Performance Evaluation 68 (2011) 201–219

Contents lists available at ScienceDirect

Performance Evaluation

journal homepage: www.elsevier.com/locate/peva

Dynamic demultiplexing in network calculus—Theory and application

Hao Wang, Jens B. Schmitt *, Ivan Martinovic

Distributed Computer Systems Lab (DISCO), University of Kaiserslautern, Germany

ARTICLE INFO

Article history:
Available online 13 December 2010

Keywords:
Performance bounds
Network calculus
Demultiplexing
Stochastic scaling

ABSTRACT

During the last two decades, starting with the seminal work by Cruz, network calculus has evolved as a new theory for the performance analysis of networked systems. In contrast to classical queueing theory, it deals with performance bounds instead of average values and thus has been the theoretical basis of quality of service proposals such as the IETF's Integrated and Differentiated Services architectures. Besides these it has, however, recently seen many other application scenarios as, for example, wireless sensor networks, switched Ethernets, avionic networks, Systems-on-Chip, or even to speed-up simulations, to name a few.

In this article, we extend network calculus by adding a new versatile modeling element: a *demultiplexer*. Conventionally, demultiplexing has been either neglected or assumed to be static, i.e., fixed at the setup time of a network. This is restrictive for many potential applications of network calculus. For example, a load balancing based on current link loads in a network could not be modeled with conventional network calculus means. Our demultiplexing element is based on *stochastic scaling*. Stochastic scaling allows one to put probabilistic bounds on how a flow is split inside the network. Fundamental results on network calculus with stochastic scaling are therefore derived in this work. We illustrate the benefits of the demultiplexer in a sample application of uncertain load balancing.

1. Introduction

1.1. Motivation

Network calculus is a min-plus system theory for deterministic queueing systems which builds on the calculus for network delay in [1,2]. The important concept of a service curve was introduced in [3–7]. The service curve-based approach facilitates the efficient analysis of tandem queues where a linear scaling of performance bounds in the number of traversed queues is achieved as elaborated in [8] and also referred to as pay bursts only once phenomenon in [9]. A detailed treatment of min-plus algebra and of network calculus can be found in [10,11,9], respectively.

Network calculus has found numerous applications, most prominently in the Internet's Quality of Service proposals IntServ and DiffServ, but also in other scenarios like wireless sensor networks [12,13], switched Ethernets [14], Systems-on-Chip [15], or even to speed-up simulations [16]. Hence, besides queueing theory it has established itself as a valuable methodology.

The basic network calculus methodology can only be applied to a tandem of nodes. So, the first analysis step usually consists of reducing a network scenario to a single flow over a tandem of servers. This reduction depends on the knowledge

* Corresponding author.
E-mail address: jschmitt@informatik.uni-kl.de (J.B. Schmitt).

0166-5316/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.peva.2010.12.002

about scheduling disciplines or worst-case assumptions on the multiplexing of flows. Conventionally, for demultiplexing it is assumed that flows can be demultiplexed based on static information contained in the flow (e.g., specific fields in packet headers). In fact, under this assumption demultiplexing can be neglected from a performance modeling perspective as it incurs a fixed delay with no queueing effects. However, what if demultiplexing decisions are made dynamically based on further characteristics as, for example, the load on outgoing links, or even in a random fashion (at least from the perspective of an external observer)? In many applications where one may wish to use network calculus this is the case as, for example, in any load-balancing systems or dynamic routing algorithms. In these cases conventional network calculus is restrictive and, from a modeling perspective, needs to be extended. While it is straightforward with respect to modeling, the difficulty in introducing a new explicit demultiplexing element into network calculus is in keeping up the attractive features with respect to the performance bound scaling. This is the topic of this article.

In the remainder of the article, we first provide some background on deterministic and stochastic network calculus as well as on our previous work with respect to *deterministic data scaling* in network calculus. Deterministic data scaling allows to accommodate flow transformations inside the network and, thus, enables us to model traffic splitting as performed by a demultiplexer. We show in detail how demultiplexing can be modeled even more flexibly by a stochastically generalized form of data scaling which allows one to put probabilistic bounds on the flow splitting performed by a demultiplexer. After that we delve into the detailed results we derived for the analysis of network calculus models involving stochastic scaling elements. Based on these we illustrate for a simple sample application of a load-balancing scenario how improved performance bounds can be achieved.

1.2. Related work

As demultiplexing has so far been rather neglected in network calculus, its directly related work is scarce. Interestingly, despite this observation, Cruz in his pioneering papers [1,2] originally introduced a demultiplexer as a member of his set of basic network calculus modeling elements. This demultiplexing element expectedly has a single input and multiple outputs, with the outputs carrying substreams of the input stream. For the demultiplexer's operation, it is assumed that data units are "marked" with information about their output path. This assumption essentially means that demultiplexing decisions are statically configured (e.g., at the connection set-up in a virtual circuit setting) and cannot be made dynamically as, for example, for load-balancing purposes.

In [11], Chang introduces a network calculus modeling element called a router. The router has one data input and output and one control input. The control input provides a functional relation between input and output data. This is very similar to our previous work in which we introduced a wider framework of such scaling behavior in network calculus [17]. As we will see in the course of this article, such a router or scaling element may constitute the basis for the modeling of more flexible demultiplexing which is not statically decided. The key to real flexibility with respect to dynamic demultiplexing is a stochastic bounding of the scaling behavior. This is neither provided in [11] nor [17] and represents an original contribution of this article.

1.3. Contributions

In this work, we extend network calculus concepts for scenarios involving dynamic demultiplexing decisions. In particular, we

- introduce a versatile demultiplexing element,
- derive fundamental results on stochastic data scaling,
- show an interesting connection between deterministic and stochastic network calculus,
- apply the theoretical results to analyze a load balancing network under uncertainty.

2. Preliminaries on network calculus and data scaling

In this section, we provide the necessary background on deterministic and stochastic network calculus. Furthermore, our previous work on data scaling in network calculus is reviewed as it provides the basis for the work in this article.

2.1. Deterministic network calculus

As network calculus is built around the notion of cumulative functions for input and output flows of data, the set $\mathcal{F}$ of real-valued, non-negative, and wide-sense increasing functions passing through the origin plays a major role:

$$\mathcal{F} = \left\{ f : \mathbb{R}^+ \to \mathbb{R}^+,\ \forall t \geq s : f(t) \geq f(s), f(0) = 0 \right\}.$$

In particular, the input function $F(t)$ and the output function $F'(t)$, which cumulatively count the number of bits that are input to, respectively output from, a system $\mathcal{S}$, are in $\mathcal{F}$.

There are two important min-plus resp. max-plus algebraic operators:

Definition 2.1 (*Min-plus and Max-plus Convolution and Deconvolution*)**.** The min-plus resp. max-plus convolution and deconvolution of two functions $f, g \in \mathcal{F}$ are defined to be (here $\wedge$ denotes the minimum and $\vee$ the maximum operator)

$$(f \otimes g)(t) = \inf_{0 \le s \le t} \{f(t-s) + g(s)\}, \quad (\wedge, + \text{ convolution})$$
$$(f \oslash g)(t) = \sup_{u \ge 0} \{f(t+u) - g(u)\}, \quad (\wedge, + \text{ deconvolution})$$
$$\left(f \bar{\otimes} g\right)(t) = \sup_{0 \le s \le t} \{f(t-s) + g(s)\}, \quad (\vee, + \text{ convolution})$$
$$\left(f \bar{\oslash} g\right)(t) = \inf_{u \ge 0} \{f(t+u) - g(u)\}, \quad (\vee, + \text{ deconvolution}).$$

It can be shown that the triple $(\mathcal{F}, \wedge, \otimes)$ constitutes a dioid [9]. Also, the min-plus convolution is a linear operator on the dioid $(\mathbb{R} \cup \{+\infty\}, \wedge, +)$, whereas the min-plus deconvolution is not. Similar statements can be made for max-plus systems. These algebraic characteristics result in a number of rules that apply to those operators, many of which can be found in [9,11].

Let us now turn to the performance characteristics of flows that can be bounded by network calculus means:

Definition 2.2 (*Backlog and Virtual Delay*)**.** Assume a flow with input function F that traverses a system $\mathcal{S}$ resulting in the output function F'. The *backlog* of the flow at time t is defined as

$$b(t) = F(t) - F'(t).$$

Assuming *FIFO* delivery, the *virtual delay* for a bit input at time t is defined as

$$d(t) = \inf\left\{\tau \ge 0 : F(t) \le F'(t+\tau)\right\}.$$

Next, the arrival and departure processes specified by input and output functions are bounded based on the central network calculus concepts of arrival and service curves:

Definition 2.3 (*Arrival Curve*)**.** Given a flow with input function F, a function $\alpha \in \mathcal{F}$ is an arrival curve for F iff

$$\forall t, s \ge 0, \quad s \le t : F(t) - F(t-s) \le \alpha(s) \Leftrightarrow F = F \otimes \alpha.$$

A typical example of an arrival curve is given by an affine arrival curve $\gamma_{r,b}(t) = b + rt,\ t > 0$ and $\gamma_{r,b}(t) = 0,\ t \le 0$, which corresponds to token-bucket traffic regulation.

Definition 2.4 (*Service Curve*)**.** If the service provided by a system $\mathcal{S}$ for a given input function F results in an output function F' we say that $\mathcal{S}$ offers a minimum resp. maximum service curve β resp. γ iff

$$F' \ge F \otimes \beta, \quad \text{resp.}$$
$$F' \le F \otimes \gamma.$$

A typical example of a service curve is given by a so-called rate-latency function $\beta_{R,T}(t) = R(t-T) \cdot 1_{\{t>T\}}$, where $1_{\{\text{cond}\}}$ is 1 if the condition cond is satisfied and 0 otherwise.

By using those concepts it is possible to derive tight performance bounds on backlog, delay, and output:

Theorem 2.5 (*Performance Bounds*)**.** *Consider a system $\mathcal{S}$ that offers a minimum and maximum service curve β and γ, respectively. Assume that a flow F traversing the system has an arrival curve α. Then we obtain the following performance bounds:*

backlog: $\forall t : b(t) \le (\alpha \oslash \beta)(0) =: v(\alpha, \beta)$,
delay: $\forall t : d(t) \le \inf\{t \ge 0 : (\alpha \oslash \beta)(-t) \le 0\} =: h(\alpha, \beta)$,
output (*arrival curve α' for F'*) : $\alpha' = (\alpha \otimes \gamma) \oslash \beta$.

One of the strongest results of network calculus is the concatenation theorem that enables us to investigate tandems of systems as if they were single systems:

Theorem 2.6 (*Concatenation Theorem for Tandem Systems*)**.** *Consider a flow that traverses a tandem of systems $\mathcal{S}_1$ and $\mathcal{S}_2$. Assume that $\mathcal{S}_i$ offers a minimum service curve β_i resp. a maximum service curve γ_i to the flow. Then the concatenation of the two systems offers a minimum service curve $\beta_1 \otimes \beta_2$ resp. a maximum service curve $\gamma_1 \otimes \gamma_2$ to the flow.*

Using the concatenation theorem, it is ensured that an end-to-end analysis of a tandem of servers still achieves tight performance bounds, which in general is not the case for an iterative per-node application of Theorem 2.5.

2.2. Stochastic network calculus

In recent years, many efforts towards a stochastic network calculus have been made (see, e.g., [18–23,8,24–26]). Many different definitions of stochastic extensions of arrival and service curves have been proposed and discussed. In particular, providing a stochastic service curve definition that still allows a favorable concatenation has shown to be a hard problem for some time. In this section, we simply provide the necessary definitions and basic results as they pertain to the work in this article, without delving into deep discussions on alternative definitions. Our definitions are mainly based on [22] and can be seen as direct generalizations of deterministic network calculus counterparts.

Definition 2.7 (*Stochastic Arrival Curve*)**.** Given a flow with an input function F, a function $\alpha^\epsilon \in \mathcal{F}$ is called a stochastic arrival curve for F iff $\forall t \geq 0$

$$P\left(F(t) \leq (F \otimes \alpha^\epsilon)(t)\right) \geq 1 - \epsilon.$$

Note that this definition provides a sample path bound as, for example, discussed in [22], where it is also called a sample-path effective envelope.

Definition 2.8 (*Stochastic Service Curve*)**.** If the service provided by a system $\mathcal{S}$ for a given input function F results in an output function F', we say that $\mathcal{S}$ offers a stochastic minimum resp. maximum service curve β^ϵ resp. γ^ϵ iff $\forall t \geq 0$

$$P\left(F'(t) \geq (F \otimes \beta^\epsilon)(t)\right) \geq 1 - \epsilon, \quad \text{resp.}$$
$$P\left(F'(t) \leq (F \otimes \gamma^\epsilon)(t)\right) \geq 1 - \epsilon.$$

These definitions again follow [22], where they are called statistical or effective service curves. The maximum stochastic service curve has actually not yet been introduced in stochastic network calculus, but we provide it here as it is an analogous generalization as the stochastic minimum service curve from its deterministic counterpart. Based on these definitions, the following stochastic performance bounds can be derived (see again [22] for the proof, where, however, the stochastic maximum service curve is not used to improve the output bound, yet, its integration is straightforward).

Theorem 2.9 (*Stochastic Performance Bounds*)**.** *Consider a system $\mathcal{S}$ that offers a stochastic minimum and maximum service curve β^{ϵ_β} and γ^{ϵ_γ}, respectively. Assume a flow F traversing the system has an arrival curve α^{ϵ_α}. Then we obtain the following stochastic performance bounds:*

$$\text{backlog: } \forall t : P(b(t) \leq v(\alpha^{\epsilon_\alpha}, \beta^{\epsilon_\beta})) \geq 1 - \epsilon_\alpha - \epsilon_\beta,$$
$$\text{delay: } \forall t : P(d(t) \leq h(\alpha^{\epsilon_\alpha}, \beta^{\epsilon_\beta})) \geq 1 - \epsilon_\alpha - \epsilon_\beta,$$
$$\text{output: } \alpha' = (\alpha^{\epsilon_\alpha} \otimes \gamma^{\epsilon_\gamma}) \oslash \beta^{\epsilon_\beta}$$
$$\text{with} \quad P\left(F' = F' \otimes \alpha'\right) \geq 1 - \epsilon_\alpha - \epsilon_\beta - \epsilon_\gamma.$$

It should be noted that under the stochastic service curve definition being used here, the concatenation of nodes is problematic without further assumptions. In particular, violation probabilities for concatenated service curves are time-dependent and can therefore be made equal to one, which makes the guarantees of the concatenated service curve void. Several resorts have been proposed in the literature, the most obvious being the introduction of time-scale bounds which avoids the degeneration of the service curve guarantee for large time durations. We refer the reader to a very good discussion on these issues in [22]. In this article, we stay with the straightforward definition of a stochastic service curve, which suffices for our purposes at least to begin with.

2.3. Data scaling in network calculus

In this subsection, we provide the necessary definitions and results for introducing scaling elements into network calculus models as presented in [17].

Definition 2.10 (*Scaling Function*)**.** A scaling function $S \in \mathcal{F}$ assigns an amount of scaled data $S(a)$ to an amount of data a.

As can be seen from the definition of scaling functions, they form a very general concept for taking into account transformations in a network calculus model. Note, however, that they do not model any queueing effects—scaling is assumed to be done infinitely fast. Queueing related effects are still modeled in the service curve element of the respective component.

Corollary 2.11 (*Inverse Scaling Functions*)**.** *Given a bijective scaling function $S \in \mathcal{F}$ it follows that its inverse scaling function S^{-1} is a scaling function, too.*

Inverse scaling functions play a role in transforming systems into alternative systems that can be analyzed more efficiently. More details are to follow.

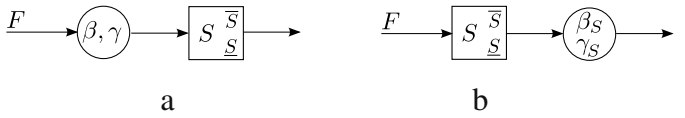


Fig. 1. Alternative systems.

Definition 2.12 (*Scaling Curves*)**.** Given a scaling function S, two functions $\underline{S}, \overline{S} \in \mathcal{F}$ are minimum and maximum scaling curves of S iff

$$\underline{S} \leq S \bar{\oslash} S,$$
$$\overline{S} \geq S \oslash S.$$

Corollary 2.13 (*Inverse Scaling Curves*)**.** *Consider a bijective scaling function S and let $\underline{S}$ and $\overline{S}$ be the respective minimum and maximum scaling curves. If $\underline{S}$ and $\overline{S}$ are bijective, a valid maximum scaling curve of the inverse scaling function $\overline{S^{-1}}$ is $\underline{S}^{-1}$ and a valid minimum scaling curve of the inverse scaling function $\underline{S^{-1}}$ is $\overline{S}^{-1}$.*

Theorem 2.14 (*Scaled Servers—Alternative Systems*)**.** *Consider the two systems in Fig. 1 and let $F(t)$ be the input function. System (a) consists of a server with minimum service curve β and maximum service curve γ whose output is scaled with a scaling function S, whereas system (b) consists of a scaling function S whose output is input to a server with minimum and maximum service curves β_S and γ_S, respectively. Given system (a), the lower and upper bounds of the output function of system (b), that are $S(F) \otimes \beta_S$ and $S(F) \otimes \gamma_S$ are also valid lower and upper bounds for the output function of system (a) if*

$$\beta_S = \underline{S}(\beta),$$

where $\overline{S}$ and $\underline{S}$ are the respective scaling curves of S. Given system (b), the lower and upper bounds for the output function of system (a), that are $S\ (F \otimes \beta)$ and $S\ (F \otimes \gamma)$ respectively, hold also for system (b) if S is bijective and

$$\beta = \underline{S^{-1}}(\beta_S),$$
$$\gamma = \overline{S^{-1}}(\gamma_S),$$

where $\overline{S^{-1}}$ and $\underline{S^{-1}}$ are the respective scaling curves of S^{-1}.

This means in effect that performance bounds for system (b) under this assumption are also valid bounds for system (a) and vice versa, as they are derived based on the bounds of the output function. This means we can effectively move a scaling function, e.g., in front of a service curve element as long as we transform the respective service curve using the minimum scaling curve of the scaling element. In [17], it is also shown that bounds computed in the alternative system, i.e., after shifting the scaling elements, remain tight. Now the utility of Corollary 2.13 becomes clear, since we can observe that it enables us to compute scaled versions of the service curves when scaling elements are shifted over service curve elements in the direction of the data stream (behind the service curve).

The following corollary states the effect scaling has on the arrival constraints of a traffic flow.

Corollary 2.15 (*Arrival Constraints Under Scaling*)**.** *Let F be an input function with arrival curve α that is fed into a scaling function S with maximum scaling curve $\overline{S}$. An arrival curve for the scaled output from the scaling element is given by*

$$\alpha_S = \overline{S}(\alpha).$$

If S is bijective and S^{-1} has a maximum scaling curve $\overline{S^{-1}}$, then given an arrival curve for the scaled output process α_S can be given as

$$\alpha = \overline{S^{-1}}(\alpha_S).$$

3. Modeling dynamic demultiplexing

In this section, we introduce a new demultiplexing element which can accommodate for dynamic demultiplexing decisions. The demultiplexer is based on the scaling element introduced in the previous section. The idea is to capture the frequent uncertainty about demultiplexing decisions within the bounds of scaling functions for each of the outputs of the demultiplexer. Sample application scenarios are provided to illustrate the versatility of the new demultiplexing element.

3.1. The demultiplexer

The demultiplexer is illustrated in Fig. 2. It is an element with one input and multiple outputs. The actual distribution of the input flow over the output flows is determined according to a vector of scaling functions $\vec{S}$, i.e., we have for each output

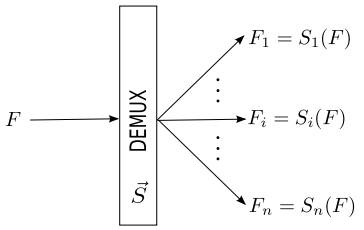


Fig. 2. The demultiplexer.

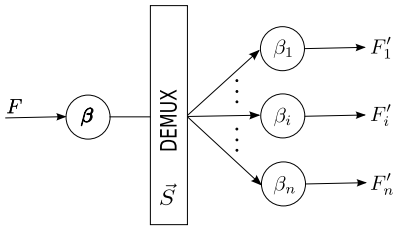


Fig. 3. A load balancing model.

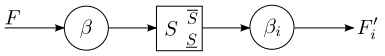


Fig. 4. Load balancing from the perspective of subflow *i*.

$i = 1, \ldots, n$ that its output function $F_i = S_i(F)$ and furthermore

$$F = \sum_{i=1}^{n} F_i = \sum_{i=1}^{n} S_i(F).$$

Hence, mathematically the demultiplexer provides the following mapping

$$\Sigma : \mathcal{F} \to \mathcal{F}^n \quad \text{with } F \to \Sigma(F) = \vec{S}(F).$$

If demultiplexing is not static, we will usually not know $\vec{S}$ explicitly, but have to work with bounds on it, the scaling curves. In many applications, as discussed in the subsequent subsections, it is also very restrictive to assume that we can bound the demultiplexer deterministically. While there are always deterministic scaling curves, they may become the trivial alternatives of $\overline{S}_i = F$ and $\underline{S}_i = 0$, which certainly results in a very pessimistic performance analysis. Hence, what is required are stochastic bounds on a demultiplexer's behavior. The required fundamental stochastic generalization of scaling in network calculus is provided in Section 4.

3.2. Application 1: load balancing

A straightforward use of the demultiplexer is the case of load balancing over a number of servers as illustrated in Fig. 3. Very often load balancing either makes randomized decisions, i.e., scheduling the next work unit on a server from the pool based on a random distribution, or demultiplexing is based on some state which we can only describe stochastically. The uncertainty about load balancing decisions may be due to data dependent switching decisions, information hiding by a network provider, or simply because of the impracticality of obtaining all the necessary switching information. In any case, the only sensible option to characterize the scaling of each of the outputs and thus demultiplexing is via stochastic bounds. Examples for load-balancing systems that fall in this category are abundant (see, e.g., [27–30] for recent systems from the networking and parallel computing domain).

In Fig. 4, we also provide a view on the load-balancing model from the perspective of subflow *i*. This shows, provided that we can setup similar results to deterministic data scaling for the stochastic setting, that an end-to-end network calculus analysis is again possible for each subflow. Note that for each subflow the whole input flow needs to be taken into account as it drives the scaling element of the subflows ($\to S_i(F)$).

3.3. Application 2: lossy links

A less obvious application of the demultiplexer is in networks with lossy links, a simple tandem scenario is shown in Fig. 5. The ground symbol here means that data units are lost. Even more obviously here than in the load-balancing case, a deterministic bound on the scaling behavior represents a mismatch and will only result in trivial scaling curves for the outputs. In particular, a deterministic bound would mean that all data units are lost, thus not providing any non-trivial

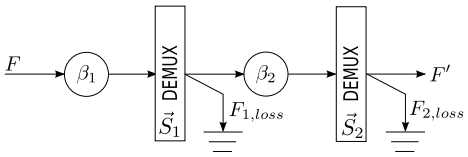


Fig. 5. A tandem network with lossy links.

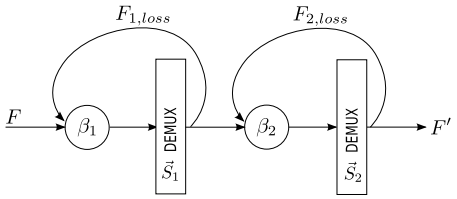


Fig. 6. A tandem network with lossy links and retransmissions.

insight into the system's performance. A stochastic generalization of scaling curves is inevitable for a system consisting of lossy links.

Under the assumption that each link uses retransmissions to cater for data loss we can modify the model according to Fig. 6.

4. Stochastic data scaling

In this section, we provide fundamental results on *stochastic* data scaling in network calculus. These can be seen as a generalization of the results for deterministic scaling in [17]. As discussed, stochastic scaling constitutes the basis for our new demultiplexing element.

We will introduce stochastic versions of minimum and maximum scaling curves. For some results, the deterministic arguments can be easily adapted, whereas for others some care needs to be taken. As in the deterministic setting, it is convenient and not too restrictive to assume bijectivity of scaling functions. However, we first need to define the notion of a stochastic scaling process of which scaling functions are its realizations.

Definition 4.1 (*Stochastic Scaling*)**.** A stochastic scaling element $\mathcal{S}$ is described as a stochastic process whose ensemble (all possible sample paths or realizations of $\mathcal{S}$) is a set of scaling functions $\{S(a,\omega) : \omega \in I_{\mathcal{S}}\}$, where $I_{\mathcal{S}}$ is an index set.

Next, we provide definitions for stochastically relaxed versions of maximum and minimum scaling curves.

Definition 4.2 (*Stochastic Scaling Curve*)**.** Let $\mathcal{S}$ be a stochastic scaling element, two functions $\underline{S}_{\underline{\epsilon}}, \overline{S}^{\overline{\epsilon}} \in \mathcal{F}$ are called stochastic minimum and maximum scaling curves of $\mathcal{S}$, respectively, iff

$$P\left(\underline{S}_{\underline{\epsilon}} \leq \mathcal{S} \bar{\oslash} \mathcal{S}\right) \geq 1 - \underline{\epsilon},$$
$$P\left(\overline{S}^{\overline{\epsilon}} \geq \mathcal{S} \oslash \mathcal{S}\right) \geq 1 - \overline{\epsilon}.$$

Here, $\underline{\epsilon}$ and $\overline{\epsilon}$ denote the violation probabilities for stochastic minimum and maximum scaling curves, respectively.

Note that the stochastic scaling curve properties are defined over sample paths (scaling functions) as realized by the respective scaling process. In the context of stochastic arrival curves this has also been coined as sample-path effective (see Section 2.2).

In the deterministic case, Corollary 2.13 provides a way to calculate scaling curves for the inverse scaling function based on knowledge of the scaling curves for the original scaling function. We can directly transfer these results under the following definitions:

Definition 4.3 (*Bijectivity of Stochastic Scaling Elements*)**.** A stochastic scaling element $\mathcal{S}$ is said to be bijective if $\forall \omega \in I_{\mathcal{S}}$ the respective scaling function $S(a,\omega)$ is bijective.

Definition 4.4 (*Inverse Stochastic Scaling Element*)**.** Given a stochastic scaling element $\mathcal{S}$ with ensemble $\{S(a,\omega) : \omega \in I_{\mathcal{S}}\}$ we define its inverse stochastic scaling element $\mathcal{S}^{-1}$ by the ensemble $\left\{S^{-1}(a,\omega) : \omega \in I_{\mathcal{S}}\right\}$. Here, it is assumed that $\mathcal{S}$ and $\mathcal{S}^{-1}$ are realized together, based on the same random experiment, such that it is justified to state that $\mathcal{S}^{-1} \circ \mathcal{S} = \mathcal{S} \circ \mathcal{S}^{-1} = \mathrm{id}$.

Now we turn to the presumably most important result, the stochastic generalization of the alternative systems theorem.

Theorem 4.5 (*Stochastically Scaled Servers—Alternative Systems*)**.** *Consider the two systems in Fig. 7 and let F be the input function. System (a) consists of a server with a deterministic minimum service curve β and maximum service curve γ whose*

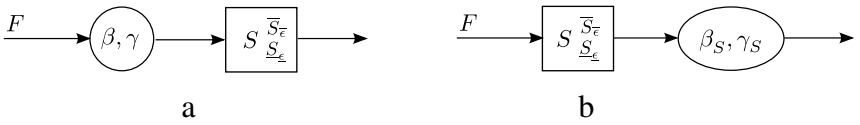


Fig. 7. Alternative systems under stochastic setting.

output is scaled by a stochastic scaling element $\mathcal{S}$, for which we have stochastic *minimum and maximum scaling curves $\underline{S}_{\underline{\epsilon}}$ and $\overline{S}^{\overline{\epsilon}}$ with violation probability $\underline{\epsilon}$ and $\overline{\epsilon}$, respectively. System (b) consists of a stochastic scaling element $\mathcal{S}$, which has stochastic minimum and maximum scaling curves $\underline{S}_{\underline{\epsilon}}$ and $\overline{S}^{\overline{\epsilon}}$ with violation probability $\underline{\epsilon}$ and $\overline{\epsilon}$, and whose output is input to a server with deterministic minimum and maximum service curves $\beta_{\mathcal{S}}$ and $\gamma_{\mathcal{S}}$, respectively.*

(1) *Given system (a) the lower and upper bounds of the output function of system (b), that are $\mathcal{S}(F) \otimes \beta_{\mathcal{S}}$ and $\mathcal{S}(F) \otimes \gamma_{\mathcal{S}}$, respectively, are also valid* stochastic *lower and upper bounds for the output function of system (a), i.e., $\forall t \geq 0$:*

$$P(\mathcal{S}(F'(t)) \geq (\mathcal{S}(F) \otimes \beta_{\mathcal{S}})(t)) \geq 1 - \underline{\epsilon},$$
$$P(\mathcal{S}(F'(t)) \leq (\mathcal{S}(F) \otimes \gamma_{\mathcal{S}})(t)) \geq 1 - \overline{\epsilon},$$

if system (b) consists of the same stochastic scaling element as in system (a) together with service curves

$$\beta_{\mathcal{S}} = \underline{S}_{\underline{\epsilon}}(\beta),$$
$$\gamma_{\mathcal{S}} = \overline{S}^{\overline{\epsilon}}(\gamma).$$

(2) *Assume bijectivity of $\mathcal{S}$. Given system (b) the lower and upper bounds of the output function of system (a), that are $\mathcal{S}(F \otimes \beta)$ and $\mathcal{S}(F \otimes \gamma)$, respectively, are also valid* stochastic *lower and upper bounds for system (b), i.e., $\forall t \geq 0$:*

$$P((\mathcal{S}(F(t)))' \geq (\mathcal{S}(F \otimes \beta))(t)) \geq 1 - \underline{\epsilon},$$
$$P((\mathcal{S}(F(t)))' \leq (\mathcal{S}(F \otimes \gamma))(t)) \geq 1 - \overline{\epsilon},$$

if system (a) consists of the same stochastic scaling element as in system (b) together with service curves

$$\beta = \underline{S^{-1}}_{\underline{\epsilon}}(\beta_{\mathcal{S}}),$$
$$\gamma = \overline{S^{-1}}^{\overline{\epsilon}}(\gamma_{\mathcal{S}}).$$

Here $\underline{S^{-1}}_{\underline{\epsilon}}$ and $\overline{S^{-1}}^{\overline{\epsilon}}$ are the respective stochastic scaling curves of inverse stochastic scaling $\mathcal{S}^{-1}$.

Proof. Some preliminary remarks: Note that a stochastic scaling curve $\underline{S}_{\underline{\epsilon}}$ only bounds a subset of all possible scaling functions from the stochastic scaling element $\mathcal{S}$. Let us denote that subset by

$$\mathcal{S}_{\underline{S}_{\underline{\epsilon}}\text{applies}} = \left\{ S(a, \omega) : \omega \in I_{\mathcal{S}}, \underline{S}_{\underline{\epsilon}} \leq S \bar{\oslash} S \right\} \subseteq \mathcal{S}.$$

Note that $P(\mathcal{S}_{\underline{S}_{\underline{\epsilon}}\text{applies}}) \geq 1 - \underline{\epsilon}$, as well as that $\underline{S}_{\underline{\epsilon}}$ is a deterministic minimum scaling curve for each scaling function $S(a, \omega) \in \mathcal{S}_{\underline{S}_{\underline{\epsilon}}\text{applies}}$.

We now start proving the first part of the theorem (going from system (a) to system (b)). We begin with the stochastic lower bound on the output function of the composite system.

For system (a), we know from the minimum service curve property that

$$F' \geq F \otimes \beta.$$

Assuming that the stochastic scaling element realizes a scaling function $S(a, \omega) \in \mathcal{S}_{\underline{S}_{\underline{\epsilon}}\text{applies}}$, we can deterministically conclude that

$$\begin{aligned}
S(F'(t)) &\geq S(F \otimes \beta)(t) \\
&= S\left(\inf_{0 \leq s \leq t} \{F(t-s) + \beta(s)\} \right) \\
&= \inf_{0 \leq s \leq t} \{S(F(t-s) + \beta(s))\} \\
&= \inf_{0 \leq s \leq t} \{S(F(t-s)) + S(F(t-s) + \beta(s)) - S(F(t-s))\} \\
&\geq \inf_{0 \leq s \leq t} \left\{ \mathcal{S}(F(t-s)) + \underline{S}_{\underline{\epsilon}}(\beta(s)) \right\} \\
&= \left(\mathcal{S}(F) \otimes \underline{S}_{\underline{\epsilon}}(\beta) \right)(t).
\end{aligned}$$

Now consider system (b). Its output function can be bounded as follows:

$$(S(F(t)))' \geq (S(F) \otimes \beta_{\mathcal{S}})(t).$$

If we let $\beta_{\mathcal{S}} = \underline{S}_{\underline{\epsilon}}(\beta)$ in system (b), we get the same bound on the output function based on the assumption that $S(a,\omega) \in \mathcal{S}_{\underline{S}_{\underline{\epsilon}}\text{applies}}$, which holds with probability $1-\underline{\epsilon}$. Hence, we obtain $\forall t \geq 0$:

$$P(\mathcal{S}(F'(t)) \geq (\mathcal{S}(F) \otimes \beta_{\mathcal{S}})(t)) \geq 1-\underline{\epsilon}.$$

Establishing the connection between the upper bound on the output functions of systems (a) and (b) follows as an immediate variation.

Now for the second part of the theorem: going from system (b) to system (a). Again, we start with the lower bound on the output function of the composite system. Similar to above, we introduce the subset of scaling functions for which their inverse adheres to $\underline{S^{-1}}_{\underline{\epsilon}}$ as

$$\mathcal{S}_{\underline{S^{-1}}_{\underline{\epsilon}}\text{applies}} = \left\{ S(a,\omega) : \omega \in I_{\mathcal{S}}, \underline{S^{-1}}_{\underline{\epsilon}} \leq S^{-1} \bar{\oslash} S^{-1} \right\} \subseteq \mathcal{S}.$$

For system (b), we know from the deterministic minimum service curve property that

$$(S(F))' \geq S(F) \otimes \beta_{\mathcal{S}}.$$

Assuming that the stochastic scaling element realizes a scaling function $S(a,\omega) \in \mathcal{S}_{\underline{S^{-1}}_{\underline{\epsilon}}\text{applies}}$, we can deterministically conclude that (since each inverse scaling function is wide-sense increasing)

$$\begin{aligned}
S^{-1}((S(F(t)))') &\geq S^{-1}\left((S(F) \otimes \beta_{\mathcal{S}})(t)\right) \\
&= S^{-1}\left(\inf_{0\leq s\leq t}\{S(F(s)) + \beta_{\mathcal{S}}(t-s)\}\right) \\
&= \inf_{0\leq s\leq t}\left\{S^{-1}(S(F(s)) + \beta_{\mathcal{S}}(t-s))\right\} \\
&= \inf_{0\leq s\leq t}\left\{S^{-1}(S(F(s))) + S^{-1}(S(F(s)) + \beta_{\mathcal{S}}(t-s)) - S^{-1}(S(F(s)))\right\} \\
&\geq \inf_{0\leq s\leq t}\left\{\mathcal{S}^{-1}(\mathcal{S}(F(s))) + \underline{S^{-1}}_{\bar{\epsilon}}(\beta_{\mathcal{S}}(t-s))\right\} \\
&= \left(F \otimes \underline{S^{-1}}_{\bar{\epsilon}}(\beta_{\mathcal{S}})\right)(t).
\end{aligned}$$

Since $S \circ S^{-1} = \mathrm{id}$, we can conclude from the above inequality that

$$(\mathcal{S}(F(t)))' \geq \mathcal{S}\left(\left(F \otimes \underline{S^{-1}}_{\bar{\epsilon}}(\beta_{\mathcal{S}})\right)(t)\right).$$

Now consider system (a). Its output function can be bounded as follows:

$$S(F'(t)) \geq S(F \otimes \beta)(t).$$

If we let $\beta = \underline{S^{-1}}_{\bar{\epsilon}}(\beta_{\mathcal{S}})$ in system (a), we get the same bound on the output function based on the assumption that $S(a,\omega) \in \mathcal{S}_{\underline{S^{-1}}_{\underline{\epsilon}}\text{applies}}$, which holds with probability $1-\underline{\epsilon}$. Hence, we obtain $\forall t \geq 0$:

$$P\left((\mathcal{S}(F(t)))' \geq (\mathcal{S}(F \otimes \beta))(t)\right) \geq 1-\underline{\epsilon}.$$

Establishing the connection between the upper bound on the output functions of systems (a) and (b) follows as an immediate variation. □

As in the deterministic case this alternative system theorem allows moving the scaling elements over service curve elements in the stochastic setting. These enable an efficient end-to-end analysis using the concatenation theorem as much as possible.

In the following corollary, we state the effect that stochastic scaling elements have on the arrival constraints of an input function.

Corollary 4.6 (*Arrival Constraints Under Stochastic Scaling*)**.** *Let F be an input function with arrival curve α that is fed into a stochastic scaling element $\mathcal{S}$ with stochastic maximum scaling curve $\overline{S}^{\bar{\epsilon}}$.*

A stochastic arrival curve for the scaled output from the scaling element is given by

$$\alpha_{S,\bar{\epsilon}} = \overline{S}^{\bar{\epsilon}}(\alpha),$$

i.e., it applies that $\forall t \geq 0$:

$$P\left(\mathcal{S}(F(t)) \leq \left(\mathcal{S}(F) \otimes \alpha_{S,\bar{\epsilon}}\right)(t)\right) \geq 1-\bar{\epsilon}.$$

If $\mathcal{S}^{-1}$ has a maximum scaling curve $\overline{S^{-1}}^{\bar{\epsilon}}$, and given an arrival curve for the scaled output process α_S, a stochastic arrival

curve for the input can be given as

$$\alpha = \overline{S^{-1}}^{\overline{\epsilon}}(\alpha_S),$$

i.e., it applies that

$$P\left(F \leq F \otimes \alpha\right) \geq 1 - \overline{\epsilon}.$$

Proof. From the stochastic maximum scaling curve property we have

$$P\left(\overline{S}^{\overline{\epsilon}} \geq \mathcal{s} \oslash \mathcal{s}\right) = P\left(\forall a, b \geq 0 : \overline{S}^{\overline{\epsilon}}(a) \geq \mathcal{s}\left(a+b\right) - \mathcal{s}(b)\right) \geq 1 - \overline{\epsilon}.$$

Setting $a = F(t) - F(t-s)$ and $b = F(t-s)$, we obtain

$$P\left(\forall t, s \geq 0 : \overline{S}^{\overline{\epsilon}}\left(F(t) - F(t-s)\right) \geq \mathcal{s}(F(t)) - \mathcal{s}(F(t-s))\right) \geq 1 - \overline{\epsilon}.$$

Now using that α is an arrival curve of F and that $\overline{S}^{\overline{\epsilon}}$ is wide-sense increasing, we obtain $\forall t \geq 0$:

$$\begin{aligned} P\left(\forall s \geq 0 : \overline{S}^{\overline{\epsilon}}\left(\alpha(s)\right) \geq \mathcal{s}(F(t)) - \mathcal{s}(F(t-s))\right) &= P\left(S(F(t)) \leq \left(S \otimes \alpha_S\right)(t)\right) \\ &\geq 1 - \overline{\epsilon}, \end{aligned}$$

which establishes α_S as a stochastic arrival curve of the scaled output function $\mathcal{s}(F)$ from the scaling element.

Now for the second part of the corollary: clearly, we have $\forall t, s \geq 0$ that

$$F(t+s) - F(s) = \mathcal{s}^{-1}(\mathcal{s}(F(t+s))) - \mathcal{s}^{-1}(\mathcal{s}(F(s))). \tag{1}$$

From the stochastic maximum scaling curve property we know that

$$P\left(\mathcal{s}^{-1}(\mathcal{s}(F(t+s))) - \mathcal{s}^{-1}(\mathcal{s}(F(s))) \leq \overline{S^{-1}}^{\overline{\epsilon}}\left(\mathcal{s}(F(t+s)) - \mathcal{s}(F(s))\right)\right) \geq 1 - \overline{\epsilon}.$$

Using the arrival curve α_S we are given for the scaled output, Eq. (1) and that $\overline{\mathcal{s}^{-1}}^{\overline{\epsilon}}$ is wide-sense increasing, we obtain $\forall t \geq 0$ that

$$P\left(F(t+s) - F(s) \leq \overline{S^{-1}}^{\overline{\epsilon}}(\alpha_S(t))\right) \geq 1 - \overline{\epsilon},$$

which establishes $\alpha = \overline{S^{-1}}^{\overline{\epsilon}}(\alpha_S)$ as a stochastic arrival curve of the input function F to the stochastic scaling element. □

Note that under stochastic scaling, again, the resulting stochastic arrival curves are sample-path bounds.

5. Sample application: delay bounds under uncertain load balancing

The point of this section is to illustrate the application of the new demultiplexing element together with the results from the previous section in the case of a load-balancing system. In particular, we assume that we have an outsider view on the network and do not know how the switching decisions are made at each of the demultiplexing points. So, we try to find stochastic delay bounds under uncertainty about the load balancing decisions made inside the network.

For ease of exposition, the example calculations are kept as simple as possible without loss of generality, wherever possible. First, we compare in a simple scenario different analysis alternatives for determining delay bounds under uncertain load-balancing based on: deterministic scaling, stochastic scaling with a node-by-node analysis, and stochastic scaling with an end-to-end analysis. As we shall see, care needs to be taken on how to actually use the theoretical results derived in the previous sections. In fact, we demonstrate that achieving the best possible stochastic delay bound with a given violation probability requires us to solve a non-trivial optimization problem. For larger scenarios, solving the respective optimization problem is outside the scope of this article, yet we provide fallback options to achieve approximate results and illustrate the increasing benefit of using a stochastic instead of a deterministic analysis when the number of nodes grows.

5.1. Scenario and preliminaries

The network scenario we assume is depicted in Fig. 8. We have aggregate flows that enter a network at a certain ingress point and while traversing the network are demultiplexed inside the network, thus resulting in many potential egress flows. From the perspective of a single aggregate flow, the network looks like a tree as shown also in Fig. 8.

The demultiplexing or, more concretely, the load-balancing decisions are assumed not to be under the control of the flow, or more concretely, the corresponding user. These decisions are either made by the provider of the network or are just random processes, for example, depending on the contents of the data units. Formally, the demultiplexing decisions can be modeled as the ratios W_{ij} of the incoming traffic at node i that is forwarded to node j, where W_{ij} are independent random

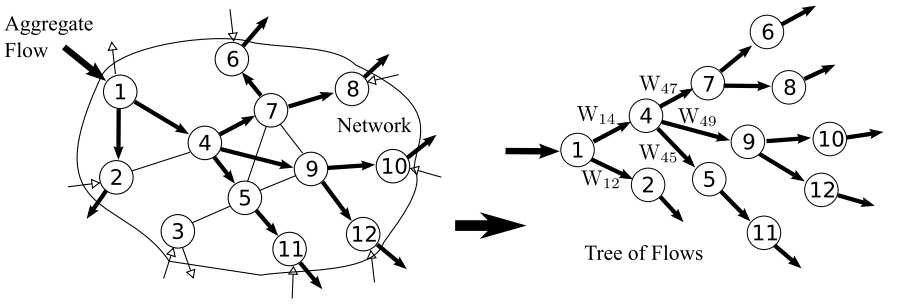


Fig. 8. Network scenario of load-balancing.

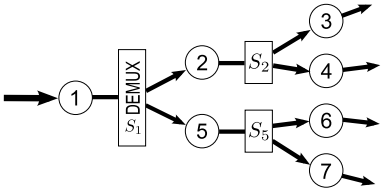


Fig. 9. Full binary tree.

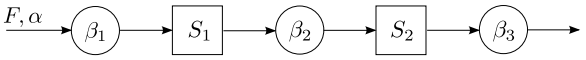


Fig. 10. Subflow from full binary tree.

variables with support [0, 1] and it applies that $\sum_j W_{ij} = 1$. The user, however, still wants to be able to compute delay bounds for all of its sub-flows (corresponding to different egress points from the network) even under uncertainty about the W_{ij}, i.e., the load-balancing decisions. Different options to model uncertainty may be assumed. Here, we mostly assume complete uncertainty. This is best modeled with a uniform distribution for W_{ij}, i.e., $W_{ij} \sim U(0, 1)$. Yet, we also demonstrate the case of assuming more knowledge by modeling W_{ij} by a triangular distribution, e.g. $W_{ij} \sim \text{Triangle}\left(0, \frac{1}{\#\text{children}(i)}, 1\right)$. The latter may correspond, e.g., to knowing that the provider aims to achieve an equal load distribution over all outgoing links for the respective flow at a demultiplexing point but can only imperfectly achieve it.

For ease of exposition, but actually mostly without loss of generality, we further constrain our discussion on fully occupied binary trees for the flow under analysis as depicted in Fig. 9. This has the nice side effect that each of the sub-flows faces the same situation with respect to the delay analysis. Hence, it does not matter which one we pick and we can actually focus on a scenario as depicted in Fig. 10. Furthermore, for the fully occupied binary tree it is clear that selecting all $W_{ij} = 0.5$ would be optimal with respect to minimizing the worst-case delay over all sub-flows. While this selection is not under our control (W_{ij} are assumed to be random variables), it can still serve as an idealistic reference to assess our alternative analysis methods below.

From now on we define the actual elements from the concrete scenario in Fig. 10. The stochastic scaling elements are defined as $\mathcal{S}_i(a) = W_i \cdot a = \gamma_{W_i,0},\ i = 1, 2$ with $W_i \sim U(0, 1)$ (later also $W_i \sim \text{Triangle}\left(0, \frac{1}{2}, 1\right)$), representing the uncertain load-balancing decisions. Under these assumptions stochastic scaling curves can be easily worked out as

$$\underline{S_i}^{\underline{\epsilon_i}} = \gamma_{\underline{\epsilon},0} \qquad \overline{S_i}^{\overline{\epsilon_i}} = \gamma_{1-\overline{\epsilon_i},0}.$$

We show this exemplarily for the maximum scaling curve under the slightly more general assumption of a general cumulative distribution function F_{W_i} for W_i, such that $\overline{S_i}^{\overline{\epsilon_i}} = \gamma_{F^{-1}_{W_i}(1-\overline{\epsilon_i}),0}$. We then have

$$\begin{aligned} P\left(\overline{S_i}^{\overline{\epsilon_i}} \geq \mathcal{S}_i \oslash \mathcal{S}_i\right) &= P\left(\forall a, b \geq 0 : \overline{S_i}^{\overline{\epsilon_i}}(a) \geq \mathcal{S}_i(a+b) - \mathcal{S}_i(b)\right) \\ &= P\left(\forall a, b \geq 0 : F^{-1}_{W_i}(1-\overline{\epsilon_i}) \cdot a \geq W_i \cdot (a+b) - W_i \cdot (b)\right) \\ &= P\left(F^{-1}_{W_i}(1-\overline{\epsilon_i}) \geq W_i\right) \\ &\geq 1 - \overline{\epsilon_i}, \end{aligned}$$

which establishes $\gamma_{F^{-1}_{W_i}(1-\overline{\epsilon_i}),0}$ as a stochastic maximum scaling curve for $\mathcal{S}_i$ with violation probability $\overline{\epsilon_i}$.

For the service curves we assume that they are rate-latency functions, i.e., $\beta_i = \beta_{R_i,T_i},\ i = 1,2,3$. Here, we set $R_1 = 10\left(\frac{\text{kp}}{\text{s}}\right)$, $R_2 = 7\left(\frac{\text{kp}}{\text{s}}\right)$, $R_3 = 4\left(\frac{\text{kp}}{\text{s}}\right)$ and $T_i = 0.01(s),\ i = 1,2,3$. The arrival curve of the input flow is assumed to be given as a token bucket, i.e., $\alpha = \gamma_{r,b}$ with $r = 4\left(\frac{\text{kp}}{\text{s}}\right)$ and $b = 0.8$ (kp). For simplicity, we assumed unit packets in these settings.

5.2. Comparison of alternative analyses

We now compare different alternatives available for the performance analysis of the uncertain load-balancing scenario. If nothing else is mentioned, we assume all the violation probabilities $\underline{\epsilon_i} = \overline{\epsilon_i} = 0.1,\ i = 1,2$.

5.2.1. Idealistic analysis

In this analysis, which we provide for reference purposes, we assume that we exactly and deterministically know the scaling functions to be $S_1(F) = S_2(F) = 0.5F$, and thus their scaling curves $\underline{S_i}(a) = \overline{S_i}(a) = 0.5a,\ i = 1,2$. In this case, we can invoke (trivially) Theorem 2.14 (in which direction to move the scaler does not matter here; so, assume we move them to the front) and Corollary 2.15 to obtain a worst-case delay bound as (further applying Theorems 2.5 and 2.6)

$$d^{\text{ideal}} \leq h\left(\overline{S_2}\left(\overline{S_1}(\alpha)\right), \underline{S_2}\left(\underline{S_1}(\beta_1)\right) \otimes \underline{S_2}(\beta_2) \otimes \beta_3\right) = 110\ (\text{ms}).$$

Clearly, none of the following analysis alternatives can go below this value as we now make more realistic assumptions on the knowledge we have on demultiplexing or, more concretely, the load-balancing process. d^{ideal} serves as a target value for the following alternatives.

5.2.2. Deterministic scaling

First, we ignore the stochastic extension of scaling we provided in this article and show what results can be achieved based on purely deterministic scaling bounds.

Based solely on the assumption of $\mathcal{S}_1, \mathcal{S}_2$ being $U(0,1)$, the only deterministic bounds we can put on scaling are that $\underline{S_1} = \underline{S_2} = 0$ and $\overline{S_1} = \overline{S_2} = \text{id}$. In this case we can, in fact, ignore scaling and calculate the following worst-case delay bound (using Theorems 2.5 and 2.6)

$$d^{\text{det}} \leq h\left(\alpha, \beta_1 \otimes \beta_2 \otimes \beta_3\right) = 230\ (\text{ms}).$$

Certainly, this bound is very pessimistic, because it essentially assumes that the entire traffic goes to a single egress point in the network. Note that deterministic scaling becomes excessively pessimistic if the scenario becomes larger, i.e., if more nodes are involved (see Section 5.4). Thus, due to the small three node scenario assumed here, it is still relatively competitive.

5.2.3. Stochastic scaling with node-by-node analysis

Next, we use stochastic bounds on the scaling behavior of the demultiplexers. For the purpose of illustrating its merits, we ignore the stochastically generalized alternative system theorem, Theorem 4.5, for a moment, and perform a simple node-by-node (*nbn*) analysis. In this case, we can calculate a stochastic delay bound as follows (applying Corollary 4.6 and Theorem 2.5)

$$\begin{aligned} d^{\text{stoch,nbn}} &\leq h(\alpha, \beta_1) + h\left(\overline{S_1}^{\epsilon_1}(\alpha \oslash \beta_1), \beta_2\right) + h\left(\overline{S_2}^{\epsilon_2}\left(\overline{S_1}^{\epsilon_1}(\alpha \oslash \beta_1) \oslash \beta_2\right), \beta_3\right) \\ &\approx 398\ (\text{ms}). \end{aligned}$$

As we can observe, this stochastic delay bound is even worse than the deterministic ones calculated in the previous subsection. This illustrates how important it is to invoke the concatenation theorem to provide for good end-to-end performance bounds. Yet, this is not possible without moving the scaling elements.

5.2.4. Stochastic scaling with end-to-end analysis

Now, we use all our "weapons" and provide an end-to-end analysis under stochastic scaling bounds. Here, using Theorem 4.5 and shifting scaling elements conveniently we are able to leverage again from the concatenation theorem. We have two options, moving the scalers to the ingress or the egress of the overall system. As we will see, how we do it is of a great importance.

Move scaling elements to ingress. For the case of moving the scaling element to the front, we obtain the following stochastic bound on delay (applying Theorem 4.5 and Corollary 4.6 as well Theorems 2.5 and 2.6)

$$\begin{aligned} d^{\text{stoch,e2e,ingr}} &\leq h\left(\overline{S_2}^{\epsilon_2}\left(\overline{S_1}^{\epsilon_1}(\alpha)\right), \underline{S_2}_{\epsilon_2}\left(\underline{S_1}_{\epsilon_1}(\beta_1)\right) \otimes \underline{S_2}_{\epsilon_2}(\beta_2) \otimes \beta_3\right) \\ &\approx 6510\ (\text{ms}). \end{aligned}$$

This is clearly very disappointing because it is far off from any sensible value. So, some things go badly wrong here: First of all, we can improve the situation by concatenating scaling elements that are neighboring during the process of being

moved. In this example, we can first move $\mathcal{S}_1$ such that it is right beside $\mathcal{S}_2$, and we can treat them as one scaling element realized by their concatenation $\mathcal{S}_1 \circ \mathcal{S}_2$. Of course, the scaling curves then need to be derived based on the concatenated scaling element. In our case,

$$(\mathcal{S}_1 \circ \mathcal{S}_2)(a) = \mathcal{S}_1(\mathcal{S}_2(a)) = W_1 \cdot W_2 \cdot a = W_2 \cdot W_1 \cdot a = \mathcal{S}_2(\mathcal{S}_1(a)) = (\mathcal{S}_2 \circ \mathcal{S}_1)(a).$$

We denote the stochastic scaling curves for the concatenated stochastic scaling as $\overline{S_1 \circ S_2}^{\overline{\epsilon_{12}}}$ and $\underline{S_1 \circ S_2}_{\underline{\epsilon_{12}}}$ as well as the according violation probabilities $\overline{\epsilon_{12}}$ and $\underline{\epsilon_{12}}$, respectively. As we know $W_1, W_2 \sim U(0, 1)$, after concatenation the probability density function of the random variable $\overline{W_1} \cdot W_2$ is that of a product of two independent uniform random variables, which can be derived as $\ln\left(\frac{1}{x}\right), x \in (0, 1]$ [31]. Let $\overline{\epsilon_{12}} = \underline{\epsilon_{12}} = 0.1$ and we obtain as stochastic delay bound:

$$\begin{aligned} d^{\text{stoch,e2e,ingr}} &\leq h\left(\overline{S_1 \circ S_2}^{\overline{\epsilon_{12}}}(\alpha), \underline{S_1 \circ S_2}_{\underline{\epsilon_{12}}}(\beta_1) \otimes \underline{S_2}_{\underline{\epsilon_2}}(\beta_2) \otimes \beta_3\right) \\ &\approx 2390 \text{ (ms)}. \end{aligned}$$

While the result is considerably better, it is still far from a sensible value, as it is ≈10 times as large as the deterministic bound. The reason for this lies in the movement of the scaling elements to the ingress. This necessitates invoking both the maximum and minimum scaling curves in the computation of the delay bound. Seen from a sample path perspective, it is however clear that you cannot realize both of them together simultaneously. Since they are pretty far from each other, we observe that while the input is scaled down only insignificantly, the servers are scaled down considerably, and thus a very untight (useless) delay bound results. So, the next attempt is to move the scaling elements to the egress of the network.

Move scaling elements to egress. Now we move the scaling elements to the egress, behind the servers, effectively scaling them up. We obtain (applying Theorem 4.5 and Corollary 2.13 as well Theorems 2.5 and 2.6)

$$\begin{aligned} d^{\text{stoch,e2e,egr}} &\leq h\left(\alpha, \beta_1 \otimes \underline{S_1^{-1}}_{\overline{\epsilon_1}}(\beta_2) \otimes \underline{(S_1 \circ S_2)^{-1}}_{\overline{\epsilon_{12}}}(\beta_3)\right) \\ &= h\left(\alpha, \beta_1 \otimes \left(\overline{S_1^{\epsilon_1}}\right)^{-1}(\beta_2) \otimes \left(\overline{(S_1 \circ S_2)^{\epsilon_{12}}}\right)^{-1}(\beta_3)\right) \\ &\approx 148 \text{ (ms)}. \end{aligned}$$

Hence, we eventually found the right way to exploit the stochastic scaling results. On the one hand, this computation allows taking advantage of the pay bursts only once phenomenon due to an end-to-end analysis as well as it leverages from the concatenation of scaling elements while being moved. On the other hand, it avoids the simultaneous usage of the maximum and minimum service curve, which, as discussed in the previous results, in gross delay bounds in our scenario.

We are now considerably below the deterministic delay bound of 230 ms and fairly close to the idealistic delay bound of 110 ms. If more knowledge on the load-balancing decisions is provided we can obtain even better bounds. So, let us now assume $W_i \sim \text{Triangle}\left(0, \frac{1}{2}, 1\right)$, then moving the scaling elements to the egress and concatenating while moving we obtain:

$$d^{\text{stoch,e2e,egr,triang}} \leq 122 \text{ (ms)}.$$

As can be perceived, the latter especially constitutes a very considerable reduction and gets actually quite close to the result from the idealistic analysis.

For the sake of completeness, the triangular p.d.f. is given as

$$\text{Triangle}(x|a, m, b) = \begin{cases} \dfrac{2(x-a)}{(b-a)(m-a)} & x \in [a, m] \\ \dfrac{2(b-x)}{(b-a)(b-m)} & x \in (m, b]. \end{cases}$$

The p.d.f. of the product of two random variables from $\text{Triangle}\left(0, \frac{1}{2}, 1\right)$ can be computed [32] as

$$h(x) = \begin{cases} 16\left(2x + 2x\ln\left(\dfrac{1}{2}\right) + x\ln\left(\dfrac{1}{4x}\right)\right) & x \in \left[0, \dfrac{1}{4}\right] \\ 16(2 - 6x + 2x\ln(2x) + (1+x)\ln(4x)) & x \in \left(\dfrac{1}{4}, \dfrac{1}{2}\right] \\ 16\left(2(x-1) + (1+x)\ln\left(\dfrac{1}{x}\right)\right) & x \in \left(\dfrac{1}{2}, 1\right]. \end{cases}$$

5.2.5. Remarks on violation probabilities

While in the previous subsections we showed how to work out different stochastic delay bounds, we suppressed, for ease of exposition, the discussion on their violation probabilities. In fact, the calculation of the violation probabilities depends on the way we perform the delay analysis. Clearly, deterministic and idealistic analysis involve zero violation probability as they follow deterministic arguments. For the other cases, a simple way to obtain a violation probability is to invoke Boole's inequality, which is true under all stochastic assumptions. However, using special properties such as independence or positive correlations between scaling elements we may be able to improve violation probabilities. In the following we go through some of the analysis methods presented above and try to make use of the special stochastic assumption in our example scenario:

Stochastic node-by-node analysis. Recall that for the stochastic node-by-node analysis, we worked out the delay bound as

$$d^{\text{stoch,nbn}} \leq h(\alpha, \beta_1) + h\left(\overline{S_1^{\epsilon_1}}(\alpha \oslash \beta_1), \beta_2\right) + h\left(\overline{S_2^{\epsilon_2}}\left(\overline{S_1^{\epsilon_1}}(\alpha \oslash \beta_1) \oslash \beta_2\right), \beta_3\right).$$

The probability for the bound to apply can be calculated as (using independence between the scaling elements)

$$\begin{aligned} P\left(\left\{\overline{S_1^{\epsilon_1}} \text{ applies}\right\} \cap \left\{\left\{\overline{S_1^{\epsilon_1}} \text{ applies}\right\} \cap \left\{\overline{S_2^{\epsilon_2}} \text{ applies}\right\}\right\}\right) &= P\left(\left\{\overline{S_1^{\epsilon_1}} \text{ applies}\right\} \cap \left\{\overline{S_2^{\epsilon_2}} \text{ applies}\right\}\right) \\ &\geq (1-\overline{\epsilon_1}) \cdot (1-\overline{\epsilon_2}) \\ &= 0.81. \end{aligned}$$

Thus, we obtain a violation probability of 0.19. Had we used Boole's inequality, it would be 0.2 (apply it to the complementary probability in the second line above).

Stochastic end-to-end analysis. As moving the scaling elements to the ingress is inferior to moving to the egress, we focus on the calculation of the violation probability of the latter here. Recall that when moving the scaling elements to the egress and using concatenated scaling the delay bound is determined as

$$\begin{aligned} d^{\text{stoch,e2e,egr}} &\leq h\left(\alpha, \beta_1 \otimes \underline{S_1^{-1}}_{\overline{\epsilon_1}}(\beta_2) \otimes \underline{(S_1 \circ S_2)^{-1}}_{\overline{\epsilon_{12}}}(\beta_3)\right) \\ &= h\left(\alpha, \beta_1 \otimes \left(\overline{S_1^{\epsilon_1}}\right)^{-1}(\beta_2) \otimes \left(\overline{(S_1 \circ S_2)^{\epsilon_{12}}}\right)^{-1}(\beta_3)\right). \end{aligned}$$

Assuming $\overline{\epsilon_{12}} = 0.1$ and as we know in the uniform random case that $W_1 \cdot W_2 \sim \ln\left(\frac{1}{x}\right), x \in [0, 1]$, we determine $\overline{(S_1 \circ S_2)^{\epsilon_{12}}}(a) = z_2 \cdot a$ by

$$\int_0^{z_2} \ln\left(\frac{1}{x}\right) \mathrm{d}x = 1 - \overline{\epsilon_{12}} = 0.9 \Rightarrow z_2 \approx 0.6.$$

Hence, the above delay bound holds with the following probability

$$\begin{aligned} P\left(\{W_1 \leq 1 - \overline{\epsilon_1}\} \cap \{W_1 \cdot W_2 \leq z_2\}\right) &= P\left(W_1 \leq 0.9\right) \cdot P\left(W_1 \cdot W_2 \leq 0.6 \mid W_1 \leq 0.9\right) \\ &= 0.9 \cdot \left(1 - \left(\int_{0.6}^{0.9} \left(1 - \frac{0.6}{x}\right) \mathrm{d}x\right) / 0.9\right) \\ &= 0.843, \end{aligned}$$

where the conditional probability can be calculated as an unconditional probability for the product of a $U(0, 1)$ with a $U(0, 0.9)$ random variable. Thus, the violation probability of 0.157 compares favorably against one based on Boole's inequality of 0.2.

5.3. Optimal delay bounds

In the previous section we derived stochastic delay bounds by fixing the violation probabilities at each of the scaling elements, and as a result obtained some violation probabilities for the delay bounds. Presumably, the more typical problem setting is, however, the following: given an overall violation probability, one wants to find the best stochastic delay bound for a given system. This problem of finding optimal stochastic delay bounds is addressed in this section. The problem can alternatively be viewed as how the violation probability should be distributed over the scaling elements or, equivalently, which scaling curves should be used at each of the stochastic scaling elements. We treat this problem again for the scenario of three nodes and two scaling elements, as described in the previous section and illustrated in Fig. 10. In particular, we assume the stochastic scaling elements again as $\mathcal{S}_i = \gamma_{W_i,0}$, with $W_i \sim U(0, 1)\ i = 1, 2$.

As shown in the previous section it is best to move the stochastic scaling elements to the egress while making use of their concatenation. We introduce two decision variables z_1, z_2, which govern how the maximum scaling curves at each of the scaling elements, $\mathcal{S}_1$ and $\mathcal{S}_1 \circ \mathcal{S}_2$ are set, i.e., $\overline{S_1^{\epsilon_1}} = \gamma_{z_1,0}$ and $\overline{S_1 \circ S_2^{\epsilon_{12}}} = \gamma_{z_2,0}$.

Under these prerequisites, we can provide the general expression for the violation probability as

$$V(z_1, z_2) = 1 - P(W_1 \leq z_1) P(W_1 W_2 \leq z_2 | W_1 \leq z_1).$$

Now, if $z_1 \geq z_2$ and with $W_1' \sim U(0, z_1)$, we can calculate

$$\begin{aligned} P(W_1 W_2 \leq z_2 | W_1 \leq z_1) &= P(W_1' W_2 \leq z_2) \\ &= 1 - \frac{1}{z_1} \int_{z_1}^{z2} 1 - \frac{z_2}{W_1'} \mathrm{d}W_1' \\ &= \frac{z_2}{z_1} - \frac{z_2}{z_1} \ln\left(\frac{z_2}{z_1}\right). \end{aligned}$$

Thus, we can compute the violation probability as

$$V(z_1, z_2) = 1 - z_2 + z_2 \ln\left(\frac{z_2}{z_1}\right).$$

On the other hand, if $z_1 < z_2$:

$$P(W_1 W_2 \leq z_2 | W_1 \leq z_1) = 1,$$

and thus the violation probability can be computed as

$$V(z_1, z_2) = 1 - z_1.$$

So, altogether, we obtain the general expression in z_1 and z_2 violation probability as

$$V(z_1, z_2) = \begin{cases} 1 - z_1 & z_1 < z_2 \\ 1 - z_2 + z_2 \ln\left(\frac{z_2}{z_1}\right) & z_1 \geq z_2. \end{cases}$$

Next, we provide the general expression for the end-to-end delay bound:

$$d^{\text{stoch,e2e,egr}}(z_1, z_2) = \frac{b}{\min\left\{R_1, \frac{R_2}{z_1}, \frac{R_3}{z_2}\right\}} + T_1 + T_2 + T_3.$$

Now we can set up the optimization problem as

$$\begin{aligned} &\text{max. } d^{\text{stoch,e2e,egr}}(z_1, z_2) \\ &\text{s.t.} \\ &V(z_1, z_2) \leq \epsilon \\ &z_1, z_2 \leq 1. \end{aligned}$$

In fact, the structure of this problem is not simple at first sight as it involves a non-linear objective function as well as non-linear constraint. In a first step, we simplify the problem in order to better understand it by transforming it into the following equivalent problem

$$\begin{aligned} &\text{max. } \min\left\{R_1, \frac{R_2}{z_1}, \frac{R_3}{z_2}\right\} \\ &\text{s.t.} \\ &V(z_1, z_2) \leq \epsilon \\ &0 \leq z_1, z_2 \leq 1. \end{aligned}$$

Noting that $V(z_1, z_2)$ is a wide-sense decreasing function in z_1 and z_2, we can observe that the objective function is actually about balancing the scaled server rates $\frac{R_2}{z_1}$ and $\frac{R_3}{z_2}$ as much as possible. So let us first assume that it is actually possible, under the constraints given, that $\frac{R_2}{z_1} = \frac{R_3}{z_2}$ or, equivalently, $z_2 = \frac{R_3}{R_2} z_1$. So we can use this to reformulate our problem as

$$\begin{aligned} &\text{max. } \frac{R_2}{z_1} \\ &\text{s.t.} \\ &\epsilon \geq \begin{cases} 1 - z_1 & R_3 > R_2 \\ 1 - \frac{R_3}{R_2} z_1 + \frac{R_3}{R_2} z_1 \ln\left(\frac{R_3}{R_2}\right) & R_3 \leq R_2 \end{cases} \\ &z_1 \leq 1. \end{aligned}$$

The solution to this problem can easily be found and thus also the solution to the original problem, under the assumption that the balancing of the scaled server rates is possible, can be computed as:

$$\begin{aligned} &R_3 > R_2 : z_1^* = 1 - \epsilon, \qquad z_2^* = \frac{R_3}{R_2} z_1^*, \\ &R_3 \leq R_2 : z_1^* = \frac{1 - \epsilon}{\frac{R_3}{R_2}\left(1 - \ln\left(\frac{R_3}{R_2}\right)\right)}, \qquad z_2^* = \frac{R_3}{R_2} z_1^*. \end{aligned} \tag{2}$$

Table 1
Optimal delay bounds for some violation probabilities.

Violation probability ϵ	0.5	0.1	0.01
Optimal delay bound $d^{\text{stoch,e2e,egr}}$	110	159	203

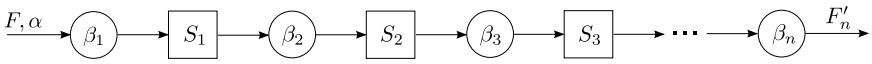


Fig. 11. N nodes subflow.

If we obtain a feasible solution, i.e., $0 \leq z_1^*, z_2^* \leq 1$, then we are done. If, however, one of the decision variables is out of its bounds then this indicates that the assumption of a possible balancing of the scaled server rates is, in fact, invalid. In this case, one of the original server rates is too small to "catch up" with the other, even if scaled up. In this case it is best to scale up the lower rate server as much as possible, meaning that we use deterministic scaling at the other server, e.g., if $R_3 < R_2$ then we set $z_1 = 1$ and make z_2 as small as possible. More precisely: Given that Eq. (2) only provides an infeasible solution, the optimal solution can be computed as:

$$R_3 < R_2 : z_1^* = 1, \qquad z_2^* = \text{solution of } 1 - z_2 + z_2 \ln(z_2) = \epsilon,$$
$$R_3 > R_2 : z_1^* = 1 - \epsilon, \qquad z_2^* = 1.$$

Clearly, for the first case we require a numerical solution. Nevertheless, this concludes our quest for the optimal stochastic delay bound under a given violation probability.

For example, using the same parameter settings as in Section 5.2 and further setting $\epsilon = 0.157$, which was the violation probability worked out there for a delay bound of 148 ms, we now obtain for the optimal delay bound:

$$z_1^* = 0.946, \qquad z_2^* = 0.541,$$
$$d^{\text{stoch,e2e,egr}} = 138 \text{ (ms)}.$$

Hence, we can improve the stochastic delay bound, at the same violation probability, by 10 ms simply by choosing the scaling curves right. For illustrative purposes, Table 1 provides some further numerical examples with different violation probabilities and corresponding optimal delay bounds. Remarkably, the median delay bound ($\epsilon = 0.5$) equals the delay bound under idealistic assumptions. In fact, this is already the case at $\epsilon = 0.376$.

5.4. Larger scenarios

In the previous sections we always worked on the small example scenario involving three nodes for the subflow under investigation. In this section, we now provide some insight into the scaling of the delay bounds when an increasing number of nodes is traversed by the subflow and thereby also demonstrate how to apply our results in more general settings. Still following the assumption that the load balancing network is a fully occupied binary tree, we depict the n nodes subflow scenario in Fig. 11. Applying the previous analyses to determine delay bounds for a subflow with n nodes, we obtain:

IDEALISTIC:

$$d^{\text{ideal}} \leq h\left(\alpha, \beta_1 \otimes 2\beta_2 \otimes 2^2\beta_3 \otimes \cdots \otimes 2^{n-1}\beta_n\right).$$

DETERMINISTIC:

$$d^{\text{det}} \leq h(\alpha, \beta_1 \otimes \beta_2 \otimes \cdots \otimes \beta_n).$$

STOCHASTIC NODE-BY-NODE:

$$d^{\text{stoch,nbn}} \leq h(\alpha, \beta_1) + h\left(\overline{S}_1^{\overline{\epsilon}_1}(\alpha \oslash \beta_1), \beta_2\right) + \cdots + h\left(\overline{S}_{n-1}^{\overline{\epsilon}_{n-1}}\left(\overline{S}_{n-2}^{\overline{\epsilon}_{n-2}}\left(\cdots\left(\overline{S}_1^{\overline{\epsilon}_1}(\alpha \oslash \beta_1) \oslash \beta_2\right)\cdots\right) \oslash \beta_{n-1}\right), \beta_n\right).$$

STOCHASTIC END-TO-END:

1. Move Stochastic Scaling to ingress
   (a) nested scalings

$$d^{\text{stoch,e2e}} \leq h\Bigg(\overline{S}_{n-1}^{\overline{\epsilon}_{n-1}}\left(\overline{S}_{n-2}^{\overline{\epsilon}_{n-2}}\left(\cdots\left(\overline{S}_1^{\overline{\epsilon}_1}(\alpha)\right)\cdots\right)\right), \underline{S_{n-1}}_{\epsilon_{n-1}}\left(\underline{S_{n-2}}_{\epsilon_{n-2}}\left(\cdots\left(\underline{S_1}_{\epsilon_1}(\beta_1)\right)\cdots\right)\right)$$
$$\times \otimes \underline{S_{n-1}}_{\epsilon_{n-1}}\left(\underline{S_{n-2}}_{\epsilon_{n-2}}\left(\cdots\left(\underline{S_2}_{\epsilon_2}(\beta_2)\right)\cdots\right)\right) \otimes \cdots \otimes \underline{S_{n-1}}_{\epsilon_{n-1}}(\beta_{n-1}) \otimes \beta_n\Bigg)$$

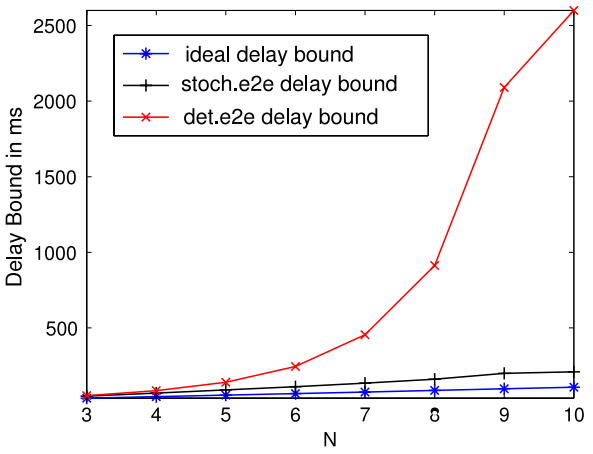


Fig. 12. Delay bounds for scenarios from 3 to 10 nodes.

(b) concatenated scaling

$$d^{\text{stoch,e2e}} \leq h\Big(\overline{S_1 \circ S_2 \circ \cdots \circ S_{n-1}}^{\overline{\epsilon_{12\cdots n-1}}}(\alpha),\ \beta_n \otimes \underline{S_{n-1}}_{\underline{\epsilon_{n-1}}}(\beta_{n-1}) \times \otimes \underline{S_{n-2} \circ S_{n-1}}_{\underline{\epsilon_{n-2,n-1}}}(\beta_{n-2}) \otimes \cdots \otimes \underline{S_1 \circ S_2 \circ \cdots \circ S_{n-1}}_{\underline{\epsilon_{12\cdots n-1}}}(\beta_1)\Big).$$

2. Move Stochastic Scaling to egress

(a) nested scalings.

As this analysis does not benefit from the concatenation of scalings, the formulations are not given here. But it is easy to get them.

(b) concatenated scaling

$$d^{\text{stoch,e2e}} \leq h\Big(\alpha,\ \beta_1 \otimes \underline{S_1^{-1}}_{\overline{\epsilon_1}}(\beta_2) \otimes \underline{(S_1 \circ S_2)^{-1}}_{\overline{\epsilon_{12}}}(\beta_3) \otimes \cdots \otimes \underline{(S_1 \circ S_2 \circ \cdots \circ S_{n-1})^{-1}}_{\overline{\epsilon_{12\cdots n-1}}}(\beta_n)\Big)$$

with probability

if independent $\geq (1-\overline{\epsilon}_1)(1-\overline{\epsilon_{12}})\cdots(1-\overline{\epsilon_{12\cdots n-1}})$;

with Boole's inequality $\geq 1 - \sum \overline{\epsilon_{12\cdots n-1}}$.

Since the better way to calculate stochastic delay bounds is to pay burst only once and avoid simultaneously using minimum and maximum scaling curves, as shown in the previous sections, we lay our focus on the branch "move to egress and concatenated scaling". Thus, we calculated the delay bound with this formulation for scenarios with 3 to 10 nodes and compare the results to the idealistic and deterministic analysis. Again, for the service curves we assume that they are rate-latency functions, i.e., $\beta_i = \beta_{R_i,T_i},\ i = 1, 2, \ldots, 10$. Here, we set $\left(\text{following the rule that } R_i = \frac{R_{i-1}}{2} - 1, i = 2, \ldots, 9\right)$: $R_1 = 1790\ \left(\frac{\text{kp}}{\text{s}}\right), R_2 = 894\ \left(\frac{\text{kp}}{\text{s}}\right), R_3 = 446\ \left(\frac{\text{kp}}{\text{s}}\right), R_4 = 222\ \left(\frac{\text{kp}}{\text{s}}\right), R_5 = 110\ \left(\frac{\text{kp}}{\text{s}}\right), R_6 = 54\ \left(\frac{\text{kp}}{\text{s}}\right), R_7 = 26\ \left(\frac{\text{kp}}{\text{s}}\right), R_8 = 12\ \left(\frac{\text{kp}}{\text{s}}\right), R_9 = 5\ \left(\frac{\text{kp}}{\text{s}}\right)$, and $R_{10} = 4\ \left(\frac{\text{kp}}{\text{s}}\right)$ as well as $T_i = 0.01$ (s), $i = 1, 2, \ldots, 10$. The arrival curve of the input flow is assumed to be given as a token bucket, i.e., $\alpha = \gamma_{r,b}$ with $r = 4\ \left(\frac{\text{kp}}{\text{s}}\right)$ and $b = 10$ (kp). For simplicity, we again assumed unit packets in these settings. Further, during this calculation we assume that all the violation probabilities are $\overline{\epsilon_{1\ldots i}} = \underline{\epsilon_{1\ldots i}} = 0.01,\ i = 1, \ldots, 10$. As above, the scaling elements are represented by straight lines through the origin with a random slope drawn from a uniform distribution $U \sim (0, 1)$. Then the concatenated scaling elements slope distributions can be calculated using the p.d.f. of the product of i independent uniform random variables [31]

$$f_{W_{12\cdots i}}(x) = \frac{\left(\ln\left(\frac{1}{x}\right)\right)^{i-1}}{(i-1)!}, \quad x \in (0, 1].$$

The results are depicted in Fig. 12. As we can see, the deterministic analysis results in a super-linear growth of the delay bounds when the number of nodes increases. On the other hand, the results from the stochastic end-to-end analysis always stay near to the idealistic analysis. Calculating the probability of the stochastic end-to-end delay bound to apply using Boole's inequality results in: $\geq$0.97, $\geq$0.96, $\geq$0.95, $\geq$0.94, $\geq$0.93, $\geq$0.92, $\geq$0.91, $\geq$0.90, for each of the scenarios. Using independence results in slightly higher probabilities.

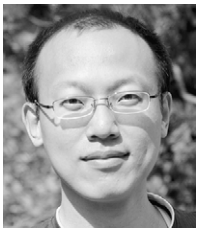
6. Conclusion

Our goal in this article was to extend the scope of network calculus to scenarios involving a non-trivial demultiplexing of data flows. To that end, we introduced a versatile demultiplexing element, which is based on stochastic data scaling. At the

heart of the article is the development of this stochastic data scaling in the network calculus framework, trying to conserve as much as possible the system-theoretic elegance of network calculus. With the aid of a sample application, bounding delay in a load-balancing network under uncertainty, we illustrated the potential benefits of using our new demultiplexing element together with the fundamental results on stochastic scaling. The application also illustrated that the new results must be carefully applied in order to become effective. In fact, the application again raised some interesting theoretical questions, like the computation of optimal delay bounds or best possible violation probabilities, thus opening avenues for future research along these lines. More practically, we are also about to integrate the new demultiplexing element into our DISCO Network Calculator toolbox [33][1] and further want to explore other interesting application scenarios.

References

[1] R.L. Cruz, A calculus for network delay, part I: network elements in isolation, IEEE Transactions on Information Theory 37 (1991) 114–131.
[2] R.L. Cruz, A calculus for network delay, part II: network analysis, IEEE Transactions on Information Theory 37 (1991) 132–141.
[3] R. Agrawal, R.L. Cruz, C. Okino, R. Rajan, Performance bounds for flow control protocols, IEEE/ACM Transactions on Networking 7 (1999) 310–323.
[4] C.-S. Chang, On deterministic traffic regulation and service guarantees: a systematic approach by filtering, IEEE Transactions on Information Theory 44 (1998) 1097–1110.
[5] R.L. Cruz, Quality of service guarantees in virtual circuit switched networks, IEEE Journal on Selected Areas in Communications 13 (1995) 1048–1056.
[6] J.-Y. Le Boudec, Application of network calculus to guaranteed service networks, IEEE Transactions on Information Theory 44 (1998) 1087–1096.
[7] H. Sariowan, R.L. Cruz, G.C. Polyzos, Scheduling for quality of service guarantees via service curves, in: Proc. IEEE ICCCN, pp. 512–520.
[8] F. Ciucu, A. Burchard, J. Liebeherr, A network service curve approach for the stochastic analysis of networks, in: Proc. ACM SIGMETRICS, pp. 279–290.
[9] J.-Y. Le Boudec, P. Thiran, Network Calculus: A Theory of Deterministic Queuing Systems for the Internet, in: Lecture Notes in Computer Science, vol. 2050, Springer-Verlag, Berlin, Germany, 2001.
[10] F. Baccelli, G. Cohen, G.J. Olsder, J.-P. Quadrat, Synchronization and Linearity: An Algebra for Discrete Event Systems, in: Probability and Mathematical Statistics, John Wiley & Sons Ltd., 1992.
[11] C.-S. Chang, Performance Guarantees in Communication Networks, in: Telecommunication Networks and Computer Systems, Springer-Verlag, 2000.
[12] A. Koubaa, M. Alves, E. Tovar, Modeling and worst-case dimensioning of cluster-tree wireless sensor networks, in: Proc. 27th IEEE International Real-Time Systems Symposium, RTSS'06, IEEE Computer Society, Rio de Janeiro, Brazil, 2006, pp. 412–421.
[13] J. Schmitt, U. Roedig, Sensor network calculus—a framework for worst case analysis, in: Proc. Distributed Computing on Sensor Systems, DCOSS, pp. 141–154.
[14] T. Skeie, S. Johannessen, O. Holmeide, Timeliness of real-time IP communication in switched industrial ethernet networks, IEEE Transactions on Industrial Informatics 2 (2006) 25–39.
[15] S. Chakraborty, S. Kuenzli, L. Thiele, A. Herkersdorf, P. Sagmeister, Performance evaluation of network processor architectures: combining simulation with analytical estimation, Computer Networks 42 (2003) 641–665.
[16] H. Kim, J. Hou, Network calculus based simulation: theorems, implementation, and evaluation, in: Proc. IEEE INFOCOM.
[17] M. Fidler, J. Schmitt, On the way to a distributed systems calculus: an end-to-end network calculus with data scaling, in: ACM SIGMETRICS/Performance 2006, SIGMETRICS'06, ACM, St. Malo, France, 2006, pp. 287–298.
[18] C.-S. Chang, Stability, queue length and delay of deterministic and stochastic queueing networks, IEEE Transactions on Automatic Control 39 (1994) 913–931.
[19] O. Yaron, M. Sidi, Performance and stability of communication networks via robust exponential bounds, IEEE/ACM Transactions on Networking 1 (1993) 372–385.
[20] R. Cruz, Quality of service management in integrated services networks, in: Proceedings of the 1st Semi-Annual Research Review, Center for Wireless Communications, UCSD.
[21] D. Starobinski, M. Sidi, Stochastically bounded burstiness for communication networks, IEEE Transactions on Information Theory 46 (2000) 206–212.
[22] A. Burchard, J. Liebeherr, S.D. Patek, A min-plus calculus for end-to-end statistical service guarantees, IEEE Transactions on Information Theory 52 (2006) 4105–4114.
[23] S. Ayyorgun, W.-C. Feng, A systematic approach for providing end-to-end probabilistic QoS guarantees, in: Proceedings of the 13th IEEE International Conference on Computer Communications and Networks, ICCCN, Chicago, IL. Also available as Technical Reports LA-UR-03-3668 and LA-UR-03-7267, Los Alamos National Laboratory.
[24] M. Fidler, An end-to-end probabilistic network calculus with moment generating functions, in: Proc. of IEEE IWQoS, pp. 261–270.
[25] Y. Jiang, A basic stochastic network calculus, ACM Computer Communication Review 36 (2006) 123–134. Proc. ACM SIGCOMM 2006.
[26] J.-Y. Le Boudec, M. Vojnović, Elements of probabilistic network calculus for packet scale rate guarantee nodes, in: Proc. of 15th Int'l Symp. of Mathematical Theory of Networks and Systems.
[27] I. Keslassy, C.-S. Chang, N. McKeown, D.-S. Lee, Optimal load-balancing, in: Proc. IEEE INFOCOM.
[28] R. Zhang-Shen, N. McKeown, Designing a predictable Internet backbone network, in: Proc. HotNets III.
[29] L.G. Valiant, A scheme for fast parallel communication, SIAM Journal on Computing 11 (1982) 350–361.
[30] F.B. Shepherd, P.J. Winzer, Selective randomized load balancing and mesh networks with changing demands, Journal of Optical Networking 5 (2006) 320–339.
[31] C. Dettmann, G. Orestis, Product of n independent uniform random variables, Statistics & Probability Letters 79 (2009) 2501–2503.
[32] T. Glickman, F. Xu, The distribution of the product of two triangular random variables, Statistics & Probability Letters 78 (2008) 2821–2826.
[33] J. Schmitt, F. Zdarsky, The DISCO network calculator—a toolbox for worst case analysis, in: Proc. of VALUETOOLS, ACM, 2006.



Hao Wang received his B.Sc. degree in Computer Science and Technology from Northeastern University, China and his M.Sc. degree in Computer Science from University of Kaiserslautern, Germany. In 2009, he joined the Distributed Computer Systems Lab (disco) at the University of Kaiserslautern. His research interests include performance modeling, analysis and simulation of distributed computer systems. Currently he is mainly working on the performance modeling of wireless networks especially using stochastic network calculus. For his research activities, he also spent extended periods of time at Deutsche Telekom Laboratories Berlin, Germany.

[1] The DISCO Network Calculator is publicly available under http://disco.informatik.uni-kl.de/content/Downloads.

Jens B. Schmitt is professor for Computer Science at the TU Kaiserslautern. Since 2003 he has been the head of the Distributed Computer Systems Lab (disco). His research interests are broadly in performance and security aspects of networked and distributed systems. He received his Ph.D. from TU Darmstadt in 2000.

Ivan Martinovic received his Diploma degree (equivalent to M.Sc.) in Computer Science in 2003 from University of Technology Darmstadt, Germany. In 2008, he received his Doctoral degree (Dr. -Ing.) from University of Kaiserslautern, Germany. Currently, he is a postdoc researcher supported by a Carl–Zeiss Foundation Fellowship. His research interests are in the area of network performance evaluation and security, in particular, analysis of the "worst-case" network behaviour from both perspectives—performance and security.