(\tau,t]$.

Again, one can easily show that any work-conserving server is also a dynamic server.

Definition 3.6 ((σ_S,ρ_S)-constrained Service)**.** We call a dynamic server (σ_S,ρ_S)-constrained if it holds for all $0 \leq s \leq t$ that

$$\mathrm{E}\left[e^{-\theta S(s,t)}\right] \leq e^{-\theta\rho_S(-\theta)(t-s)+\theta\sigma_S(-\theta)}. \tag{3.18}$$

For example, a dynamic server with constant rate $C \geq 0$ is (σ_S,ρ_S)-constrained with $\sigma_S(-\theta) = 0$ and $\rho_S(-\theta) = C$.

3.2 STOCHASTIC SINGLE-NODE PERFORMANCE BOUNDS

In this section, we present the SNC performance bounds. Therefore, we start with the bivariate sample-path bounds.

Theorem 3.7 (Sample-Path Bounds: Backlog, Delay, and Output)**.** *Consider a dynamic server $S(s,t)$ with arrival process $A(s,t)$.*

1. *The backlog at time $t \geq 0$ is upper bounded by*

$$q(t) \leq A \oslash S\,(t,t). \tag{3.19}$$

2. *The virtual delay at time $t \geq 0$ is upper bounded by*

$$d(t) \leq \inf\{s \geq 0 : A \oslash S\,(t+s,t) \leq 0\}. \tag{3.20}$$

3. *The departure process $D(s,t)$ is upper bounded for any $0 \leq s \leq t$ by*

$$D(s,t) \leq A \oslash S\,(s,t). \tag{3.21}$$

A proof can be found in [Fido6]. Note that the deconvolution for the delay extends the domain of the deconvolution in Definition 3.3 (first argument larger than second). We define it as

$$A \oslash S\,(t+s,t) := \sup_{0\leq\tau\leq t}\{A(\tau,t) - S(\tau,t+s)\}.$$

Making use of Theorem 3.7, we can now derive stochastic performance bounds. But before, let us formulate the generalized Hölder inequality that we frequently use to bound the expected value of products of dependent random variables.

Theorem 3.8 (Generalized Hölder Inequality). *Let $p_1, \ldots, p_n > 1$ be real numbers and Hölder conjugates, i.e., $\sum_{i=1}^{n} \frac{1}{p_i} = 1$. Let $X_1, \ldots, X_n \geq 0$ be such that $X_i \in L^{p_i}, i = 1, \ldots n$. Then, $\prod_{i=1}^{n} X_i \in L^1$ and*

$$\mathrm{E}\left[\prod_{i=1}^{n} X_i\right] \leq \prod_{i=1}^{n} \mathrm{E}\left[X_i^{p_i}\right]^{\frac{1}{p_i}}. \tag{3.22}$$

Proof. A proof is given in [Che01]. □

Theorem 3.9 (Violation Probability of Backlog and Delay). *Let $\theta > 0$ and $p, q > 1$ be Hölder conjugates, that is, $\frac{1}{p} + \frac{1}{q} = 1$. Suppose we have (σ_A, ρ_A)-bounded arrivals and a (σ_S, ρ_S)-bounded dynamic server. Further, we assume the* stability condition

$$\rho_A(p\theta) < \rho_S(-q\theta). \tag{3.23}$$

1. *Let $B \geq 0$. For the backlog, it holds for all $t \geq 0$ that*

$$\mathrm{P}(q(t) > B) \leq e^{-\theta B} \sum_{\tau=0}^{t-1} \left(\mathrm{E}\left[e^{p\theta A(\tau,t)}\right]\right)^{\frac{1}{p}} \left(\mathrm{E}\left[e^{-q\theta S(\tau,t)}\right]\right)^{\frac{1}{q}} \tag{3.24}$$

$$\leq e^{-\theta B} \frac{e^{\theta(\sigma_A(p\theta)+\sigma_S(-q\theta))}}{\theta\left(\rho_S(-q\theta) - \rho_A(p\theta)\right)}. \tag{3.25}$$

2. *Let $T \geq 0$. For the virtual delay, it holds for all $t \geq 0$ that*

$$\mathrm{P}(d(t) > T) \leq \sum_{\tau=0}^{t-1} \left(\mathrm{E}\left[e^{p\theta A(\tau,t)}\right]\right)^{\frac{1}{p}} \left(\mathrm{E}\left[e^{-q\theta S(\tau,t+T)}\right]\right)^{\frac{1}{q}} \tag{3.26}$$

$$\leq e^{-\theta\rho_S(-q\theta)T} \frac{e^{\theta(\sigma_A(p\theta)+\sigma_S(-q\theta))}}{\theta\left(\rho_S(-q\theta) - \rho_A(p\theta)\right)}. \tag{3.27}$$

Note that, the first (time-dependent) inequality for backlog and delay can still be used if the processes are not (σ, ρ)-bounded or if stability cannot be assumed. In addition, if A and S are independent, the bounds in Theorem 3.9 can be improved by setting $p = q = 1$ (which are obviously not Hölder conjugates). Proofs for the independent case can be found, e.g., in the seminal work by Chang ([Cha00, pp. 248]). However, since the applied techniques are key throughout this thesis, we show the proof of the delay bound. One important inequality that is typically used in the SNC is *Boole's inequality / the Union bound*.

Theorem 3.10 (Union Bound / Boole's Inequality). *Let $X_1, \ldots, X_n$ be random variables and $x \in \mathbb{R}$. Then holds the* Union Bound / Boole's Inequality*:*

$$\mathrm{P}\left(\max_{i=1,\ldots,n} X_i > x\right) \leq \sum_{i=1}^{n} \mathrm{P}(X_i > x). \tag{3.28}$$

Proof. See, e.g, [Ros10, p. 57] or [KMT11, pp. 295]. □

Proof of Theorem 3.9. By Eqn (3.20), we know that if there exists a $T \geq 0$ such that
$\sup_{0\leq\tau\leq t} \{A(\tau, t) - S(\tau, t + T)\} \leq 0$, then $d(t) \leq T$. Vice versa,

$$d(t) > T \Rightarrow \sup_{0\leq\tau\leq t} \{A(\tau, t) - S(\tau, t + T)\} > 0 \quad \text{for } T \geq 0. \tag{3.29}$$

Furthermore, we have $\mathrm{P}(A(t,t)-S(t,t+T)>0)=0$. Hence,

$$
\begin{aligned}
\mathrm{P}(d(t)>T) &\overset{(3.29)}{\leq} \mathrm{P}\left(\sup_{0\leq\tau\leq t-1}\{A(\tau,t)-S(\tau,t+T)\}>0\right) \\
&\overset{(3.28)}{\leq} \sum_{\tau=0}^{t-1}\mathrm{P}(A(\tau,t)-S(\tau,t+T)>0) \\
&\overset{(3.3)}{\leq} \sum_{\tau=0}^{t-1}\mathrm{E}\left[e^{\theta(A(\tau,t)-S(\tau,t+T))}\right] \\
&\overset{(3.22)}{\leq} \sum_{\tau=0}^{t-1}\left(\mathrm{E}\left[e^{p\theta A(\tau,t)}\right]\right)^{\frac{1}{p}}\left(\mathrm{E}\left[e^{-q\theta S(\tau,t+T)}\right]\right)^{\frac{1}{q}} \\
&\leq \sum_{\tau=0}^{t-1} e^{\theta(\rho_A(p\theta)(t-\tau)+\sigma_A(p\theta))}e^{-\theta(\rho_S(-q\theta)(t+T-\tau)-\sigma_S(-q\theta))} \\
&= \underbrace{e^{-\theta\rho_S(-q\theta)T}e^{\theta(\sigma_A(p\theta)+\sigma_S(-q\theta))}}_{=:b}\sum_{\tau=0}^{t-1}e^{\theta(\rho_A(p\theta)-\rho_S(-q\theta))(t-\tau)} \\
&= b\cdot\sum_{j=1}^{t}e^{\theta(\rho_A(p\theta)-\rho_S(-q\theta))j}.
\end{aligned}
$$

Here, we used the Union bound in the second line and the Chernoff bound in the third line. We substitute $t-\tau$ by j in the last line. It follows that

$$
\begin{aligned}
\mathrm{P}(d(t)>T) &\leq b\cdot\sum_{j=1}^{\infty}e^{\theta(\rho_A(p\theta)-\rho_S(-q\theta))j} \\
&\leq b\cdot\int_0^{\infty}e^{\theta(\rho_A(p\theta)-\rho_S(-q\theta))x}\mathrm{d}x \\
&\overset{(3.23)}{=} b\cdot\frac{1}{\theta\left(\rho_S(-q\theta)-\rho_A(p\theta)\right)} \\
&= e^{-\theta\rho_S(-q\theta)T}\frac{e^{\theta(\sigma_A(p\theta)+\sigma_S(-q\theta))}}{\theta\left(\rho_S(-q\theta)-\rho_A(p\theta)\right)}.
\end{aligned}
$$

where we used the stability condition in the third line to ensure convergence. This concludes the proof. □

The bounds in Theorem 3.9 assume discrete-time processes. If A is a continuous-time process, discretization techniques as in [CBL06] can be applied. For the respective performance bounds, see, e.g., [Bec16a, pp. 30].

3.3 END-TO-END ANALYSIS AND OPEN PROBLEMS

Similar to the deterministic analysis, we have results for multiple flows at a server, an output bound, as well as the multi-node case. We present the state of the art in the following. Note that, if arrivals and service are assumed to be (σ,ρ)-bounded, all network operations can be shown to be also in this class, see also [BS13, ZBHB16, BH17].

We start off with the multi-flow case.

Theorem 3.11 (Leftover Service under Arbitrary Multiplexing)**.** *Consider two flows, f_1 and f_2, with respective arrival processes $A_1(s,t)$ and $A_2(s,t)$, that receive*

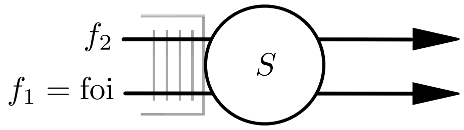


Figure 3.4: Two flows at one node with service process S.

service process $S(s,t)$ at a work-conserving server as in Figure 3.4. Further, we assume f_1 to be our flow of interest (foi) and the scheduling to be arbitrary multiplexing. *Then, the foi sees the dynamic server*

$$S^1_{\text{l.o.}}(s,t) := S^1(s,t) := [S(s,t) - A_2(s,t)]^+, \quad 0 \le s \le t, \tag{3.30}$$

the so-called leftover service. This is also denoted by $S^1 = S \ominus A_2$.

Let $\theta > 0$ and $p, q > 1$ be Hölder conjugates. If we further assume the arrival process A_2 and the service process to be $(\sigma_{A_2}, \rho_{A_2})$ and (σ_S, ρ_S)-bounded, respectively, then the leftover service is $(\sigma_{S^1}, \rho_{S^1})$-bounded with

$$\begin{aligned}\sigma_{S^1}(-\theta) &= \sigma_S(-q\theta) + \sigma_{A_2}(p\theta),\\ \rho_{S^1}(-\theta) &= \rho_S(-q\theta) - \rho_{A_2}(p\theta).\end{aligned}$$

Proof. See [Chao00, pp. 179] or [Fid06] for the dynamic server, and [BS13] for the $(\sigma_{S^1}, \rho_{S^1})$-bound. □

As for the performance bounds, if A and S are independent, then Theorem 3.11 can be improved by setting $p = q = 1$. Moreover, the result can easily be generalized to $n-1$ cross flows.

Apart from being a bivariate result, there is a fundamental difference from Theorem 2.7, namely that it directly takes the arrival process A_2 rather than an arrival curve α_2. Hence, for stochastic arrivals, the leftover service is a stochastic process, even if S was assumed to be a deterministic process.

Next, we present a bound for the departures of a flow. Note that, since we work with moment-generating functions, instead of bounding the output burstiness, we need a bound on the MGF of the departures.

Theorem 3.12 (MGF-Output Bound)**.** *Let $\theta > 0$ and $p, q > 1$ be Hölder conjugates. Suppose we have (σ_A, ρ_A)-bounded arrivals and a (σ_S, ρ_S)-bounded dynamic server. Further, we assume the stability condition in Eqn. (3.23). For the MGF of the output, it holds for all $0 \le s \le t$ that*

$$\begin{aligned}\mathrm{E}\left[e^{\theta D(s,t)}\right] &\le \sum_{\tau=0}^{s} \left(\mathrm{E}\left[e^{p\theta A(\tau,t)}\right]\right)^{\frac{1}{p}} \left(\mathrm{E}\left[e^{-q\theta S(\tau,s)}\right]\right)^{\frac{1}{q}}\\ &\le e^{\theta \rho_A(p\theta)(t-s)} \frac{e^{\theta(\sigma_A(p\theta)+\sigma_S(-q\theta))}}{1 - e^{\theta(\rho_A(p\theta)-\rho_S(-q\theta))}}.\end{aligned} \tag{3.31}$$

Further, the output is (σ_D, ρ_D)-bounded with

$$\begin{aligned}\sigma_D(\theta) &= \sigma_A(p\theta) + \sigma_S(-q\theta) + \frac{1}{\theta}\log\left(\frac{1}{1 - e^{\theta(\rho_A(p\theta)-\rho_S(-q\theta))}}\right),\\ \rho_D(\theta) &= \rho_A(p\theta).\end{aligned}$$

Proof. See [Bec16a, pp. 29]. □

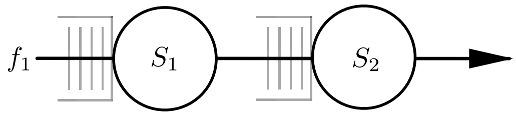


Figure 3.5: One flow - two servers

Again, if A and S are independent, then Theorem 3.12 can be improved by setting $p = q = 1$. For the continuous-time version, see again [Bec16a, pp. 29].

As the last piece of this section, we state the result for the multi-node case, the Concatenation Theorem.

Theorem 3.13 (Concatenation Theorem). *Assume the topology in Figure 3.5, where the arrivals of f_1 are $A_{\text{e2e}} = A_1$. We denote by $D_1 = A_2$ the output of S_1, which equals the input at S_2, and by $D_{\text{e2e}} = D_2$ the departures of S_2. Further, we assume the servers to be dynamic servers. Then, the end-to-end service, S_{e2e}, is a dynamic server. Let $\theta > 0$ and let $p, q > 1$ be Hölder conjugates. Under the additional assumption of $(\sigma_{S_1}, \rho_{S_1})$ and $(\sigma_{S_2}, \rho_{S_2})$-constrained servers with $\rho_{S_1}(-p\theta) \neq \rho_{S_2}(-q\theta)$, the end-to-end service is $(\sigma_{S_{\text{e2e}}}, \rho_{S_{\text{e2e}}})$-bounded, where*

$$\sigma_{S_{\text{e2e}}}(-\theta) = \sigma_{S_1}(-p\theta) + \sigma_{S_2}(-q\theta) + K,$$
$$\rho_{S_{\text{e2e}}}(-\theta) = \min\left\{\rho_{S_1}(-p\theta), \rho_{S_2}(-q\theta)\right\},$$

where

$$K := \frac{1}{\theta} \log\left(\frac{1}{1 - e^{-\theta\left|\rho_{S_2}(-q\theta) - \rho_{S_1}(-p\theta)\right|}}\right).$$

Proof. See [Bec16a, p. 28]. □

Theorem 3.13 shows that that the convolution operation $\otimes$, similar to the other operations, is closed under all (σ, ρ)-constrained processes. As a consequence, we can conduct a sequential end-to-end performance analysis by, e.g., implementing the separated flow analysis (SFA) known from the DNC (Section 2.5). In the following, we call this the sequential separated flow analysis (seqSFA). However, it does not scale well for larger networks as $\sigma_{S_{\text{e2e}}}(-\theta)$ increases by K for each convolution. This problem can be avoided when, in contrast to a sequential order, the deconvolution for the performance bounds and the convolution of the service processes is calculated in one step. The idea of avoiding this sequencing property is first reported in [Fid06, Theorem 3] and further discussed in [Bec16a, pp. 34]. In the following, we state the result for the delay for heterogeneous servers without cross-traffic which is less accurate compared to the homogeneous version in [Fid06].

Theorem 3.14 (End-to-End Delay Bound). *Assume the topology in Figure 3.6, where the (σ_A, ρ_A)-bounded arrivals of f_1 are $A_{\text{e2e}} = A_1$. Further, we assume all servers, S_j, $j = 1, \ldots, n$, to be dynamic servers for flow f_1 and $\left(\sigma_{S_j}, \rho_{S_j}\right)$-bounded. Let $\theta > 0$ and let p_j, $j = 1, \ldots, n$ be Hölder conjugates, i.e., all $p_j > 1$ and $\sum_{j=1}^{n} \frac{1}{p_j} = 1$. Let us assume the stability condition*

$$\rho_A(\theta) < \rho_{S_j}(-p_j\theta), \quad j = 1, \ldots, n.$$

Then, bounds on the delay's violation probability are given by

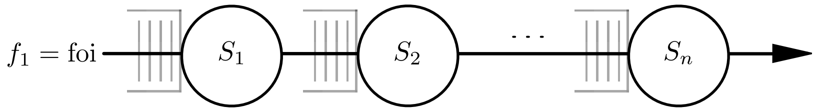


Figure 3.6: One flow - n servers

1.
$$\mathrm{P}(d(t) > T) \le e^{-\theta\rho_A(\theta)\cdot T} e^{\theta\sigma_A(\theta)}\gamma,$$
where
$$\gamma := \prod_{j=1}^{n} \frac{e^{\theta\sigma_{S_j}(-p_j\theta)}}{1 - e^{\theta\left(\rho_A(\theta) - \rho_{S_j}(-p_j\theta)\right)}}.$$

2.
$$e^{-\theta \min_{j=1,\dots,n}\left\{\rho_{S_j}(-p_j\theta)\right\}\cdot T} e^{\theta\sigma_A(\theta)} e^{\theta\sum_{j=1}^{n}\sigma_{S_j}(-p_j\theta)}\zeta^n,$$
where
$$\zeta := \frac{\left(1 + \frac{T}{n}\right)^{\left(1+\frac{T}{n}\right)}}{\left(\frac{T}{n}\right)^{\frac{T}{n}}}.$$

As discussed above, Theorem 3.14 gives us an alternative to seqSFA. In order to have a clear separation, we call it SFA when applying Theorem 3.14 to calculate performance bounds.

Let us now assume that all arrivals and servers are independent. If we fix the delay violation probability to ε and solve for T, we obtain the stochastic delay bounds

1.
$$T = \frac{\theta\left(\sigma_A(\theta) + \sum_{j=1}^{n}\sigma_{S_j}(\theta)\right) + \log\left(\frac{1}{\varepsilon}\right) + \log(\gamma)}{\theta\rho_A(\theta)},$$

2.
$$T = \frac{\theta\left(\sigma_A(\theta) + \sum_{j=1}^{n}\sigma_{S_j}(\theta)\right) + \log\left(\frac{1}{\varepsilon}\right) + n\cdot\log(\zeta)}{\theta\min_{j=1,\dots,n}\left\{\rho_{S_j}(\theta)\right\}}.$$

In other words, under the independence assumption, both stochastic delay bounds scale linearly in the number of traversed servers ($\mathcal{O}(n)$) [Fid06]. Without assuming independence, it has been shown for the class of EBB traffic using the so-called network service curve (NSC), that delay bounds scale in $\Theta(n\log(n))$ [CBL06, BLC07, BLC11].

For a numerical comparison of (σ,ρ)-constrained arrivals and EBB, we refer the interested reader to [RF11, RF12b].

In fact, one can also consider the case $\rho_{S_1}(-p\theta) = \rho_{S_2}(-q\theta)$ in a sequential analysis. The idea we make use of is inspired by the rate reduction technique in [CBL05, CBL06].

Theorem 3.15 (Concatenation Theorem, pt. 2)**.** *Assume the scenario in Theorem 3.13. Let $\theta, \delta > 0$ and let $p, q > 1$ be Hölder conjugates. Under the additional assumption of $(\sigma_{S_1}, \rho_{S_1})$ and $(\sigma_{S_2}, \rho_{S_2})$-constrained servers with*

$$\delta < \rho_{S_1}(-p\theta) = \rho_{S_2}(-q\theta) \tag{3.32}$$

the end-to-end service is $(\sigma_{S_{\text{e2e}}}, \rho_{S_{\text{e2e}}})$*-bounded, where*

$$\sigma_{S_{\text{e2e}}}(-\theta) = \sigma_{S_1}(-p\theta) + \sigma_{S_2}(-q\theta) + V,$$
$$\rho_{S_{\text{e2e}}}(-\theta) = \rho_{S_2}(-q\theta) - \delta,$$

where

$$V := \frac{1}{\theta} \log\left(\frac{1}{1 - e^{-\theta\delta}}\right).$$

Proof. We bound the Laplace transform of $S_1 \otimes S_2\,(s,t)$ by

$$\begin{aligned}
&\mathrm{E}\left[e^{-\theta S_1 \otimes S_2\,(s,t)}\right] \\
&\leq \sum_{\tau=s}^{t} \mathrm{E}\left[e^{-\theta S_1(s,\tau) - \theta S_2(\tau,t)}\right] \\
&\leq \sum_{\tau=s}^{t} \left(\mathrm{E}\left[e^{-p\theta S_1(s,\tau)}\right]^{\frac{1}{p}}\right) \left(\mathrm{E}\left[e^{-q\theta S_2(\tau,t)}\right]\right)^{\frac{1}{q}} \\
&\leq \sum_{\tau=s}^{t} e^{-\theta\left(\rho_{S_1}(-p\theta)(\tau-s) - \theta\sigma_{S_2}(-p\theta)\right)} e^{-\theta\left(\left(\rho_{S_2}(-q\theta)\right)(t-\tau) - \theta\sigma_{S_2}(-q\theta)\right)} \\
&\leq \sum_{\tau=s}^{t} e^{-\theta\left(\rho_{S_1}(-p\theta)(\tau-s) - \theta\sigma_{S_2}(-p\theta)\right)} e^{-\theta\left(\left(\rho_{S_2}(-q\theta) - \delta\right)(t-\tau) - \theta\sigma_{S_2}(-q\theta)\right)} \\
&\overset{(3.32)}{=} e^{-\theta\left(\left(\rho_{S_2}(-q\theta) - \delta\right)(t-s) - \sigma_{S_1}(-p\theta) - \sigma_{S_2}(-q\theta)\right)} \cdot \sum_{j=0}^{t-s} e^{-\theta\delta j} \\
&\leq e^{-\theta\left(\left(\rho_{S_2}(-q\theta) - \delta\right)(t-s) - \sigma_{S_1}(-p\theta) - \sigma_{S_2}(-q\theta)\right)} \frac{1}{1 - e^{-\theta\delta}} \\
&= e^{-\theta\left(\rho_{S_2}(-q\theta) - \delta\right)(t-s)} e^{\theta\left(\sigma_{S_1}(-p\theta) + \sigma_{S_2}(-q\theta) - \frac{1}{\theta}\log\left(1 - e^{-\theta\delta}\right)\right)}.
\end{aligned}$$

This finishes the proof. □

Part II

DEALING WITH DEPENDENCE

4 DEALING WITH DEPENDENCE USING PMOO FOR TANDEM QUEUES AND SINK TREES

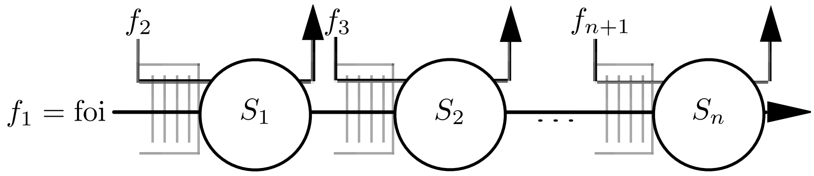


Figure 4.1: Canonical tandem

Results presented in this chapter are joint work with Jens Schmitt [NS17, NS20b].

In Section 3.3, we discussed that delay bounds scale in $\mathcal{O}(n)$ under independence [Fid06]. The technique to prove this result is based on the separated flow analysis (SFA) (see also Section 2.5). In other words, instead of summing stochastic per-hop delay bounds (which scales in $\mathcal{O}(n^3)$), we convolve service processes and benefit from the pay bursts only once (PBOO) property [CBL06]. This result has been derived for the canonical tandem (Figure 4.1) under arbitrary multiplexing. However, if we directly applied the SFA to a tandem queue [Bur64, LH08] as in Figure 4.2, the flows would become dependent in the analysis, even if they were assumed to be independent when entering the network. In order to continue the analysis, one typically applies Hölder's inequality (Eqn. (3.22)). A similar observation with cross-flows becoming dependent can be made for sink trees. Sink trees are interesting for various application scenarios and have been subject to SNC-based analyses:

- In Multiprotocol Label Switching (MPLS) networks, The option to set up multipoint-to-point label-switched paths between several ingress edge routers and one egress edge router [RVC01] creates a sink tree.
- Multi-hop wireless sensor networks with a central base station collecting data from sensor nodes induce a sink tree topology [KAT06, SBP17]. More generally, any data collection by a central point results in a sink tree and if time-critical decisions are made based on that data, performance guarantees are desirable, see, e.g., [ZAV02].
- We see tree topologies frequently in network-on-chip (NoC) architectures frequently [Jaf+10, Qia+16].
- Switched Ethernets set up spanning trees to avoid cycles in frame forwarding, hence, again sink trees emerge as a natural choice to support resource allocation in such installations [Jas+02].
- Sink trees are also related to so-called fat trees in supercomputing [Lei85]; in fact, fat trees have also been proposed in data center interconnects [ZBHB16, Wan+18], recently.

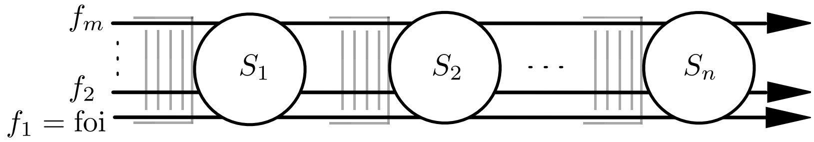
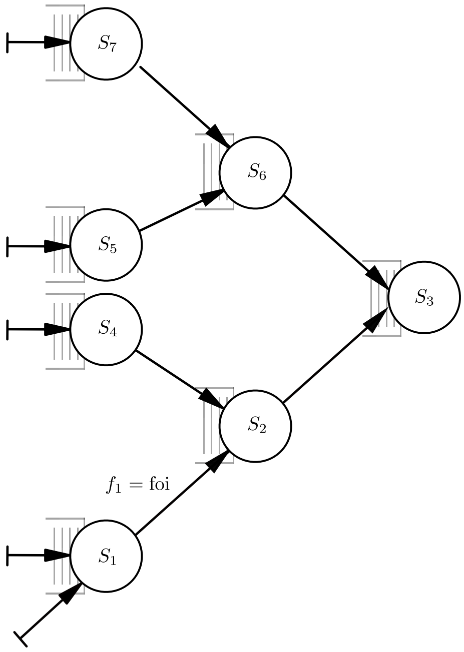
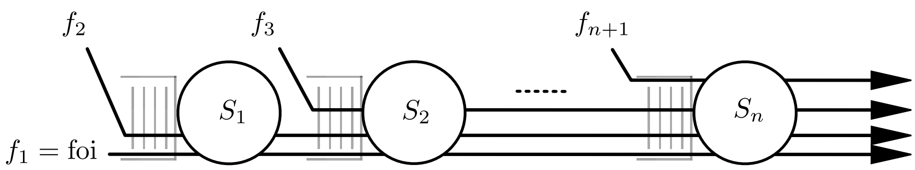


Figure 4.2: Tandem queue with m flows and n servers.



(a) Sink tree



(b) Sink tree tandem after reduction with n servers and $n+1$ flows.

Figure 4.3: Sink tree reduction

Here, we assume this reduction from network to tandem has already been performed (see Figure 4.3).

In this chapter, we show by applying the sequential pay multiplexing only once (seqPMOO) for nested interference known from the DNC (Section 2.6), we do not need to consider stochastic dependencies in the analysis. In other words, the dependencies are not *method-pertinent* for these topologies. Therefore, we can avoid the application of Hölder's inequality (Eqn. (3.22)) and show that this leads to significantly better delay bounds. As we discussed in Section 2.6, the possibility of improving delay bounds by using the PMOO analysis has been investigated extensively in DNC literature. Yet, to the best of our knowledge, it has not been investigated much in the literature on SNC with MGFs. We derive e2e delay bounds for the class of general arrivals and service with moment generating functions. For instance, this includes the fractional Brownian motion (fBm) arrival model (Eqn. (3.12)). FBm has been shown to be useful for Internet traffic modeling [Nor95, FMN00], since it is able to capture long-range dependence, which is why we also use it in our numerical experiments. On

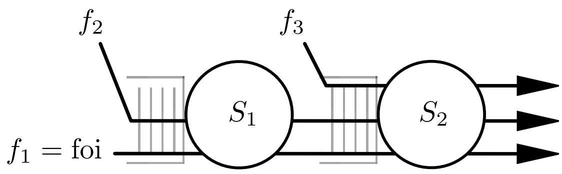


Figure 4.4: Sink tree with 3 flows and 2 servers.

the other hand, it is a non-trivial traffic type for SNC to deal with and we do not provide stationary (time-independent) delay bounds (for Hurst parameter $H > 0.5$), but transient (time-dependent) delay bounds as in Eqn. (3.26) only. Note that, however, some applications specifically look for bounds on the transient phase when only short-term performance guarantees are desired, see, e.g., [MSZ02, BF15, Bec16b, CAG18, Bec21].

OUTLINE The rest of the chapter is structured as follows: In Section 4.1, we present the derivations for the e2e delay analysis of sink tree tandems and tandem queues under independent and dependent cross-flows using different algorithms (SFA and PMOO). Section 4.2 provides a numerical evaluations of different aspects: influence of the time horizon on the transient delay bounds, effects of traffic parameters and sink tree depths / tandem lengths, comparisons between different analysis algorithms and the independent and dependent scenarios. In Section 4.3, we summarize the chapter.

4.1 END-TO-END DELAY BOUND

We start by analyzing the illustrative sink tree example in Figure 4.4, since it already enables us to point at some key differences between SFA and PMOO. We extend the results to general sink trees and tandem queues in the following subsection.

4.1.1 *Separated flow analysis (SFA)*

Here, we compute the leftover service at each server (assuming arbitrary multiplexing) until we convolve all service processes in a final step.

For the two-server sink tree in Figure 4.4, SFA yields the end-to-end service

$$S_{\text{e2e}}^{\text{SFA}} = [S_1 - A_2]^+ \otimes [S_2 - (A_3 + (A_2 \oslash S_1))]^+ . \tag{4.1}$$

Regard that the arrival process A_2 appears twice. Therefore, in the analysis, we invoke Hölder's inequality to upper bound the MGF of dependent processes. Let $\theta > 0$. Using that A_1 and S_{e2e} are independent, it follows for the bound on the delay's violation probability that

$$\begin{aligned}
&\mathrm{P}(d(t) > T) \\
&\overset{(3.26)}{\leq} \sum_{s_0=0}^{t-1} \mathrm{E}\left[e^{\theta A_1(s_0,t)}\right] \mathrm{E}\left[e^{-\theta S_{\text{e2e}}^{\text{SFA}}(s_0,t+T)}\right] \\
&\overset{(4.1)}{=} \sum_{s_0=0}^{t-1} \mathrm{E}\left[e^{\theta A_1(s_0,t)}\right] \mathrm{E}\left[e^{-\theta\left([S_1-A_2]^+\otimes[S_2-(A_3+(A_2\oslash S_1))]^+(s_0,t+T)\right)}\right]
\end{aligned}$$

$$
\begin{aligned}
&\vdots \\
&\le \sum_{s_0=0}^{t-1} \mathrm{E}\left[e^{\theta A_1(s_0,t)}\right] \\
&\quad \cdot \left(\sum_{s_1=s_0}^{t+T} \mathrm{E}\left[e^{p_1\theta A_2(s_0,s_1)}\right]^{\frac{1}{p_1}} \mathrm{E}\left[e^{e^{-p_1\theta S_1(s_0,s_1)}}\right]^{\frac{1}{p_1}} \mathrm{E}\left[e^{p_2\theta A_3(s_1,t+T)}\right]^{\frac{1}{p_2}} \right.\\
&\quad \cdot \left. \left(\sum_{s_2=0}^{s_1} \mathrm{E}\left[e^{p_2\theta A_2(s_2,t+T)}\right] \mathrm{E}\left[e^{-p_2\theta S_1(s_2,s_1)}\right] \right)^{\frac{1}{p_2}} \mathrm{E}\left[e^{-p_2\theta S_2(s_1,t+T)}\right]^{\frac{1}{p_2}} \right),
\end{aligned}
$$

where $\frac{1}{p_1} + \frac{1}{p_2} = 1$.

4.1.2 *Pay multiplexing only once (PMOO)*

For the two-server sink tree, PMOO for nested interference, seqPMOO, yields the end-to-end service

$$
S_{\mathrm{e2e}}^{\mathrm{seqPMOO}} = \left[\left([S_2 - A_3]^+ \otimes S_1 \right) - A_2 \right]^+ . \tag{4.2}
$$

In contrast to SFA, A_2 appears only once.

$$
\begin{aligned}
&\mathrm{P}(d(t) > T) \\
&\overset{(3.26)}{\le} \sum_{s_0=0}^{t-1} \mathrm{E}\left[e^{\theta A_1(s_0,t)}\right] \mathrm{E}\left[e^{-\theta S_{\mathrm{e2e}}^{\mathrm{seqPMOO}}(s_0,t+T)}\right] \\
&\overset{(4.2)}{=} \sum_{s_0=0}^{t-1} \mathrm{E}\left[e^{\theta A_1(s_0,t)}\right] \mathrm{E}\left[e^{-\theta\left[\left(S_1 \otimes [S_2-A_3]^+\right)-A_2\right]^+(s_0,t+T)}\right] \\
&\le \sum_{s_0=0}^{t-1} \mathrm{E}\left[e^{\theta A_1(s_0,t)}\right] \mathrm{E}\left[e^{\theta A_2(s_0,t+T)}\right] \mathrm{E}\left[e^{-\theta\left(\left(S_1 \otimes [S_2-A_3]^+\right)\right)(s_0,t+T)}\right] \\
&\le \sum_{s_0=0}^{t-1} \mathrm{E}\left[e^{\theta A_1(s_0,t)}\right] \mathrm{E}\left[e^{\theta A_2(s_0,t+T)}\right] \left(\sum_{s_1=s_0}^{t+T} \mathrm{E}\left[e^{-\theta S_1(s_1,t+T)}\right] \cdot \mathrm{E}\left[e^{-\theta [S_2-A_3]^+(s_0,s_1)}\right] \right) \\
&\le \sum_{s_0=0}^{t-1} \mathrm{E}\left[e^{\theta A_1(s_0,t)}\right] \mathrm{E}\left[e^{\theta A_2(s_0,t+T)}\right] \left(\sum_{s_1=s_0}^{t+T} \mathrm{E}\left[e^{-\theta S_1(s_1,t+T)}\right] \mathrm{E}\left[e^{\theta A_3(s_0,s_1)}\right] \cdot \mathrm{E}\left[e^{-\theta S_2(s_0,s_1)}\right] \right).
\end{aligned}
$$

Even though we consider only a two-server sink tree, we can already observe the key difference between SFA and PMOO, as only the SFA has to apply Hölder's inequality. We see in the following subsection, that this insight is even more evident in the general case.

4.1.3 *The general case*

In this section, we generalize the results from the two-server scenario. We start off with the tandem queue in Figure 4.2 and extend it to the general case of n servers and m flows.

Proposition 4.1 (Tandem Queue with SFA)**.** *With the SFA, the end-to-end service for m arrival flows and n servers is*

$$S_{\mathrm{e2e}}^{\mathrm{SFA}} = \left[S_1 - \sum_{j=2}^{m} A_j\right]^+ \otimes \left[S_2 - \left(\sum_{j=2}^{m} A_j\right) \oslash S_1\right]^+ \otimes \cdots \otimes \left[S_n - \left(\left(\left(\sum_{j=2}^{m} A_j\right) \oslash S_1\right) \dots\right) \oslash S_{n-1}\right]^+,$$

This yields the following bound on the delay's violation probability:

$$\begin{aligned}
&\mathrm{P}(d(t) > T) \\
&\overset{(3.26)}{\leq} \sum_{s_0=0}^{t-1} \mathrm{E}\left[e^{\theta A_1(s_0,t)}\right] \mathrm{E}\left[e^{-\theta S_{\mathrm{e2e}}(s_0,t+T)}\right] \\
&\leq \sum_{s_0=0}^{t-1} \Bigg(\mathrm{E}\left[e^{\theta A_1(s_0,t)}\right] \sum_{s_0 \leq s_1 \leq t+T} \cdots \sum_{s_{n-2} \leq s_{n-1} \leq t+T} \mathrm{E}\left[e^{p_1 \theta \sum_{j=2}^{m} A_j(s_0,s_1)} e^{-p_1 \theta S_1(s_0,s_1)}\right]^{\frac{1}{p_1}} \\
&\quad \cdots \mathrm{E}\left[e^{p_n \theta \left(\left(\left(\left(\sum_{j=2}^{m} A_j\right) \oslash S_1\right) \dots\right) \oslash S_{n-1}\right)(s_{n-1},t+T)} \cdots e^{-p_n \theta S_n(s_{n-1},t+T)}\right]^{\frac{1}{p_n}} \Bigg),
\end{aligned}$$

such that

$$\sum_{i=1}^{n} \frac{1}{p_i} = 1.$$

Proof. See Appendix B.1. □

On the contrary, the PMOO is able to circumvent the necessity to take into account a large number of dependencies:

Proposition 4.2 (Tandem Queue with PMOO)**.** *With the SFA, the end-to-end service for m arrival flows and n servers is*

$$S_{\mathrm{e2e}}^{\mathrm{seqPMOO}} = \left[\bigotimes_{i=1}^{n} S_i - \sum_{j=2}^{m} A_j\right]^+.$$

This yields the following bound on the delay's violation probability:

$$\begin{aligned}
&\mathrm{P}(d(t) > T) \\
&\overset{(3.26)}{\leq} \sum_{s_0=0}^{t-1} \mathrm{E}\left[e^{\theta A_1(s_0,t)}\right] \mathrm{E}\left[e^{-\theta S_{\mathrm{e2e}}(s_0,t+T)}\right] \\
&\leq \sum_{s_0=0}^{t-1} \Bigg(\mathrm{E}\left[e^{\theta A_1(s_0,t)}\right] \mathrm{E}\left[e^{\theta \sum_{j=2}^{m} A_j\left(s_0,t+T\right)}\right] \\
&\quad \cdot \sum_{s_0 \leq s_1 \leq t+T} \cdots \sum_{s_{n-2} \leq s_{n-1} \leq t+T} \mathrm{E}\left[e^{-\theta S_1(s_0,s_1)}\right] \cdots \mathrm{E}\left[e^{-\theta S_n(s_{n-1},t+T)}\right] \Bigg).
\end{aligned}$$

Proof. See Appendix B.1. □

For the special case of constant rate servers with rate $C_i \geq 0$, $i = 1, \dots, n$, Proposition 4.2 can be simplified to

$$\mathrm{P}(d(t) > T) \overset{(2.28)}{\leq} \sum_{s_0=0}^{t-1} \mathrm{E}\left[e^{\theta A_1(s_0,t)}\right] \mathrm{E}\left[e^{\theta \sum_{j=2}^{m} A_j(s_0,t+T)}\right] e^{-\theta C_{i^*}(t+T-s_0)}, \tag{4.3}$$

where we define $C_{i^*} := \arg\min_{i=1,\dots,n} \{C_i\}$ (this follows by Eqn. (2.29)).

Next, we state the results for sink trees as in Figure 4.3b.

Proposition 4.3 (Sink Tree with SFA)**.** *With the SFA, the end-to-end service for $n+1$ arrival flows and n servers in a sink tree is*

$$S_{\mathrm{e2e}}^{\mathrm{SFA}} = [S_1 - A_2]^+ \otimes [S_2 - (A_3 + (A_2 \oslash S_1))]^+ \cdots \otimes \left[S_n - \left(A_n + (A_{n-1} \oslash S_{n-1}) + \cdots + \left((A_1 \oslash S_1) \oslash [S_2 - A_2]^+\right) \oslash \cdots \oslash [S_{n-1} - (A_2 + \cdots + A_{n-1})]^+\right)\right]^+.$$

This yields the following bound on the delay's violation probability:

$$\begin{aligned}
&\mathrm{P}(d(t) > T) \\
&\leq \sum_{s_0=0}^{t-1} \mathrm{E}\left[e^{\theta A_1(s,t)}\right] \left(\sum_{s_1=s_0}^{t+T} \cdots \sum_{s_{n-1}=s_{n-2}}^{t+T} \mathrm{E}\left[e^{p_1\theta A_2(s_0,s_1)}\right]^{\frac{1}{p_1}} \mathrm{E}\left[e^{-p_1\theta S_1(s_0,s_1)}\right]^{\frac{1}{p_1}}\right. \\
&\cdots \mathrm{E}\left[e^{p_n\theta\left(A_{n+1}+(A_n\oslash S_{n-1})+\cdots+\left(((A_2\oslash S_1)\oslash[S_2-A_3]^+)\oslash\cdots\oslash[S_{n-1}-(A_3+\cdots+A_n)]^+\right)\right)(s_{n-1},t+T)}\right]^{\frac{1}{p_n}} \\
&\left.\cdot \mathrm{E}\left[e^{-p_n\theta S_n(s_{n-1},t+T)}\right]^{\frac{1}{p_n}}\right)
\end{aligned}$$

such that

$$\sum_{i=1}^{n} \frac{1}{p_i} = 1.$$

Again, the PMOO does not have to take into account the dependencies between cross-flows that share servers.

Proposition 4.4 (Sink Tree with PMOO)**.** *With the sequential PMOO, the end-to-end service for $n+1$ arrival flows and n servers in a sink tree is*

$$S_{\mathrm{e2e}}^{\mathrm{seqPMOO}} = \left[\left(\left[\left([S_n - A_{n+1}]^+ \otimes S_{n-1}\right) - A_n\right]^+ \otimes \cdots \otimes S_1\right) - A_2\right]^+. \tag{4.4}$$

This yields the following bound on the delay's violation probability:

$$\begin{aligned}
&\mathrm{P}(d(t) > T) \\
&\leq \sum_{s_0=0}^{t-1} \mathrm{E}\left[e^{\theta A_1(s_0,t)}\right] \mathrm{E}\left[e^{\theta A_2(s_0,t+T)}\right] \left(\sum_{s_1=s_0}^{t+T} \mathrm{E}\left[e^{-\theta S_1(s_1,t+T)}\right] \mathrm{E}\left[e^{\theta A_3(s_0,s_1)}\right]\right. \\
&\cdot \left(\sum_{s_2=s_0}^{s_1} \mathrm{E}\left[e^{-\theta S_2(s_2,s_1)}\right] \mathrm{E}\left[e^{\theta A_4(s_0,s_2)}\right] \cdots \left(\sum_{s_k=s_0}^{s_{k-1}} \mathrm{E}\left[e^{-\theta S_k(s_k,s_{k-1})}\right] \mathrm{E}\left[e^{\theta A_{k+2}(s_0,s_k)}\right]\right.\right. \\
&\left.\left.\left.\cdots \left(\sum_{s_{n-1}=s_0}^{s_{n-2}} \mathrm{E}\left[e^{-\theta S_{n-1}(s_{n-1},s_{n-2})}\right] \mathrm{E}\left[e^{\theta A_{n+1}(s_0,s_{n-1})}\right] \mathrm{E}\left[e^{-\theta S_n(s_0,s_{n-1})}\right]\right)\right)\right)\right).
\end{aligned}$$

Proof. See Appendix B.2. □

4.1.4 *Delay bounds with PMOO under dependent cross-flows*

So far, the analysis only considered originally independent arrival flows. Now, if we assume the cross-flow arrivals to be dependent, even with the PMOO, we have to apply Hölder's inequality. Such dependencies may be due to resource sharing between cross-flows before they hit the flow of interest (foi), or simply because the original data sources are already dependent, as, e.g., in an environmental sensor network where the range of sensor nodes is overlapping and, thus, an observed physical phenomenon is reported by several neighboring nodes at the same time. Again, we start with tandem queues.

Proposition 4.5 (Tandem Queue with PMOO and Dependent Cross-Flows)**.** *If all m flows are dependent, the PMOO yields*

$$\begin{aligned}
&\mathrm{P}(d(t) > T) \\
&\leq \sum_{s_0=0}^{t-1} \mathrm{E}\left[e^{p_1\theta A_1(s_0,t)}\right]^{\frac{1}{p_1}} \\
&\quad \cdot \Bigg(\sum_{s_1=s_0}^{t+T} \mathrm{E}\left[e^{p_2 p_3\theta A_2(s_0,t+T)}\right]^{\frac{1}{p_3}} \mathrm{E}\left[e^{-p_2\theta S_1(s_0,s_1)}\right] \cdots \\
&\quad \cdot \sum_{s_{l-1}=s_{l-2}}^{t+T} \mathrm{E}\left[e^{p_2 p_{l+1}\theta A_l(s_0,t+T)}\right]^{\frac{1}{p_{l+1}}} \cdot \mathrm{E}\left[e^{-p_2\theta S_{l-1}(s_{l-2},s_{l-1})}\right] \\
&\quad \cdot \sum_{s_{n-1}=s_{n-2}}^{t+T} \mathrm{E}\left[e^{p_2 p_{m+1}\theta A_m(s_0,t+T)}\right]^{\frac{1}{p_{m+1}}} \\
&\quad \cdot \mathrm{E}\left[e^{-p_2\theta S_{n-1}(s_{n-2},s_{n-1})}\right] \mathrm{E}\left[e^{-p_2\theta S_n(s_{n-1},t+T)}\right]\Bigg)^{\frac{1}{p_2}}
\end{aligned}$$

such that

$$\begin{aligned}
\frac{1}{p_1} + \frac{1}{p_2} &= 1, \\
\frac{1}{p_3} + \cdots + \frac{1}{p_{m+1}} &= 1.
\end{aligned}$$

Proof. See Appendix B.1. □

For the special case of constant rate servers with rate $C_i \geq 0$, $i = 1, \ldots, n$, Proposition 4.2 can be simplified to

$$\begin{aligned}
&\mathrm{P}(d(t) > T) \\
&\overset{(2.29),(3.22)}{\leq} \sum_{s_0=0}^{t-1} \mathrm{E}\left[e^{p_1\theta A_1(s_0,t)}\right]^{\frac{1}{p_1}} \left(\prod_{j=2}^{m} \mathrm{E}\left[e^{p_2 p_{j+1}\theta A_j(s_0,t+T)}\right]^{\frac{1}{p_{j+1}}(s_0,t+T)} e^{-p_2\theta C_{i^*}(t+T-s_0)}\right)^{\frac{1}{p_2}}
\end{aligned} \tag{4.5}$$

such that

$$\begin{aligned}
\frac{1}{p_1} + \frac{1}{p_2} &= 1, \\
\frac{1}{p_3} + \cdots + \frac{1}{p_{n+1}} &= 1,
\end{aligned}$$

where we define $C_{i^*} := \arg\min_{i=1,\ldots,n} \{C_i\}$.

For sink trees, we obtain the following bound on the delay violation probability.

Proposition 4.6 (Sink Tree with PMOO and Dependent Cross-Flows)**.** *If all n cross-flows are dependent, the PMOO yields*

$$
\begin{aligned}
&\mathrm{P}(d(t) > T) \\
&\leq \sum_{s_0=0}^{t-1} \mathrm{E}\left[e^{\theta A_1(s_0,t)}\right] \left(\mathrm{E}\left[e^{p_1\theta A_2(s_0,t+T)}\right]\right)^{\frac{1}{p_1}} \\
&\quad \cdot \left(\sum_{s_1=s_0}^{t+T} \mathrm{E}\left[e^{-p_2\theta S_1(s_1,t+T)}\right] \mathrm{E}\left[e^{p_2p_3\theta A_3(s_0,s_1)}\right]^{\frac{1}{p_3}} \right. \\
&\quad \cdots \left(\sum_{s_{n-2}=s_0}^{s_{n-3}} \mathrm{E}\left[e^{-p_2p_4\cdots p_{2n-4}\theta S_{n-2}(s_{n-2},s_{n-3})}\right] \right. \\
&\quad \cdot \mathrm{E}\left[e^{p_2p_4\cdots p_{2n-4}p_{2n-3}\theta A_n(s_0,s_{n-1})}\right]^{\frac{1}{p_{2n-3}}} \\
&\quad \cdot \left(\sum_{s_{n-1}=s_0}^{s_{n-2}} \mathrm{E}\left[e^{-p_2p_4\cdots p_{2n-2}\theta S_{n-1}(s_{n-1},s_{n-2})}\right] \mathrm{E}\left[e^{p_2p_4\cdots p_{2n-2}\theta A_{n+1}(s_0,s_{n-1})}\right] \right. \\
&\quad \left. \left. \left. \cdot \mathrm{E}\left[e^{-p_2p_4\cdots p_{2n-2}\theta S_n(s_0,s_{n-1})}\right] \right)^{\frac{1}{p_{2n-2}}} \cdots \right)^{\frac{1}{p_4}} \right)^{\frac{1}{p_2}},
\end{aligned}
$$

such that

$$
\begin{aligned}
\frac{1}{p_1} + \frac{1}{p_2} &= 1, \\
&\vdots \\
\frac{1}{p_{2n-1}} + \frac{1}{p_{2n-2}} &= 1.
\end{aligned}
$$

Proof. See Appendix B.2. □

4.2 NUMERICAL EVALUATION

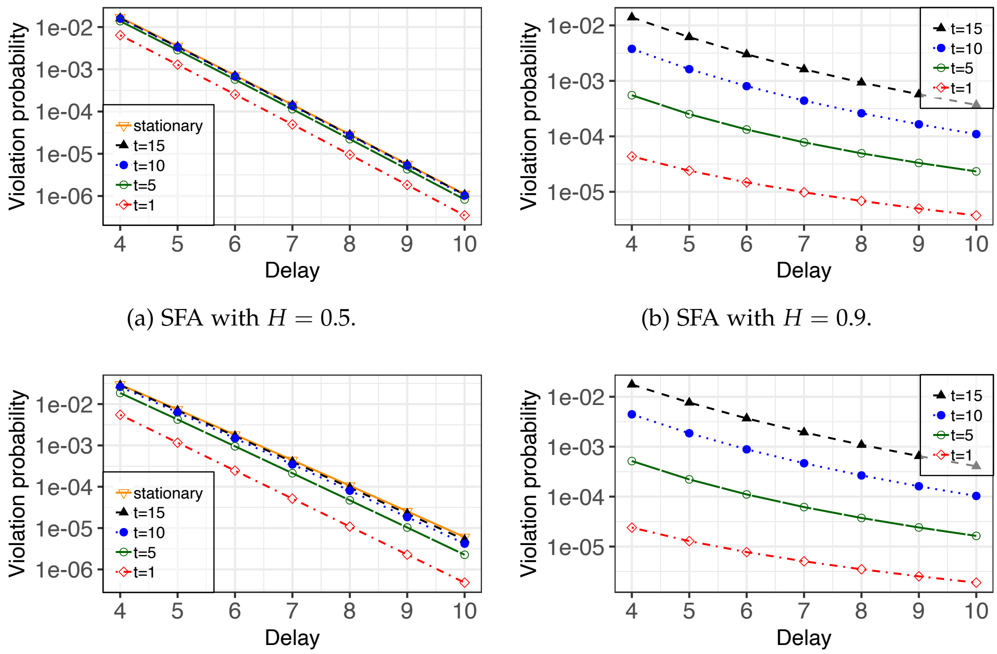
In this section, we evaluate and compare the delay bounds of sink trees and tandem queues for different techniques and parameters. At the beginning, all flows are assumed to be independent. At first, we investigate the impact of the transient time horizon t on the bound and how it relates to the assumed fractional Brownian motion traffic model. Then, we compare the SFA with the PMOO, before taking a look at the sensitivity of the model with respect to the fBm parameters. Furthermore, we consider the scaling behavior when increasing the tree depth / tandem length. In the last experiment, we relax the independence assumption and consider the case of dependent cross-flows as well as mixed scenarios (the latter only for tandem queues).

For the arrival, we choose the fractional Brownian motion (fBm) arrival model (Eqn. (3.12)):

$$A(s,t) \overset{(3.12)}{=} \lambda \cdot (t-s) + \sigma \cdot Z(t-s),$$

with

$$\mathrm{E}\left[e^{\theta A(s,t)}\right] \overset{(3.13)}{=} e^{\theta\lambda(t-s)+\frac{\theta^2\sigma^2}{2}(t-s)^{2H}}.$$



(a) SFA with $H = 0.5$.

(b) SFA with $H = 0.9$.

(c) PMOO with $H = 0.5$.

(d) PMOO with $H = 0.9$.

Figure 4.5: Delay violation probability for the two-server sink tree and different t.

For $H \in (0.5, 1)$, fBm exhibits a property called long-range dependence (LRD). If not mentioned otherwise, throughout the experiments, we choose $\lambda = 1$, $\sigma^2 = 1$, and $H = 0.7$.

Given the continuous nature of fBm, the arrivals in Eqn. (3.12) are a continuous-time process (Eqn. (2.2)) that has to be discretized in order to be applicable to our discrete-time arrival model (cf. Eqn. (2.1)).

We proceed as in [CBL06]. Let $\tau > 0$ be a discretization parameter and $t \geq 0$. Then, assuming a dynamic server and arrivals and service to be independent, it can be shown for the delay bound that

$$\mathrm{P}(d(t) > T) \leq \sum_{j=0}^{\lfloor \frac{t}{\tau} \rfloor} \mathrm{E}\left[e^{\theta A(t-(j+1)\tau, t)}\right] \mathrm{E}\left[e^{-\theta S(t-j\tau, t+T)}\right].$$

The rest follows along similar lines as in the discrete-time case (see, e.g., [Bec16a, pp. 30] for MGF performance bounds of continuous-time processes).

If not explicitly specified, by default, the cross-flows are assumed to be independent and t is equal to 20. For the fBm arrivals, we fix $\lambda = 1.0$, $\sigma = 1.0$, and $H = 0.7$. All servers are assumed to be work-conserving with constant rate (we assume homogeneous sink trees and tandem queues) is denoted by $C \geq 0$. Further, all results are obtained by numerically optimizing $\theta > 0$ and the Hölder parameters $p_i > 1$.

4.2.1 *Impact of a finite time horizon*

We compare the delay bounds for different time horizons t, applying the sink tree bounds for SFA (Proposition 4.3) and PMOO (Proposition 4.4), respectively. The results are depicted in Figure 4.5.

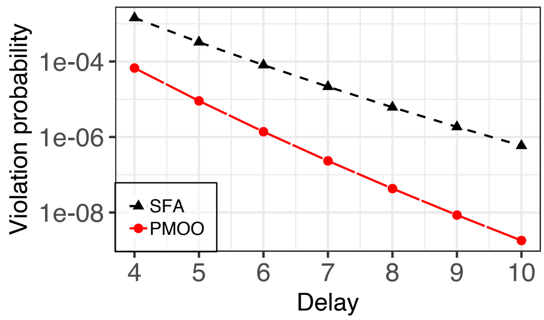


Figure 4.6: Comparison between delay violation probabilities using SFA and PMOO (sink tree).

Table 4.1: Comparison between runtimes [s] using SFA and PMOO (sink tree).

Delay	SFA	PMOO
4	1.398	0.036
5	1.492	0.037
6	1.580	0.039
7	1.680	0.040
8	2.072	0.049
9	2.111	0.041
10	2.337	0.043

We observe that the delay bounds do not change significantly for larger t when the Hurst parameter is $H = 0.5$ (Figure 4.5a and 4.5c) and rate $C = 4.5$. Since for this particular H, the fBm traffic model is (σ, ρ)-bounded (Definition 3.2), we can also derive stationary bounds that hold for all $0 \leq s \leq t$:

$$\mathrm{E}\left[e^{\theta A(s,t)}\right] \overset{(3.13)}{=} e^{\theta\left(\lambda+\theta\frac{\sigma^2}{2}\right)(t-s)} = e^{\theta\rho_A(\theta)\cdot(t-s)+\theta\sigma_A(\theta)}, \quad \theta > 0,$$

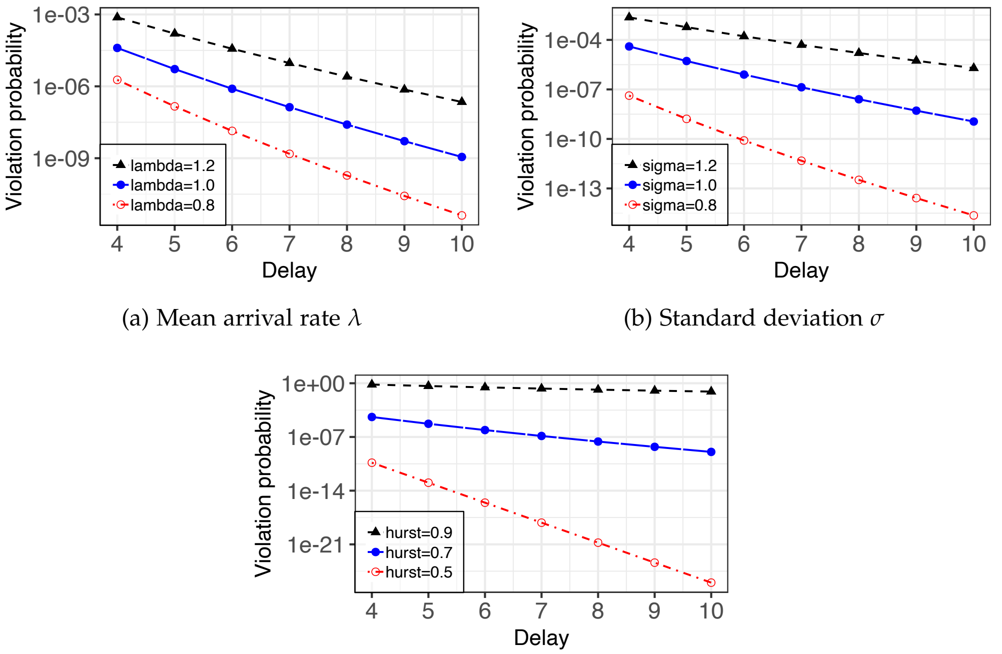
where

$$\sigma_A(\theta) = 0,$$
$$\rho_A(\theta) = \lambda + \theta\frac{\sigma^2}{2}.$$

However, for $H = 0.9$ (Figure 4.5b and 4.5d), when the fBm traffic model exhibits a long-range dependence, the delay bounds vary strongly for different t and server rate $C = 8.0$. This indicates that, if one is aiming at transient bounds, results obtained from a stationary analysis may be too conservative. The tandem queue exhibit similar trends, for a numerical evaluation, see therefore Figure B.1 in Appendix B.3.

4.2.2 *Comparison between SFA and PMOO*

For a sink tree with two servers, we compare the delay bounds using SFA and PMOO. To that end, we consider a two-server sink tree with server rate $C = 7.0$.



(a) Mean arrival rate λ

(b) Standard deviation σ

(c) Hurst parameter H

Figure 4.7: Parameter sensitivity of fractional Brownian motion on the delay bounds using PMOO (sink tree).

The results in Figure 4.6 indicate a significant gap in the delay bounds. While the difference in the violation probability is about two orders of magnitude, in the delay space, the PMOO bound exhibits an improvement of roughly 40%. This is caused by the additional application of Hölder's inequality, that is only necessary in the SFA. Moreover, this additional parameter also significantly increases the runtimes, as Table 4.1 shows for this particular scenario. Hence, in the following experiments, we only use PMOO. For the tandem queue, we observe similar results; see therefore Figure B.2 in Appendix B.3.

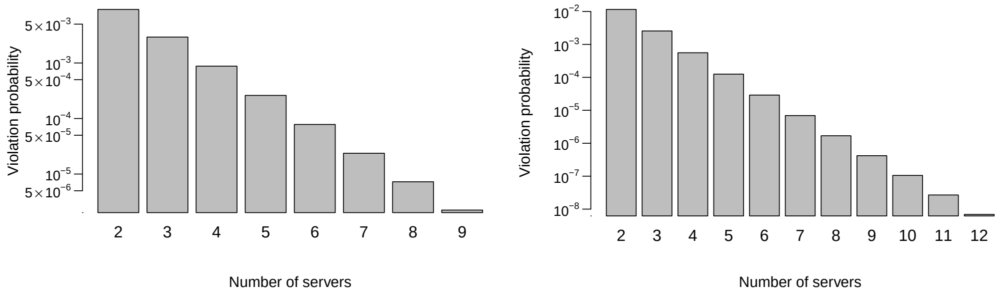
4.2.3 *Parameter sensitivity of fractional Brownian motion*

In this subsection, we investigate the impact of the fBm traffic model parameters on the delay bounds. Therefore, for a three-server sink tree, we fixed the server rates to $c = 9.0$. The resulting PMOO bounds are shown in Figure 4.7.

We see that, while all parameters clearly influence the outcome, the parameter sensitivity significantly differs. As expected, it is evident that, at the same load, the Hurst parameter H can be decisive whether the system suffers from long queues ($H = 0.9$), or hardly sees any queueing effects ($H = 0.5$) (see Figure 4.7c). For tandem queues we make similar observations, see therefore Figure B.3 in Appendix B.3.

4.2.4 *Scaling effects of PMOO*

In this experiment, we focus on how the delay violation probability scales with the number of servers for a delay of $T = 4$. For the sink tree, we keep the



(a) Sink tree and constant utilization at the last server.

(b) Tandem queue and constant utilization at each server.

Figure 4.8: Delay violation probability for different lengths using PMOO.

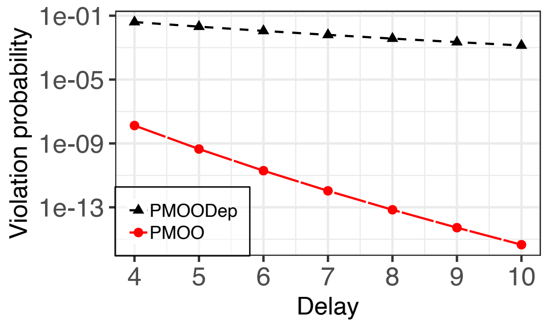


Figure 4.9: Comparison between delay violation probabilities for independent and dependent cross-flows using PMOO (sink tree).

utilization at the last server (since it is the server with the heaviest load in a homogeneous sink tree, $\frac{(n+1)\lambda}{c}$) constant, i.e. we scale its capacity with the number of flows. For the tandem queue, on the other hand, we choose the parameters such that the utilization $\frac{n\lambda}{C}$ is constant at each server.

The results in Figure 4.8 show that the delay bounds improve with the number of servers. This improvement is due to statistical multiplexing gains as the number of flows grows. It has been shown to scale with $\Omega(\sqrt{n})$ [CS12] which is supported by both Figure 4.8a and 4.8b.

4.2.5 *Comparison between independent and dependent cross-flows*

So far, all experiments considered the cross-flows to be independent. In this last experiment, we now omit the independence assumption, i.e., we apply Hölder's inequality to the MGF of the cross-flows. The delay bounds for a sink tree of three servers with server rate $c = 9.0$ are depicted in Figure 4.9.

As expected, the impact of dependence (and therefore Hölder's inequality) is strong. The delay violation probability is more than 9 orders of magnitude higher compared to the independent case. This indicates the importance of treating and, if possible, avoiding the invocation of Hölder's inequality. We observe similar results for the tandem queue, see Figure B.4 in Appendix B.3.

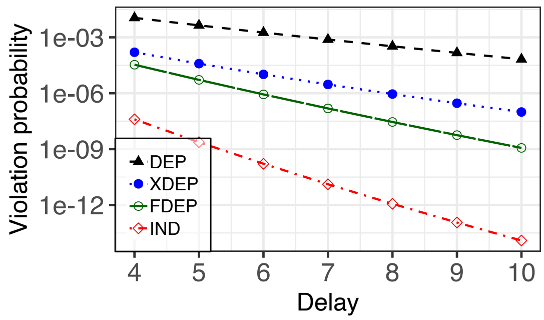


Figure 4.10: Delay bound comparison for three servers with increasing degree of dependence.

4.2.6 *Mixed independence / dependence scenarios*

In the previous experiments, we have seen that dependencies can be very detrimental for the delay bounds. In this last experiment, we now investigate a mixed scenario with some flows being dependent and others being independent. More specifically, we use 3 servers in the tandem queue and 3 flows correspondingly. We evaluate the (optimized) PMOO delay bounds for four scenarios:

1. (IND) all flows are independent;
2. (DEP) all flows are dependent;
3. (XDEP) the cross-flows are dependent, but the foi is independent of them;
4. (FDEP) the foi is dependent with one of the cross-flows, but not with the other one, the cross-flows are independent of each other.

The analytical derivation of the new scenarios XDEP and FDEP can be found in Appendix B.4.

In this setting, we set the server rate to $C = 8.0$. The results are depicted in Figure 4.10.

As expected, the cases with partial dependencies appear in the gap between full dependence and independence, respectively. XDEP's violation probability is three orders of magnitude better than the full dependence for a delay bound equal to 3. This gap increases over time. In the delay space, this leads to an improvement of about 40%. Switching the dependence from dependent cross flows to the dependence of the foi and one cross flow, we gain another approx. 25%. The independent case, on the other hand, outperforms the FDEP by far, especially for larger delays.

The results suggest again that dependence leads to a high penalty in the analysis. If, however, partial independence can be assumed, making use of this property yields significantly better bounds. As we observe, not only the number of dependent flows is important, but also their relation. Apparently, having a dependence among the cross flows is worse for our analysis than between the foi and one of the cross flows instead.

4.3 SUMMARY

In this chapter, we have derived end-to-end delay bounds for sink trees and tandem queues. It has been shown that sequential pay multiplexing only once (seqPMOO) has to consider less stochastic dependencies, and therefore applies less Hölder inequalities in the analysis. Further, our numerical experiments with a fractional Brownian motion traffic model indicate that this leads to significantly improved delay bounds.

5 DEALING WITH DEPENDENCE USING PMOO FOR TREE NETWORKS

Results presented in this chapter are joint work with Anne Bouillard and Jens Schmitt [BNS21, BNS22].

In the larger context of probabilistic end-to-end performance analysis, stochastic dependencies play a significant role. In the previous chapter, we have seen the sequential pay multiplexing only once (seqPMOO) analysis has no method-pertinent dependencies (see also Chapter 4) for tandem queues and sink trees, if we assume all arrivals and servers to be independent initially.

However, to the best of our knowledge, research in the field of non-nested or overlapping interference has not yet been a focus in the SNC (with the notable exception of [NS20a]). For instance, for a network with overlapping interference as in Figure 5.1, state-of-the-art analysis still requires at least one application of Hölder's inequality (Eqn. (3.22)), even if we assume all external arrival processes to be independent. This is also illustrated in Figure 5.2b (anticipating some of the results from Section 5.3): method-pertinent dependencies force the state-of-the-art SNC bounds to deteriorate significantly; in particular, it is observable that, in comparison to the single-node case, even the scaling of the delay bounds is not captured correctly anymore.

The overall goal of this chapter is therefore to unleash the power of the PMOO principle in the SNC framework in order to reduce the gap between simulations and bounds even in more complex and larger networks of queues. To that end, we make the following contributions:

- We present a PMOO-based SNC end-to-end analysis for an extension of tandems, so-called tree networks. It achieves *zero* method-pertinent stochastic dependencies when external arrivals and service processes are independent. In other words, if all input flows are assumed to be independent, we can derive bounds without a single application of Hölder's inequality.
- We conduct an extensive numerical evaluation with respect to the accuracy of the new bounds for several traffic classes and different network topologies.

5.1 TREE NETWORK ANALYSIS

In this section, we derive a result enabling us to unleash the power of the PMOO principle for the SNC when all arrival and service processes are originally independent. In particular, this result is directly applicable to tree networks.

Let us start with an illustrative example and consider the overlapping tandem in Figure 5.1. Let S_1, S_2, S_3 be work-conserving servers with eponymous service processes. Assume that for all $t \geq 0$, t_j is the start of the backlogged period (as in Eqn. (2.15)) of server S_j before t_{j+1} (t for the last server). Further, we denote

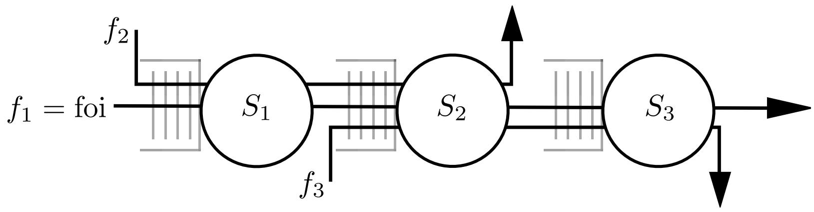


Figure 5.1: Overlapping tandem

the output of flow i at server S_j by D_i^j (for the output of the last server of a flow, we omit the superscript). Mimicking the analysis in Section 2.6, yet, this time for bivariate processes, we obtain

$$\begin{aligned} D_1(t) + D_3(t) - D_1(t_3) - D_3(t_3) &\geq S_3(t_3, t) \\ D_1^2(t_3) + D_2^2(t_3) + D_3^2(t_3) - D_1^2(t_2) - D_2^2(t_2) - D_3^2(t_2) &\geq S_2(t_2, t_3) \\ D_1^1(t_2) + D_2^1(t_2) - D_1^1(t_1) - D_2^1(t_1) &\geq S_1(t_1, t_2). \end{aligned}$$

Since t_j is the start of the backlogged period of server S_j, we have that $D_i^j(t_j) = A_i^j(t_j)$ for $j = 1, 2$. Using this yields

$$\begin{aligned} D_1(t) + D_3(t) - D_1^2(t_3) - D_3^2(t_3) &\geq S_3(t_3, t) \\ D_1^2(t_3) + D_2^2(t_3) + D_3^2(t_3) - D_1^1(t_2) - D_2^1(t_2) - A_3(t_2) &\geq S_2(t_2, t_3) \\ D_1^1(t_2) + D_2^1(t_2) - A_1(t_1) - A_2(t_1) &\geq S_1(t_1, t_2). \end{aligned}$$

Summing all three inequalities and simplifying leads to

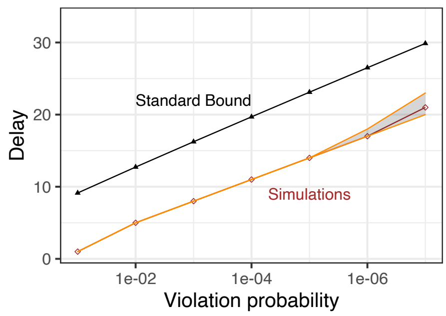
$$D_1(t) + \underbrace{D_2^2(t_3)}_{\leq A_2(t_3)} + \underbrace{D_3(t)}_{\leq A_3(t)} - A_1(t_1) - A_2(t_1) - A_3(t_2) \geq S_1(t_1, t_2) + S_2(t_2, t_3) + S_3(t_3, t).$$

Making use of causality (see also Eqn. (2.3)) and isolating $D_1(t)$ on the left-hand side yields

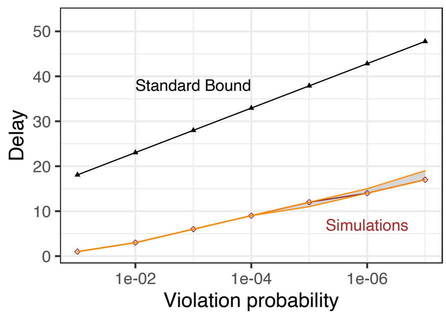
$$D_1(t) \geq A_1(t_1) + S_1(t_1, t_2) + S_2(t_2, t_3) + S_3(t_3, t) - A_2(t_1, t_3) - A_3(t_2, t).$$

Moreover, we have by definition of t_1 that

$$D_1(t) \geq D_1(t_1) = A_1(t_1).$$



(a) Single-node case



(b) Overlapping tandem

Figure 5.2: Stochastic delay bounds: Simulation results and SNC bounds (for traffic with exponentially distributed increments and constant rate servers). For more details, see Section 5.3.

Combining both inequalities proves a dynamic server (Definition 3.4):

$$\begin{aligned}
&D_1(t)\\
\geq &A_1(t_1) + [S_1(t_1,t_2) + S_2(t_2,t_3) + S_3(t_3,t) - A_2(t_1,t_3) - A_3(t_2,t)]^+\\
\geq &\inf_{0\leq s\leq t}\left\{A_1(s) + \left[\inf_{s\leq t_2\leq t_3\leq t}\{S_1(s,t_2) + S_2(t_2,t_3) + S_3(t_3,t) - A_2(s,t_3) - A_3(t_2,t)\}\right]^+\right\}\\
= &A_1 \otimes S_{\text{e2e}}^{\text{PMOO}}(0,t),
\end{aligned}$$

where we define the end-to-end service process

$$S_{\text{e2e}}^{\text{PMOO}}(s,t) := \left[\inf_{s\leq t_2\leq t_3\leq t}\{S_1(s,t_2) + S_2(t_2,t_3) + S_3(t_3,t) - A_2(s,t_3) - A_3(t_2,t)\}\right]^+. \tag{5.1}$$

Let us now formalize this observation for tandems. We denote the path of flow f_i, $i = 1,\dots,m$ by

$$\pi_i = (\pi_i(1),\dots,\pi_i(l_i)),$$

where l_i is defined as the length of the path of flow f_i. If server S_j is on the path of f_i, we write $j \in \pi_i$. Let A_i^j be the arrival process of flow f_i at server S_j and, if the successor of this flow is S_{j+1}, $D_i^j = A_i^{j+1}$ its respective departure process. For A_i^1 and $D_i^{\pi_i(l_i)}$, we omit the superscripts and we define $D_i^{\pi_i(1)-1} := A_i$ for all $i = 1,\dots,m$.

Theorem 5.1. *Assume a tandem network with flows $f_1,\dots,f_m$ and work-conserving servers $S_1,\dots,S_n$. Assume a flow of interest (foi) traversing it. By abuse of notation, each server offers the service $S_j(s,t)$, $j = 1,\dots,n$ for $0 \leq s \leq t$. Then, the foi sees the dynamic server for all $0 \leq t_1 \leq t_{n+1}$*

$$S_{\text{e2e}}^{\text{PMOO}}(t_1,t_{n+1}) = \left[\inf_{t_1\leq t_2\leq\cdots\leq t_n\leq t_{n+1}}\left\{\sum_{j=1}^{n} S_j(t_j,t_{j+1}) - \sum_{i=2}^{m} A_i\left(t_{\pi_i(1)}, t_{\pi_i(l_i)+1}\right)\right\}\right]^+. \tag{5.2}$$

Proof. The proof formalizes the illustrative example above and is therefore postponed to Appendix C.1. □

Less formally, if we used the times $0 \leq s \leq t$ instead, we can rewrite it more intuitively. Let us assume that $\pi_2(1) = 1$, $\pi_i(1) > 1$, $\pi_i(l_i) < n$, $i = 2,\dots,m-1$, and $\pi_m(l_m) = n$. Then, this is equivalent to

$$\begin{aligned}
&S_{\text{e2e}}^{\text{PMOO}}(s,t)\\
= &\left[\inf_{s\leq t_2\leq\cdots\leq t_n\leq t}\left\{S_1(s,t_2) + \sum_{j=2}^{n-1} S_j(t_j,t_{j+1}) + S_n(t_n,t)\right.\right.\\
&\left.\left. - A_2\left(s,t_{\pi_2(l_2)+1}\right) - \sum_{i=3}^{m-1} A_i\left(t_{\pi_i(1)},t_{\pi_i(l_i)+1}\right) - A_m\left(t_{\pi_m(1)},t\right)\right\}\right]^+.
\end{aligned}$$

The form of the leftover service in Theorem 5.1 is quite appealing, as the indices for the service processes are just increased by 1, whereas for the arrivals, we just have to consider the server index when entering the tandem and the server index plus 1 when leaving the network. In other words, it contains all

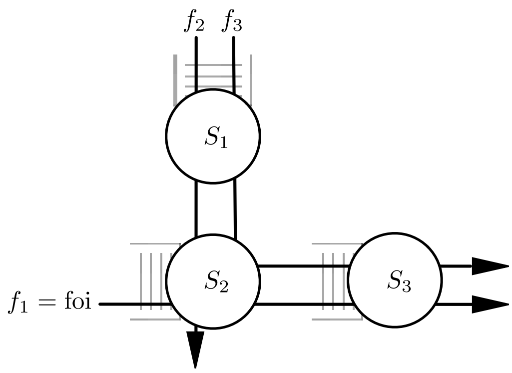


Figure 5.3: The $\mathbb{L}$

the topological information. The only exceptions to this rule are the first (time variable s) and last server (time variable t) of the tandem.

The above theorem can even be generalized to tree networks, a special case of a feedforward network where we do not have any rejoining flows and where we can define a unique successor to each server. In the following, we consider another illustrative example, the $\mathbb{L}$ (Figure 5.3).

Assume that for all $t \geq 0$, t_3 is the start of the backlogged period of server S_3 before t, $t_1, (t_2)$ the start of the backlogged period of Server $S_1(S_2)$ before $t_2(t_3)$. We obtain

$$\begin{aligned} D_1(t) + D_3(t) - D_1(t_3) - D_3(t_3) &\geq S_3(t_3, t) \\ D_1^2(t_3) + D_2(t_3) + D_3^2(t_3) - D_1^2(t_2) - D_2(t_2) - D_3^2(t_2) &\geq S_2(t_2, t_3) \\ D_2^1(t_2) + D_3^1(t_2) - D_2^1(t_1) - D_3^1(t_1) &\geq S_1(t_1, t_2). \end{aligned}$$

Since t_j is the start of the backlogged period of server S_j, this is equivalent to

$$\begin{aligned} D_1(t) + D_3(t) - D_1^2(t_3) - D_3^2(t_3) &\geq S_3(t_3, t) \\ D_1^2(t_3) + D_2(t_3) + D_3^2(t_3) - A_1(t_2) - D_2^1(t_2) - D_3^1(t_2) &\geq S_2(t_2, t_3) \\ D_2^1(t_2) + D_3^1(t_2) - A_2(t_1) - A_3(t_1) &\geq S_1(t_1, t_2). \end{aligned}$$

Summing all three inequalities and simplifying leads to

$$D_1(t) + \underbrace{D_2(t_3)}_{\leq A_2(t_3)} + \underbrace{D_3(t)}_{\leq A_3(t)} - A_1(t_2) - A_2(t_1) - A_3(t_1) \geq S_1(t_1, t_2) + S_2(t_2, t_3) + S_3(t_3, t).$$

Making use of causality and isolating $D_1(t)$ on the left-hand side yields

$$D_1(t) \geq A_1(t_2) + S_1(t_1, t_2) + S_2(t_2, t_3) + S_3(t_3, t) - A_2(t_1, t_3) - A_3(t_1, t).$$

Following along the lines for the overlapping tandem, this results in the end-to-end service

$$S_{\text{e2e}}^{\text{PMOO}}(s, t) = \left[\inf_{\substack{s \leq t_3 \leq t, \\ 0 \leq t_1 \leq s}} \{ S_1(t_1, s) + S_2(s, t_3) + S_3(t_3, t) - A_2(t_1, t_3) - A_3(t_1, t) \} \right]^+ .$$

The important difference here is that t_1 is not lower bounded by s in the infimum. In general terms, if S_k is succeeded by S_j (in tree network, a server can only have at most one successor), then we have $t_k \leq t_j$ as a constraint in the infimum. We observe that this unleashed PMOO analysis takes "byways" into consideration, whereas a previous stochastic network calculus analysis such as the seqPMOO would calculate an output bound for the flows f_2 and f_3 at server S_2 and continue with a tandem analysis. However, this would result in

$$S_{\text{e2e}}^{\text{seqPMOO}} = \left[\left(\left[S_1 - A_2 \oslash [S_2 - A_3]^+ \right]^+ \otimes S_3 \right) - A_3 \oslash [S_2 - A_2]^+ \right]^+$$

and hence has method-pertinent dependencies, since A_2 and A_3 appear twice (see also dynamic priority assignment in Section 2.6). Theorem 5.2 can therefore be used to generalize the results of [SZ06, SZM08, Bou+08] in deterministic network calculus (DNC).

Theorem 5.2. *Assume a tree network with flows $f_1, \ldots, f_m$ and work-conserving servers $S_1, \ldots, S_n$. Assume a flow of interest (foi) f_1 traversing the servers $(\pi_1(1), \ldots, \pi_1(l_1))$ inside the tree network. We denote by $S_{j^\bullet}$ is the successor of S_j (successors are unique for tree networks), $j = 1, \ldots, n$ and denote the indices of the time variables accordingly. By abuse of notation, each server offers the service $S_j(s,t)$, $j = 1, \ldots, n$ for $0 \leq s \leq t$. Then, the foi sees the dynamic server for all $0 \leq t_{\pi_1(1)} \leq t_{\pi_1(l_1)+1}$*

$$S_{\text{e2e}}^{\text{PMOO}}(t_1, t_{\pi_1(l_1)+1}) = \left[\inf_{t_j \leq t_{j^\bullet}, j=1,\ldots,n} \left\{ \sum_{j=1}^{n} S_j\left(t_j, t_{j^\bullet}\right) - \sum_{i=2}^{m} A_i \left(t_{\pi_i(1)}, t_{\pi_i(l_i)^\bullet} \right) \right\} \right]^+ . \tag{5.3}$$

Proof. The proof is very similar to the one of Theorem 5.1. See also [BNS21]. □

5.2 PERFORMANCE BOUNDS

With the end-to-end service in Theorem 5.1, we can now derive performance bounds by unleashing PMOO. For the following theorem, we focus on the most important performance metric, end-to-end bounds on the delay's violation probability. We generalize the results in Theorem 3.14 from the canonical tandem to tree networks.

Theorem 5.3 (End-to-End Delay Bound under PMOO). *Assume a tree network with flows $f_1, \ldots, f_m$ and work-conserving servers $S_1, \ldots, S_n$. Assume a flow of interest (foi) traversing the servers $(\pi_1(1), \ldots, \pi_1(l_1))$ inside the tree network. By abuse of notation, each server offers the service $S_j(s,t)$, $j = 1, \ldots, n$ for $0 \leq s \leq t$. Let all respective arrivals A_i, $i = 1, \ldots, m$ be independent and $(\sigma_{A_i}, \rho_{A_i})$-bounded and all servers, $S_i, i = 1, \ldots, n$, be independent and $\left(\sigma_{S_j}, \rho_{S_j}\right)$-bounded, $j = 1, \ldots, n$. Let $\theta > 0$ and assume the stability condition*

$$\sum_{i: j \in \pi_i} \rho_{A_i}(\theta) < \rho_{S_j}(-\theta), \quad \textit{for all } j = 1, \ldots, n.$$

Moreover, define

$$\sigma_{\text{total}}(\theta) := \sum_{i=1}^{m} \sigma_{A_i}(\theta) + \sum_{j=1}^{n} \sigma_{S_j}(-\theta),$$

the residual rate

$$C_{\text{res},j}(-\theta) := \rho_{S_j}(-\theta) - \sum_{i \neq 1: j \in \pi_i} \rho_{A_i}(\theta), \tag{5.4}$$

and the minimum residual rate *(on the foi's path)*

$$C_{\min}(-\theta) := \min_{j\in\pi_1}\left\{\rho_{S_j}(-\theta) - \sum_{i\neq 1: j\in\pi_1} \rho_{A_i}(\theta)\right\} = \min_{j\in\pi_1}\left\{C_{\text{res},j}(-\theta)\right\}. \tag{5.5}$$

In order to take servers into account that are not on the foi's path, we also define:

$$W := \prod_{j\notin\pi_1} \frac{1}{1-e^{-C_{\text{res},j}(-\theta)}}.$$

(If the tree network is a tandem, it holds that $W = 1$.) Then, bounds on the delay's violation probability are given by

1.
$$\mathrm{P}(d(t) > T) \leq e^{-\theta\rho_{A_1}(\theta)\cdot T} e^{\theta\sigma_{\text{total}}(\theta)}\gamma \cdot W,$$
where
$$\gamma := \prod_{j\in\pi_1} \frac{1}{1-e^{\theta\left(\rho_{A_1}(\theta)-C_{\text{res},j}(-\theta)\right)}}.$$

2. *If*
$$T \geq \frac{l_1 e^{-\theta\left(C_{\min}(-\theta)-\rho_{A_1}(\theta)\right)}}{1-e^{-\theta\left(C_{\min}(-\theta)-\rho_{A_1}(\theta)\right)}},$$
then
$$\mathrm{P}(d(t) > T) \leq e^{-\theta C_{\min}(-\theta)\cdot T} e^{\theta\sigma_{\text{total}}(\theta)}\zeta^{l_1} \cdot W,$$
where
$$\zeta := \frac{\left(1+\frac{T}{l_1}\right)^{\left(1+\frac{T}{l_1}\right)}}{\left(\frac{T}{l_1}\right)^{\frac{T}{l_1}}}.$$

3. *Let us assume that $C_{\min}(-\theta) = \min_{j\in\pi_1}\left\{\rho_{S_j}(-\theta) - \sum_{i\neq 1: j\in\pi_i} \rho_{A_i}(\theta)\right\}$ is a unique minimum and attained for index j^*. Then,*
$$\mathrm{P}(d(t) > T) \leq e^{-\theta C_{\min}(-\theta)\cdot T} e^{\theta\sigma_{\text{total}}(\theta)} \frac{\psi}{1-e^{\theta\left(\rho_{A_1}(\theta)-C_{\min}(-\theta)\right)}} \cdot W,$$
where
$$\psi := \prod_{j^*\neq j\in\pi_1} \frac{1}{1-e^{\theta\left(C_{\min}(-\theta)-C_{\text{res},j}(-\theta)\right)}}.$$

Proof. The proof is given in Appendix C.1. □

If we fix the delay violation probability to ε and solve for T, we obtain the stochastic delay bounds

1.
$$T = \frac{\theta\sigma_{\text{total}}(\theta) + \log\left(\frac{1}{\varepsilon}\right) + \log(\gamma W)}{\theta\rho_{A_1}(\theta)}.$$

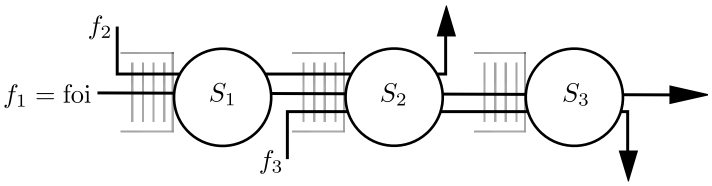


Figure 5.4: Overlapping tandem

2.
$$T = \frac{\theta\sigma_{\text{total}}(\theta) + \log\left(\frac{1}{\varepsilon}\right) + n \cdot \log(\zeta) + \log(W)}{\theta C_{\min}(-\theta)},$$

if
$$T \geq \frac{l_1 e^{-\theta\left(C_{\min}(-\theta) - \rho_{A_1}(\theta)\right)}}{1 - e^{-\theta\left(C_{\min}(-\theta) - \rho_{A_1}(\theta)\right)}}.$$

3.
$$T = \frac{\theta\sigma_{\text{total}}(\theta) + \log\left(\frac{1}{\varepsilon}\right) + \log\left(\frac{\psi W}{1 - e^{\theta\left(\rho_{A_1}(\theta) - C_{\min}(-\theta)\right)}}\right)}{\theta C_{\min}(-\theta)},$$

if $C_{\min}(-\theta) = \min_{j \in \pi_1}\left\{\rho_{S_j}(-\theta) - \sum_{i \neq 1 : j \in \pi_i} \rho_{A_i}(\theta)\right\}$ is a unique minimum.

5.3 NUMERICAL EVALUATION

In this section, we compute stochastic delay bounds applying state-of-the-art techniques as well as our unleashed PMOO. We perform several experiments for different network topologies. For the arrivals, we assume three discrete-time processes all adhering to the class of (σ_A, ρ_A)-constrained arrivals (see also Section 3.1):

- Independent exponentially distributed arrival increments with parameter λ,
- Independent Weibull distributed arrival increments with fixed shape parameter $k = 2$ and scale parameter λ,
- discrete-time Markov-modulated On-Off (MMOO) arrivals.

The latter can be described by three parameters: the probability to stay in the "On"-state in the next time step, p_{on}, the probability to stay in the "Off"-state, p_{off}, and a constant peak rate π_A, at which data is sent during the "On"-state. For the service, we always assume work-conserving servers with a constant rate.

5.3.1 *Overlapping tandem*

In our first numerical evaluation, we calculate stochastic delay bounds for a standard example in network calculus when PMOO effects shall be illustrated: the "overlapping tandem network" in Figures 5.4. These are compared to simulation results. Of course, simulation results cannot be conducted with "arbitrary scheduling". Therefore, we implemented the simulations with shortest-to-destination first (SDF) scheduling, since it can be used to characterize the worst case [BN15]. To provide a strong "standard bound", we combined several techniques known from the literature, such as

- a sequential separated flow analysis (seqSFA), where we apply each network calculus operation one after the other (see the discussion after Theorem 3.13 in Section 3.3) in contrast to a simultaneous technique as in Theorem 5.3 [BS13]:

$$S_{\text{e2e}}^{\text{seqSFA}} = [S_1 - A_2]^+ \otimes [S_2 - ((A_2 \oslash S_1) + A_3)]^+ \otimes \left[S_3 - A_3 \oslash [S_2 - (A_2 \oslash S_1)]^+\right]^+;$$

- an SFA where the deconvolution of the performance bound and the convolution of the service processes is done in one step (see Theorem 3.14); it is known that this avoids a sequencing penalty to some degree [Bec16a, pp. 33];
- an analysis that convolves servers as much as possible before subtracting cross-flows (sequential pay multiplexing only once (seqPMOO), see also Section 2.6 and Chapter 4):

$$S_{\text{e2e}}^{\text{seqPMOO,1}} = \left[\left(S_1 \otimes [S_2 - A_3]^+\right) - A_2\right]^+ \otimes \left[S_3 - A_3 \oslash [S_2 - (A_2 \oslash S_1)]^+\right]^+$$

  and

$$S_{\text{e2e}}^{\text{seqPMOO,2}} = [S_1 - A_2]^+ \otimes \left[\left([S_2 - (A_2 \oslash S_1)]^+ \otimes S_3\right) - A_3\right]^+;$$

  In [NS20a], this is called "PMOO", however, it is not fully able to pay multiplexing only once, as it involves a sequential order of network calculus operations. Clearly, a sequencing penalty is incurred again.

We then always pick the analysis that results in the best stochastic delay bound.

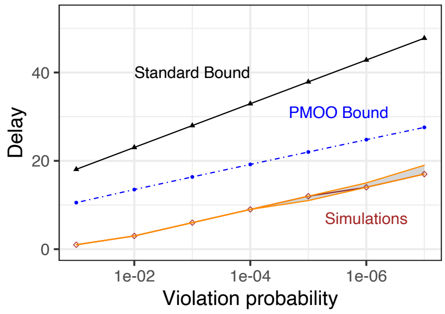
This set of state-of-the art bounding techniques is then compared to our new PMOO based on Theorem 5.3. For this topology, our new PMOO leads to (see Eqn. (5.1)):

$$S_{\text{e2e}}^{\text{PMOO}}(s,t) = \left[\inf_{s \le t_2 \le t_3 \le t} \{S_1(s,t_2) + S_2(t_2,t_3) + S_3(t_3,t) - A_2(s,t_3) - A_3(t_2,t)\}\right]^+.$$

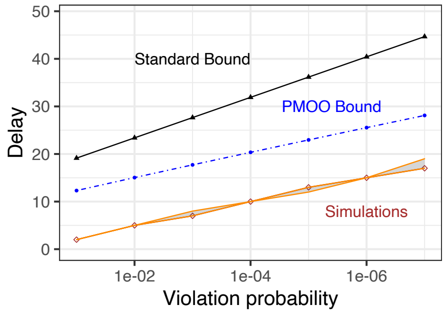
We calculate all three bounds and then pick the one with the tightest result. In addition, we also provide simulation results with respective pointwise 95%-confidence bands that are based on order statistics of a binomially distributed sample [HM11] to see how the bounds relate to the empirical delay. The results are provided in Figure 5.5.

We observe that the PMOO bound outperforms the standard bound significantly. For example, for a delay violation probability of 10^{-3}, the delay bound is improved from 28 to 18, and for a violation probability of 10^{-7}, from 45 to 31. These examples indicate an improvement of more than 35%.

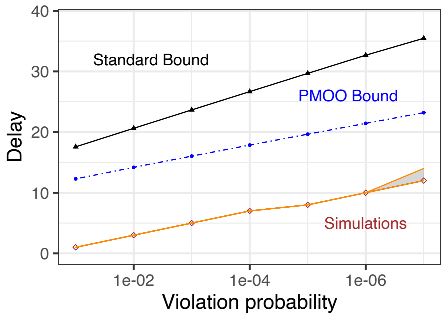
These positive results are mainly caused by the fact that our unleashed PMOO analysis is able to provide bounds without introducing any method-pertinent dependencies. The standard bound, in contrast, suffers from such



(a) Exponential distribution with $\lambda_i = 1.5$, $i = 1, 2, 3$



(b) Weibull distribution with $\lambda_i = 1.0$, $i = 1, 2, 3$



(c) MMOO with $p_{\text{on},i} = 0.5$, $p_{\text{off},i} = 0.5$, $\pi_{A_i} = 1.4$, $i = 1, 2, 3$

Figure 5.5: Stochastic delay bounds and simulation results for the overlapping tandem with constant rate servers and rates $C_1 = 2.5$, $C_2 = 3.0$, and $C_3 = 2.0$.

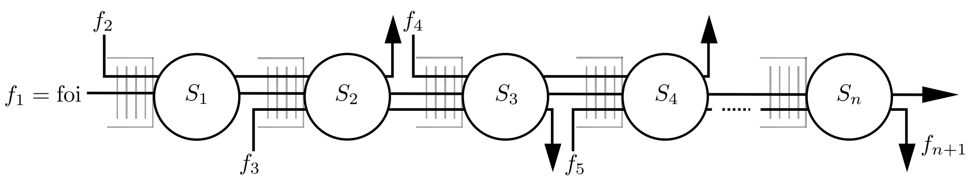


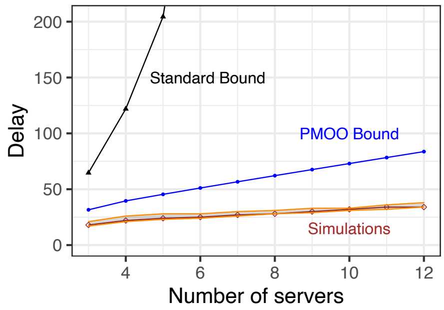
Figure 5.6: Extended overlapping tandem

a dependency in the calculation and, consequently, needs to apply Hölder's inequality. Furthermore, it also potentially loses accuracy due to the sequencing penalty.

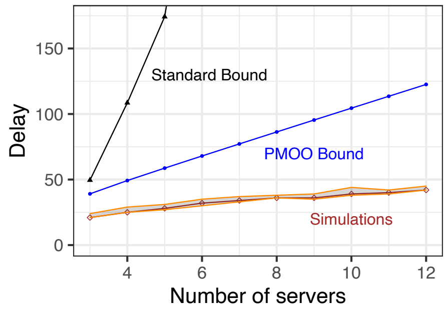
In comparison, we see that the gap between simulation results and bounds is considerably reduced and the scaling of the delay in the simulations is captured well (getting closer to single-node results again). Recalling that Figure 5.5a without the new bound has been used at the beginning of this chapter to illustrate that the known SNC gap will widen too much, we now provide a new prospect for this again.

5.3.2 *Extended overlapping tandem*

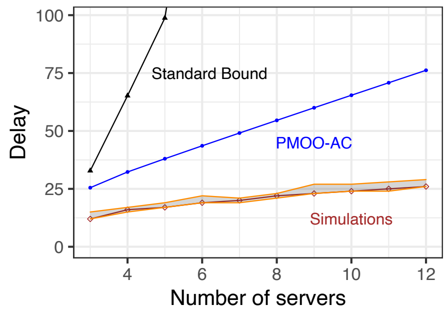
In the next experiment, we generalize the case of an overlapping tandem by varying its lengths while keeping the interference structure (Figure 5.6).



(a) Exponential distribution with $\lambda_i = 2.0$



(b) Weibull distribution with $\lambda_i = 0.7$



(c) MMOO with $p_{\text{on},i} = 0.5$, $p_{\text{off},i} = 0.5$, $\pi_{Ai} = 1.0$

Figure 5.7: Stochastic delay bounds for the extended overlapping tandem under constant rate servers with rates $C_i = 2.0$ for $i = 1, \ldots, 13$.

Here, the standard bound only includes the sequential and simultaneous SFA approaches:

$$
\begin{aligned}
& S_{\text{e2e}}^{\text{seqSFA}} \\
=& S_{1,\text{l.o.}} \otimes S_{2,\text{l.o.}} \otimes S_{3,\text{l.o.}} \otimes S_{4,\text{l.o.}} \otimes \cdots \otimes S_{n-1,\text{l.o.}} \otimes S_{n,\text{l.o.}} \\
=& [S_1 - A_2]^+ \otimes [S_2 - (A_2 \oslash S_1) + A_3]^+ \otimes \left[S_3 - \left(A_3 \oslash [S_2 - A_2 \oslash S_1]^+\right) + A_4\right]^+ \\
& \otimes \left[S_4 - \left(A_4 \oslash \left[S_3 - A_3 \oslash [S_2 - A_2 \oslash S_1]^+\right]^+\right) + A_5\right]^+ \otimes \cdots \\
& \otimes \left[S_{n-1} - \left(A_{n-1} \oslash \left[S_{n-2} - A_{n-2} \oslash [S_{n-3} - \cdots A_2 \oslash S_1]^+\right] + A_n\right)\right]^+ \\
& \otimes \left[S_n - \left(A_n \oslash \left[S_n - A_n \oslash [S_n - \cdots A_2 \oslash S_1]^+\right] + A_{n+1}\right)\right]^+ .
\end{aligned}
$$

The number of possible (combinations of) network calculus operations in the seqPMOO grows exponentially with the number of servers and, therefore, is computationally prohibitive.

We compute stochastic delay bounds for a fixed violation probability of 10^{-6}. The results are shown in Figure 5.7.

While the standard bound explodes in the number of servers (we have only included the results from 3 to 5 servers), the new technique scales significantly better. This is mainly due to the fact that the SFA leads to the application of $n-1$ Hölder inequalities for an overlapping tandem of size n, whereas the new PMOO does not incur any method-pertinent dependencies in the analysis of this topology.

Table 5.1: Runtimes [s] for the extended overlapping tandem (exponential distribution).

Tandem length	Standard bound	PMOO bound
3	4.846	0.012
4	182.397	0.012
5	8540.171	0.027
6	·	0.021
7	·	0.022
8	·	0.016
9	·	0.0215
10	·	0.023
11	·	0.023
12	·	0.025

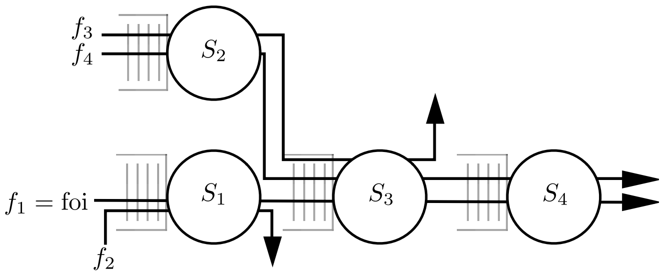


Figure 5.8: Tree network

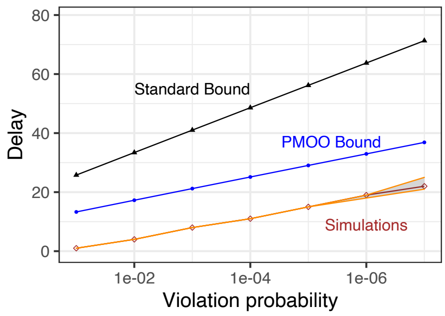
Not only does this lead to tighter performance bounds, it also impacts runtimes significantly as it reduces an n-dimensional non-linear optimization problem (one θ and $n-1$ Hölder parameters) to a 1-dimensional optimization problem.[1] We present and discuss only the results for the exponential distribution (Table 5.1), results for the other distributions are in the same orders of magnitude. Specifically, for the standard bound, runtimes increase quickly from 4.8 seconds (3 servers) over 3 minutes (4 servers) to almost 2.5 hours (5 servers). On the other hand, PMOO did not take longer than 0.027 seconds in all examples even though we have to compute 3 bounds.

5.3.3 *Case study: tree network*

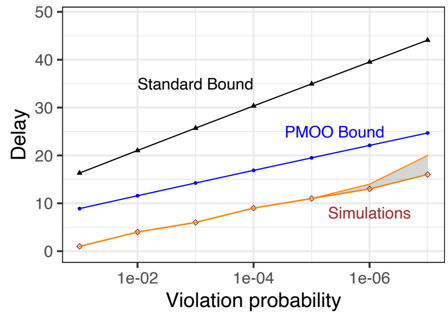
A typical network calculus analysis consists of three separate sequential steps:

1. Reducing the network to a tandem that is traversed by the flow of interest (foi), by invoking output bounds.
2. Reducing the tandem to an end-to-end server that represents the complete system.
3. Computation of performance bounds.

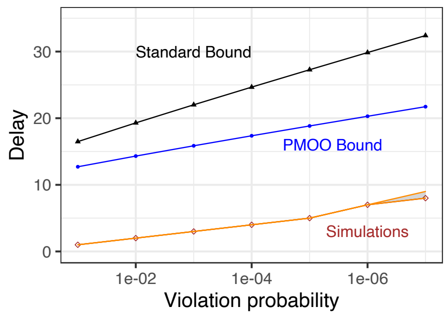
1 In our numerical experiments, we apply a grid search followed by a downhill simplex algorithm to optimize the parameters. For more details see Chapter 8.



(a) Exponential distribution with $\lambda_i = 2.0$



(b) Weibull distribution with $\lambda_i = 0.67$



(c) MMOO with $p_{\text{on},i} = 0.5$, $p_{\text{off},i} = 0.5$, $\pi_{A_i} = 1.0$

Figure 5.9: Stochastic delay bounds and simulation results for the tree network with constant rate servers and rates $C_i = 2.0$, $i = 1, \ldots, 4$.

It is already known that combining the last two steps can lead to tighter performance bounds [Bec16a, pp. 33]. However, to the best of our knowledge, simultaneously performing all three steps, as in the unleashed PMOO, has not been done before in SNC. The following case study shall mainly investigate the benefit of this.

To that end, we consider the tree network in Figure 5.8. Standard SNC techniques first compute the output bounds of flows 3 and 4 at server 2 in order to compute the residual service at servers 3 and 4:

$$
\begin{aligned}
&S_{\text{e2e}}^{\text{seqSFA}} \\
&= [S_1 - A_2]^+ \otimes [S_3 - (A_3 + A_4) \oslash S_2]^+ \\
&\otimes \left[S_4 - \left(\left(A_4 \oslash [S_2 - A_3]^+ \right) \oslash \left[S_3 - \left(\left(A_1 \oslash [S_1 - A_2]^+ \right) + \left(A_3 \oslash [S_2 - A_4]^+ \right) \right) \right]^+ \right) \right]^+ .
\end{aligned}
$$

This incurs again method-pertinent dependencies and therefore applications of Hölder's inequality. The application of Theorem 5.3 allows us to circumvent these dependencies. In contrast, state-of-the-art analysis using SFA needs three applications of Hölder's inequality and seqPMOO requires one, respectively:

$$
\begin{aligned}
&S_{\text{e2e}}^{\text{seqPMOO}} \\
&= [S_1 - A_2]^+ \otimes \left[\left(\left[S_3 - A_3 \oslash [S_2 - A_4]^+ \right]^+ \otimes S_4 \right) - \left(A_4 \oslash [S_2 - A_3]^+ \right) \right]^+ .
\end{aligned}
$$

The results are shown in Figure 5.9. Similar to the results for the overlapping tandem, we observe that the unleashed PMOO considerably improves delay bounds. Again, we are able to achieve a similar scaling compared to the simulation results, in contrast to the standard bound. Further, while the state-of-the-art analysis requires an optimization of up to 4 parameters (three Hölder and θ), the unleashed PMOO only has θ to optimize, substantially improving the runtime.

5.4 SUMMARY

We have presented a new network analysis method that unleashes the pay multiplexing only once (PMOO) principle in the stochastic network calculus. Equipped with this, it is now possible to calculate rigorous probabilistic performance bounds for tree networks without incurring any method-pertinent stochastic dependencies. In numerical evaluations, we observed that we are largely successful in reducing the known gap between simulations and bounds, and, at least, closely capture the scaling of the performance bounds.

While our method can benefit from improvements based on a preliminary network transformation (e.g., flow prolongation [NS20a]) at no cost, it might also be possible to integrate h-mitigators (Chapter 7 below).

6 DEALING WITH DEPENDENCE USING NEGATIVE DEPENDENCE

Results presented in this chapter are joint work with Jens Schmitt and Florin Ciucu [NSC19a, NSC19b].

In Chapter 5, we have presented a pay multiplexing only once (PMOO)-based analysis to obtain performance bounds for tree networks. Analyzing more general networks, e.g., feedforward networks however, requires to consider also dependent flows at some points in the network, as the sharing of a resource clearly has a mutual impact on the flows' output behavior. Therefore, if we want to obtain the moment-generating function (MGF) of aggregated, yet dependent arrival processes $A_1(s,t)$ and $A_2(s,t)$, we typically invoke Hölder's inequality (Eqn. (3.22)): Let $\theta > 0$ and let $p, q > 1$ be Hölder conjugates. Then,

$$\mathrm{E}\left[e^{\theta(A_1(s,t)+A_2(s,t))}\right] \leq \mathrm{E}\left[e^{p\theta A_1(s,t)}\right]^{\frac{1}{p}} \cdot \mathrm{E}\left[e^{q\theta A_2(s,t)}\right]^{\frac{1}{q}}, \quad \text{for all } 0 \leq s \leq t.$$

Hölder's inequality is completely oblivious of the actual dependence structure, thus it often leads to very conservative bounds. Furthermore, it places the burden of an additional, nonlinear parameter for each application to optimize.

Dependence of arrivals does not have to be a negative property per se. Taking advantage of the information about the dependence structure to improve upon the bounds has been attempted before. In [DWS15, Don17], the functional dependence is estimated using a copula-based approach. In our work, we investigate a simpler alternative, using the independent scenario as an upper bound. To that end, we rely on a characteristic called *negative dependence*. We explain the main idea with the help of the following, simplistic example.

Consider a single time slot assuming two arrival processes, A_1 and A_2, that are multiplexed at one server. Both arrivals send one packet, each independently with probability $p \in (0,1)$, and the server serves one packet but strictly prioritizes A_2. Clearly, their two outputs, D_1 and D_2, are strongly dependent, as an arrival of the prioritized flow forces the other one to wait in the queue. Simply put, if one flow gets a larger share of the server's capacity, the other is more likely to have less output. For the joint distribution of the output, we have by assumption for the departures both being equal to 1, that $\mathrm{P}(D_1 = 1, D_2 = 1) = 0$. On the other hand, we compute for the product distribution by a simple conditioning, that $\mathrm{P}(D_1 = 1) \cdot \mathrm{P}(D_2 = 1) = (p \cdot (1-p)) \cdot (1-p) > 0$. Hence, if we deliberately forego the knowledge about the dependence structure, we only obtain an upper bound, yet, it allows us to consider just the marginal distributions.

OUTLINE The rest of the chapter is structured as follows. Section 6.1 introduces the necessary background on negative dependence. Section 6.2 contains the main results obtained in two case studies assuming a conjecture on dependence. The numerical evaluation is presented in Section 6.3. Section 6.4 discusses the chapter.

6.1 NEGATIVE DEPENDENCE AND ACCEPTABLE RANDOM VARIABLES

As we discussed in the introduction, we would like to bound the joint distribution of two random variables by their respective product distribution. This concept was captured in the 1960s by Lehmann and his notion of negative dependence.

Definition 6.1 (Negative Dependence [Leh66])**.** A finite family of random variables $\{X_1, \ldots, X_n\}$ is said to be *negatively (orthant) dependent (ND)* if the two following inequalities hold:

$$\mathrm{P}(X_1 \leq x_1, \ldots, X_n \leq x_n) \leq \prod_{i=1}^{n} \mathrm{P}(X_i \leq x_i)\,,$$
$$\mathrm{P}(X_1 > x_1, \ldots, X_n > x_n) \leq \prod_{i=1}^{n} \mathrm{P}(X_i > x_i)\,,$$

for all $(x_1, \ldots, x_n) \in \mathbb{R}^n$.

The following lemma shows how this characteristic can be used directly in the context of MGFs.

Lemma 6.2 ([JDP83, Sun11])**.** *Let* $\theta > 0$*. If* $\{X_1, \ldots, X_n\}$ *is a set of ND random variables, then*

$$\mathrm{E}\left[e^{\theta \sum_{i=1}^{n} X_i}\right] \leq \prod_{i=1}^{n} \mathrm{E}\left[e^{\theta X_i}\right]. \tag{6.1}$$

In other words, treating the aggregate of ND random variables as if they were independent yields an upper bound for the respective MGFs. Random variables that suffice Eqn. (6.1) are called "acceptable" [AKV08], but are studied in an unrelated context.

Proving that random variables are negatively dependent is a challenging task. Some results exist, e.g., the multinomial and multivariate hypergeometric distribution are ND, or the "Zero-One Lemma" [DR98], which proves the property for $X_1, \ldots, X_n \in \{0, 1\}$ such that $\sum_i X_i = 1$. This means that the output processes in the example at the beginning of this chapter are indeed ND. Furthermore, it has been shown a related result in [JDP83] that a permutation distribution, and therefore random sampling without replacement, is ND. This result has been used to prove near-perfect load balancing for switches called "Sprinklers" [Din+14]. In our context, this provides a result for a single time slot. In the following, we confine ourselves to conjecture this property for intervals.

Conjecture 6.3. *Let two independent flows with according arrival processes* A_1 *and* A_2 *traverse a work-conserving server with finite capacity. Further, both arrival processes have iid increments.*

Then, we assume their respective output processes $D_1(s, t)$ *and* $D_2(s, t)$ *to be ND for all* $0 \leq s \leq t$*.*

We do not have a proof but Conjecture 6.3 held in all our experiments using 10^6 samples to estimate the joint and product (complementary) cumulative distribution functions (CDFs), respectively: For two flows at one server, we tried over 5500 different combinations of intervals, x_1, x_2, (as in the CDF), utilizations

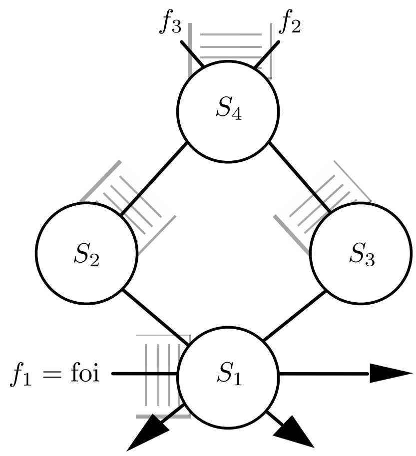


Figure 6.1: Diamond network

(between 0.4 and 0.9), and random packet sizes that were drawn from either exponential, Weibull, Gumbel, or log-normal distribution.

The focus on the same interval for both process is important, as the following, admittedly simplifying, argument suggests: Assume the high priority (HP) flow to send a lot of packets consecutively, i.e., the low priority (LP) flow has no output in this period and queues all its packets. Then, it is more likely for the LP flow to have outputs when the HP flow stops sending, as it is more likely for it to have queued packets.

6.2 INDEPENDENCE AS A BOUND

In this section, we investigate two case studies to show in which part of the analysis we exploit the negative dependence.

In the following, we consider the flow f_1 to be the flow of interest (foi) whose delay we stochastically upper bound. All arrival processes A_i are assumed to be discrete time and to have iid increments and all servers S_j are work-conserving and provide a constant rate $C_j \geq 0$. To simplify notation, we denote again by D_i^j the output of flow i at server S_j.

6.2.1 *Diamond network*

In this case study, we consider the topology in Fig. 6.1. Assume the foi to have the lowest priority and f_3 to have the highest priority. Moreover, we assume the rates of S_2 and S_3, C_2 and C_3, to be equal. By SNC literature [CBL06, Fid06], the end-to-end service provided for the flow of interest, also known as the network service curve, can be described by $S_{\text{e2e}} = S_{\text{e2e}}^{\text{seqPMOO}} = S_{\text{e2e}}^{\text{seqSFA}}$ with

$$S_{\text{e2e}} = \left[S_1 - \left(\left(\left(A_2 \oslash [S_4 - A_3]^+\right) \oslash S_2\right) + \left(\left(A_3 \oslash S_4\right) \oslash S_3\right)\right)\right]^+.$$

Since Conjecture 6.3 is made on output processes, we postpone the application of the output bound in Eqn. (2.21) by keeping the exact output at first. That is, we start with

$$S_{\text{e2e}}^{\text{ND}} = \left[S_1 - \left(D_2^2 + D_3^3\right)\right]^+, \tag{6.2}$$

use then the conjecture to bound the MGF of the aggregate by their product (Eqn. (6.1)), and apply the output bound in a final step.

The probability that the delay process $d(t)$ exceeds a value $T \geq 0$ is upper bounded by

$$
\begin{aligned}
&\mathrm{P}(d(t) > T) \\
&\overset{(3.26)}{\leq} \sum_{t_0=0}^{t-1} \mathrm{E}\left[e^{\theta A_1(t_0,t)}\right] \mathrm{E}\left[e^{-\theta S_{\mathrm{e2e}}(t_0,t+T)}\right] \\
&\overset{(6.2)}{=} \sum_{t_0=0}^{t-1} \mathrm{E}\left[e^{\theta A_1(t_0,t)}\right] \mathrm{E}\left[e^{-\theta \left[S_1 - \left(D_2^2 + D_3^3\right)\right]^+ (t_0,t+T)}\right] \\
&\leq \sum_{t_0=0}^{t-1} \mathrm{E}\left[e^{\theta A_1(t_0,t)}\right] e^{-\theta C_1 (t+T-t_0)} \mathrm{E}\left[e^{\theta \left(D_2^2 + D_3^3\right)(t_0,t+T)}\right], \qquad (6.3)
\end{aligned}
$$

where we used Theorem 3.9 in the first inequality. Since the flows f_2 and f_3 share the server S_4, their according output processes D_2^4 and D_3^4 are dependent and, as a consequence, D_2^2 and D_3^3, as well (note that we assumed $C_2 = C_3$). However, by the conjecture above, we assume that the resource sharing at S_4 indicates that the dependence on $[t_0, t+T]$ is negative which, in turn, is the reason why we upper bound their joint MGF by the product of the marginal MGFs.

This can be interpreted as if we analyzed a new system, where the server S_4 would be split into two servers. That is, one provides the same service as the original (for the high priority flow f_3), and the other provides the leftover service $[S_4' - A_3']^+$, where S_4' has the same service rate as S_4 and A_3' is a new arrival process, but with the same distribution as A_3.

Hence, the second factor is upper bounded by

$$
\begin{aligned}
\mathrm{E}\left[e^{\theta \left(D_2^4 + D_3^4\right)(t_0,t+T)}\right] &\leq \mathrm{E}\left[e^{\theta D_2^4 (t_0,t+T)}\right] \mathrm{E}\left[e^{\theta D_3^4 (t_0,t+T)}\right] \\
&\leq \mathrm{E}\left[e^{\theta \left(A_2 \oslash [S_4 - A_3]^+\right)(t_0,t+T)}\right] \mathrm{E}\left[e^{\theta (A_3 \oslash S_4)(t_0,t+T)}\right].
\end{aligned}
$$

Further assuming all A_i to be (σ_A, ρ_A)-bounded yields a closed-form for the delay bound under stability:

$$
\begin{aligned}
\mathrm{P}(d(t) > T) \leq &\frac{e^{\theta\left(\left(\rho_{A_2}(\theta) + \rho_{A_3}(\theta) - C_1\right)T + \sigma_1(\theta) + \sigma_{A_2}(\theta) + 2\sigma_{A_3}(\theta)\right)}}{1 - e^{\theta\left(\rho_{A_1}(\theta) + \rho_{A_2}(\theta) + \rho_{A_3}(\theta) - C_1\right)}} \\
&\cdot \frac{1}{1 - e^{\theta\left(\rho_{A_2}(\theta) - C_2\right)}} \cdot \frac{1}{1 - e^{\theta\left(\rho_{A_3}(\theta) - C_3\right)}} \\
&\cdot \frac{1}{1 - e^{\theta\left(\rho_{A_2}(\theta) + \rho_{A_3}(\theta) - C_4\right)}} \cdot \frac{1}{1 - e^{\theta\left(\rho_{A_3}(\theta) - C_4\right)}}.
\end{aligned}
$$

For detailed calculations we refer to Appendix D.1. For the sake of a uniform presentation, we bound all sums by the geometric series instead of the integral bound as in Theorem 3.9.

In contrast, standard techniques proceed at Eqn. (6.3) by applying the output Bound Eqn. (2.21) immediately and continue with Hölder's inequality to deal with the dependence.

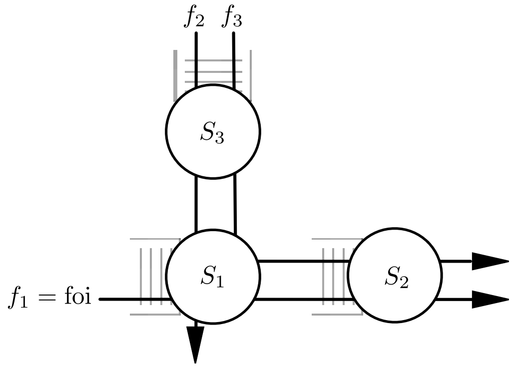


Figure 6.2: The $\mathbb{L}$

6.2.2 *The* $\mathbb{L}$

In this case study, we analyze the topology in Fig. 6.2. The foi has the lowest priority and f_2 the highest. Similarly to Subsection 6.2.1, we assume the outputs processes of f_2 and f_3 to be ND, based on Conjecture 6.3. Here, the end-to-end service is

$$S_{\text{e2e}}^{\text{seqPMOO}} = \left[\left([S_1 - (A_2 \oslash S_3)]^+ \otimes S_2\right) - \left(A_3 \oslash [S_3 - A_2]^+\right)\right]^+.$$

Again, we postpone the output bounding and start with

$$S_{\text{e2e}}^{\text{ND}} = \left[\left(\left[S_1 - D_2^3\right]^+ \otimes S_2\right) - D_3^3\right]^+. \tag{6.4}$$

The crucial difference is that, in order to obtain a bound on foi's delay, the min-plus convolution (Eqn. (3.14)) has to be applied to the service processes of S_1 and S_2 forcing us to analyze the output processes at different intervals:

$$\begin{aligned}
&\mathrm{P}(d(t) > T) \\
\overset{(3.26)}{\leq}& \sum_{t_0=0}^{t-1} \mathrm{E}\left[e^{\theta A_1(t_0,t)}\right] \mathrm{E}\left[e^{-\theta S_{\text{e2e}}(t_0,t+T)}\right] \\
\overset{(6.4)}{=}& \sum_{t_0=0}^{t-1} \mathrm{E}\left[e^{\theta A_1(t_0,t)}\right] \mathrm{E}\left[e^{-\theta\left[\left(\left[S_1 - D_2^3\right]^+ \otimes S_2\right) - D_3^3\right]^+ (t_0,t+T)}\right] \\
\leq& \sum_{t_0=0}^{t-1} \mathrm{E}\left[e^{\theta A_1(t_0,t)}\right] \sum_{t_1=t_0}^{t+T} \mathrm{E}\left[e^{\theta D_3^3(t_0,t+T)} e^{-\theta\left[S_1 - D_2^3\right]^+ (t_0,t_1)} e^{-\theta S_2(t_1,t+T)}\right] \\
\leq& \sum_{t_0=0}^{t-1} \mathrm{E}\left[e^{\theta A_1(t_0,t)}\right] \sum_{t_1=t_0}^{t+T} e^{-\theta C_1 \cdot (t_1 - t_0)} e^{-\theta C_2 \cdot (t+T-t_1)} \, \mathrm{E}\left[e^{\theta D_3^3(t_0,t+T)} e^{\theta D_2^3(t_0,t_1)}\right],
\end{aligned}$$

where we used the Union bound for each application of the convolution / deconvolution. This scenario is not covered by Conjecture 6.3 (see also the discussion at the end of Section 6.1). Our workaround is to leverage the monotonicity of D_2^3:

$$\mathrm{E}\left[e^{\theta D_3^3(t_0,t+T)} e^{\theta D_2^3(t_0,t_1)}\right] \leq \mathrm{E}\left[e^{\theta D_3^3(t_0,t+T)} e^{\theta D_2^3(t_0,t+T)}\right].$$

The rest of the analysis employs similar techniques as for the diamond network. See also Appendix D.1. Under the assumption of (σ_A, ρ_A)-bounded arrivals,

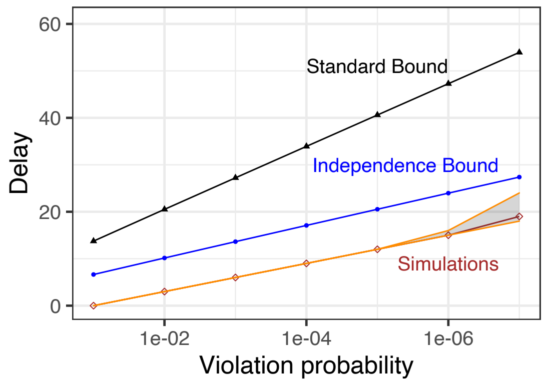


Figure 6.3: Stochastic delay bounds for the diamond network ($\lambda_1 = 7.4, \lambda_2 = 7.7, \lambda_3 = 6.3, C_1 = 0.6, C_2 = C_3 = 5.5, C_4 = 1.5$).

we obtain again a closed form for a bound on the delay's violation probability under stability:

$$P(d(t) > T) \leq \frac{e^{\theta\left(\left(\rho_{A_2}(\theta)+\rho_{A_3}(\theta)-\min\{C_1,C_2\}\right)\cdot T+\sigma_{A_1}(\theta)+2\sigma_{A_2}(\theta)+\sigma_{A_3}(\theta)\right)}}{1-e^{\theta\left(\rho_{A_1}(\theta)+\rho_{A_2}(\theta)+\rho_{A_3}(\theta)-\min\{C_1,C_2\}\right)}} \cdot \frac{1}{1-e^{\theta\left(\rho_{A_2}(\theta)-C_3\right)}} \cdot \frac{1}{1-e^{\theta\left(\rho_{A_2}(\theta)+\rho_{A_3}(\theta)-C_3\right)}} \cdot \frac{1}{1-e^{-\theta|C_1-C_2|}}.$$

6.3 NUMERICAL EVALUATION

We present the results of a numerical evaluation for both case studies. We ran 10^4 Monte-Carlo simulations to sample the parameters for different server rates and packet sizes, the latter sampled from an exponential distribution. The scenarios are then filtered to ensure a utilization $\in [0.5, 1)$.

6.3.1 *Quality of the bounds*

Diamond Network: This topology, after above the mentioned filtering, yields 485 remaining scenarios, of which 371 are improved. The fact that not all are improved despite the avoidance of Hölder's inequality can be explained as follows: In the analysis, the Union bound is applied after Hölder's inequality. The exponentiation before the summing followed by a square root can have a "mitigating" effect. We exploit a similar observation in Chapter 7.

We also measured the extent of the improvement by computing the ratio of the delay violation probability of the standard approach over the "independence bound":

$$\frac{\text{Standard bound}}{\text{Independence bound}}. \tag{6.5}$$

Clearly, values above 1 are desirable. Here, we obtain a median improvement of 6.04. In Fig. 6.3, we depict the delay bounds for specific parameters.

The $\mathbb{L}$: For this topology, we expect a weaker performance, as our approach using independence as a bound requires the additional step of extending the interval of one output process. The numerical results confirm this expectation: Out of the 729 scenarios, only half of them (384) yield a performance gain

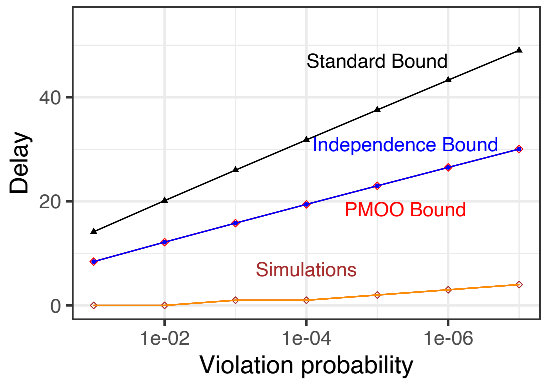


Figure 6.4: Stochastic delay bounds for the $\mathbb{L}$ ($\lambda_1 = 2.8, \lambda_2 = 3.4, \lambda_3 = 5.1, C_1 = 1.1, C_2 = 7.7, C_3 = 6.6$).

over the standard bound. Moreover, in contrast to the diamond network, we can directly apply the unleashed pay multiplexing only once (PMOO) from Chapter 5 for trees. The two bounds (independence bound and PMOO) exhibit only very small gaps. Given that the PMOO has a rigorous proof (Theorem 5.2), this gives numerical evidence that the independence bound could also be valid. The median of the improvement ratio confirms this, being relatively close to 1 (1.27). Again, we show the delay bounds for fixed parameters (Fig. 6.4).

6.3.2 *Computation runtime*

Our proposed approach does not only often substantially improve the bounds but it also has a much lower computation complexity than the standard approach. The reason is that the latter relies on an additional Hölder parameter. The optimizations are conducted using a grid search followed by a downhill simplex algorithm. The improvement ratios are in the median 337.5 (1.62 sec compared to 0.0048 sec) for the diamond scenario and 458.1 for the $\mathbb{L}$ (1.42 sec compared to 0.0031 sec). These improvements due to the reduction of the optimization parameters indicates a significant potential for an analysis of larger networks, as the optimization step in the MGF-based SNC can severely limit its scalability.

6.4 DISCUSSION

In this chapter, we found interesting results indicating that by using independence as a bound, one can often times improve the delay bound while also speeding up the runtime significantly. Obviously, the crucial next step is to find scenarios in which the conjecture can be proved rigorously. One potential technique might be to use the coupling method [Thoo0, Lin02], as it is can be applied to derive relations between tail probabilities.

Part III

END-TO-END ANALYSIS

7

h-MITIGATORS

Results presented in this chapter are joint work with Jens Schmitt and Malte Schütze [NS18, NSS19].

Typically, a DNC/SNC network analysis proceeds along the following steps:

1) Reducing the network to a tandem of servers traversed by the flow of interest (foi) by invoking the output bound calculation to characterize cross-traffic flows at the servers where they join the foi.

2) Reducing the tandem of servers traversed by the foi to a single server representing the whole system.

3) Calculating the delay bound of the foi at the single server representing the whole system.

Most of the existing NC literature has mainly focused on steps 2) and 3). In DNC, step 1) has seen some advanced treatment recently [BNS17a], but in SNC it has been largely neglected in the sense that no work beyond the standard output bound calculation was invested. In contrast to this, we focus on step 1) and, in particular, try to improve the SNC output bound calculation in this chapter. As the output bound calculation has to be invoked numerous times in step 1), we believe its accuracy to be key in larger network analyses. For example: assume a full binary tree of height h where each node represents a server and each of these servers has an arrival flow that is transmitted to the sink; let the foi be starting from one of the leaf nodes (see also Figure 7.1), then the number of output bound calculations is $2^h - h - 1$, whereas we only need to invoke the delay bound calculation once (in step 3)). Thus, any improvement in the output bound calculation pays off tremendously in larger network analyses.

In this chapter, we present a modification of the MGF-based SNC that mitigates the Union bound's effect in the output bound calculation. It consists of the application of Jensen's inequality via a convex function h just before the invocation of the Union bound and does not impose any additional assumptions. It is thus minimally invasive and, using the power function for h, all existing results and procedures of the SNC are literally still applicable while, as we see below, it improves the performance bounds. In fact, we prove this new bound with the power function, the so-called "power-mitigator", to be always at least as good as the state-of-the-art method. Evaluations in a very simple heterogeneous two-server setting show that it can improve the delay's violation probability already by a factor of up to 478.9.

It comes, however, at the price of an additional parameter per invocation of Jensen's inequality. Thus, we trade higher computational effort in the optimization of these parameters for improved bounds. However, as we also show this effort is moderate if the optimization is done carefully.

In [NSS19], we also investigate an alternative h-mitigator using the exponential function. Yet, it does not lead to an improvement of the SNC analysis.

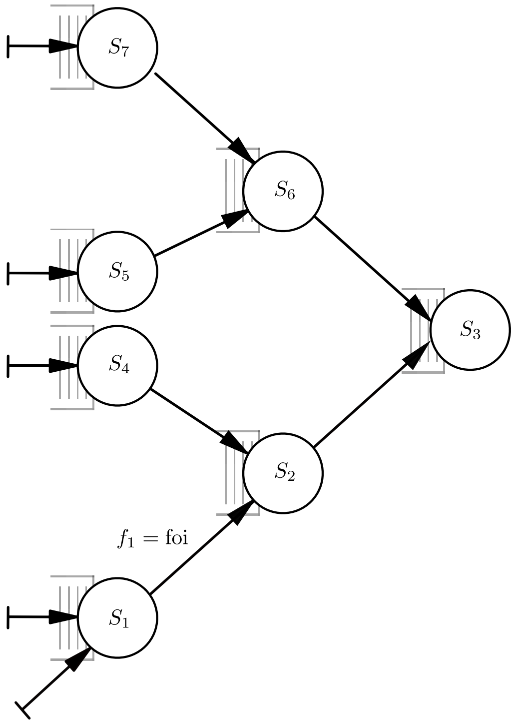


Figure 7.1: Full binary sink tree with seven nodes.

OUTLINE The rest of the chapter is structured as follows: In Section 7.1, we present our new output bound calculation and prove its validity. A numerical evaluation is given in Section 7.2: we compare output bounds for a single server and delay bounds for a two-server setting as well as a fat tree topology with the current state-of-the-art method. In Section 7.3, we prove that Jensen's inequality with the power function cannot be applied directly to delay bounds. Section 7.4 summarizes the chapter.

7.1 NEW OUTPUT BOUND CALCULATION

In this section, we derive our new approach to compute the MGF-output bound. Furthermore, we apply this idea to (σ, ρ)-bounded arrivals and service.

7.1.1 *Insertion of Jensen's inequality*

The standard approach to bound the output-MGF (Theorem 3.12) is as follows:

$$\begin{aligned} \mathrm{E}\left[e^{\theta D(s,t)}\right] &\overset{(3.21)}{\leq} \mathrm{E}\left[e^{\theta A \oslash S\,(s,t)}\right] \\ &= \mathrm{E}\left[e^{\theta \sup_{0\leq\tau\leq s}\{A(\tau,t)-S(\tau,s)\}}\right] \\ &\leq \sum_{\tau=0}^{s} \mathrm{E}\left[e^{\theta(A(\tau,t)-S(\tau,s))}\right], \end{aligned} \tag{7.1}$$

where the $\sup\{\cdot\}$ is always less than or equal to the sum since we have only non-negative terms. Eqn. (7.1) is similar to the application of the Union bound[1],

$$P\left(\sup_{i=1,\dots,n}\{X_i\} > a\right) \overset{(3.28)}{\leq} \sum_{i=1}^{n} P(X_i > a)\,.$$

The idea in the following is to insert Jensen's inequality to mitigate the inaccuracy imposed by Eqn. (7.1). Therefore, we call this approach in the following "h-mitigator". Nonetheless, this approach is able to preserve end-to-end analyses for certain functions h, as we show in the subsequent Subsection 7.1.2.

Theorem 7.1 (Jensen's Inequality). *Suppose that h is a differentiable convex function on $\mathbb{R}$ and let $X \in \mathcal{L}^1$. Then*

$$h(E[X]) \leq E[h(X)]\,. \tag{7.2}$$

Proof. See, e.g., [Nel95, pp. 176]. □

This enables us to prove a new output bound.

Proposition 7.2 (h-Mitigator). *Let $h_p : \mathbb{R}^+ \to \mathbb{R}^+$ be a differentiable, strictly increasing, and convex function with parameter p over a set $P \subset \mathbb{R}$.*

1. *It holds that*

$$E\left[e^{\theta D(s,t)}\right] \leq \inf_{p\in P}\left\{h_p^{-1}\left(\sum_{\tau=0}^{s} E\left[h_p\left(e^{\theta(A(\tau,t)-S(\tau,s))}\right)\right]\right)\right\}. \tag{7.3}$$

2. *If we additionally assume that h_p is the identity function for $\bar{p} \in P$, i.e.,*

$$h_{\bar{p}}(x) = x, \quad \forall x \geq 0, \tag{7.4}$$

then this bound is always at least as good as the standard approach in Eqn. (7.1).

Proof. We know by Jensen's inequality that

$$\begin{aligned} E\left[e^{\theta D(s,t)}\right] &\overset{(3.21)}{\leq} E\left[e^{\theta \sup_{0\leq\tau\leq s}\{A(\tau,t)-S(\tau,s)\}}\right] \\ &\overset{(7.2)}{\leq} \inf_{p\in P}\left\{h_p^{-1}\left(E\left[h_p\left(e^{\theta \sup_{0\leq\tau\leq s}\{A(\tau,t)-S(\tau,s)\}}\right)\right]\right)\right\} \\ &= \inf_{p\in P}\left\{h_p^{-1}\left(E\left[\sup_{0\leq\tau\leq s} h_p\left(e^{\theta(A(\tau,t)-S(\tau,s))}\right)\right]\right)\right\} \\ &\leq \inf_{p\in P}\left\{h_p^{-1}\left(\sum_{\tau=0}^{s} E\left[h_p\left(e^{\theta(A(\tau,t)-S(\tau,s))}\right)\right]\right)\right\}, \end{aligned} \tag{7.5}$$

1 For probability bounds such as the backlog or the delay, it is even equivalent to the Union bound, as

$$\begin{aligned} &P\left(\sup_{i=1,\dots,n} X_i > a\right) \overset{(3.28)}{\leq} \sum_{i=1}^{n} P(X_i > a) \overset{(3.3)}{\leq} e^{-\theta a}\sum_{i=1}^{n} E\left[e^{\theta X_i}\right] \\ \Leftrightarrow\ &P\left(\sup_{i=1,\dots,n} X_i > a\right) \overset{(3.3)}{\leq} e^{-\theta a} E\left[\sup_{i=1,\dots,n} e^{\theta X_i}\right] \overset{(7.1)}{\leq} e^{-\theta a}\sum_{i=1}^{n} E\left[e^{\theta X_i}\right] \end{aligned}$$

Therefore, we call the inequality in Eqn. (7.1) in the following "quasi-Union bound."

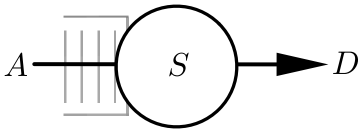


Figure 7.2: One server topology.

where we used that strictly increasing functions on $\mathbb{R}$ always have an inverse in the second line and the quasi-Union bound in the last inequality. This proves the first part of the proposition.

For the second part, we simply observe

$$\inf_{p \in P} \left\{ h_p^{-1} \left(\sum_{\tau=0}^{s} \mathrm{E}\left[h_p \left(e^{\theta(A(\tau,t)-S(\tau,s))} \right) \right] \right) \right\} \leq \sum_{\tau=0}^{s} \mathrm{E}\left[e^{\theta(A(\tau,t)-S(\tau,s))} \right],$$

where we used that there is one $\bar{p} \in P$ such that Eqn. (7.4) holds. This finishes the proof. □

The goal of the parameterization of h_p is to enable a whole set of functions that, ideally, lead to tighter bound as well as the possibility to provide a guarantee to not worsen the bound as in Proposition 7.2.2.

7.1.2 *Power-mitigator and (σ, ρ)-bounds*

In this subsection, we show that Proposition 7.2 generalizes the results in [NS18]. Moreover, we restate the compatibility with the (σ, ρ)-bounds in Definition 3.2.

Proposition 7.2 yields an output bound given a parameterized function h_p. A suitable candidate for h_p is the power function

$$\begin{aligned} h_p : \mathbb{R}_+ &\to \mathbb{R}_+, \\ x &\mapsto x^p, \end{aligned} \tag{7.6}$$

where $p \geq 1$, because it suffices the necessary conditions of both parts of Proposition 7.2 being differentiable, strictly increasing, convex, and is the identity for $p = 1$. Hence, we call the h-mitigator with this choice the "power-mitigator".

Corollary 7.3 (Power-Mitigator)**.** *Let h_p be defined as in Eqn. (7.6). Then it holds that*

$$\begin{aligned} \mathrm{E}\left[e^{\theta D(s,t)} \right] &\leq \inf_{p \geq 1} \left\{ \left(\sum_{\tau=0}^{s} \mathrm{E}\left[e^{p\theta(A(\tau,t)-S(\tau,s))} \right] \right)^{\frac{1}{p}} \right\} \\ &\leq \sum_{\tau=0}^{s} \mathrm{E}\left[e^{\theta\{A(\tau,t)-S(\tau,s)\}} \right], \end{aligned} \tag{7.7}$$

i.e., we receive a new output bound Eqn. (7.7) that guarantees to be as good as the standard approach in Eqn. (7.1).

Here, we see that the subadditivity of the root function implies that the insertion can mitigate the effect of the quasi-Union bound (7.1).

SINGLE SERVER SETTING Assume now a single flow - single server setting as in Figure 7.2. We have already deduced that

$$\begin{aligned} \mathrm{E}\left[e^{\theta D(s,t)}\right] &\leq \mathrm{E}\left[e^{\theta A \oslash S\,(s,t)}\right] \\ &\overset{(7.1)}{\leq} \sum_{\tau=0}^{s} \mathrm{E}\left[e^{\theta(A(\tau,t)-S(\tau,s))}\right]. \end{aligned}$$

We now require the arrivals and service to have (σ, ρ)-constraints (Definition 3.2). Under stability, $\rho_A(\theta) < \rho_S(-\theta)$, the standard approach leads to

$$\begin{aligned} \mathrm{E}\left[e^{\theta D(s,t)}\right] &\leq \sum_{\tau=0}^{s} \mathrm{E}\left[e^{\theta A(\tau,t)}\right] \mathrm{E}\left[e^{-\theta S(\tau,s)}\right] \\ &\leq \sum_{\tau=0}^{s} e^{\theta\rho_A(\theta)(t-\tau)+\theta\sigma_A(\theta)} e^{-\theta\rho_S(-\theta)(s-\tau)+\theta\sigma_S(-\theta)} \\ &\leq e^{\theta\rho_A(\theta)(t-s)} \frac{e^{\theta(\sigma_A(\theta)+\sigma_S(-\theta))}}{1-e^{\theta(\rho_A(\theta)-\rho_S(-\theta))}}, \end{aligned} \tag{7.8}$$

where we have used the independence of arrivals and service in the first line, (σ, ρ)-bounds in the second line and the convergence of the geometric series in the last line. This shows that the output is (σ, ρ)-bounded as well (see Proposition 7.4 below).

If we use Jensen's inequality with the power function Eqn. (7.7) instead, we obtain in comparison

$$\begin{aligned} \mathrm{E}\left[e^{\theta D(s,t)}\right] &\leq \inf_{p \geq 1} \left\{ \left(\frac{e^{p\theta(\rho_A(p\theta)(t-s)+\sigma_A(p\theta)+\sigma_S(-p\theta))}}{1-e^{p\theta(\rho_A(p\theta)-\rho_S(-p\theta))}} \right)^{\frac{1}{p}} \right\} \\ &= \inf_{p \geq 1} \left\{ \frac{e^{\theta(\rho_A(p\theta)(t-s)+\sigma_A(p\theta)+\sigma_S(-p\theta))}}{\left(1-e^{p\theta(\rho_A(p\theta)-\rho_S(-p\theta))}\right)^{\frac{1}{p}}} \right\}. \end{aligned} \tag{7.9}$$

Thus, the power-mitigator can also be used under (σ, ρ)-constraints (see Proposition 7.5). That is, it can easily be integrated in existing end-to-end analyses. Let us restate Theorem 3.12 for independent processes first.

Proposition 7.4. *Consider a (σ_A, ρ_A)-bounded arrival process $A(s,t)$ with (σ_S, ρ_S)-bounded dynamic server $S(s,t)$, as in Figure 7.2. If the stability condition $\rho_A(\theta) < \rho_S(-\theta)$ holds, then the output D is (σ_D, ρ_D)-bounded with*

$$\begin{aligned} \sigma_D(\theta) &= \sigma_A(\theta) + \sigma_S(-\theta) - \frac{1}{\theta} \log\left(1 - e^{\theta(\rho_A(\theta)-\rho_S(-\theta))}\right) \\ \rho_D(\theta) &= \rho_A(\theta). \end{aligned}$$

This property also holds for the new output bound, as the following proposition shows.

Proposition 7.5 (The Output Bound with the Power-Mitigator is (σ, ρ)-Bounded)**.** *Under the assumptions in Proposition 7.4 plus a modified stability condition $\rho_A(p\theta) < \rho_S(-p\theta)$, we obtain that the output A' is (σ_D, ρ_D)-bounded with*

$$\begin{aligned} \sigma_D(\theta) &= \sigma_A(p\theta) + \sigma_S(-p\theta) - \frac{1}{p\theta} \log\left(1 - e^{p\theta(\rho_A(p\theta)-\rho_S(-p\theta))}\right) \\ \rho_D(\theta) &= \rho_A(p\theta), \end{aligned}$$

where $p \geq 1$.

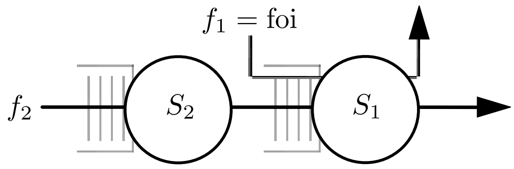


Figure 7.3: Two-server topology.

Proof. See Appendix E.1. □

7.2 NUMERICAL EVALUATION

In this section, we investigate the increased accuracy of our new output bound introduced in Section 7.1. That is, we evaluate the gain of the delay's violation probability by the improved output bound calculation for a two-server topology and a fat tree. The improvement factor is measured by calculating

$$\frac{\text{Bound standard approach}}{\text{Bound power-mitigator}}, \tag{7.10}$$

where clearly larger values are desirable.

For the arrivals, we assume three processes all adhering to the class of (σ_A, ρ_A)-constrained arrivals (see also Section 3.1):

- Independent exponentially distributed increments with parameter λ,
- Independent Weibull distributed arrival increments with fixed shape parameter $k = 2$ and scale parameter λ,
- Continuous-time Markov-modulated On-Off (MMOO) arrivals.

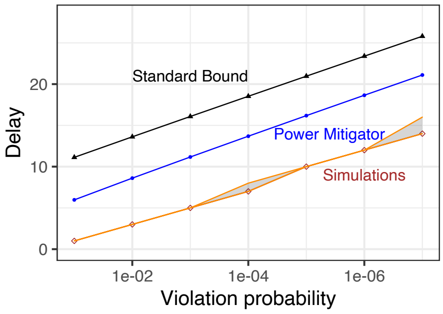
The latter can be described by three parameters: the transition rates, λ and μ, to switch between the "Off"-state and the "On"-state, and a constant peak rate π_A, at which data is sent during the "On"-state. The service is always chosen to be work-conserving and of constant rate.

If not stated otherwise, θ and the Jensen parameters p_i are optimized by a brute force optimization along a grid followed by a downhill simplex algorithm (see Chapter 8).

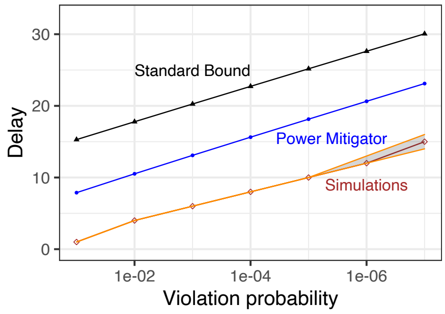
With each application of this new inequality, an additional parameter has to be optimized. On the other hand, since the costs of incorporating the power-mitigator in a given implementation are rather moderate, it gives us convenient new options: Either we prioritize accuracy and optimize all p_i (at the cost of higher computational effort), or focus more on speed setting many $p_i = 1$ (setting all p_i equal to 1 would yield the conventional approach). Hence, we gain more flexibility while being minimally invasive at the same time.

7.2.1 *Two-server topology*

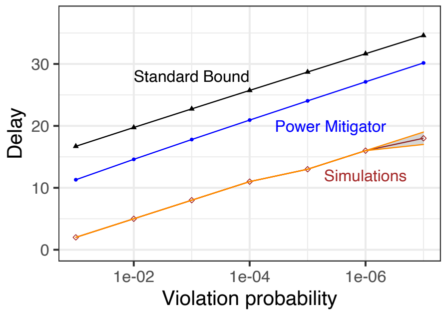
In this section, we investigate the effect of Jensen's inequality on the delay bound. Therefore, we consider the two-server setting in Figure 7.3. Here, a cross flow f_2 enters server S_2 and its output $(\leq (A_2 \oslash S_2))$ is prioritized over the flow of interest f_1 at server S_1. The improved output bound impacts the delay



(a) Exponential distribution with (λ_1, λ_2) = (0.2, 8.0), server rates (r_1, r_2) = (8.0, 0.2)



(b) Weibull distribution with (λ_1, λ_2) = (8.9, 0.1), server rates (r_1, r_2) = (9.0, 0.1)



(c) MMOO with (μ_1, μ_2) = (1.2, 3.7), (λ_1, λ_2) = (2.1, 1.5), (π_{A1}, π_{A2}) = (3.5, 0.4), server rates (r_1, r_2) = (2.0, 0.3)

Figure 7.4: Stochastic delay bounds and simulation results in the two-server setting with constant rate servers.

by being more accurate in terms of the foi's leftover service. Mathematically speaking, this leftover service at S_1 is described by $S_{\text{e2e}} = S_{\text{e2e}}^{\text{PMOO}} = S_{\text{e2e}}^{\text{seqPMOO}} = S_{\text{e2e}}^{\text{seqSFA}}$, where

$$S_{\text{e2e}} = [S_1 - (A_2 \oslash S_2)]^+ .$$

In this topology, we calculate the stochastic delay bound (solving for T in Theorem 3.9) but take the new output bound invocation into account. For the exponential distribution, Weibull distribution, and Markov-modulated On-Off (MMOO) traffic, one example for each distribution is depicted in Figure 7.4. These are compared to simulation results. Similar to Chapter 5, for the simulations, we use shortest-to-destination first (SDF) scheduling which translates to a simple static priority (SP) for the cross-flow in this scenario. As we can observe from these examples, the actual gain from our new power function output bound calculation can vary strongly depending on the scenarios' parameters. For that reason, we decided to systematically sample the parameter spaces in a Monte Carlo-type fashion. That is, we took samples with a size of 10^5 from a uniform distribution as well as an exponential distribution (since the parameter space is only lower bounded) and computed the average and largest improvement as well as the share of improved bounds. The parameters of the arrival and service distribution are drawn from the same distribution, i.e., the stability condition $\rho_A(\theta) < \rho_S(-\theta)$ is approximately half of the time violated. These cases are removed from the results given in Table 7.1. Furthermore, since we aim to focus

Table 7.1: Improvement of the delay's violation probability for the two-server setting and delay = 10 (above: uniform sampling, below: exponential sampling).

Distribution	Exponential	Weibull	MMOO
Average gain	1.40	1.35	1.04
Maximum gain	135.0	85.8	36.6
Share of improved bounds	99.8%	99.8%	99.8%

Distribution	Exponential	Weibull	MMOO
Average gain	1.47	1.64	1.30
Maximum gain	93.2	478.9	68.5
Share of improved bounds	100%	99.9%	99.3%

our analysis on queueing-relevant load situations, we also removed all cases with a utilization < 0.5.

We often observe an improved delay bound, as one can see in the examples of Figure 7.4. It shows that for the delay, the difference is up to 50%. Depending on the parameters, the gap between the simulation results and the analytically derived bounds can be reduced considerably. Considering again bounds on the delay's violation probability, average behavior on the other hand is less significant. Table 7.1 indicates a highly non-linear behavior where some violation probabilities are improved by a factor of 478.9, whereas average gain is moderate with a total mean of 1.34.

7.2.2 *Fat tree*

Starting off with the two-server topology in Figure 7.3, we investigate the delay bound's scaling behavior for multiple invocations of Jensen's inequality. We now take a look at n flows, where $n-1$ are cross flows with corresponding server and their outputs jointly enter server S_1 (see Figure 7.5). The flow of interest is again, due to arbitrary multiplexing, assumed to be served after the cross traffic. In terms of leftover service provided for the foi, this means

$$S_{\text{e2e}} = \left[S_1 - \sum_{i=2}^{n} (A_i \oslash S_i) \right]^+ .$$

We calculated the delay's violation probability for the following setting: The foi has exponentially distributed increments with parameter $\lambda_1 = 0.5$ and enters server S_1 with rate $r_1 = 4$. The $n-1$ cross flows are also exponential, but with parameters $\lambda_i = 8$, $i = 2, \ldots, n$ and corresponding servers S_i with rates $r_i = 2$, $i = 2, \ldots, n$. The accuracy gains for different numbers of servers is depicted in Figure 7.6.

We observe that the ratio increases quickly to 32.8 in the case of 8 servers, even though only an improvement of 1.5 was achieved for the two-server setting. This shows that the power-mitigator can fully develop its strengths in larger networks, when more output bound calculations have to be invoked.

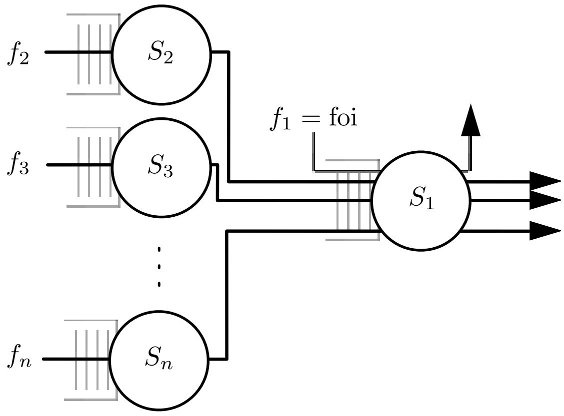


Figure 7.5: Fat tree topology.

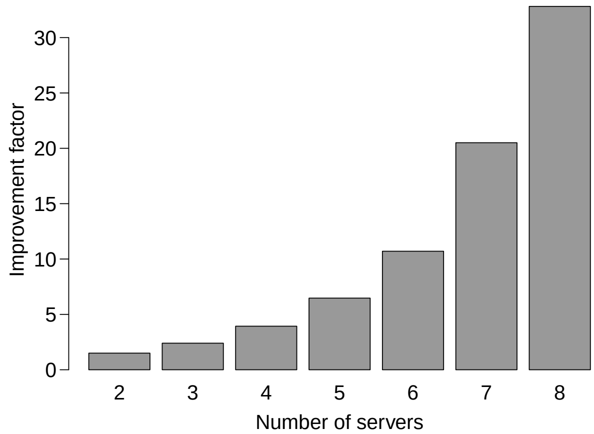


Figure 7.6: Delay bound improvement for different numbers of servers (delta time = 8).

7.2.3 *Runtime*

So far, we focused on the power-mitigator's accuracy gain and observed favorable outcomes. Yet, the other side of the coin is the computational effort the new output bound calculation must invest to optimize over the higher-dimensional parameter space. To investigate this in more detail, we ran 10^5 experiments for exponential as well as MMOO-traffic in the two-server topology (Figure 7.3) and the fat tree (Figure 7.5) with $2, 4, \ldots, 12$ flows. In this scenario, the aforementioned naive grid optimization runs quickly into computational problems, as a computation for 4 flows already took approximately a day. Therefore, we implemented the "Pattern Search" [HJ61] heuristic. Here, a function is minimized by changing arguments only in a single direction. If multiple modifications lead to a descent, a step in the direction of all successful intermediate steps is attempted. The results of the ratio

$$\frac{\text{Computation time power-mitigator}}{\text{Computation time standard approach}}$$

for these experiments are depicted in Figure 7.7.

For Pattern Search, we observe that the computational overhead scales only linearly with the number of invocations of Jensen's inequality. This indicates

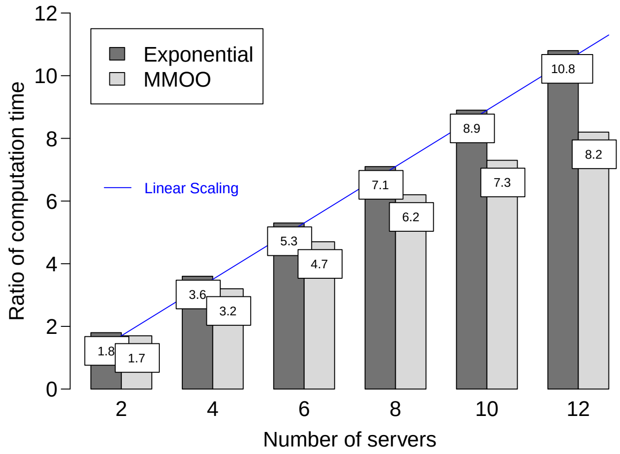


Figure 7.7: Computation time comparison for the state-of-the-art and power-mitigator approach.

that a good trade-off between cost and accuracy gain can be achieved, if optimization is done carefully.

7.3 DIRECT APPLICATION TO DELAY BOUNDS

At first glance, it is tempting to apply the power-mitigator to the delay bound calculation as well, given its results in Section 7.2. That is, we would modify the computation of the delay's violation probability as follows:

$$\mathrm{P}(d(t) > T) \overset{(3.29)}{\leq} \mathrm{E}\left[e^{\theta \sup_{0\leq\tau\leq t-1}\{A(\tau,t)-S(\tau,t+T)\}}\right]$$

$$= \inf_{p\geq 1}\left\{\left(\mathrm{E}\left[e^{p\theta \sup_{0\leq\tau\leq t-1}\{A(\tau,t)-S(\tau,t+T)\}}\right]\right)^{\frac{1}{p}}\right\} \tag{7.11}$$

$$\overset{(7.1)}{\leq} \inf_{p\geq 1}\left\{\left(\sum_{\tau=0}^{t-1}\mathrm{E}\left[e^{p\theta(A(\tau,t)-S(\tau,t+T))}\right]\right)^{\frac{1}{p}}\right\}, \tag{7.12}$$

where we used the identity-property for $p = 1$ in Eqn. (7.11), and the quasi-Union bound in the last line. Owing to the fact that this estimates a probability, only values below 1 are of interest for Eqn. (7.12). Disappointingly for this case, no improvement can be obtained, as the next theorem states.

Theorem 7.6. *Let a delay bound T according to* (7.12) *exist such that*

$$\sum_{\tau=0}^{t-1}\mathrm{E}\left[e^{p\theta(A(\tau,t)-S(\tau,t+T))}\right] < 1. \tag{7.13}$$

If p and θ are optimized (denoted by p^* and θ^*), then $p^* = 1$, i.e., no improvement can be achieved.

Proof. Assume that p^* and θ^* are the optimal parameters for Eqn. (7.12) and that $p^* > 1$. This means that there exist $1 \leq p' < p^*$ and $\theta' > \theta^*$ such that $p'\theta' = p^*\theta^*$. But this means

$$\left(\sum_{\tau=0}^{t-1}\mathrm{E}\left[e^{p^*\theta^*(A(\tau,t)-S(\tau,t+T))}\right]\right)^{\frac{1}{p^*}} = \left(\sum_{\tau=0}^{t-1}\mathrm{E}\left[e^{p'\theta'(A(\tau,t)-S(\tau,t+T))}\right]\right)^{\frac{1}{p^*}}$$

$$> \left(\sum_{\tau=0}^{t-1} \mathrm{E}\left[e^{p'\theta'(A(\tau,t)-S(\tau,t+T))} \right] \right)^{\frac{1}{p'}},$$

where we inserted $p^*\theta^* = p'\theta'$ in the first line. In the second line, we used that $x^{\frac{1}{p^*}} > x^{\frac{1}{p'}}$ holds for all $x \in (0,1)$ and $p^* > p' \geq 1$. Clearly, this is a contradiction to our assumption that we had an optimal solution. Thus, the optimal p^* must be equal to 1. □

As a consequence, the power-mitigator approach can only indirectly decrease delay bounds via the output bound calculation. The same holds for the backlog bound (the proof follows along the same lines).

7.4 SUMMARY

In this chapter, we proposed a novel approach to improve the MGF output bound calculation in the stochastic network calculus using Jensen's inequality with *h*-mitigators. We also gave a proof showing why this is a valid bound and, when using the power function (Eqn. (7.6)), that it is always at least as accurate as the state-of-the-art method. It is also shown in comprehensive numerical evaluations that the delay's violation probability can be improved for two-server topologies as well as fat trees. Our evaluation indicated a significant gain in some cases while leading to more moderate improvements on average. For a fat tree, we observed a very high gain as the number of cross flows is increased. These gains come conceptually for free, as no additional constraints have to be imposed, thus making our approach minimally invasive. Yet, from a computational perspective, the gain comes at the price of a higher-dimensional optimization in the last stage of computing the bounds. Fortunately, our experiments indicate that the computational overhead only scales linearly with the invocations of Jensen's inequality under a carefully chosen optimization method.

8 A TOOLBOX FOR STOCHASTIC NETWORK CALCULUS WITH MOMENT-GENERATING FUNCTIONS

In this short chapter, we present the SNC-MGF toolbox[1]. It is a Python implementation that can be used to derive stochastic end-to-end performance bounds for (σ, ρ)-constrained arrivals (Definition 3.2) and servers (Definition 3.6). We discuss its components below. Those include:

- Arrivals and service (Section 8.1)
- Performance bounds (Section 8.2)
- Network operations and a modular analysis (Section 8.3)
- End-to-end delay bounds (Section 8.4)

8.1 ARRIVALS AND SERVICE

All arrival classes inherit from the *Arrival()* class (Listing 1). It contains the necessary information to characterize the distribution: The $\sigma(\theta)$ and $\rho(\theta)$ part, as well as whether the process is discrete- or continuous-time. In a similar fashion, service processes inherit from the *Server()* class in Listing 2.

8.2 PERFORMANCE BOUNDS

Performance bounds, on the other hand, are implemented in performance_bounds.py. For example, the code for the delay's violation probability, *delay_prob()*, is given in Listing 3. Note that we directly implement the case for dependent arrivals and service with $p = q = 1$ being the special case under independence. The function *stability_check()* guarantees the stability condition in Eqn. (3.23). Depending on whether we have discrete- or continuous-time arrivals, we invoke different bounds.

8.3 NETWORK OPERATIONS AND MODULAR ANALYSIS

For an end-to-end analysis, we also implemented all network operations from Section 3.3. They enable to conduct a modular analysis. Here, we exemplify the deconvolution operation for the output bound (Theorem 3.12) in Listing 4. Note that each network operation class is implemented for the independent as well es the dependent case.

Let us explain the procedure for the sink tree tandem in Figure 4.3b under seqPMOO. In Eqn. (4.4), it is stated that

$$S_{\text{e2e}}^{\text{seqPMOO}} = \left[\left(\left[\left([S_n - A_{n+1}]^+ \otimes S_{n-1} \right) - A_n \right]^+ \otimes \cdots \otimes S_1 \right) - A_2 \right]^+ .$$

1 https://github.com/paulnikolaus/snc-mgf-toolbox

```python
"""Abstract Arrival class."""

from abc import abstractmethod, ABC


class Arrival(ABC):
        """Abstract Arrival class."""

        @abstractmethod
        def sigma(self, theta:  float) ->  float:
                """
                sigma(theta)
                :param theta: mgf parameter
                """
                pass

        @abstractmethod
        def rho(self, theta:  float) ->  float:
                """
                rho(theta)
                :param theta: mgf parameter
                """
                pass

        @abstractmethod
        def is_discrete(self) ->  bool:
                """
                :return True if the arrival distribution is discrete, False if not
                """
                pass
```

Listing 1: Arrival() class

The Python class to implement this end-to-end service is presented in Listing 5. It inherits from the class *Setting()* to ensure that the method *standard_bound()* is implemented for the subsequent parameter optimization. the class *LeftoverARB()* yields the leftover service using Theorem 3.11, whereas *Convolve()* implements Theorem 3.13 for the two server case.

8.4 END-TO-END DELAY BOUNDS

For the One flow - n servers topology in Figure 3.6, we have seen that Theorem 3.14 can be used to derive end-to-end delay bounds. Note that we can easily incorporate more complex topologies with the help of the leftover service. Yet, we present the code of Theorem 5.3, as it often leads to significantly improved performance bounds. It consists of the *Flow()* class (Listing 6) that contains the topological information of each flow plus the e2e delay bounds given in Listing 7. Since the end-to-end performance bounds in Theorem 5.3 distinguishes between servers in π_1 and servers $\notin \pi_1$, we take this into account in the function arguments. *E2EEnum()* is an enum(eration) class that enables us to choose which of the three bounds to compute. It is important to note that this implementation only allows for independent processes and only for trees, yet it can be extended.

```python
"""Implemented service class"""

from abc import abstractmethod, ABC


class Server(ABC):
        """Abstract Server class"""

        @abstractmethod
        def sigma(self, theta:  float) ->  float:
                """Sigma method"""
                pass

        @abstractmethod
        def rho(self, theta:  float) ->  float:
                """Rho method"""
                pass
```

Listing 2: Server() class

In a final step, we optimize all parameters. As a default choice, we use a brute force optimization along a grid using the "brute()" method in the "scipy.optimize" library [Oli07, Vir+20]. When applying a second optimization step (a downhill simplex), this method also evaluates points outside the grid. However, we also implemented other heuristics to conduct the parameter optimization such as "pattern search" [HJ61], the "Nelder–Mead (downhill simplex) method" [NM65], "simulated annealing" [KGV83], and interfaces to SciPy functions such as "Basin-Hopping" [WD97], "differential evolution" [SP97], or "dual annealing" [TS96].

```
def delay_prob(arr: Arrival, ser: Server, theta:  float, delay_value:
    int, indep=True, p=1.0, geom_series=False) ->  float:
        """Implements stationary standard_bound method"""
        if indep:
                p = 1.0
                q = 1.0
        else:
                q = get_q(p=p)

        stability_check(arr=arr, ser=ser, theta=theta, indep=indep, p=p, q=q)

        if indep:
                sigma_sum, rho_diff arr.sigma(theta=theta) + ser.sigma(theta=theta
                    ), arr.rho(theta=theta) - ser.rho(theta=theta)

        else:
                sigma_sum, rho_diff = arr.sigma(theta=p * theta) + ser.sigma(theta
                    =q * theta), arr.rho(theta=p * theta) - ser.rho(theta=q *
                    theta)

        if arr.is_discrete():
                return exp(
                -theta * ser.rho(theta=q * theta) * delay_value) * exp(theta *
                    sigma_sum) / (-rho_diff * theta)
        else:
                tau_opt = 1 / (theta * ser.rho(theta=q * theta))
                return exp(-theta * ser.rho(theta=q * theta) * delay_value) * exp(
                    theta * (ser.rho(theta=q * theta) * tau_opt + sigma_sum)) / (-
                    rho_diff * theta * tau_opt)
```

Listing 3: delay_prob()-function

```python
class Deconvolve(Arrival):
        """Deconvolution class."""
        def __init__(self, arr: Arrival, ser: Server, indep=True, p=1.0) -> None:
                self.arr = arr
                self.ser = ser
                self.indep = indep

        if indep:
                self.p = 1.0
                self.q = 1.0
        else:
                self.p = p
                self.q = get_q(p=p)

        def sigma(self, theta:  float) ->  float:
                """

                :param theta: mgf parameter
                :return:      sigma(theta)
                """

                arr_sigma_p = self.arr.sigma(self.p * theta)
                ser_sigma_q = self.ser.sigma(self.q * theta)

                arr_rho_p = self.arr.rho(self.p * theta)
                k_sig = -log(1 - exp(theta * (arr_rho_p - self.ser.rho(self.q *
                    theta)))) / theta

                if self.arr.is_discrete():
                        return arr_sigma_p + ser_sigma_q + k_sig
                else:
                        return arr_sigma_p + ser_sigma_q + arr_rho_p + k_sig

        def rho(self, theta:  float) ->  float:
                """

                :param theta: mgf parameter
                :return: rho(theta)
                """

                arr_rho_p = self.arr.rho(self.p * theta)

                if arr_rho_p < 0 or self.ser.rho(self.q * theta) < 0:
                        raise ParameterOutOfBounds("The rhos must be >= 0")

                stability_check(arr=self.arr, ser=self.ser, theta=theta, indep=
                    self.indep, p=self.p, q=self.q)

                return arr_rho_p

        def is_discrete(self):
                return self.arr.is_discrete()
```

Listing 4: Deconvolve() class

```
class SinkTreePMOO(Setting):
        """Canonical sink tree with PMOO analysis"""
        def __init__(self, arr_list: List[ArrivalDistribution], ser_list: List[
            ConstantRateServer], perform_param: PerformParameter) -> None:
                # The first element in the arrival list in dedicated to the foi
                if  len(arr_list) != ( len(ser_list) + 1):
                        raise IllegalArgumentError(f"number of arrivals {len(
                            arr_list)} and servers {len(ser_list)} + 1 have to
                            match")

                self.arr_list = arr_list
                self.ser_list = ser_list
                self.perform_param = perform_param

                self.number_servers =  len(ser_list)

        def standard_bound(self, param_list: List[ float]) ->  float:
                theta = param_list[0]

                s_net: Server = LeftoverARB(ser=self.ser_list[self.number_servers
                    - 1], cross_arr=self.arr_list[self.number_servers])

                for _i in  range(self.number_servers - 2, -1, -1):
                        s_net = Convolve(ser1=s_net, ser2=self.ser_list[_i])
                        s_net = LeftoverARB(ser=s_net, cross_arr=self.arr_list[_i
                            + 1])

                return single_hop_bound(foi=self.arr_list[0], s_e2e=s_net, theta=
                    theta, perform_param=self.perform_param)
```

Listing 5: SinkTreePMOO() class

```
class Flow( object):
def __init__(self, arr: ArrivalDistribution, server_indices: List[ int]):
self.arr = arr
self.server_indices = server_indices
```

Listing 6: Flow() class

```
def pmoo_tandem_bound(foi: Flow, cross_flows_on_foi_path: List[Flow],
    ser_on_foi_path: List[Server], theta:
    float, perform_param: PerformParameter, e2e_enum: E2EEnum,
    cross_flows_not_on_foi_path=None, ser_not_on_foi_path=None, indep=True,
    geom_series=True) ->
    float:
        if indep is False:
                raise NotImplementedError("Only implemented for independent
                    processes")

        residual_rate_with_foi_list = [0.0] *  len(ser_on_foi_path)
        residual_rate_list = [0.0] *  len(ser_on_foi_path)
        sigma_sum = 0.0
        foi_rate = foi.arr.rho(theta=theta)

        for k, server in  enumerate(ser_on_foi_path):
                residual_rate_list[k] = server.rho(theta=theta)
                residual_rate_with_foi_list[k] = residual_rate_list[k] - foi_rate
                sigma_sum += server.sigma(theta=theta)

        sigma_sum += foi.arr.sigma(theta=theta)

        for cross_flow in cross_flows_on_foi_path:
                sigma_sum += cross_flow.arr.sigma(theta=theta)

        for server_index in cross_flow.server_indices:
                cross_arr_rate = cross_flow.arr.rho(theta=theta)
                residual_rate_list[server_index] -= cross_arr_rate
                residual_rate_with_foi_list[server_index] -= cross_arr_rate

        for res_rate_with_foi in residual_rate_with_foi_list:
                if res_rate_with_foi <= 0:
                        raise ParameterOutOfBounds("Stability condition is
                            violated")

        if ser_not_on_foi_path is not None:
                residual_rate_not_on_foi_list = [0.0] *  len(ser_not_on_foi_path)

                for j, server in  enumerate(ser_not_on_foi_path):
                        residual_rate_not_on_foi_list[j] = server.rho(theta=theta)
                        sigma_sum += server.sigma(theta=theta)

                for cross_flow in cross_flows_not_on_foi_path:
                        sigma_sum += cross_flow.arr.sigma(theta=theta)

                for server_index in cross_flow.server_indices:
                        cross_arr_rate = cross_flow.arr.rho(theta=theta)
                        residual_rate_not_on_foi_list[server_index] -=
                            cross_arr_rate

                for res_rate_not_foi in residual_rate_not_on_foi_list:
                        if res_rate_not_foi <= 0:
                                raise ParameterOutOfBounds("Stability condition is
                                    violated")
        ...
```

Listing 7: pmoo_tandem_bound() function

Part IV

FAIR QUEUEING IN SNC

9

STOCHASTIC ANALYSIS OF GENERALIZED PROCESSOR SHARING

Generalized process sharing (GPS) [PG93, PG94] is an idealized (theoretical) scheduler that aims at *fair* resource allocation. In other words, it provides the "theoretical underpinnings for fair packet scheduling algorithms" [BL18]. It is a "natural generalization" [PG93] of uniform processor sharing [Kle75] and the packet-based version weighted fair queuing (WFQ) [DKS89]. There is a long list of fair packet schedulers, including deficit round robin (DRR) [SV95], weighted round robin (WRR) [KSC91], worst-case fair weighted fair queueing [BZ96], start-time fair queueing (SFQ) [GVC97], and many others. In recent years, GPS has seen some treatment in the DNC analysis [BL18], [BBC18, pp. 171], [Bou21].

Yet, to be the best of our knowledge, there has only been limited investigation in the stochastic counterpart, with the notable exceptions of [ZTK95, Fid05, LBL07, CPS13]. In [CPS13], it was shown in the single server case, that bounds obtained via martingale-based techniques not only significantly outperform SNC results, they also provide tight performance bounds by comparing them to simulations. However, this comes with the notable exception of GPS; here, even the martingale bounds indicate a more conservative decay rate compared to simulations (see Figure 9.1). Yet, only a homogeneous case where all arrivals have the same parameters are considered.

In this chapter, we compare different GPS analyses in the stochastic network calculus and give an overview of what can be considered as state-of-the-art analysis. We consider the SNC with MGFs as well as with tail bounds / envelope functions, since recent GPS results in the DNC [BL18], [BBC18, pp. 171] can be incorporated into the latter. We show in a numerical evaluation that either of the two branches can outperform the so-called "GPS Basic". Additionally, we show in this evaluation that, in case we cannot make independence assumptions on the arrivals, GPS Basic (Theorem 9.2 below) still provides the best trade-off solution between accuracy and runtimes.

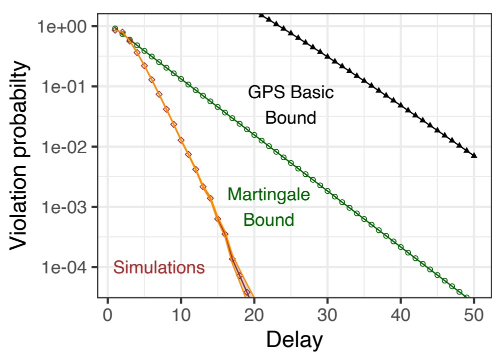


Figure 9.1: Simulation results, Martingale bounds, and SNC bounds on the delay's violation probability (for continuous-time MMOO arrivals).

OUTLINE The rest of the chapter is structured as follows. In Section 9.1, we introduce the necessary background on GPS. Section 9.2 then includes the different SNC techniques to provide stochastic performance bounds for GPS. A numerical evaluation is given in Section 9.3. Section 9.4 summarizes the chapter.

9.1 GPS BACKGROUND

In this section, we give a formal definition of a server implementing generalized processor sharing (GPS) according to [PG93]. Moreover, we introduce the necessary notation and give an overview of the state-of-the-art results known from the literature.

Definition 9.1 (GPS Server [PG93])**.** Let S be a work-conserving server (Definition 3.5). Assume a set of flows $f_j, \in \mathcal{N} = \{1, \ldots, m\}$ traversing this server and let the weights $\phi_j > 0, j = 1, \ldots, m$. Then, a *GPS server* is defined if it holds that

$$\frac{D_i(s,t)}{\phi_i} \geq \frac{D_j(s,t)}{\phi_j}, \quad \forall j \in \mathcal{N} \tag{9.1}$$

for any flow i that is continuously backlogged in the interval (s,t).

In the following, we always assume the foi to have index i.

Theorem 9.2 (GPS Basic)**.** *Let S be a work-conserving GPS server for all flows $f_j, j \in \mathcal{N}$. Then, for all $t \geq 0$, f_i sees a dynamic server with service*

$$S^i_{\text{l.o.}}(s,t) := S^i(s,t) := \frac{\phi_i}{\sum_{k \in \mathcal{N}} \phi_k} \cdot S(s,t). \tag{9.2}$$

Proof. For an initial proof for dynamic servers with constant rate, see [PG93]. For a generalization to work-conserving servers (actually, strict service curves since the univariate case is considered), see [BBC18, pp. 171]. □

In the following, we refer to this result as "GPS Basic". Note that in the case of a constant rate server with rate $C \geq 0$, $S^i(s,t) = \frac{\phi_i}{\sum_{k \in \mathcal{N}} \phi_k} C$ is also called *backlog clearing rate* [PG94].

By abuse of notation, we denote by $i \in M \subset \mathcal{N}$ an arbitrary subset M of $\mathcal{N}$ that contains the foi's index i.

Theorem 9.3 (GPS M)**.** *Let S be a work-conserving GPS server for the aggregate of flows $f_j, j \in \mathcal{N}$. We denote the respective departure processes by $D_j, j \in \mathcal{N}$. Then, for all $t \geq 0$, f_i sees a dynamic server with service*

$$S^i_{\text{l.o.}}(s,t) := S^i(s,t) := \max_{i \in M \subset \mathcal{N}} \left\{ \frac{\phi_i}{\sum_{j \in M} \phi_j} \left(S(s,t) - \sum_{j \notin M} D_j(s,t) \right) \right\}. \tag{9.3}$$

Proof. A proof for the special case of token bucket arrival and constant rate service is given in [PG93], [Chaoo, pp. 68]. Yet, since the proof provides us with useful inequalities, we show it here.

Let $0 \le s \le t$ and $i \in M \subset \mathcal{N}$. Let s_i be the beginning of the backlogged period of flow f_i up to time t. Since we have a work-conserving server S for the aggregate and since $(s_i, t]$ is a backlogged period, it holds that

$$\begin{aligned} \sum_{j \in \mathcal{N}} D_j(s_i, t) &= S(s_i, t) \\ \Leftrightarrow \sum_{j \in M} D_j(s_i, t) &= S(s_i, t) - \sum_{j \notin M} D_j(s_i, t). \end{aligned} \tag{9.4}$$

Thus, it follows that:

$$\begin{aligned} \left(\sum_{j \in M} \phi_j \right) D_i(s_i, t) &\overset{(9.1)}{\ge} \phi_i \sum_{j \in M} D_j(s_i, t) \\ &\overset{(9.4)}{=} \phi_i \left(S(s_i, t) - \sum_{j \notin M} D_j(s_i, t) \right) \end{aligned}$$

which is equivalent to

$$D_i(t) \ge D_i(s_i) + \frac{\phi_i}{\sum_{j \in M} \phi_j} \left(S(s_i, t) - \sum_{j \notin M} D_j(s_i, t) \right) \tag{9.5}$$

for all $i \in M \subset \mathcal{N}$. Since s_i is the start of the backlogged period of flow f_i, we have

$$D_i(s_i) = A_i(s_i).$$

Hence, it follows that

$$\begin{aligned} D_i(t) &\ge A_i(s_i) + \max_{i \in M \subset \mathcal{N}} \left\{ \frac{\phi_i}{\sum_{j \in M} \phi_j} \left(S(s_i, t) - \sum_{j \notin M} D_j(s_i, t) \right) \right\} \\ &\ge \inf_{0 \le s \le t} \left\{ A_i(s) + \max_{i \in M \subset \mathcal{N}} \left\{ \frac{\phi_i}{\sum_{j \in M} \phi_j} \left(S(s, t) - \sum_{j \notin M} D_j(s, t) \right) \right\} \right\}. \end{aligned}$$

This proves the theorem. □

In the following, we refer to this result as "GPS M". The formula in Equation (9.3) is not a "closed-form" solution, as it contains the sample path of the output. In the above proof, it is tempting to bound $D_j(s_i, t)$ in Eqn. (9.5) by $A_j(s_i, t)$ with the help of causality [OMP06]. However, one can construct a counterexample such that flow f_j is already backlogged at s_i (in other words, s_i is not the start of the backlogged period of flow f_j).

One approach is to compute an output bound with the help of GPS Basic in Eqn. (9.2) and the output bound in Eqn. (3.21) [Chaoo, pp. 68], [Fid10]. Let $i \in M \subset \mathcal{N}$. Then,

$$\begin{aligned} D_j(s, t) &\le A_j \oslash \left(\frac{\phi_j}{\sum_{k \in \mathcal{N}} \phi_k} S \right) (s, t), \quad \forall j \in M \\ \Rightarrow \quad \sum_{j \notin M} D_j(s, t) &\le \sum_{j \notin M} \left\{ A_j \oslash \left(\frac{\phi_j}{\sum_{k \in \mathcal{N}} \phi_k} S \right) (s, t) \right\}. \end{aligned} \tag{9.6}$$

This leads to

$$S^i(s,t) = \max_{i \in M \subset \mathcal{N}} \left\{ \frac{\phi_i}{\sum_{j \in M} \phi_j} \left(S(s,t) - \sum_{j \notin M} \left\{ A_j \oslash \left(\frac{\phi_j}{\sum_{k \in \mathcal{N}} \phi_k} S \right) (s,t) \right\} \right) \right\}. \tag{9.7}$$

In the deterministic network calculus (DNC), an improved state-of-the-art leftover service curve has been derived recently. In contrast to the other results, we employ strict service curves (Definition 2.4) and arrival curves (Definition 2.3).

Theorem 9.4. *Let S be a work-conserving GPS server that offers a convex strict service curve β to the aggregate of flows $f_j, j \in \mathcal{N}$. Moreover, let the arrivals A_j be constrained by a concave arrival curve α_j, $j \in \mathcal{N} \setminus \{i\}$. Then, for all $t \geq 0$, f_i sees a strict service curve*

$$\beta^i(t) := \max_{i \in M \subset \mathcal{N}} \left\{ \frac{\phi_i}{\sum_{j \in M} \phi_j} \left(\beta(t) - \sum_{j \notin M} \alpha_j(t) \right) \right\}. \tag{9.8}$$

This leftover service curve is tight in the sense that the bound holds with equality in the "greedy/lazy scenario".

Proof. A proof is given in [BL18], [BBC18, pp. 172]. □

Remark 9.5. For the special case of a leaky bucket arrival curves $\gamma_{r_j,b_j}, j \in \mathcal{N}$, (Eqn. (2.14)) and a rate-latency service curve $\beta_{R,T}$ (Eqn. (2.18)), one receives

$$\beta^i(t) = \max_{i \in M \subset \mathcal{N}} \left\{ \beta_{\frac{\phi_i}{\sum_{j \in M} \phi_j} \left(R - \sum_{j \notin M} r_j \right), \left(\frac{\sum_{j \notin M} b_j + R \cdot T}{R - \sum_{j \notin M} r_j} \right)}(t) \right\}.$$

We show in Appendix F.1 that in a homogeneous scenario ($\gamma_{r_j,b_j} = \gamma_{r,b}$ and $\phi_j = \phi \, \forall j \in \mathcal{N}$), under stability, $M = \mathcal{N}$ yields the best delay bound for the foi.

However, in the SNC with MGFs, we work directly on the sample path A and hence, we cannot make any concavity assumptions.

9.2 STOCHASTIC ANALYSIS OF GPS

In this section, we show how to obtain stochastic performance bounds under GPS in the SNC. At first, we consider SNC with MGFs and give (σ_S, ρ_S)-bounds for the GPS leftover service in case S was (σ_S, ρ_S)-constrained. This enables us to calculate stochastic performance bounds by directly applying Theorem 3.9. Then, we show how to obtain performance bounds for the SNC with tail bounds / envelope functions, where we can make use of Theorem 9.4. In order to facilitate a concise notation, we omit the $\max_{i \in M \subset \mathcal{N}}$ in the following and only write that $i \in M \subset \mathcal{N}$ is arbitrary.

9.2.1 (σ_S, ρ_S)-*Bounds of the GPS Leftover Service*

We start off with the leftover service under GPS Basic.

Proposition 9.6. *Let $\theta > 0$. Assume a work-conserving GPS server S and let S be (σ_S, ρ_S)-constrained. The leftover service S^i for flow f_i under GPS Basic is $(\sigma_{S^i}, \rho_{S^i})$-constrained with*

$$\sigma_{S^i}(-\theta) = \psi_i \sigma_S(-\psi_i\theta) = \frac{\phi_i}{\sum_{k\in\mathcal{N}}\phi_k}\sigma_S\left(-\frac{\phi_i}{\sum_{k\in\mathcal{N}}\phi_k}\theta\right),$$
$$\rho_{S^i}(-\theta) = \psi_i \rho_S(-\psi_i\theta) = \frac{\phi_i}{\sum_{k\in\mathcal{N}}\phi_k}\rho_S\left(-\frac{\phi_i}{\sum_{k\in\mathcal{N}}\phi_k}\theta\right),$$

where we define

$$\psi_i := \frac{\phi_i}{\sum_{k\in\mathcal{N}}\phi_k}.$$

Proof. See Appendix F.2. □

In the next proposition, we derive a (σ_S, ρ_S)-bound for GPS M.

Proposition 9.7. *Let $\theta > 0$ and $i \in M \subset \mathcal{N}$. Assume a work-conserving GPS server S and let S be deterministic and (σ_S, ρ_S)-constrained. We define for any $i \in M \subset \mathcal{N}$*

$$\psi_{i,M} := \frac{\phi_i}{\sum_{j\in M}\phi_j}$$

and

$$\psi_j := \frac{\phi_j}{\sum_{k\in\mathcal{N}}\phi_k}.$$

Further, we assume for all $j \in \mathcal{N}\setminus\{i\}$ that A_j is $\left(\sigma_{A_j}, \rho_{A_j}\right)$-constrained such that the stability condition

$$\rho_{A_j}(p_j\psi_{i,M}\theta) < \psi_j\rho_S(-\psi_{i,M}\psi_j\theta)$$

holds for all $j \in \mathcal{N}$. Then, the leftover service S^i for flow f_i under GPS M is $(\sigma_{S^i}, \rho_{S^i})$-constrained with

$$\begin{aligned}\sigma_{S^i}(-\theta) =& \psi_{i,M}\sigma_S(-\psi_{i,M}\theta) + \psi_{i,M}\sum_{j\notin M}\Big\{\sigma_{A_j}(p_j\psi_{i,M}\theta) + \psi_j\sigma_S(-\psi_{i,M}\psi_j\theta)\\ &-\frac{1}{\theta\psi_{i,M}}\log\left(1 - e^{\psi_{i,M}\theta\left(\rho_{A_j}(p_j\psi_{i,M}\theta) - \psi_j\rho_S(-\psi_{i,M}\psi_j\theta)\right)}\right)\Big\},\\ \rho_{S^i}(-\theta) =& \psi_{i,M}\left(\rho_S(-\psi_{i,M}\theta) - \sum_{j\notin M}\rho_{A_j}(p_j\psi_{i,M}\theta)\right),\end{aligned}$$

where

$$\sum_{j\notin M}\frac{1}{p_j} = 1.$$

If the flows can be assumed to be independent, then $p_j = 1, j \notin M$.

Proof. See Appendix F.2. □

Note that, in case the optimization of Hölder parameters is computationally infeasible (e.g., due to runtime constraints) we can always revert to using GPS Basic, where we do not even need the $A_j, j \in \mathcal{N}\setminus\{i\}$ to have any arrival constraints.

9.2.2 *Tail Bound Analysis of GPS*

For the SNC branch with tail bounds / envelope functions, the analysis looks a bit different. Here, we need to prove a stochastic service curve (Definition F.2 in the appendix) in order to calculate performance bounds (Theorem F.3 in the appendix). At first, we introduce a proposition to calculate bounds on the delay's violation probability in the general case oblivious to any dependence of the arrivals. Then, we also provide delay bounds for independent arrivals.

The definitions and notations are mainly inspired by [CBL06]. For some *slack rate* $\delta > 0$, we introduce the notation $f_\delta(t) = f(t) + \delta \cdot t$. Moreover, we also abuse notation and write

$$A \otimes [\beta - \sigma]^+ (t)$$

for

$$\inf_{0 \le s \le t} \left\{ A(s) + [\beta(t-s) - \sigma]^+ \right\}.$$

For the sake of closed-form results, we limit ourselves to $(\sigma_A(\theta), \rho_A(\theta))$-constrained arrivals. One can easily show that this traffic class has a stochastic arrival curve $\alpha(t) := \rho_A(\theta) \cdot t$ with error function $\varepsilon_a(\sigma) := e^{\theta \sigma_A(\theta)} e^{-\theta\sigma}$. To be precise, it belongs to the class of exponentially bounded burstiness (EBB) (see Eqn. (3.4)).

GENERAL CASE: We start off with the general case. As the only exception, we give a general stochastic leftover service curve without assuming $(\sigma_A(\theta), \rho_A(\theta))$-constrained arrivals.

Proposition 9.8. *Let $\delta > 0$ and $i \in M \subset \mathcal{N}$. Assume a work-conserving GPS server offering a convex strict service curve β to an aggregate of flows $f_j, j \in \mathcal{N}$ with concave stochastic arrival curves $\alpha_j, j \notin M$ and error function $\varepsilon_{a_j}(\sigma)$. Assume the stability condition*

$$\limsup_{t \to \infty} \sum_{j \in \mathcal{N} \setminus \{i\}} \alpha_j(t) - \beta(t) < 0$$

for all $t \ge 0$. Then,

$$\beta^i_{\text{l.o.}}(t) := \beta^i(t) := \frac{\phi_i}{\sum_{j \in M} \phi_j} \left(\beta(t) - \sum_{j \notin M} \alpha_{j,\delta}(t) \right)$$

is a stochastic service curve for flow f_i for flow f_i with error function

$$\varepsilon_s(\sigma) = \inf_{\sum_{j \notin M} \sigma_j = \sigma} \left\{ \sum_{j \notin M} \sum_{k=0}^{\infty} \varepsilon_{a_j}(\sigma_j + k\delta) \right\}.$$

Proof. See Appendix F.4. □

Now, we can use this to derive a bound on the delay's violation probability.

Proposition 9.9. *Let $\theta, \delta > 0$ and $i \in M \subset \mathcal{N}$. Assume a work-conserving GPS server with strict service curve $\beta(t) = C \cdot t$ to an aggregate of flows f_j that are $\left(\sigma_{A_j}, \rho_{A_j}\right)$-constrained, $j \in \mathcal{N}$. Let $\alpha_j(t) := \rho_{A_j}(\theta), j \in \mathcal{N}$ and assume the stability condition*

$$\rho_{A_i,\delta}(\theta) + \frac{\phi_i}{\sum_{j \in M} \phi_j} \sum_{j \notin M} \rho_{A_j,\delta}(\theta) \le \frac{\phi_i}{\sum_{j \in M} \phi_j} C. \quad (9.9)$$

Then, it holds for flow f_1 that

$$\mathrm{P}(d(t) > T) \leq (1 + |\mathcal{N} \setminus M|) \frac{1}{1 - e^{-\theta\delta}} \cdot e^{-\frac{\theta}{1+|\mathcal{N}\setminus M|} T \cdot \frac{\phi_i}{\sum_{j\in M}\phi_j} \cdot \left(C - \sum_{j\notin M} \rho_{A_j}(\theta) - |\mathcal{N}\setminus M| \cdot \delta\right)} \cdot e^{\frac{\theta}{1+|\mathcal{N}\setminus M|}\left(\sigma_{A_i}(\theta) + \sum_{j\notin M} \sigma_{A_j}(\theta)\right)}.$$

Proof. See Appendix F.4. □

INDEPENDENT CASE: Next, we continue by assuming arrivals to be independent.

Proposition 9.10. *Let $\theta, \delta > 0$ and $i \in M \subset \mathcal{N}$. Assume a work-conserving GPS server with service curve $\beta(t) = C \cdot t$ to an aggregate of* independent *flows f_j that are $\left(\sigma_{A_j}, \rho_{A_j}\right)$-constrained, $j \in \mathcal{N}$. Let $\alpha_j(t) := \rho_{A_j}(\theta), j \in \mathcal{N}$ and assume the stability condition*

$$\sum_{j\in\mathcal{N}} \rho_{A_j,\delta}(\theta) \leq C.$$

for all $t \geq 0$. Then,

$$\beta^i_{\mathrm{l.o.}}(t) := \beta^i(t) := \frac{\phi_i}{\sum_{j\in M}\phi_j}\left(C - \sum_{j\notin M} \rho_{A_j,\delta}(\theta)\right) \cdot t$$

is a stochastic service curve for flow f_i with error function

$$\varepsilon_s(\sigma) = \begin{cases} \frac{e^{\theta \sum_{j\notin M} \sigma_j(\theta)}}{1 - e^{-\theta|\mathcal{N}\setminus M|\delta}} e^{-\theta\sigma}, & \text{if } M \neq \mathcal{N} \\ 0, & \text{else.} \end{cases}$$

Proof. See Appendix F.4. □

Similar to the general case, we now derive a bound on the delay's violation probability.

Proposition 9.11. *Let $\theta, \delta > 0$ and $i \in M \subset \mathcal{N}$. Assume a work-conserving GPS server with service curve $\beta(t) = C \cdot t$ to an aggregate of* independent *flows f_j that are $\left(\sigma_{A_j}, \rho_{A_j}\right)$-constrained, $j \in \mathcal{N}$. Let $\alpha_j(t) := \rho_{A_j}(\theta), j \in \mathcal{N}$ and assume the stability condition*

$$\rho_{A_i,\delta}(\theta) + \frac{\phi_i}{\sum_{j\in M}\phi_j} \sum_{j\notin M} \rho_{A_j,\delta}(\theta) \leq \frac{\phi_i}{\sum_{j\in M}\phi_j} C. \tag{9.10}$$

Then, it holds for flow f_i that

$$\mathrm{P}(d(t) > T) \leq 2 \left(\frac{1}{1 - e^{-\theta|\mathcal{N}\setminus M|\cdot\delta}}\right)^{\frac{1}{2}} \cdot \left(\frac{1}{1 - e^{-\theta\delta}}\right)^{\frac{1}{2}} \cdot e^{-\frac{\theta}{2} T \cdot \frac{\phi_i}{\sum_{j\in M}\phi_j} \cdot \left(C - \sum_{j\notin M} \rho_{A_j}(\theta) - |\mathcal{N}\setminus M|\cdot\delta\right)} \cdot e^{\theta\left(\sigma_{A_i}(\theta) + \frac{\phi_i}{\sum_{j\in M}\phi_j} \cdot \sum_{j\notin M} \sigma_{A_j}(\theta)\right)},$$

if $M \neq \mathcal{N}$, and that

$$\mathrm{P}(d(t) > T) \leq \frac{1}{1 - e^{-\theta\delta}} e^{-\theta T \cdot \frac{\phi_i}{\sum_{k \in \mathcal{N}} \phi_k} \cdot C} e^{\theta \sigma_{A_i}(\theta)},$$

if $M = \mathcal{N}$. In the latter case, the optimal δ is

$$\delta^* = \frac{\phi_i}{\sum_{k \in \mathcal{N}} \phi_k} \cdot C - \rho_{A_i}(\theta).$$

Proof. See Appendix F.4. □

Proposition 9.11 is the formula with the closest relation to [ZTK95, Theorem 8], yet, to the best of our knowledge, we obtained smaller delay bounds because of the factor $\frac{\phi_i}{\sum_{j \in M} \phi_j} \cdot \left(C - \sum_{j \notin M} \rho_{A_j}(\theta) - |\mathcal{N} \setminus M| \cdot \delta \right)$ that gives more flexibility compared to the backlog clearing rate.

9.3 NUMERICAL EVALUATION

In this section, we compare stochastic delay bounds for a flow of interest (foi) with cross-flows at a single server under GPS scheduling. In all our experiments, we implement GPS Basic (Theorem 9.2) with (σ_S, ρ_S)-bound in Proposition 9.6 as a benchmark. For the arrivals, we assume three processes all adhering to the class of (σ_A, ρ_A)-constrained arrivals (see also Section 3.1):

- Independent exponentially distributed arrival increments with parameter λ,
- Independent Weibull distributed arrival increments with fixed shape parameter $k = 2$ and scale parameter λ,
- discrete-time Markov-modulated On-Off (MMOO) arrivals.

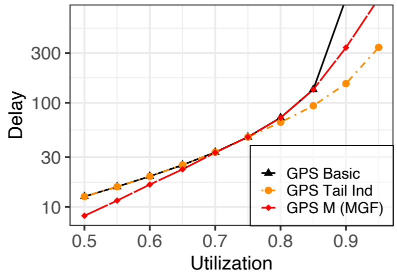
The latter can be described by three parameters: the probability to stay in the "On"-state in the next time step, p_{on}, the probability to stay in the "Off"-state, p_{off}, and a constant peak rate π_A, at which data is sent during the "On"-state. For the service, we always assume work-conserving servers with a constant rate.

We consider two scenarios: in the first, we assume the arrivals of the incoming flows to be independent; and in the second, we forego the independence assumption.

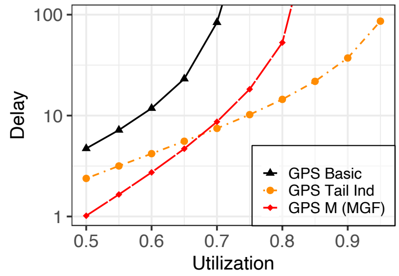
9.3.1 *Independent Arrival Processes*

Apart from GPS Basic, we calculated stochastic bounds with the leftover service assuming the arrivals to be independent:

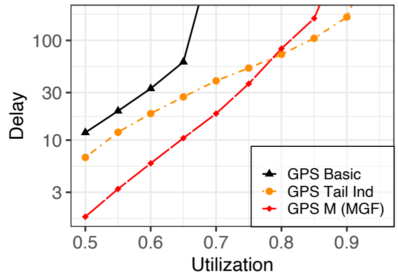
- GPS M (Theorem 9.3) with (σ_S, ρ_S)-bound in Proposition 9.7,
- GPS with tail bounds assuming all arrivals to be independent (stochastic service curve in Proposition 9.10) with bounds on the delay's violation probability obtained via Proposition 9.11.



(a) Exponential distribution



(b) Weibull distribution



(c) MMOO

Figure 9.2: Stochastic delay bounds for a violation probability of 10^{-6} under independent arrivals.

Similar to θ and δ, the choice of M is part of an optimization problem that we solve with a heuristic (otherwise, we would need to try $2^{|\mathcal{N}|-1}$ subsets of $\mathcal{N} \setminus \{i\}$).

In our first numerical experiment, we investigate how the different techniques compare under different utilizations. We consider a scenario with 16 flows (one foi and 15 cross-flows). The 15 cross-flows are heterogeneous meaning that they consist of three groups where each group has different traffic parameters. We opt for a heterogeneous scenario since we know for the DNC, that if all parameters are equal, then GPS Basic already gives the optimal $M \subset \mathcal{N}$ (see Remark 9.5). The results are given in Figure 9.2.

We observe that both, GPS M and GPS with tail bounds, are able to outperform GPS Basic depending on the utilization. To be precise, GPS M often gives the best bounds for smaller utilizations (0.5 to 0.8 depending on the traffic class), whereas GPS with tail bounds can give the best bound for higher loads (from 0.7 or 0.8 on).

It is important to note that the improvement highly depends on the parameters; in particular, on the level of heterogeneity of the parameters as well as the weights ϕ_j. An important special case of the weight assignment is to choose all ϕ_j according to the arrival rates. This is known as rate proportional processor sharing (RPPS) [PG94] and the results for this scenario are depicted in Figure 9.3. Here, we observe that GPS Basic can only be improved by GPS M and only for utilization below 80%.

In our next scenario, we keep the heterogeneous traffic classes and parameters and vary the violation probability instead. Inspired by the results in Figure 9.2, we consider a scenario with a smaller load of 65% as we always obtain finite results (see Figure 9.4). We observe that the violation probability largely impacts the gap between the different delay bounds. To be precise, the smaller the probability, the larger the improvement over GPS Basic. Again, the parameters as well as the chosen traffic class have a strong influence on the improvement.

9.3.2 *General Arrival Processes*

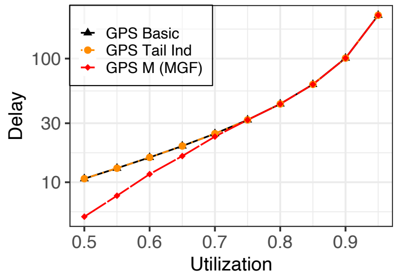
In our last experiment, we consider the general case where we do not assume the arrivals to be independent. Apart from our benchmark GPS Basic, we also implement GPS with tail bounds (stochastic service curve in Proposition 9.8) with bounds on the delay's violation probability obtained via Proposition 9.9. We could also implement GPS M (Theorem 9.3), since the (σ_S, ρ_S)-bound in Proposition 9.7 is given for the case that the arrival are dependent; yet, the number of parameters to optimize for 16 flows (θ, 15 Hölder parameters, and the set M with possibly 2^15 subsets) is computationally infeasible.

The results are given in Figure 9.5. We observe that GPS with tail bounds is not able to improve upon GPS Basic in any of the examples. In other words, they overlap in all scenarios completely. Given that in Proposition 9.9, we additionally need to optimize over all $i \in M \subset \mathcal{N}$, GPS Basic provides a better trade off by leading to the same performance bounds but in a significantly less complex calculation (wee only need to optimize θ).

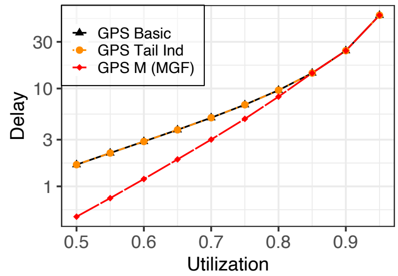
9.4 SUMMARY

In this chapter, we provided a stochastic analysis of a generalized processor sharing (GPS) server applying different SNC techniques. Therefore, we considered leftover service processes in the SNC with MGFs as well as the SNC with tail bounds. If flows are assumed to be independent, we have seen that either of the two branches can outperform GPS Basic. The obtained improvement highly depends on the chosen parameters and on the utilization. However, this could be seen as a step towards reducing the gap between simulations and calculated bounds. In case arrivals are not assumed to be independent, we have seen that new SNC techniques could not improve upon GPS Basic, and further come with the downside of introducing more complexity to the calculation.

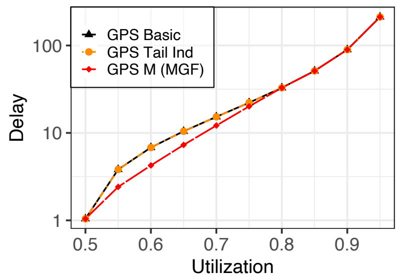
Yet, the stochastic analysis of GPS still remains a challenging task in the network calculus. We hope that this chapter can be a useful tool in order to enhance understanding of the GPS performance analysis. It would also be interesting whether a leftover service as in Theorem 9.3 could be integrated into the martingale-based approaches in the future.



(a) Exponential distribution

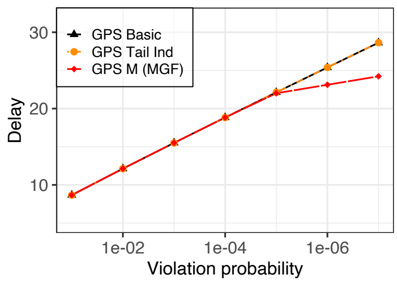


(b) Weibull distribution

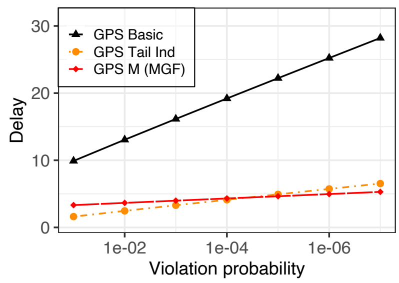


(c) MMOO

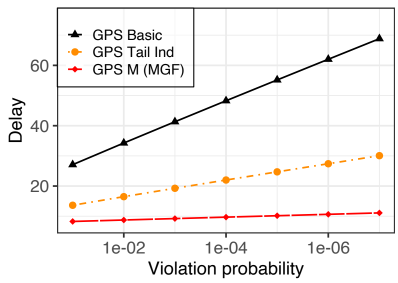
Figure 9.3: Stochastic delay bounds for a violation probability of 10^{-6} under independent arrivals (RPPS).



(a) Exponential distribution

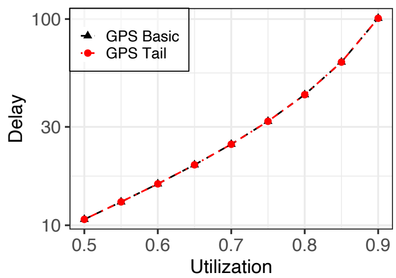


(b) Weibull distribution

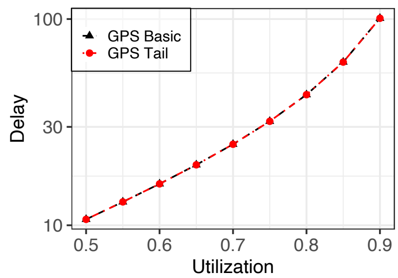


(c) MMOO

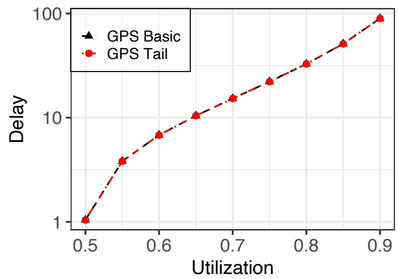
Figure 9.4: Stochastic delay bounds for a utilization = 0.65 under independent arrivals.



(a) Exponential distribution



(b) Weibull distribution



(c) MMOO

Figure 9.5: Stochastic delay bounds for a violation probability of 10^{-6} under general arrivals.

10 CONCLUSION AND OUTLOOK

In this chapter, we conclude this thesis by summarizing its contribution. In addition, we provide some outlook to future research directions.

10.1 CONCLUSION

This thesis makes several contributions to the stochastic network calculus (SNC).

1. We show that the pay multiplexing only once (PMOO) property, originally known from the deterministic network calculus (DNC) [Fid03, SZ06, SZM06, Bou+08], can be achieved in the SNC with a significant gain in accuracy of performance bounds and a reduction of complexity. Moreover, while the technique originates from the DNC, PMOO has a measurably larger impact in the SNC since it also reduces the method-pertinent dependencies. That is, dependencies that are caused when flows interfere in the network, even though they are assumed to be independent when entering the network. As a result, this new analysis reduces runtimes by several orders of magnitude depending on the used optimization heuristic. Given that the usage of the Union bound is already responsible for a notable gap between simulations and bounds, PMOO helps us to not widen the gap even further by other artifacts from the analysis.

2. We give a perspective on how these method-pertinent dependencies can further be reduced for flows that directly or indirectly shared the capacity of a server by incorporating the concept of negative dependence. While it lacks a rigorous proof, numerical evidence indicates a larger potential of improved bound accuracy and reduced runtimes. Maybe even more importantly, it gives an outlook that dependence does not have to be a negative property per se. While this perspective is not completely new, see e.g., [DWS15, Don17], to the best our knowledge, it is the first time to be introduced to the basic SNC with MGFs.

3. We introduce h-mitigators that can be used to improve the calculation of MGF-output bounds. Similar to the negative dependence, it can particularly benefit when considering larger, more complex topologies. Moreover, it yields an opportunity to reduce the effect of the Union bound without compromising on generality as it requires no additional assumptions. While it comes at the price of additional parameters to optimize, the number of parameters can be tailored to the according problem.

4. We made a Python toolbox publicly available that includes all network operations, state-of-the-art end-to-end analyses and contains all the above contributions. In particular, it allows for a replication of all presented results.

5. Apart from the above results for arbitrary multiplexing, we conduct different SNC analyses (with MGFs as well as with tail bounds) of generalized processor sharing (GPS), a popular fair scheduler. We give an overview of the state-of-the-art analyses in the SNC and provide numerical evaluations.

10.2 OUTLOOK

Providing probabilistic end-to-end performance bounds in packet-switched networks remains a timeless challenge. [Ciu07, pp. 173] reported that a problem of interest is the existence of "convolution-form networks", i.e., the "class of networks in which the service given to flows can be expressed in terms of min-plus convolution formulas". In particular, this leads to the question whether networks of arbitrary topologies, in particular, with cycles or probabilistic routing, are contained in this class. Even though we claim in this thesis to extend the class of analyzable topologies within a reasonable amount of time, the original problem basically still remains. In fact, it is not clear whether this problem can be solved in a single, large step, or whether it requires many smaller steps that further "push the boundaries" towards a general tractability and applicability of the stochastic network calculus.

In the introduction (Chapter 1), we mentioned that the uniform framework of the SNC comes with a price of a noticeable gap between exact distribution and bounds. However, it is an open problem whether such a uniform framework can still yield bounds as tight as, e.g., the single-node bounds obtained with martingale techniques as in [CPS14, PC15]. While such an accomplishment appears to be elusive from today's point of view, hypothetically, it would be considered a major breakthrough in the performance analysis of distributed systems.

One of the "smaller", yet still challenging steps to advance the SNC, would be to establish the PMOO to arbitrary feedforward networks. [BNS21] already made a first attempt by the transformation into a so-called "tree-reducible network". The transformation has been conducted by "cutting" a network; however, other techniques are also possible and a comparison is beyond the scope of this thesis. Solving this problem would make a significant step towards catching up with the DNC in terms of a general end-to-end analysis of feedforward networks.

The concept of using negative dependence to simplify the analysis of dependent structure in Chapter 6 is basically a first attempt. Fully unleashing its potential is an open, yet very relevant problem.

It would also be interesting whether the simulation-calculation gap of the martingale analysis for generalized processor sharing (GPS) could be reduced by integrating a state-of-the-art leftover service (see Chapter 9).

Part V

APPENDIX

A

Stochastic Network Calculus Traffic Classes

A.1 (σ_A, ρ_A)-bounds for discrete-time MMOO arrivals

In [Cha94], [Cha00, pp. 244], a Markov-modulated process (MMP) is analyzed. Yet, only a θ-envelope rate is derived which basically is the $\rho_A(\theta)$ of (σ_A, ρ_A)-constrained arrivals [Cha00, pp. 243]:

$$\rho_A(\theta) = \frac{1}{\theta} \log\left(\mathrm{sp}(\phi(\theta)P)\right).$$

For the special case of MMOO arrivals with peak rate b, we can derive a closed form solution for the spectral radius (and hence for $\rho_A(\theta)$) [Soh92], [Cha00, pp. 243]:

$$\mathrm{sp}(\phi(\theta)P) = \frac{p_{11} + p_{22}e^{\theta\pi_A} + \sqrt{\left(p_{11} + p_{22}e^{\theta\pi_A}\right)^2 - 4\left(p_{11} + p_{22} - 1\right)e^{\theta\pi_A}}}{2}.$$

Coming back to the MMP, a bound on the remaining $\sigma_A(\theta)$ part is derived in [Bec16a, pp. 124]:

$$\sigma_A(\theta) = \frac{1}{\theta} \log\left(\left(\max_{k=1,\dots,M} \phi_k(\theta)\right) \frac{\max_{k=1,\dots,M}\{v_k\}}{\min_{k=1,\dots,M}\{v_k\}} \cdot \frac{1}{\mathrm{sp}(\phi(\theta)P)}\right),$$

where $\boldsymbol{v} \in \mathbb{R}_+^M$ is an eigenvector with only positive entries corresponding to the eigenvalue $\mathrm{sp}(\phi(\theta)P)$. The eigenvector $\boldsymbol{v}$ can be computed numerically, yet, for the sake of completeness, we now provide an analytical solution under MMOO arrivals.

First, it is trivial that

$$\max_{k=1,2} \phi_k(\theta) = e^{\theta\pi_A}.$$

Based on the definition of $\boldsymbol{v}$, it holds that

$$\begin{aligned} & \left(\phi(\theta)P - \mathrm{sp}(\phi(\theta)P)\, I\right)\boldsymbol{v} && = \mathbf{0} \\ \Leftrightarrow\; & \begin{pmatrix} p_{11}\phi_1(\theta) - \mathrm{sp}(\phi(\theta)P) & p_{12}\phi_1(\theta) \\ p_{21}\phi_2(\theta) & p_{22}\phi_2(\theta) - \mathrm{sp}(\phi(\theta)P) \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} && = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \end{aligned}$$

which leads to the eigenvector

$$\boldsymbol{v} = \begin{pmatrix} p_{12}\phi_1(\theta) \\ \mathrm{sp}(\phi(\theta)P) - p_{11}\phi_1(\theta) \end{pmatrix}.$$

Now, we can compute $\frac{\max\{v_1, v_2\}}{\min\{v_1, v_2\}}$ directly by checking the sign of $\boldsymbol{v}$.

A.2 FRACTIONAL BROWNIAN MOTION

Definition A.1 (Fractional Brownian Motion [Nor94, Nor95])**.** A stochastic process $Z(t)$ is called (normalized) fractional Brownian motion (fBm) with (self-similarity) Hurst parameter $H \in (0.5, 1)$, if it can be characterized by the following properties:

- $Z(t)$ has stationary increments,
- $Z(0) = 0$ and $E[Z(t)] = 0$ for all t,
- $E\left[Z(t)^2\right] = |t|^{2H}$ for all t,
- $Z(t)$ has continuous paths,
- $Z(t)$ is Gaussian, i.e., all its finite-dimensional marginal distributions are Gaussian.

The increments of $Z(t)$, $Z(t+1) - Z(t)$, are called fractional Gaussian noise (fGn).

B APPENDIX OF CHAPTER 4

B.1 TANDEM QUEUE PERFORMANCE BOUNDS

PROOF OF PROPOSITION 4.1:

Proof. We compute

$$
\begin{aligned}
&\mathrm{P}(d(t) > T) \\
&\overset{(3.26)}{\leq} \sum_{s_0=0}^{t-1} \mathrm{E}\left[e^{\theta A_1(s_0,t)}\right] \mathrm{E}\left[e^{-\theta S_{\mathrm{e2e}}(s_0,t+T)}\right] \\
&= \sum_{s_0=0}^{t-1} \Bigg(\mathrm{E}\left[e^{\theta A_1(s_0,t)}\right] \\
&\quad \cdot \mathrm{E}\left[e^{-\theta\left(\left[S_1 - \sum_{j=2}^m A_j\right]^+ \otimes \cdots \otimes \left[S_n - \left(\left(\left(\sum_{j=2}^m A_j\right) \oslash S_1\right) \ldots\right) \oslash S_{n-1}\right]^+ (s_0, t+T)\right)}\right] \\
&\leq \sum_{s_0=0}^{t-1} \Bigg(\mathrm{E}\left[e^{\theta A_1(s_0,t)}\right] \sum_{s_1=s_0}^{t+T} \cdots \\
&\quad \sum_{s_{n-1}=s_{n-2}}^{t+T} \mathrm{E}\Bigg[e^{-\theta\left(\left[S_1 - \sum_{j=2}^m A_j\right]^+ (s_0,s_1) + \right.} \\
&\quad \cdots \left. {}^{\cdots + \left[S_n - \left(\left(\left(\sum_{j=2}^m A_j\right) \oslash S_1\right) \ldots\right) \oslash S_{n-1}\right]^+ (s_{n-1}, t+T)\right)} \Bigg] \Bigg)
\end{aligned}
$$

All of the factors are dependent and hence we apply the generalized Hölder inequality (Eqn. (3.22)):

$$
\begin{aligned}
&\mathrm{P}(d(t) > T) \\
&\overset{(3.22)}{\leq} \sum_{s_0=0}^{t-1} \Bigg(E\left[e^{\theta A_1(s_0,t)}\right] \sum_{s_1=s_0}^{t+T} \cdots \sum_{s_{n-1}=s_{n-2}}^{t+T} \mathrm{E}\left[e^{-p_1\theta\left[S_1 - \sum_{j=2}^m A_j\right]^+ (s_0,s_1)}\right]^{\frac{1}{p_1}} \\
&\quad \cdots \mathrm{E}\left[e^{p_n\theta\left[S_n - \left(\left(\left(\sum_{j=2}^m A_j\right) \oslash S_1\right) \ldots\right) \oslash S_{n-1}\right]^+ (s_{n-1}, t+T)}\right]^{\frac{1}{p_n}} \Bigg) \\
&\leq \sum_{s_0=0}^{t-1} \Bigg(\mathrm{E}\left[e^{\theta A_1(s_0,t)}\right] \sum_{s_1=s_0}^{t+T} \cdots \sum_{s_{n-1}=s_{n-2}}^{t+T} \mathrm{E}\left[e^{p_1\theta \sum_{j=2}^m A_j(s_0,s_1)} e^{-p_1\theta S_1(s_0,s_1)}\right]^{\frac{1}{p_1}} \\
&\quad \cdots \mathrm{E}\left[e^{p_n\theta\left(\left(\left(\left(\sum_{j=2}^m A_j\right) \oslash S_1\right) \ldots\right) \oslash S_{n-1}\right)(s_{n-1},t+T)} e^{-p_n\theta S_n(s_{n-1},t+T)}\right]^{\frac{1}{p_n}} \Bigg),
\end{aligned}
$$

such that

$$\sum_{i=1}^{n} \frac{1}{p_i} = 1.$$

□

PROOF OF PROPOSITION 4.2:

Proof. We compute

$$
\begin{aligned}
&\mathrm{P}(d(t) > T) \\
&\overset{(3.26)}{\leq} \sum_{s_0=0}^{t-1} \mathrm{E}\left[e^{\theta A_1(s_0,t)}\right] \mathrm{E}\left[e^{-\theta S_{\mathrm{e2e}}(s_0,t+T)}\right] \\
&= \sum_{s_0=0}^{t-1} \mathrm{E}\left[e^{\theta A_1(s_0,t)}\right] \mathrm{E}\left[e^{-\theta\left[\otimes_{i=1}^{n} S_i(s_0,t+T) - \sum_{j=2}^{n} A_j(s_0,t+T)\right]^+}\right] \\
&\leq \sum_{s_0=0}^{t-1} \mathrm{E}\left[e^{\theta A_1(s_0,t)}\right] \mathrm{E}\left[e^{\theta \sum_{j=2}^{m} A_j(s_0,t+T)}\right] \mathrm{E}\left[e^{-\theta \otimes_{i=1}^{n} S_i(s_0,t+T)}\right] \\
&\leq \sum_{s_0=0}^{t-1} \Bigg(\mathrm{E}\left[e^{\theta A_1(s_0,t)}\right] \mathrm{E}\left[e^{\theta \sum_{j=2}^{m} A_j(s_0,t+T)}\right] \\
&\qquad \cdot \sum_{s_1=s_0}^{t+T} \cdots \sum_{s_{n-1}=s_{n-2}}^{t+T} \mathrm{E}\left[e^{-\theta(S_1(s_0,s_1)+\cdots+S_n(s_{n-1},t+T))}\right]\Bigg) \\
&\overset{(\mathrm{indep.})}{=} \sum_{s_0=0}^{t-1} \Bigg(\mathrm{E}\left[e^{\theta A_1(s_0,t)}\right] \prod_{j=2}^{m} \mathrm{E}\left[e^{\theta A_j(s_0,t+T)}\right] \\
&\qquad \cdot \sum_{s_1=s_0}^{t+T} \cdots \sum_{s_{n-1}=s_{n-2}}^{t+T} \left(\mathrm{E}\left[e^{-\theta S_1(s_0,s_1)}\right] \cdots \mathrm{E}\left[e^{-\theta S_n(s_{n-1},t+T)}\right]\right)\Bigg),
\end{aligned}
$$

where we used that

$$
\mathrm{E}\left[e^{-\theta(S_1(s_0,s_1)+\cdots+S_n(s_{n-1},t+T))}\right] \leq \sum_{s_1=s_0}^{t+T} \cdots \sum_{s_{n-1}=s_{n-2}}^{t+T} \left(\mathrm{E}\left[e^{-\theta S_1(s_0,s_1)}\right] \cdots \mathrm{E}\left[e^{-\theta S_n(s_{n-1},t+T)}\right]\right). \tag{B.1}
$$

□

PROOF OF PROPOSITION 4.5:

Proof. At first, we compute a bound on the leftover service:

$$
\begin{aligned}
&\mathrm{E}\left[e^{-\theta S_{\mathrm{e2e}}(s_0,t+T)}\right] \\
&= \mathrm{E}\left[e^{-\theta\left[\otimes_{i=1}^{n} S_i(s_0,t+T) - \sum_{j=2}^{m} A_j(s_0,t+T)\right]^+}\right] \\
&\leq \mathrm{E}\left[e^{\theta \sum_{j=2}^{m} A_j(s_0,t+T)}\right] \mathrm{E}\left[e^{-\theta \otimes_{i=1}^{n} S_i(s_0,t+T)}\right] \\
&\overset{(3.22)}{\leq} \prod_{j=2}^{m} \mathrm{E}\left[e^{p_{j+1}\theta A_j(s_0,t+T)}\right]^{\frac{1}{p_{j+1}}} \sum_{s_1=s_0}^{t+T} \cdots \sum_{s_{n-1}=s_{n-2}}^{t+T} \mathrm{E}\left[e^{-\theta(S_1(s_0,s_1)+S_n(s_{n-1},t+T))}\right],
\end{aligned}
$$

where the last inequality follows by the generalized Hölder inequality (Eqn. (3.22)) and Eqn. (B.1). Thus, we obtain for the delay bound

$$
\begin{aligned}
&\mathrm{P}(d(t) > T) \\
&\overset{(3.26)}{\leq} \sum_{s_0=0}^{t-1} \mathrm{E}\left[e^{p_1\theta A_1(s_0,t)}\right]^{\frac{1}{p_1}} \mathrm{E}\left[e^{-p_2\theta S_{\mathrm{e2e}}(s_0,t+T)}\right]^{\frac{1}{p_2}} \\
&\leq \sum_{s_0=0}^{t-1} \mathrm{E}\left[e^{p_1\theta A_1(s_0,t)}\right]^{\frac{1}{p_1}} \left(\prod_{j=2}^{m} E\left[e^{p_2 p_{j+1}\theta A_j(s_0,t+T)}\right]^{\frac{1}{p_{j+1}}}\right.
\end{aligned}
$$

$$\cdot \sum_{s_1=s_0}^{t+T} \cdots \sum_{s_{n-1}=s_{n-2}}^{t+T} E\left[e^{-p_2\theta(S_1(s_0,s_1)+\cdots+S_n(s_{n-1},t+T))}\right]\Bigg)^{\frac{1}{p_2}}$$

$$\overset{\text{(indep.)}}{=} \sum_{s_0=0}^{t-1} \Bigg(E\left[e^{p_1\theta A_1(s_0,t)}\right]^{\frac{1}{p_1}} \left(\prod_{j=2}^{m} E\left[e^{p_2 p_{j+1}\theta A_j(s_0,t+T)}\right]^{\frac{1}{p_{j+1}}}\right.$$

$$\left.\cdot \sum_{s_1=s_0}^{t+T} \cdots \sum_{s_{n-1}=s_{n-2}}^{t+T} \left(E\left[e^{-p_2\theta S_1(s_0,s_1)}\right] \cdots E\left[e^{-p_2\theta S_n(s_{n-1},t+T)}\right]\right)\right)^{\frac{1}{p_2}}\Bigg),$$

such that

$$\frac{1}{p_1} + \frac{1}{p_2} = 1,$$

$$\frac{1}{p_3} + \cdots + \frac{1}{p_{m+1}} = 1.$$

□

B.2 SINK TREE PERFORMANCE BOUNDS

PROOF OF PROPOSITION 4.4

Proof. We prove the theorem via induction. The base case $n = 2$ is already treated in Subsection 4.1.2.

Assume now that the induction hypothesis (IH) is true for some $n \in \mathbb{N}$. We denote the end-to-end service of tandems of length n by S^n_{e2e}. Observe that extending the sink tree basically means that we prolong all flows and add one flow that only traverses the last hop. Therefore, we apply the induction hypothesis on the last server n servers $S_2, \ldots, S_{n+1}$ and receive S^n_{e2e}. Afterwards, we basically apply the base case, as the network is reduced to the network consisting of S_1 and S^n_{e2e}. This gives

$$S^{n+1}_{\text{e2e}}$$

$$= [(S^n_{\text{e2e}} \otimes S_1) - A_2]^+$$

$$\overset{\text{(IH)}}{=} \left[\left(\left[\left(\left[\left([S_{n+1} - A_{n+2}]^+ \otimes S_n\right) - A_{n+1}\right]^+ \otimes \cdots \otimes S_2\right) - A_3\right]^+ \otimes S_1\right) - A_2\right]^+.$$

For the delay bound, it follows that

$$\mathrm{P}(d(t) > T)$$

$$\overset{(3.26)}{\leq} \sum_{s_0=0}^{t-1} \mathrm{E}\left[e^{\theta A_1(s_0,t)}\right] \mathrm{E}\left[e^{-\theta S_{\text{e2e}}(s_0,t+T)}\right]$$

$$\leq \sum_{s_0=0}^{t-1} \mathrm{E}\left[e^{\theta A_1(s_0,t)}\right]$$

$$\cdot\, \mathrm{E}\left[e^{-\theta\left[\left(\left[\left(\left[\left([S_{n+1}-A_{n+2}]^+\otimes S_n\right)-A_{n+1}\right]^+\otimes\cdots\otimes S_2\right)-A_3\right]^+\otimes S_1\right)-A_2\right]^+(s_0,t+T)}\right]$$

$$\overset{\text{(IH)}}{\leq} \sum_{s_0=0}^{t-1} \mathrm{E}\left[e^{\theta A_1(s_0,t)}\right] \mathrm{E}\left[e^{\theta A_2(s_0,t+T)}\right] \left(\sum_{s_1=s_0}^{t+T} \mathrm{E}\left[e^{-\theta S_1(s_1,t+T)}\right] \mathrm{E}\left[e^{\theta A_3(s_0,s_1)}\right]\right.$$

$$
\begin{aligned}
&\cdot\left(\sum_{s_2=s_0}^{s_1} \mathrm{E}\left[e^{-\theta S_2(s_2,s_1)}\right]\mathrm{E}\left[e^{\theta A_4(s_0,s_2)}\right]\cdots\left(\sum_{s_{n-1}=s_0}^{s_{n-2}} \mathrm{E}\left[e^{-\theta S_{n-1}(s_{n-1},s_{n-2})}\right]\mathrm{E}\left[e^{\theta A_{n+1}(s_0,s_{n-1})}\right]\right.\right.\\
&\left.\left.\left.\cdot\,\mathrm{E}\left[e^{-\theta\left([S_{n+1}-A_{n+2}]^+\otimes S_n\right)(s_0,s_{n-1})}\right]\right)\right)\right)\\
&\leq \sum_{s_0=0}^{t-1}\mathrm{E}\left[e^{\theta A_1(s_0,t)}\right]\mathrm{E}\left[e^{\theta A_2(s_0,t+T)}\right]\left(\sum_{s_1=s_0}^{t+T}\mathrm{E}\left[e^{-\theta S_1(s_1,t+T)}\right]\mathrm{E}\left[e^{\theta A_3(s_0,s_1)}\right]\right.\\
&\cdot\left(\sum_{s_2=s_0}^{s_1} \mathrm{E}\left[e^{-\theta S_2(s_2,s_1)}\right]\mathrm{E}\left[e^{\theta A_4(s_0,s_2)}\right]\cdots\left(\sum_{s_{n-1}=s_0}^{s_{n-2}} \mathrm{E}\left[e^{-\theta S_{n-1}(s_{n-1},s_{n-2})}\right]\mathrm{E}\left[e^{\theta A_{n+1}(s_0,s_{n-1})}\right]\right.\right.\\
&\left.\left.\left.\cdot\left(\sum_{s_n=s_0}^{s_{n-1}}\mathrm{E}\left[e^{-\theta S_n(s_n,s_{n-1})}\right]\mathrm{E}\left[e^{\theta A_{n+2}(s_0,s_n)}\right]\mathrm{E}\left[e^{-\theta S_{n+1}(s_0,s_n)}\right]\right)\right)\right)\right).
\end{aligned}
$$

This finishes the proof. □

PROOF OF PROPOSITION 4.6:

Proof.

$$
\begin{aligned}
&\mathrm{P}(d(t)>T)\\
&\overset{(3.26)}{\leq}\sum_{s_0=0}^{t-1}\mathrm{E}\left[e^{\theta A_1(s_0,t)}\right]\mathrm{E}\left[e^{-\theta S_{\mathrm{e2e}}(s_0,t+T)}\right]\\
&=\sum_{s_0=0}^{t-1}\mathrm{E}\left[e^{\theta\left(A_1(s_0,t)-\left[\left(\left[\left([S_n-A_{n+1}]^+\otimes S_{n-1}\right)-A_n\right]^+\otimes\cdots\otimes S_1\right)-A_2\right]^+(s_0,t+T)\right)}\right]\\
&\leq\sum_{s_0=0}^{t-1}\mathrm{E}\left[e^{\theta A_1(s_0,t)}\right]\mathrm{E}\left[e^{\theta\left(A_2(s_0,t+T)-\left(\left[\left([S_n-A_{n+1}]^+\otimes S_{n-1}\right)-A_n\right]^+\otimes\cdots\otimes S_1\right)(s_0,t+T)\right)}\right].
\end{aligned}
$$

Here, we have A_1 being dependent on the other A_i's and hence, Hölder's inequality comes into play.

$$
\begin{aligned}
&\mathrm{P}(d(t)>T)\\
&\overset{(3.22)}{\leq}\sum_{s_0=0}^{t-1}\mathrm{E}\left[e^{\theta A_1(s_0,t)}\right]\mathrm{E}\left[e^{p_1\theta A_2(s_0,t+T)}\right]^{\frac{1}{p_1}}\\
&\cdot\left(\mathrm{E}\left[e^{-p_2\theta\left(\left[\left[\left([S_n-A_{n+1}]^+\otimes S_{n-1}\right)-A_n\right]^+\otimes\cdots-A_3\right]^+\otimes S_1\right)(s_0,t+T)}\right]\right)^{\frac{1}{p_2}}\\
&\leq\sum_{s_0=0}^{t-1}\mathrm{E}\left[e^{\theta A_1(s_0,t)}\right]\mathrm{E}\left[e^{p_1\theta A_2(s_0,t+T)}\right]^{\frac{1}{p_1}}\\
&\cdot\left(\sum_{s_1=s_0}^{t+T}\mathrm{E}\left[e^{-p_2\theta\left[\left[\left([S_n-A_{n+1}]^+\otimes S_{n-1}\right)-A_n\right]^+\otimes\cdots-A_3\right]^+(s_0,s_1)}\right]\mathrm{E}\left[e^{-p_2\theta S_1(s_1,t+T)}\right]\right)^{\frac{1}{p_2}}\\
&\leq\sum_{s_0=0}^{t-1}\mathrm{E}\left[e^{\theta A_1(s_0,t)}\right]\mathrm{E}\left[e^{p_1\theta A_2(s_0,t+T)}\right]^{\frac{1}{p_1}}\\
&\cdot\left(\sum_{s_1=s_0}^{t+T}\mathrm{E}\left[e^{-p_2\theta S_1(s_1,t+T)}\right]\mathrm{E}\left[e^{p_2\theta\left(A_2-\left[\left([S_n-A_{n+1}]^+\otimes S_{n-1}\right)-A_n\right]^+\otimes\cdots\otimes S_2\right)(s_0,s_1)}\right]\right)^{\frac{1}{p_2}}
\end{aligned}
$$

and we see that Hölder has to be applied n-times:

$$\begin{aligned}
&\mathrm{P}(d(t) > T) \\
\overset{(3.22)}{\leq} &\sum_{s_0=0}^{t-1} \mathrm{E}\left[e^{\theta A_1(s_0,t)}\right] \mathrm{E}\left[e^{p_1\theta A_1(s_0,t+T)}\right]^{\frac{1}{p_1}} \\
&\cdot \left(\sum_{s_1=s_0}^{t+T} \mathrm{E}\left[e^{-p_2\theta S_1(s_1,t+T)}\right] \mathrm{E}\left[e^{p_2p_3\theta A_2(s_0,s_1)}\right]^{\frac{1}{p_3}} \right. \\
&\left. \cdot \left(\mathrm{E}\left[e^{-p_2p_4\theta\left[\left([S_n - A_{n+1}]^+ \otimes S_{n-1}\right) - A_{n-1}\right]^+ \otimes \cdots \otimes S_2(s_0,s_1)}\right]\right)^{\frac{1}{p_4}} \right)^{\frac{1}{p_2}} \\
&\vdots \\
\leq &\sum_{s_0=0}^{t-1} \mathrm{E}\left[e^{\theta A_1(s_0,t)}\right] \mathrm{E}\left[e^{p_1\theta A_2(s_0,t+T)}\right]^{\frac{1}{p_1}} \\
&\cdot \left(\sum_{s_1=s_0}^{t+T} E\left[e^{-p_2\theta S_1(s_1,t+T)}\right] \mathrm{E}\left[e^{p_2p_3\theta A_3(s_0,s_1)}\right]^{\frac{1}{p_3}} \right. \\
&\cdots \left(\sum_{s_{n-1}=s}^{s_{n-2}} \mathrm{E}\left[e^{-p_2p_4\cdots p_{2n-2}\theta S_{n-1}(s_{n-1},s_{n-2})}\right] \mathrm{E}\left[e^{p_2p_4\cdots p_{2n-2}\theta A_{n+1}(s_0,s_{n-1})}\right] \right. \\
&\left. \cdot\ \mathrm{E}\left[e^{-p_2p_4\cdots p_{2n-2}\theta S_n(s_0,s_{n-1})}\right] \right)^{\frac{1}{p_{2n-2}}} \\
&\left. \cdots \right)^{\frac{1}{p_2}}
\end{aligned}$$

such that

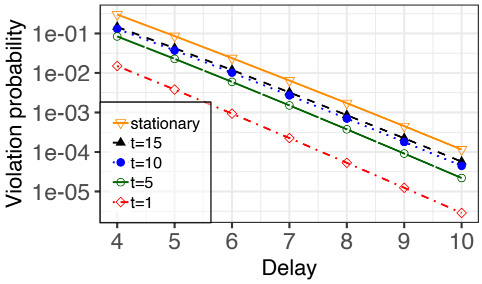
$$\frac{1}{p_1} + \frac{1}{p_2} = 1$$

$$\vdots$$

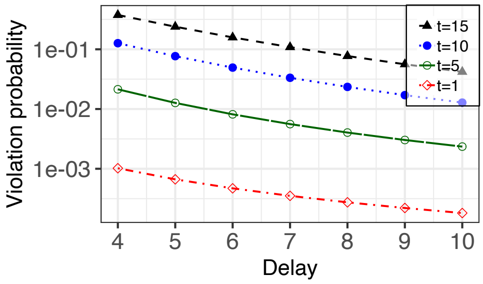
$$\frac{1}{p_{2n-1}} + \frac{1}{p_{2n-2}} = 1.$$

□

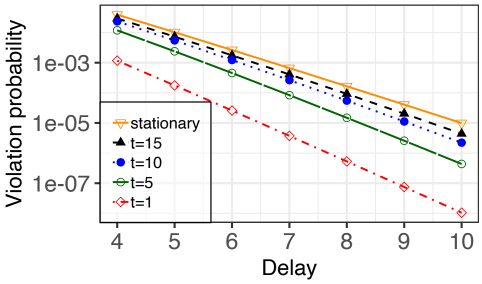
B.3 TANDEM QUEUE NUMERICAL EVALUATION



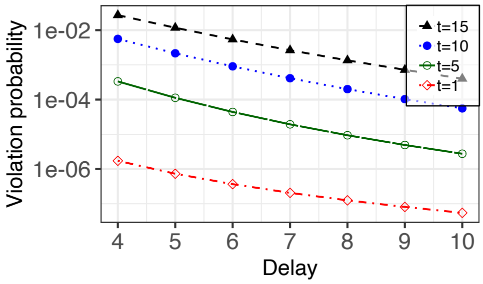
(a) SFA with $H = 0.5$.



(b) SFA with $H = 0.9$.



(c) PMOO with $H = 0.5$.



(d) PMOO with $H = 0.9$.

Figure B.1: Delay violation probability for the two-server tandem queue and different t.

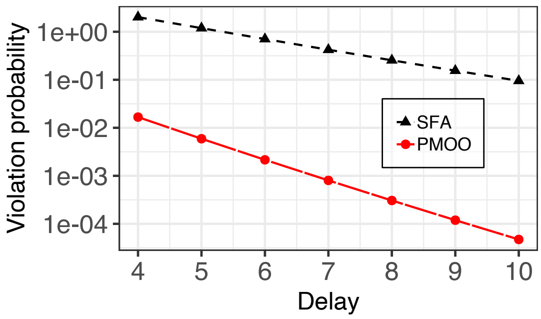
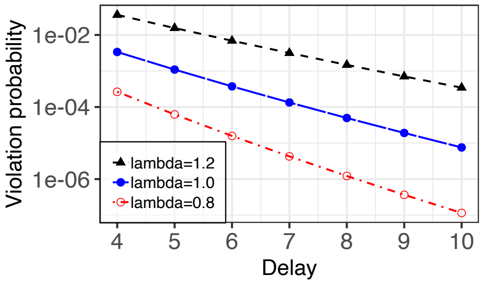
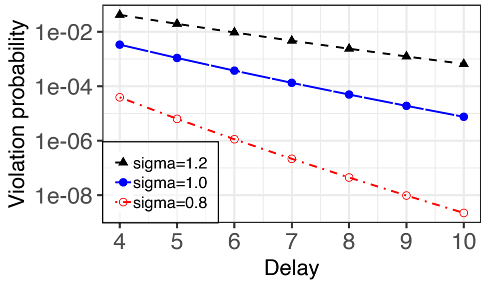


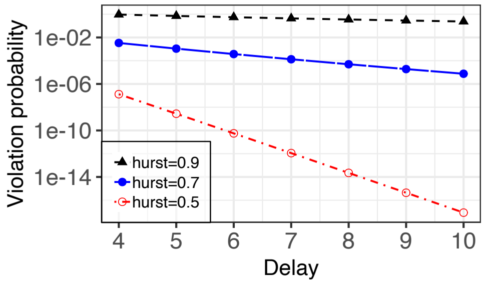
Figure B.2: Comparison between delay violation probabilities using SFA and PMOO (tandem queue).



(a) Mean arrival rate λ



(b) Standard deviation σ



(c) Hurst parameter H

Figure B.3: Parameter sensitivity of fractional Brownian motion on the delay bounds (tandem queue).

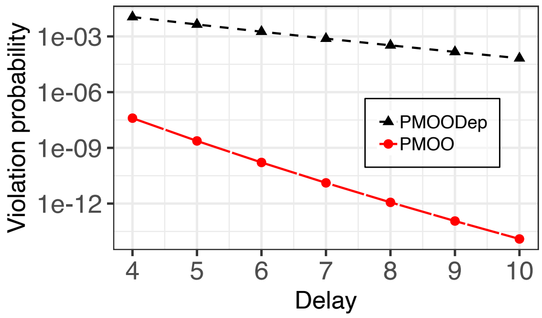


Figure B.4: Comparison between delay violation probabilities for independent and dependent cross-flows using PMOO (tandem queue).

B.4 TANDEM QUEUE MIXED SCENARIO

Proposition B.1 (XDEP Scenario)**.** *The PMOO-SFA yields for the delay bound with dep. cross flow*

$$\begin{aligned} &\mathrm{P}(d(t) > T) \\ \leq &\sum_{k_0=0}^{t-1} E\left[e^{\theta A_1(k_0,t)}\right] E\left[e^{p_1\theta A_2(k_0,t+T)}\right]^{\frac{1}{p_1}} \\ &\cdot E\left[e^{p_2\theta A_3(k_0,t+T)}\right]^{\frac{1}{p_2}} \sum_{k_1=k_0}^{t+T} \sum_{k_2=k_1}^{t+T} E\left[e^{-\theta S_1(k_0,k_1)}\right] \\ &\cdot E\left[e^{-\theta S_2(k_1,k_2)}\right] E\left[e^{-\theta S_3(k_2,t+T)}\right], \end{aligned}$$

with

$$\frac{1}{p_1}+\frac{1}{p_2}=1.$$

Proof. At first, we compute a bound on the leftover service curve:

$$\begin{aligned}
&\mathrm{E}\left[e^{-\theta(S_{\mathrm{l.o.}}(k_0,t+T))}\right]\\
=&\,\mathrm{E}\left[e^{-\theta\left([S_1\otimes S_2\otimes S_3(k_0,t+T)-(A_2+A_3)(k_0,t+T)]^+\right)}\right]\\
\leq&\,\mathrm{E}\left[e^{\theta(A_2+A_3)(k_0,t+T)}\right]\cdot\mathrm{E}\left[e^{-\theta(S_1\otimes S_2\otimes S_3(k_0,t+T))}\right]\\
\overset{(3.22)}{\leq}&\,\mathrm{E}\left[e^{p_1\theta A_2(k_0,t+T)}\right]^{\frac{1}{p_1}}\mathrm{E}\left[e^{p_2\theta A_3(k_0,t+T)}\right]^{\frac{1}{p_2}}\\
&\cdot\sum_{k_1=k_0}^{t+T}\sum_{k_2=k_1}^{t+T}\mathrm{E}\left[e^{-\theta(S_1(k_0,k_1)+S_2(k_1,k_2)+S_3(k_2,t+T))}\right]\\
=&\,E\left[e^{p_1\theta A_2(k_0,t+T)}\right]^{\frac{1}{p_1}}E\left[e^{p_2\theta A_3(k_0,t+T)}\right]^{\frac{1}{p_2}}\\
&\cdot\sum_{k_1=k_0}^{t+T}\sum_{k_2=k_1}^{t+T}\mathrm{E}\left[e^{-\theta S_1(k_0,k_1)}\right]\\
&\cdot\mathrm{E}\left[e^{-\theta S_2(k_1,k_2)}\right]\mathrm{E}\left[e^{-\theta S_3(k_2,t+T)}\right].
\end{aligned}$$

Thus, we obtain for the delay bound by applying the foi's independence of the two remaining flows:

$$\begin{aligned}
&\mathrm{P}(d(t)>T)\\
\leq&\sum_{k_0=0}^{t-1}\mathrm{E}\left[e^{\theta A_1(k_0,t)}\right]\mathrm{E}\left[e^{-\theta S_{\mathrm{e2e}}(s_0,t+T)}\right]\\
\leq&\sum_{k_0=0}^{t-1}\mathrm{E}\left[e^{\theta A_1(k_0,t)}\right]\mathrm{E}\left[e^{p_1\theta A_2(k_0,t+T)}\right]^{\frac{1}{p_1}}\\
&\cdot\mathrm{E}\left[e^{p_2\theta A_3(k_0,t+T)}\right]^{\frac{1}{p_2}}\sum_{k_1=k_0}^{t+T}\sum_{k_2=k_1}^{t+T}\mathrm{E}\left[e^{-\theta S_1(k_0,k_1)}\right]\\
&\cdot\mathrm{E}\left[e^{-\theta S_2(k_1,k_2)}\right]\mathrm{E}\left[e^{-\theta S_3(k_2,t+T)}\right],
\end{aligned}$$

with

$$\frac{1}{p_1}+\frac{1}{p_2}=1.$$

□

Proposition B.2 (FDEP Scenario)**.** *The PMOO-SFA yields for the delay bound with semi dependence, i.e. that the foi f_1 and a cross flow f_2 are dependent, f_3 is independent:*

$$\begin{aligned}
&\mathrm{P}(d(t)>T)\\
\leq&\sum_{k_0=0}^{t-1}\mathrm{E}\left[e^{p_1\theta A_1(k_0,t)}\right]^{\frac{1}{p_1}}\mathrm{E}\left[e^{p_2\theta A_2(k_0,t+T)}\right]^{\frac{1}{p_2}}\\
&\cdot\mathrm{E}\left[e^{\theta A_3(k_0,t+T)}\right]\\
&\cdot\sum_{k_1=k_0}^{t+T}\sum_{k_2=k_1}^{t+T}E\left[e^{-\theta S_1(k_0,k_1)}\right]E\left[e^{-\theta S_2(k_1,k_2)}\right]
\end{aligned}$$

$$E\left[e^{-\theta S_3(k_2,t+T)}\right],$$

with

$$\frac{1}{p_1}+\frac{1}{p_2}=1.$$

Proof. We obtain for the delay bound:

$$\begin{aligned}
&\mathrm{P}(d(t)>T)\\
&\leq \sum_{k_0=0}^{t-1}\mathrm{E}\Big[e^{\theta A_1(k_0,t)}e^{-\theta S_{\mathrm{e2e}}(k_0,t+T)}\Big]\\
&= \sum_{k_0=0}^{t-1}\mathrm{E}\Big[e^{\theta A_1(k_0,t)}e^{-\theta[S_1\otimes S_2\otimes S_3(k_0,t+T)-(A_2+A_3)(k_0,t+T)]^+}\Big]\\
&\leq \sum_{k_0=0}^{t-1}\mathrm{E}\Big[e^{\theta A_1(k_0,t)}e^{\theta((A_2+A_3)(k_0,t+T)-S_1\otimes S_2\otimes S_3(k_0,t+T))}\Big]\\
&\overset{\text{(indep.)}}{=} \sum_{k_0=0}^{t-1}E\left[e^{\theta(A_1(k_0,t)+A_2(k_0,t+T)+A_3(k_0,t+T))}\right]\\
&\quad\cdot E\left[e^{\theta(-S_1\otimes S_2\otimes S_3(k_0,t+T))}\right]\\
&\overset{(3.22)}{\leq} \sum_{k_0=0}^{t-1}E\left[e^{p_1\theta A_1(k_0,t)}\right]^{\frac{1}{p_1}}E\left[e^{p_2\theta A_2(k_0,t+T)}\right]^{\frac{1}{p_2}}\\
&\quad\cdot E\left[e^{\theta A_3(k_0,t+T)}\right]\\
&\quad\cdot\sum_{k_1=k_0}^{t+T}\sum_{k_2=k_1}^{t+T}E\left[e^{-\theta(S_1(k_0,k_1)+S_2(k_1,k_2)+S_3(k_2,t+T))}\right]\\
&= \sum_{k_0=0}^{t-1}E\left[e^{p_1\theta A_1(k_0,t)}\right]^{\frac{1}{p_1}}E\left[e^{p_2\theta A_2(k_0,t+T)}\right]^{\frac{1}{p_2}}\\
&\quad\cdot E\left[e^{\theta A_3(k_0,t+T)}\right]\\
&\quad\cdot\sum_{k_1=k_0}^{t+T}\sum_{k_2=k_1}^{t+T}E\left[e^{-\theta S_1(k_0,k_1)}\right]E\left[e^{-\theta S_2(k_1,k_2)}\right]\\
&\quad E\left[e^{-\theta S_3(k_2,t+T)}\right],
\end{aligned}$$

with

$$\frac{1}{p_1}+\frac{1}{p_2}=1.$$

□

APPENDIX OF CHAPTER 5

C

C.1 PROOFS

PROOF OF THEOREM 5.1

Proof. Note that all servers S_j, $j = 1, \ldots, n$, are work-conserving servers. Assume that for all $t \geq 0$, t_j is the start of the backlogged period (as in Eqn. (2.15)) of server S_j before t_{j+1} (t for the last server). Then, by the definition of a work-conserving server,

$$\sum_{i:j\in\pi_i} D_i^j\left(t_j, t_{j+1}\right) \geq S_j\left(t_j, t_{j+1}\right).$$

Summing over all $j = 1, \ldots, n$, this leads to

$$\sum_{j=1}^{n} \sum_{i:j\in\pi_i} D_i^j\left(t_j, t_{j+1}\right) \geq \sum_{j=1}^{n} S_j\left(t_j, t_{j+1}\right).$$

Since t_j is the start of the backlogged period of server S_j, we have that $D_i^j(t_j) = A_i^j(t_j) = D_i^{j-1}(t_j)$ for all $j = 1, \ldots, n$ (we defined $D_i^0 = A_i$) and for i such that $j \in \pi_i$. Moreover, we can exchange the sums of the left-hand side and receive a telescoping sum. This simplifies to

$$\begin{aligned}
& \textstyle\sum_{i=1}^{m} \sum_{j\in\pi_i} D_i^j\left(t_{j+1}\right) - D_i^j\left(t_j\right) && \geq \sum_{j=1}^{n} S_j\left(t_j, t_{j+1}\right) \\
\Leftrightarrow \quad & \textstyle\sum_{i=1}^{m} \sum_{j\in\pi_i} D_i^j\left(t_{j+1}\right) - D_i^{j-1}\left(t_j\right) && \geq \sum_{j=1}^{n} S_j\left(t_j, t_{j+1}\right) \\
\Leftrightarrow \quad & \textstyle\sum_{i=1}^{m} D_i^{\pi_i(l_i)}\left(t_{\pi_i(l_i)+1}\right) - D_i^{\pi_i(1)-1}\left(t_{\pi_i(1)}\right) && \geq \sum_{j=1}^{n} S_j\left(t_j, t_{j+1}\right).
\end{aligned}$$

Using that $D_i^{\pi_i(1)-1} = A_i$ as well as $D_i^{\pi_i(l_i)} = D_i$ for all i, we receive

$$\sum_{i=1}^{m} D_i\left(t_{\pi_i(l_i)+1}\right) - A_i\left(t_{\pi_i(1)}\right) \geq \sum_{j=1}^{n} S_j\left(t_j, t_{j+1}\right).$$

Now, in order to prove the theorem, we need to isolate D_1 on the left-hand side of the inequality. Hence, we subtract both sides, use that $D_i(t) \leq A_i(t)$ for all i and $t \geq 0$ by causality (see also Eqn. (2.3)) and $\pi_1(1) = 1, \pi_1(l_i) = n$, leading to

$$\begin{aligned}
D_1\left(t_{n+1}\right) &\geq A_1(t_1) + \sum_{j=1}^{n} S_j\left(t_j, t_{j+1}\right) - \sum_{i=2}^{m}\left(D_i\left(t_{\pi_i(l_i)}\right) - A_i\left(t_{\pi_i(1)}\right)\right) \\
&\geq A_1(t_1) + \sum_{j=1}^{n} S_j\left(t_j, t_{j+1}\right) - \sum_{i=2}^{m} A_i\left(t_{\pi_i(1)}, t_{\pi_i(l_i)}\right).
\end{aligned}$$

In addition, by the definition of t_1 being the start of a backlogged period before $t_{\pi_i(l_i)}$, it holds that

$$D_1(t_{n+1}) \geq D_1(t_1) = A_1(t_1).$$

Combining both inequalities yields

$$\begin{aligned}
&D_1(t_{n+1})\\
\geq& A_1(t_1) + \left[\sum_{j=1}^{n} S_j(t_j, t_{j+1}) - \sum_{i=2}^{m} A_i\left(t_{\pi_i(1)}, t_{\pi_i(l_i)}\right)\right]^+\\
\geq& \inf_{0\leq t_1 \leq t_{n+1}} \left\{A_1(t_1) + \left[\inf_{t_1\leq t_2\leq\cdots\leq t_n\leq t_{n+1}} \left\{\sum_{j=1}^{n} S_j(t_j, t_{j+1}) - \sum_{i=2}^{m} A_i\left(t_{\pi_i(1)}, t_{\pi_i(l_i)}\right)\right\}\right]^+\right\}\\
=& \inf_{0\leq t_1\leq t_{n+1}} \left\{A_1(t_1) + S_{\mathrm{e2e}}(t_1, t_{n+1})\right\}\\
=& A_1 \otimes S_{\mathrm{e2e}}(0, t_{n+1}).
\end{aligned}$$

This finishes the proof. □

PROOF OF THEOREM 5.3

Proof. For the ease of notation, we only provide a proof for the tandem case. This means that $\pi_1(l_1) = l_1 = n$. Therefore, we start all proofs with n and switch to l_1 whenever the result would be different when considering trees.

For the first bound, we calculate

$$\begin{aligned}
&\mathrm{P}(d(t_{n+1}) > T)\\
\overset{(3.26)}{\leq}& \sum_{t_1=0}^{t_{n+1}} \mathrm{E}\left[e^{\theta(A_1(t_1,t) - S_{\mathrm{e2e}}(t_1, t_{n+1}+T))}\right]\\
\overset{(5.3)}{=}& \sum_{t_1=0}^{t_{n+1}} \mathrm{E}\left[e^{\theta\left(A_1(t_1,t) - \left[\inf_{t_1\leq t_2\leq\cdots\leq t_n\leq t_{n+1}+T}\left\{\sum_{j=1}^{n} S_j\left(t_j,t_{j+1}\right) - \sum_{i=2}^{m} A_i\left(t_{\pi_i(1)}, t_{\pi_i(l_i)+1}\right)\right\}\right]^+\right)}\right]\\
\leq& \sum_{t_1=0}^{t_{n+1}} \mathrm{E}\left[e^{\theta A_1(t_1,t)}\right] \sum_{t_1\leq t_2\leq\cdots\leq t_n\leq t_{n+1}+T} \mathrm{E}\left[e^{-\theta\sum_{j=1}^{n} S_j\left(t_j,t_{j+1}\right)}\right] \mathrm{E}\left[e^{\theta\sum_{i=2}^{m} A_i\left(t_{\pi_i(1)}, t_{\pi_i(l_i)+1}\right)}\right]\\
\leq& \sum_{t_1=0}^{t_{n+1}} e^{\theta\left(\sigma_{A_1}(\theta) + \rho_{A_1}(\theta)(t-t_1)\right)}\\
&\cdot \sum_{t_1\leq t_2\leq\cdots\leq t_n\leq t_{n+1}+T} e^{-\theta\sum_{j=1}^{n}\left(\sigma_{S_j}(-\theta) + \rho_{S_j}(-\theta)(t_{j+1}-t_j)\right)} e^{\theta\sum_{i=2}^{m}\left(\sigma_{A_i}(\theta) + \rho_{A_i}(\theta)\left(t_{\pi_i(l_i)+1} - t_{\pi_i(1)}\right)\right)}\\
=& e^{\theta\sigma_{\mathrm{total}}(\theta)} \sum_{t_1=0}^{t_{n+1}} e^{\theta\rho_{A_1}(\theta)(t-t_1)} \sum_{t_1\leq t_2\leq\cdots\leq t_n\leq t_{n+1}+T} e^{-\theta\sum_{j=1}^{n}\rho_{S_j}(-\theta)(t_{j+1}-t_j)} e^{\theta\sum_{i=2}^{m}\rho_{A_i}(\theta)\left(t_{\pi_i(l_i)+1} - t_{\pi_i(1)}\right)},
\end{aligned} \tag{C.1}$$

where we used the PMOO leftover service for trees from Theorem 5.2 in the third line. We continue and receive

$$\begin{aligned}
&\mathrm{P}(d(t_{n+1}) > T)\\
\leq& e^{\theta\sigma_{\mathrm{total}}(\theta)} e^{-\theta\rho_{A_1}(\theta)T} \sum_{t_1=0}^{t_{n+1}} e^{\theta\rho_{A_1}(\theta)(t+T-t_1)}
\end{aligned}$$

$$
\begin{aligned}
&\cdot \sum_{t_1 \le t_2 \le \cdots \le t_n \le t_{n+1}+T} e^{-\theta \sum_{j=1}^{n} \rho_{S_j}(-\theta)(t_{j+1}-t_j)} e^{\theta \sum_{i=2}^{m} \rho_{A_i}(\theta)\left(t_{\pi_i(l_i)+1}-t_{\pi_i(1)}\right)} \\
=& e^{\theta\sigma_{\text{total}}(\theta)} e^{-\theta\rho_{A_1}(\theta)T} \sum_{t_1=0}^{t_{n+1}} \sum_{t_1 \le t_2 \le \cdots \le t_n \le t_{n+1}+T} e^{-\theta \sum_{j=1}^{n} \left(\rho_{S_j}(-\theta) - \sum_{i:j\in\pi_i} \rho_{A_i}(\theta)\right)(t_{j+1}-t_j)} \\
\le& e^{\theta\sigma_{\text{total}}(\theta)} e^{-\theta\rho_{A_1}(\theta)T} \sum_{t_1'=0}^{\infty} \cdots \sum_{t_n'=0}^{\infty} e^{-\theta \sum_{j=1}^{n} \left(\rho_{S_j}(-\theta) - \sum_{i:j\in\pi_i} \rho_{A_i}(\theta)\right)t_1'} \cdots e^{-\theta \sum_{j=1}^{n} \left(\rho_{S_j}(-\theta) - \sum_{i:j\in\pi_i} \rho_{A_i}(\theta)\right)t_n'} \\
=& e^{\theta\sigma_{\text{total}}(\theta)} e^{-\theta\rho_{A_1}(\theta)T} \prod_{j=1}^{n} \frac{1}{1-e^{-\theta\left(C_{\text{res},j}(-\theta)-\rho_{A_1}(\theta)\right)}} \\
=& e^{\theta\sigma_{\text{total}}(\theta)} e^{-\theta\rho_{A_1}(\theta)T} \underbrace{\prod_{j\in\pi_1} \frac{1}{1-e^{\theta\left(\rho_{A_1}(\theta)-C_{\text{res},j}(-\theta)\right)}}}_{=\gamma} \\
=& e^{\theta\sigma_{\text{total}}(\theta)} e^{-\theta\rho_{A_1}(\theta)T}\gamma.
\end{aligned}
$$

This proves the first bound.

For the second result, we start with Eqn. (C.1) and follow along the lines of [Fido6, Theorem 3]. Wlog, assume that all n servers are traversed by the foi (otherwise we have the additional factor W as for the first bound).

$$
\begin{aligned}
&\mathrm{P}(d(t_{n+1}) > T) \\
\overset{(C.1)}{\le}& e^{\theta\sigma_{\text{total}}(\theta)} \sum_{t_1=0}^{t_{n+1}} e^{\theta\rho_{A_1}(\theta)(t-t_1)} \sum_{t_1 \le t_2 \le \cdots \le t_n \le t_{n+1}+T} e^{-\theta \sum_{j=1}^{n} \rho_{S_j}(-\theta)(t_{j+1}-t_j)} e^{\theta \sum_{i=2}^{m} \rho_{A_i}(\theta)\left(t_{\pi_i(l_i)+1}-t_{\pi_i(1)}\right)} \\
=& e^{\theta\sigma_{\text{total}}(\theta)} e^{-\theta\rho_{A_1}(\theta)T} \sum_{t_1=0}^{t_{n+1}} e^{\theta\rho_{A_1}(\theta)(t+T-t_1)} \\
&\cdot \sum_{t_1 \le t_2 \le \cdots \le t_n \le t_{n+1}+T} e^{-\theta \sum_{j=1}^{n} \rho_{S_j}(-\theta)(t_{j+1}-t_j)} e^{\theta \sum_{i=2}^{m} \rho_{A_i}(\theta)\left(t_{\pi_i(l_i)+1}-t_{\pi_i(1)}\right)} \\
\le& e^{\theta\sigma_{\text{total}}(\theta)} e^{-\theta\rho_{A_1}(\theta)T} \sum_{t_1=0}^{t_{n+1}} \sum_{t_1 \le t_2 \le \cdots \le t_n \le t_{n+1}+T} e^{-\theta \sum_{j=1}^{n} \left(\rho_{S_j}(-\theta) - \sum_{i:j\in\pi_i} \rho_{A_i}(\theta)\right)(t_{j+1}-t_j)} \\
\le& e^{\theta\sigma_{\text{total}}(\theta)} e^{-\theta\rho_{A_1}(\theta)T} \sum_{t_1=T}^{\infty} \sum_{u_i \ge 0 : \sum_{j\in\pi_1} u_j = t_1} \Bigg(\underbrace{e^{-\theta \min_{j\in\pi_1}\left\{C_{\text{res},j}(-\theta)-\rho_{A_1}(\theta)\right\}}}_{=:q} \Bigg)^{t_1} \\
=& e^{\theta\sigma_{\text{total}}(\theta)} e^{-\theta\rho_{A_1}(\theta)T} \sum_{t_1=T}^{\infty} \binom{t_1+l_1-1}{l_1-1} q^{t_1} \\
=& e^{\theta\sigma_{\text{total}}(\theta)} e^{-\theta\rho_{A_1}(\theta)T} \frac{1}{(1-q)^{l_1}} \sum_{t_1=T}^{\infty} \binom{t_1+l_1-1}{l_1-1} \Bigg(\underbrace{1-q}_{=:p} \Bigg)^{l_1} q^{t_1}.
\end{aligned}
$$

We observe that the series is the CCDF of a negatively, binomially distributed random variable X with success probability p (to be precise, the probability that $T+n$ or more trials are needed for n successes). Hence,

$$
\begin{aligned}
&\mathrm{P}(d(t_{n+1}) > T) \\
=& e^{\theta\sigma_{\text{total}}(\theta)} e^{-\theta\rho_{A_1}(\theta)T} \frac{1}{(1-q)^{l_1}} \mathrm{P}(X \ge T + l_1)
\end{aligned}
$$

$$
\begin{aligned}
&\overset{\left(\text{if } T \geq \frac{l_1 q}{1-q}\right)}{\leq} e^{\theta\sigma_{\text{total}}(\theta)} e^{-\theta\rho_{A_1}(\theta)T} \frac{1}{(1-q)^{l_1}} \left(\underbrace{\frac{\left(1+\frac{T}{l_1}\right)^{\left(1+\frac{T}{l_1}\right)}}{\left(\frac{T}{l_1}\right)^{\frac{T}{l_1}}}}_{=:\zeta} \right)^{l_1} q^T \\
&= e^{-\theta\rho_{A_1}(\theta)T} e^{\theta\left(\min_{j\in\pi_1}\left\{\rho_{S_j}(-\theta)-\sum_{i:j\in\pi_1}\rho_{A_i}(\theta)\right\}\right)T} e^{\theta\sigma_{\text{total}}(\theta)} \zeta^{l_1} \\
&= e^{\theta\left(\min_{j\in\pi_1}\left\{\rho_{S_j}(-\theta)-\sum_{i\neq 1:j\in\pi_1}\rho_{A_i}(\theta)\right\}\right)T} e^{\theta\sigma_{\text{total}}(\theta)} \zeta^{l_1} \\
&= e^{-\theta C_{\min}(-\theta)T} e^{\theta\sigma_{\text{total}}(\theta)} \zeta^{l_1}.
\end{aligned}
$$

where the inequality is obtained by applying Chernoff's bound and optimizing θ (for more details, see again [Fid06, Theorem 3]).

Let us now prove the third bound.

$$
\begin{aligned}
&\mathrm{P}(d(t_{n+1}) > T) \\
&\overset{(C.1)}{\leq} e^{\theta\sigma_{\text{total}}(\theta)} \sum_{t_1=0}^{t_{n+1}} e^{\theta\rho_{A_1}(\theta)(t-t_1)} \sum_{t_1\leq t_2\leq\cdots\leq t_n\leq t_{n+1}+T} e^{-\theta\sum_{j=1}^{n}\rho_{S_j}(-\theta)(t_{j+1}-t_j)} e^{\theta\sum_{i=2}^{m}\rho_{A_i}(\theta)\left(t_{\pi_i(l_i)+1}-t_{\pi_i(1)}\right)} \\
&= e^{\theta\sigma_{\text{total}}(\theta)} \sum_{t_1=0}^{t_{n+1}} e^{\theta\rho_{A_1}(\theta)(t-t_1)} \sum_{t_1\leq t_2\leq\cdots\leq t_{l_1}\leq t_{n+1}+T} e^{-\theta\sum_{j\in\pi_1} C_{\text{res},j}(-\theta)(t_{j+1}-t_j)} \\
&= e^{\theta\sigma_{\text{total}}(\theta)} e^{-\theta C_{\min}(-\theta)\cdot T} \sum_{t_1=0}^{t_{n+1}} e^{\theta\left(\rho_{A_1}(\theta)-C_{\min}(-\theta)\right)(t-t_1)} \\
&\quad \cdot \sum_{t_1\leq t_2\leq\cdots\leq t_{l_1}\leq t_{n+1}+T} e^{\theta\sum_{j\in\pi_1}\left(C_{\min}(-\theta)-C_{\text{res},j}(-\theta)\right)(t_{j+1}-t_j)} \\
&\leq e^{\theta\sigma_{\text{total}}(\theta)} e^{-\theta C_{\min}(-\theta)\cdot T} \\
&\quad \cdot \sum_{t'_1=0}^{\infty}\sum_{t'_2=0}^{\infty}\cdots\sum_{t'_{l_1}=0}^{\infty} e^{\theta\left(\rho_{A_1}(\theta)-C_{\min}(-\theta)\right)t'_1} e^{\theta(C_{\min}(-\theta)-C_{\text{res},1}(-\theta))t'_2} \\
&\quad \cdots e^{\theta\left(C_{\min}(-\theta)-C_{\text{res},\pi(l_1)}(-\theta)\right)t'_{l_1}} \\
&= e^{\theta\sigma_{\text{total}}(\theta)} e^{-\theta C_{\min}(-\theta)\cdot T} \\
&\quad \cdot \frac{1}{1-e^{\theta\left(\rho_{A_1}(\theta)-C_{\min}(-\theta)\right)}} \cdot \underbrace{\prod_{j^*\neq j\in\pi_1} \frac{1}{1-e^{\theta\left(C_{\min}(-\theta)-C_{\text{res},j}(-\theta)\right)}}}_{=\psi} \\
&= e^{\theta\sigma_{\text{total}}(\theta)} e^{-\theta C_{\min}(-\theta)\cdot T} \frac{\psi}{1-e^{\theta\left(\rho_{A_1}(\theta)-C_{\min}(-\theta)\right)}},
\end{aligned}
$$

where we note that the product is equal to 1 for $j = j^*$. This finishes the proof. □

APPENDIX OF CHAPTER 6

D

D.1 PROOFS

DIAMOND NETWORK By using the conjecture, we have obtained so far that

$$
\begin{aligned}
&\mathrm{P}(d(t) > T) \\
&\leq \sum_{t_0=0}^{t-1} \mathrm{E}\left[e^{\theta A_1(t_0,t)}\right] e^{-\theta C_1(t+T-t_0)}\, \mathrm{E}\left[e^{\theta\left(D_2^2+D_3^3\right)(t_0,t+T)}\right] \\
&\leq \sum_{t_0=0}^{t-1} \mathrm{E}\left[e^{\theta A_1(t_0,t)}\right] e^{-\theta C_1(t+T-t_0)}\, \mathrm{E}\left[e^{\theta\left(\left(A_2 \oslash [S_4-A_3]^+\right)\oslash S_2\right)(t_0,t+T)}\right] \mathrm{E}\left[e^{\theta((A_3\oslash S_4)\oslash S_3)(t_0,t+T)}\right].
\end{aligned}
$$

This leads to

$$
\begin{aligned}
&\mathrm{P}(d(t) > T) \\
&\leq \sum_{t_0=0}^{t-1} \mathrm{E}\left[e^{\theta A_1(t_0,t)}\right] e^{-\theta C_1(t+T-t_0)} \left\{\sum_{t_1=0}^{t_0} \mathrm{E}\left[e^{\theta\left(A_2\oslash[S_4-A_3]^+\right)(t_1,t+T)}\right] \mathrm{E}\left[e^{-\theta S_2(t_1,t_0)}\right]\right\} \\
&\quad \cdot \left\{\sum_{t_1=0}^{t_0} \mathrm{E}\left[e^{\theta(A_3\oslash S_4)(t_1,t+T)}\right] \mathrm{E}\left[e^{-\theta S_3(t_1,t_0)}\right]\right\} \\
&\leq \sum_{t_0=0}^{t-1} \mathrm{E}\left[e^{\theta A_1(t_0,t)}\right] e^{-\theta C_1(t+T-t_0)} \\
&\quad \cdot \left\{\sum_{t_1=0}^{t_0} \left\{\sum_{t_2=0}^{t_1} \mathrm{E}\left[e^{\theta A_2(t_2,t+T)}\right] \mathrm{E}\left[e^{\theta A_3(t_2,t_1)}\right] \mathrm{E}\left[e^{-\theta S_4(t_2,t_1)}\right]\right\} \mathrm{E}\left[e^{-\theta S_2(t_1,t_0)}\right]\right\} \\
&\quad \cdot \left\{\sum_{t_1=0}^{t_0} \left\{\sum_{t_2=0}^{t_1} \mathrm{E}\left[e^{\theta A_3(\tau_3,t+T)}\right] \mathrm{E}\left[e^{-\theta S_4(\tau_{3,},t_1)}\right]\right\} \mathrm{E}\left[e^{-\theta S_3(t_1,t_0)}\right]\right\},
\end{aligned}
$$

after applying the Union bound for each usage of the deconvolution. Further assuming all A_i to be (σ_A, ρ_A)-bounded yields a closed-form for the delay bound under the stability condition

$$
\begin{aligned}
\rho_{A_1}(\theta) + \rho_{A_2}(\theta) + \rho_{A_3}(\theta) &< C_1, \\
\rho_{A_2}(\theta) &< C_2, \\
\rho_{A_3}(\theta) &< C_3, \\
\rho_{A_2}(\theta) + \rho_{A_3}(\theta) &< C_4 :
\end{aligned}
$$

$$
\begin{aligned}
&\mathrm{P}(d(t) > T) \\
&\overset{(3.2)}{\leq} \sum_{t_0=0}^{t-1} e^{\theta\left(\rho_{A_1}(\theta)(t-t_0)+\sigma_1(\theta)\right)} e^{-\theta C_1(t+T-t_0)} \\
&\quad \cdot \left\{\sum_{t_1=0}^{t_0} \left\{\sum_{t_2=0}^{t_1} e^{\theta\left(\rho_{A_2}(\theta)(t+T-t_2)+\sigma_{A_2}(\theta)\right)} e^{\theta\left(\rho_{A_3}(\theta)(t_1-t_2)+\sigma_{A_3}(\theta)\right)} e^{-\theta C_4(t_1-t_2)}\right\} e^{-\theta C_2(t_0-t_1)}\right\}
\end{aligned}
$$

$$
\begin{aligned}
&\cdot \left\{ \sum_{t_1=0}^{t_0} \left\{ \sum_{t_2=0}^{t_1} e^{\theta\left(\rho_{A_3}(\theta)(t+T-t_2)+\sigma_{A_3}(\theta)\right)} e^{-\theta C_4(t_1-t_2)} \right\} e^{-\theta C_3(t_0-t_1)} \right\} \\
\leq & e^{\theta\left(\left(\rho_{A_2}(\theta)+\rho_{A_3}(\theta)-C_1\right)T+\sigma_1(\theta)+\sigma_{A_2}(\theta)+2\sigma_{A_3}(\theta)\right)} \\
&\cdot \sum_{t_0=0}^{t-1} e^{\theta\left(\rho_{A_1}(\theta)-C_1\right)(t-t_0)} \left\{ \sum_{t_1=0}^{t_0} \frac{e^{\theta\rho_{A_2}(\theta)(t-t_1)}}{1-e^{\theta\left(\rho_{A_2}(\theta)+\rho_{A_3}(\theta)-C_4\right)}} e^{-\theta C_2(t_0-t_1)} \right\} \\
&\cdot \left\{ \sum_{t_1=0}^{t_0} \frac{e^{\theta\rho_{A_3}(\theta)(t-t_1)}}{1-e^{\theta\left(\rho_{A_3}(\theta)-C_4\right)}} e^{-\theta C_3(t_0-t_1)} \right\} \\
\leq & e^{\theta\left(\left(\rho_{A_2}(\theta)+\rho_{A_3}(\theta)-C_1\right)T+\sigma_1(\theta)+\sigma_{A_2}(\theta)+2\sigma_{A_3}(\theta)\right)} \\
&\cdot \sum_{t_0=0}^{t-1} \frac{e^{\theta\left(\rho_{A_1}(\theta)+\rho_{A_2}(\theta)+\rho_{A_3}(\theta)-C_1\right)(t-t_0)}}{1-e^{\theta\left(\rho_{A_2}(\theta)-C_2\right)}} \cdot \frac{1}{1-e^{\theta\left(\rho_{A_3}(\theta)-C_3\right)}} \\
&\cdot \frac{1}{1-e^{\theta\left(\rho_{A_2}(\theta)+\rho_{A_3}(\theta)-C_4\right)}} \cdot \frac{1}{1-e^{\theta\left(\rho_{A_3}(\theta)-C_4\right)}} \\
\leq & \frac{e^{\theta\left(\left(\rho_{A_2}(\theta)+\rho_{A_3}(\theta)-C_1\right)T+\sigma_1(\theta)+\sigma_{A_2}(\theta)+2\sigma_{A_3}(\theta)\right)}}{1-e^{\theta\left(\rho_{A_1}(\theta)+\rho_{A_2}(\theta)+\rho_{A_3}(\theta)-C_1\right)}} \cdot \frac{1}{1-e^{\theta\left(\rho_{A_2}(\theta)-C_2\right)}} \cdot \frac{1}{1-e^{\theta\left(\rho_{A_3}(\theta)-C_3\right)}} \\
&\cdot \frac{1}{1-e^{\theta\left(\rho_{A_2}(\theta)+\rho_{A_3}(\theta)-C_4\right)}} \cdot \frac{1}{1-e^{\theta\left(\rho_{A_3}(\theta)-C_4\right)}},
\end{aligned}
$$

where we used the convergence of the geometric series.

THE $\mathbb{L}$ We have that

$$
\begin{aligned}
&\mathrm{P}(d(t) > T) \\
\leq & \sum_{t_0=0}^{t-1} \mathrm{E}\left[e^{\theta A_1(t_0,t)}\right] \sum_{t_1=t_0}^{t+T} e^{-\theta C_1\cdot(t_1-t_0)} e^{-\theta C_2\cdot(t+T-t_1)}\, \mathrm{E}\left[e^{\theta D_3^3(t_0,t+T)} e^{\theta D_2^3(t_0,t_1)}\right] \\
\leq & \sum_{t_0=0}^{t-1} \mathrm{E}\left[e^{\theta A_1(t_0,t)}\right] \sum_{t_1=t_0}^{t+T} e^{-\theta C_1\cdot(t_1-t_0)} e^{-\theta C_2\cdot(t+T-t_1)}\, \mathrm{E}\left[e^{\theta D_3^3(t_0,t+T)} e^{\theta D_2^3(t_0,t+T)}\right].
\end{aligned}
$$

With the conjecture, we compute

$$
\begin{aligned}
&\mathrm{P}(d(t) > T) \\
\leq & \sum_{t_0=0}^{t-1} \mathrm{E}\left[e^{\theta A_1(t_0,t)}\right] \sum_{t_1=t_0}^{t+T} e^{-\theta C_1\cdot(t_1-t_0)} e^{-\theta C_2\cdot(t+T-t_1)}\, \mathrm{E}\left[e^{\theta D_3^3(t_0,t+T)}\right] \mathrm{E}\left[e^{\theta D_2^3(t_0,t+T)}\right] \\
\leq & \sum_{t_0=0}^{t-1} \mathrm{E}\left[e^{\theta A_1(t_0,t)}\right] \sum_{t_1=t_0}^{t+T} e^{-\theta C_1\cdot(t_1-t_0)} e^{-\theta C_2\cdot(t+T-t_1)}\, \mathrm{E}\left[e^{\theta(A_2\oslash S_3)(t_0,t+T)}\right] \mathrm{E}\left[e^{\theta\left(A_3\oslash[S_3-A_2]^+\right)(t_0,t+T)}\right] \\
\leq & \sum_{t_0=0}^{t-1} \mathrm{E}\left[e^{\theta A_1(t_0,t)}\right] \sum_{t_1=t_0}^{t+T} e^{-\theta C_1\cdot(t_1-t_0)} e^{-\theta C_2\cdot(t+T-t_1)} \left\{ \sum_{t_2=0}^{t_0} \mathrm{E}\left[e^{\theta A_2(t_2,t+T)}\right] e^{-\theta C_3(t_0-t_2)} \right\} \\
&\cdot \left\{ \sum_{t_2=0}^{t_0} \mathrm{E}\left[e^{\theta A_3(t_2,t+T)}\right] \mathrm{E}\left[e^{-\theta[S_3-A_2]^+(t_2,t_0)}\right] \right\} \\
\leq & \sum_{t_0=0}^{t-1} \mathrm{E}\left[e^{\theta A_1(t_0,t)}\right] \sum_{t_1=t_0}^{t+T} e^{-\theta C_1\cdot(t_1-t_0)} e^{-\theta C_2\cdot(t+T-t_1)} \left\{ \sum_{t_2=0}^{t_0} \mathrm{E}\left[e^{\theta A_2(t_2,t+T)}\right] e^{-\theta C_3(t_0-t_2)} \right\} \\
&\cdot \left\{ \sum_{t_2=0}^{t_0} \mathrm{E}\left[e^{\theta A_3(t_2,t+T)}\right] \mathrm{E}\left[e^{\theta A_2(t_2,t_0)}\right] e^{-\theta C_3(t_0-t_2)} \right\}.
\end{aligned}
$$

If we again assume all A_i to be (σ_A, ρ_A)-bounded, we obtain for

$$\rho_{A_1}(\theta) + \rho_{A_2}(\theta) + \rho_{A_3}(\theta) < \min\{C_1, C_2\},$$
$$\rho_{A_2}(\theta) + \rho_{A_3}(\theta) < C_3,$$

and $C_1 \neq C_2$:

$$\begin{aligned}
&\mathrm{P}(d(t) > T) \\
\overset{(3.2)}{\leq} &\sum_{t_0=0}^{t-1} e^{\theta\left(\rho_{A_1}(\theta)(t-t_0)+\sigma_{A_1}(\theta)\right)} \sum_{t_1=t_0}^{t+T} e^{-\theta C_1\cdot(t_1-t_0)} e^{-\theta C_2\cdot(t+T-t_1)} \\
&\cdot \left\{ \sum_{t_2=0}^{t_0} e^{\theta\left(\rho_{A_2}(\theta)(t+T-t_2)+\sigma_{A_2}(\theta)\right)} e^{-\theta C_3(t_0-t_2)} \right\} \\
&\cdot \left\{ \sum_{t_2=0}^{t_0} e^{\theta\left(\rho_{A_2}(\theta)(t_0-t_2)+\sigma_{A_2}(\theta)\right)} e^{\theta\left(\rho_{A_3}(\theta)(t+T-t_2)+\sigma_{A_3}(\theta)\right)} e^{-\theta C_3(t_0-t_2)} \right\} \\
\leq &e^{\theta\left(\left(\rho_{A_2}(\theta)+\rho_{A_3}(\theta)\right)\cdot T+\sigma_{A_1}(\theta)+2\sigma_{A_2}(\theta)+\sigma_{A_3}(\theta)\right)} \\
&\cdot \sum_{t_0=0}^{t-1} e^{\theta\left(\rho_{A_1}(\theta)+\rho_{A_2}(\theta)+\rho_{A_3}(\theta)\right)(t-t_0)} \sum_{t_1=t_0}^{t+T} e^{-\theta C_1\cdot(t_1-t_0)} e^{-\theta C_2\cdot(t+T-t_1)} \\
&\cdot \left\{ \sum_{t_2=0}^{t_0} e^{\theta\left(\rho_{A_2}(\theta)-C_3\right)(t_0-t_2)} \right\} \left\{ \sum_{t_2=0}^{t_0} e^{\theta\left(\rho_{A_2}(\theta)+\rho_{A_3}(\theta)-C_3\right)(t_0-t_2)} \right\} \\
\leq &e^{\theta\left(\left(\rho_{A_2}(\theta)+\rho_{A_3}(\theta)\right)\cdot T+\sigma_{A_1}(\theta)+2\sigma_{A_2}(\theta)+\sigma_{A_3}(\theta)\right)} \\
&\cdot \sum_{t_0=0}^{t-1} \frac{e^{\theta\left(\rho_{A_1}(\theta)+\rho_{A_2}(\theta)+\rho_{A_3}(\theta)\right)(t-t_0)}}{1-e^{\theta\left(\rho_{A_2}(\theta)-C_3\right)}} \cdot \frac{1}{1-e^{\theta\left(\rho_{A_2}(\theta)+\rho_{A_3}(\theta)-C_3\right)}} \\
&\cdot \sum_{t_1=t_0}^{t+T} e^{-\theta C_1\cdot(t_1-t_0)} e^{-\theta C_2\cdot(t+T-t_1)} \\
\leq &e^{\theta\left(\left(\rho_{A_2}(\theta)+\rho_{A_3}(\theta)\right)\cdot T+\sigma_{A_1}(\theta)+2\sigma_{A_2}(\theta)+\sigma_{A_3}(\theta)\right)} \\
&\cdot \sum_{t_0=0}^{t-1} \frac{e^{\theta\left(\rho_{A_1}(\theta)+\rho_{A_2}(\theta)+\rho_{A_3}(\theta)\right)(t-t_0)}}{1-e^{\theta\left(\rho_{A_2}(\theta)-C_3\right)}} \cdot \frac{1}{1-e^{\theta\left(\rho_{A_2}(\theta)+\rho_{A_3}(\theta)-C_3\right)}} \\
&\cdot \sum_{t_1=t_0}^{t+T} e^{-\theta C_1\cdot(t_1-t_0)} e^{-\theta C_2\cdot(t+T-t_1)} \\
\leq &\frac{e^{\theta\left(\left(\rho_{A_2}(\theta)+\rho_{A_3}(\theta)-\min\{C_1,C_2\}\right)\cdot T+\sigma_{A_1}(\theta)+2\sigma_{A_2}(\theta)+\sigma_{A_3}(\theta)\right)}}{1-e^{\theta\left(\rho_{A_1}(\theta)+\rho_{A_2}(\theta)+\rho_{A_3}(\theta)-\min\{C_1,C_2\}\right)}} \\
&\cdot \frac{1}{1-e^{\theta\left(\rho_{A_2}(\theta)-C_3\right)}} \cdot \frac{1}{1-e^{\theta\left(\rho_{A_2}(\theta)+\rho_{A_3}(\theta)-C_3\right)}} \cdot \frac{1}{1-e^{-\theta|C_1-C_2|}},
\end{aligned}$$

where we used again the convergence of the geometric series.

APPENDIX OF CHAPTER 7

E

E.1 PROOFS

Proof of Proposition 7.5. We have already seen in Eqn. (7.5) that

$$\mathrm{E}\left[e^{\theta D(s,t)}\right] \leq \inf_{p\geq 1}\left\{\left(\sum_{\tau=0}^{s}\mathrm{E}\left[e^{p\theta(A(\tau,t)-S(\tau,s))}\right]\right)^{\frac{1}{p}}\right\},$$

which can be continued with

$$\begin{aligned}\mathrm{E}\left[e^{\theta D(s,t)}\right] &\leq \inf_{p\geq 1}\left\{\left(\sum_{\tau=0}^{s}\mathrm{E}\left[e^{p\theta A(\tau,t)}\right]\mathrm{E}\left[e^{-p\theta S(\tau,s)}\right]\right)^{\frac{1}{p}}\right\}\\ &\leq \inf_{p\geq 1}\left\{e^{\theta(\sigma_A(p\theta)+\sigma_S(-p\theta))}\left(\sum_{\tau=0}^{s}e^{p\theta(\rho_A(p\theta)(t-\tau)-\rho_S(-p\theta)(s-\tau))}\right)^{\frac{1}{p}}\right\},\end{aligned}$$

where we, again, used the independence of arrivals and service in the second line and the (σ,ρ)-constraints for arrivals and service in the third line.

Since we assume stability, $\rho_A(p\theta) < \rho_S(-p\theta)$, we obtain by convergence of the geometric series

$$\mathrm{E}\left[e^{\theta D(s,t)}\right] \leq \inf_{p\geq 1}\left\{e^{\theta(\rho_A(p\theta)(t-s)+\sigma_A(p\theta)+\sigma_S(-p\theta))}\left(\frac{1}{1-e^{p\theta(\rho_A(p\theta)-\rho_S(-p\theta))}}\right)^{\frac{1}{p}}\right\}.$$

This yields

$$\begin{aligned}\sigma_D(\theta) =& \sigma_A(p\theta)+\sigma_S(-p\theta)-\frac{1}{p\theta}\log\left(1-e^{p\theta(\rho_A(p\theta)-\rho_S(-p\theta))}\right),\\ \rho_D(\theta) =& \rho_A(p\theta)\end{aligned}$$

as the theorem states. □

F

APPENDIX OF CHAPTER 9

F.1 PROOFS OF OPTIMAL M IN HOMOGENEOUS SCENARIO

Proof. We assume that $\gamma_{r_j,b_j} = \gamma_{r,b}$ and $\phi_j = \phi \, \forall j \in \mathcal{N}$. Theorem 9.4 then yields the strict service curve

$$\beta^i(t) = \max_{i \in M \subset \mathcal{N}} \left\{ \beta_{\frac{\phi_i}{\sum_{j \in M} \phi_j}\left(R - \sum_{j \notin M} r_j\right), \left(\frac{\sum_{j \notin M} b_j + R \cdot T}{R - \sum_{j \notin M} r_j}\right)}(t) \right\}$$
$$= \max_{i \in M \subset \mathcal{N}} \left\{ \beta_{\frac{1}{|M|}\left(R - \sum_{j \notin M} r_j\right), \left(\frac{(|\mathcal{N}| - |M|) \cdot b + R \cdot T}{R - \sum_{j \notin M} r_j}\right)}(t) \right\}.$$

In other words, only the cardinality of M is relevant for the delay bound. By stability,

$$r \leq \frac{1}{|M|}\left(R - \sum_{j \notin M} r_j\right) =: R_{\text{l.o.},M}, \quad \forall i \in M \subset \mathcal{N},$$

we obtain for the delay bound

$$d \overset{(2.24)}{\leq} \frac{b}{R_{\text{l.o.},M}} + T_{\text{l.o.}}$$
$$= \min_{i \in M \subset \mathcal{N}} \left\{ \frac{b}{\frac{1}{|M|}\left(R - \sum_{j \notin M} r_j\right)} + \frac{(|\mathcal{N}| - |M|) \cdot b + R \cdot T}{R - \sum_{j \notin M} r_j} \right\}$$
$$= \min_{i \in M \subset \mathcal{N}} \left\{ \frac{|M| \cdot b + (|\mathcal{N}| - |M|) \cdot b + R \cdot T}{R - \sum_{j \notin M} r_j} \right\}$$
$$= \min_{i \in M \subset \mathcal{N}} \left\{ \frac{|\mathcal{N}| \cdot b + R \cdot T}{R - \sum_{j \notin M} r_j} \right\}.$$

Since the denominator is increasing in the cardinality of M (and therefore the fraction is decreasing), the optimal choice of M is equal to $\mathcal{N}$ with delay bound

$$d \leq \frac{|\mathcal{N}| \cdot b + R \cdot T}{R}$$
$$= \frac{|\mathcal{N}| \cdot b}{R} + T.$$

□

F.2 PROOFS FOR SNC WITH MGFS

PROOF OF PROPOSITION 9.6:

Proof. We obtain

$$\begin{aligned}
\mathrm{E}\left[e^{-\theta S^i(s,t)}\right] &= \mathrm{E}\left[e^{-\theta\left(\frac{\phi_i}{\sum_{k\in\mathcal{N}}\phi_k}\right)S(s,t)}\right] \\
&\leq e^{-\left(\frac{\phi_i}{\sum_{k\in\mathcal{N}}\phi_k}\theta\right)\left(\rho_S\left(-\frac{\phi_i}{\sum_{k\in\mathcal{N}}\phi_k}\theta\right)(t-s)-\sigma_S\left(-\frac{\phi_i}{\sum_{k\in\mathcal{N}}\phi_k}\theta\right)\right)} \\
&= e^{-(\psi_i\theta)\rho_S(-\psi_i\theta)(t-s)+(\psi_i\theta)\sigma_S(-\psi_i\theta)} \\
&= e^{-\theta\psi_i\rho_S(-\psi_i\theta)(t-s)+\theta\psi_i\sigma_S(-\psi_i\theta)},
\end{aligned}$$

where we define

$$\psi_i := \frac{\phi_i}{\sum_{k\in\mathcal{N}}\phi_k}.$$

□

PROOF OF PROPOSITION 9.7:

Proof. Let p_j such that

$$\sum_{j\notin M}\frac{1}{p_j} = 1.$$

We calculate

$$\begin{aligned}
&\mathrm{E}\left[e^{-\theta S^i(s,t)}\right] \\
&= \mathrm{E}\left[e^{-\theta\left(\frac{\phi_i}{\sum_{j\in M}\phi_j}\left(S(s,t)-\left(\sum_{j\notin M}\left\{A_j \oslash\left(\frac{\phi_j}{\sum_{k\in\mathcal{N}}\phi_k}S\right)(s,t)\right\}\right)\right)\right)}\right] \\
&\overset{\text{(S determ.)}}{=} \mathrm{E}\left[e^{-\theta\frac{\phi_i}{\sum_{j\in M}\phi_j}S(s,t)}\right]\cdot \mathrm{E}\left[e^{\theta\frac{\phi_i}{\sum_{j\in M}\phi_j}\sum_{j\notin M}\left\{A_j\oslash\left(\frac{\phi_j}{\sum_{k\in\mathcal{N}}\phi_k}S\right)(s,t)\right\}}\right] \\
&\leq \mathrm{E}\left[e^{-\theta\frac{\phi_i}{\sum_{j\in M}\phi_j}S(s,t)}\right]\cdot\prod_{j\notin M}\mathrm{E}\left[e^{p_j\theta\frac{\phi_i}{\sum_{j\in M}\phi_j}\left(A_j\oslash\left(\frac{\phi_j}{\sum_{k\in\mathcal{N}}\phi_k}S\right)(s,t)\right)}\right]^{\frac{1}{p_j}},
\end{aligned}$$

where we used Hölder's inequality in the last line. If we have independent arrivals, we also independence of the output bounds (but not of the actual outputs). Similar to above, we define for $i \in M \subset \mathcal{N}$

$$\psi_{i,M} := \frac{\phi_i}{\sum_{j\in M}\phi_j}$$

and

$$\psi_j := \frac{\phi_j}{\sum_{k\in\mathcal{N}}\phi_k}.$$

Further, we assume the stability condition

$$\rho_{A_j}\left(p_j\psi_{i,M}\theta\right) < \psi_j\rho_S\left(-\psi_{i,M}\psi_j\theta\right)$$

for all $j \notin M$. We continue by

$$
\begin{aligned}
&\mathrm{E}\left[e^{-\theta S^i(s,t)}\right]\\
&\leq \mathrm{E}\left[e^{-\theta \frac{\phi_i}{\sum_{j\in M}\phi_j} S(s,t)}\right] \cdot \prod_{j\notin M} \mathrm{E}\left[e^{p_j\theta \frac{\phi_i}{\sum_{j\in M}\phi_j}\left(A_j \oslash \left(\frac{\phi_j}{\sum_{k\in\mathcal{N}}\phi_k} S\right)(s,t)\right)}\right]^{\frac{1}{p_j}}\\
&\leq \mathrm{E}\left[e^{-\theta \frac{\phi_i}{\sum_{j\in M}\phi_j} S(s,t)}\right] \cdot \prod_{j\notin M} \left\{\sum_{\tau=0}^{s} \mathrm{E}\left[e^{p_j\theta \frac{\phi_i}{\sum_{j\in M}\phi_j}\left(A_j(\tau,t) - \frac{\phi_j}{\sum_{k\in\mathcal{N}}\phi_k} S(\tau,s)\right)}\right]^{\frac{1}{p_j}}\right\}\\
&\overset{\text{(S determ.)}}{=} \mathrm{E}\left[e^{-\theta\psi_{i,M}S(s,t)}\right] \cdot \prod_{j\notin M} \left\{\sum_{\tau=0}^{s} \mathrm{E}\left[e^{p_j\theta\psi_{i,M}A_j(\tau,t)}\right]^{\frac{1}{p_j}} \mathrm{E}\left[e^{-\theta\psi_{i,M}\cdot\frac{\phi_j}{\sum_{k\in\mathcal{N}}\phi_k}S(\tau,s)}\right]\right\}\\
&\leq e^{-\psi_{i,M}\theta\rho_S(-\psi_{i,M}\theta)(t-s)+\psi_{i,M}\theta\sigma_S(-\psi_{i,M}\theta)}\\
&\quad \cdot \prod_{j\notin M} \left\{\sum_{\tau=0}^{s} e^{\psi_{i,M}\theta\rho_{A_j}\left(p_j\psi_{i,M}\theta\right)(t-\tau)+\psi_{i,M}\theta\sigma_{A_j}\left(p_j\psi_{i,M}\theta\right)}\right.\\
&\quad \left. \cdot e^{-\psi_{i,M}\psi_j\theta\rho_S\left(-\psi_{i,M}\psi_j\theta\right)(s-\tau)+\psi_{i,M}\psi_j\theta\sigma_S\left(-\psi_{i,M}\psi_j\theta\right)}\right\}\\
&\leq e^{-\psi_{i,M}\theta(\rho_S(-\psi_{i,M}\theta)(t-s)-\sigma_S(-\psi_{i,M}\theta))}\\
&\quad \cdot \prod_{j\notin M} \left\{\frac{e^{\psi_{i,M}\theta\rho_{A_j}\left(p_j\psi_{i,M}\theta\right)(t-s)+\psi_{i,M}\theta\sigma_{A_j}\left(p_j\psi_{i,M}\theta\right)+\psi_{i,M}\psi_j\theta\sigma_S\left(-\psi_{i,M}\psi_j\theta\right)}}{1-e^{\psi_{i,M}\theta\left(\rho_{A_j}\left(p_j\psi_{i,M}\theta\right)-\psi_j\rho_S\left(-\psi_{i,M}\psi_j\theta\right)\right)}}\right\}\\
&= e^{-\psi_{i,M}\theta\left(\rho_S(-\psi_{i,M}\theta)-\sum_{j\notin M}\rho_{A_j}(\psi_{i,M}\theta)\right)(t-s)+\psi_{i,M}\theta\sigma_S(-\psi_{i,M}\theta)}\\
&\quad \cdot e^{\sum_{j\notin M}\psi_{i,M}\psi_j\theta\sigma_S\left(-\psi_{i,M}\psi_j\theta\right)+\psi_{i,M}\theta\sigma_{A_j}\left(p_j\psi_{i,M}\theta\right)}\\
&\quad \cdot \prod_{j\notin M} \left\{\frac{1}{1-e^{\psi_{i,M}\theta\left(\rho_{A_j}\left(p_j\psi_{i,M}\theta\right)-\psi_j\rho_S\left(-\psi_{i,M}\psi_j\theta\right)\right)}}\right\}.
\end{aligned}
$$

We continue with

$$
\begin{aligned}
&\mathrm{E}\left[e^{-\theta S^i(s,t)}\right]\\
&\leq e^{-\psi_{i,M}\theta\left(\rho_S(-\psi_{i,M}\theta)-\sum_{j\notin M}\rho_{A_j}(\psi_{i,M}\theta)\right)(t-s)+\psi_{i,M}\theta\sigma_S(-\psi_{i,M}\theta)}\\
&\quad \cdot \prod_{j\notin M} \left\{\frac{e^{\psi_{i,M}\theta\sigma_{A_j}\left(p_j\psi_{i,M}\theta\right)+\psi_{i,M}\psi_j\theta\sigma_S\left(-\psi_{i,M}\psi_j\theta\right)}}{1-e^{\psi_{i,M}\theta\left(\rho_{A_j}\left(p_j\psi_{i,M}\theta\right)-\psi_j\rho_S\left(-\psi_{i,M}\psi_j\theta\right)\right)}}\right\}\\
&= e^{-\psi_{i,M}\theta\left(\rho_S(-\psi_{i,M}\theta)-\sum_{j\notin M}\rho_{A_j}(\psi_{i,M}\theta)\right)(t-s)}\\
&\quad \cdot e^{\psi_{i,M}\theta\left(\sigma_S(-\psi_{i,M}\theta)+\sum_{j\notin M}\left\{\sigma_{A_j}\left(p_j\psi_{i,M}\theta\right)+\psi_j\sigma_S\left(-\psi_{i,M}\psi_j\theta\right)-\frac{1}{\psi_{i,M}\theta}\log\left(1-e^{\psi_{i,M}\theta\left(\rho_{A_j}\left(p_j\psi_{i,M}\theta\right)-\psi_j\rho_S\left(-\psi_{i,M}\psi_j\theta\right)\right)}\right)\right\}\right)}.
\end{aligned}
$$

This proves the claim. □

F.3 BACKGROUND ON SNC WITH TAIL BOUNDS

In this section, we give a background on SNC with tail bounds. The definitions and notations are mainly inspired by [CBL06]. First, as a reminder, we state Definition 3.1 again.

Definition F.1. An increasing function $\alpha(t-s)$ is said to be a *stochastic envelope* or *stochastic arrival curve* for an arrival process A if, for all $\sigma \in \mathbb{R}$

$$\mathrm{P}(A(s,t) > \alpha(t-s) + \sigma) \leq \varepsilon_a(\sigma), \quad 0 \leq s \leq t, \tag{F.1}$$

where $\varepsilon_a(\sigma) \geq 0$ is a decreasing function, the *error function* or *overflow / deficit profile*.

Definition F.2 (Stochastic Service Curve [Cru96], [CBL06], [JL08, p. 65])**.** Assume A and D are the arrival respectively departure process. A function $\beta(t)$ is a *stochastic service curve* for an arrival process A, if for any fixed sample path and every choice of $\sigma \geq 0$

$$\mathrm{P}\Big(D(t) < A \otimes [\beta - \sigma]^+ (t)\Big) \leq \varepsilon_s(\sigma), \quad \forall t \geq 0, \tag{F.2}$$

where $\varepsilon_s(\sigma)$ is a decreasing error function.

Moreover, for some *slack rate* $\delta > 0$, we introduce the notation $f_\delta(t) = f(t) + \delta \cdot t$. Based on above definitions, on can derive performance bounds.

Theorem F.3 (Violation Probability of Backlog, Delay, and Output)**.** *Let A and D denote the arrivals and departure processes at a node which provides a stochastic service curve $\beta(t)$ satisfying Eqn.* (F.2) *for a given sample path with error function $\varepsilon_s(\sigma)$. Let the arrivals be bounded by a stochastic envelope α with error function $\varepsilon_a(\sigma)$ satisfying Eqn.* (F.1) *and the integrability condition*

$$\int_0^\infty \varepsilon(\sigma) \mathrm{d}\sigma < \infty. \tag{F.3}$$

Fix $\delta > 0$ and define

$$\varepsilon(\sigma) := \inf_{\sigma_a + \sigma_s = \sigma} \left\{ \varepsilon_s(\sigma_s) + \sum_{k=0}^{\infty} \varepsilon_a(\sigma_a + k\delta) \right\}. \tag{F.4}$$

Then we have the following bounds:

1. *Backlog: A bound on the backlog's violation probability is given by*

$$\mathrm{P}(q(t) > \alpha_\delta \oslash \beta\,(0) + \sigma) \leq \varepsilon(\sigma), \quad t \geq 0.$$

2. *Delay: A bound on the delay's violation probability is given by*

$$\mathrm{P}(d(t) > h(\sigma)) \leq \varepsilon(\sigma), \quad t \geq 0, \tag{F.5}$$

where

$$h(\sigma) = \inf\left\{ s \geq 0 : \beta(\tau + s) \geq \alpha_\delta(\tau) + \sigma \textit{ for all } \tau \geq 0 \right\}. \tag{F.6}$$

3. *Output burstiness:* $\alpha \oslash \beta_{-\delta}$ *provides a stochastic envelope for* D*, i.e.,*

$$\mathrm{P}(D(s,t) > \alpha \oslash \beta_{-\delta}\,(t-s) + \sigma) \leq \varepsilon(\sigma), \quad 0 \leq s \leq t.$$

Proof. For a proof, see [CBL06, Theorem 2]. □

Moreover, a useful lemma is given in [CBL06, Lemma 3].

Lemma F.4. *For any positive numbers* M_k, θ_k, $k = 1, \ldots, K$ *and any* $\sigma \geq 0$*,*

$$\inf_{\sigma_1+\cdots+\sigma_K=\sigma} \left\{ \sum_{k=1}^{K} M_k e^{-\theta_k \sigma_k} \right\} = \prod_{k=1}^{K} (M_k \theta_k w)^{\frac{1}{\theta_k w}} e^{-\frac{\sigma}{w}}, \tag{F.7}$$

where $w = \sum_{k=1}^{K} \frac{1}{\theta_k}$.

F.4 PROOFS FOR TAIL BOUND ANALYSIS OF GPS

PROOF OF PROPOSITION 9.8

Proof. Suppose now, for a particular sample path, we have

$$A_j(s,t) \leq \alpha_{j,\delta}(t-s) + \sigma_j, \quad \forall 0 \leq s \leq t, \forall j \notin M. \tag{F.8}$$

and define

$$\sigma := \sum_{j \notin M} \sigma_j.$$

Note that the sum of concave functions is concave. Further, adding a constant does not change concavity. By Theorem 9.4 and the "arrival curve assumption" in Eqn. (F.8), we conclude that

$$\begin{aligned} D_1(t) &\geq \inf_{0\leq s\leq t} \left\{ A_i(s) + \left[\frac{\phi_i}{\sum_{j\in M}\phi_j} \left(\beta(t-s) - \sum_{j\notin M} \alpha_{j,\delta}(t-s) - \sum_{j\notin M} \sigma_j \right) \right]^+ \right\} \\ &\geq \inf_{0\leq s\leq t} \left\{ A_i(s) + \left[\beta^i(t-s) - \sigma \right]^+ \right\} \\ &= A_1 \otimes \left[\beta^i(t) - \sigma \right]^+ (t). \end{aligned}$$

and abused the notation in the last line. Hence, we reverse the implication and receive the bound

$$\begin{aligned} &\mathrm{P}\Big(D_1(t) < A_1 \otimes [\beta_{\text{l.o.},M}(t-s) - \sigma]^+ (t) \Big) \\ &\leq \inf_{\sum_{j\notin M}\sigma_j = \sigma} \left\{ \sum_{j\notin M} \mathrm{P}\Big(A_j(t) > \inf_{0\leq s\leq t} \{ A_j(s) + \alpha_{j,\delta}(t-s) + \sigma_j \} \Big) \right\} \\ &= \inf_{\sum_{j\notin M}\sigma_j = \sigma} \left\{ \sum_{j\notin M} \mathrm{P}\Big(\sup_{0\leq s\leq t} \{ A_j(s,t) - \alpha_{j,\delta}(t-s) \} > \sigma_j \Big) \right\} \\ &\overset{(3.28)}{\leq} \inf_{\sum_{j\notin M}\sigma_j = \sigma} \left\{ \sum_{j\notin M} \sum_{s=0}^{t} \mathrm{P}(A_j(s,t) > \alpha_{j,\delta}(t-s) + \sigma_j) \right\} \\ &\leq \inf_{\sum_{j\notin M}\sigma_j = \sigma} \left\{ \sum_{j\notin M} \sum_{k=0}^{\infty} \varepsilon_{a_j}(\sigma_j + k\delta) \right\}. \end{aligned}$$

This finishes the proof. □

PROOF OF PROPOSITION 9.9

Proof. Observe that, since the arrivals are $(\sigma_A(\theta), \rho_A(\theta))$-bounded, they have a concave stochastic envelope

$$\alpha(t-s) = \rho_A(\theta) \cdot (t-s).$$

Further, the constant rate function is obviously is convex.

In the following, we apply Theorem F.3.2. For $h(\sigma)$, we calculate using Proposition 9.8

$$\begin{aligned}
&h(\sigma) \\
&= \inf\left\{s \geq 0 : \beta^i(\tau+s) \geq \alpha_{i,\delta}(\tau) + \sigma \text{ for all } \tau \geq 0\right\} \\
&= \inf\left\{s \geq 0 : \frac{\phi_i}{\sum_{j\in M}\phi_j} \cdot \left(\beta(\tau+s) - \sum_{j\notin M}\alpha_{j,\delta}(\tau+s)\right) \geq \alpha_i(\tau) + \delta\cdot\tau + \sigma \text{ for all } \tau \geq 0\right\} \\
&= \frac{\frac{\sum_{j\in M}\phi_j}{\phi_i}\cdot\sigma}{C - \sum_{j\notin M}\rho_{A_j}(\theta) - |\mathcal{N}\setminus M|\cdot\delta}.
\end{aligned}$$

Let us replace σ by

$$T \cdot \frac{\phi_i}{\sum_{j\in M}\phi_j} \cdot \left(C - \sum_{j\notin M}\rho_{A_j}(\theta) - |\mathcal{N}\setminus M|\cdot\delta\right) \geq 0.$$

Then, by Theorem F.3, it holds that

$$\begin{aligned}
\mathrm{P}(d(t) > T) &= \mathrm{P}(d(t) > h(\sigma)) \\
&\leq \inf_{\sigma_a+\sigma_s=\sigma}\left\{\varepsilon_s(\sigma_s) + \sum_{k=0}^{\infty}\underbrace{\varepsilon_a(\sigma_a + k\delta)}_{\text{error fun. of foi}}\right\} \\
&= \inf_{\sigma_a+\sigma_s=\sigma}\left\{\varepsilon_s(\sigma_s) + \sum_{k=0}^{\infty} e^{\theta\sigma_{A_j}(\theta)} e^{-\theta(\sigma_a+k\delta)}\right\}.
\end{aligned}$$

Further, we have that

$$\begin{aligned}
\varepsilon_s(\sigma_s) &= \inf_{\sum_{j\notin M}\sigma_j=\sigma_s}\left\{\underbrace{\sum_{j\notin M}\sum_{k=0}^{\infty}\varepsilon_{a_j}(\sigma_j + k\delta)}_{\text{error fun. of cross-flows}}\right\} \\
&= \inf_{\sum_{j\notin M}\sigma_j=\sigma_s}\left\{\sum_{j\notin M}\sum_{k=0}^{\infty} e^{\theta\sigma_{A_j}(\theta)} e^{-\theta\left(\sigma_j+k\delta\right)}\right\} \\
&= \frac{1}{1-e^{-\theta\delta}} \cdot \inf_{\sum_{j\notin M}\sigma_j=\sigma_s}\left\{\sum_{j\notin M} e^{\theta\sigma_{A_j}(\theta)} e^{-\theta\sigma_j}\right\}.
\end{aligned}$$

Applying Lemma F.4 yields

$$\begin{aligned}
\inf_{\sum_{j\notin M}\sigma_j=\sigma_s}\left\{\sum_{j\notin M} e^{\theta\sigma_{A_j}(\theta)} e^{-\theta\sigma_j}\right\} &= e^{\frac{\theta}{|\mathcal{N}\setminus M|}\sum_{j\notin M}\sigma_{A_j}(\theta)} e^{-\frac{\sigma_s}{w}} \underbrace{\prod_{j\notin M}|\mathcal{N}\setminus M|^{\frac{1}{|\mathcal{N}\setminus M|}}}_{=|\mathcal{N}\setminus M|} \\
&= |\mathcal{N}\setminus M|\, e^{\frac{\theta}{|\mathcal{N}\setminus M|}\sum_{j\notin M}\sigma_{A_j}(\theta)} e^{-\frac{\theta}{|\mathcal{N}\setminus M|}\sigma_s}.
\end{aligned}$$

Thus,

$$\varepsilon_s(\sigma_s) = \frac{|\mathcal{N} \setminus M|}{1-e^{-\theta\delta}} e^{\frac{\theta}{|\mathcal{N}\setminus M|}\sum_{j\notin M}\sigma_{A_j}(\theta)} \cdot e^{-\frac{\theta}{|\mathcal{N}\setminus M|}\sigma_s}.$$

It follows that

$$\begin{aligned}
&\mathrm{P}(d(t) > T) \\
&\le \inf_{\sigma_a+\sigma_s=\sigma}\left\{\varepsilon_s(\sigma_s) + \sum_{k=0}^{\infty} e^{\theta\sigma_{A_i}(\theta)} e^{-\theta(\sigma_a+k\delta)}\right\} \\
&= \inf_{\sigma_a+\sigma_s=\sigma}\left\{\frac{|\mathcal{N}\setminus M|}{1-e^{-\theta\delta}} e^{\frac{\theta}{|\mathcal{N}\setminus M|}\sum_{j\notin M}\sigma_{A_j}(\theta)} e^{-\frac{\theta}{|\mathcal{N}\setminus M|}\sigma_s} + \frac{1}{1-e^{-\theta\delta}} e^{\theta\sigma_{A_i}(\theta)} e^{-\theta\sigma_a}\right\}.
\end{aligned}$$

We compute

$$w = \frac{1+|\mathcal{N}\setminus M|}{\theta}$$

in Lemma F.4 and hence

$$\begin{aligned}
&\mathrm{P}(d(t) > T) \\
&\le \inf_{\sigma_a+\sigma_s=\sigma}\left\{\frac{|\mathcal{N}\setminus M|}{1-e^{-\theta\delta}} e^{\frac{\theta}{|\mathcal{N}\setminus M|}\sum_{j\notin M}\sigma_{A_j}(\theta)} e^{-\frac{\theta}{|\mathcal{N}\setminus M|}\sigma_s} + e^{\theta\sigma_{A_i}(\theta)}\frac{1}{1-e^{-\theta\delta}} e^{-\theta\sigma_a}\right\} \\
&= \left(e^{\frac{\theta}{|\mathcal{N}\setminus M|}\sum_{j\notin M}\sigma_{A_j}(\theta)}\right)^{\frac{1}{\frac{\theta}{|\mathcal{N}\setminus M|}w}} \left(e^{\theta\sigma_{A_i}(\theta)}\right)^{\frac{1}{\theta w}} e^{-\frac{\sigma}{w}} \left(\frac{|\mathcal{N}\setminus M|}{1-e^{-\theta\delta}}\cdot\frac{\theta}{|\mathcal{N}\setminus M|}w\right)^{\frac{1}{\frac{\theta}{|\mathcal{N}\setminus M|}w}} \cdot \left(\frac{1}{1-e^{-\theta\delta}}\cdot\theta w\right)^{\frac{1}{\theta w}} \\
&= e^{\frac{\theta}{1+|\mathcal{N}\setminus M|}\left(\sigma_{A_i}(\theta)+\sum_{j\notin M}\sigma_{A_j}(\theta)\right)} (1+|\mathcal{N}\setminus M|)\left(\frac{1}{1-e^{-\theta\delta}}\right) e^{-\frac{\theta\sigma}{1+|\mathcal{N}\setminus M|}} \\
&= (1+|\mathcal{N}\setminus M|)\left(\frac{1}{1-e^{-\theta\delta}}\right) \\
&\quad \cdot e^{-\frac{\theta}{1+|\mathcal{N}\setminus M|}\left(T\cdot\frac{\phi_i}{\sum_{j\in M}\phi_j}\cdot\left(C-\sum_{j\notin M}\rho_{A_j}(\theta)-|\mathcal{N}\setminus M|\cdot\delta\right)\right)} e^{\frac{\theta}{1+|\mathcal{N}\setminus M|}\left(\sigma_{A_i}(\theta)+\sum_{j\notin M}\sigma_{A_j}(\theta)\right)}.
\end{aligned}$$

□

PROOF OF PROPOSITION 9.10

Proof. We only show the case $M \neq \mathcal{N}$. Suppose now, for a particular sample path, we have

$$\sum_{j\notin M} A_j(s,t) \le \sum_{j\notin M}\alpha_{j,\delta}(t-s) + \sigma \quad \forall 0 \le s \le t. \tag{F.9}$$

By Theorem 9.4 and the "arrival curve assumption" in Eqn. (F.9), we conclude that

$$\begin{aligned}
D_i(t) &\ge \inf_{0\le s\le t}\left\{A_i(s) + \left[\frac{\phi_i}{\sum_{j\in M}\phi_j}\left(C\cdot(t-s) - \sum_{j\notin M}\alpha_{j,\delta}(t-s) - \sigma\right)\right]^+\right\} \\
&\ge \inf_{0\le s\le t}\left\{A_i(s) + \left[\beta^i(t-s)-\sigma\right]^+\right\} \\
&= A_i \otimes \left[\beta^i - \sigma\right]^+(t).
\end{aligned}$$

Hence, we reverse the implication and receive the bound

$$
\begin{aligned}
&\mathrm{P}\left(D_i(t) < A_i \otimes \left[\beta^i(t-s)-\sigma\right]^+(t)\right)\\
\leq &\mathrm{P}\left(\sum_{j\notin M} A_j(t) > \inf_{0\leq s\leq t}\left\{\sum_{j\notin M} A_j(s) + \sum_{j\notin M}\alpha_{j,\delta}(t-s)+\sigma\right\}\right)\\
= &\mathrm{P}\left(\sup_{0\leq s\leq t}\left\{\sum_{j\notin M} A_j(s,t) - \sum_{j\notin M}\alpha_{j,\delta}(t-s)\right\} > \sigma\right)\\
\overset{(3.28)}{\leq} &\sum_{s=0}^{t}\mathrm{P}\left(\sum_{j\notin M} A_j(s,t) > \sum_{j\notin M}\alpha_{j,\delta}(t-s)+\sigma\right)\\
\overset{(3.3)}{\leq} &\sum_{k=0}^{t} e^{-\theta\left(\sum_{j\notin M}\rho_{A_j}(\theta)k+|\mathcal{N}\setminus M|\delta\cdot k+\sigma\right)}\,\mathrm{E}\left[e^{\theta\sum_{j\notin M}A_j(t-k,t)}\right]\\
\leq &\sum_{k=0}^{t} e^{-\theta\left(\sum_{j\notin M}\rho_{A_j}(\theta)k+|\mathcal{N}\setminus M|\delta\cdot k+\sigma\right)}e^{\theta\sum_{j\notin M}\rho_{A_j}(\theta)k+\sum_{j\notin M}\sigma_{A_j}(\theta)}\\
= &e^{\theta\sum_{j\notin M}\sigma_{A_j}(\theta)}\sum_{k=0}^{t} e^{-\theta(|\mathcal{N}\setminus M|\delta\cdot k+\sigma)}\\
\leq &\frac{e^{\theta\sum_{j\notin M}\sigma_{A_j}(\theta)}}{1-e^{-\theta|\mathcal{N}\setminus M|\delta}}e^{-\theta\sigma},
\end{aligned}
$$

where we used the Union bound in the fourth line and the Chernoff bound in the subsequent line. This finishes the proof. □

PROOF OF PROPOSITION 9.11

Proof. We start with the case $M \neq \mathcal{N}$:

Again, we want to apply Theorem F.3.2. For $h(\sigma)$, we calculate using Proposition 9.10

$$
\begin{aligned}
&h(\sigma)\\
&=\inf\left\{s\geq 0:\frac{\phi_i}{\sum_{j\in M}\phi_j}\cdot\left(\beta(\tau+s)-\sum_{j\notin M}\alpha_{j,\delta}(\tau+s)\right)\geq\alpha_{1,\delta}(\tau)+\sigma\text{ for all }\tau\geq 0\right\}\\
&=\frac{\sigma}{\frac{\phi_i}{\sum_{j\in M}\phi_j}\left(C-\sum_{j\notin M}\rho_{A_j}(\theta)-|\mathcal{N}\setminus M|\cdot\delta\right)}.
\end{aligned}
$$

Let us replace σ by

$$
T\cdot\frac{\phi_i}{\sum_{j\in M}\phi_j}\cdot\left(C-\sum_{j\notin M}\rho_{A_j}(\theta)-|\mathcal{N}\setminus M|\cdot\delta\right)\geq 0.
$$

Then, by Theorem F.3.2, it holds that

$$
\begin{aligned}
\mathrm{P}(d(t)>T) &= \mathrm{P}(d(t)>h(\sigma))\\
&\leq \inf_{\sigma_a+\sigma_s=\sigma}\left\{\varepsilon_s(\sigma_s)+\sum_{k=0}^{\infty}\underbrace{\varepsilon_a(\sigma_a+k\delta)}_{\text{error fun. of foi}}\right\}\\
&= \inf_{\sigma_a+\sigma_s=\sigma}\left\{\frac{e^{\theta\sum_{j\notin M}\sigma_{A_j}(\theta)}e^{-\theta\sigma_s}}{1-e^{-\theta|\mathcal{N}\setminus M|\delta}}+\frac{e^{\theta\sigma_{A_i}(\theta)}e^{-\theta\sigma_a}}{1-e^{-\theta\delta}}\right\}.
\end{aligned}
$$

We compute

$$w = \frac{2}{\theta}$$

in Lemma F.4 and hence

$$\begin{aligned}
&\mathrm{P}(d(t) > T) \\
&\leq \inf_{\sigma_a+\sigma_s=\sigma} \left\{ \frac{e^{\theta \sum_{j\notin M} \sigma_{A_j}(\theta)} e^{-\theta\sigma_s}}{1-e^{-\theta|\mathcal{N}\setminus M|\delta}} + \frac{e^{\theta\sigma_{A_i}(\theta)} e^{-\theta\sigma_a}}{1-e^{-\theta\delta}} \right\} \\
&= e^{\frac{\theta}{2}\left(\sigma_{A_i}(\theta)+\sum_{j\notin M}\sigma_{A_j}(\theta)\right)} e^{-\frac{\sigma}{w}} \left(\frac{1}{1-e^{-\theta|\mathcal{N}\setminus M|\delta}} \cdot \theta w \right)^{\frac{1}{\theta w}} \cdot \left(\frac{1}{1-e^{-\theta\delta}} \cdot \theta w \right)^{\frac{1}{\theta w}} \\
&= e^{\frac{\theta}{2}\left(\sigma_{A_i}(\theta)+\sum_{j\notin M}\sigma_{A_j}(\theta)\right)} 2 \left(\frac{1}{1-e^{-\theta|\mathcal{N}\setminus M|\delta}} \right)^{\frac{1}{2}} \cdot \left(\frac{1}{1-e^{-\theta\delta}} \right)^{\frac{1}{2}} e^{-\frac{\theta}{2}\sigma} \\
&= 2 \left(\frac{1}{1-e^{-\theta|\mathcal{N}\setminus M|\delta}} \right)^{\frac{1}{2}} \cdot \left(\frac{1}{1-e^{-\theta\delta}} \right)^{\frac{1}{2}} \\
&\quad \cdot e^{-\frac{\theta}{2} T \cdot \frac{\phi_i}{\sum_{j\in M}\phi_j} \cdot \left(C - \sum_{j\notin M} \rho_{A_j}(\theta) - |\mathcal{N}\setminus M|\cdot\delta \right)} e^{\frac{\theta}{2}\left(\sigma_{A_i}(\theta)+\sum_{j\notin M}\sigma_{A_j}(\theta)\right)}.
\end{aligned}$$

Next, we proof the case $M = \mathcal{N}$. By Proposition 9.10, we know that

$$\beta^i(t) = \frac{\phi_i}{\sum_{k\in\mathcal{N}} \phi_k} \beta(t)$$

is a stochastic service curve with error function $\varepsilon_s = 0$. Using Theorem F.3, we obtain for $h(\sigma)$

$$\begin{aligned}
&h(\sigma) \\
&= \inf \left\{ s \geq 0 : \frac{\phi_i}{\sum_{k\in\mathcal{N}}\phi_k} \cdot \beta(\tau+s) \geq \alpha_i(\tau) + \delta\cdot\tau + \sigma \text{ for all } \tau \geq 0 \right\} \\
&= \frac{\sigma}{\frac{\phi_i}{\sum_{k\in\mathcal{N}}\phi_k} \cdot C}.
\end{aligned}$$

Let us replace σ by

$$T \cdot \frac{\phi_i}{\sum_{k\in\mathcal{N}}\phi_k} C.$$

Then, by Theorem F.3.2, it holds that

$$\begin{aligned}
\mathrm{P}(d(t) > T) &= \mathrm{P}(d(t) > h(\sigma)) \\
&\leq \inf_{\sigma_a+\sigma_s=\sigma} \left\{ \varepsilon_s(\sigma_s) + \sum_{k=0}^{\infty} \underbrace{\varepsilon_a(\sigma_a + k\delta)}_{\text{error fun. of foi}} \right\} \\
&= \sum_{k=0}^{\infty} \varepsilon_a(\sigma + k\delta) \\
&= \sum_{k=0}^{\infty} e^{\theta\sigma_{A_i}(\theta)} e^{-\theta(\sigma+k\delta)} \\
&= \frac{e^{\theta\sigma_{A_i}(\theta)}}{1-e^{-\theta\delta}} e^{-\theta\sigma} \\
&= \frac{1}{1-e^{-\theta\delta}} e^{-\theta T \cdot \frac{\phi_i}{\sum_{k\in\mathcal{N}}\phi_k} \cdot C} e^{\theta\sigma_{A_i}(\theta)}.
\end{aligned}$$

Since this function is decreasing in δ, the optimal value of δ its upper bound (that is given by the stability condition Eqn. (9.10) by inserting $M = \mathcal{N}$):

$$
\begin{aligned}
\rho_{A_i}(\theta) + \delta &\overset{!}{=} \frac{\phi_i}{\sum_{k \in \mathcal{N}} \phi_k} \cdot C \\
\Leftrightarrow \qquad \delta &= \frac{\phi_i}{\sum_{k \in \mathcal{N}} \phi_k} \cdot C - \rho_{A_i}(\theta).
\end{aligned}
$$

□

BIBLIOGRAPHY

[Agr+99] Rajeev Agrawal, Rene L Cruz, Clayton Okino, and Rajendran Rajan. "Performance bounds for flow control protocols." In: *IEEE/ACM Transactions on Networking (ToN)* 7.3 (1999), pp. 310–323.

[AR96] Rajeev Agrawal and Rajendran Rajan. *Performance bounds for guaranteed and adaptive services*. IBM TJ Watson Research Center Yorktown Heights, NY, 1996.

[AMS82] David Anick, Debasis Mitra, and Man M Sondhi. "Stochastic theory of a data-handling system with multiple sources." In: *Bell System technical journal* 61.8 (1982), pp. 1871–1894.

[AKV08] Rita Giuliano Antonini, Yuriy Kozachenko, and Andrei Volodin. "Convergence of series of dependent φ-subGaussian random variables." In: *J. Math. Anal. Appl.* (2008).

[Bac+92] François Baccelli, Guy Cohen, Geert Jan Olsder, and Jean-Pierre Quadrat. *Synchronization and linearity: an algebra for discrete event systems*. John Wiley & Sons Ltd, 1992.

[Bas+75] Forest Baskett, K Mani Chandy, Richard R Muntz, and Fernando G Palacios. "Open, closed, and mixed networks of queues with different classes of customers." In: *Journal of the ACM (JACM)* 22.2 (1975), pp. 248–260.

[Bec16a] Michael A Beck. "Advances in Theory and Applicability of Stochastic Network Calculus." PhD thesis. TU Kaiserslautern, 2016.

[BS13] Michael A. Beck and Jens Schmitt. "The DISCO Stochastic Network Calculator Version 1.0 - When Waiting Comes to an End." In: *Proc. 7th International Conference on Performance Evaluation Methodologies and Tools (VALUETOOLS 2013)*. 2013.

[Bec16b] Michael Beck. "Towards the Analysis of Transient Phases with Stochastic Network Calculus." In: *IEEE 17th International Network Strategy and Planning Symposium (Networks'16)*. 2016.

[BH17] Michael Beck and Sebastian Henningsen. "Technical Report The Stochastic Network Calculator." In: *arXiv preprint arXiv:1707.07739* (2017).

[Bec21] Nico Becker. "Non-stationary service curves: model and estimation method with application to cellular sleep scheduling." PhD thesis. Hannover: Institutionelles Repositorium der Leibniz Universität Hannover, 2021.

[BF15] Nico Becker and Markus Fidler. "A non-stationary service curve model for performance analysis of transient phases." In: *2015 27th International Teletraffic Congress*. IEEE. 2015, pp. 116–124.

[BZ96] Jon CR Bennett and Hui Zhang. "WF2Q: worst-case fair weighted fair queueing." In: *Proceedings of IEEE INFOCOM'96. Conference on Computer Communications*. Vol. 1. 1996, pp. 120–128.

[Bon16] Steffen Bondorf. "Worst-Case Performance Analysis of Feed-Forward Networks – An Efficient and Accurate Network Calculus." PhD thesis. 2016, pp. IX, 157.

[BNS17a] Steffen Bondorf, Paul Nikolaus, and Jens B. Schmitt. "Quality and Cost of Deterministic Network Calculus – Design and Evaluation of an Accurate and Fast Analysis." In: *Proceedings of the ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS 2017)*. 2017.

[BNS17b] Steffen Bondorf, Paul Nikolaus, and Jens B. Schmitt. "Quality and Cost of Deterministic Network Calculus – Design and Evaluation of an Accurate and Fast Analysis." In: *Proceedings of the ACM on Measurement and Analysis of Computing Systems (POMACS)* 1.1 (2017), p. 34.

[BS16] Steffen Bondorf and Jens B. Schmitt. "Should Network Calculus Relocate? An Assessment of Current Algebraic and Optimization-based Analyses." In: *Proceedings of the International Conference on Quantitative Evaluation of Systems (QEST 2016)*. 2016.

[Boo+00] Robert Boorstyn, Almut Burchard, Joerg Liebeherr, and Chaiwat Oottamakorn. "Effective envelopes: Statistical bounds on multiplexed traffic in packet networks." In: *Proc. IEEE International Conference on Computer Communications (INFOCOM'00)*. Vol. 3. 2000, pp. 1223–1232.

[Bou14] Anne Bouillard. "Algorithms and efficiency of Network calculus." habilitation. Ecole Normale Supérieure (Paris), 2014.

[Bou21] Anne Bouillard. "Individual Service Curves for Bandwidth-Sharing Policies using Network Calculus." In: *IEEE Networking Letters* (2021).

[BBC18] Anne Bouillard, Marc Boyer, and Euriell Le Corronc. *Deterministic Network Calculus: From Theory to Practical Implementation*. John Wiley & Sons, 2018.

[Bou+08] Anne Bouillard, Bruno Gaujal, Sébastien Lagrange, and Eric Thierry. "Optimal routing for end-to-end guarantees using network calculus." In: *Performance Evaluation* 65.11-12 (2008), pp. 883–906.

[BJT09] Anne Bouillard, Laurent Jouhet, and Eric Thierry. *Service curves in Network Calculus: dos and don'ts*. Tech. rep. INRIA, France, 2009.

[BJT10] Anne Bouillard, Laurent Jouhet, and Eric Thierry. "Tight performance bounds in the worst-case analysis of feed-forward networks." In: *Proc. IEEE International Conference on Computer Communications (INFOCOM'10)*. 2010, pp. 1–9.

[BNS21] Anne Bouillard, Paul Nikolaus, and Jens B. Schmitt. "Fully Unleashing the Power of Paying Multiplexing Only Once in Stochastic Network Calculus." In: *CoRR* abs/2104.14215 (2021). arXiv: 2104.14215.

[BNS22] Anne Bouillard, Paul Nikolaus, and Jens Schmitt. "Unleashing the Power of Paying Multiplexing Only Once in Stochastic Network Calculus." In: *Proceedings of the ACM on Measurement and Analysis of Computing Systems (POMACS)* 6.2 (2022), p. 34.

[BN15] Anne Bouillard and Thomas Nowak. "Fast symbolic computation of the worst-case delay in tandem networks and applications." In: *Performance Evaluation* 91 (2015), pp. 270–285.

[BT16] Anne Bouillard and Éric Thierry. "Tight performance bounds in the worst-case analysis of feed-forward networks." In: *Discrete Event Dynamic Systems* 26.3 (2016), pp. 383–411.

[BNF12] Marc Boyer, Nicolas Navet, and Marc Fumey. "Experimental assessment of timing verification techniques for AFDX." In: *ERTS2012-Embedded Real Time Software and Systems*. 2012.

[Boy+18] Hugh Boyes, Bil Hallaq, Joe Cunningham, and Tim Watson. "The industrial internet of things (IIoT): An analysis framework." In: *Computers in Industry* 101 (2018), pp. 1–12.

[BL18] Almut Burchard and Jörg Liebeherr. "A General Per-Flow Service Curve for GPS." In: *2018 30th International Teletraffic Congress (ITC 30)*. Vol. 2. 2018, pp. 31–36.

[BLC07] Almut Burchard, Jörg Liebeherr, and Florin Ciucu. "On Θ (H log H) scaling of network delays." In: *Proc. IEEE International Conference on Computer Communications (INFOCOM'07)*. 2007.

[BLC11] Almut Burchard, Jörg Liebeherr, and Florin Ciucu. "On superlinear scaling of network delays." In: *IEEE/ACM Transactions on Networking* 19.4 (2011), pp. 1043–1056.

[BLP06] Almut Burchard, Jörg Liebeherr, and Stephen D Patek. "A min-plus calculus for end-to-end statistical service guarantees." In: *IEEE Transactions on Information Theory* 52.9 (2006), pp. 4105–4114.

[Bur64] PJ Burke. "The dependence of delays in tandem queues." In: *The Annals of Mathematical Statistics* 35.2 (1964), pp. 874–875.

[CAG18] Jaya Prakash Champati, Hussein Al-Zubaidy, and James Gross. "Transient Delay Bounds for Multi-Hop Wireless Networks." In: *CoRR* (2018).

[Cha94] Cheng-Shang Chang. "Stability, queue length, and delay of deterministic and stochastic queueing networks." In: *IEEE Transactions on Automatic Control* 39.5 (1994), pp. 913–931.

[Cha97] Cheng-Shang Chang. "A filtering theory for deterministic traffic regulation." In: *Proc. IEEE International Conference on Computer Communications (INFOCOM'97)*. Vol. 2. IEEE. 1997, pp. 436–443.

[Cha00] Cheng-Shang Chang. *Performance guarantees in communication networks*. London: Springer-Verlag, 2000.

[CC99] Cheng-Shang Chang and Rene L Cruz. "A time varying filtering theory for constrained traffic regulation and dynamic service guarantees." In: *Proc. IEEE International Conference on Computer Communications (INFOCOM'99)*. Vol. 1. 1999, pp. 63–70.

[Cha+02] Cheng-Shang Chang, Rene L Cruz, J-Y Le Boudec, and Patrick Thiran. "A min,+ system theory for constrained traffic regulation and dynamic service guarantees." In: *IEEE/ACM Transactions on Networking* 10.6 (2002), pp. 805–817.

[Che01] Wing-Sum Cheung. "Generalizations of Hölder's inequality." In: *International Journal of Mathematics and Mathematical Sciences* 26.1 (2001), pp. 7–10.

[Ciu07] Florin Ciucu. "Network calculus delay bounds in queueing networks with exact solutions." In: *Managing Traffic Performance in Converged Networks*. Springer, 2007, pp. 495–506.

[CBL05] Florin Ciucu, Almut Burchard, and Jörg Liebeherr. "A network service curve approach for the stochastic analysis of networks." In: *Proc. ACM International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS'05)*. Vol. 33. 1. 2005, pp. 279–290.

[CBL06] Florin Ciucu, Almut Burchard, and Jörg Liebeherr. "Scaling properties of statistical end-to-end bounds in the network calculus." In: *IEEE/ACM Transactions on Networking (ToN)* 14.SI (2006), pp. 2300–2312.

[CP18] Florin Ciucu and Felix Poloczek. "Two extensions of Kingman's GI/G/1 bound." In: *Proceedings of the ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS 2018)*. Vol. 2. 3. 2018, pp. 1–33.

[CPR19] Florin Ciucu, Felix Poloczek, and Amr Rizk. "Queue and loss distributions in finite-buffer queues." In: *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 3.2 (2019), pp. 1–29.

[CPS13] Florin Ciucu, Felix Poloczek, and Jens B. Schmitt. "Sharp Bounds in Stochastic Network Calculus." In: *CoRR* abs/1303.4114 (2013).

[CPS14] Florin Ciucu, Felix Poloczek, and Jens Schmitt. "Sharp Per-Flow Delay Bounds for Bursty Arrivals: The Case of FIFO, SP, and EDF Scheduling." In: *Proc. IEEE International Conference on Computer Communications (INFOCOM'14)*. Toronto, Canada, 2014.

[CPS16] Florin Ciucu, Felix Poloczek, and Jens Schmitt. "Stochastic Upper and Lower Bounds for General Markov Fluids." In: *Proc. IEEE International Teletraffic Congress (ITC 28'16)*. Vol. 1. 2016, pp. 184–192.

[CS12] Florin Ciucu and Jens Schmitt. "Perspectives on Network Calculus – No Free Lunch, But Still Good Value." In: *Proc. ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM'12)*. Helsinki, Finland, Aug. 2012, pp. 311–322.

[CW96] Costas Courcoubetis and Richard Weber. "Buffer overflow asymptotics for a buffer handling many traffic sources." In: *Journal of Applied Probability* 33 (1996), pp. 886–903.

[CB97] Mark E Crovella and Azer Bestavros. "Self-similarity in World Wide Web traffic: evidence and possible causes." In: *IEEE/ACM Transactions on Networking (ToN)* 5.6 (1997), pp. 835–846.

[Cru91a] Rene L Cruz. "A calculus for network delay, part I: Network elements in isolation." In: *IEEE Transactions on information theory* 37.1 (1991), pp. 114–131.

[Cru91b] Rene L Cruz. "A calculus for network delay, part II: Network analysis." In: *IEEE Transactions on information theory* 37.1 (1991), pp. 132–141.

[Cru95] Rene L Cruz. "Quality of service guarantees in virtual circuit switched networks." In: *IEEE Journal on Selected areas in Communications* 13.6 (1995), pp. 1048–1056.

[Cru96] Rene L Cruz. "Quality of service management in integrated services networks." In: *Proc. 1st Semi-Annual Research Review, CWC*. Vol. 1. 7. 1996, pp. 4–5.

[Cru98] Rene L Cruz. "SCED+: Efficient Management of Quality of Service Guarantees." In: *Proc. IEEE International Conference on Computer Communications (INFOCOM'98)*. 1998, pp. 625–634.

[CL93] Rene L Cruz and Hai-Ning Liu. "Single server queues with loss: A formulation." In: *Proc. CISS*. 1993, pp. 1–5.

[CO96] Rene L Cruz and Clayton Okino. "Service guarantees for window flow control." In: *Proc. 34th Allerton Conference on Communications, Control and Computing*. Vol. 34. 1996, pp. 10–21.

[DKS89] Alan Demers, Srinivasan Keshav, and Scott Shenker. "Analysis and simulation of a fair queueing algorithm." In: *ACM SIGCOMM Computer Communication Review* 19.4 (1989), pp. 1–12.

[Din+14] Weijun Ding, Jun Xu, Jim Dai, Yang Song, and Bill Lin. "Sprinklers: A randomized variable-size striping approach to reordering-free load-balanced switching." In: *Proc. 10th ACM International on Conference on emerging Networking Experiments and Technologies*. 2014, pp. 89–100.

[Don17] Fang Dong. "Copula theory and its applications in computer networks." PhD thesis. 2017.

[DWS15] Fang Dong, Kui Wu, and Venkatesh Srinivasan. "Copula analysis for statistical network calculus." In: *Proc. IEEE INFOCOM'15*. 2015, pp. 1535–1543.

[Doo53] Joseph Leo Doob. *Stochastic processes*. Vol. 10. New York Wiley, 1953.

[DR98] Devdatt Dubhashi and Desh Ranjan. "Balls and bins: A study in negative dependence." In: *Random Structures & Algorithms* 13.2 (1998), pp. 99–124.

[Erl09] Agner Krarup Erlang. “The theory of probabilities and telephone conversations.” In: *Nyt Tidsskrift for Matematik B* 20.33-39 (1909), p. 16.

[Erl17] Agner Krarup Erlang. “Solution of some problems in the theory of probabilities of significance in automatic telephone exchanges.” In: *Elektrotkeknikeren* 13 (1917), pp. 5–13.

[Fel+99] Anja Feldmann, Anna C Gilbert, Polly Huang, and Walter Willinger. “Dynamics of IP traffic: A study of the role of variability and the impact of control.” In: *Proc. ACM SIGCOMM'99*. Vol. 29. 4. 1999, pp. 301–313.

[Fet14] Gerhard Paul Fettweis. “The tactile internet: Applications and challenges.” In: *IEEE Vehicular Technology Magazine* 9.1 (2014), pp. 64–70.

[Fid03] Markus Fidler. “Extending the network calculus pay bursts only once principle to aggregate scheduling.” In: *International Workshop on Quality of Service in Multiservice IP Networks*. Springer. 2003, pp. 19–34.

[Fid05] Markus Fidler. “Server models for probabilistic network calculus.” In: *Informatik 2005. Informatik Live! Band 1* (2005).

[Fid06] Markus Fidler. “An end-to-end probabilistic network calculus with moment generating functions.” In: *Proc. IEEE IWQoS'06*. 2006, pp. 261–270.

[Fid10] Markus Fidler. “Survey of deterministic and stochastic service curve models in the network calculus.” In: *IEEE Communications surveys & tutorials* 12.1 (2010), pp. 59–86.

[FR15] Markus Fidler and Amr Rizk. “A guide to the stochastic network calculus.” In: *IEEE Communications Surveys & Tutorials* 17.1 (2015), pp. 92–105.

[FMN00] Nelson LS Fonseca, Gilberto S Mayor, and Cesar AV Neto. “On the equivalent bandwidth of self-similar sources.” In: *ACM Transactions on Modeling and Computer Simulation (TOMACS)* 10.2 (2000), pp. 104–124.

[FFG06] Fabrice Frances, Christian Fraboul, and Jérôme Grieu. “Using network calculus to optimize the AFDX network.” In: (2006).

[GMOB00] José R Gallardo, Dimitrios Makrakis, and Luis Orozco-Barbosa. “Use of α-stable self-similar stochastic processes for modeling traffic in broadband networks.” In: *Performance Evaluation* 40.1-3 (2000), pp. 71–98.

[GB19] Fabien Geyer and Steffen Bondorf. “DeepTMA: Predicting effective contention models for network calculus using graph neural networks.” In: *Proc. IEEE International Conference on Computer Communications (INFOCOM'19)*. IEEE. 2019, pp. 1009–1017.

[GFC12] Yashar Ghiassi-Farrokhfal and Florin Ciucu. “On the impact of finite buffers on per-flow delays in FIFO queues.” In: *2012 24th International Teletraffic Congress (ITC 24)*. IEEE. 2012, pp. 1–8.

[Gol+08] Nicos Gollan, Frank A. Zdarsky, Ivan Martinovic, and Jens Schmitt. "The DISCO Network Calculator." In: *14th GI/ITG Conference on Measurement, Modeling, and Evaluation of Computer and Communication Systems (MMB 2008)*. GI/ITG. 2008.

[GVC97] Pawan Goyal, Harrick M Vin, and Haichen Cheng. "Start-time fair queueing: A scheduling algorithm for integrated services packet switching networks." In: *IEEE/ACM Transactions on networking* 5.5 (1997), pp. 690–704.

[HM11] Gerald J Hahn and William Q Meeker. *Statistical intervals: a guide for practitioners*. Vol. 92. John Wiley & Sons, 2011.

[HB13] Mor Harchol-Balter. *Performance modeling and design of computer systems: queueing theory in action*. Cambridge University Press, 2013.

[HJ61] Robert Hooke and Terry A Jeeves. ""Direct Search"Solution of Numerical and Statistical Problems." In: *Journal of the ACM (JACM)* 8.2 (1961), pp. 212–229.

[Jac57] James R Jackson. "Networks of waiting lines." In: *Operations research* 5.4 (1957), pp. 518–521.

[Jac63] James R Jackson. "Jobshop-like queueing systems." In: *Management science* 10.1 (1963), pp. 131–142.

[Jaf+10] Fahimeh Jafari, Zhonghai Lu, Axel Jantsch, and Mohammad Hossein Yaghmaee. "Buffer optimization in network-on-chip through flow regulation." In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 29.12 (2010), pp. 1973–1986.

[Jas+02] Jürgen Jasperneite, Peter Neumann, Michael Theis, and Kym Watson. "Deterministic real-time communication with switched Ethernet." In: *4th IEEE International Workshop on Factory Communication Systems*. IEEE. 2002, pp. 11–18.

[JL08] Yuming Jiang and Yong Liu. *Stochastic network calculus*. Vol. 1. Springer, 2008.

[JDP83] Kumar Joag-Dev and Frank Proschan. "Negative association of random variables with applications." In: *The Annals of Statistics* 11.1 (1983), pp. 286–295.

[KH01] Anestis Karasaridis and Dimitrios Hatzinakos. "Network heavy traffic modeling using/spl alpha/-stable self-similar processes." In: *IEEE Transactions on Communications* 49.7 (2001), pp. 1203–1214.

[KSC91] Manolis Katevenis, Stefanos Sidiropoulos, and Costas Courcoubetis. "Weighted round-robin cell multiplexing in a general-purpose ATM switch chip." In: *IEEE Journal on selected Areas in Communications* 9.8 (1991), pp. 1265–1279.

[Kel75] Frank P Kelly. "Networks of queues with customers of different types." In: *Journal of applied probability* 12.3 (1975), pp. 542–554.

[Kel76] Frank P Kelly. "Networks of queues." In: *Advances in Applied Probability* 8.2 (1976), pp. 416–432.

[Kel91] Frank Patrick Kelly. "Effective bandwidths at multi-class queues." In: *Queueing systems* 9.1-2 (1991), pp. 5–15.

[Kel96] Frank Patrick Kelly. "Notes on Effective Bandwidths." In: *Stochastic Networks: Theory and Applications*. Ed. by Frank Patrick Kelly, Stan Zachary, and Ilze Ziedins. Vol. 4. Royal Statistical Society Lecture Notes Series. Oxford University Press: Oxford, 1996, pp. 141–168.

[KGV83] Scott Kirkpatrick, C Daniel Gelatt, and Mario P Vecchi. "Optimization by simulated annealing." In: *science* 220.4598 (1983), pp. 671–680.

[Kle75] Leonard Kleinrock. *Queueing systems*. Vol. 66. wiley New York, 1975.

[KMT11] Hisashi Kobayashi, Brian L Mark, and William Turin. *Probability, random processes, and statistical analysis: applications to communications, signal processing, queueing theory and mathematical finance*. Cambridge University Press, 2011.

[KAT06] Anis Koubaa, Mario Alves, and Eduardo Tovar. "Modeling and worst-case dimensioning of cluster-tree wireless sensor networks." In: *2006 27th IEEE International Real-Time Systems Symposium (RTSS'06)*. IEEE. 2006, pp. 412–421.

[Kur92] Jim Kurose. "On computing per-session performance bounds in high-speed multi-hop computer networks." In: *Proc. ACM SIGMETRICS '92*. ACM, 1992, pp. 128–139.

[Le 98] Jean-Yves Le Boudec. "Application of network calculus to guaranteed service networks." In: *IEEE Transactions on Information theory* 44.3 (1998), pp. 1087–1096.

[LT01] Jean-Yves Le Boudec and Patrick Thiran. *Network calculus: a theory of deterministic queuing systems for the internet*. New York: Springer-Verlag, 2001.

[LH08] Long Le and Ekram Hossain. "Tandem queue models with applications to QoS routing in multihop wireless networks." In: *IEEE Transactions on Mobile Computing* 7.8 (2008), pp. 1025–1040.

[Leh66] Erich Leo Lehmann. "Some concepts of dependence." In: *The Annals of Mathematical Statistics* (1966), pp. 1137–1153.

[Lei85] Charles E Leiserson. "Fat-trees: universal networks for hardware-efficient supercomputing." In: *IEEE transactions on Computers* 100.10 (1985), pp. 892–901.

[Lel+94] Will E Leland, Murad S Taqqu, Walter Willinger, and Daniel V Wilson. "On the self-similar nature of Ethernet traffic." In: *IEEE/ACM Transactions on Networking (ToN)* 2.1 (1994), pp. 1–15.

[LMS07] Luciano Lenzini, Enzo Mingozzi, and Giovanni Stea. "End-to-end delay bounds in FIFO-multiplexing tandems." In: *Proc. Second International Conference on Performance Evaluation Methodologies and Tools (VALUETOOLS'07), Pisa, Italy*. 2007, pp. 1–10.

[LBL07] Chengzhi Li, Almut Burchard, and Jörg Liebeherr. "A network calculus with effective bandwidth." In: *IEEE/ACM Transactions on Networking* 15.6 (2007), pp. 1442–1453.

[Lie17] Jörg Liebeherr. "Duality of the max-plus and min-plus network calculus." In: *Foundations and Trends in Networking* 11.3-4 (2017), pp. 139–282.

[LBC12] Jörg Liebeherr, Almut Burchard, and Florin Ciucu. "Delay bounds in communication networks with heavy-tailed and self-similar traffic." In: *IEEE Transactions on Information Theory* 58.2 (2012), pp. 1010–1024.

[LGFB11] Jörg Liebeherr, Yashar Ghiassi-Farrokhfal, and Almut Burchard. "On the impact of link scheduling on end-to-end delays in large networks." In: *IEEE Journal on Selected Areas in Communications* 29.5 (2011), pp. 1009–1020.

[Lin52] David V Lindley. "The theory of queues with a single server." In: *Mathematical Proceedings of the Cambridge Philosophical Society*. Cambridge University Press. 1952, pp. 277–289.

[Lin02] Torgny Lindvall. *Lectures on the coupling method*. Courier Corporation, 2002.

[Liu93] Haining Liu. *Buffer size and packet loss in a tandem queueing network*. University of California at San Diego, 1993.

[LGOBM05] Miguel Lopez-Guerrero, Luis Orozco-Barbosa, and Dimitrios Makrakis. "Probabilistic envelope processes for α-stable self-similar traffic models and their application to resource provisioning." In: *Performance Evaluation* 61.2-3 (2005), pp. 257–279.

[MS97] Gilberto Mayor and John Silvester. "Time scale analysis of an ATM queueing system with long-range dependent traffic." In: *Proc. IEEE International Conference on Computer Communications (INFOCOM'97)*. Vol. 1. IEEE. 1997, pp. 205–212.

[MSZ02] Marco Mellia, Ion Stoica, and Hui Zhang. "TCP model for short lived flows." In: *IEEE communications letters* 6.2 (2002), pp. 85–87.

[NM65] John A Nelder and Roger Mead. "A simplex method for function minimization." In: *The computer journal* 7.4 (1965), pp. 308–313.

[Nel95] Randolph Nelson. *Probability, stochastic processes, and queueing theory: the mathematics of computer performance modeling*. Springer, 1995.

[NS17] Paul Nikolaus and Jens Schmitt. "On Per-Flow Delay Bounds in Tandem Queues under (In)Dependent Arrivals." In: *Proc. IFIP Networking 2017 Conference (NETWORKING'17)*. Stockholm, Sweden, 2017.

[NS18] Paul Nikolaus and Jens Schmitt. "Improving Output Bounds in the Stochastic Network Calculus Using Lyapunov's Inequality." In: *Proc. IFIP Networking 2018 Conference (NETWORKING'18)*. Zurich, Switzerland, 2018.

[NS20a] Paul Nikolaus and Jens Schmitt. "Improving Delay Bounds in the Stochastic Network Calculus by Using less Stochastic Inequalities." In: *Proc. 13th EAI International Conference on Performance Evaluation Methodologies and Tools (VALUETOOLS 2020)*. Tsukuba, Japan, 2020.

[NS20b] Paul Nikolaus and Jens Schmitt. "On the Stochastic End-to-End Delay Analysis in Sink Trees Under Independent and Dependent Arrivals." In: *Proc. Conference on Measurement, Modelling and Evaluation of Computing Systems (MMB'20)*. Springer. 2020.

[NSC19a] Paul Nikolaus, Jens Schmitt, and Florin Ciucu. *Dealing with Dependence in Stochastic Network Calculus – Using Independence as a Bound*. Tech. rep. TU Kaiserslautern, Germany, May 2019.

[NSC19b] Paul Nikolaus, Jens Schmitt, and Florin Ciucu. "Dealing with Dependence in Stochastic Network Calculus – Using Independence as a Bound." In: *Proc. International Conference on Analytical and Stochastic Modeling Techniques and Applications (ASMTA'19)*. Springer. Moscow, Russia, 2019.

[NSS19] Paul Nikolaus, Jens Schmitt, and Malte Schütze. "h-Mitigators: Improving your stochastic network calculus output bounds." In: *Computer Communications* 144 (2019), pp. 188–197.

[Nor94] Ilkka Norros. "A storage model with self-similar input." In: *Springer Queueing Systems* 16.3 (1994), pp. 387–396.

[Nor95] Ilkka Norros. "On the use of fractional Brownian motion in the theory of connectionless networks." In: *IEEE Journal on selected Areas in Communications* 13.6 (1995), pp. 953–962.

[Oli07] Travis E Oliphant. "Python for scientific computing." In: *Computing in science & engineering* 9.3 (2007), pp. 10–20.

[OMP06] Chaiwat Oottamakorn, Shiwen Mao, and Shivendra S Panwar. "On generalized processor sharing with regulated multimedia traffic flows." In: *IEEE Transactions on Multimedia* 8.6 (2006), pp. 1209–1218.

[PG93] A. K. Parekh and R. G. Gallager. "A generalized processor sharing approach to flow control in integrated services networks: the single-node case." In: *IEEE /ACM Transactions on Networking* 1.3 (1993), pp. 344–357.

[PG94] Abhay K Parekh and Robert G Gallager. "A generalized processor sharing approach to flow control in integrated services networks: the multiple node case." In: *IEEE /ACM transactions on Networking* 2.2 (1994), pp. 137–150.

[PF95] Vern Paxson and Sally Floyd. "Wide area traffic: the failure of Poisson modeling." In: *IEEE/ACM Transactions on Networking (ToN)* 3.3 (1995), pp. 226–244.

[PC14] Felix Poloczek and Florin Ciucu. "Scheduling analysis with martingales." In: *Elsevier Performance Evaluation* 79 (2014), pp. 56–72.

[PC15] Felix Poloczek and Florin Ciucu. "Service-martingales: Theory and applications to the delay analysis of random access protocols." In: *Proc. IEEE Conference on Computer Communications (INFOCOM'15)*. 2015, pp. 945–953.

[Qia+16] Zhiliang Qian, Paul Bogdan, Chi-Ying Tsui, and Radu Marculescu. "Performance evaluation of NoC-based multicore systems: From traffic analysis to noc latency modeling." In: *ACM Transactions on Design Automation of Electronic Systems (TODAES)* 21.3 (2016), p. 52.

[Raj+10] Ragunathan Raj Rajkumar, Insup Lee, Lui Sha, and John Stankovic. "Cyber-physical systems: the next computing revolution." In: *Proc. ACM Design Automation Conference'10*. 2010, pp. 731–736.

[Riz13] Amr Rizk. "Non-asymptotic Performance Evaluation and sampling-based Parameter Estimation for Communication Networks with long Memory traffic." PhD thesis. Hannover: Gottfried Wilhelm Leibniz Universität Hannover, 2013.

[RF11] Amr Rizk and Markus Fidler. "Leveraging statistical multiplexing gains in single-and multi-hop networks." In: *Proc. IEEE IWQoS '11*. 2011, pp. 1–9.

[RF12a] Amr Rizk and Markus Fidler. "Non-asymptotic end-to-end performance bounds for networks with long range dependent fBm cross traffic." In: *Computer Networks* 56.1 (2012), pp. 127–141.

[RF12b] Amr Rizk and Markus Fidler. "On multiplexing models for independent traffic flows in single-and multi-node networks." In: *IEEE Transactions on Network and Service Management* 10.1 (2012), pp. 15–28.

[RVC01] E. Rosen, A. Viswanathan, and R. Callon. *Multiprotocol Label Switching Architecture*. RFC 3031. RFC Editor, 2001.

[Ros96] Sheldon M Ross. *Stochastic processes*. Vol. 2. Wiley New York, 1996.

[Ros10] Sheldon M Ross. *Introduction to probability models*. Academic press, 2010.

[SCP95] Hanrijanto Sariowan, Rene L Cruz, and George C Polyzos. "Scheduling for quality of service guarantees via service curves." In: *Proceedings of Fourth International Conference on Computer Communications and Networks-IC3N'95*. 1995, pp. 512–520.

[SBS18] Alexander Scheffler, Steffen Bondorf, and Jens B. Schmitt. "Worst-Case Performance Analysis with the Disco Deterministic Network Calculator." In: *Proceedings of the 22nd IEEE International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD 2018)*. 2018.

[SZF08] Jens B Schmitt, Frank A Zdarsky, and Markus Fidler. "Delay Bounds under Arbitrary Multiplexing: When Network Calculus Leaves You in the Lurch ..." In: *Proc. IEEE International Conference on Computer Communications (INFOCOM'08)*. Phoenix, AZ, USA, Apr. 2008.

[Sch03] Jens Schmitt. "On average and worst case behaviour in non-preemptive priority queueing." In: *Simulation Series* 35.4 (2003), pp. 197–204.

[SBP17] Jens Schmitt, Steffen Bondorf, and Wint Yi Poe. "The Sensor Network Calculus as Key to the Design of Wireless Sensor Networks with Predictable Performance." In: *Journal of Sensor and Actuator Networks* 6.3 (2017).

[Sch+11] Jens Schmitt, Nicos Gollan, Steffen Bondorf, and Ivan Martinovic. "Pay Bursts Only Once Holds for (Some) Non-FIFO Systems." In: *The 30th IEEE International Conference on Computer Communications (INFOCOM 2011)*. Shanghai, China, 2011.

[SGM08] Jens Schmitt, Nicos Gollan, and Ivan Martinovic. *End-to-End Worst-Case Analysis of Non-FIFO Systems*. Technical Report 370/08. TU Kaiserslautern, Germany, 2008.

[SZ06] Jens Schmitt and Frank A. Zdarsky. "The DISCO Network Calculator - A Toolbox for Worst Case Analysis." In: *Proc. First International Conference on Performance Evaluation Methodologies and Tools (VALUETOOLS'06), Pisa, Italy*. 2006.

[SZM06] Jens Schmitt, Frank A. Zdarsky, and Ivan Martinovic. *Performance Bounds in Feed-Forward Networks under Blind Multiplexing*. Technical Report 349/06. University of Kaiserslautern, Germany, 2006.

[SZM08] Jens Schmitt, Frank A Zdarsky, and Ivan Martinovic. "Improving Performance Bounds in Feed-Forward Networks by Paying Multiplexing Only Once." In: *Proc. GI/ITG Conference on Measurement, Modeling, and Evaluation of Computer and Communication Systems (MMB'08)*. 2008, pp. 1–15.

[SV95] Madhavapeddi Shreedhar and George Varghese. "Efficient fair queueing using deficit round robin." In: *Proc. conference on Applications, technologies, architectures, and protocols for computer communication*. 1995, pp. 231–242.

[Sin+14] Ankit Singla, Balakrishnan Chandrasekaran, Philip Brighten Godfrey, and Bruce Maggs. "The internet at the speed of light." In: *Proc. ACM Workshop on Hot Topics in Networks'14*. HotNets-XIII. 2014, pp. 1–7.

[Soh92] Khosrow Sohraby. "On the asymptotic behavior of heterogeneous statistical multiplexer with applications." In: *Proc. IEEE International Conference on Computer Communications (INFOCOM'92)*. 1992, pp. 839–847.

[SS00] David Starobinski and Moshe Sidi. "Stochastically bounded burstiness for communication networks." In: *Proc. IEEE International Conference on Computer Communications (INFOCOM'99)*. Vol. 1. 2000, pp. 36–42.

[SP97] Rainer Storn and Kenneth Price. "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces." In: *Journal of global optimization* 11.4 (1997), pp. 341–359.

[Sun11] Soo Hak Sung. "On the exponential inequalities for negatively dependent random variables." In: *J. Math. Anal. Appl.* 381.2 (2011), pp. 538–545.

[Sup+10] Urban Suppiger, Simon Perathoner, Kai Lampka, and Lothar Thiele. *Modular performance analysis of large-scale distributed embedded systems: An industrial case study*. Tech. rep. ETH Zurich, 2010.

[Tho00] Hermann Thorisson. *Coupling, Stationarity and Regeneration*. Springer, 2000.

[TS96] Constantino Tsallis and Daniel A Stariolo. "Generalized simulated annealing." In: *Physica A: Statistical Mechanics and its Applications* 233.1-2 (1996), pp. 395–406.

[Vir+20] Pauli Virtanen, Ralf Gommers, Travis E Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, et al. "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python." In: *Nature Methods* 17 (2020), pp. 261–272.

[WD97] David J Wales and Jonathan PK Doye. "Global optimization by basin-hopping and the lowest energy structures of Lennard-Jones clusters containing up to 110 atoms." In: *The Journal of Physical Chemistry A* 101.28 (1997), pp. 5111–5116.

[Wan+18] Hao Wang, Haoyun Shen, Philipp Wieder, and Ramin Yahyapour. "A Data Center Interconnects Calculus." In: *2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS)*. IEEE. 2018, pp. 1–10.

[WADX15] Andrew Whitmore, Anurag Agarwal, and Li Da Xu. "The Internet of Things–A survey of topics and trends." In: *Springer Information Systems Frontiers* 17.2 (2015), pp. 261–274.

[Wil+97] Walter Willinger, Murad S Taqqu, Robert Sherman, and Daniel V Wilson. "Self-similarity through high-variability: statistical analysis of Ethernet LAN traffic at the source level." In: *IEEE/ACM Transactions on Networking (ToN)* 5.1 (1997), pp. 71–86.

[YS93] Opher Yaron and Moshe Sidi. "Performance and stability of communication networks via robust exponential bounds." In: *IEEE/ACM Transactions on Networking* 1.3 (1993), pp. 372–385.

[Yin+02] Qinghe Yin, Yuming Jiang, Shengming Jiang, and Peng Yong Kong. "Analysis on generalized stochastically bounded bursty traffic for communication networks." In: *Proc. 27th Annual IEEE Conference on Local Computer Networks*. 2002, pp. 141–149.

[ZTK95] Zhi-Li Zhang, Don Towsley, and Jim Kurose. "Statistical analysis of the generalized processor sharing scheduling discipline." In: *IEEE Journal on Selected Areas in Communications* 13.6 (1995), pp. 1071–1080.

[ZBHB16] Timothy Zhu, Danel S Berger, and Mor Harchol-Balter. "SNC-Meister: Admitting More Tenants with Tail Latency SLOs." In: *Proc. ACM Symposium on Cloud Computing (SoCC'16)*. 2016.

[ZAV02] Konstantinos G Zografos, Konstantinos N Androutsopoulos, and George M Vasilakis. "A real-time decision support system for roadway network incident response logistics." In: *Transportation Research Part C: Emerging Technologies* 10.1 (2002), pp. 1–18.

INDEX

Paul Nikolaus

Curriculum Vitae

Education

Academic Qualifications

- 2017–2022 **PhD in Computer Science**, *Technische Universität Kaiserslautern*, Distributed Computer Systems (DISCO) Lab
- 2013–2016 **M.Sc. Business Mathematics**, *Technische Universität Kaiserslautern*, major: Financial Mathematics
- 2009–2013 **B.Sc. Mathematics**, *Technische Universität Kaiserslautern*, major: Probability and Optimization, elective: Economics

Research Experience

- 2018–2020 **Cooperation Project**, *Huawei Technologies Co.,Ltd.*, IP Research Department
- Oct 2018 **Research Visit**, *University of Warwick*, Department of Computer Science

Professional Experience

- 2015 **Intern**, *KfW-IPEX*, Frankfurt am Main, X4a4 - Risk Instruments and Risk Control
- 2015 **Student Research Assistant**, *Technische Universität Kaiserslautern*, Distributed Computer Systems (DISCO) Lab
- 2014 – 2015 **Intern**, *Deutsche Bank AG*, Frankfurt am Main, Credit Risk Management Private & Business Clients (CRM PBC)
- 2014 **Student Research Assistant**, *Technische Universität Kaiserslautern*, Distributed Computer Systems (DISCO) Lab

Honors & Awards

- 2018 **Best Presentation Award**, *WoNeCa '18*
- 2017 **Student Travel Grant**, *ACM SIGMETRICS '17*

LIST OF PUBLICATIONS

CONFERENCE PROCEEDINGS

1. Steffen Bondorf, Paul Nikolaus, and Jens B. Schmitt. "Quality and Cost of Deterministic Network Calculus – Design and Evaluation of an Accurate and Fast Analysis." In: Proceedings of the ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS 2017). 2017.

2. Steffen Bondorf, Paul Nikolaus, and Jens B. Schmitt. "Catching Corner Cases in Network Calculus – Flow Segregation Can Improve Accuracy." In: Proceedings of 19th International GI/ITG Conference on Measurement, Modelling and Evaluation of Computing Systems (MMB). 2018.

3. Paul Nikolaus, Sebastian Henningsen, Michael A. Beck, and Jens Schmitt. "Integrating Fractional Brownian Motion Arrivals into the Statistical Network Calculus." In: 2018 30th International Teletraffic Congress (ITC 30). Vol. 2. Vienna, Austria, 2018, pp. 37–42.

4. Paul Nikolaus and Jens Schmitt. "On Per-Flow Delay Bounds in Tandem Queues under (In)Dependent Arrivals." In: Proc. IFIP Networking 2017 Conference (NETWORKING'17). Stockholm, Sweden, 2017.

5. Paul Nikolaus and Jens Schmitt. "Improving Output Bounds in the Stochastic Network Calculus Using Lyapunov's Inequality." In: Proc. IFIP Networking 2018 Conference (NETWORKING'18). Zurich, Switzerland: IEEE, May 2018.

6. Paul Nikolaus and Jens Schmitt. "Improving Delay Bounds in the Stochastic Network Calculus by Using less Stochastic Inequalities." In: Proc. 13th EAI International Conference on Performance Evaluation Methodologies and Tools (VALUETOOLS 2020). Tsukuba, Japan, 2020.

7. Paul Nikolaus and Jens Schmitt. "On the Stochastic End-to-End Delay Analysis in Sink Trees Under Independent and Dependent Arrivals." In: Proc. Conference on Measurement, Modelling and Evaluation of Computing Systems (MMB'20). Springer. 2020.

8. Paul Nikolaus, Jens Schmitt, and Florin Ciucu. "Dealing with Dependence in Stochastic Network Calculus – Using Independence as a Bound." In: Proc. International Conference on Analytical and Stochastic Modeling Techniques and Applications (ASMTA'19). Springer. Moscow, Russia, 2019.

journal articles

1. Steffen Bondorf, Paul Nikolaus, and Jens B. Schmitt. "Quality and Cost of Deterministic Network Calculus – Design and Evaluation of an Accurate and Fast Analysis." In: Proceedings of the ACM on Measurement and Analysis of Computing Systems (POMACS) 1.1 (2017), p. 34.
2. Anne Bouillard, Paul Nikolaus, and Jens Schmitt. "Unleashing the Power of Paying Multiplexing Only Once in Stochastic Network Calculus." In: Proceedings of the ACM on Measurement and Analysis of Computing Systems (POMACS) 6.2 (2022), p. 34.
3. Paul Nikolaus, Jens Schmitt, and Malte Schütze. "h-Mitigators: Improving your stochastic network calculus output bounds." In: Computer Communications 144 (2019), pp. 188–197.

technical reports

1. Steffen Bondorf, Paul Nikolaus, and Jens B. Schmitt. "Delay Bounds in Feed-Forward Networks - A Fast and Accurate Network Calculus Solution." In: CoRR abs/1603.02094 (2016). arXiv: 1603.02094.
2. Anne Bouillard, Paul Nikolaus, and Jens B. Schmitt. "Unleashing the Power of Paying Multiplexing Only Once in Stochastic Network Calculus." In: CoRR abs/2104.14215 (2021). arXiv: 2104.14215.
3. Paul Nikolaus, Jens B. Schmitt, and Florin Ciucu. Dealing with Dependence in Stochastic Network Calculus – Using Independence as a Bound. Tech. rep. TU Kaiserslautern, Germany, May 2019.

COLOPHON

This document was typeset using the typographical look-and-feel `classicthesis` developed by André Miede and Ivo Pletikosić. The style was inspired by Robert Bringhurst's seminal book on typography *"The Elements of Typographic Style"*. `classicthesis` is available for both LaTeX and LyX:

https://bitbucket.org/amiede/classicthesis/