Fast symbolic computation of the worst-case delay in tandem networks and applications

Anne Bouillard, Thomas Nowak

Fast Symbolic Computation of the Worst-Case Delay in Tandem Networks and Applications

Anne Bouillard, Thomas Nowak

Département d'Informatique, École Normale Supérieure, Paris, France

Abstract

Computing deterministic performance guarantees is a defining issue for systems with hard real- time constraints, like reactive embedded systems. In this paper, we use burst-rate constrained arrivals and rate-latency servers to deduce tight worst-case delay bounds in tandem networks under arbitrary multiplexing. We present a constructive method for computing the exact worst-case delay, which we prove to be a linear function of the burstiness and latencies; our bounds are hence symbolic in these parameters. Our algorithm runs in quadratic time in the number of servers. We also present an application of our algorithm to the case of stochastic arrivals and server capacities. For a generalization of the exponentially bounded burstiness (EBB) model, we deduce a polynomial-time algorithm for stochastic delay bounds that strictly improve the state-of-the-art separated flow analysis (SFA) type bounds.

Keywords: performance analysis, worst-case delay, network calculus, stochastic network calculus

1. Introduction

With the explosive growth of embedded systems, which often have hard real-time constraints, having efficient tools for computing worst-case performance bounds for those systems has become a necessity. Several methods for deriving network performance bounds have been subsumed under the network calculus formalism: From local and simple description of the system elements, one derives deterministic performance bounds for the network. Servers and the flows circulating in the network are described by service curves and arrival curves. These functions provide guarantees on the minimal amount of service provided by the servers and on the shape of incoming traffic. In the past, deterministic guarantees have provided the analytical framework for Internet QoS (DiffServ/IntServ) [1, 2], switched networks [3, 4, 5], and Video-on-Demand [6]. More recently, deterministic bounds have become especially useful in the context of large embedded networks

Email addresses: anne.bouillard@ens.fr (Anne Bouillard), thomas.nowak@ens.fr (Thomas Nowak)

like AFDX (Avionics Full Duplex) [7]. Among other methods for computing worst-case performance bounds, one can cite model-checking, which enables to compute accurate performance bounds but has a prohibitive algorithmic cost [8], or the trajectorial method [9], where worst-case delay bounds are given by the solution of a fixed-point equation. One main advantage of network calculus over these methods is its modularity and its scalability that permits to analyze large and complex networks.

Recent years have seen several advances for deducing tight network performance bounds. On one hand, there were several attempts to improve the computations of the end-to-end performances. Indeed, classically, the performances are computed using (min,plus) operations, which introduce a certain pessimism to the analysis. The methods used range from the exhaustive computations, which can be performed only for small networks [10], approximate methods that are exact for subclasses of networks [11]. More recently, exact methods [12, 13] using linear programming have been proposed. Unfortunately, the solution is given as the solution of an optimization problem, which makes the use of this method difficult for solving more general issues, such as optimizing on a parameter.

On the other hand, stochastic versions of network calculus have been defined. This stochastic network calculus (SNC) started with the work of Chang [14], and several models have since been derived. The main approach is to mix the basic results of network calculus with large deviation theory. Reviews of these results can be found in [15, 16, 17]. Many studies focus on the case of a single server and multiple flows. Different service policies are studied [18], and very sophisticated results show that SNC can be adapted to a variety of service policies [19]. Network analysis has received less attention. End-to-end delay bounds are computed in presence of disjoint cross-traffic in [20]. Also, the study of stochastic network calculus opens new fields of application for network calculus, such as smart-grids [21] and soft real-time systems [22].

In the present paper, we present two main contributions. The first concerns deterministic networks. When the arrival curves are burst-rate and the service curves are rate-latency, we present an explicit construction for the optimal residual service curve and exact worst-case delay of an arbitrary flow crossing a tandem network. In particular, we show that the residual service curve is rate-latency and that the latency (and hence also the worst-case delay) is a polynomial of degree one in the latencies of the servers and the bursts of the flows.

Secondly, we extend our analysis to the stochastic settings and compare it to results that can be obtained from the literature. Our algorithm produces stochastic delay bounds in polynomial time for the case of convex error probability functions, like the exponentially bounded burstiness (EBB) model. The bounds are always better than those coming from a separated flow analysis (SFA) of the network. We also discuss how

to adapt our results to more general stochastic frameworks and the difficulties that are involved.

The remaining of the paper is organized as follows. In Section 2, we give our model definition for deterministic networks. In Section 3 we present our algorithm to compute the worst-case delay in deterministic tandem networks. In Section 4, we show how the result can be used in the stochastic setting to optimize the error probability in tandem networks. The conclusion is in Section 5. The appendix contains serveral supplementary proofs and discussions of our approach.

2. Framework and Model

We denote by $\mathbb{N}$ the set of non-negative integers $\{0, 1, \ldots\}$ and for all $n \in \mathbb{N}$, we set $[\![n]\!] = \{1, \ldots, n\}$. For $x \in \mathbb{R}$, we set $(x)_+ = \max(x, 0)$. We write $\mathbb{R}_+$ for the set of non-negative reals.

2.1. General Model

While our model is in line with the standard definitions of networks calculus, we present only the material that is needed in this paper. A more complete presentation of the network calculus framework can be found in the reference books [23, 17].

Flows of data are represented by non-decreasing and left-continuous functions that model the cumulative processes. More precisely, if A represents a flow at a certain point in the network, $A(t)$ is the amount of data of that flow that crossed that point in the time interval $[0, t)$, with the convention $A(0) = 0$. More formally, let $\mathcal{F} = \{f : \mathbb{R}_+ \to \mathbb{R} \mid f \text{ is non-decreasing, left-continuous, and satisfies } f(0) = 0\}$.

A system $\mathcal{S}$ is a non-deterministic relation between input and output flows, where the number of inputs is the same as the number of outputs: $\mathcal{S} \subseteq \mathcal{F}^m \times \mathcal{F}^m$ and there is a one-to-one relation between the inputs and the outputs of the system, such that to each input flow corresponds one and only one output flow that is causal—no data is created inside the system—meaning that for $((A_i)_{i=1}^m, (B_i)_{i=1}^m) \in \mathcal{S}$, $\forall i \in [\![m]\!]$, $A_i \geq B_i$. The vector $((A_i)_{i=1}^m, (B_i)_{i=1}^m)$ is an *(admissible) trajectory* of $\mathcal{S}$ if $((A_i)_{i=1}^m, (B_i)_{i=1}^m) \in \mathcal{S}$. If $m = 1$, *i.e.*, the system has exactly one incoming and one outgoing flow, then we will also refer to it as a *server*.

Arrival curves. The notion of arrival curve is quite simple: The amount of data that arrived during an interval of time is a function of the length of this interval. More formally, let $\alpha \in \mathcal{F}$. A flow is constrained by the arrival curve α, or is α-constrained if $\forall s, t \in \mathbb{R}_+$ with $s \leq t$: $\quad A(t) - A(s) \leq \alpha(t - s)$.

A typical example of such arrival curve are the pseudo-affine *burst-rate* functions: $\alpha_{b,r} : 0 \mapsto 0;\ t \mapsto b + rt$, if $t > 0$. The burstiness parameter b can be interpreted as the maximal amount of data that can arrive simultaneously and the rate r as a maximal long-term arrival rate.

Service curves. The role of a service curve is to constrain the relation between the input of a server and its output. Let A be an cumulative arrival process to a server and B be its cumulative departure process. Several types of service curves have been defined in the literature, and the main types are the simple and strict service curves, which we now define.

We say that β is a *simple service curve* for a server $\mathcal{S}$ if $\forall (A,B) \in \mathcal{S}\colon \quad A \geq B \geq A * \beta$, with the convolution $A * \beta(t) = \inf_{0 \leq s \leq t} A(s) + \beta(t-s)$.

An interval I is a *backlogged period* for $(A,B) \in \mathcal{F} \times \mathcal{F}$ if $\forall u \in I,\ A(u) > B(u)$. The start of the backlogged period of an instant $t \in \mathbb{R}_+$ is $\text{start}(t) = \sup\{u \leq t \mid A(u) = B(u)\}$. As both A and B are left-continuous, we have $A\big(\text{start}(t)\big) = B\big(\text{start}(t)\big)$, and $\big(\text{start}(t), t\big]$ is always a backlogged period.

We say that β is a *strict service curve* for server $\mathcal{S}$ if $\forall (A,B) \in \mathcal{S}\colon\ A \geq B$ and for all backlogged periods $(s,t]\colon\ B(t) - B(s) \geq \beta(t-s)$. The name strict service curves implies a difference to simple service curves. Works on the comparisons between the different types of service curves can be found in [23, 24]. In this article, we will mainly deal with strict service curves, but will make use of the convolution $A * \beta$ used to define simple service curves.

A typical example of a service curve are the *rate-latency* functions: $\beta_{R,T} : t \mapsto R \cdot (t-T)_+$, where T is the latency until the server has to become active and R is its minimal service rate after this latency.

Note that a server $\mathcal{S}$ may not be deterministic, as the function β only corresponds to a guarantee on the service offered. Among this non-determinism, we will focus on two modes of operation:

- **Exact service mode:** During a backlogged period $(s,t]$, the service is *exact* if for all $u \in (s,t]$ we have $B(u) = A\big(\text{start}(t)\big) + \beta\big(u - \text{start}(t)\big)$. (In this case, $\text{start}(u) = \text{start}(t)$.)
- **Infinite service mode:** During an interval of time $(s,t]$, the service is *infinite* if $\forall u \in (s,t]\colon B(u) = A(u)$, *i.e.*, the server serves all data instantaneously.

For a system $\mathcal{S} \in \mathcal{F}^m \times \mathcal{F}^m$ with m inputs and outputs, we say that $\mathcal{S}$ offers a strict service curve β if the aggregate system is, that is, if

$$\forall \big((A_i)_{i \in [\![m]\!]}, (B_i)_{i \in [\![m]\!]}\big) \in \mathcal{S}\colon \left(\sum_{i \in [\![m]\!]} A_i, \sum_{i \in [\![m]\!]} B_i \right) \in \mathcal{S}(\beta).$$

We assume no knowledge about the service policy in this system (except that it is FIFO per flow).

Performance guarantees. In this article, we focus on the worst-case delay of a flow in a network. Let $(A,B) \in \mathcal{F}^2$ be an admissible trajectory of a server. The delay of a bit that arrives at time t is $d(t) =$

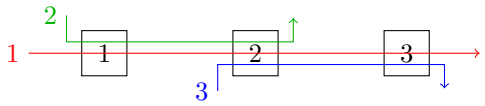

Figure 1: Tandem network with 3 servers and 3 flows.

$\sup\{d \in \mathbb{R}_+ \mid A(t) > B(t+d)\}$. The worst-case delay is then $d_{\max} = \sup_{t \geq 0} d(t)$. Graphically this is the maximal horizontal distance between A and B.

We are interested in the bit of data that suffers the maximal delay. This bit will be called in the following the *bit of interest*. Due to the left-continuity assumption, its formal definition is not straightforward. A bit of interest can be associated with a quantity of data x and for each process A having that bit of data, there must exists $t_{A,x}$ such that $\forall t > t_{A,x} \colon A(t) \geq x$ and $\forall t < t_{A,x} \colon A(t) < x$. It coincides with the infimum $t_{A,x} = \inf\{t \in \mathbb{R}_+ \mid A(t) \geq x\}$. We call $t_{A,x}$ the time that data x leaves flow A. Denote by $\mathrm{rg}(A) = \{x \in \mathbb{R}_+ \mid \exists t \in \mathbb{R}_+ \colon A(t) \geq x\}$ the *range* of A. The worst-case delay of trajectory (A, B) is then equal to $d_{\max} = \sup\limits_{x \in \mathrm{rg}(A)} \big(t_{B,x} - t_{A,x}\big)$.

2.2. Tandem Networks

Throughout the paper, we consider networks in tandem only: Assume a sequence of n servers numbered from 1 to n and m flows, flow i crossing servers s_i to d_i in order, with $s_i \leq d_i$. Figure 1 depicts a tandem network with 3 servers and 3 flows. We have $s_1 = s_2 = 1$, $d_2 = s_3 = 2$ and $d_1 = d_3 = 3$. For a server j, we set $\mathrm{Fl}(j)$ the set of flows that crosses that server. In the example network, $\mathrm{Fl}(1) = \{1, 2\}$, $\mathrm{Fl}(2) = \{1, 2, 3\}$, and $\mathrm{Fl}(3) = \{1, 3\}$.

Concerning the arrival and service guarantees, we assume the following:

- Flow $i \in [\![m]\!]$ has a burst-rate arrival curve $\alpha_i : 0 \mapsto 0;\ t \mapsto b_i + r_i t$.
- Server $j \in [\![n]\!]$ offers the strict rate-latency service curve $\beta_j : t \mapsto R_j(t - T_j)_+$.

We aim at finding the worst-case delay of a given flow of an admissible trajectory. Formally, we call trajectory a collection of cumulative arrival and departure processes for each flow at each server it crosses. We say that a trajectory is admissible if the processes satisfy the network calculus constraints given by the arrival curves of the flows and the strict service curves of the servers. We denote by $F_i^{(j-1)}$ (resp. $F_i^{(j)}$) the cumulative arrival (resp. departure) process of flow i at server j. The trajectory $(F_i^{(j)})$ is admissible if

- $\forall i \colon F_i^{(s_i - 1)}$ has arrival curve α_i and
- $\forall j \colon ((F_i^{(j-1)}), (F_i^{(j)}))$ has a strict service curve β_j.

We assume that we focus on the worst-case delay of a flow that crosses every server. Assume it is flow 1. The goal is to compute $\sup_{x \in \mathrm{rg}(F_1^{(0)})} t_{F_1^{(n)},x} - t_{F_1^{(0)},x}$.

To this aim, let us introduce the dates we will focus on for given admissible trajectory and bit of interest x. Set $t_n = t_{B,x}$, the departure date of the system of the bit of interest. Then, t_j, $j \leq n$ are defined recursively by setting $t_{j-1} = \mathrm{start}_j(t_j)$ where start_j is the start of the backlogged period at server j.

3. Worst-Case Delay in Deterministic Tandem Networks

In this section we present the main contribution of this paper, that is, the construction of a worst-case delay trajectory for tandem networks. We prove that we need to focus on limited number of trajectories only. Then, the delay of the bit of interest can be explicitly computed for these trajectories. Moreover, it is not necessary to compute the delays for all those trajectories, as a greedy algorithm enables to find one with the largest delay. Interestingly, the worst-case delay turns out to be an affine function of the burstiness of the arrivals and the latencies of the servers.

We first present several properties of a worst-case trajectory that can be assumed to hold, and then present the construction of the worst-case trajectory that lead to the worst-case delay.

3.1. Properties of a Worst-Case Scenario

Several scenarios can lead to the worst-case delay. In this subsection, we show that we can construct a worst-case scenario with certain additional properties that facilitate its analysis. This permits to consider only a limited number of scenarios to find the worst-case delay. Let us first state them and next comment them.

($\mathbf{H_1}$) Service policy is SDF (shortest-to-destination first): for two flows i and j, if $d_i < d_j$, then flow i is served with higher priority than flow j.

($\mathbf{H_2}$) Server j has the unique backlogged period (t_{j-1}, t_j) and provides infinite service outside its backlogged period.

($\mathbf{H_3}$) Each server provides exact service in its backlogged period and $t_j - t_{j-1} \geq T_j$.

($\mathbf{H_4}$) The new arrivals at server j are maximal from time t_{j-1} on and zero before that.

($\mathbf{H_5}$) The bit of interest that suffers the longest delay is b_1, the burstiness of flow 1.

It is proved in [25, Theorem 8] that any admissible trajectory can be replaced by another satisfying ($\mathbf{H_1}$) and ($\mathbf{H_2}$) and such that the delay of the bit of interest increases. The transformation does not modify the arrival processes and the global service offered by a server during its backlogged period. The result is quite intuitive: Firstly, the SDF policy is known for being among the ones that can make a cyclic network unstable. Secondly, we focus on the delay suffered by one bit of interest. Its delay depends only on the service provided in the backlogged periods it belongs to. So these backlogged periods should be charged as much as possible for the worst-case delay. Hence, the upstream servers should transmit their backlogs instantaneously and simultaneously to the newly backlogged server. A direct consequence is that the bit of interest leaves server j at time t_j.

Lemma 1. *For any admissible trajectory $\mathcal{T}$, there exists an admissible trajectory $\tilde{\mathcal{T}}$ satisfying ($\mathbf{H_3}$) and that does not modify the arrival processes and whose maximum delay is larger.*

Proof. Let $\mathcal{T} = (F_i^{(j)})$ be a trajectory and B the bit with the worst-case delay. Suppose there exists a backlogged period (t_{j-1}, t_j) where the service offered by server j is not exact. The quantity $\sum_{i\in \mathrm{Fl}(j)}(F_i^{(j)}(t_j) - F_i^{(j-1)}(t_{j-1})) = B$ is the quantity of data served during $[t_{j-1}, t_j)$ by server j. There exists t' such that $\beta_j(t') = B$ and, because β_j is a strict service curve for server j, we have $t' > t_j - t_{j-1}$. Set $d = t' - t_j + t_{j-1}$. As a consequence, one can replace $F_i^{(j)}$ by $\tilde{F}_i^{(j)}$, which is modified only in its backlogged period (t_{j-1}, t_j). The backlogged period becomes $(t_{j-1}, t_j + d)$, the service is exact and for all $i \in \mathrm{Fl}(j)$, $\tilde{F}_i^{(j)}(t_j + d) = F_i^{(j)}(t_j)$. Replace every cumulative function $F_i^{(k)}$ after t_j by $\tilde{F}_i^{(k)}(t) = F_i^{(k)}(t + d)$. The trajectory is still admissible and the delay of bit B is increased by d. Note that if $B = 0$, then one can take $t' = T_j$. □

Lemma 2. *For any admissible trajectory $\mathcal{T}$, there exists an admissible trajectory $\hat{\mathcal{T}}$ satisfying ($\mathbf{H_4}$) and whose maximum delay is larger.*

Proof. From ($\mathbf{H_2}$), one can assume without loss of generality that (t_{j-1}, t_j) is the unique backlogged period for server j, and then that if flow i starts at server j, then $F_i^{(s_i-1)}(t) = 0$ for all $t \le t_{j-1}$. Let $\mathcal{T} = (F_i^{(j)})$ and consider the trajectory $\tilde{\mathcal{T}} = (\tilde{F}_i^{(j)})$ where for all $t \ge 0$: $\tilde{F}_i^{(s_i-1)}(t) = \alpha_i((t - t_{s_i-1})_+)$ and the rest of the trajectory is the same as $\mathcal{T}$. Note that by doing so, the arrival time of the bit of interest decreases.

As the arrival processes increase, for server 1, (t_0, t_1) is still a backlogged period, and the backlog transmitted to server 2 at time t_1 is greater than for trajectory $\mathcal{T}$. By induction, this is also true at each server: (t_{j-1}, t_j) is still a backlogged period for server j and the backlog transmitted at time t_j to server $j + 1$ increases. Then t_n is still in the backlogged period of server n, and if all the backlog is served at time t_n, then the departing time of the bit of interest is unchanged, hence the result. □

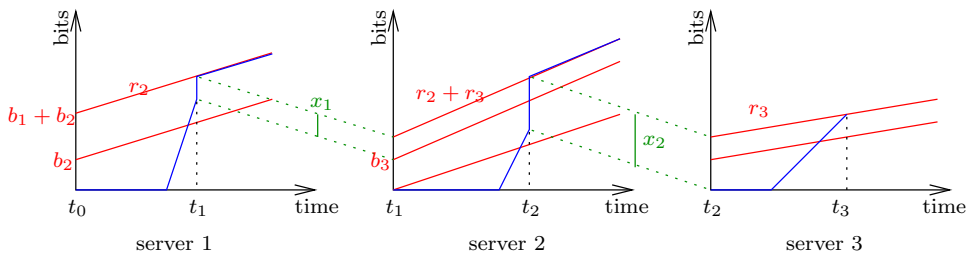

Figure 2: Trajectories satisfying $(\mathbf{H_1}), \ldots, (\mathbf{H_5})$.

Lemma 3. *Consider a trajectory satisfying* $(\mathbf{H_1})$–$(\mathbf{H_4})$. *There exists a trajectory* $\hat{\mathcal{T}}$ *where bit* b_1 *has the maximum delay.*

Proof. Let x the bit of flow 1 that suffers the largest delay. First, if $x \leq b_1$, then x and b_1 arrive in the system at the same time. So the policy is FIFO per flow, bit b_1 leaves the network after x.

Now, suppose that $x > b_1$. One can assume without loss of generality that $t_0 = 0$ and $F_1^{(0)}(t) \leq x$ for all $t \geq 0$: the arrival of flow 1 after The arrival of bit x has no influence on the delay of bit x.

If there is no server j (including server n) for which the backlog transmitted to the next server is 0 at time t_j, then bit x does not suffers the largest delay: There exists a bit $y < x$ of flow 1 that exits at the same time, but arrives strictly before. Bit y is transmitted at the same time as x.

Now suppose that there is a server j that transmits no backlog at time t_j. This means that there is an $\varepsilon > 0$ such that during the interval of time $[t_j - \varepsilon, t_j]$, flow 1 is served at rate

$$\rho = R_j - \sum_{i \in \mathrm{Fl}(j) \setminus \{1\}} r_i > r_1, \tag{1}$$

since $(\mathbf{H_1}), \ldots, (\mathbf{H_4})$ are satisfied. The delay of bit x can be decomposed in $d = d_1 + d_2$, where $d_2 = t_n - t_j$ and $d_1 = t_j - t'$, where t' is the arrival time of bit x at server 1.

Replace t_k by $t_k - \varepsilon$ for all $k > j$ and replace the arrival processes accordingly (if $s_i > j$, then $\tilde{F}_i^{(s_i-1)}(t) = F_i^{(s_i-1)}(t + \varepsilon)$). By (1), the delay of bit $y = x - \rho\varepsilon$ is $d_1' + d_2$, where $d_1' < d_1$. As a consequence, x does not suffer the largest delay. This is true for all $x > b_1$, and, hence, b_1 suffers the largest delay for some trajectory. □

As the bit of interest is b_1, we can and will set the rate of the flow of interest to $r_1 = 0$.

Figure 2 illustrates the properties $(\mathbf{H_1}), \ldots, (\mathbf{H_5})$ a scenario should have for the analysis in the next section. The only non-determinism now are the start times t_j of the backlogged periods, which also define the bursts that are transmitted from a server to the next one. In server 1, the dashed line is $F_2^{(0)}$, the plain

line above is $F_1^{(0)} + F_2^{(0)}$, and the bottom is $F_1^{(1)} + F_2^{(1)}$. When $F_2^{(0)} \leq F_1^{(1)} + F_2^{(1)}$, flow 2 is not backlogged in server 1 (because of the SDF service policy). Then at time t_1, the burst of size x_1 transmitted belongs to flow 1 only. The same hold for server 2 (from bottom to top, according to the SDF policy, $F_2^{(1)}$, $F_2^{(1)} + F_3^{(1)}$, $F_2^{(1)} + F_3^{(1)} + F_1^{(1)}$ are drawn). At time t_2, the burst of size x_2 is sent to server 3, which is shared between flows 1 and 3.

3.2. Computation of the Worst-Case Scenario

Using the properties above, we can build a worst-case trajectory with the following properties: Each server j has only two modes: infinite server or exact server (during the backlogged period (t_{j-1}, t_j)). This means that at time t_{j-1}, server $j-1$ transmits all its backlog to server j. Furthermore, the flows i entering the network at server j send their burst b_i at time t_{j-1}. Afterward, the arrival rate of data at server j is exactly $\sum_{i \in \mathrm{Fl}(j)} r_i$.

The service policy is SDF, but it should be clear that flows crossing server j and having the same destination do not need to be distinguished. We will introduce notation that distinguishes flows only by their destination.

To compute the worst-case trajectory, we only need to decide when the servers are exact. For example, if from time t_{j-1} server $j-1$ serves as an infinite server, it will transmit data to server j, and the delay from time t_{j-1} depends only on the quantity of these data for any destination and on the parameters of servers $j, \ldots, n$. We denote this delay by $\Delta_j(x_j, \ldots, x_n)$. It is the delay from server j of the bit of interest, that is $t_n - t_{j-1}$ if x_k is the burst transmitted at t_{j-1} to server j by the flows ending at server k, for $k \geq j$. Finally, the worst-case delay is $D = \Delta(0, \ldots, 0)$. We will show by induction that:

(**A**) Δ_j is affine in the x_k, T_k, and b_i. More precisely we can write $\Delta_j(x_j, \ldots, x_n) = D_j + \sum_{k=j}^{n} \frac{x_k}{\mathcal{R}_j^k}$, where $\mathcal{R}_j^k$ only depends on the R_k and r_i, and D_j is a polynomial of degree 1 in the T_k and b_i (with coefficients depending on the R_k and r_i only).

(**B**) $\mathcal{R}_j^k \geq \mathcal{R}_j^{k+1}$.

This inequality (**B**) is quite intuitive: the coefficient $\mathcal{R}_j^k$ roughly corresponds to the service rate of flows ending at server k in the servers $j, \ldots, k$. It means that a flow that has higher priority received a larger service rate than a flow with lower priority.

Algorithm 1 outputs the worst-case delay of the tandem network as an affine function

$$D = \sum_{j \in [\![n]\!]} \lambda_j T_j + \sum_{i \in [\![m]\!]} \mu_i b_i.$$

It uses the following additional notations:

- $b_j^k = \sum_{i,\ s_i=j, d_i=k} b_i$ size of the burst arriving at server j at time t_{j-1} belonging to flows starting at server j and ending at server k;
- $r_j^k = \sum_{i,\ s_i \le j, d_i=k} r_i$ arrival rate at server j for all flows ending at server k and crossing j (remember that we fixed $r_1 = 0$);

Algorithm 1: Worst-case delay algorithm

begin
  $\mathcal{R}_n^n \leftarrow R_n - r_n^n$;
  for j **from** $n-1$ **to** 1 **do**
    $k \leftarrow n$;
    while $\mathcal{R}_{j+1}^k < (R_j - \sum_{\ell=j}^k r_j^\ell) \left(1 + \sum_{\ell=k+1}^n \frac{r_j^\ell}{\mathcal{R}_{j+1}^\ell}\right)^{-1}$ **do**
      $\mathcal{R}_j^k \leftarrow \mathcal{R}_{j+1}^k$;
      $k \leftarrow k-1$;
    for ℓ **from** j **to** k **do** $\mathcal{R}_{j+1}^\ell \leftarrow (R_j - \sum_{\ell'=j}^k r_j^{\ell'}) \left(1 + \sum_{\ell'=k+1}^n \frac{r_j^{\ell'}}{\mathcal{R}_{j+1}^{\ell'}}\right)^{-1}$; ;
  for j **from** 1 **to** n **do** $\lambda_j \leftarrow 1 + \sum_{\ell=j}^n r_j^\ell / \mathcal{R}_j^\ell$;
  ;
  for i **from** 1 **to** m **do** $\mu_i \leftarrow (\mathcal{R}_{s_i}^{d_i})^{-1}$;

Theorem 1. *Consider a tandem network of n servers offering rate-latency service curves $\beta_j : t \mapsto R_j(t - T_j)_+$, $j \in [\![n]\!]$ and crossed by m flows with pseudo-affine arrival curves $\alpha_i : t \mapsto b_i + r_i t$, $i \in [\![m]\!]$. The expression $D = \sum_{j \in [\![n]\!]} \lambda_j T_j + \sum_{i \in [\![m]\!]} \mu_i b_i$ where the λ_j and μ_i are the outputs of Algorithm 1 is the worst-case delay of a tandem network. The coefficients λ_j and μ_i depend only on $(r_i)_{i \in [\![m]\!]}$ and $(R_j)_{j \in [\![n]\!]}$. D can be effectively computed in time $O(n^2 + m)$.*

Due to space limitations, the proof of this theorem is postponed to Appendix A.

From this theorem, a (simple) service curve can be derived for flow 1. Intuitively, it is the inverse of the function $\Delta(0, 0, \ldots, 0) = D(b_1)$ seen as a function of b_1. It is then a rate-latency function. The next theorem proves this and states that this is the best service curve that can be found for flow 1 crossing this system. Note that the rate of this function is easyly computed by $\min_{j \in [\![n]\!]} R_j - \sum_{i \in \mathrm{Fl}(j)} r_i$, the latency is $D(0)$.

Ser. 2	$\frac{1}{R_2-r_1-r_2} \geq \frac{1}{R_3-r_2}$		$\frac{1}{R_2-r_1-r_2} < \frac{1}{R_3-r_2}$		
Ser. 1	$\frac{1}{R_1-r_1} \geq \frac{1}{R_2-r_1-r_2}$	$\frac{1}{R_1-r_1} > \frac{1}{R_2-r_1-r_2}$	$\frac{1}{R_1-r_1} \geq \frac{1}{R_3-r_2}$	$\frac{R_3}{(R_2-r_1)(R_3-r_2)} \leq \frac{1}{R_1-r_1} < \frac{1}{R_3-r_2}$	otherwise
a	$b_2+b_1+r_2T_1$	0	$b_1+b_2+r_2T_1$	0	0
b	$b_3+(r_2+r_3)T_2$	$b_1+b_2+b_3$ $+r_2T_1+(r_2+r_3)T_2$	0	0	0
c	0	0	r_3T_2	r_2T_2	$b_2+r_2(T_1+T_2)$
d	0	0	$b_3+r_3T_2$	$b_3+b_1+r_3T_2$	$b_3+b_1+r_3T_2$

Table 1: Worst-case delay for the tandem network in Figure 1.

Theorem 2. *The curve $\beta_{R,T}$ with $R = \min_{j\in[\![n]\!]} R_j - \sum_{i\in \mathrm{Fl}(j)} r_i$ and $T = D(0)$ is a service curve for flow 1 and there is no function $\beta \in \mathcal{F}$ greater than $\beta_{R,T}$ that is a service curve for flow 1.*

Proof. Let us first prove that there exists no $\beta > \beta_{R,T}$ that is a service curve for the system. Suppose that $\beta(s) > \beta_{R,T}(s)$. Then there exists $u > s$ such that $\beta(s) = \beta_{R,T}(u) = b$. By Theorem 1, the worst-case delay for the arrival process $A : t \mapsto b\mathbf{1}_{t>0}$ is u. But $A * \beta(s) = b$, so the delay obtained using β is less than u. So β cannot be a service curve.

We now show that $\beta_{R,T}$ is a service curve. Let A be a cumulative arrival process for flow 1 and B its output process. Let $t_n > 0$, and define t_0 as in the notations, that is, $t_0 = \big(\mathrm{start}_1 \circ \mathrm{start}_2 \circ \cdots \circ \mathrm{start}_n\big)(t_n)$. Set $b = B(t_n) - A(t_0)$. Then from Theorem 1, $t_n - t_0 \leq \Delta(0,\ldots,0) = D(b_1)$, so $b_1 \geq \beta_{R,T}(t_n - t_0)$. As a consequence, $B(t_n) = A(t_0) + b_1 \geq A(t_0) + \beta_{R,T}(t_n - t_0) \geq A * \beta_{R,T}(t_n)$. □

3.3. Example

Consider the example of Figure 1 and assume we want to compute the delay of flow 3.

We compute Δ_3, Δ_2 and then Δ_1.

Server 3: $\Delta_3(x_3) = T_3 + \frac{x_3+r_3T_3}{R_3-r_3}$ (we do not take r_1 into account as the worst-case delay is obtained at the burst);

Server 2: if $\frac{1}{R_2-r_2-r_3} \geq \frac{1}{R_3-r_3}$, then $\Delta_2(x_2,x_3) = T_2 + T_3 + \dfrac{x_2+r_2T_2+b_2+x_3+r_3T_2}{R_2-r_2-r_3}$. Otherwise $\Delta_2(x_2,x_3) = T_2 + T_3 + \dfrac{R_3(x_2+r_2T_2)}{(R_2-r_2)(R_3-r_3)} + \dfrac{b_3+x_3+r_3T_2}{R_3-r_3}$.

Server 1: According to our construction, there are be 5 cases to consider. The delay has a general form $D = T_1+T_2+T_3+\frac{r_3T_3}{R_3-r_3}+\frac{a}{R_1-r_2}+\frac{b}{R_2-r_2-r_3}+\frac{cR_3}{(R_2-r_2)(R_3-r_3)}+\frac{d}{R_3-r_3}$, where a,b,c,d are shown in Table 1.

3.4. Comparison with the State of the Art

The problem of computing exact worst-case delays is a long-standing problem in the literature. As far as arbitrary multiplexing is concerned, this problem has been extensively studied in [10], where the authors first present the difficulties of obtaining closed formulas, even for the trivial case of two flows crossing two servers.

Then they completely study the case of the tandem network depicted in Figure 1. Their analysis exhibits 5 cases leading to 4 different formulas. The worst-case scenario the author construct is slightly different than the one constructed in Section 3.3 as they do not assume that the backlogged periods are disjoint, leading to a case distinction that is more intricate than the one we presented. However, we checked that the cases we found for this network are exactly the same as theirs, and so we compute exactly the same delays. Note that we have 5 different cases instead of 4 because we take into account the service time of the burst of the flow of interest. Apart from this service time, the third and fourth cases are similar.

Secondly, some work has been done to compute worst-case delays in feed-forward network using linear programs in [12]. In the case of tandem network the authors show that it can be done by solving a single linear program making the procedure polynomial in time. They can handle more general arrival and service curves. We believe that our approach can easily be generalized to obtain good approximations in general feed-forward network, which is, due to space limitation, beyond the scope of this article. The generalization to general concave or convex arrival and service curves would be more demanding, as delay would not be an affine function anymore. One drawback of the method in [12] is that no intuition is provided for the construction of the worst-case trajectory and then a slight modification of the parameters cannot be handled, which is a major issue with this method. In particular, computing the residual service curve of the flow of interest is intractable with the method of [12], as is explained in [26]. Theorem 2 states that it can be done in polynomial time with our algorithm.

Moreover it is not possible to use the method of [12] directly to optimize on a parameters b_i or T_j. We will provide a more explicit application in the next section, where we give an application to Stochastic Network Calculus, that requests to optimize on the parameters. We explain in Appendix B why we cannot apply [12] to solve this problem.

4. Application to Stochastic Arrivals and Servers

In this section, we apply our results to a stochastic extension of network calculus. The results of this section provide an upper bound on the probability that a given stochastic system fails to satisfy a given delay bound. Along the way, we introduce a new definition for stochastic service curves, which is based on the notion of variable capacity nodes (VCNs), that plays the role of strict service curves. Indeed, until now, strict service curves have not received a lot of attention, with the notable exception of guaranteed rate curves (curves of type $t \mapsto Rt$).

In Section 4.1, we give the definitions of stochastic network calculus, including the new stochastic VCN

service curves. In Section 4.2, we present the state-of-the-art method for deriving delay bounds in stochastic network calculus, and discuss a new result on the composability of stochastic curves. Section 4.3 introduces our new method for deriving delay bounds and proves its correctness. We provide numerical examples, including quantitative comparison to the state-of-the-art method, in Section 4.4. We discuss the stochastic model assumptions in Appendix C.

4.1. Stochastic Arrivals and Servers

Stochastic network calculus is a generalization of deterministic network calculus, in which the arrival processes and departures processes are stochastic processes. We now give the definition of the stochastic model we use in this paper.

4.1.1. Stochastic Arrival Curves

Let $\alpha \in \mathcal{F}$ and $\varepsilon_t : \mathbb{R}_+ \to \mathbb{R}_+$ for every $t \geq 0$. We say that the arrival process A is $\langle \varepsilon_t, \alpha \rangle$-constrained if we have: $\forall t\, \forall \sigma\colon\ \mathbf{P}\big(\exists u \leq s \leq t\colon A(s) - A(u) > \alpha(s-u) + \sigma\big) \leq \varepsilon_t(\sigma)$, where "$\exists u \leq s \leq t$" signifies the existence of u and s with $u \leq s \leq t$. We call ε_t the error probability function. This definition is called the maximum virtual backlog centric (m.b.c.) model by Jiang and Liu [15]. An important example process are families of burst-rate curves with a common arrival rate. In this case, we can choose the failure probability function to directly depend on the burst parameter b:

$$\forall t\, \forall b\colon\ \mathbf{P}\big(u \leq s \leq t \text{ and } A(s) - A(u) > b + r(s-u)\big) \leq \varepsilon_t(b). \tag{2}$$

An important family of typical choices for the error probability function, originating from an exponentially bounded burstiness (EBB) assumption, is of the form

$$\varepsilon_t(b) = Ke^{ct-ab}, \tag{3}$$

with positive real constants a, c, and K. We call it the time-dependent EBB (tEBB) model. The EBB assumption was originally introduced by Yaron and Sidi [27] in the form $\forall s,t\colon\ \mathbf{P}\big(A(t) - A(s) > r_1 \cdot (t-s) + b\big) \leq Me^{-ab}$ to compute stochastic backlog bounds. They also showed [27, Theorem 1] that it is possible to translate this form of EBB constraint of t.a.c.-type (traffic amount centric) into a constraint of v.b.c.-type (virtual backlog centric [28]): For all $r_2 > r_1$ there exists an N such that $\forall t\colon\ \mathbf{P}\big(\exists s\colon A(t) - A(s) > r_2 \cdot (t-s) + b\big) \leq Ne^{-ab}$. It is further possible [15, Theorem 3.24(ii)] to translate this into a constraint of type (3) for burst-rate arrivals with rate $r > r_2$. It is hence possible to translate an EBB constraint in its

weakest form into a stronger form, described in (2) and (3), by sacrificing an epsilon of its rate bound. On the other hand, in a heavy load scenario, this translation cannot be done, as there is no epsilon rate to spare.

4.1.2. Stochastic VCN Service Curves

We now go on to define a similar stochastic definition for strict service curves. A desirable property of a service curve process would be to be independent of the arrival process. To come up with such a definition is not obvious from the definition of a strict service curve, which depends on the start times of backlogged periods, which clearly depend on the input. Fortunately, there exists an almost equivalent definition for strict service curves, the *variable capacity node* (VCN) service curve. For the classes of service curves we are interested in, these notions coincide (see [24] for mode details).

In deterministic network calculus, variable capacity nodes are defined as follows: A server offers a VCN service curve β if there is some $C \in \mathcal{F}$ such that $\forall s < t\colon\ B(t) = \inf_{0\le u\le t} A(u) + C(t) - C(u)$ and $C(t)-C(s) \ge \beta(t-s)$. Intuitively, $C(t)$ represents the amount of service offered by the server up to time t, and the relation between A, B, and C is a direct consequence of the Lindley equation, that is widely used in the context of queueing theory. For rate-latency curves, the notions of strict service and VCN service coincide (see next theorem). However, the extension to the stochastic setting is more natural to define with VCN service curves as it is not necessary to distinguish backlogged and non-backlogged periods of a stochastic process.

Theorem 3 ([24, Theorem 3])**.** *Let β be a rate-latency curve. Then a deterministic server $\mathcal{S}$ offers β as a strict service curve if and only if it offers β as a VCN service curve.*

Starting from this formulation, it is now easy to adapt the notion of service curve to the stochastic case:

Definition 1. Let $\beta \in \mathcal{F}$ and $\eta_t : \mathbb{R}_+ \to \mathbb{R}_+$ for every $t \ge 0$. The service C offers a stochastic variable capacity node β with failure probability η if $\forall t\,\forall\theta\colon\ \mathbf{P}\big(\exists u \le s \le t\colon C(s) - C(u) < \beta(s-u) - \theta\big) \le \eta_t(\theta)$. In this case, we will say that C offers $\langle \eta_t, \beta\rangle$.

The equality $B(t) = \inf_{0\le u\le t} A(u) + C(t) - C(u)$ does not appear in the definition, as we assume that B is a stochastic process that deterministically depends on A and C.

Similarly to arrival curves, if the service curve is rate-latency, one can make η_t depend on the latency T, *i.e.*, $\forall t\,\forall T\colon\ \mathbf{P}\big(\exists u \le s \le t\colon C(s) - C(u) < R(s-u-T)_+\big) \le \eta_t(RT)$.

In analogy to the tEBB constraint for arrival curves, we are led to consider a time-dependent exponentially

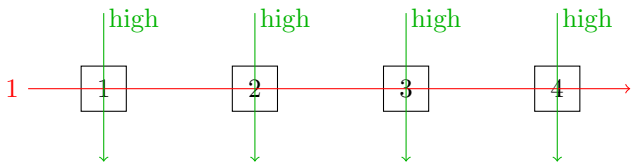

Figure 3: Tandem network with crossing high-priority flows.

bounded latency (tEBL) constraint, *i.e.*, an error probability function for VCN service curves of the form

$$\eta_t(RT) = Le^{ft-gRT}, \tag{4}$$

with positive real constants f, g, and L.

An example where this kind of stochastic service guarantees naturally occur is the case of crossing high-priority flows that, themselves, have stochastically constrained arrivals; see Figure 3. Take, *e.g.*, a single server with deterministic VCN service $\beta(t) = Rt$ and a crossing stochastic tEBB arrival with $\alpha(t) = r_h t$ and error probability function $\varepsilon_t(b) = Me^{ct-ab}$. If the server always serves the stochastic arrival with highest priority, then the residual service provided is tEBL constrained with service curve $\tilde{\beta}(t) = (R-r_h)t$ and error probability function $\eta_t\big((R-r_h)T\big) = Me^{ct-a(R-r_h)T}$.

4.2. State-of-the-Art Delay Bounds

In this section, we present the state of the art for deriving probabilistic delay bounds in feed-forward networks. The following theorem sums up the single-node performance bounds in stochastic network calculus that are used as the basis for performance bounds of a network. It uses the definitions of the vertical distance $v(\alpha,\beta) = \sup_{s\geq 0}\big(\alpha(s)-\beta(s)\big)$, horizontal distance $h(\alpha,\beta) = \sup\{d \geq 0 \mid \exists t \geq 0\colon \alpha(t) > \beta(t+d)\}$, and the deconvolution $\alpha \oslash \beta(t) = \sup_{\tau\geq 0}\big(\alpha(t+\tau)-\beta(\tau)\big)$ of α and β. It is a generalization of existing results (cf. Theorems 5.1, 5.4, and 5.12 in [15]) to the case of VCN service curves. Our algorithm does not rely on them.

Theorem 4. *Let A be the arrival process that is constrained by $\langle \varepsilon_t, \alpha\rangle$ and B the departure process that offers a VCN service of $\langle \eta_t, \beta\rangle$.*

(i) Worst-case backlog: For all $t \geq 0$ and all $\gamma \geq 0$, $\mathbf{P}\big(\exists s \leq t\colon A(s) - B(s) > v(\alpha,\beta)+\gamma\big) \leq \varepsilon_t * \eta_t(\gamma)$;

(ii) Worst-case delay: For all $t \geq 0$ and all $\gamma \geq 0$, $\mathbf{P}\big(\exists s \leq t\colon A(s-h(\alpha+\gamma,\beta)) - B(s) > 0\big) \leq \varepsilon_t * \eta_t(\gamma)$.

(iii) Constraint propagation: For all $t \geq 0$ and $\gamma \geq 0$, $\mathbf{P}\big(\exists u \leq s \leq t\colon B(s) - B(u) > \alpha \oslash \beta(s-u)+\gamma\big) \leq \varepsilon_t * \eta_t(\gamma)$.

Theorem 4, together with similar results on the concatenation, superposition, and leftover service curve (cf. [15, Theorems 5.36, 5.42, 5.64]), allows for a worst-case delay analysis very similar to the deterministic case. Note, however, that the notion of stochastic worst-case delay bound is stronger than that used in [15], which consists in bounding the probability of a delay occuring at a specific time instant. In particular, separated flow analysis (SFA) techniques can be adapted to the stochastic setting. We discuss the SFA technique and its generalization to stochastic arrivals and servers in more detail together with a comparison to our new algorithm in Section 4.3.

4.3. New Stochastic Delay Bounds

This section presents our algorithm for deriving upper bounds on the probability that the maximum delay passes a given delay bound. It requires the arrivals and servers to be burst-rate and rate-latency, respectively, and the error probability functions to be convex in the burst parameters σ_i and the latency parameters θ_j. These assumptions allow us to use convex programming to find the optimal bound among the family of bounds our analytic results of this section establish. Our algorithm is based on Algorithm 1, *i.e.*, it focuses on sample paths instead of focusing on residual services at each server.

We again assume that we are given a tandem network of n servers with a set of m flows crossing it, and a distinguished flow of interest, flow 1. We assume that every arrival curve $F_i^{(0)}$ is $\langle \varepsilon_t^{(i)}, \alpha_i \rangle$ constrained where α_i is a burst-rate curve and the error probability function $\varepsilon_t^{(i)}(\sigma_i)$ is convex in σ_i. For the servers, we assume that they provide a $\langle \eta_t^{(j)}, \beta_j \rangle$ VCN service where β_j is rate-latency and $\eta_t^{(j)}(\theta_j)$ is convex in θ_j. In our numerical examples, we will use the exponentially bounded burstiness and latency (tEBB and tEBL) functions of (3) and (4), which are clearly convex in σ and θ.

Let $D(\sigma_1, \ldots, \sigma_m, \theta_1, \ldots, \theta_n)$ denote the worst-case delay of the deterministic tandem network with burst parameters $\sigma_1, \ldots, \sigma_m$ and delay parameters $\theta_1, \ldots, \theta_n$: The arrival of flow i is deterministically $(\alpha_i + \sigma_i)$-constrained and server j provides a deterministic VCN service that is $(\beta_j - \theta_j)_+$-constrained. Theorem 1 shows that D is an affine polynomial in the σ_i and θ_j, and Algorithm 1 allows to compute their coefficients.

The convexity assumption, together with the affinity of the system's delay in the burst and delay parameters, will allow us to use convex programming to find, for a given delay, the optimal assignment of the parameters that (a) allows the system to experience the given delay and (b) minimizes the probability bound given by the sum of the flows' and servers' error probability functions.

Theorem 5. *Let $\mathcal{S}$ be a tandem network with n servers and m flows. Assume that the arrival of flow i is stochastically constrained by a burst-rate arrival curve with error probability function $\varepsilon_t^{(i)}$ and that server j*

is stochastically constrained by a rate-latency VCN service curve with error probability function $\eta_t^{(j)}$. *Let* $\sigma_1, \ldots, \sigma_m, \theta_1, \ldots, \theta_n \geq 0$ *and* $t \geq 0$. *Denote by* $d = D(\sigma_1, \ldots, \sigma_m, \theta_1, \ldots, \theta_n)$ *the delay of the corresponding deterministic tandem network with burst parameters* σ_i *and latency parameters* θ_j.

Then the probability that flow 1 *experiences a delay of more than* d *up to time* t *is upper-bounded by:*

$$\mathbf{P}\big(\exists s \leq t\colon F_1^{(0)}(s-d) - F_1^{(n)}(s) > 0\big) \leq \sum_{i=1}^{m} \varepsilon_t^{(i)}(\sigma_i) + \sum_{j=1}^{n} \eta_t^{(j)}(\theta_j).$$

Proof. Because the function $D(\sigma_1, \ldots, \sigma_m, \theta_1, \ldots, \theta_n)$ is nondecreasing in all arguments, the event $\exists s \leq t\colon F_1^{(0)}(s-d) - F_1^{(n)}(s) > 0$ of violating the delay bound of d before time t implies that one of deterministic arrival or service assumptions was violated before time t. That is, one of the events $\exists v \leq u \leq t\colon F_i^{(0)}(u) - F_i^{(0)}(v) > \alpha_i(u-v) + \sigma_i$ or $\exists v \leq u \leq t\colon C_j(u) - C_j(v) < (\beta_j(u-v) - \theta_j)_+$ where C_j denotes the capacity of server j. Taking the union bound now proves the claimed inequality. □

As the sum of convex functions is convex and the constraint of the deterministic delay being equal to d is linear, hence convex, in the parameters σ_i and θ_j, the optimization problem in Figure 4 is a convex program. By Theorem 5, its optimal value is indeed an upper bound on the probability that flow 1 experiences a delay of more than d up to time t. In fact, it is the best bound that we can achieve with Theorem 5.

We compare our approach with the SFA technique, which relies on the modularity of Network calculus. For each server j it computes a *residual service curve* for the flow of interest (flow 1) as $\tilde{\beta}_j = (\beta_j - \tilde{\alpha}_j)_+$, where $\tilde{\alpha}_j$ is a constraint on the cross traffic at server j, and then computes the global service for flow 1 ,$\tilde{\beta}$, by computing the convolution of all $\tilde{\beta}_j$. Then the end-to-end delay is bounded by the horizontal distance between the arrival curve of flow 1 and $\tilde{\beta}$.

For example, on the network of Figure 1, $\tilde{\beta}_1 = (\beta_1 - \alpha_2)_+$, $\tilde{\beta}_2 = \big(\beta_2 - (\alpha_3 + \alpha_2 \oslash \beta_1)\big)_+$ (we make an improvement compared to the classical SFA as from [25, Theorem 8], the worst-case delay is obtained for the SDF service policy) and $\tilde{\beta}_3 = \big(\beta_3 - (\alpha_3 \oslash [\beta_2 - (\alpha_2 \oslash \beta_1)]_+)\big)_+$, hence SFA obtains a delay bound of

$$h\Big(\alpha_1\,,\, (\beta_1 - \alpha_2)_+ * \big(\beta_2 - (\alpha_3 + \alpha_2 \oslash \beta_1)\big)_+ * \big(\beta_3 - (\alpha_3 \oslash [\beta_2 - (\alpha_2 \oslash \beta_1)]_+)\big)_+\Big).$$

With rate-latency VCN and burst-rate arrival curves, the delay bound obtained is also affine in the bursts and latencies. The optimization problem of Figure 4 can then be applied to obtain an error probability bound, which we call the sSFA bound. We denote sExact the solution of the convex program in Figure 4 that uses Algorithm 1 for the computation of the exact worst-case delay. The delays obtained with sExact will always

$$
\begin{array}{lll}
\text{minimize} & \displaystyle\sum_{i=1}^{m} \varepsilon_t^{(i)}(\sigma_i) + \sum_{j=1}^{n} \eta_t^{(j)}(\theta_j) & \\
\text{subject to} & D(\sigma_1, \ldots, \sigma_m, \theta_1, \ldots, \theta_n) = d & \\
& \sigma_i \geq 0, & i = 1 \ldots, m \\
& \theta_j \geq 0, & j = 1 \ldots, n
\end{array}
$$

Figure 4: Convex program to bound the probability of a delay $> d$ up to time t.

be better than those obtained with sSFA, simply because both use a deterministic bound and then the union bound as in the proof of Theorem 5. This technique benefits from having a tighter deterministic bound. Hence, sExact gives a tighter stochastic bound than sSFA since it starts with an optimal deterministic bound.

4.4. Numerical Examples and Comparison

In this section, we give numerical examples and comparisons for the tandem network in Figure 1. We used a MATLAB implementation of our algorithm and the CVX optimization package to carry out the computations.

We assume identical Bernoulli arrival processes that have a probability $p = 0.15$ to generate a bit every millisecond, *i.e.*, they have an expected rate of $r = 150\,\text{b/s}$. An application of [27, Proposition 3] with $p = 0.15$ and $\varepsilon = 0.01$ shows the t.a.c. guarantee that $\mathbf{P}\big(A(t) - A(s) \geq 0.16(t-s) + b\big) \leq e^{-\alpha b}$ where α is maximal such that $pe^{\alpha} + 1 - p \leq e^{(p+\varepsilon)\alpha}$ for all s and t measured in ms. The value of α is approximately $\alpha \approx 0.151$, for which the inequality still holds. Applying [15, Theorem 3.13(ii)] with $\theta = 0.01$ gives the v.b.c. guarantee of $\mathbf{P}\big(\exists s \leq t\colon A(t) - A(s) > 0.17(t-s) + b\big) \leq e^{-0.151 \cdot b} + \theta^{-1} \int_b^{\infty} e^{-0.151 \cdot y}\, dy = (1 + 100/0.151) \cdot e^{-0.151 \cdot b} \leq 659 \cdot e^{-0.151 \cdot b}$ for all t. Finally, using [15, Theorem 3.24(ii)] with $\theta = 0.01$, *i.e.*, by relaxing the rate by another 10 b/s, we get the m.b.c. bound of $\mathbf{P}\big(\exists u \leq s \leq t\colon A(s) - A(u) > 0.18(s-u) + b\big) \leq \theta^{-1} \int_{b-\theta t}^{b} 659 \cdot e^{-0.151 \cdot y}\, dy \leq 100 \cdot \int_{b-0.01 \cdot t}^{\infty} 659 \cdot e^{-0.151 \cdot y}\, dy = (65\,900/0.151) \cdot e^{0.00151 \cdot t - 0.151 \cdot b} \leq 436\,424 \cdot e^{0.00151 \cdot t - 0.151 \cdot b}$ for all t. This is an explicit tEBB error probability bound for the Bernoulli arrivals. We assume constant-rate servers with a rate of $R = 1\,180\,\text{b/s}$ and additional independent high-priority cross-traffic at every server identical to the arrivals, which, by the discussion at the end of Section 4.1.2, gives a tEBL error probability bound for the service to flows 1, 2, and 3 at each server of $\mathbf{P}\big(\exists u \leq s \leq t\colon C(s) - C(u) < (s - u - T)_+\big) \leq 436\,424 \cdot e^{0.00151 \cdot t - 0.151 \cdot T}$ where u, s, t, and T are measured in ms. This makes server 2 have a utilization rate of $\rho = 4 \cdot 150/1\,180 \approx 50.85\%$.

Figure 5 shows the bounds of the sExact and sSFA algorithms in the three-server system for the probability that the delay up to time $t = 10\,\text{s}$ is more than d for values of d between 0 s and 2 s. Both bounds are close to each other and larger than 10^{12} for $d = 0.02\,\text{s}$. After that, however, their ratio gets larger as d increases.

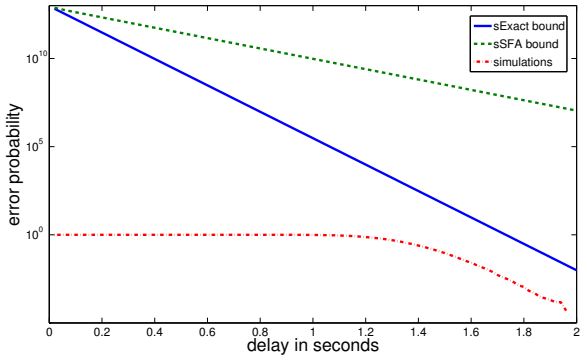

Figure 5: Stochastic delay bounds and simulations for three-server system.

(Note that the figure uses a logarithmic scale and thus their distance does not.) We observe that the bound of our sExact algorithm becomes meaningful, *i.e.*, becomes less than 1, at $d \approx 1.74\,\mathrm{s}$, whereas the bound of the sSFA algorithm becomes meaningful only at $d \approx 4.4\,\mathrm{s}$. Figure 5 also contains simulation data for the network with Bernoulli arrivals. It uses $N = 20\,000$ simulation runs and depicts the relative amount of runs in which the maximum delay up to $t = 10\,\mathrm{s}$ is greater than d. This relative amount is strictly between 0 and 1 only for values of d between $0.78\,\mathrm{s}$ and $2\,\mathrm{s}$. In this interval, the sSFA bound is still larger than 1. While there is still a considerable gap between the simulated maximum delays and the sExact bound, the plots suggest a similar slope of the simulated data and the sExact bound in logarithmic scale. This observation spawns further questions and directions for future research.

5. Conclusion

This paper presented a new and fast algorithm for computing the exact worst-case delay of tandem networks with arbitrary multiplexing when the arrivals are constrained by burst-rate arrival curves and the servers provide a rate-latency service curve. As a byproduct, we proved that the worst-case delay is, in fact, an affine function of the burstiness and latency parameters of the arrivals and servers. Our algorithm is the first that computes the linear coefficients and runs in time $O(n^2)$.

Linearity in the burstiness and latency parameters allowed the extension of our bounds to the case where arrivals and server capacities are stochastic processes. For the case that the error probability functions are convex, *e.g.*, the exponentially bounded burstiness (tEBB) and exponentially bounded latency (tEBL) models, we gave a polynomial-time algorithm based for finding stochastic delay bounds. We showed that our algorithm gives bounds at least as good as the state-of-the-art SFA-type method and provided a quantitative numerical comparison.

Future work will focus on finding the worst-case delay for specific multiplexing policies and more general topologies. As some of these problems are NP-hard even in feed-forward topologies, only approximations can be efficiently computed, but the quantification of such an approximation is an important issue. We would also like to further explore the gap between v.b.c.-type and m.b.c.-type stochastic models with respect to delay bounds. In particular, the correlation between the arrival processes and the events of breaking a constraint should be worth investigating.

Acknowledgments

The authors would like to thank Florin Ciucu for many comments on earlier versions of this paper.

References

[1] A. Parekh, R. Gallager, A generalized processor sharing approach to flow control in integrated services networks: the single-node case, IEEE/ACM Transactions on Networking 1 (1993) 344–357.

[2] A. Parekh, R. Gallager, A generalized processor sharing approach to flow control in integrated services networks: the multiple-node case, IEEE/ACM Transactions on Networking 2 (1994) 137–150.

[3] C.-S. Chang, D.-S. Lee, Y.-S. Jou, Load balanced Birkhoff-von Neumann switches, in: 2nd IEEE Workshop on High Performance Switching and Routing, HPSR, IEEE Press, New York, 2001.

[4] C.-S. Chang, D.-S. Lee, C.-M. Lien, Load balanced Birkhoff–von Neumann switches, part II: multi-stage buffering, Computer Communications 25 (2002) 623–634.

[5] R. Cruz, Quality of service guarantees in virtual circuit switched networks, IEEE Journal on Selected Areas in Communication 13 (1995) 1048–1056.

[6] J. McManus, K. Ross, Video-on-demand over ATM: constant-rate transmission and transport, IEEE Journal on Selected Areas in Communication 14 (2006) 1087–1098.

[7] M. Boyer, J. Migge, M. Fumey, PEGASE – a robust and efficient tool for worst-case network traversal time evaluation on AFDX, in: SAE AeroTech, 2011.

[8] S. Chakraborty, L. Phan, P. Thiagarajan, Event count automata: a state-based model for stream processing systems, in: 26th IEEE Real-Time Systems Symposium, RTSS, IEEE Press, New York, 2005.

[9] S. Martin, P. Minet, Worst case end-to-end response times of flows scheduled with FP/FIFO, in: 5th International Conference on Networking, ICN, IEEE Press, New York, 2006.

[10] J. Schmitt, F. Zdarsky, M. Fidler, Delay bounds under arbitrary multiplexing: When network calculus leaves you in the lurch. . . , in: 27th Conference on Computer Communications, INFOCOM, IEEE Press, New York, 2008.

[11] L. Lenzini, L. Martorini, E. Mingozzi, G. Stea, Tight end-to-end per-flow delay bounds in FIFO multiplexing sink-tree networks, Performance Evaluation 63 (2006) 956–987.

[12] A. Bouillard, L. Jouhet, E. Thierry, Tight performance bounds in the worst-case analysis of feed-forward networks, in: 29th Conference on Computer Communications, INFOCOM, IEEE Press, New York, 2010.

[13] A. Bouillard, G. Stea, Exact worst-case delay for FIFO-multiplexing tandems, in: 6th International ICST Conference on Performance Evaluation Methodologies and Tools, ValueTools, ICST, Gent, 2012.

[14] C.-S. Chang, Stability, queue length and delay of deterministic and stochastic queueing networks, IEEE Transactions on Automatic Control 39 (1994) 913–931.

[15] Y. Jiang, Y. Liu, Stochastic Network Calculus, Springer, Heidelberg, 2008.

[16] M. Fidler, Survey of deterministic and stochastic service curve models in the network calculus, IEEE Communications Surveys & Tutorials 12 (2010) 59–86.

[17] C. Chang, Performance Guarantees in Communication Networks, Springer, Heidelberg, 2000.

[18] J. Liebeherr, Y. Ghiassi-Farrokhfal, A. Burchard, On the impact of link scheduling on end-to-end delays in large networks, IEEE Journal on Selected Areas in Communications 29 (2011) 1009–1020.

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

[20] A. Burchard, J. Liebeherr, S. Patek, A min-plus calculus for end-to-end statistical service guarantees, IEEE Transactions on Information Theory 52 (2006) 4105–4114.

[21] K. Wang, F. Ciucu, C. Lin, S. Low, A stochastic power network calculus for integrating renewable energy sources into the power grid, IEEE Journal on Selected Areas in Communications 30 (2012) 1037–1048.

[22] L. Santinelli, P. Yomsi, D. Maxim, L. Cucu-Grosjean, A component-based framework for modeling and analyzing probabilistic real-time systems, in: 16th IEEE Conference on Emerging Technologies & Factory Automation, ETFA, IEEE Press, New York, 2011.

[23] J.-Y. Le Boudec, P. Thiran, Network Calculus: A Theory of Deterministic Queuing Systems for the Internet, Springer, Heidelberg, 2001.

[24] A. Bouillard, L. Jouhet, E. Thierry, Comparison of different classes of service curves in network calculus, in: 10th International Workshop on Discrete Event Systems, WODES, IFAC, New York, 2010.

[25] A. Bouillard, A. Junier, Worst-case delay bounds with fixed priorities using network calculus, in: 5th International ICST Conference on Performance Evaluation Methodologies and Tools, ValueTools, ICST, Gent, 2011.

[26] A. Bouillard, L. Jouhet, E. Thierry, Tight performance bounds in the worst-case analysis of feed-forward networks, Tech. Rep. RR-7012, INRIA, Le Chesnay (2009).

[27] O. Yaron, M. Sidi, Performance and stability of communication networks via robust exponential bounds, IEEE/ACM Transactions on Networking 1 (1993) 372–385.

[28] R. Cruz, Quality of service management in integrated services networks, in: 1st Semi-Annual Research Review, Center for Wireless Communications, University of California, San Diego, 1996.

[29] F. Ciucu, J. Schmitt, Perspectives on network calculus: No free lunch, but still good value, in: SIGCOMM 2012, ACM, New York, 2012.

Appendix A. Proof of Theorem 1

The worst-case delay is $\Delta_1(0,\dots,0)$ and we need to to compute Δ_j.

Define

$$Q_j^k = b_j^k + x_j^k + r_j^k T_j.$$

As Q_j^ℓ is an affine function in T_j, b_j^ℓ and x_j^ℓ, we only need to show that every Δ_j is affine in the Q_j^ℓ.

Initialization: Computation of Δ_n. The only burst that can be transmitted by server $n-1$ is for destination n. Suppose that a burst of size x_n is transmitted at time t_{n-1}. It is obvious that the worst-case scenario is that the server is exact until the bit on interest is served.

The delay of the bit of interest is then the solution $\delta = \Delta_n(x_n^n)$ of the equation $x_n^n + b_n^n + r_n^n \delta = R_n(\delta - T_n)_+$, see Figure A.6. It is equal to

$$\Delta_n(x_n^n) = \frac{R_n T_n + x_n^n + b_n^n}{R_n - r_n^n} = T_n + \frac{Q_n^n}{\mathcal{R}_n^n},$$

where $\mathcal{R}_n^n = R_n - r_n^n$.

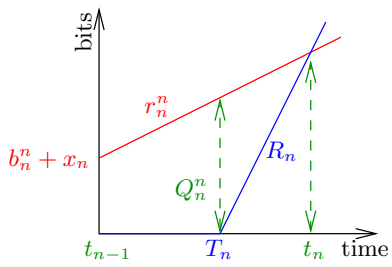

Figure A.6: Delay incurred at server n.

Inductive step: Computation of Δ_j *from* Δ_{j+1}. Suppose that Δ_{j+1} is known and satisfies the properties above. We first compute the delay induced from server j assuming that the service is exact for flows ending at server k (of before) and infinite for flows ending at server $k+1$ (or after). Call Δ_j^k this delay. This is a function of x_j^k the bursts transmitted by server $j-1$.

$\Delta_j^k(x_j^j, \ldots, x_j^n)$ is computed the following way: It takes time δ to serve flows ending at $j, \ldots, k$, and data from any other flow is transmitted to server $j+1$. This quantity of data is $x_{j+1}^\ell = b_j^\ell + x_j^\ell + r_j^\ell \delta$ for $\ell > k$ where δ satisfies

$$\sum_{\ell=j}^{k} x_j^\ell + b_j^\ell + r_j^\ell \delta = R_j(\delta - T_j)_+ \ ,$$

i.e., $\delta = T_j + \sum_{\ell=j}^k Q_j^\ell / \big(R_j - \sum_{\ell=j}^k r_j^\ell\big)$. For $\ell > k$, we then have the equality

$$x_{j+1}^\ell = Q_j^\ell + r_j^\ell \frac{\sum_{\ell=j}^k Q_j^\ell}{R_j - \sum_{\ell=j}^k r_j^\ell} \ .$$

Then

$$\begin{aligned}
\Delta_j^k(x_j^j, \ldots, x_j^n) &= \delta + \Delta_{j+1}(0, \ldots, 0, x_{j+1}^{k+1}, \ldots, x_{j+1}^n) \\
&= T_j + \frac{\sum_{\ell=j}^k Q_j^\ell}{R_j - \sum_{\ell=j}^k r_j^\ell} + D_{j+1} + \sum_{\ell'=k+1}^{n} \frac{1}{\mathcal{R}_{j+1}^{\ell'}} \left(Q_j^{\ell'} + r_j^{\ell'} \frac{\sum_{\ell=j}^k Q_j^\ell}{R_j - \sum_{\ell=j}^k r_j^\ell} \right) \\
&= T_j + D_{j+1} + \sum_{\ell=k+1}^{n} \frac{Q_j^\ell}{\mathcal{R}_{j+1}^\ell} + \sum_{\ell=j}^{k} \frac{Q_j^\ell}{R_j - \sum_{\ell'=j}^k r_j^{\ell'}} \left(1 + \sum_{\ell'=k+1}^{n} \frac{r_j^{\ell'}}{\mathcal{R}_{j+1}^{\ell'}} \right)
\end{aligned}$$

The other scenarios that should be taken into account is when part of the flows ending at k are served, and the other part is transmitted instantaneously to server $j+1$ (due to the SDF service policy, only one

flow can be partially served). When y_k is this latter quantity, the worst-case delay from time t_{j-1} is

$$\begin{aligned}\Delta_j^{k,y_k}(x_j,\dots,x_n) &= T_j + X + \Delta_{j+1}\big(0,\dots,0,y_k,Q_j^{k+1}+r_j^{k+1}X,\dots,Q_j^n+r_j^nX\big)\\ &= T_j + D_{j+1} + X\left(1+\sum_{\ell'=k+1}^{n}\frac{r_j^{\ell'}}{\mathcal{R}_{j+1}^{\ell'}}\right)+\sum_{\ell'=k+1}^{n}\frac{Q_j^{\ell'}}{\mathcal{R}_{j+1}^{\ell'}}+\frac{y_k}{\mathcal{R}_{j+1}^{k}}\end{aligned}$$

where $X = \big((\sum_{\ell=j}^{k} Q_j^\ell) - y_k\big)/\big(R_j - \sum_{\ell=j}^{k} r_j^\ell\big)$. This is a linear function in y_k.

The next lemma shows that we can drop these "mixed" scenarios as they lead to shorter delays.

Lemma 4. $\forall y_k$, $\Delta_j^{k,y_k} \le \max(\Delta_j^k, \Delta_j^{k-1})$.

Proof. First notice that there exists α such that $\Delta_j^{k,y_k} = \Delta_j^k + \alpha y_k$ with

$$\alpha = \frac{1}{\mathcal{R}_{j+1}^k} - \frac{1+\sum_{\ell'=k+1}^{n} r_j^{\ell'}/\mathcal{R}_{j+1}^{\ell'}}{R_j - \sum_{\ell=k}^{k} r_j^\ell}.$$

Depending on whether α is positive or negative (the null case is trivial), Δ_j^{k,y_k} is maximized when y_k is maximized or for $y_k = 0$.

If α is positive, then the delay is maximized with the maximal value for y_k, which corresponds to not serving the flow ending at k at all and to Δ_j^{k-1}.

If it is negative, then the delay is maximized by $y_k = 0$, which corresponds to reducing the flow's backlog to 0 and to Δ_j^k. □

We now prove that there Δ_j is in fact on of the Δ_j^k. In other words, Δ_j is affine.

Lemma 5. *There exists k such that $\Delta_j = \Delta_j^k$ (that is, for all $x_j,\dots,x_n$, we have $\Delta_j(x_j,\dots,x_n) = \Delta_j^k(x_j,\dots,x_n)$) and for all k, $\mathcal{R}_j^k \ge \mathcal{R}_j^{k+1}$.*

Proof. The proof is also by induction. We prove that $\forall j$, we have the equivalence

$$\begin{aligned}\Delta_j = \Delta_j^k \Leftrightarrow \forall k' > k\ \frac{1}{\mathcal{R}_{j+1}^{k'}} &> \frac{1}{R_j - \sum_{\ell=j}^{k} r_j^\ell}\left(1+\sum_{\ell'=k'+1}^{n}\frac{r_j^{\ell'}}{\mathcal{R}_{j+1}^{\ell'}}\right)\\ \text{and } \frac{1}{\mathcal{R}_{j+1}^{k}} &\le \frac{1}{R_j - \sum_{\ell=j}^{k} r_j^\ell}\left(1+\sum_{\ell'=k+1}^{n}\frac{r_j^{\ell'}}{\mathcal{R}_{j+1}^{\ell'}}\right).\end{aligned}$$

Assertion **(B)** is proved at the same time.

The equivalence above also state that if for all $k' > k$ $\Delta_j \neq \Delta_j^{k'}$, then $\Delta_j^n \leq \Delta_j^{n-1} \leq \cdots \leq \Delta_j^k$. Let us assume this.

The coefficient of Q_j^ℓ in $\Delta_j^{k'}$ for all $k' \leq k$ and $\ell > k$ is $\frac{1}{\mathcal{R}_{j+1}^\ell}$ and that of Q_j^k is

- either $\frac{1}{R_j - \sum_{\ell=j}^k r_j^\ell} \left(1 + \sum_{\ell'=k+1}^n \frac{r_j^{\ell'}}{\mathcal{R}_{j+1}^{\ell'}}\right)$ (for Δ_j^k)
- or $\frac{1}{\mathcal{R}_{j+1}^k}$ (for Δ_j^ℓ, $\ell < k$).

Suppose that

$$\frac{1}{R_j - \sum_{\ell=j}^k r_j^\ell} \left(1 + \sum_{\ell'=k+1}^n \frac{r_j^{\ell'}}{\mathcal{R}_{j+1}^{\ell'}}\right) \geq \frac{1}{\mathcal{R}_{j+1}^k}. \tag{A.1}$$

For $\ell \leq k' < k$, the coefficient of Q_j^ℓ in $\Delta_j^{k'}$ is

$$\begin{aligned}
&\frac{1}{R_j - \sum_{\ell'=j}^{k'} r_j^{\ell'}} \left(1 + \sum_{\ell'=k'+1}^n \frac{r_j^{\ell'}}{\mathcal{R}_{j+1}^{\ell'}}\right) \leq \\
&\qquad \leq \frac{1}{R_j - \sum_{\ell'=j}^{k'} r_j^{\ell'}} \left(1 + \sum_{\ell'=k+1}^n \frac{r_j^{\ell'}}{\mathcal{R}_{j+1}^{\ell'}} + \sum_{\ell'=k'+1}^k \frac{r_j^{\ell'}}{\mathcal{R}_{j+1}^k}\right) \\
&\qquad \leq \frac{1}{R_j - \sum_{\ell=j}^{k'} r_j^\ell} \left(1 + \sum_{\ell'=k+1}^n \frac{r_j^{\ell'}}{\mathcal{R}_{j+1}^{\ell'}} + \sum_{\ell'=k'+1}^k \frac{r_j^{\ell'}}{R_j - \sum_{\ell''=j}^k r_j^{\ell''}} \left[1 + \sum_{\ell''=k+1}^n \frac{r_j^{\ell''}}{\mathcal{R}_{j+1}^{\ell''}}\right]\right) \\
&\qquad = \left[1 + \sum_{\ell'=k+1}^n \frac{r_j^{\ell'}}{\mathcal{R}_{j+1}^{\ell'}}\right] \frac{1}{R_j - \sum_{\ell'=j}^{k'} r_j^{\ell'}} \left(1 + \frac{\sum_{\ell'=k'+1}^k r_j^{\ell'}}{R_j - \sum_{\ell'=j}^k r_j^{\ell'}}\right) \\
&\qquad = \left[1 + \sum_{\ell'=k+1}^n \frac{r_j^{\ell'}}{\mathcal{R}_{j+1}^{\ell'}}\right] \frac{1}{R_j - \sum_{\ell'=j}^k r_j^{\ell'}},
\end{aligned}$$

which is the coefficient of Q_j^ℓ in Δ_j^k. The first inequality uses (**B**) for server $j+1$ and the second uses Ineq. (A.1).

Moreover, for $k' < \ell < k$, the coefficient of Q_j^ℓ in $\Delta_j^{k'}$ is

$$\frac{1}{\mathcal{R}_{j+1}^{k'}} \leq \frac{1}{\mathcal{R}_{j+1}^k} \leq \frac{1}{R_j - \sum_{\ell=j}^k r_j^\ell} \left(1 + \sum_{\ell'=k+1}^n \frac{r_j^{\ell'}}{\mathcal{R}_{j+1}^{\ell'}}\right),$$

so $\Delta_j^k \geq \Delta_j^{k'}$ and $\Delta_j = \Delta_j^k$.

The same kinds of computations lead to $\Delta_j^{k-1} > \Delta_j^k$ if $\frac{1}{\mathcal{R}_{j+1}^k} > \frac{1}{R_j - \sum_{\ell=j}^k r_j^\ell} \left(1 + \sum_{\ell'=k+1}^n \frac{r_j^{\ell'}}{\mathcal{R}_{j+1}^{\ell'}}\right)$

Set $\mathcal{R}_j^k$ according to the Δ_j that is computed (that is Δ_j^k for some k). Then we still have $\mathcal{R}_j^\ell \leq \mathcal{R}_j^{\ell+1}$ and (**B**) is true for server j. □

To conclude, just remark that the delay for flow 1 is $\Delta_1(0,\ldots,0)$, which is a polynomial of degree 1 of the variables b_i and T_j. Lemma 5 also gives a polynomial greedy algorithm to compute the coefficients for every b_i and T_j in polynomial time, and an explicit construction for a worst-case scenario: Proceed backward from server n to server 1, and for each server k, there are $n-k+1$ potential worst-case trajectories. Checking that one case is the actual worst-case trajectory can be done by checking one inequality. Those inequalities imply the computation of some constants (new flows arriving). As a flow enters the system at one server only, the total number of computations for constants is $O(m)$. So, finding the actual worst-case can be done in $O(n-k)$ operations (there is one inequality to check for each case) apart from the computation of the constants r_j^ℓ and b_j^ℓ, and the computation of the worst-case delay can be done in $O(n^2+m)$. Note that if every flow has a different path, then $m = O(n^2)$ and computing the worst-case delay can be done in $O(n^2)$.

Appendix B. About the Linear Programming Approach

Since other approaches to worst-case delay bounds already exist in the literature, we now want to explain the differences of our new formulation to existing ones.

In [12], the linear program presented to compute the worst-case delay is of the form

$$\text{maximize} \quad AX \quad \text{subject to} \quad BX \leq C \text{ and } X \geq 0.$$

Vector A and matrix B do not depend on the parameters $\text{Params} = (\sigma_1,\ldots,\sigma_m,\theta_1\ldots,\theta_n)$. Translating our approach naïvely into this formalism would lead to compute the minimum of a function with the constraint

$$\min\big\{f(\sigma_1,\ldots,\sigma_m,\theta_1\ldots,\theta_n) \mid \max\{AX \mid BX \leq C\} = d \text{ and } \sigma_i \geq 0,\ \theta_j \geq 0\big\}.$$

The maximum cannot be dropped since we want to ensure a worst-case trajectory.

Another solution, following the lines of [26], would be to consider the dual problem: as A and B do not depend on $Param$, the dual problem is (expressing the dependencies in the parameters)

$$\text{minimize} \quad C^t(\text{Param})Y \quad \text{subject to} \quad B^TY \leq A^T,\ Y \geq 0.$$

Now, only the objective function depends on the parameters, and the optimal value is then obtained at a vertex of the convex set of the constraints. The delay can then be expressed as a minimum of a finite number of linear functions, but this number can be exponential in the number of constraints, which makes

this solution intractable.

Appendix C. On the Choice of the Stochastic Model

As mentioned in Section 4.1, there are three principal types of stochastic model assumptions for arrival curves: The traffic amount centric (t.a.c.) model, the virtual backlog centric (v.b.c.) model, and the maximum virtual backlog centric (m.b.c.) model. The t.a.c. and v.b.c. models are ordered in increasing strength in the sense that an v.b.c. arrival process is also an t.a.c. arrival process if the arrival constraint curve α and the error probability function ε are equal. Such an ordering does not exist between the v.b.c and m.b.c. models, if only for the fact that the error probability functions ε_t in the m.b.c. model also depend on the time parameter t. Moreover, the m.b.c. model is not a meaningful stochastic model if one chooses the error probability function independent of t: If $\varepsilon_t = \varepsilon$ for all t, then Ciucu and Schmitt [29, Section 4.4] have shown that every stationary ergodic m.b.c. arrival process is quasi-deterministic. Thus, typically, $\varepsilon_t(\sigma) \to 1$ as $t \to \infty$ in the m.b.c. model.

In this paper, we chose the m.b.c. model for arrivals (and an analogous one for server capacities). This poses the question why we made this choice, and did not, say, choose the v.b.c. model. Even though we explained in Section 4.1 that even t.a.c. arrival curves can be translated into m.b.c. curves, this translation requires to augment the rate bound by a small bit and is thus not applicable to heavy load scenarios near the stability point. In fact, we chose the m.b.c. model not to simplify the exposition, but because we see a fundamental hurdle to prove analogous results in the v.b.c. model. Burchard *et al.* [20, Section III-C] already discussed some of the difficulties that arise in a v.b.c.-type model. They explained that the difficulty of composing two stochastic service curves lies in the fact that v.b.c.-type conditions would need to be evaluated at a time t that is a random variable and not a constant value. In the rest of this section, we want to extend on this point by giving an in-depth explanation of the problem. By that, we give an argument for the usage of the m.b.c. model by showing that we would otherwise need to make very strong independence assumptions.

Let us focus on the simple example of the composition of two VCN servers and two flows crossing both servers with two stochastic service constraints of v.b.c.-type. That is,

$$\forall t\colon\ \mathbf{P}\big(\exists s \le t\colon C_1(t) - C_1(s) < \beta_1(t-s)\big) \le \eta_1$$
$$\forall t\colon\ \mathbf{P}\big(\exists s \le t\colon C_2(t) - C_2(s) < \beta_2(t-s)\big) \le \eta_2.$$

We want to derive a (simple) service curve for the composed server. It is a result of deterministic network

calculus (see, *e.g.*, [23]) that

$$\Big(\forall s \le t\colon C_2(t) - C_2(s) \ge \beta_2(t-s) \wedge \forall u \le s = \mathrm{start}(t)\colon C_1(s) - C_1(u) \ge \beta_1(s-u)\Big)$$
$$\Rightarrow B_2(t) \ge A * \beta_1 * \beta_2(t).$$

As a consequence, we can write:

$$\begin{aligned}\mathbf{P}\big(B_2(t) < A * \beta_1 * \beta_2(t)\big) \le{}& \mathbf{P}\big(\exists s \le t\colon C_2(t) - C_2(s) < \beta_2(t-s)\big) \\ &+ \mathbf{P}\big(\exists u \le \hat{s} = \mathrm{start}(t)\colon C_1(\hat{s}) - C_1(u) < \beta_1(\hat{s}-u)\big)\end{aligned} \tag{C.1}$$

The term $\mathbf{P}\big(\exists s \le t\colon C_2(t) - C_2(s) < \beta_2(t-s)\big)$ is at most η_2 by the service constraint for server 2.

Difficulties arise when wanting to bound the second probability. One cannot simply use the definition of the service constraint for server 1 because $\hat{s} = \mathrm{start}(t)$ is not a constant value. Buchard *et al.* [20] stated that the fact that $\hat{s}$ is not a constant value, but a random variable, is what causes problems. We want to make the somewhat more precise statement that the problem lies in the potential *dependency* of $\hat{s}$ and the violation of the service constraint. Denote by E_t the event $\exists s \le t\colon C_1(t) - C_1(s) < \beta_1(t-s)$. In fact, we will show that the second probability of the sum in (C.1) is bounded by η_1 if $\hat{s}$ is independent of $E_{\hat{s}}$ in the sense that the conditional probability $\mathbf{P}\big(E_{\hat{s}} \mid \hat{s} = s\big)$ is equal to (or upper-bounded by) $\mathbf{P}(E_s)$.

But first, to have the conditional probability and also the event $E_{\hat{s}}$ well-defined, we need to show that $\hat{s}$ is indeed a random variable, which is not trivial.

Lemma 6. *Let A be the arrival process and B the departure process of a server. Then* $\mathrm{start}(t)$ *is a random variable for every $t \ge 0$.*

Proof. We start by proving that every arrival and service process is separable. Indeed, we show the stronger claim that every (random) element of the function class $\mathcal{F}$ only depends on its values at the positive rationals $\mathbb{Q}_+$: So let $F \in \mathcal{F}$ and $t \ge 0$. We choose an increasing sequence of positive rationals $t_k \in \mathbb{Q}_+$ with limit $t_k \to t$. By left-continuity, $F(t) = \lim F(t_k)$, *i.e.*, the function value $F(t)$ only depends on the function values $F(t_k)$ at positive rationals.

Set $D(t) = A(t) - B(t)$. Because of left-continuity, for all $0 \le s < t$, we have the equivalence

$$\exists s < \tau \le t\colon\ D(\tau) = 0 \iff \exists s < \tau_1 < \tau_2 < \cdots < t\colon\ \forall k\colon\ \tau_k \in \mathbb{Q}_+ \wedge D(\tau_k) \le 2^{-k} \tag{C.2}$$

with probability 1. Setting V_K equal to

$$\bigcup_{\substack{s<\tau_1<t\\ \tau_1\in T}} \bigcup_{\substack{\tau_1<\tau_2<t\\ \tau_2\in T}} \cdots \bigcup_{\substack{\tau_{K-1}<\tau_K<t\\ \tau_K\in T}} \bigcap_{k=1}^{K} \{\omega\in\Omega \mid D_\omega(\tau_k)\le 2^{-k}\}$$

we see that every V_K is measurable and the sequence of the V_K is nonincreasing. Hence the monotone limit

$$\{\omega\in\Omega \mid \exists s<\tau\le t\colon\ D_\omega(\tau)=0\} = \lim_{K\to\infty} V_K$$

is measurable. Thus for every $t\in\mathbb{R}_+$, $\mathrm{start}(t)$ is a random variable as we can write

$$\{\omega\in\Omega \mid \mathrm{start}_\omega(t)\le s\} = \Omega\setminus\{\omega\in\Omega \mid \exists s<\tau\le t\colon\ D_\omega(\tau)=0\},$$

which is clearly a measurable set for every $0\le s\le t$. □

We can then use the law of total probability $\mathbf{P}(E_{\hat{s}}) = \mathbf{E}\big[\mathbf{P}(E_{\hat{s}} \mid \hat{s})\big]$ to show that the probability of interest $\mathbf{P}(E_{\hat{s}})$ is bounded by η_1 by monotony of the expected value because the conditional probability $\mathbf{P}(E_{\hat{s}} \mid \hat{s})$ is. This then shows that (C.1) is bounded by $\eta_1+\eta_2$.

On the other hand, if there is a tight correlation between the instant of violating the service constraint and $\hat{s}$, then this argument is not valid. Take, for instance, $\hat{s}$ as the random variable

$$\hat{s} = \inf\big\{s\le t \mid \exists u\le s\colon C_1(s)-C_1(u)<\beta_1(s-u)\big\}\cup\{0\}$$

of the earliest instance $\le t$ where the service constraint is violated, or 0 if it is not violated before time t. Then, in fact, $\mathbf{P}(E_{\hat{s}}) = \mathbf{P}\big(\exists u\le s\le t\colon C_1(s)-C_1(u)<\beta_1(s-u)\big)$, which suggests that an m.b.c.-type constraint is more natural to consider if correlation cannot be ruled out.

A method for getting delay bounds for a chain of server with disjoint cross-traffic was introduced by Burchard *et al.* [20]. The key ingredient to their proof is to relax the latency of almost each server by a parameter $a>0$. By that, they are able to circumvent the formally stronger assumptions of an m.b.c. model, which are know to lead to several zero-one-laws in certain cases. We believe that an adaption of their method can be combined with our approach.