Performance Evaluation 91 (2015) 270–285

Contents lists available at ScienceDirect

Performance Evaluation

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

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

ARTICLE INFO

Article history:
Available online 4 July 2015

Keywords:
Performance analysis
Worst-case delay
Network calculus
Stochastic network calculus

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.

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 [3–5], and Video-on-Demand [6]. More recently, deterministic bounds have become especially useful in the context of large embedded networks 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.

* Corresponding author.
E-mail addresses: anne.bouillard@ens.fr (A. Bouillard), thomas.nowak@ens.fr (T. Nowak).

http://dx.doi.org/10.1016/j.peva.2015.06.016
0166-5316/

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 [15–17]. 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 several supplementary proofs and discussions of our approach.

2. Framework and model

We denote by $\mathbb{N}$ the set of non-negative integers $\{0, 1, \dots\}$ and for all $n \in \mathbb{N}$, we set $[\![n]\!] = \{1, \dots, n\}$. 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 $\mathcal{F} = \{f : \mathbb{R}_+ \to \mathbb{R} | f$ 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 $s \leq t$: $A(t) - A(s) \leq \alpha(t - s)$.

A typical example of such arrival curve is 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 $\forall (A, B) \in \mathcal{S}$: $A \geq B \geq A * \beta$, 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 $\forall (A, B) \in \mathcal{S}$: $A \geq B$ and for all backlogged periods $(s, t]$: $B(t) - B(s) \geq \beta(t - s)$. The name strict service curves implies a difference to simple service curves. Works on the comparisons