Improving Delay Bounds in the Stochastic Network Calculus by Using less Stochastic Inequalities

Paul Nikolaus
Distributed Computer Systems (DISCO) Lab,
TU Kaiserslautern
Kaiserslautern, Germany
nikolaus@cs.uni-kl.de

Jens Schmitt
Distributed Computer Systems (DISCO) Lab,
TU Kaiserslautern
Kaiserslautern, Germany
jschmitt@cs.uni-kl.de

ABSTRACT

Stochastic network calculus is a versatile framework to derive probabilistic end-to-end delay bounds. Its popular subbranch using moment-generating function bounds allows for accurate bounds under the assumption of independence. However, in the dependent flow case, standard techniques typically invoke Hölder's inequality, which in many cases leads to loose bounds. Furthermore, optimization of the Hölder parameters is computationally expensive. In this work, we show that two simple, yet effective techniques related to the deterministic network calculus are able to improve the delay analysis in many scenarios, while at the same time enabling a considerably faster computation. Specifically, in a thorough numerical evaluation of two case studies, we show that using the proposed techniques: 1. we can improve the stochastic delay bounds often considerably and sometimes even obtain a bound where the standard technique provides no finite bound; 2. computation times are decreased by about two orders of magnitude.

CCS CONCEPTS

• **Networks** → **Network performance analysis**; **Network performance modeling**.

KEYWORDS

Network calculus, Moment-generating functions, Stochastic inequalties

ACM Reference Format:
Paul Nikolaus and Jens Schmitt. 2020. Improving Delay Bounds in the Stochastic Network Calculus by Using less Stochastic Inequalities. In *13th EAI International Conference on Performance Evaluation Methodologies and Tools (VALUETOOLS '20), May 18–20, 2020, Tsukuba, Japan.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3388831.3388848

1 INTRODUCTION

Stochastic network calculus (SNC) provides a flexible mathematical framework to provide probabilistic end-to-end delay bounds in packet-switched networks. It has its roots in the work on deterministic performance guarantees [15, 16] and was put forward in the following years by making use of min-plus algebra [1]. It was then extended to provide probabilistic performance bounds [11, 13, 19, 20], in particular by the seminal work of C.S. Chang [8, 9].

However, recent work [12] emphasized the bounds' lack of accuracy, 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, there is no concatenation result in the martingale-based SNC that would enable an end-to-end analysis over multiple nodes. It is well-known that the bounds' tightness is dominated by the chosen stochastic inequalities [13]. While the the approach using envelope functions [11, 17, 20] offers a wider modeling scope, e.g., heavy-tailed distributions [22], the SNC branch using moment-generating functions [9, 19] is shown to provide tighter bounds when leveraging the independence of arrival flows [26]. On the other hand, dependent arrivals, denoted by $A_1(s,t)$ and $A_2(s,t)$ with $0 \le s \le t$, are usually treated by invoking Hölder's inequality:

$$\mathrm{E}\left[e^{\theta(A_1(s,t)+A_2(s,t))}\right] \le \mathrm{E}\left[e^{p\theta A_1(s,t)}\right]^{1/p} \cdot \mathrm{E}\left[e^{q\theta A_2(s,t)}\right]^{1/q},$$

where $\theta > 0$, and $\frac{1}{p} + \frac{1}{q} = 1$, $p, q \in [1, \infty]$. Not only does this often lead to loose delay bounds, the additional parameter to optimize (one for each application) can significantly increase the computational effort. Some previous work already observed that analysis techniques that avoid Hölder's inequality tend to result in significantly tighter delay bounds [6, 18, 24, 25, 30]. One technique to mitigate the problem is to use the "paying multiplexing only once" (PMOO) principle known from deterministic network calculus [29].

In this paper, we investigate two alternative techniques: *flow prolongation* (FP) and the *maximum service output bound* (MSOB). Both techniques add pessimism to the analyzed scenario in order to avoid stochastic inequalities, most notably Hölder's inequality and the Union bound / Boole's inequality, speculating on consequently improved delay bounds.

Flow prolongation modifies the topology by extending cross-flows such that the obtained delay bounds are still rigorous. Introduced in [28] to improve computation time, [4] investigates the impact of flow prolongation on the bound's accuracy in the deterministic network calculus. While the number of improved flows varies between 57% and 92%, the maximum improvement is below 1.2%. Yet, as we show in this paper, the impact is much more significant in the stochastic network calculus, where it had not been applied before.

Flow prolongation has also been used in the context of FIFO multiplexing analysis [3] to prove that the least upper delay bound

VALUETOOLS '20, May 18–20, 2020, Tsukuba, Japan

ACM ISBN 978-1-4503-7646-4/20/05…$15.00
https://doi.org/10.1145/3388831.3388848

(LUDB) does not provide a tight worst-case delay bound. A similar technique to change the topology is used in the context of window flow controllers [2]. Yet there, the goal is to enforce subadditivity in order to obtain a converging bound.

The second technique, the maximum service output bound, mainly targets at avoiding Hölder's inequality by upper bounding the server output by its maximum service capabilities. This results in removing all the randomness from the output process and enables a more tractable analysis. It comes with the obvious disadvantage of being overly pessimistic, since we effectively assume a server to be busy at all times.

The remainder of the paper is structured as follows. In Section 2, we list existing stochastic network calculus results we need in the following. Section 3 motivates the problem and introduces our two alternative techniques. Two case studies of small, but typical analysis scenarios are investigated in Section 4. Their numerical evaluation is presented in Section 5. Section 6 concludes the paper.

2 SNC BACKGROUND AND NOTATION

We use the MGF-based SNC in order to bound the probability that the delay exceeds a given value $T \geq 0$. The MGF bound on a probability is established by applying Chernoff's bound [23]

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

We define the arrival process of a flow f by the stochastic process A with discrete time space $\mathbb{N} = \{0, 1, 2, \dots\}$ and continuous state space $\mathbb{R}^+ = [0, \infty)$ as

$$A(s,t) \coloneqq \sum_{i=s+1}^{t} a_i,$$

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. Let $x(s,t)$ and $y(s,t)$ be real-valued, bivariate functions. The *min-plus convolution* of x and y is defined as

$$x \otimes y\,(s,t) \coloneqq \inf_{s \leq \tau \leq t} \{x(s,\tau) + y(\tau,t)\}.$$

The *min-plus deconvolution* of x and y is defined as

$$x \oslash y\,(s,t) \coloneqq \sup_{0 \leq \tau \leq s} \{x(\tau,t) - y(\tau,s)\}.$$

The characteristics of the service process are captured by the notion of a dynamic S-server [9].

Definition 2.1. Assume a server has an arrival flow A as its input and the respective output is denoted by D. 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* if for all $t \geq 0$ it holds that

$$D(0,t) \geq A \otimes S\,(0,t) = \inf_{0 \leq s \leq t} \{A(0,s) + S(s,t)\}. \tag{1}$$

Definition 2.2 (Work-Conserving Server [19]). For any $t \geq 0$, let $\tau \coloneqq \sup\{s \in [0,t] : D(0,s) = A(0,s)\}$ 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 non-idling and uses the entire available service, i.e., $D(0,t) = D(0,\tau) + S(\tau,t)$.

The analysis 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 policy between flows to be arbitrary multiplexing [27].

Proposition 2.3 (Leftover Service [19]). *Consider two arrivals flows f_1 and f_2 at a work-conserving dynamic S-server with service process S. Then, the corresponding arrival A_1 sees under arbitrary multiplexing the* leftover service

$$S_{\text{l.o.}}(s,t) = [S(s,t) - A_2(s,t)]^+.$$

We define the *virtual delay* at time $t \geq 0$ as

$$d(t) \coloneqq \inf\{\tau \geq 0 : A(0,t) \leq D(0,t+\tau)\}.$$

Theorem 2.4 (Output and Delay Bound [9]). *Consider an arrival process $A(s,t)$ with dynamic S-server $S(s,t)$.*

The departure process D is upper bounded for any $0 \leq s \leq t$ according to

$$D(s,t) \leq A \oslash S\,(s,t).$$

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

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

We focus on the analogue of Theorem 2.4 for moment-generating functions:

Theorem 2.5 (MGF Delay Bound [9, 19]). *For the assumptions as in Theorem 2.4, we obtain:*

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

$$\mathrm{P}(d(t) > T) \leq \mathrm{E}\left[e^{\theta(A \oslash S\,(t+T,t))}\right], \quad \forall \theta > 0. \tag{2}$$

In order to obtain the tightest possible result, the bound in Equation (2) should be optimized in θ.

The next theorem shows how network calculus leverages min-plus algebra to derive end-to-end results.

Theorem 2.6 (End-to-End Service [19]). *Consider a flow f crossing a tandem of n work-conserving servers with service processes S_i, $i = 1, \dots, n$. Then, the overall service offered to f can be described by the* end-to-end service

$$S_{\text{e2e}}(s,t) = S_1 \otimes S_2 \otimes \cdots \otimes S_n\,(s,t).$$

In the following, we introduce (σ,ρ)-constraints that enable us to derive time-independent, stationary bounds under stability [9]. An arrival flow is (σ_A,ρ_A)-bounded for some $\theta > 0$, if for all $0 \leq s \leq t$

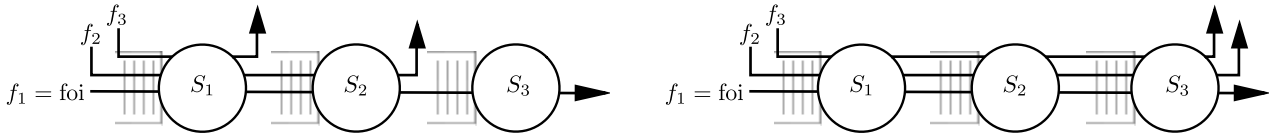
$$\mathrm{E}\left[e^{\theta A(s,t)}\right] \leq e^{\theta\rho_A(\theta)\cdot(t-s)+\theta\sigma_A(\theta)}.$$

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

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

3 THE CASE FOR USING LESS STOCHASTIC INEQUALITIES

In this section, we present our two techniques to improve SNC delay bounds: flow prolongation and the maximum service output bound. At first, however, we illustrate by means of a simplistic example how inaccuracies stemming from the use of stochastic inequalities can sometimes be easily circumvented.



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

(b) Flow prolongation leads to $S_{\text{e2e}} = [(S_1 \otimes S_2 \otimes S_3) - (A_2 + A_3)]^+$.

Figure 1: Network transformation through flow prolongation.

3.1 Inaccuracy Caused by Stochastic Inequalities

A well-known consequence of Theorem 2.6 is that, if the service processes are (σ, ρ)-bounded, so is their convolution [2]. Yet, its derivation requires the usage of stochastic inequalities and therefore causes more inaccuracy, as the following example shows.

Corollary 3.1. *Let $\theta > 0$. Assume two servers S_1 and S_2 in tandem, such that the departures of flow f_1, D_1, are the arrivals at server S_2. Further, we assume the servers to be independent dynamic S_1- and S_2-servers, respectively. Under the assumption of (σ_S, ρ_S)-constrained servers, the MGF of the end-to-end service*

$$\mathrm{E}\left[e^{-\theta S_{\text{e2e}}(s,t)}\right] = \mathrm{E}\left[e^{-\theta(S_1 \otimes S_2\,(s,t))}\right]$$

is $\left(\sigma_{S_{\text{e2e}}}, \rho_{S_{\text{e2e}}}\right)$-bounded, where

$$\begin{aligned}\sigma_{S_{\text{e2e}}}(-\theta) =& \sigma_{S_1}(-\theta) + \sigma_{S_2}(-\theta) \\ &+ \frac{1}{\theta}\log\left(\frac{1}{1 - e^{-\theta\left|\rho_{S_2}(-\theta) - \rho_{S_1}(-\theta)\right|}}\right), \\ \rho_{S_{\text{e2e}}}(-\theta) =& \min\left\{\rho_{S_1}(-\theta), \rho_{S_2}(-\theta)\right\}\end{aligned}$$

in case $\rho_{S_1}(-\theta) \neq \rho_{S_2}(-\theta)$, and

$$\begin{aligned}\sigma_{S_{\text{e2e}}}(-\theta) &= \sigma_{S_1}(-\theta) + \sigma_{S_2}(-\theta), \\ \rho_{S_{\text{e2e}}}(-\theta) &= \rho_{S_1}(-\theta) - \frac{1}{\theta}\end{aligned}$$

if $\rho_{S_1}(-\theta) = \rho_{S_2}(-\theta)$.

It provides a solution to obtain a bound on the convolution. However, it can lead to inaccuracy, as the proof requires the use of the Union bound. This issue can be illustrated with the following example.

Example 3.2. Let S_1 and S_2 be constant rate servers with rate c_1 and c_2, respectively. Then, both are (σ, ρ)-constrained with

$$\begin{aligned}\sigma_{S_i}(-\theta) &= 0, \\ \rho_{S_i}(-\theta) &= c_i.\end{aligned}$$

Since both processes are deterministic, we can perform a deterministic bivariate convolution and obtain the exact result

$$\mathrm{E}\left[e^{-\theta S_{\text{e2e}}(s,t)}\right] = e^{-\theta \min\left\{\rho_{S_1}(-\theta), \rho_{S_2}(-\theta)\right\}\cdot(t-s)}. \tag{3}$$

However, using Corollary 3.1, we either obtain

$$\begin{aligned}&\mathrm{E}\left[e^{-\theta S_{\text{e2e}}(s,t)}\right] \\ =& e^{-\theta \min\left\{\rho_{S_1}(-\theta), \rho_{S_2}(-\theta)\right\}\cdot(t-s) + \log\left(\frac{1}{1 - e^{-\theta\left|\rho_{S_2}(-\theta) - \rho_{S_1}(-\theta)\right|}}\right)},\end{aligned}$$

if $c_1 \neq c_2$, or

$$\mathrm{E}\left[e^{-\theta S_{\text{e2e}}(s,t)}\right] = e^{-\theta\left(\rho_{S_1}(-\theta) - \frac{1}{\theta}\right)\cdot(t-s)},$$

for $c_1 = c_2$. In both cases, the result is strictly worse compared to Equation (3).

3.2 Flow Prolongation

Flow prolongation transforms the network model into a more pessimistic one by extending cross-flows that interfere with the flow of interest along its path. More formally, it is defined as follows. Consider a feed-forward network $\mathcal{S}$ from the perspective of the foi, i.e., after reducing it to a tandem with servers $(1, \ldots, n)$, where 1 is the first server of the foi and n the last, respectively. Let another flow f_i share its path with the foi, i.e., it traverses the servers $(j, j+1, \ldots, j+k)$ such that $j, k \geq 1$ and $j + k < n$. Then, a flow prolongation results in a tandem network $\overline{\mathcal{S}}$ where f_i is extended to $\overline{f}_i$ such that it traverses the servers $(j, j+1, \ldots, j+k, \ldots, j+m)$, where $m > k$ and $j + m \leq n$. In other words, $\overline{f}_i$ extends its shared path with the foi. Such a flow prolongation is only performed if the servers $j+k+1, \ldots, j+m$ remain stable in $\overline{\mathcal{S}}$. Further, at the servers $j+k+1, \ldots, j+m$, we set the priorities of the prolonged flow $\overline{f}_i$ lower than for all other cross-flows, but higher than for the foi. By this, it is ensured that for all other cross-flows this prolongation is effectively transparent, whereas for the foi the scenario in $\overline{\mathcal{S}}$ has worsened compared to $\mathcal{S}$ due to the reduced service at the servers $j+k+1, \ldots, j+m$. Consequently, its *delay process* $d(t)$ is larger in $\overline{\mathcal{S}}$ than in $\mathcal{S}$ (in a stochastic ordering sense). However, the speculation is, that the *delay bound* is smaller in $\overline{\mathcal{S}}$.

The illustrative example in Figure 1 shows the result of a flow prolongation. The foi's delay in this transformed network is always at least as large as in the original system, as the interference at more hops can only worsen the situation [3]. However, the delay bound obtained by a network calculus analysis can yield an improvement. One can see in Figure 1 that it enables us to consider the aggregate of the cross-flows rather than bounding them individually. However, this obviously is only feasible as long as this transformation in Figure 1(b) does not cause the servers S_2 or S_3 to become unstable.

Compared to its deterministic counterpart, the impact of flow prolongation can be stronger in the stochastic network calculus, as computation of operations like leftover service or deconvolution requires several stochastic inequalities, i.e., it can be more beneficial to circumvent them. Assuming the S_i to be constant rate servers, we can also benefit from the efficient convolution of deterministic processes $S_1 \otimes S_2 \otimes S_3$, whereas applying the leftover operation before the convolution transforms the service processes into a random one which necessitates inaccurate stochastic inequalities

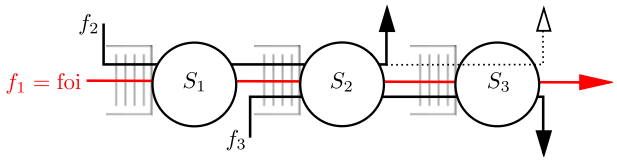


Figure 2: Overlapping tandem (dashed line indicates a flow prolongation).

(see Example 3.2). Further, as we see in the following, it can be used to avoid the usage of Hölder's inequality.

3.3 Maximum Service Output Bound

The intuitive idea behind this technique is that the output of a server is upper bounded by its maximum capacity. With the notation of a maximum dynamic $\bar{S}$-server, we can make this intuition mathematically rigorous.

Definition 3.3 (Maximum Dynamic $\bar{S}$-Server). Let $\bar{S}(s,t)$, $0 \le s \le t$, be a stochastic process that is nonnegative and increasing in t. The service element is a *maximum dynamic $\bar{S}$-server* if for all $t \ge 0$ it holds that

$$D(0,t) \le A \otimes \bar{S}(0,t).$$

In other words, it is a dynamic $\bar{S}$-server, where "$\ge$" in Equation (1) is replaced by "$\le$".

Proposition 3.4. *Assume an arrival process A and a server that is a dynamic S-server and maximum dynamic $\bar{S}$-server. Then, for $0 \le s \le t$, it holds that*

$$D(s,t) \le \bar{S} \oslash S(s,t). \tag{4}$$

We would like to remark that this bound basically provides a generalization of a similar result for greedy shapers [21]. A proof of this output bound is given in Appendix A.1.

If S and $\bar{S}$ coincide, a simple consequence follows for constant rate servers. In this case, if the server has capacity c, then the output in the interval $[s,t]$ is upper bounded by

$$D(s,t) \le c \cdot (t-s). \tag{5}$$

In contrast to the standard output bound from Theorem 2.4, $D(s,t) \le A \oslash S(s,t)$, Equation (5) has the advantage of providing a deterministic bounding process. The subsequent goal is to avoid the dependence by exploiting this determinism. However, it is obvious that this upper bound is inaccurate, the more underloaded a server is.

4 CASE STUDIES

In this section, we derive the end-to-end service using the standard approach as well as the proposed techniques and show how to use less stochastic inequalities in the analysis. For our case studies, we use two canonical networks from deterministic network calculus literature, the overlapping tandem [27], and the square network [7].

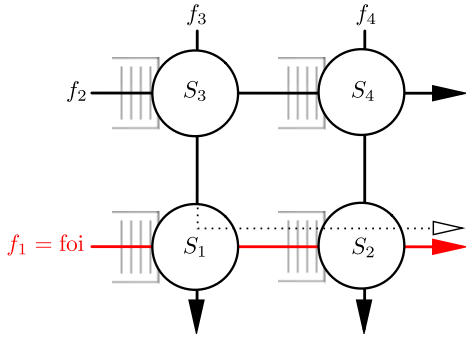


Figure 3: Square network (dashed line indicates a flow prolongation).

4.1 Overlapping Tandem

In this topology (see Figure 2), there are two options for PMOO: subtract f_3 and convolve S_1 and S_2, or subtract f_2 and apply the convolution to the last two servers. We assume the priorities $f_1 \le f_2 \le f_3$. It follows a leftover computation with the remaining cross-flow and a convolution with the residual server. We compute the bounds for both options and take the tightest. This exhaustive analysis has been shown to improve bounds in the deterministic network calculus significantly [5]. For instance, opting for convolving S_1 and S_2 first yields

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

However, we observe that A_3 appears twice which requires Hölder's inequality when applying the outer convolution.

The flow prolongation, extending f_2 to traverse S_3, yields

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

Hence, due to the prolongation, we do not have to consider dependencies anymore, as each process appears only once in the end-to-end service.

The maximum service output bound, since it is based on the PMOO analysis, also considers two cases. As for the PMOO, we discuss only the first option. Let the capacity of server S_2 be c_2. Then, we can use it to replace the output bound $A_3 \oslash S_2$:

$$S_{\text{e2e}}^{\text{MSOB}} = \left[\left(S_1 \otimes [S_2 - A_3]^+\right) - A_2\right]^+ \otimes [S_3 - c_2]^+ .$$

As for the FP, the MSOB avoids consideration of dependencies in the analysis.

4.2 Square Network

Consider the topology in Figure 3 and assume the priorities $f_1 \le \{f_3, f_4\} \le f_2$. Here, both cross-flows, f_2 and f_3 need to be subtracted so that the servers can be convolved:

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

In comparison, prolonging flow f_3 results in

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

Here, similar to the PMOO analysis, in both end-to-end service processes, arrival process A_2 appears twice.

The maximum service output, due to the appearance of two output bound calculations, needs to distinguish two cases:

$$S_{\text{e2e}}^{\text{MSOB}} = [S_1 - c_3]^+ \otimes \left[S_2 - A_4 \oslash [S_4 - (A_2 \oslash S_3)]^+\right]^+$$

and

$$S_{\text{e2e}}^{\text{MSOB}} = \left[S_1 - A_3 \oslash [S_3 - A_2]^+\right]^+ \otimes [S_2 - c_4]^+$$

which avoids the need to consider stochastic dependencies in the analysis.

5 NUMERICAL EVALUATION

For our numerical experiments, we used three different arrival classes. Let $0 \le s \le t$:

- **D/M/1** Exponentially distributed packet sizes with parameter λ:

$$\mathrm{E}\left[e^{\theta A(s,t)}\right] = \left(\frac{\lambda}{\lambda - \theta}\right)^{t-s}, \quad 0 < \theta < \lambda.$$

  As the packets arrive with constant inter-arrival times, for a single server system this would correspond to a D/M/1-queue.
- **M/D/1** Arrivals follow a Poisson process with parameter λ:

$$\mathrm{E}\left[e^{\theta A(s,t)}\right] = e^{\lambda(t-s)(e^{\theta}-1)}, \quad \theta > 0.$$

  Because of the continuous-time assumption of the Poisson process and our usage of the Union bound, it needs to be discretized in the analysis, e.g. by using techniques like in [10].
- **Continuous Markov-Modulated On-Off (MMOO)** The traffic model is governed by a continuous-time Markov chain with two states, 0 and 1, and transition rates μ and λ. In state 0, no traffic is sent, and in state 1, a constant peak rate b is sent. Its MGF is upper bounded by

$$\mathrm{E}\left[e^{\theta A(s,t)}\right] \le e^{\theta\omega(\theta)\cdot(t-s)}, \quad \theta > 0,$$

  where

$$\omega(\theta) = \frac{-d + \sqrt{d^2 + 4\mu\theta b}}{2\theta}$$

  and $d = \mu + \lambda - \theta b$ [14]. Again, we have to apply discretization in the analysis.

All these traffic classes are (σ, ρ)-bounded and therefore, they yield closed-form solutions for all network calculus operations and performance bounds. With regard to the service, we always use work-conserving constant rate servers.

Further, since the results highly depend on the chosen parameter values for the arrivals' packet sizes and service rates, we sample the parameter spaces in a Monte Carlo-type fashion. I.e., we used uniformly distributed random parameters and compared their bounds for the standard analysis with the two proposed alternatives, FP and MSOB. Since we are only interested in higher loads with significant queueing, we only considered networks with a utilization of at least 70%. This results in 6216 analyzed scenarios for the overlapping tandem and 4850 for the square network.

In all experiments, we optimize θ and the Hölder parameters numerically.

Metric	Share
Standard bound: share of finite bounds	92.6 %
MSOB: share of finite bounds	9.1 %
FP: share of finite bounds	74.3 %
MSOB: improvement over standard	7.4 %
FP: improvement over standard	63.4 %

Table 1: Numerical results for the overlapping tandem.

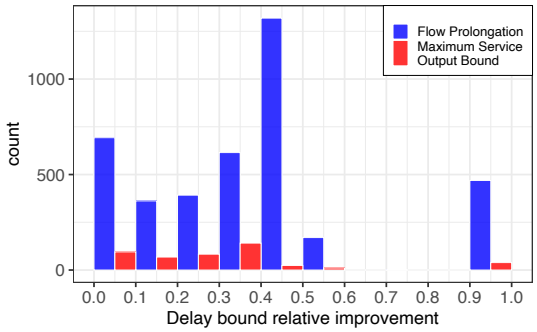


Figure 4: Histogram of the relative improvement of the stochastic delay bound (overlapping tandem).

5.1 Overlapping Tandem

For the overlapping tandem as in Figure 2, a brief summary of our numerical evaluation can be found in Table 1. The evaluation is conducted using sampled parameters for all three traffic classes.

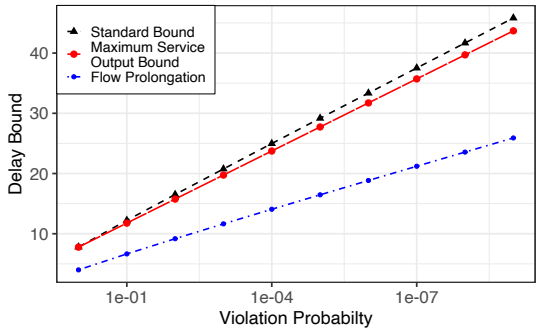
For the flow prolongation (FP), we observe that in 63.4% of the cases, it is capable of improving the standard technique. The maximum service output bound (MSOB), on the other hand, only provides a finite bound in 9.1% of the scenarios. However, 7.4% yield a tighter delay bound than the state of the art. Considering only the improved bounds, a histogram of the relative improvement, i.e.,

$$\frac{\text{Standard Bound} - \text{New Bound}}{\text{Standard Bound}} \in [0, 1)$$

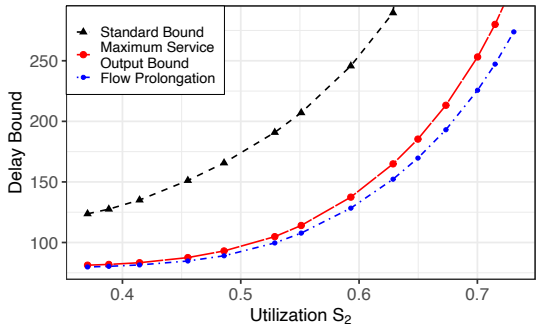
is depicted in Figure 4. If only the new technique yields a finite delay bound, the improvement is set equal to 1. Here, we observe that the relative improvement of the FP has a peak at roughly 40% and at 100%, when a finite bound is provided even though the standard technique does not.

For MMOO traffic, examples of different parameter sets are visualized in Figure 5. At first, we look at the stochastic delay bound for different violation probabilities (Figure 5(a)). The FP and the MSOB provide tighter bounds than the standard analysis. In Figure 5(b), we take a different perspective and increase the arrivals A_3, which in turn increases the utilization at S_2. Here, we see that the gap between the standard techniques and the new ones increases.

Last, but not least, run times are dominated by the number of parameters to optimize. Since both new techniques, FP and MSOB, do not have to consider any Hölder parameters to optimize, their run times are 159 and 138 times, respectively, faster on average.



(a) Utilization = 70%.



(b) Delay violation probability = 10^{-3}.

Figure 5: Stochastic delay bounds for the overlapping tandem with MMOO traffic.

Metric	Share
Standard bound: share of finite bounds	68.2 %
MSOB: share of finite bounds	58.1 %
FP: share of finite bounds	36.5 %
MSOB: improvement over standard	51.1 %
FP: improvement over standard	27.6 %

Table 2: Numerical results for the square network.

5.2 Square Network

A summary of our numerical experiments for the square network containing all three traffic classes is given in Table 2. While the flow prolongation improves the standard analysis in 27.6% of the scenarios, the maximum service output bound yields an improvement in roughly 51% of the cases. Further, if the MSOB obtains an improvement, this usually also yields the best delay bound. Looking only at the improved delay bounds, we observe that, while FP only leads to small improvements in most cases, the MSOB significantly improves the delay bound and often even provides finite delay bounds as the only approach (see Figure 6).

Again, we depicted in Figure 7 the square delay bounds for some exemplified parameter sets. For instance, if server S_3 has a high utilization while having a rather moderate capacity, the maximum service output bounds can give significantly better delay bounds

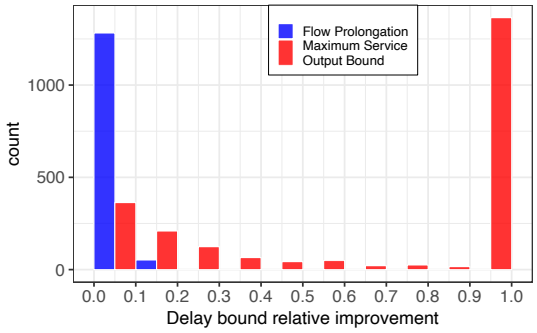
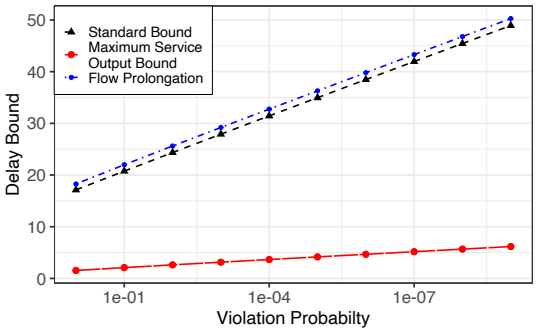
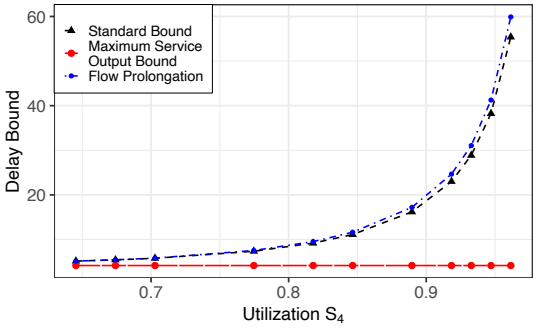


Figure 6: Histogram of the relative improvement of the stochastic delay bound (square network).



(a) Utilization = 70%.



(b) Delay violation probability = 10^{-3}.

Figure 7: Stochastic delay bounds for the square network with MMOO traffic.

(Figure 7(a)). If we increase the arrivals A_3, and therefore increase the utilization at S_3, we observe constant delay for the server bound but increasing bounds for the other techniques (Figure 7(b)).

As expected, run times of the standard technique and FP are very similar in this case. The MSOB, on the other hand, is on average roughly 75 times faster, as it avoids the consideration of dependencies, and therefore Hölder's inequality, in the analysis.

6 CONCLUSION

In this paper, we have proposed two new techniques to improve the delay bound computation in the stochastic network calculus by using less stochastic inequalities. The first, called flow prolongation, adds more pessimism by extending cross-flows to make the analysis more tractable (and consider less dependencies), The second technique, called maximum service output bound, assumes a server to run at maximum capacity. While this again leads to a more conservative analysis, it introduces more determinism that, in turn, can improve the bounding in the analysis by avoiding dependencies as well. Numerical evaluations clearly indicate that both techniques, despite being based on simple insights, can lead to largely improved stochastic delay bounds. Furthermore, run times are significantly improved when we have to invoke Hölder's inequality less frequently.

Taking into account the crucial role of dependencies in the network analysis, we believe that this paper made a significant step towards an accurate end-to-end analysis in the stochastic network calculus. However, this area still leaves room for future work; in particular, we plan to investigate the impact of both techniques on the analysis of large-scale networks.

REFERENCES

[1] François Baccelli, Guy Cohen, Geert Jan Olsder, and Jean-Pierre Quadrat. 1992. *Synchronization and linearity: an algebra for discrete event systems.* John Wiley & Sons Ltd.

[2] Michael A Beck. 2016. *Advances in Theory and Applicability of Stochastic Network Calculus.* Ph.D. Dissertation. TU Kaiserslautern.

[3] Luca Bisti, Luciano Lenzini, Enzo Mingozzi, and Giovanni Stea. 2008. Estimating the worst-case delay in FIFO tandems using network calculus. In *Proc. 3rd International Conference on Performance Evaluation Methodologies and Tools (VALUETOOLS'08).*

[4] Steffen Bondorf. 2017. Better Bounds by Worse Assumptions – Improving Network Calculus Accuracy by Adding Pessimism to the Network Model. In *Proc. IEEE Internat. Conference on Communications (ICC'17).*

[5] Steffen Bondorf, Paul Nikolaus, and Jens Schmitt. 2017. 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).*

[6] Anne Bouillard, Céline Comte, Élie de Panafieu, and Fabien Mathieu. 2018. Of Kernels and Queues: when network calculus meets analytic combinatorics. In *2018 30th International Teletraffic Congress (ITC 30).* 49–54.

[7] Anne Bouillard and Éric Thierry. 2016. Tight performance bounds in the worst-case analysis of feed-forward networks. *Discrete Event Dynamic Systems* 26, 3 (2016), 383–411.

[8] Cheng-Shang Chang. 1994. Stability, queue length, and delay of deterministic and stochastic queueing networks. *IEEE Trans. Automat. Control* 39, 5 (1994), 913–931.

[9] Cheng-Shang Chang. 2000. *Performance guarantees in communication networks.* Springer, London.

[10] Florin Ciucu. 2007. Network calculus delay bounds in queueing networks with exact solutions. In *International Teletraffic Congress (ITC 20)).* 495–506.

[11] Florin Ciucu, Almut Burchard, and Jörg Liebeherr. 2006. Scaling properties of statistical end-to-end bounds in the network calculus. *IEEE/ACM Transactions on Networking (ToN)* 14, 6 (2006), 2300–2312.

[12] Florin Ciucu, Felix Poloczek, and Jens Schmitt. 2014. 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).* 1896–1904.

[13] Florin Ciucu and Jens Schmitt. 2012. 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).* 311–322.

[14] Costas Courcoubetis and Richard Weber. 1996. Buffer overflow asymptotics for a buffer handling many traffic sources. *Journal of Applied Probability* 33 (1996), 886–903.

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

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

[17] Rene L Cruz. 1996. Quality of service management in integrated services networks. In *Proc. Semi-Annual Research Review, CWC.* 4–5.

[18] Fang Dong, Kui Wu, and Venkatesh Srinivasan. 2015. Copula analysis for statistical network calculus. In *Proc. IEEE INFOCOM'15.* 1535–1543.

[19] Markus Fidler. 2006. An end-to-end probabilistic network calculus with moment generating functions. In *Proc. IEEE International Workshop on Quality of Service (IWQoS'06).* 261–270.

[20] Yuming Jiang and Yong Liu. 2008. *Stochastic network calculus.* Vol. 1. Springer.

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

[22] Jörg Liebeherr, Almut Burchard, and Florin Ciucu. 2012. Delay bounds in communication networks with heavy-tailed and self-similar traffic. *IEEE Transactions on Information Theory* 58, 2 (2012), 1010–1024.

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

[24] Paul Nikolaus and Jens Schmitt. 2017. On Per-Flow Delay Bounds in Tandem Queues under (In)Dependent Arrivals. In *Proc. 16th IFIP Networking Conference (NETWORKING'17).* 1–9.

[25] Paul Nikolaus, Jens Schmitt, and Florin Ciucu. 2019. Dealing with Dependence in Stochastic Network Calculus – Using Independence as a Bound. Tech. Rep. 394/19, TU Kaiserslautern. https://disco.cs.uni-kl.de/discofiles/publicationsfiles/NSC19-1_TR.pdf.

[26] Amr Rizk and Markus Fidler. 2011. Leveraging statistical multiplexing gains in single-and multi-hop networks. In *Proc. IEEE International Workshop on Quality of Service (IWQoS'17).* 1–9.

[27] Jens Schmitt, Frank A Zdarsky, and Markus Fidler. 2008. Delay Bounds under Arbitrary Multiplexing: When Network Calculus Leaves You in the Lurch .... In *Proc. IEEE International Conference on Computer Communications (INFOCOM'08).* 1669–1677.

[28] Jens Schmitt, Frank A. Zdarsky, and Ivan Martinovic. 2006. *Performance Bounds in Feed-Forward Networks under Blind Multiplexing.* Technical Report. TU Kaiserslautern, Germany.

[29] Jens Schmitt, Frank A Zdarsky, and Ivan Martinovic. 2008. 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).* 1–15.

[30] Timothy Zhu, Danel S Berger, and Mor Harchol-Balter. 2016. SNC-Meister: Admitting More Tenants with Tail Latency SLOs. In *Proc. ACM Symposium on Cloud Computing (SoCC'16).* 374–387.

A APPENDIX

A.1 Proof of the Maximum Service Output Bound (MSOB)

Proof. Since $A \otimes S \leq D \leq A \otimes \bar{S}$, we obtain

$$\begin{aligned} D(s,t) \leq & D \oslash D\,(s,t) \\ &= \sup_{0\leq\tau\leq s} \{D(\tau,t) - D(\tau,s)\} \\ &\leq \sup_{0\leq\tau\leq s} \left\{A \otimes \bar{S}\,(\tau,t) - (A\otimes S)(\tau,s)\right\} \\ &= \sup_{0\leq\tau\leq s} \left\{ \inf_{\tau\leq u\leq t} \left\{A(\tau,u) + \bar{S}(u,t)\right\} \right. \\ &\qquad \left. - \inf_{\tau\leq v\leq s} \{A(\tau,v) + S(v,s)\} \right\}. \end{aligned}$$

In the first inequality, we used that $D(s,t)$ is included in

$$\sup_{0\leq\tau\leq s} \{D(\tau,t) - D(\tau,s)\}$$

for $\tau = s$. Let $v^* \in [\tau, s]$ be the solution of

$$\inf_{\tau\leq v\leq s} \{A(\tau,v) + S(v,s)\}.$$

By choosing $u \in [\tau, t] \supseteq [\tau, s]$ to be equal to v^*, we obtain the upper bound

$$\begin{aligned} D(s,t) &\leq \sup_{0\leq\tau\leq s} \left\{ \inf_{\tau\leq u\leq t} \left\{ A(\tau,u) + \bar{S}(u,t) - A(\tau,v^*) - S(v^*,s) \right\} \right\} \\ &\overset{(\text{set } u=v^*)}{\leq} \sup_{0\leq\tau\leq s} \left\{ \bar{S}(v^*,t) - S(v^*,s) \right\} \\ &= \bar{S}(v^*,t) - S(v^*,s) \\ &\leq \sup_{0\leq r\leq s} \left\{ \bar{S}(r,t) - S(r,s) \right\} = \bar{S} \oslash S\,(s,t). \end{aligned}$$

In the last inequality, we used that v^* is in $[0, s]$, therefore the difference at v^* is upper bounded by the supremum. □

Since a work-conserving constant rate server is a dynamic S-server as well as a maximum dynamic $\bar{S}$-server with

$$S(s,t) = \bar{S}(s,t) = c \cdot (t-s),$$

the proposition also yields the output bound

$$D(s,t) \leq \bar{S} \oslash S\,(s,t) = c \cdot (t-s).$$