Performance Evaluation 93 (2015) 27–46

Contents lists available at ScienceDirect

Performance Evaluation

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

Performance evaluation using periodic system-state measurements

W. Ellens[a],*, M. Mandjes[a], H. van den Berg[b], D. Worm[b], S. Błaszczuk[c]

[a] Korteweg–de Vries Institute for Mathematics, University of Amsterdam, P.O. Box 94248 1090 GE, Amsterdam, Netherlands

[b] TNO, P.O. Box 96800 2509 JE, The Hague, The Netherlands

[c] Faculty of Sciences, VU University Amsterdam, De Boelelaan 1081, 1081 HV, Amsterdam, Netherlands

ARTICLE INFO

Article history:
Received 15 September 2014
Received in revised form 21 April 2015
Accepted 14 July 2015
Available online 22 July 2015

Keywords:
Transient birth–death process
Measurements
System behavior inference
Exceedance probability
Reversible Markov process

ABSTRACT

This paper deals with the problem of inferring short time-scale fluctuations of a system's behavior from periodic state measurements. In particular, we devise a novel, efficient procedure to compute four interesting performance metrics for a transient birth–death process on an interval of fixed length with given begin and end states: the probability to exceed a predefined (critical) level m, and the expectation of the time, area, and number of arrivals above level m. Moreover, our procedure allows to compute the variances and cross-correlations of the latter three metrics. The asymptotic behavior of the metrics for small and large measurement intervals is also derived.

An extensive numerical study in the context of communication networks reveals the impact of important system parameters on the considered performance metrics, and shows that the three latter metrics are very highly correlated. We also illustrate through this numerical study how our analysis can be used in practical situations to support e.g., capacity management and SLA verification.

1. Introduction

There is still a growing interest in the use of traffic and performance measurements for capacity and service management in communication networks [1,2]. For example, these measurements can be used for network link dimensioning purposes or to verify whether the quality-of-service (QOS) objectives agreed upon in service level agreements (SLA s) between the provider and the users are met or violated. For technical reasons regarding the limited speed of the measurement equipment and in order to keep the amount of measurement data manageable, network operators often rely on periodic measurements (sampling) instead of continuously monitoring their network [1,3–5].

Periodic measurements obviously always yield just partial information: between the observations the underlying process may in principle have behaved extremely erratically. Hence, as one would like to have a sound impression of the state of the network, one should sample relatively frequently. On the other hand, frequent sampling may lead to a substantial overhead or slow down the network. In order to strike an appropriate balance, one would like to have an understanding of the information that the measurements provide about the behavior of the system under study *between* consecutive observations. This requires a methodology to infer these fluctuations from the observations, so as to be able, for example, to verify whether the SLA is satisfied with a given (high) probability.

[*] Corresponding author.
E-mail addresses: w.ellens@uva.nl (W. Ellens), m.r.h.mandjes@uva.nl (M. Mandjes), j.l.vandenberg@tno.nl (H. van den Berg), daniel.worm@tno.nl (D. Worm), s.blaszczuk@gmail.com (S. Błaszczuk).

http://dx.doi.org/10.1016/j.peva.2015.07.004
0166-5316/© 2015 Elsevier B.V. All rights reserved.

The goal of this paper is to devise fast, analytical techniques to analyze metrics related to random fluctuations between consecutive observations. Let t be the inter-observation time. We focus on the situation that the underlying stochastic process is of a *birth–death* nature; this class of processes is rather general and versatile, and covers several relevant queueing models as special cases. An important performance measure that we consider concerns the probability that within the interval of length t the number of clients exceeds a specific (high) level m, given that at the beginning (end, resp.) of the interval $X_0 = i$ ($X_t = j$, resp.) clients are present. In the SLA it is typically agreed that for a suitably chosen number m (corresponding e.g., to an acceptable throughput for active clients sharing a communication link) this exceedance probability is below a certain threshold (for instance 1%).

The metric described above provides us with important insight into the performance perceived by clients, but it may be desirable that the SLA contains more detailed performance indicators. When only considering the probability of exceeding level m, no information is provided about the 'severity' of such a congestion. For that reason, one could include a wider set of metrics in the SLA; think of performance measures such as the amount of time U_t that the number of customers is larger than m, the area A_t below the graph of the number of customers and above m, or the number N_t of customers that enters the system while the number of customers is larger than m.

In [6] we introduced a recursive procedure to compute the exceedance probability. The main contribution of the present paper is that we develop a fast procedure that is capable of evaluating the expected values of the more detailed performance metrics. This procedure relies on numerically solving a system of equations for the *joint* Laplace transform

$$\mathbb{E}\left(e^{-\alpha U_t-\beta A_t}z^{N_t} \mid X_0=i, X_t=j\right), \tag{1}$$

thus enabling the evaluation of the correlations between the individual metrics as well. Both procedures are presented in this paper and make clever use of the fact that a deterministic time can be approximated by a convolution of exponential times. In addition to their evaluation, we prove some interesting asymptotic results regarding the performance metrics, for the cases that the size of the sample period t becomes very small or very large. In particular, for $t \downarrow 0$ we show that the process moves from state i to j with overwhelming probability via a straight path, staying in each of the intermediate states for an equally short time, whereas for $t \to \infty$ we determine the trivariate central limit theorem for (U_t, A_t, N_t). Further, through numerous numerical examples, we investigate the impact of the system parameters on the various performance metrics and their mutual correlations, and illustrate practical application of our work by a simulation experiment.

From an application point-of-view, this paper can be seen as an approach to effectively using sample information for traffic management purposes. In this sense, it is part of a long tradition; think of the work on network tomography (see e.g. [7,8]), work on admission control based on probe information (see e.g. [9]), and studies on the inference of the input traffic process when observing the buffer process (see e.g. [5]). These measurement-based techniques are often used to perform traffic management in a distributed manner: without 'global knowledge' of the system at hand, the objective is to use 'local' measurement information for QOS control.

At the methodological level (in terms of performance evaluation techniques), there is a strong connection of the present work to the work of Preater [10] and Roijers et al. [11], who also analyze the joint distribution of various congestion-related metrics, but with the focus on the process' behavior during so called *congestion periods*, i.e., intervals during which the number of users is consistently above a certain threshold. Importantly, the focus in both [10,11] is on an important subclass of the birth–death process, viz. the $M|M|\infty$ queue, while our derivations hold for general birth–death processes.

The work presented in this paper was inspired by a case study for cable access networks; see [6] for preliminary results. In these cable access networks information regarding network performance is often obtained via throughput sampling ('speed tests', see e.g. [12]), i.e., measuring the throughputs of up- and downloads to/from locally installed servers, through regularly requests generated by PC s that act as fake users. In this setting the goal is to assess to what extent throughput sampling provides a representative view on the actual throughputs obtained by the users. The language and the input parameters used in the numerical examples of this paper originate from the context of cable access networks. Nevertheless, the results are also applicable to other contexts in which a process is observed regularly and can be modeled as a birth–death process.

The remainder of this paper is organized as follows. First, in Section 2, the probability of exceeding level m, conditional on the observations at the start and end of the interval is analyzed; we also provide explicit results for the important special cases $M|M|1$ and $M|M|\infty$. Extending this technique, Section 3 presents a procedure for fast and accurate evaluation of (1). In Section 4 we provide an extensive numerical study of the performance metrics using the analytical evaluation tools developed in the previous sections; we also give an example based on simulation in order to illustrate how operators can use our technique. The paper is concluded by a brief summary and some topics for further research in Section 5 and a set of Appendices analyzing the small and large time-scale behavior of the metrics.

2. Probability to stay below a certain level

This section presents a technique for determining the distribution of the maximum of a birth–death process (as formally defined below) over an interval with given initial and terminal conditions. The underlying ideas follow, by and large, concepts presented in our earlier paper [6]. The procedure follows the following three steps.

- We first (in Section 2.1) derive the distribution of the maximum over an *exponentially* distributed interval with a *given begin state*. We do so by setting up a recursion; in specific cases this recursion can be solved explicitly (viz. for $M|M|\infty$ and $M|M|1$ queues).