On Expressing Networks with Flow Transformations in Convolution-Form

Florin Ciucu
T-Labs / TU Berlin
florin@net.t-labs.tu-berlin.de

Jens Schmitt and Hao Wang
TU Kaiserslautern
{jschmitt,wang}@informatik.uni-kl.de

***Abstract*—Convolution-form networks have the property that the end-to-end service of network flows can be expressed in terms of a $(min,+)$-convolution of the per-node services. This property is instrumental for deriving end-to-end queueing results which fundamentally improve upon alternative results derived by a node-by-node analysis. This paper extends the class of convolution-form networks with stochastic settings to scenarios with flow transformations, e.g., by loss, dynamic routing or retransmissions. In these networks, it is shown that by using the tools developed in this paper end-to-end delays grow as $\mathcal{O}(n)$ in the number of nodes n; in contrast, by using the alternative node-by-node analysis, end-to-end delays grow as $\mathcal{O}\left(n^2\right)$.**

I. Introduction

Network calculus has been emerging as a promising alternative methodology—to the classical queueing theory [14]—for the performance analysis of packet-switched networks [5], [2], [12]. By its deterministic and stochastic formulations, the calculus is able to model a large set of practical scenarios, and yet to achieve concise results for end-to-end queueing measures with reasonable accuracy [2], [6]. For instance, it has been applied to several important network engineering problems: traditionally in the Internet's Quality of Service proposals IntServ and DiffServ, and more recently in diverse environments such as wireless sensor networks [15], [17], switched Ethernets [19], Systems-on-Chip (SoC) [4], or even to speed-up simulations [13].

The fundamental result of network calculus is the convolution-form representation of networks, within an underlying $(min,+)$ algebra, whereby the service given to a single flow by the whole network can be expressed using a single process. In this way, for broad classes of arrival processes (e.g., light-tailed Markovian or even heavy-tailed and self-similar [16]), end-to-end per-flow queueing measures can be obtained using single-node results, i.e., as if the flow had traversed a single node. The key benefit of using such a $(min,+)$ system theoretic approach is the exact scaling of queueing measures. For instance, end-to-end delay bounds scale as $\Theta(n)$ for both deterministic [2], as well as stochastic settings under complete statistical independence assumptions [9]; in stochastic settings without any independence assumptions the bounds scale as $\Theta(n \log n)$ [7], [3]. In contrast, when neglecting the overall system perspective and conducting a straightforward node-by-node analysis, the resulting additive end-to-end delays scale much more poorly: as $\mathcal{O}(n^2)$ for the deterministic case [2], and even as $\mathcal{O}(n^3)$ for the general stochastic case [7].

A major limitation of the scope of convolution-form networks is caused by an underlying assumption that flows are transported unaltered, for instance lossless, over the network. Concretely, it is still an open problem whether many networks, in which data flows are being transformed on the way to their destinations, can be expressed in convolution-form. Ordinary examples are networks with lossy links, dynamic routing or load balancing, and more sophisticated ones are wireless sensor networks with their typical in-network processing, P2P content distribution systems, media streaming applications with some transcoding happening inside the network (e.g., to accommodate heterogeneous multicast receivers), or even network coding scenarios and distributed real-time systems with heterogeneous resources. Without the convolution-form representation, such networks can be still in principle analyzed with network calculus by conducting an additive node-by-node analysis, but the resulting network queueing measures would scale, as evidenced earlier, poorly.

An extension of network calculus to deal with flow transformations, and yet achieve the desirable convolution-form representations, has been proposed in purely deterministic settings [10]. By introducing the so-called data scaling elements to model actual transformation processes, and by controlling the movement of these elements in the network, the authors showed that the exact scaling properties from the deterministic network calculus are preserved.

Although widening the scope of convolution-form networks, for example to wireless sensor networks with in-network processing [18], the approach from [10] still largely leaves open the limitation in scope of convolution-form networks. The reason lies in the deterministic modelling which can very loosely capture the behavior of networks with stochastic settings. For instance, modelling scaling elements with deterministic bounds can be extremely impractical to capture loss processes in wireless networks, since extreme situations must be accounted for (e.g., all data is lost), which would further lead to trivial results (e.g., zero end-to-end delays). Other impractical situations include scenarios with random routing, load-balancing, or network coding with bursty arrivals. It thus becomes clear that in order to accurately capture the inherent stochastic behavior of flow transformations in networks, one must resort to stochastic modelling techniques.

Towards this goal, this paper contributes by introducing a stochastic scaling element, in the framework of the stochastic network calculus, to model flow transformations in great generality. The new scaling element is carefully defined to achieve (1) convolution-form network representations, and (2) a flexible means of capturing actual transformation processes inside a network. The former allows to preserve the exact scaling properties in network calculus and the latter opens up the modelling scope widely. More technically speaking, the new stochastic scaling element has several useful algebraic properties, of which the most important is that it can be commuted with a service curve element for service modelling. In this way, end-to-end performance bounds can be computed by first reordering a series of service curve and scaling elements, and then applying concatenation properties for each type.

A. Related Work

In network calculus, Chang is the first to introduce an element called router, which has a single data input, a control input and an output [5]. The control input determines which packets appear at the output. In fact, this is a special scaling element introduced to specifically model traffic routing, respectively splitting. He derives the effect a router has on arrival constraints, but convolution-form expressions of end-to-end service are not investigated.

Wandeler introduces so-called workload curves [21] into the real-time calculus, a recent customization of network calculus for hard real-time systems [20]. Workload curves are a special scaling element that translates an event stream into specific requirements for a certain resource, e.g., CPU time or link bandwidth, thus bridging between different subsystems with heterogeneous resources. While [21] focuses on how to actually compute such workload curves (e.g., based on finite-state machine representations of the actual processing components) and it basically just applies a variant of Chang's router element, it is interesting to note that this application of scaling opened up an active research avenue for distributed real-time systems. Convolution-form expressions of end-to-end service are not yet dealt with in that work.

In [10] a general scaling element with minimum and maximum scaling curves is introduced. It is shown how to perform an end-to-end analysis in the presence of flow transformations inside the network based on a convolution-form expression of the end-to-end service. In that sense, it goes beyond [5], [21] and also generalizes previous scaling elements to some degree. The main message is that the scaling properties of the network calculus are preserved. Yet, the work is purely *deterministic* and a stochastic extension is not straightforward. While our work could be seen as a stochastic extension of [10], in that we also strive for convolution-form expressions, we emphasize that we had to depart significantly from [10] by defining the scaling at the level of the arrival processes, instead of as *stochastic* scaling curves.

An interesting and complementary work is the calculus for information-driven networks from [23]. Similar to our work, this calculus also deals with processing inside the network and has many applications such as network coding. The difference lies in the queueing analysis of the data information, in the sense of Shannon entropy, rather than of the data itself. The calculus from [23] can be thus regarded as an important step to bridging the gap between communication networks and classical information theory.

The rest of this paper is structured as follows. In Section II we introduce stochastic scaling elements with some of their properties. Then we use these elements in Section III to derive end-to-end delays in a network with flow transformations. These results are numerically illustrated in Section IV. Brief conclusions are presented in Section V.

II. Stochastic Scaling Element

In this section we first define a (stochastic) scaling element, state its basic properties, and give an example of a Markov-modulated scaling process. Then we show the commutativity property of scaling and service curve elements, which is instrumental for expressing networks with flow transformations in convolution-form.

The time model is discrete. Arrival processes are modelled with non-decreasing and non-negative random processes, taking integer values, and defined on some joint probability space. For an arrival process $A(t)$ we denote for convenience its bivariate extension as $A(s,t) := A(t) - A(s)$. Also, for two r.v.'s X and Y, we denote equality in distribution by $X =_d Y$.

Definition 1: (Scaling Element) A scaling element consists of an arrival process $A(t)$, a scaling random process $\mathbf{X} = (X_i)_{i\geq 1}$ taking non-negative integer values, and a scaled process $A^{\mathbf{X}}(t)$ defined for all $t \geq 0$ as

$$A^{\mathbf{X}}(t) = \sum_{i=1}^{A(t)} X_i \ . \tag{1}$$

For an illustration see Figure 1.

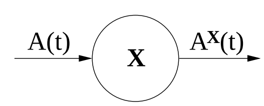


Fig. 1. A scaling element with arrival process $A(t)$, scaling random process $\mathbf{X} = (X_i)_{i\geq 1}$, and scaled process $A^{\mathbf{X}}(t)$.

When $X_i \in \{0,1\}$ for all $i \geq 1$ we say that the scaling process $\mathbf{X}$ is a loss process, which is useful for modelling losses at a link. In turn, if $X_i > 1$ for some i, then the scaling process $\mathbf{X}$ is useful for modelling retransmissions of previous losses or redundant transmissions. For notation convenience, when the elements of $\mathbf{X}$ are *i.i.d.*, we generically refer to X_i by X. Also, two scaling processes $\mathbf{X} = (X_i)_{i\geq 1}$ and $\mathbf{Y} = (Y_i)_{i\geq 1}$ are *i.i.d.* if $\{X_{i\geq 1}, Y_{i\geq 1}\}$ are (mutually) *i.i.d.*

For an arrival process $A(t)$ and a scaling process $\mathbf{X} = (X_i)_{i\geq 1}$, it is helpful to define the corresponding scaled process in *bivariate* form as:

$$A^{\mathbf{X}}(s,t) := A^{\mathbf{X}}(t) - A^{\mathbf{X}}(s) = \sum_{i=A(s)+1}^{A(t)} X_i \ . \tag{2}$$

Note that in general $A^{\mathbf{X}}(s,t) \neq (A(s,t))^{\mathbf{X}}$, but they are equal in distribution under appropriate stationarity assumption on $\mathbf{X}$ and independence between $A(t)$ and $\mathbf{X}$.

We remark that the scaled process $A^{\mathbf{X}}(t)$ from Eq. (1) is defined by space scaling at the granularity of data units, i.e., packets. A coarser scaling may be defined by scaling at the granularity of time units. Concretely, for a scaling random process $\mathbf{Y} = (Y_s)_{s \geq 1}$, the scaled process of $A(t)$ is the process $A^{\mathbf{Y}}(t)$ defined as

$$A^{\mathbf{Y}}(t) = \sum_{s=1}^{t} a(s) Y_s \ , \tag{3}$$

for all $t \geq 1$, where $a(s) = A(s) - A(s-1)$ is the instantaneous arrival process of $A(t)$.

The space and time scaling models are tightly related in the following way. On one hand, given the scaled process $A^{\mathbf{X}}(t)$ from Eq. (1), one can define for all $s \geq 1$ the process $Y_s = \frac{\sum_{i=A(s-1)+1}^{A(s)} X_i}{a(s)}$ such that $A^{\mathbf{Y}}(t) = A^{\mathbf{X}}(t)$. Conversely, given the scaled process $A^{\mathbf{Y}}(t)$ from Eq. (3), one can define for all $i \geq 1$ the process $X_i = Y_{\min\{s : i \leq A(s)\}}$ such that $A^{\mathbf{X}}(t) = A^{\mathbf{Y}}(t)$.

The rest of the paper considers the space scaling model only. The next lemma states some basic properties of scaling elements which are useful in analyzing networks with flow transformations.

Lemma 1: (PROPERTIES OF SCALING ELEMENTS) For an arrival process $A(t)$ and an independent and stationary scaling process $\mathbf{X} = (X_i)_{i \geq 1}$ the following basic properties hold.

1) (Scaling Additivity) If $\mathbf{Y} = (Y_i)_{i \geq 1}$ is a scaling process then

$$A^{\mathbf{X}+\mathbf{Y}} = A^{\mathbf{X}} + A^{\mathbf{Y}} \ ,$$

as illustrated in Figure 2.

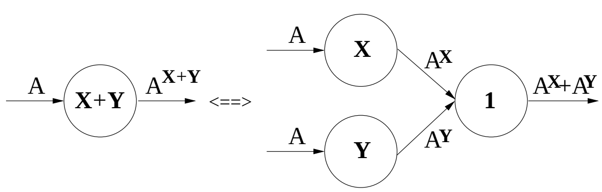


Fig. 2. Scaling with a sum. The identity scaling element $\mathbf{1} = (1, 1, \dots)$ plays the role of a multiplexer.

2) (Arrival Additivity) If $B(t)$ is an arrival process and X_i's are *i.i.d.* then

$$(A + B)^{\mathbf{X}} =_d A^{\mathbf{X}} + B^{\mathbf{Y}} \ ,$$

as illustrated in Figure 3, where $\mathbf{Y} = (Y_i)_{i \geq 1}$ is a scaling process such that $\mathbf{X}$ and $\mathbf{Y}$ are *i.i.d.*.

3) (Stationarity) If both $A(t)$ and $\mathbf{X}$ are stationary and independent then the scaled process $A^{\mathbf{X}}(t)$ is stationary, i.e., for all $s, t \geq 0$

$$A^{\mathbf{X}}(s, s+t) =_d A^{\mathbf{X}}(t) \ . \tag{4}$$

However, if $B(s)$ is an additional arrival process, not necessarily independent of $A(t)$, and $\mathbf{X}$ is independent

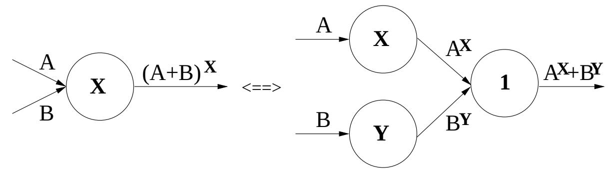


Fig. 3. Scaling of a sum; $\mathbf{X}$ and $\mathbf{Y}$ are *i.i.d.*

of $(A(t), B(s))$, then we have the following bound for $s, t \geq 0$ and $x \geq 0$

$$Pr\left(A^{\mathbf{X}}(t) - B^{\mathbf{X}}(s) \geq x\right) \leq Pr\left((A(t) - B(s))^{\mathbf{X}} \geq x\right) \ . \tag{5}$$

4) (Concatenation and Commutativity) If $\mathbf{X}$ is a loss process, $\mathbf{Y} = (Y_i)_{i \geq 1}$ is a scaling process independent of $\mathbf{X}$, and X_i's, Y_i's are *i.i.d.* then

$$\left(A^{\mathbf{X}}\right)^{\mathbf{Y}} =_d A^{\mathbf{XY}} \ ,$$

where $\mathbf{XY}$ is the scalar product, as illustrated in Figure 4.

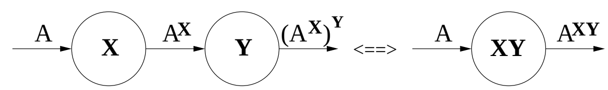


Fig. 4. Concatenation; $\mathbf{X}$ is a loss process and $\mathbf{X}$ and $\mathbf{Y}$ are *i.i.d.*

If $\mathbf{Y}$ is also a loss process then we have the commutativity property

$$\left(A^{\mathbf{X}}\right)^{\mathbf{Y}} =_d \left(A^{\mathbf{Y}}\right)^{\mathbf{X}} \ .$$

5) (Absorbing Zero and Identity Elements) If $\mathbf{0} = (0, 0, \dots)$ and $\mathbf{1} = (1, 1, \dots)$ then

$$A^{\mathbf{0}} = 0, \quad A^{\mathbf{1}} = A.$$

We point out that the properties of concatenation and commutativity require the strong condition that the processes are *i.i.d.*; as it will be evident from the proof, this condition cannot be relaxed to the stationarity of the processes.

PROOF. The two additivity properties and the absorbing zero and identity elements' properties follow directly from the definition of the scaled process.

For the stationary bound from Eq. (5) we have for some $x > 0$

$$\begin{aligned}
&Pr\left(A^{\mathbf{X}}(t) - B^{\mathbf{X}}(s) \geq x\right) \\
&\quad = Pr\left(\sum_{i=B(s)+1}^{A(t)} X_i \geq x,\ A(t) > B(s)\right) \\
&\quad = Pr\left(\sum_{i=1}^{A(t)-B(s)} X_i \geq x,\ A(t) > B(s)\right) \\
&\quad = Pr\left(\sum_{i=1}^{A(t)-B(s)} X_i \geq x\right) \ .
\end{aligned}$$

In the second line we used the formula of total probability. In the third line we used the stationarity of $\mathbf{X}$ and its independence of $A(t)$ and $B(s)$. The inequality in Eq. (5) appears in the case of $x = 0$. Using the same argument, one may prove a generalized statement that for $x \geq 0$

$$Pr\left(\left(A^{\mathbf{X}}(t) - B^{\mathbf{X}}(s) - C\right)^{\mathbf{Y}} \geq x\right)$$
$$\leq Pr\left(\left(\left(A(t) - B(s)\right)^{\mathbf{X}} - C\right)^{\mathbf{Y}} \geq x\right) \ , \qquad (6)$$

where C and $\mathbf{Y}$ are additional r.v./scaling process, stationary and independent of the rest; critical for the proof is the positivity of $\mathbf{X}$ and $\mathbf{Y}$, as in the derivation of Eq. (5).

For the concatenation property we first condition on $A(t)$ and thus it is sufficient to prove that

$$\sum_{j=1}^{X_1+X_2+\cdots+X_n} Y_j =_d \sum_{i=1}^{n} X_i Y_i$$

for all $n \geq 1$. Further conditioning on $(X_1, X_2, \ldots, X_n)$, and taking into account that $\mathbf{X}$ is a loss process, it becomes sufficient to prove that

$$Y_1 + Y_2 + \cdots + Y_m =_d Y_{i_1} + Y_{i_2} + \cdots + Y_{i_m} \ ,$$

for all $m \geq 1$ and $i_1 \leq i_2 \leq \cdots \leq i_m$. This is true from the *i.i.d.* property of Y_i's. The commutativity property follows from applying the concatenation property. $\square$

A. Example: Markov-Modulated Scaling Processes (MMSP)

Here we give an example of a scaling process $\mathbf{X} = (X_i)_{i \geq 1}$ as being modulated by a discrete and homogeneous Markov process $S(i)$ with states $1, 2, \ldots, M$ and transition probabilities $\lambda_{i,j}$ for all $1 \leq i, j \leq M$, as in Figure 5. Let also the *i.i.d.* random processes $L_i(n)_{n \geq 1}$ for all $1 \leq i \leq M$. The corresponding scaling process is defined as

$$X_i = L_{S(i)}(i) \ ,$$

i.e., the scaling follows the distribution of $L_{S(i)}(1)$ while in state $S(i)$. In the special case when the Markov chain has a single state, i.e., $M = 1$, or even two states with $\lambda_{1,2} + \lambda_{2,1} = 1$, then the entire scaling process $\mathbf{X} = (X_i)_{i \geq 1}$ is *i.i.d.*

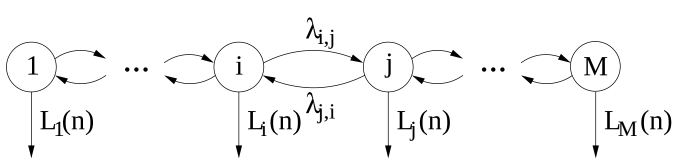


Fig. 5. A Markov chain $S(i)$ with M states and transition probabilities $\lambda_{i,j}$, modulating the scaling process $\mathbf{X}$ as $X_i = L_{S(i)}(i)$.

A loss process can be fitted from the classical two-state Gilbert-Elliott loss model [11], [8] or the finite-state Markov channel for modelling Rayleigh fading [22]. Further, by inversing these models and adding a time offset, one may consequently determine a retransmission scaling process.

In order to carry out a queueing algebra with scaling elements it is useful to compute the moment generating functions (MGFs) of the scaled processes in terms of the MGFs of the arrival processes. First we introduce the notation $M_X(\theta) := E\left[e^{\theta X}\right]$ for some r.v. X and $\theta > 0$.

Lemma 2: (MOMENT GENERATING FUNCTION OF A SCALED PROCESS) Let an arrival process $A(t)$ and an MMSP $\mathbf{X} = (X_i)_{i \geq 1}$ defined as above. Then we have the MGFs for some $\theta > 0$:

1) (General case) If the matrix $\lambda = (\lambda_{i,j})_{i,j}$ is irreducible and aperiodic then

$$M_{A^{\mathbf{X}}(t)}(\theta) \leq M_{A(t)}\left(\log \text{sp}\left(\phi(\theta)\lambda\right)\right) \ , \qquad (7)$$

where $\phi(\theta) := \text{diag}\left(M_{L_1(1)}(\theta), \ldots, M_{L_M(1)}(\theta)\right)$ and $\text{sp}\left(\phi(\theta)\lambda\right)$ denotes the spectral radius of $\phi(\theta)\lambda$.

2) (*I.i.d.* case) If X_i's are *i.i.d.* then

$$M_{A^{\mathbf{X}}(t)}(\theta) = M_{A(t)}\left(\log M_X(\theta)\right) \ . \qquad (8)$$

The proof follows using conditioning (e.g., see [5]).

The above lemma can be applied recursively to derive the MGF of a scaled process through a series of scaling elements. For instance, if $\mathbf{X}$ and $\mathbf{Y}$ are *i.i.d.* then

$$\begin{aligned} M_{(A^{\mathbf{X}})^{\mathbf{Y}}(t)}(\theta) &= M_{A^{\mathbf{X}}(t)}\left(\log M_Y(\theta)\right) \\ &= M_{A(t)}\left(\log M_X\left(\log M_Y(\theta)\right)\right) \ . \end{aligned}$$

If $\mathbf{X}$ is additionally a loss process then the last line is further equal to $M_{A(t)}\left(\log M_{XY}(\theta)\right)$, by applying the concatenation property from Lemma 1.

B. Commuting with Service Curve Elements

Here we show how to commute a series of a service curve element and a scaling element. As mentioned earlier, this property is instrumental for expressing networks with flow transformations in convolution-form. First we briefly introduce service curve elements.

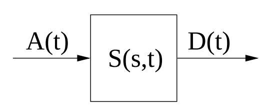


Fig. 6. A service curve element with arrival process $A(t)$, service random process $S(s,t)$, and departure process $D(t)$.

Service curves provide lower bounds on the service received by an arrival flow $A(t)$ at a network node (see Figure 6), and are formally defined by a bivariate random process $S(s,t)$ such that the departure process $D(t)$ satisfies [5]

$$D(t) \geq A * S(t) \ , \qquad (9)$$

for all $t \geq 0$, where '$*$' denotes the $(min, +)$-convolution defined as $A * S(t) = \inf_{0 \leq s \leq t}\left\{A(s) + S(s,t)\right\}$. If the inequality above holds with equality then the service curve is said to be exact.

Lemma 3: (COMMUTING SERVICE CURVE AND SCALING ELEMENTS) Consider a system with an arrival process $A(t)$ which goes through a service curve $S(s,t)$ and then a scaling element $\mathbf{X} = (X_i)_{i \geq 1}$. In another system, $A(t)$ goes first through $\mathbf{X}$ and then through the exact service curve $T(s,t) :=$

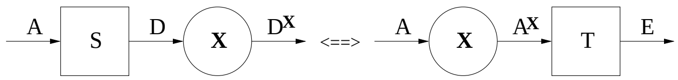


Fig. 7. Commuting service curve and scaling elements

$\sum_{i=A(s)+1}^{A(s)+S(s,t)} X_i$, as shown in Figure 7. If $A(t)$, $\mathbf{X}$, and $S(s,t)$ are independent, then for all $t \geq 0$

$$E(t) \leq D^{\mathbf{X}}(t) \ , \tag{10}$$

where $D^{\mathbf{X}}(t)$ and $E(t)$ are the departure processes in the two systems. Moreover, if A, S and $\mathbf{X}$ are independent, and $\mathbf{X}$ is stationary, then $M_{T(s,t)}(-\theta) = M_{S(s,t)}(\log M_X(-\theta))$ for any $\theta > 0$.

PROOF. From the definition of scaling and service elements we have immediately for some $t \geq 0$

$$\begin{aligned} E(t) &= \inf_{0\leq s\leq t} \left\{ A^{\mathbf{X}}(s) + T(s,t) \right\} \\ &= \inf_{0\leq s\leq t} \left\{ \sum_{i=1}^{A(s)} X_i + \sum_{i=A(s)+1}^{A(s)+S(s,t)} X_i \right\} \\ &= \inf_{0\leq s\leq t} \sum_{i=1}^{A(s)+S(s,t)} X_i \leq \sum_{i=1}^{D(t)} X_i = D^{\mathbf{X}}(t) \ . \end{aligned}$$

The rest of proof follows by successive conditioning. $\square$

The lemma ensures that backlog/delay processes in the transformed system are bigger in distribution than in the original system. For instance, for an arrival process $A(t)$ and departure process $D(t)$, the delay process is defined by the process $W(t) = \inf \{d : A(t-d) \leq D(t)\}$. We also point out that although the expression of $T(s,t)$ depends on $A(s)$, the expression of $T(s,t)$'s Laplace transform is sufficient to elegantly carry out end-to-end computations. The next section presents a detailed end-to-end delay analysis of a network with flow transformations by means of Lemma 3.

III. SCALING OF END-TO-END DELAYS

In this section we compute end-to-end delays in a flow transformation network consisting of a series of alternate service and scaling elements. In particular we demonstrate that by using Lemma 3, which allows the transformation of this network in a convolution-form network by repeatedly commuting scaling and service elements, the end-to-end delays scale linearly in the number of service elements. In contrast, we also show that by applying the alternative node-by-node and additive analysis, end-to-end delays scale quadratically.

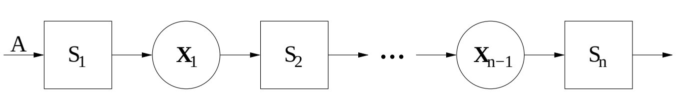


Fig. 8. A flow transformation network consisting of sequence of alternating service and scaling elements.

We consider the flow transformation network scenario from Figure 8. A stationary arrival process $A(t)$ crosses a series of alternate service and scaling elements denoted by S_i and $\mathbf{X}_i$, respectively. We assume that all the service and scaling processes are stationary and (mutually) independent. This network scenario can be seen as a flow's view, or a part of it, in a network with loss or random routing.

A. Transformation in Convolution-Form

The next theorem provides a bound on the end-to-end delay and the corresponding order of growth in Figure 8.

Theorem 1: (END-TO-END DELAYS IN A FLOW TRANSFORMATION NETWORK) Consider the network scenario from Figure 8 where a stationary arrival process $A(t)$ crosses a series of alternate stationary and (mutually) independent service and scaling elements denoted by $S_1, S_2, \ldots, S_n$ and *i.i.d.* $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_{n-1}$, respectively. Assume the MGF bounds $M_{A(s,t)}(\theta) \leq e^{\theta r(\theta)(t-s)}$ and $M_{S_k(t)}(-\theta) \leq e^{-\theta C_k t}$, for $k = 1, \ldots, n$, and some $\theta > 0$. Under a stability condition, to be explicitly given in the proof, we have the following end-to-end steady-state delay bounds for all $d \geq 0$

$$Pr\Big(W > d\Big) \leq K^n b^d \ , \tag{11}$$

where the constants K and b are to be given in the proof as well. Moreover, the ε-quantiles scale as $\mathcal{O}(n)$, for some $\varepsilon > 0$.

PROOF. Fix $t, d \geq 0$ and denote for convenience for all $k, s \geq 0$

$$A^{(k)}(s) := \left(\cdots \left(A^{\mathbf{X}_1}\right)^{\mathbf{X}_2^{\cdot^{\cdot^{\cdot}}}} \right)^{\mathbf{X}_k} (s) \tag{12}$$

the iterative scaling of A by $\mathbf{X}_1$, $\mathbf{X}_2$, $\ldots$, $\mathbf{X}_k$. Also, for $k \geq 0$ we introduce the scaled processes $U_k(s, u_k)$ defined as

$$U_0(s, u_0) = A(s) \ ,$$

for $u_0 = s$, and then recursively

$$U_k(s, u_k) = \Big(U_{k-1}(s, u_{k-1}) + S_k(u_{k-1}, u_k)\Big)^{\mathbf{X}_k} \tag{13}$$

for $k \geq 1$ and $u_{k-1} \leq u_k$.

We are going to prove the claim from the theorem by induction. For $k \geq 1$ we let the following two statements $(\mathcal{S}_1)$ and $(\mathcal{S}_2)$ for the induction process:

$$\begin{aligned} (\mathcal{S}_1) \quad & Pr\Big(W_k(t) > d\Big) \leq \sum_{0\leq s\leq t-d} \ \sum_{s\leq u_1\leq \cdots\leq u_{k-1}\leq t} \\ & Pr\left(A^{(k-1)}(t-d) > U_{k-1}\left(s, u_{k-1}\right) + S_k\left(u_{k-1}, t\right)\right) \ , \end{aligned}$$

and for fixed s and u_k

$$\begin{aligned} (\mathcal{S}_2) \quad & \Big(A^{(k-1)}(s) + T_{k-1} * S_k(s, u_k)\Big)^{\mathbf{X}_k} \\ & \qquad = \inf_{s\leq u_1\leq \cdots\leq u_k} U_k\left(s, u_k\right) \ , \end{aligned}$$

where T_k is defined recursively as $T_0(0) = 0$, $T_0(s) = \infty$ for all $s > 0$, and

$$T_k(s,t) := \sum_{i=A^{(k-1)}(s)+1}^{A^{(k-1)}(s)+T_{k-1}*S_k(s,t)} X_{k,i} \ . \tag{14}$$

For the initial step of the induction, i.e., $k = 1$, we have

$$\begin{aligned} Pr\Big(W_1(t) > d\Big) &= Pr\left(A(t-d) > D(t)\right) \\ &\le Pr\left(A(t-d) > A * S_1(t)\right) \\ &\le \sum_{0 \le s \le t-d} Pr\left(A(t-d) > U_0\left(s,s\right) + S_1(s,t)\right) \ , \end{aligned} \quad (15)$$

which verifies the first statement $(\mathcal{S}_1)$. In the first line $D(t)$ is the output process from the service element S_1 and we used the equivalence $W_1(t) > 0 \Leftrightarrow A(t-d) > D(t)$, in the second line we used the definition of the service curve, and in the third line we expanded the $(min, +)$ convolution and applied the union bound.

In turn, for the second statement $(\mathcal{S}_2)$, we have

$$\begin{aligned} \left(A(s) + T_0 * S_1(s,u_1)\right)^{\mathbf{X}_1} &= \left(A(s) + S_1\left(s,u_1\right)\right)^{\mathbf{X}_1} \\ &= \inf_{s \le u_1} U_1\left(s,u_1\right) \ , \end{aligned}$$

which verifies $(\mathcal{S}_2)$. In the first line we used that $T_0(0) = 0$, $T_0(s) = \infty$ for $s > 0$, and then we used the definition of $U_k(s, u_k)$ from Eq. (13).

For the inductive step we assume that $(\mathcal{S}_1)$ and $(\mathcal{S}_2)$ hold for some $k \ge 1$ and we will prove them for $k + 1$.

First, we observe that after iteratively commuting for k times the service and service elements from Figure 8, we obtain the system from Figure 9; note that, according to Lemma 3, the output before the k^{th} service element in the transformed system is smaller than in the original system.

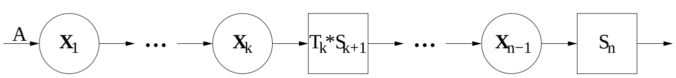


Fig. 9. Transformation of the system from Figure 8 after iteratively applying Lemma 3 for k times.

Using the argument from Eq. (15) we can write for the end-to-end delay until the $k+1^{th}$ scaling element

$$\begin{aligned} & Pr\left(W_{k+1}(t) > d\right) \\ \le & \sum_{0 \le s \le t-d} Pr\Big(A^{(k)}(t-d) > A^{(k)}(s) + T_k * S_{k+1}(s,t)\Big) \\ \le & \sum_{0 \le s \le t-d} \sum_{s \le u_k \le t} Pr\Big(A^{(k)}(t-d) > \\ & \qquad A^{(k)}(s) + T_k(s,u_k) + S_{k+1}(u_k,t)\Big) \\ \le & \sum_{0 \le s \le t-d} \sum_{s \le u_k \le t} Pr\Big(A^{(k)}(t-d) > \\ & \qquad \Big(A^{(k-1)}(s) + T_{k-1} * S_k(s,u_k)\Big)^{\mathbf{X}_k} + S_{k+1}(u_k,t)\Big) \\ \le & \sum_{0 \le s \le t-d} \sum_{s \le u_1 \le \dots \le u_k \le t} Pr\Bigg(A^{(k)}(t-d) > \\ & \qquad U_k(s,u_k) + S_{k+1}(u_k,t)\Bigg) \ , \end{aligned}$$

which proves the statement $(\mathcal{S}_1)$ for $k + 1$. In the third line we expanded the convolution and applied the union bound, and finally we used the induction hypothesis for $(\mathcal{S}_2)$ together with the union bound.

Lastly, for the induction argument, we need to prove the statement $(\mathcal{S}_2)$ for $k + 1$. We have

$$\begin{aligned} & \Big(A^{(k)}(s) + T_k * S_{k+1}(s,u_{k+1})\Big)^{\mathbf{X}_{k+1}} \\ &= \inf_{s \le u_k \le u_{k+1}} \Big(A^{(k)}(s) + T_k(s,u_k) + S_{k+1}(u_k,u_{k+1})\Big)^{\mathbf{X}_{k+1}} \\ &= \inf_{s \le u_k \le u_{k+1}} \Bigg(\Big(A^{(k-1)}(s) + T_{k-1} * S_k(s,u_k)\Big)^{\mathbf{X}_k} \\ & \qquad\qquad + S_{k+1}(u_k,u_{k+1})\Bigg)^{\mathbf{X}_{k+1}} \\ &= \inf_{s \le u_k \le u_{k+1}} \left(U_k(s,u_k) + S_{k+1}(u_k,u_{k+1})\right)^{\mathbf{X}_{k+1}} \\ &= \inf_{s \le u_k \le u_{k+1}} U_{k+1}(s,u_{k+1}) \ , \end{aligned}$$

which proves the claim. In the next to last line we used the induction hypothesis and then the definition of $U_k(s, u_k)$ from Eq. (13). The induction argument is thus complete.

Next we compute the end-to-end delay bound on $W_n(t)$ by using the statement $(\mathcal{S}_1)$ for $k = n$. We have

$$\begin{aligned} & Pr\Big(W_n(t) > d\Big) \le \sum_{0 \le s \le t-d} \sum_{s \le u_1 \le \dots \le u_{n-1} \le t} Pr\Bigg(A^{(n-1)}(t-d) \\ & > \Bigg(\dots\Big(\left(A(s) + S_1(s,u_1)\right)^{\mathbf{X}_1} + S_2(u_1,u_2)\Big)^{\mathbf{X}_2} \\ & \qquad + \dots + S_{n-1}(u_{n-2},u_{n-1})\Bigg)^{\mathbf{X}_{n-1}} + S_n(u_{n-1},t)\Bigg) \\ & \le \sum_{0 \le s \le t-d} \sum_{s \le u_1 \le \dots \le u_{n-1} \le t} \\ & \quad Pr\Bigg(\Bigg(\dots\Big(\left(A(t-d-s) - S_1(s,u_1)\right)^{\mathbf{X}_1} - S_2(u_1,u_2)\Big)^{\mathbf{X}_2} \\ & \qquad - \dots - S_{n-1}(u_{n-2},u_{n-1})\Bigg)^{\mathbf{X}_{n-1}} > S_n(u_{n-1},t)\Bigg) \ , \end{aligned}$$

after repeatedly applying the stationary bounds in Eqs. (5) and (6) from Lemma 1. Next, using the Chernoff bound for some $\theta > 0$, we obtain

$$\begin{aligned} Pr\Big(W_n(t) > d\Big) \le & \sum_{0 \le s \le t-d} \sum_{s \le u_1 \le \dots \le u_{n-1} \le t} \\ & e^{a_{n-1} r(a_{n-1})(t-d-s)} e^{-a_{n-1} C_1(u_1 - s)} e^{-a_{n-2} C_2(u_2 - u_1)} \\ & \qquad \dots e^{-a_0 C_n(t - u_{n-1})} \ . \end{aligned}$$

Here we recursively used Lemma 2 and obtained the bounds

$$E\left[e^{\theta A^{(k)}(t)}\right] \le e^{a_k r(a_k) t} \ , \quad (16)$$

for $k \ge 0$ (refer to Eq. (12) for the definition of $A^{(k)}(t)$). Note that $a_0 = \theta$ and a_k's for $k \ge 1$ do not depend on $A(t)$ but instead are formed recursively as

$$a_{k+1} = \log E\left[e^{a_k X_{1,1}}\right] \quad (17)$$

in the case when $X_{1,i}$'s are *i.i.d.* More generally, when $\mathbf{X}_k$'s are MMSPs, the recursion is given by

$$a_{k+1} = \log \mathrm{sp}\left(\phi_{\mathbf{X}_{k+1}}(a_k)\lambda\right) \tag{18}$$

by Lemma 2, where the diagonal matrices $\phi_{\mathbf{X}_{k+1}}(a_k)$ are defined as $\phi(\theta)$ from Lemma 2, but now relative to each $\mathbf{X}_{k+1}$ for $k \geq 0$.

Denoting

$$b = \sup_{k=0,\ldots,n-1} e^{-a_k C_{n-k}} \tag{19}$$

and imposing the following stability condition

$$a_{n-1} r(a_{n-1}) + \log b < 0 \ , \tag{20}$$

we can bound the sums as

$$\begin{aligned} &Pr\Big(W_n(t) > d\Big) \\ &\leq \sum_{0 \leq s \leq t-d} \ \sum_{s \leq u_1 \leq \cdots \leq u_{n-1} \leq t} e^{a_{n-1} r(a_{n-1})(t-d-s)} b^{t-s} \\ &= \sum_{0 \leq s \leq t-d} \binom{t-s+n-1}{n-1} e^{a_{n-1} r(a_{n-1})(t-d-s)} b^{t-s} \\ &\leq K^n b^d \ . \end{aligned}$$

where $\binom{t-s+n-1}{n-1}$ is the number of combinations with repetition and $K = \frac{\left(1+\frac{d}{n}\right)^{1+\frac{d}{n}}}{\left(\frac{d}{n}\right)^{\frac{d}{n}}}$. Here we used that $e^{a_{n-1}r(a_{n-1})} b < 1$ (note the stability condition from Eq. (20)), and then applied the summation results from Theorem 3 in [9]. Taking $t \to \infty$ proves the result from Eq. (11). Finally, the order of growth of the ε-quantiles for some $0 < \varepsilon < 1$ follow directly from Eq. (11) by observing that K is bounded in n. The proof is now complete. $\square$

B. Alternative Node-by-Node Analysis

Here we show that end-to-end delays obtained by a straightforward node-by-node analysis fundamentally differ from those obtained in Theorem 1 in that the order of growth is quadratic in number of nodes, as opposed to linear.

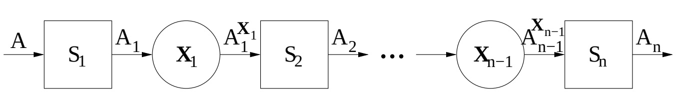


Fig. 10. A flow transformation network consisting of a sequence of alternating service and scaling elements.

Consider the same network setting as in the previous subsection, and with the notations from Figure 10, where A_k represents the output from the k^{th} service element, for $k = 1, \ldots, n$. Also, denote by convention $A_0 = A$ and let $\mathbf{X}_0 = \mathbf{1}$.

Assume as in Theorem 1 that $M_{A(s,t)}(\theta) \leq e^{\theta r(\theta)(t-s)}$ and $M_{S_k(t)}(-\theta) \leq e^{-\theta C_k t}$, for $k = 1, \ldots, n$, and some $\theta > 0$. Assume also the stability condition $a_{k-i+1} C_i > a_k r(a_k)$ for all $1 \leq i \leq k$, where a_k's are defined as in Eq. (17) or (18).

We will first prove by induction the statement

$$(\mathcal{S}) \quad E\left[e^{\theta A_k^{\mathbf{X}_k}(t)}\right] \leq M_k e^{a_k r(a_k) t} \ , \tag{21}$$

where $M_0 = 1$ and for $k \geq 1$

$$M_k = \prod_{i=1}^{k} \frac{1}{a_{k-i+1} C_i - a_k r(a_k)} \ . \tag{22}$$

The statement is immediately true for $k = 0$ from the initial assumptions. Let us now assume that the statement $(\mathcal{S})$ holds for k and prove it for $k+1$. We can write for the MGF of the output $A_{k+1}(s,t)$ from the $(k+1)^{th}$ service element (see [9])

$$\begin{aligned} E\left[e^{\theta A_{k+1}(s,t)}\right] &\leq E\left[e^{\theta\left(A_k^{\mathbf{X}_k}(t) - A_{k+1}(s)\right)}\right] \\ &\leq E\left[e^{\theta\left(A_k^{\mathbf{X}_k}(t) - A_k^{\mathbf{X}_k} * S_{k+1}(s)\right)}\right] \\ &\leq \sum_{0 \leq u < s} M_k e^{a_k r(a_k)(t-u)} e^{-\theta C_{k+1}(s-u)} \\ &\leq \frac{M_k}{\theta C_{k+1} - a_k r(a_k)} e^{a_k r(a_k)(t-s)} \ . \end{aligned} \tag{23}$$

In the first line we used that the output at time t is dominated by the corresponding input. In the second line we used the definition of the service curve. In the third line we used the union bound and the induction hypothesis, and finally we estimated the sum by an integral.

Next, for the MGF of $A_{k+1}^{\mathbf{X}_{k+1}}(s,t)$, Lemma 2 yields

$$E\left[e^{\theta A_{k+1}^{\mathbf{X}_{k+1}}(s,t)}\right] = E\left[e^{a_1 A_{k+1}(s,t)}\right]$$

Since the bound on the MGF of $A_{k+1}(s,t)$ from Eq. (23) is derived for θ, we need to replace all the occurrences of θ by a_1 (note that all the a_k's depend on θ, according to their definitions from either Eq. (17) or (18)). Consequently, a_k is to be replaced by a_{k+1} for all $k \geq 0$, which yields the MGF bound

$$\begin{aligned} E\left[e^{\theta A_{k+1}^{\mathbf{X}_{k+1}}(s,t)}\right] &\leq \frac{1}{a_{k+1} C_1 - a_{k+1} r(a_{k+1})} \\ &\cdots \frac{1}{a_2 C_k - a_{k+1} r(a_{k+1})} \frac{e^{a_{k+1} r(a_{k+1})(t-s)}}{a_1 C_{k+1} - a_{k+1} r(a_{k+1})} \\ &= M_{k+1} e^{a_{k+1} r(a_{k+1})(t-s)} \ , \end{aligned}$$

which proves that the statement $(\mathcal{S})$ holds for $k+1$, and thus the induction proof is complete.

In the following, having the MGF bounds for the arrivals at the k^{th} service element, and the service curves $S_k(s,t)$, we can derive the corresponding per-node delay bounds $W_k(t)$. Using the same arguments as in Eq. (15) we obtain for all $k \geq 1$ and $d \geq 0$

$$Pr\Big(W_k > d\Big) \leq L_k e^{-\theta C_k d} \ , \tag{24}$$

where the prefactors L_k's are defined as

$$L_k = \prod_{i=1}^{k} \frac{1}{a_{k-i} C_i - a_{k-1} r(a_{k-1})} \ ,$$

Note that replacing all the occurrences of θ from L_k with a_1 yields the M_k's defined earlier in Eq. (22).

Let us next make the convenient choices

$$b_k = \inf_{i=1\ldots k} a_{k-i} C_i$$

such that $L_k \leq \left(\frac{1}{b_k - a_{k-1} r(a_{k-1})}\right)^k$, and

$$b = \sup_{k=1,\ldots,n} \frac{1}{b_k - a_{k-1} r(a_{k-1})} \ .$$

We can thus bound the bound on W_k by

$$Pr\Big(W_k > d\Big) \leq b^k e^{-\theta C_k d} \ ,$$

for all $k \geq 1$.

Finally, a bound on the end-to-end delay $W = \sum_k W_k$ can be formulated as the optimization problem

$$Pr\Big(W > d\Big) \leq \inf_{d_1+\cdots+d_n=d} \left\{ b e^{-\theta C_1 d_1} + \cdots + b^n e^{-\theta C_n d_n} \right\} \ .$$

Letting $C = \sup_{k=1,\ldots,n} C_k$ we find the infimum (see [7])

$$Pr\Big(W > d\Big) \leq n b^{\frac{n+1}{2}} e^{-\frac{\theta}{n} C d} \ . \tag{25}$$

From here one may easily determine that the quantiles of the end-to-end delay grow as $\mathcal{O}\left(n^2\right)$, which proves the claim from the beginning of this section.

IV. Numerical Results

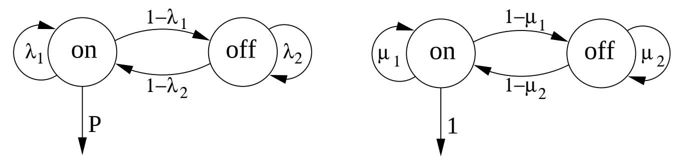
In this section we numerically compare the end-to-end delay bounds from Theorem 1 with those obtained by the alternative node-by-node analysis presented in Subsection III-B, and illustrate the corresponding order of growths. Then we draw some insights on how the burstiness in arrival vs. scaling processes affect the results from Theorem 1.

For Theorem 1 we directly use the delay bound from Eq. (11). In turn, for the node-by-node analysis, we use the end-to-end delay bound

$$P\Big(W > d\Big) \leq \inf_{\sum_k d_k = d} \left\{ \sum_k L_k e^{-\theta C_k d_k} \right\} \tag{26}$$

for which the infimum can be computed exactly using a convex optimization result from [7] (see also Eq. (24) for the values of L_k). We point out that we do not use the expression from Eq. (25) which was subject to several convenient bounding choices; these were made in order to derive a result which can concisely express the $\mathcal{O}(n^2)$ growth.

We consider the flow transformation network scenario from Figure 8 with two examples of arrival processes $A(t)$: Poisson with rate λ and Markov-Modulated On-Off (MMOO). The MMOO process is represented in Figure 11.(a) in terms of the transition probabilities λ_1 and λ_2, and also the peak-rate P, i.e., the process transmits at rate P while in state 'on' and is idle while in state 'off'. When $\lambda_1 + \lambda_2 = 1$ then $A(t)$ has independent increments and is thus a sum of Bernoulli random variables $B(\lambda_1)$. We consider the scaling (loss) processes from Figure 8 as MMOO processes as represented in Figure 11.(b) (note that for the loss process the rate while in the 'on' state is 1).



(a) Arrival process $A(t)$ (b) Scaling processes $\mathbf{X}_i$'s

Fig. 11. Representation of Markov-Modulated On-Off (MMOO) processes.

For the MMOO process $A(t)$ from Figure 11.(a) we have the following bound on its MGF [5]

$$E\left[e^{\theta A(t)}\right] \leq e^{\theta r(\theta) t} \ ,$$

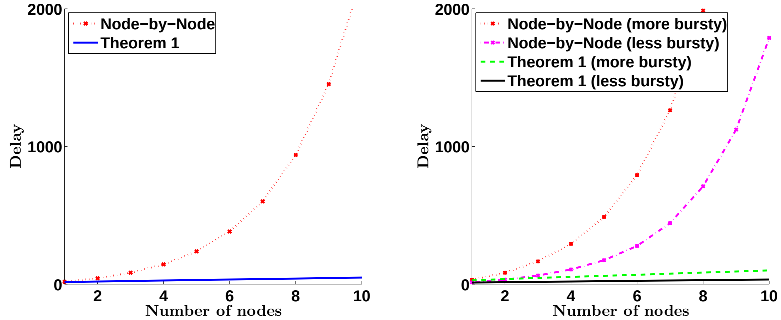
where $r(\theta) = \frac{1}{\theta} \log \frac{\lambda_1 e^{\theta P} + \lambda_2 + \sqrt{(\lambda_1 e^{\theta P} + \lambda_2)^2 - 4(\lambda_1 + \lambda_2 - 1) e^{\theta P}}}{2}$. Similar bounds apply for the MGF's of the cumulative processes of the $\mathbf{X}_i$'s, with different parameters.

We next use the following numerical settings. The average rates of the arrivals are normalized to one packet per one time unit. The utilization at the first node is .75, i.e., $C_1 = 1.33$. The capacities at the rest of the nodes are set as $C_k = \frac{a_{n-1}}{a_{n-k}} C_1$, such that there is no loss in accuracy in the end-to-end delay from Theorem 1, as a result of the bounding from Eq. (19).

Figures 12.(a,b) illustrate the order of growths of the end-to-end delay bounds obtained with Theorem 1 and the node-by-node analysis for $n = 1, \ldots, 10$ service elements in Figure 8, by plotting the corresponding ε-quantiles (in time units) with $\varepsilon = 10^{-3}$. We consider Bernoulli scaling processes (with $\mu_1 = 1 - \mu_2 = .75$ in Figure 11.(b)) and different arrival processes (Poisson in (a) and two MMOO's with different levels of burstiness in (b); the parameters are displayed in the caption). For all the arrival processes the figure clearly illustrates the $\mathcal{O}(n)$ vs. $\mathcal{O}(n^2)$ order of growths of the end-to-end delays. Moreover, as illustrated by (b), the $\mathcal{O}(n)$ results are much less sensitive to the arrivals' burstiness than the $\mathcal{O}(n^2)$ results, indicating large pre-constants in the latter.

Figure 13 illustrates the impact of burstiness in the scaling processes over the burstiness in the arrival processes for $n = 1, \ldots, 10$ service elements in Figure 8. We consider four arrival processes (one Poisson and three MMOO's, each with three levels of burstiness, by adjusting the transition probability λ_1 from Figure 11.(a)). For each arrival process we consider three scaling processes as MMOO's, each with three levels of burstiness, by adjusting μ_1 from Figure 11.(b), while keeping the same average rate of .75 (for the values of all the parameters see the caption).

Interestingly, the figure indicates that the burstiness in the arrivals completely dominates the burstiness in the scaling processes (the plots with different scalings' burstiness are visually indistinguishable, for all four arrival cases). This phenomenon can be justified by the independence assumption between arrival and scaling processes, i.e., unless some forms of correlations exist between the two (e.g., always drop when bursty traffic occurs), the scalings' burstiness has only a negligible impact on the arrivals' burstiness. The figure also illustrates that when there is almost no burstiness in the arrivals



(a) Poisson (b) MMOO

Fig. 12. Scaling of end-to-end delay bounds with Theorem 1 and the method of node-by-node analysis (arrival process with rate 1 (Poisson in (a) and two MMOO's in (b) with $P = 2$ (less bursty) and $P = 3$ (more bursty) and $\lambda_1 = \frac{1}{P}$), Bernoulli (0.75) scaling processes, violation probability $\varepsilon = 10^{-3}$).

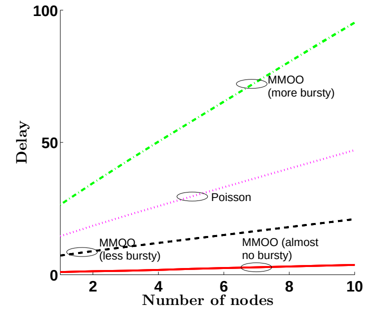


Fig. 13. Arrivals' burstiness dominates scalings' burstiness (three MMOO scalings ($\mu_1 =$ .67, .75, .99, average=.75) for each arrivals with average 1 (three MMOO's with $\lambda_1 = .7, .4, .01$, $P = 2$), $\varepsilon = 10^{-3}$).

(i.e., $\lambda_1 \to 0$ which means that the arrival process looks roughly like a periodic source of 2 (the peak rate P) packets every two time units) the delays converge to roughly zero time units. As this would actually be the case for such periodic arrivals, the figure provides evidence that the delay bounds from Theorem 1 are reasonably accurate.

V. Conclusions

Since flow transformations are manifold and frequent, we believe that this paper opens up the modelling scope of the stochastic network calculus widely. To that end, we have introduced a versatile stochastic scaling element and have shown how networks with flow transformations could still be expressed in convolution-form. Consequently, as demonstrated analytically as well as by numerical examples, the fundamental scaling properties of the network calculus are retained. This paper lays the theoretical foundation for a rich set of new applications of network calculus, ranging from lossy networks over dynamic routing or network coding scenarios.

For instance, in a network coding scenario [1] with multiple bursty flows, one may model the coding element with scaling elements as correlated loss processes, for each of the flows, which essentially remove all but the last packet in a tuple of packets to be coded. It becomes of interest to compare this approach with an alternative and recent approach of the stochastic network calculus for computing queueing measures in acyclic networks with network coding [24]. Another particularly interesting research problem is to embed the scaling element proposed in this paper within the data information network calculus from [23]. While some of the applications of scaling elements to various network scenarios may require further thought, especially when dealing with cycles, many opportunities lie ahead.

References

[1] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung. Network information flow. *IEEE Transactions on Information Theory*, 46(4):1204–1216, July 2000.
[2] J.-Y. Le Boudec and P. Thiran. *Network Calculus*. Springer Verlag, Lecture Notes in Computer Science, LNCS 2050, 2001.
[3] A. Burchard, J. Liebeherr, and F. Ciucu. On $\Theta(H \log H)$ scaling of network delays. In *Proceedings of IEEE Infocom*, May 2007.
[4] S. Chakraborty, S. Kuenzli, L. Thiele, A. Herkersdorf, and P. Sagmeister. Performance evaluation of network processor architectures: Combining simulation with analytical estimation. *Computer Networks*, 41(5):641–665, Apr. 2003.
[5] C.-S. Chang. *Performance Guarantees in Communication Networks*. Springer Verlag, 2000.
[6] F. Ciucu. Network calculus delay bounds in queueing networks with exact solutions. In *International Teletraffic Congress (ITC)*, pages 495–506, 2007.
[7] F. Ciucu, A. Burchard, and J. Liebeherr. Scaling properties of statistical end-to-end bounds in the network calculus. *IEEE Transactions on Information Theory*, 52(6):2300–2312, June 2006.
[8] E. O. Elliott. Estimates of error rates for codes on bursty-noise channels. *Bell System Technical Journal*, 42(9):1977–1997, Sept. 1963.
[9] M. Fidler. An end-to-end probabilistic network calculus with moment generating functions. In *IEEE International Workshop on Quality of Service (IWQoS)*, pages 261–270, 2006.
[10] M. Fidler and J. Schmitt. On the way to a distributed systems calculus: an end-to-end network calculus with data scaling. In *ACM Sigmetrics/Performance*, pages 287–298, 2006.
[11] E. N. Gilbert. Capacity of a bursty-noise channel. *Bell System Technical Journal*, 39(5):1253–1265, Sept. 1960.
[12] Y. Jiang and Y. Liu. *Stochastic Network Calculus*. Springer, 2008.
[13] H. Kim and J. Hou. Network calculus based simulation: theorems, implementation, and evaluation. In *IEEE Infocom*, Mar. 2004.
[14] L. Kleinrock. *Queueing Systems*, volume 1. John Wiley and Sons, 1975.
[15] A. Koubaa, M. Alves, and E. Tovar. Modeling and worst-case dimensioning of cluster-tree wireless sensor networks. In *Real-Time Systems Symposium*, pages 412–421, 2006.
[16] J. Liebeherr, A. Burchard, and F. Ciucu. Non-asymptotic delay bounds for networks with heavy-tailed traffic. In *IEEE Infocom*, 2010.
[17] J. Schmitt and U. Roedig. Sensor network calculus - a framework for worst case analysis. In *IEEE International Conference on Distributed Computing in Sensor Systems*, pages 141–154, 2005.
[18] J. B. Schmitt, F. A. Zdarsky, and L. Thiele. A comprehensive worst-case calculus for wireless sensor networks with in-network processing. In *IEEE Real-Time Systems Symposium (RTSS)*, 2007.
[19] T. Skeie, S. Johannessen, and O. Holmeide. Timeliness of real-time IP communication in switched industrial ethernet networks. *IEEE Transactions on Industrial Informatics*, 2(1):25–39, Feb. 2006.
[20] L. Thiele, S. Chakraborty, and M. Naedele. Real-time calculus for scheduling hard real-time systems. In *IEEE International Symposium on Circuits and Systems*, volume 4, pages 101–104, 2000.
[21] E. Wandeler, A. Maxiaguine, and L. Thiele. Quantitative Characterization of Event Streams in Analysis of Hard Real-Time Applications. *Real-Time Systems*, 29(2):205–225, March 2005.
[22] H. S. Wang and N. Moayeri. Finite-state Markov channel – a useful model for radio communication channels. *IEEE Transactions on Vehicular Technology*, 44(1):163–171, Feb. 1995.
[23] K. Wu, Y. Jiang, and G. Hu. A calculus for information-driven networks. In *IEEE International Workshop on Quality of Service (IWQoS)*, 2009.
[24] Y. Yuan, K. Wu, W. Jia, and Y. Jiang. Performance of acyclic stochastic networks with network coding. *IEEE Transactions on Parallel and Distributed Systems*, 99(PP), 2010.