h-Mitigators: Improving Your Stochastic Network Calculus Output Bounds

Paul Nikolaus[1], Jens Schmitt, Malte Schütze

Distributed Computer Systems (DISCO) Lab
TU Kaiserslautern, Germany

Preprint submitted to Elsevier Computer Communications

Abstract

Giving tight estimates for output bounds is key to an accurate network analysis using the stochastic network calculus (SNC) framework. In order to upper bound the delay for a flow of interest in the network, one typically has to calculate output bounds of cross-traffic flows several times. Thus, an improvement in the output bound calculation pays off considerably. In this paper, we propose a new output bound calculation in the SNC framework by making use of Jensen's inequality. In consists of inserting a convex function h into the bound, the so-called h-mitigator. We prove the bound's validity and also show that, by choosing h as the power function, that it is always at least as accurate as the state-of-the-art method. Numerical evaluations demonstrate that even in small heterogeneous two-server topologies, our approach can improve a delay bound's violation probability by a factor of over 135. For a set of randomly generated parameters, the bound is still decreased by a factor of 1.23 on average. Furthermore, our approach can be easily integrated in existing end-to-end analyses. Last but not least, we investigated another variant for h, the exponential function and showed numerically that this approach is mostly disadvantageous.

Keywords: Stochastic Network Calculus, Output Bound, Jensen Inequality

1. Introduction

1.1. Motivation

Providing delay bounds in packet-switched networks is a timeless challenge with recent sample applications as, e.g., Internet at the speed of light [1], Tactile Internet [2], Internet of Things [3], or the envisioned cyber-physical systems [4], which often face real-time requirements.

The network calculus (NC) holds the promise to enable a tight end-to-end delay analysis in such advanced applications building on a modular and uniform mathematical framework based on min-plus algebra [5]. Starting from the 1990s with two papers by Cruz [6, 7], NC demonstrated its benefits providing tight bounds for deterministic worst-case end-to-end delay bounds. In the following, the deterministic network calculus (DNC) was further elaborated and mathematically cast into a min-plus algebra setting [8, 9]. More recently, NC was generalized into a stochastic setting providing probabilistic worst-case bounds: The stochastic network calculus (SNC) framework [8, 10, 11, 12, 13]. SNC's main features can be summarized as providing a very general scheduling abstraction (the service curve) and the ability to enable system-wide end-to-end analysis (the concatenation theorem) [13].

SNC results can be categorized into different branches such as tail-bound based [10, 14, 12], moment generating functions (MGF) based [8, 11], and martingale based [15] approaches. Recent work evidences its applicability to modern problems, e.g., in the analysis of parallel systems (using the fork-join pattern) or multi-tenancy [16, 17, 18].

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 flow of interest (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 [19], 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 paper. 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

Email addresses: nikolaus@cs.uni-kl.de (Paul Nikolaus), jschmitt@cs.uni-kl.de (Jens Schmitt), m_schuetze13@cs.uni-kl.de (Malte Schütze)

[1] This paper was presented in part at IFIP Networking 2018, Zurich, CH, February 2018.

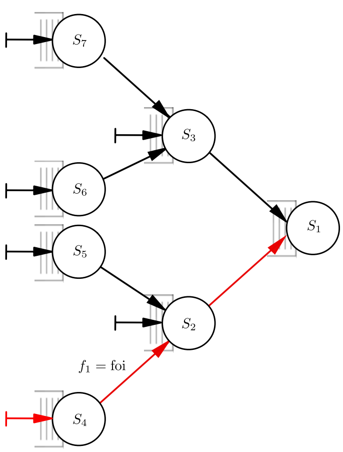


Figure 1: Full binary sink tree with seven nodes.

transmitted to the sink; let the foi be starting from one of the leaf nodes (see also Figure 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.

Yet, how can we improve upon the standard SNC output bound calculation? The tail bound and MGF SNC analyses have the application of the so-called Union bound or Boole's inequality in common. In a series of publications, [15, 20, 21, 22], the authors emphasized its poor performance and suggested an appealing martingale-based approach. It provides tight single hop lower and upper bounds on the delay for different scheduling disciplines. Yet, to the best of our knowledge, so far there is no concatenation result in the martingale-based SNC and thus step 2) from above cannot be performed and, thus, an elegant end-to-end analysis remains elusive. Hence, we decided to remain within the standard SNC framework and, yet, try to counteract the inherent problems of the Union bound.

1.2. Main Contribution

In this paper, 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. We try different functions h for Jensen's inequality and show that its choice is key to a tight and computationally fast analysis. 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 bound already by a factor of up to 33.

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.

1.3. Outline

The rest of the paper is structured as follows: In Section 2, we introduce the necessary notations for SNC and its main results as we need them in this paper. In Section 3, we present our new output bound calculation and prove its validity. A numerical evaluation is given in Section 4: 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 5, we prove that Jensen's inequality with the power function cannot be applied directly to delay bounds. We investigate alternatives by choosing h as the exponential function in Section 6. Section 7 concludes the paper.

2. SNC Background and Notation

In this section, we introduce some of the basic terms and concepts in SNC.

We use the MGF-based SNC in order to calculate per-flow delay bounds. To be precise, we bound the probability that the delay exceeds a given value, typically denoted by T. The connection between probability bounds and MGFs is established by Chernoff's bound [23]

$$\mathrm{P}(X > a) \leq e^{-\theta a}\,\mathrm{E}\left[e^{\theta X}\right], \quad \theta > 0, \tag{1}$$

with $\mathrm{E}\left[e^{\theta X}\right]$ as the moment-generating function (MGF) of a random variable X. We define an *arrival flow* by the stochastic process A with discrete time space $\mathbb{N}$ and continuous state space $\mathbb{R}_0^+$ as $A(s,t) := \sum_{i=s+1}^{t} a(i)$, with $a(i)$ as the traffic increment process in time slot i. Network calculus provides an elegant system-theoretic analysis by employing min-plus algebra:

Definition 1 (Convolution in Min-Plus Algebra [5]). The min-plus (de-)convolution of real-valued, bivariate functions $x(s,t)$ and $y(s,t)$ is defined as

$$\begin{aligned} (x \otimes y)\,(s,t) &:= \min_{s \leq i \leq t} \{x(s,i) + y(i,t)\}\,, \\ (x \oslash y)\,(s,t) &:= \max_{0 \leq i \leq s} \{x(i,t) - y(i,s)\}\,. \end{aligned} \tag{2}$$

The characteristics of the service process are captured by the notion of a dynamic S-server.

Definition 2 (Dynamic S-Server [8]). Assume a service element has an arrival flow A as its input and the respective output is denoted by A'. Let $S(s,t)$, $0 \leq s \leq t$, be a stochastic process that is nonnegative and increasing in t. The service element is a *dynamic S-server* iff for all $t \geq 0$ it holds that:

$$A'(0,t) \geq (A \otimes S)\,(0,t) \overset{(2)}{=} \min_{0 \leq i \leq t} \{A(0,i) + S(i,t)\}\,.$$

The analysis in this paper is based on a per-flow perspective. That is, we consider a certain flow, the so-called *flow of interest* (foi). Throughout this paper, for the sake of simplicity, we assume the servers' scheduling to be arbitrary multiplexing [24]. That is, if flow f_2 is prioritized over flow f_1, the leftover service at a dynamic S-server for the corresponding arrival A_1 is $S_{\mathrm{l.o.}}(s,t) = [S(s,t) - A_2(s,t)]^+$. Therefore, we require the server to be work-conserving.

Definition 3 (Work-Conserving Server [8][11]). For any $t \geq 0$ let $\tau := \sup\{s \in [0,t] : A'(0,s) = A(0,s)\}$ be the beginning of the last backlogged period before t. Assume again the service $S(s,t)$, $0 \leq s \leq t$, to be a stochastic process that is nonnegative and increasing in t with $S(\tau,\tau) = 0$. A server is said to be *work-conserving* if for any fixed sample path the server is not idle and uses the entire available service, i.e., $A'(0,t) = A'(0,\tau) + S(\tau,t)$.

In the following definition, we introduce (σ,ρ)-constraints [8] as they enable us to give bounds under stability conditions.

Definition 4 ((σ,ρ)-Bound [8]). An arrival flow is (σ_A,ρ_A)-bounded for some $\theta > 0$, if its MGF exists and for all $0 \leq s \leq t$

$$\mathrm{E}\left[e^{\theta A(s,t)}\right] \leq e^{\theta(\rho_A(\theta)(t-s)+\sigma_A(\theta))}.$$

A dynamic S-server is (σ_S,ρ_S)-bounded for some $\theta > 0$, if its MGF exists and for all $0 \leq s \leq t$

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

Definition 5 (Virtual Delay). The *virtual delay* at time $t \geq 0$ is defined as

$$d(t) := \inf\{s \geq 0 : A(0,t) \leq A'(0,t+s)\}.$$

It can briefly be described as the time it takes for the cumulated departures to "catch up with" the cumulated arrivals.

Theorem 6 (Output and Delay Bound). *[8][11] Consider an arrival process $A(s,t)$ with dynamic S-server $S(s,t)$.*

The departure process A' is upper bounded for any $0 \leq s \leq t$ according to

$$A'(s,t) \leq (A \oslash S)\ (s,t).$$

The delay at $t \geq 0$ is upper bounded by

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

We focus on the analogue of Theorem 6 for moment generating functions:

Theorem 7 (Output and Delay MGF-Bound [11][25]). *For the assumptions as in Theorem 6, we obtain:*

The MGF of the departure process A' is upper bounded for any $0 \leq s \leq t$ according to

$$\mathrm{E}\left[e^{\theta A'(s,t)}\right] \leq \mathrm{E}\left[e^{\theta(A \oslash S)\ (s,t)}\right]. \tag{3}$$

The violation probability of a given stochastic delay bound $T \geq 0$ at time $t \geq 0$ is bounded by

$$\mathrm{P}(d(t) > T) \leq \mathrm{E}\left[e^{\theta(A \oslash S)\ (t+T,t)}\right]. \tag{4}$$

3. 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.

3.1. Insertion of Jensen's Inequality

The standard approach to bound the output-MGF (3) is as follows

$$\begin{aligned}\mathrm{E}\left[e^{\theta A'(s,t)}\right] &\overset{(3)}{\leq} \mathrm{E}\left[e^{\theta(A \oslash S)\ (s,t)}\right] \\ &\overset{(2)}{=} \mathrm{E}\left[e^{\theta \max_{0 \leq i \leq s}\{A(i,t)-S(i,s)\}}\right] \\ &\leq \sum_{i=0}^{s} \mathrm{E}\left[e^{\theta(A(i,t)-S(i,s))}\right],\end{aligned} \tag{5}$$

where the max is always less than or equal to the sum since we have only non-negative terms. Inequality (5) is similar to the application of the Union bound[2],

$$\mathrm{P}\left(\max_{i=1,\dots,n} X_i > a\right) \leq \sum_{i=1}^{n} \mathrm{P}(X_i > a). \tag{6}$$

In the following, we call the bound in (5) "standard approach" given that it is the most intuitive way to proceed. It has been shown to often perform poorly, in particular for correlated increments. The authors of [15] suggested instead a martingale-based approach that allows for significantly more accurate delay bounds. To the best of our knowledge, however, achieving a concatenation property to enable an end-to-end analysis remains an elusive goal in the martingale-based approach.

The idea in this paper is to insert Jensen's inequality to mitigate the inaccuracy imposed by the (5). 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 3.2.

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

$$\mathrm{P}\left(\max_{i=1,\dots,n} X_i > a\right) \overset{(6)}{\leq} \sum_{i=1}^{n} \mathrm{P}(X_i > a) \overset{(1)}{\leq} e^{-\theta a} \sum_{i=1}^{n} \mathrm{E}\left[e^{\theta X_i}\right]$$

$$\Leftrightarrow \mathrm{P}\left(\max_{i=1,\dots,n} X_i > a\right) \overset{(1)}{\leq} e^{-\theta a}\, \mathrm{E}\left[\max_{i=1,\dots,n} e^{\theta X_i}\right] \overset{(5)}{\leq} e^{-\theta a} \sum_{i=1}^{n} \mathrm{E}\left[e^{\theta X_i}\right]$$

Therefore, we call the inequality in (5) in the following "quasi-Union bound."

Theorem 8 (Jensen's Inequality [23])**.** *Suppose that h is a differentiable convex function on $\mathbb{R}$ and let $X \in \mathcal{L}^1$. Then*

$$h(\mathrm{E}[X]) \leq \mathrm{E}[h(X)]\,. \tag{7}$$

This enables us to prove a new output bound.

Proposition 9 (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*

$$\begin{aligned}&\mathrm{E}\left[e^{\theta A'(s,t)}\right]\\ &\leq \inf_{p \in P}\left\{h_p^{-1}\left(\sum_{i=0}^{s}\mathrm{E}\left[h_p\left(e^{\theta(A(i,t)-S(i,s))}\right)\right]\right)\right\}.\end{aligned} \tag{8}$$

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

$$h_{\bar{p}}(x) = x, \tag{9}$$

then this bound is always at least as good as the standard approach in (5)*.*

Proof. We know by Jensen's inequality that

$$\begin{aligned}&\mathrm{E}\left[e^{\theta A'(s,t)}\right]\\ &\overset{(3)}{\leq} \mathrm{E}\left[e^{\theta \max_{0\leq i\leq s}\{A(i,t)-S(i,s)\}}\right]\\ &\overset{(7)}{\leq} \inf_{p\in P}\left\{h_p^{-1}\left(\mathrm{E}\left[h_p\left(e^{\theta \max_{0\leq i\leq s}\{A(i,t)-S(i,s)\}}\right)\right]\right)\right\}\\ &= \inf_{p\in P}\left\{h_p^{-1}\left(\mathrm{E}\left[\max_{0\leq i\leq s} h_p\left(e^{\theta(A(i,t)-S(i,s))}\right)\right]\right)\right\}\\ &\leq \inf_{p\in P}\left\{h_p^{-1}\left(\sum_{i=0}^{s}\mathrm{E}\left[h_p\left(e^{\theta(A(i,t)-S(i,s))}\right)\right]\right)\right\},\end{aligned}$$

where we used that strictly increasing function on $\mathbb{R}$ always have an inverse in the third 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

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

where we used that there is one $\bar{p} \in P$ such that (9) 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 9.2.

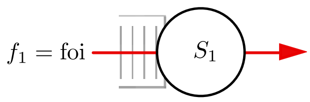


Figure 2: One server topology.

3.2. Power-Mitigator and (σ, ρ)-Bounds

In this subsection, we show that Proposition 9 generalizes the results in [26]. Moreover, we restate the compatibility with the (σ, ρ)-bounds in Definition 4.

Proposition 9 yields an output bound given a parameterized function h_p. A suitable candidate for h_p is the power function

$$\begin{aligned}h_p : \mathbb{R}^+ &\to \mathbb{R}^+\\ x &\mapsto x^p,\end{aligned} \tag{10}$$

where $p \geq 1$, because it suffices the necessary conditions of both parts of Proposition 9 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 10 (Power-Mitigator)**.** *Let h_p be defined as in* (10)*. Then it holds that*

$$\begin{aligned}\mathrm{E}\left[e^{\theta A'(s,t)}\right] &\leq \inf_{p\geq 1}\left\{\left(\sum_{i=0}^{s}\mathrm{E}\left[e^{p\theta(A(i,t)-S(i,s))}\right]\right)^{\frac{1}{p}}\right\} \\ &\leq \sum_{i=0}^{s}\mathrm{E}\left[e^{\theta\{A(i,t)-S(i,s)\}}\right],\end{aligned} \tag{11}$$

i.e., we receive a new output bound (11) *that guarantees to be as good as the standard approach in* (5)*.*

Here, we see that the subadditivity of the root function implies that the insertion can mitigate the effect of the quasi-Union bound (5).

Single Server Setting. Assume now a single flow - single server setting as in Figure 2. We have already deduced that

$$\begin{aligned}\mathrm{E}\left[e^{\theta\left(A'(s,t)\right)}\right] &\overset{(3)}{\leq} \mathrm{E}\left[e^{\theta(A \oslash S)\,(s,t)}\right]\\ &\overset{(5)}{\leq} \sum_{i=0}^{s}\mathrm{E}\left[e^{\theta(A(i,t)-S(i,s))}\right].\end{aligned}$$

We now require the arrivals and service to have (σ, ρ)-constraints (Definition 4). For $\rho_A(\theta) < \rho_S(-\theta)$ the standard approach leads to

$$\begin{aligned}\mathrm{E}\left[e^{\theta\left(A'(s,t)\right)}\right] &\leq \sum_{i=0}^{s}\mathrm{E}\left[e^{\theta(A(i,t)-S(i,s))}\right]\\ &= \sum_{i=0}^{s}\mathrm{E}\left[e^{\theta A(i,t)}\right]\mathrm{E}\left[e^{-\theta S(i,s)}\right]\end{aligned}$$

$$
\begin{aligned}
&\leq \sum_{i=0}^{s} e^{\theta\rho_A(\theta)(t-i)+\theta\sigma_A(\theta)} e^{-\theta\rho_S(-\theta)(s-i)+\theta\sigma_S(-\theta)} \\
&= e^{\theta(\rho_A(\theta)(t-s)+\sigma_A(\theta)+\sigma_S(-\theta))} \\
&\quad \cdot \sum_{j=0}^{s} e^{\theta(\rho_A(\theta)-\rho_S(-\theta))j} \\
&\leq \frac{e^{\theta(\rho_A(\theta)(t-s)+\sigma_A(\theta)+\sigma_S(-\theta))}}{1-e^{\theta(\rho_A(\theta)-\rho_S(-\theta))}}, \qquad (12)
\end{aligned}
$$

where we have used the independence of arrivals and service in the second line, (σ, ρ)-bounds in the third line and the convergence of the geometric series in the last line. The shows that the output is (σ, ρ)-bounded as well (see Proposition 11 below).

If we use Jensen's inequality with the power function (11) instead, we obtain in comparison

$$
\begin{aligned}
&\mathrm{E}\Big[e^{\theta A'(s,t)}\Big] \\
&\leq \inf_{p\geq 1}\left\{\left(\frac{e^{p\theta(\rho_A(p\theta)(t-s)+\sigma_A(p\theta)+\sigma_S(-p\theta))}}{1-e^{p\theta(\rho_A(p\theta)-\rho_S(-p\theta))}}\right)^{\frac{1}{p}}\right\} \\
&= \inf_{p\geq 1}\left\{\frac{e^{\theta(\rho_A(p\theta)(t-s)+\sigma_A(p\theta)+\sigma_S(-p\theta))}}{\left(1-e^{p\theta(\rho_A(p\theta)-\rho_S(-p\theta))}\right)^{\frac{1}{p}}}\right\}. \qquad (13)
\end{aligned}
$$

Thus, the power-mitigator can also be used under (σ, ρ)-constraints (see Proposition 12). That is, it can easily be integrated in existing end-to-end analyses.

Proposition 11 ([8][25]). *Consider a (σ_A, ρ_A)-bounded arrival process $A(s,t)$ with (σ_S, ρ_S)-bounded dynamic S-server $S(s,t)$, as in Figure 2. If the stability condition $\rho_A(\theta) < \rho_S(-\theta)$ holds, then the output A' is $(\sigma_{A'}, \rho_{A'})$-bounded with*

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

Proposition 12 (The Output Bound with the Power-Mitigator is (σ, ρ)-Bounded). *Under the assumptions in Proposition 11 plus a modified stability condition $\rho_A(p\theta) < \rho_S(-p\theta)$, we obtain that the output A' is $(\sigma_{A'}, \rho_{A'})$-bounded with*

$$
\begin{aligned}
\sigma_{A'}(p\theta) &= \sigma_A(p\theta) + \sigma_S(-p\theta) \\
&\quad - \frac{1}{p\theta}\log\left(1-e^{p\theta(\rho_A(p\theta)-\rho_S(-p\theta))}\right) \\
\rho_{A'}(p\theta) &= \rho_A(p\theta),
\end{aligned}
$$

where $p \geq 1$.

Proof. See Appendix A.1.

Remark 13 (Computational Advantage). The bounds in (5) and (8) give an estimate for the min-plus operators in Theorem 7, but are computationally infeasible for

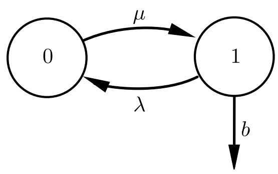


Figure 3: MMOO Model.

larger networks. Since the number of sums in these calculations typically scales linearly with the number of invoked min-plus operators, one usually seeks for stationary closed-form solutions. Using (σ, ρ)-bounds conveniently solves this problem by letting these sums converge. The computational advantage can be observed as follows:
The quasi-Union bound yields

$$
\mathrm{E}\Big[e^{\theta A'(s,t)}\Big] \overset{(5)}{\leq} \sum_{i=0}^{s} \mathrm{E}\Big[e^{\theta(A(i,t)-S(i,s))}\Big],
$$

i.e., we have to compute a sum with $s+1$ summands. With the additional assumption of (σ, ρ)-constraints, the output can be bounded by the closed form $\mathrm{E}\Big[e^{\theta A'(s,t)}\Big] \overset{(12)}{\leq} \frac{e^{\theta(\rho_A(\theta)(t-s)+\sigma_A(\theta)+\sigma_S(-\theta))}}{1-e^{\theta(\rho_A(\theta)-\rho_S(-\theta))}}$.

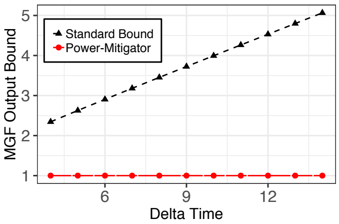
Remark 14 (Improvement of the Output Bound). The power-mitigator's MGF output bound (13) can be moved arbitrarily close to its lower bound 1. This can be seen as follows: Assume a sequence p_n and θ_n such that $p_n\theta_n \to c$ with constant $c > 0$ and that the MGF $\mathrm{E}\big[e^{\theta(A(i,t)-S(i,s))}\big]$ exists, where $p_n \to \infty$ and $\theta_n \to 0$ but $\theta_n > 0$ for all n (for example, choose $\theta_n = \frac{1}{n}$ and $p_n = c \cdot n$). Then it holds that

$$
\begin{aligned}
\mathrm{E}\Big[e^{\theta A'(s,t)}\Big] &\leq \lim_{n\to\infty} \frac{e^{\theta_n(\rho_A(p_n\theta_n)(t-s)+\sigma_A(p_n\theta_n)+\sigma_S(p_n\theta_n))}}{\left(1-e^{p_n\theta_n(\rho_A(p_n\theta_n)-\rho_S(p_n\theta_n))}\right)^{\frac{1}{p_n}}} \\
&\simeq \lim_{n\to\infty} \frac{e^{\theta_n(\rho_A(c)(t-s)+\sigma_A(c)+\sigma_S(c))}}{\left(1-e^{c(\rho_A(c)-\rho_S(c))}\right)^{\frac{1}{p_n}}} \\
&= 1,
\end{aligned}
$$

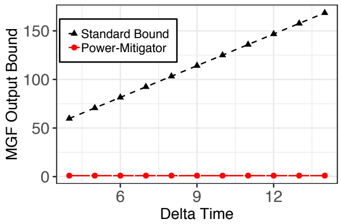
where we used that the numerator converges to 1 for arbitrary small θ and the denominator converges also to 1 as $\rho_A(c) < \rho_S(c) \Rightarrow 1 - e^{c(\rho_A(c)-\rho_S(c))} \in (0,1)$. Examples of this MGF output bound behavior are depicted in Figure 4. Therefore, we focus in the evaluation on the impact on the delay in topologies with multiple server.

4. Evaluation

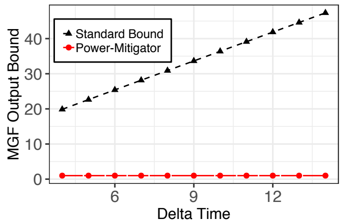
In this section, we investigate the increased accuracy of our new output bound introduced in Section 3. That is, we evaluate the gain of the delay bound by the improved output bound calculation calculation for a two-server topology and a fat tree. The improvement factor is measured



(a) D/M/1 with $\lambda = 3.8$, service rate $r = 3$



(b) MMOO with $\mu = 8$, $\lambda = 12$, $b = 3$, service rate $r = 1.5$



(c) M/D/1 with $\lambda = 0.5$, service rate $r = 1.0$

Figure 4: MGF Output bound comparison in the single server setting.

by calculating

$$\frac{\text{Bound standard approach}}{\text{Bound power-mitigator}}, \tag{14}$$

where clearly larger values are desirable.

The formulae are implemented in the general-purpose programming language Python[3]. We made the source code publicly available[4]. Throughout this paper, we only consider arrivals that are (σ, ρ)-bounded. Therefore, all arrival processes inherit in an object-oriented manner from the class "Arrival" that has the abstract methods "sigma(theta)" and "rho(theta)".

We have implemented three different arrival distributions:

1. **D/M/1** Independent exponentially distributed increments with parameter λ:
   For this model, the MGF is given by
   $$\begin{aligned} \mathrm{E}\left[e^{\theta A(s,t)}\right] &= \mathrm{E}\left[e^{\theta \sum_{i=s+1}^{t} a(i)}\right] \\ &= \left(\frac{\lambda}{\lambda - \theta}\right)^{t-s}, \quad 0 < \theta < \lambda,\ s \leq t. \end{aligned}$$
   Since the packets arrive with constant inter-arrival times due the discrete-time model and by the choice of the arrivals' distribution, this basically corresponds to a D/M/1-queue.
2. **MMOO** Markov-Modulated On-Off traffic model:
   It consists of a continuous-time Markov chain with two states, 0 and 1, together with transition rates μ and λ. If the chain is in state 0, it means that no traffic arrives, whereas in state 1, data with burst rate b are sent (see Figure 3). It has been shown in [27] that, for this arrival model, the MGF can be bounded by
   $$\mathrm{E}\left[e^{\theta A(s,t)}\right] \leq e^{\theta\omega(\theta)\cdot(t-s)}, \quad \theta > 0,\ s \leq t,$$
   where $\omega(\theta) = \frac{-d+\sqrt{d^2+4\mu\theta b}}{2\theta}$ and $d = \mu + \lambda - \theta b$. However, in contrast to the exponentially distributed increments above, the MMOO traffic model is a continuous process $A(s,t) = \int_s^t a(x)\mathrm{d}x$, $s \leq t$. Therefore, we also need discretizing techniques such as in [10] as we use a discrete time model.
3. **M/D/1** Independent exponentially distributed inter-arrival times with parameter λ:
   In contrast to 1), the time between arrival of size 1 is distributed exponentially, yielding the MGF [28]
   $$\mathrm{E}\left[e^{\theta A(s,t)}\right] = e^{\lambda(t-s)\left(e^{\theta}-1\right)},\ s \leq t.$$
   This arrival class corresponds to the M/D/1-queue and is, as for the MMOO traffic, a continuous-time model that needs to be discretized.

[3]https://www.python.org

[4]https://github.com/paulnikolaus/snc-mgf-toolbox

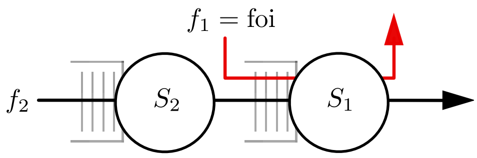
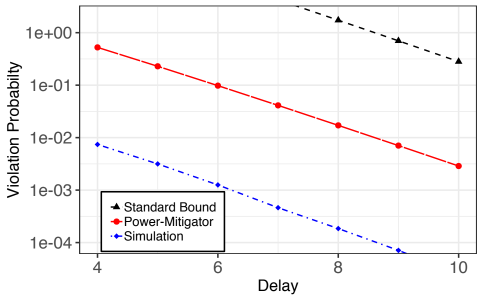


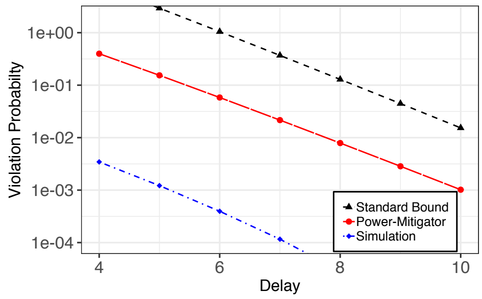
Figure 5: Two-server topology.

The service is always chosen to be work-conserving and of constant rate. Its according class also has the methods "sigma(theta)" and "rho(theta)". All mentioned arrival and service processes can be described by (σ, ρ)-bounds, therefore all network calculus operations such as the convolution of server or the computation of the leftover service yield closed-form solutions that can also be represented by (σ, ρ)-bounds [29, 17]. Implementation-wise, this means that all operations inherit either from the arrival or the server class. As a consequence, all performance bounds can be obtained via closed-form solution, as well. The same holds for the output bound with the power-mitigator (13) as shown in Proposition 12.

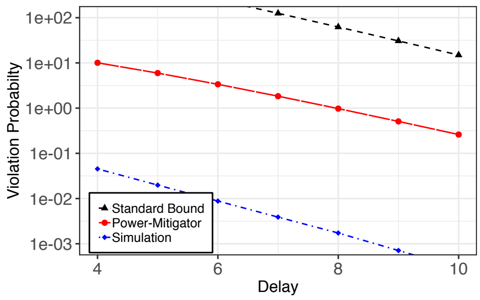
If not stated otherwise, θ and the Jensen parameters p_i are optimized by a brute force optimization along a grid using the "brute()" method in the "scipy.optimize" library [30]. When applying a second optimization with the best grid point as initial solution, this method also evaluates points outside the grid meaning that we actually apply an unconstrained optimization to a constrained problem. Therefore, we set the function value to ∞ ("math.inf") if θ or p are not in the feasible set ensuring that these points are not in an optimal solution.



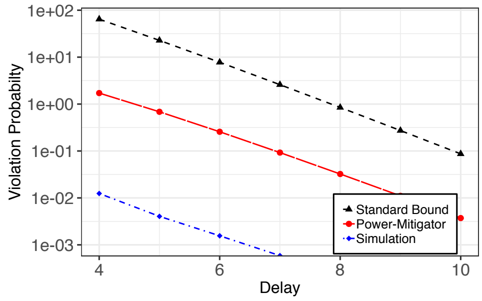
(a) D/M/1 with $(\lambda_1, \lambda_2) = (0.2, 8.0)$, service rates $(r_1, r_2) = (8.0, 0.2)$



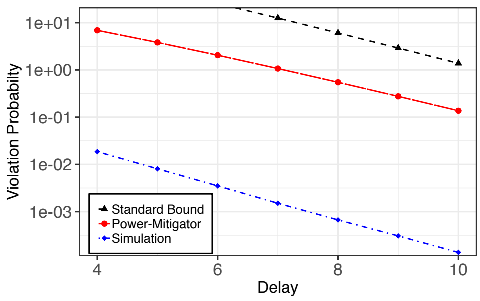
(b) D/M/1 with $(\lambda_1, \lambda_2) = (0.4, 3.5)$, service rates $(r_1, r_2) = (4.5, 0.4)$



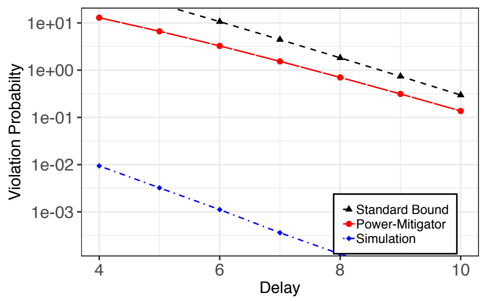
(c) MMOO with $(\mu_1, \mu_2) = (1.2, 3.7)$, $(\lambda_1, \lambda_2) = (2.1, 1.5)$, $(b_1, b_2) = (3.5, 0.4)$, service rates $(r_1, r_2) = (2.0, 0.3)$



(d) MMOO with $(\mu_1, \mu_2) = (1.0, 3.6)$, $(\lambda_1, \lambda_2) = (2.2, 1.6)$, $(b_1, b_2) = (3.4, 0.4)$, service rates $(r_1, r_2) = (2.0, 0.3)$



(e) M/D/1 with $(\lambda_1, \lambda_2) = (1.6, 0.01)$, service rates $(r_1, r_2) = (2.0, 0.15)$



(f) M/D/1 with $(\lambda_1, \lambda_2) = (3.6, 0.28)$, service rates $(r_1, r_2) = (4.4, 0.7)$

Figure 6: Delay bound comparison in the two-server setting.

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 old approach). Hence, we gain more flexibility while being minimally invasive at the same time.

4.1. 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 5. 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 by being more accurate in terms of the foi's leftover service. Mathematically speaking, this leftover service at S_1 is described by $S_{1,\text{l.o.}} = [S_1 - (A_2 \oslash S_2)]^+$. In this topology, we calculate the delay bound (4) but take the new output bound invocation into account. For the D/M/1, M/D/1, and Markov-Modulated On-Off (MMOO) traffic, two examples for each distribution are depicted in Figure 6. The plot is complemented by delay measurements in a packet-level simulation. Here, the violation probability is estimated by the empirical distribution computing the average number of occurred delays. 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

Distribution	D/M/1	MMOO	M/D/1
Average gain	1.40	1.04	1.03
Maximum gain	135.0	36.6	15.5
Share of improved bounds	99.8%	99.8%	99.9%

Distribution	D/M/1	MMOO	M/D/1
Average gain	1.47	1.30	1.13
Maximum gain	93.2	68.5	10.9
Share of improved bounds	100%	99.3%	100%

Table 1: Improvement of the delay's violation probability for the two-server setting and delay = 10 (above: uniform sampling, below: exponential sampling).

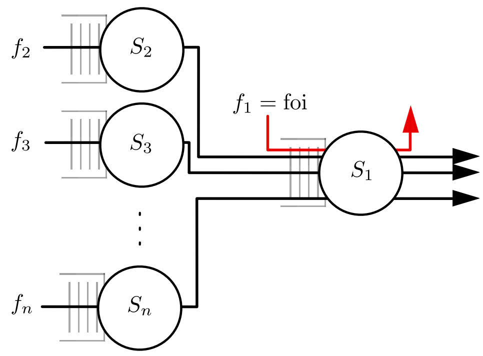


Figure 7: Fat tree topology.

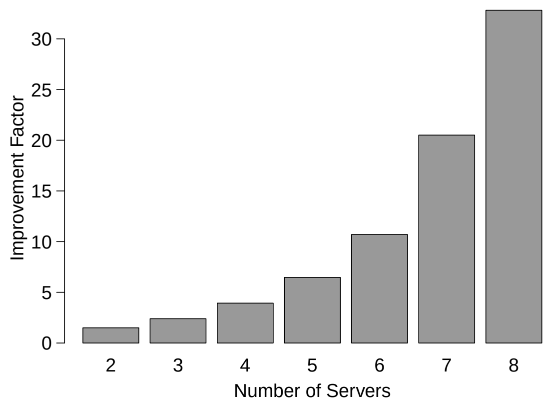


Figure 8: Delay bound improvement for different numbers of servers (delta time = 8).

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 1. Furthermore, since we aim to focus 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 6. It shows that even in the delay space (the difference in the delay bound for a given probability), the difference is up to 50%. Depending on the parameters, the gap between the simulation results and the analytically derived bounds can be closed considerably. Average behavior on the other hand is less significant. Table 1 indicates a highly non-linear behavior where some violation probabilities are improved by a factor of 135.0, whereas average gain is moderate with a total mean of 1.23.

4.2. Fat Tree

Starting off with the two-server topology in Figure 5, 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). 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 $S_{1,\text{l.o.}} = \left[S_1 - \sum_{i=2}^{n} (A_i \oslash S_i)\right]^+$.

We calculated the delay's violation probability for the following setting: The foi has exponentially distributed increments (D/M/1) with parameter $\lambda_1 = 0.5$ and enters server S_1 with rate $r_1 = 4$. The $n-1$ cross flows are also D/M/1, 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 8.

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.

4.3. Run Time

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 D/M/1 as well as MMOO-traffic in the two-server topology (Figure 5) and the fat tree (Figure 7) 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 so-called Pattern Search [31] 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

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

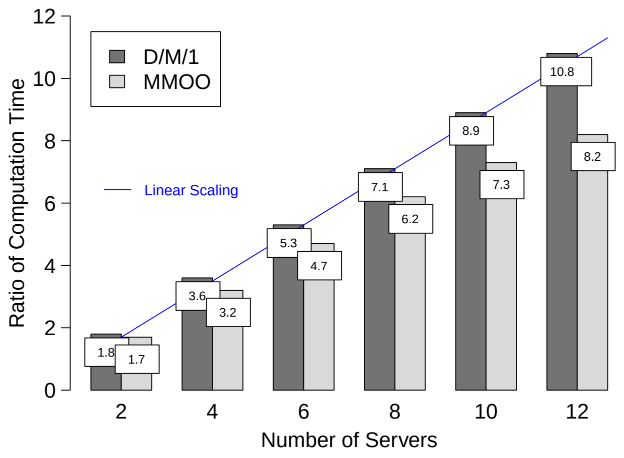


Figure 9: Computation time comparison for the state-of-the-art and power function approach.

for these experiments are depicted in Figure 9.

For Pattern Search, we observe that the computational overhead scales only linearly with the number of invocations of Jensen's inequality. This indicates that a good trade-off between cost and accuracy gain can be achieved, if optimization is done carefully.

5. 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 4. That is, we would modify the computation of the delay's violation probability as follows:

$$
\begin{aligned}
&\mathrm{P}(d(t) > T) \\
&\overset{(4)}{\leq} \mathrm{E}\Big[e^{\theta (A \oslash S)\,(t+T,t)}\Big] \\
&\overset{(2)}{=} \mathrm{E}\Big[e^{\theta \max_{0\leq i\leq t+T}\{A(i,t)-S(i,t+T)\}}\Big] \\
&= \mathrm{E}\Big[e^{\theta \max_{0\leq i\leq t}\{A(i,t)-S(i,t+T)\}}\Big] \\
&= \inf_{p\geq 1}\left\{\left(\mathrm{E}\Big[e^{p\theta \max_{0\leq i\leq t}\{A(i,t)-S(i,t+T)\}}\Big]\right)^{\frac{1}{p}}\right\} \qquad (15)\\
&\overset{(5)}{\leq} \inf_{p\geq 1}\left\{\left(\sum_{i=0}^{t}\mathrm{E}\Big[e^{p\theta(A(i,t)-S(i,t+T))}\Big]\right)^{\frac{1}{p}}\right\}, \qquad (16)
\end{aligned}
$$

where we used that $A(s,t) = 0$ for $s \geq t$ in the third line, the identity-property for $p = 1$ in (15), 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 (16). Disappointingly for this case, no improvement can be obtained, as the next theorem states.

Theorem 15. *Let a delay bound T according to* (16) *exist such that*

$$\sum_{i=0}^{t}\mathrm{E}\Big[e^{p\theta(A(i,t)-S(i,t+T))}\Big] < 1. \qquad (17)$$

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 (16) and that $l^* > 1$. This means that there exist $1 \leq p' < p^*$ and $\theta' > \theta^*$ such that $p'\theta' = p^*\theta^*$. But this means

$$
\begin{aligned}
&\left(\sum_{i=0}^{t}\mathrm{E}\Big[e^{p^*\theta^*(A(i,t)-S(i,t+T))}\Big]\right)^{\frac{1}{p^*}} \\
&= \left(\sum_{i=0}^{t}\mathrm{E}\Big[e^{p'\theta'(A(i,t)-S(i,t+T))}\Big]\right)^{\frac{1}{p^*}} \\
&> \left(\sum_{i=0}^{t}\mathrm{E}\Big[e^{p'\theta'(A(i,t)-S(i,t+T))}\Big]\right)^{\frac{1}{p'}},
\end{aligned}
$$

where we inserted $p^*\theta^* = p'\theta'$ in the second line. In the third 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).

6. The Exp-Mitigator

In this subsection, we investigate an alternative h-mitigator using the exponential function

$$
\begin{aligned}
h_p &: \mathbb{R}^+ \to \mathbb{R}^+ \\
x &\mapsto p^x, \qquad (18)
\end{aligned}
$$

where $p > 1$, as it is differentiable, strictly increasing, and convex. But, since there is no p such that p^x equals the identity, only the first part of Proposition 9 can be utilized.

Corollary 16 (Exp-Mitigator)**.** *Let h_p be defined as in* (18)*. Then it holds that*

$$\mathrm{E}\Big[e^{\theta A'(s,t)}\Big] \leq \inf_{p>1}\left\{\log_p\left(\sum_{i=0}^{s}\mathrm{E}\Big[p^{\left(e^{\theta(A(i,t)-S(i,s))}\right)}\Big]\right)\right\}. \qquad (19)$$

Above all, we cannot guarantee that this bound does not deteriorate in comparison to the standard bound (5).

The bound in (19) is difficult to obtain analytically even for simple distributions such as exponentially distributed increments (D/M/1). Taking the expectation inside, i.e.,

$$\inf_{p>1}\left\{\log_p\left(\sum_{i=0}^{s}p^{\left(e^{\theta\,\mathrm{E}[A(i,t)-S(i,s)]}\right)}\right)\right\} \qquad (20)$$

would lead to a lower bound on the upper bound (19) (again, by Jensen's inequality). This significantly restricts the applicability of the exp-mitigator.

6.1. Numerical Evaluation of the Exp-Mitigator

In the following, we compare the delay bounds obtained with the exp-mitigator and the standard approach. Given that we do not have a closed-form solution for the exp-mitigator, we limit ourselves to the single server topology (Figure 2). Based on (14), we measure the improvement via

$$\frac{\text{Bound standard approach}}{\text{Bound exp-mitigator}},$$

because for this topology, the power-mitigator could not improve the bound on the delay violation probability (see Section 5). The exp-mitigator's delay violation probability (derived in Appendix A.2),

$$\mathrm{P}(d(t) > T) \leq \inf_{p>1} \left\{ \log_p \left(\sum_{i=0}^{t} \mathrm{E}\left[p^{\left(e^{\theta(A(i,t)-S(i,t+T))}\right)} \right] \right) \right\},$$

lacking a closed form, is implemented with the non-stationary lower bound

$$\inf_{p>1} \left\{ \log_p \left(\sum_{i=0}^{t} p^{\left(e^{\theta(\mathrm{E}[A(i,t)]-\mathrm{E}[S(i,t+T)])}\right)} \right) \right\}$$

to avoid calculating the double exponential of the expectation. As a consequence, the exp-mitigator has a strong advantage in the evaluation.

In a supplementary evaluation, we approximate the violation probability by using the strong law of large numbers [32]. That is, for i.i.d. random variables $\in \mathcal{L}^1$ and measurable function f it holds that

$$\lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} f(X_i) \to f(\mathrm{E}[X_1]) \quad \text{a.s.}$$

For the sake of simplicity, we only assume the arrivals to be of exponential distributed increments (D/M/1) and a service of constant rate. As for the parameter choice, we apply the same Monte Carlo-type analysis like in the previous numerical evaluation. The results are shown in Table 2.

On average, the lower bound is about 19% worse. On the other hand, a maximum improvement factor of 16.5 is achieved. Yet, given that this is only a lower bound, it is quite surprising that only at most 2.2% of the bounds are improved. The performance gap is for sample mean approximation is even larger, bearing in mind that nearly no enhancement at all is obtained despite the choice of stationary bounds as competitors.

Remark 17 (Relation p and t). Let us fix a $p > 1$ for the moment. This results for the bound on the violation probability $\mathrm{P}(d(t) > T)$ in

$$\begin{aligned} &\log_p \left(\sum_{i=0}^{t} \mathrm{E}\left[p^{\left(e^{\theta(A(i,t)-S(i,t+T))}\right)} \right] \right) \\ &\geq \log_p \left(\sum_{i=0}^{t} \mathrm{E}\left[p^0\right] \right) \\ &= \log_p (t+1), \end{aligned}$$

Distribution	Lower Bound	Sample Mean
Average gain	1.05	0.09
Maximum gain	14.2	0.96
Share of improved bounds	18.4%	0%

Distribution	Lower Bound	Sample Mean
Average gain	0.63	0.06
Maximum gain	16.5	1.03
Share of improved bounds	9.7%	1.0%

Table 2: Delay bound improvement for a single server and delay = 10 (above: uniform sampling, below: exponential sampling).

where we used in the second line that $e^x > 0$ for all $x \in \mathbb{R}$. This means that, in order to make sure that the bound on the violation probability is less than 1, p has to be greater than $t + 1$. As a consequence, there is no finite p in this approach for a stationary delay bound.

Besides this strong quantitative evidence, the exp-mitigator also exhibits a number of qualitative disadvantages:

- The lack of closed-form solutions significantly increases the run time as the sums have to be calculated from every single summand making it rather elusive for a network analysis.
- The double exponential causes many floating-point errors that have to be caught and subsequently removed from the evaluation.

Following these arguments, we can clearly recommend the usage of the power-mitigator.

7. Conclusion

In this paper, 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 that shows why this is a valid bound and, when using the power function, 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 the Jensen's inequality under a carefully chosen optimization method. We also showed in a numerical evaluation, that, in contrast, the application of the exp-mitigators often times provides worse delay bounds apart from having far higher computation times caused by the lack of a closed-form solution.

Considering the crucial role of the output bound, we believe that we have made a significant contribution to the SNC network analysis. On the other hand, there are still many open challenges in the analysis of larger and more complex networks, e.g., dealing effectively with correlations in the traffic flows, which are left for future work.

References

[1] A. Singla, B. Chandrasekaran, P. B. Godfrey, B. Maggs, The internet at the speed of light, in: Proc. ACM Workshop on Hot Topics in Networks'14, HotNets-XIII, 2014, pp. 1–7.

[2] G. P. Fettweis, The tactile internet: Applications and challenges, IEEE Vehicular Technology Magazine 9 (1) (2014) 64–70.

[3] A. Whitmore, A. Agarwal, L. Da Xu, The internet of things–a survey of topics and trends, Springer Information Systems Frontiers 17 (2) (2015) 261–274.

[4] R. R. Rajkumar, I. Lee, L. Sha, J. Stankovic, Cyber-physical systems: the next computing revolution, in: Proc. ACM Design Automation Conference'10, 2010, pp. 731–736.

[5] F. Baccelli, G. Cohen, G. J. Olsder, J.-P. Quadrat, Synchronization and linearity: an algebra for discrete event systems, John Wiley & Sons Ltd, 1992.

[6] R. L. Cruz, A calculus for network delay, part I: Network elements in isolation, IEEE Transactions on information theory 37 (1) (1991) 114–131.

[7] R. L. Cruz, A calculus for network delay, part II: Network analysis, IEEE Transactions on information theory 37 (1) (1991) 132–141.

[8] C.-S. Chang, Performance guarantees in communication networks, Springer-Verlag, London, 2000.

[9] J.-Y. Le Boudec, P. Thiran, Network calculus: a theory of deterministic queuing systems for the internet, Springer-Verlag, New York, 2001.

[10] F. Ciucu, A. Burchard, J. 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, 2005, pp. 279–290.

[11] M. Fidler, An end-to-end probabilistic network calculus with moment generating functions, in: Proc. IEEE IWQoS'06, 2006, pp. 261–270.

[12] Y. Jiang, Y. Liu, Stochastic network calculus, Vol. 1, Springer, 2008.

[13] F. Ciucu, J. 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), New York, NY, USA, 2012, pp. 311–322.

[14] C. Li, A. Burchard, J. Liebeherr, A network calculus with effective bandwidth, IEEE/ACM Transactions on Networking 15 (6) (2007) 1442–1453.

[15] F. Ciucu, F. Poloczek, J. 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.

[16] A. Rizk, F. Poloczek, F. Ciucu, Computable bounds in fork-join queueing systems, in: Proc. ACM International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS'15), Vol. 43, 2015, pp. 335–346.

[17] T. Zhu, D. S. Berger, M. Harchol-Balter, SNC-meister: Admitting more tenants with tail latency SLOs, in: Proc. ACM Symposium on Cloud Computing (SoCC'16), 2016.

[18] M. Fidler, Y. Jiang, Non-asymptotic delay bounds for (k, l) fork-join systems and multi-stage fork-join networks, in: Proc. IEEE International Conference on Computer Communications (INFOCOM'16), 2016, pp. 1–9.

[19] S. Bondorf, P. Nikolaus, J. B. Schmitt, Quality and cost of deterministic network calculus - design and evaluation of an accurate and fast analysis, in: Proc. ACM International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS'17), 2017.

[20] F. Poloczek, F. Ciucu, Scheduling analysis with martingales, Elsevier Performance Evaluation 79 (2014) 56–72.

[21] F. Poloczek, F. 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.

[22] F. Ciucu, F. Poloczek, J. 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.

[23] R. Nelson, Probability, stochastic processes, and queueing theory: the mathematics of computer performance modeling, Springer, 1995.

[24] J. B. Schmitt, F. A. Zdarsky, M. 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, 2008.

[25] M. A. Beck, Advances in theory and applicability of stochastic network calculus, Ph.D. thesis, TU Kaiserslautern (2016).

[26] P. Nikolaus, J. Schmitt, Improving Output Bounds in the Stochastic Network Calculus Using Lyapunov's Inequality, in: Proc. IFIP Networking 2018 Conference (NETWORKING'18), Zurich, Switzerland, 2018.

[27] C. Courcoubetis, R. Weber, Buffer overflow asymptotics for a buffer handling many traffic sources, Journal of Applied Probability 33 (1996) 886–903.

[28] F. Ciucu, Network calculus delay bounds in queueing networks with exact solutions, in: Proc. IEEE International Teletraffic Congress (ITC 20'07), 2007, pp. 495–506.

[29] M. A. Beck, J. Schmitt, The disco stochastic network calculator version 1.0 - when waiting comes to an end, in: Proc. International Conference on Performance Evaluation Methodologies and Tools (VALUETOOLS 2013), 2013.

[30] E. Jones, T. Oliphant, P. Peterson, et al., SciPy: Open source scientific tools for Python (2001–). URL `http://www.scipy.org/`

[31] R. Hooke, T. A. Jeeves, "Direct Search" Solution of Numerical and Statistical Problems, Journal of the ACM (JACM) 8 (2) (1961) 212–229.

[32] S. Ross, A First Course in Probability, Vol. 8, Pearson, 2010.

Appendix A. Appendix

Appendix A.1. Proof of Proposition 12

We have already seen in Subsection 3.2 that

$$\mathrm{E}\left[e^{\theta A'(s,t)}\right] \le \inf_{p\ge 1}\left\{\left(\sum_{i=0}^{s}\mathrm{E}\left[e^{p\theta(A(i,t)-S(i,s))}\right]\right)^{\frac{1}{p}}\right\},$$

which can be continued with

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

$$
\begin{aligned}
&\leq \inf_{p\geq 1}\Bigg\{ e^{\theta(\sigma_A(p\theta)+\sigma_S(-p\theta))} \\
&\quad \cdot \left(\sum_{i=0}^{s} e^{p\theta(\rho_A(p\theta)(t-i)-\rho_S(-p\theta)(s-i))}\right)^{\frac{1}{p}} \Bigg\},
\end{aligned}
$$

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 that $\rho_A(p\theta) < \rho_S(-p\theta)$, we obtain by convergence of the geometric series

$$
\begin{aligned}
\cdots &= \inf_{p\geq 1}\Bigg\{ e^{\theta(\rho_A(p\theta)(t-s)+\sigma_A(p\theta)+\sigma_S(-p\theta))} \\
&\quad \cdot \left(\sum_{j=0}^{s} e^{p\theta(\rho_A(p\theta)-\rho_S(-p\theta))j}\right)^{\frac{1}{p}} \Bigg\} \\
&\leq \inf_{p\geq 1}\Bigg\{ e^{\theta(\rho_A(p\theta)(t-s)+\sigma_A(p\theta)+\sigma_S(-p\theta))} \\
&\quad \cdot \left(\frac{1}{1-e^{p\theta(\rho_A(p\theta)-\rho_S(-p\theta))}}\right)^{\frac{1}{p}} \Bigg\}.
\end{aligned}
$$

This finishes the proof, as it is equal to

$$
\begin{aligned}
\cdots &= \inf_{p\geq 1}\Bigg\{ e^{\theta(\rho_A(p\theta)(t-s)+\sigma_A(p\theta)+\sigma_S(-p\theta))} \\
&\quad \cdot e^{\theta\left(-\frac{1}{p\theta}\log\left(1-e^{p\theta\left(\rho_A(p\theta)-\rho_S(-p\theta)\right)}\right)\right)} \Bigg\},
\end{aligned}
$$

which yields

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

as the theorem states.

Appendix A.2. Delay Bound of Exp-Mitigator

The delay bound using the Jensen bound with the exponential function yields

$$
\begin{aligned}
\mathrm{P}(d(t) > T) &\leq \mathrm{E}\Big[e^{\theta(A\oslash S)\,(t+T,t)}\Big] \\
&= \mathrm{E}\Big[e^{\theta\max_{0\leq i\leq t}\{A(i,t)-S(i,t+T)\}}\Big] \\
&\leq \inf_{p>1}\left\{\log_p\left(\mathrm{E}\Big[p^{\left(e^{\theta\max_{0\leq i\leq t}\{A(i,t)-S(i,t+T)\}}\right)}\Big]\right)\right\} \\
&\leq \inf_{p>1}\left\{\log_p\left(\sum_{i=0}^{t}\mathrm{E}\Big[p^{\left(e^{\theta(A(i,t)-S(i,t+T))}\right)}\Big]\right)\right\}.
\end{aligned}
$$