Performance Evaluation ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at ScienceDirect

Performance Evaluation

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

Generalizing window flow control in bivariate network calculus to enable leftover service in the loop

Michael Beck *, Jens Schmitt

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

ARTICLE INFO

Article history:
Available online xxxx

Keywords:
Network calculus
Feedback
Window flow controlled systems
Performance of systems

ABSTRACT

Network calculus is a methodology to analyze queueing networks in a (probabilistic) worst-case setting. To that end, many results have been presented for different arrival and service processes and under different network topologies. Yet, in the stochastic setting of network calculus it has been found fundamentally hard to deal with feedback systems such as under window flow control constraints. First results on the bivariate feedback equation have been given by Chang et al. (2002). In this solution, the minimum in a sequence of self-convolutions must be determined. Chang et al. show that under certain conditions the – theoretically infinite – sequence of self-convolutions can be cut off after evaluating a finite number of them; the reason being that any further terms are greater than previous terms. As it turns out the required conditions do not allow for leftover service descriptions in the feedback loop. As leftover descriptions are essential in a network analysis, a more general result as in Chang et al. (2002) is needed.

The results of Chang et al. can be translated into the notation of σ-additive operators (see Chang (2000)). This paper's main result is a solution to the feedback equation for such operators. It generalizes present-day solutions and eventually allows to solve the feedback equation for leftover service descriptions inside the feedback loop.

1. Introduction

With the works of Cruz [1–3] the theory of network calculus emerged. As methodology it has grown to capture more complex queueing systems, including a wide variety of scheduling disciplines and topologies (see also [4]); furthermore, this theory has been extended into stochastic versions that effectively capture random effects like the statistical multiplexing gain [5] or arrivals and service as stochastic processes [6–8]. Many of the original, deterministic results of network calculus have been carried over to these extensions – known today as Stochastic Network Calculus (SNC) (see also [9]).

The results for window flow controlled systems (WFC systems), however, eluded a corresponding stochastic analysis ever since. This has been marked as an open and challenging problem for over a decade [4,10–12]. Only recently, progress towards a stochastic analysis of WFC systems has been made [12–15].

We briefly sketch the mechanics of the stochastic extension of network calculus: The event one is interested in (e.g., a delay bound) is first analyzed for sample paths satisfying certain conditions on the arrival and service processes. This first step is a logical implication, which takes place inside the space of possible sample paths. The probability of the event happening is then bounded by the probability of the conditions being fulfilled—the second step. That second step usually invokes bounds on the processes' tails or moment-generating functions. This two step procedure is characteristic for SNC [9].

* Corresponding author.
E-mail address: michaelalexbeck@gmail.com (M. Beck).

http://dx.doi.org/10.1016/j.peva.2017.04.008
0166-5316/

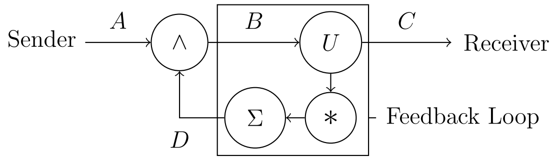

Fig. 1. A window flow controlled system. The element denoted by $*$ is a placeholder.

The main contribution of the recent results towards window flow controlled systems [14,15] is related to the second step, i.e., how to bound the set of favorable sample paths. Regarding the first step, results of (deterministic) bivariate network calculus [16] are re-used. This paper shows that the existing available results for the first step are not sufficient to analyze window flow controlled systems in the presence of crossflows, i.e., in the highly relevant case of multiple flows sharing the network. Indeed, the first step of SNC, using deterministic arguments, must be developed further to cover such scenarios. We address this by formulating a general result about WFC systems within the notation of σ-additive operators. With this theorem we achieve two goals: First, we have a single formulation for WFC systems and are able to recover previous results as special cases. Second, and more importantly, we cover a wide variety of systems, including those with crossflows. As such this paper provides a missing piece for the further research towards the stochastic analysis of window flow controlled systems.

The rest of the paper is structured as follows: We start with the needed notations in Section 2. In Section 3, we introduce window flow controlled systems and show that the current results are not sufficient to describe bivariate window flow controlled systems in the presence of crossflows. Afterwards, we introduce σ-additive operators in Section 4. In Section 5 our main result is presented and proven. Section 6 shows that our main result recovers previous results as special cases. Section 7 provides an illustration of the benefits of our result for a set of numerical experiments.

2. Notations

This paper uses the methodology of network calculus. Introductions to this topic are found in [4,16–18]. The following notations are used.

Definition 1. Call a real sequence $(a(t))_{t\in\mathbb{N}_0}$ of some quantity (e.g., bits, packets, data, etc.) *arrival increments* and its cumulatives $\sum_{s=0}^{t} a(s) =: A(t)$ a *flow* A. Denote an ordered pair $s \leq t$ of integers by (s,t) and the set of ordered pairs by $\Lambda(\mathbb{N}_0) = \Lambda$. A set indexed by Λ is called a *triangular array*. The set of bivariate functions is denoted by $\tilde{\mathcal{F}}$, i.e., $F \in \tilde{\mathcal{F}}$ if and only if

$$\begin{aligned} F : \Lambda(\mathbb{N}_0) &\to \mathbb{R} \cup \{-\infty\} \cup \{\infty\} \\ (s,t) &\mapsto A(s,t). \end{aligned}$$

Each flow A extends to a bivariate function by $A(s,t) := A(t) - A(s)$.

The relations $A < B$, $A \leq B$, and $A = B$ are to be understood pointwisely. Note that bivariate functions are, in general, not defining a flow. In contrast, to general bivariate functions, a flow and its bivariate extension take real values on the whole Λ. Furthermore a flow fulfills Chasles' relation: $A(s,r) + A(r,t) = A(s,t)$ for all $(s,r), (r,t) \in \Lambda$; or stated differently: A flow is always *additive*.

Definition 2. Consider a network element that has a flow A as input and the output B. Let $U \in \tilde{\mathcal{F}}$ and assume that the causality condition $A \geq B$ holds. The network element is a *dynamic U-server* if

$$B(t) \geq \min_{0\leq s\leq t}\{A(0,s) + U(s,t)\}$$

holds for all $t \in \mathbb{N}_0$. The expression

$$\min_{s\leq r\leq t}\{A(s,r) + U(r,t)\} =: A \otimes U(s,t)$$

is called *min-plus convolution* and it maps from $\tilde{\mathcal{F}} \times \tilde{\mathcal{F}}$ to $\tilde{\mathcal{F}}$.

The type of feedback systems we consider in this paper is depicted in Fig. 1. The elements involved are the following: The throttle $\wedge$ is a network element that has two inputs (A and D) and produces the minimum of these as output: $B(t) = A(t) \wedge D(t)$. The element denoted by U is a dynamic U-server. The element denoted by $*$ is a wildcard element that differs for specific examples. The element denoted by Σ is a window element.

Please cite this article in press as: M. Beck, J. Schmitt, Generalizing window flow control in bivariate network calculus to enable leftover service in the loop, Performance Evaluation (2017), http://dx.doi.org/10.1016/j.peva.2017.04.008.