Fluid Analysis of Queueing in Random Environments

Giuliano Casale
Department of Computing
Imperial College London
SW7 2AZ, London, UK
g.casale@imperial.ac.uk

Mirco Tribastone
Institut für Informatik
Ludwig-Maximilians-Universität
Munich, Germany
tribastone@pst.ifi.lmu.de

***Abstract*—A large number of random environments leads to Markov processes where average-environment (AVG) and near-complete-decomposability (DEC) approximations suffer unacceptably large errors. This is problematic for queueing networks in particular, where state-space explosion hinders the application of numerical methods. In this paper we introduce *blending*, a novel fluid-based approximation for queueing models in random environments. Blending estimates the equilibrium of the model by iteratively evaluating transient-analysis subproblems for each state of the random environment. Each subproblem is solved by means of a very small system of ordinary differential equations, making the approach scalable and simple to implement. Random environments supported by blending are either state-independent, as for models with breakdown and repair, or state-dependent, such as for Markov-modulated queues where the service phase changes only while busy. Comparative results with AVG and DEC approximations prove that blending tackles the limitations of existing methods for evaluating queues in random environments.**

I. Introduction

Random environments are a useful abstraction to represent variability in the behavior of a system. For example, a network with server breakdowns may be seen as operating in an environment that causes failures according to a stochastic law [1]. Similarly, the recent diffusion in server farms of dynamic voltage-scaling technologies has spawned interest for random environments (policies) that change over time the frequency of a processor (service rate) and thus its power consumption [9]. Furthermore, non-exponential arrival processes described by Markov-modulated Poisson processes (MMPPs) or Markovian arrival processes (MAPs) may be seen as random environments, where the active state determines the current rate of arrival to the system [1]. Random environments that evolve depending on the system state allow to easily generalize the methodology, e.g., to queueing networks with non-exponential MMPP/MAP service times, state-dependent routing, variable threading levels, and blocking.

Markov random environments generate *events* according to Poisson processes, thus they are compositional with large Markov models used in performance evaluation and reliability analysis. Here, we focus on closed queueing network models, where the analysis of random environments have received limited attention due to the difficulty of dealing with the state-space explosion problem. Because of this issue, queueing networks in random environments are usually tackled either by average-environment (AVG) or near-complete-decomposability (DEC) approximations [1], [21]. In the AVG approximation, the system is studied with exponential state transitions having rates equal to their average value in the environment at equilibrium. This is equivalent to studying a single exponential network that enjoys a product-form solution. In the DEC approximation, a separate model is formulated for each environment state and the solutions weighted using the equilibrium distribution of the environment process. In general, AVG captures well environments where events happen *frequently* with respect to the representative time-scales of the system. Conversely, DEC describes accurately random environments that evolve *rarely* compared to the system dynamics. However, in intermediate situations both methods tend to be loose, especially when their estimates are significantly different. This makes a case for developing more robust approximation techniques for random environment analysis.

To cope with these difficulties, we introduce *blending*, a simple and scalable approximation for queueing models in random environments. Similarly to decomposition methods, blending estimates mean performance metrics by evaluating a submodel for each state of the random environment. However, rather than seeking equilibrium solutions for the submodels, blending studies their transient behavior using a simple fluid limit approximation. The transient analysis is initialized with the system state observed at the instants of occurrence of the events, thus it is able to capture their representative time scales, which are instead ignored by the AVG and DEC approximations. An iterative approach is used to identify the average system state seen upon arrival of an event. Experimental results indicate that blending is surprisingly accurate for both state-independent and state-dependent random environments.

The paper is organized as follows. Section II motivates this study using examples. Fluid models and blending are introduced in sections III and IV. Numerical results are given

in Section V. Related work is overviewed in Section VI followed by conclusions in Section VII. The final Appendix proves a characterization result used in Section II.

II. Motivating Example

The challenges in analyzing random environments are illustrated by a simple tandem closed network composed by a first-come first-served queue and a delay station (infinite server queue). We provide two examples which show the two classes of environments studied in this paper, namely a state-independent and a state-dependent one consisting in a MMPP service process.

A. State-independent environment

The population is composed by $N = 2$ jobs which receive an exponential delay with rate λ between successive visits at the queue. The network is immersed in a state-independent Markov random environment that causes server breakdown at the queue with rate β and successive repair with rate α. Thus, the queue normally processes jobs with rate $\mu_1 > 0$, but upon a breakdown it proceeds with reduced rate $0 \leq \mu_0 < \mu_1$ until repaired. This basic model can be described by a continuous-time Markov chain (CTMC) with infinitesimal generator Q immersed in an environment with generator E, where

$$Q = \left[\begin{array}{c|c} Q_0 - \alpha I & \alpha I \\ \hline \beta I & Q_1 - \beta I \end{array}\right] \quad E = \begin{bmatrix} -\alpha & \alpha \\ \beta & -\beta \end{bmatrix}$$

$$Q_0 = \begin{bmatrix} -2\lambda & 2\lambda & 0 \\ \mu_0 & -\mu_0 - \lambda & \lambda \\ 0 & \mu_0 & -\mu_0 \end{bmatrix} \quad Q_1 = \begin{bmatrix} -2\lambda & 2\lambda & 0 \\ \mu_1 & -\mu_1 - \lambda & \lambda \\ 0 & \mu_1 & -\mu_1 \end{bmatrix}$$

where as usual each state in Q represents a reachable state of the network. The states of the random environments are 0 and 1 and are called *stages* in the rest of the paper. Let the equilibrium distribution of the network be the probability vector $\pi = [\pi_0, \pi_1]$ such that $\pi Q = 0$. Here, π_0 (resp., π_1) represents the equilibrium distribution vector for the states in Q_0 (resp., Q_1). We derive in the Appendix the following necessary conditions for π_0 and π_1:

$$\pi_0 \left(\left(\frac{\beta}{\alpha+\beta} \right) Q_0 + \left(\frac{\alpha}{\alpha+\beta} \right) Q_1 - \frac{Q_0 Q_1}{\alpha+\beta} \right) = 0 \quad (1)$$

$$\pi_1 \left(\left(\frac{\beta}{\alpha+\beta} \right) Q_0 + \left(\frac{\alpha}{\alpha+\beta} \right) Q_1 - \frac{Q_1 Q_0}{\alpha+\beta} \right) = 0. \quad (2)$$

This characterization is interesting as it explains the relation between the exact solution $\pi = [\pi_0, \pi_1]$ and the AVG and DEC approximations, as we show next. First, note that the coefficients $\beta/(\alpha+\beta)$ and $\alpha/(\alpha+\beta)$ are simply the equilibrium probabilities of the random environment for states in Q_0 and Q_1, respectively. Thus, $Q^{avg} = \left(\frac{\beta}{\alpha+\beta}\right) Q_0 + \left(\frac{\alpha}{\alpha+\beta}\right) Q_1$ is just an equivalent exponential network where we have averaged the queue rates according to the environment equilibrium, i.e., the AVG approximation for the model under study. Instead, Q_0 and Q_1 are exactly the infinitesimal generators for the subproblems studied in the DEC approximation. Observe now that, as the aggregate event rate $\nu = \alpha + \beta$ decreases, the solution approaches that of the DEC approximation up to scaling factors, i.e.,

$$\lim_{\nu \to 0^+} \pi_0 \left(Q^{avg} - \frac{1}{\nu} Q_0 Q_1 \right) = \pi_0 Q_0 = c_0 \pi_0^{dec} Q_0 = 0$$

$$\lim_{\nu \to 0^+} \pi_1 \left(Q^{avg} - \frac{1}{\nu} Q_1 Q_0 \right) = \pi_1 Q_1 = c_1 \pi_1^{dec} Q_1 = 0$$

where by standard decomposition arguments $c_0 = \beta/(\alpha+\beta)$ and $c_1 = \alpha/(\alpha+\beta)$. Conversely, if ν becomes large compared to the rates in Q_0 and Q_1 one finds that the AVG solution is asymptotically exact:

$$\lim_{\nu \to +\infty} \pi_0 \left(Q^{avg} - \frac{1}{\nu} Q_0 Q_1 \right) = \pi_0 Q^{avg} = c_0 \pi^{avg} Q^{avg} = 0$$

$$\lim_{\nu \to +\infty} \pi_1 \left(Q^{avg} - \frac{1}{\nu} Q_1 Q_0 \right) = \pi_1 Q^{avg} = c_1 \pi^{avg} Q^{avg} = 0$$

Perhaps not too surprisingly, it can be shown that for $\nu \to +\infty$ the service process of the queue seen by an external observer as the environment evolves describes a Poisson process. This confirms the qualitative intuition of the limits.

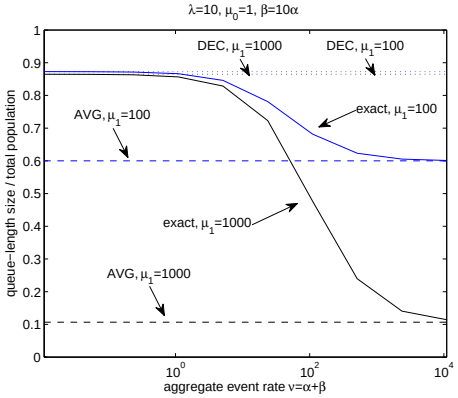
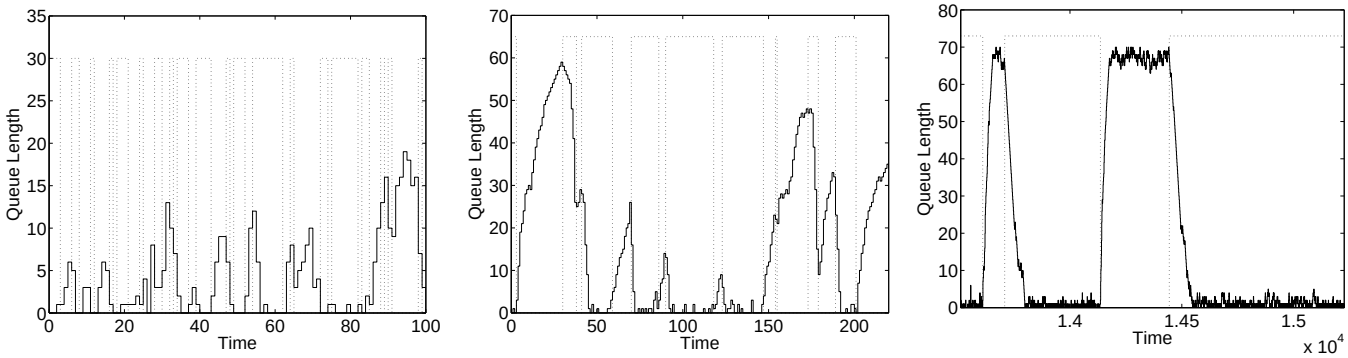


Figure 1: Accuracy of AVG and DEC approximations

Figure 1 illustrates the behavior of the approximation error incurred by AVG and DEC as a function of ν. The considered metric is the ratio between the length of the queue and the total population N, for different values of ν. The experiments show that AVG and DEC are *insensitive* to ν for fixed ratio between α and β, whereas this can significantly affect the values of the queue-length. For general networks, it is not simple to identify the switching point between large and small values of the queue-length since this depends on the dominating eigenvalues of the infinitesimal generators Q_0 and Q_1, which are difficult to estimate. Note also that the the error becomes larger with the variability between the rates, μ_0 and μ_1, thus suggesting that Markov environments describing rare events (e.g., fat-tailed distributions) may be the hardest to approximate by AVG and DEC.



(a) High ν rate (AVG approx. accurate). (b) Intermediate ν rate (no accurate method). (c) Low ν rate (DEC approx. accurate).

Figure 2: Transient analysis of tandem network (delay station and queue) for decreasing aggregate event rate ν. Dashed lines indicate stage transitions in the random environment.

To summarize, AVG and DEC provide asymptotically exact results for models in random environments where the frequency of events is either very large or very small compared to the state transition rates of the system. However, in intermediate cases, (1)–(2) show that the probability distributions π_0 and π_1 depend also on matrices Q_0Q_1 and Q_1Q_0, which are not in general CTMC infinitesimal generators.[1] Furthermore, these matrices do not preserve the structure of Q_0 and Q_1, hence they do not even describe queueing network models. In light of this, our results suggest that there may not exist a robust way to approximate π_0 and π_1 based on simple linear combinations of the equilibrium solutions for Q^{avg}, Q_0, and Q_1.

B. MMPP environment

To better illustrate the intuition given in this section about the impact of the random environment on the behavior of the queueing network, let us consider an environment describing a MMPP service process. It has rate matrix is $\Lambda = \text{diag}(10, 0.01)$ and infinitesimal generator is E as in the previous example. However, the random environment is state-dependent since it may change stage only during busy times of the queue. The job population size—which was previously set to 2 to be able to analyze the network exactly—is now increased to $N = 70$ to show another feature of random environments, i.e., the transient effects due to job migrations.

Figures 2 illustrate representative queue-length sample paths for different values of the aggregate event rate. Figure 2(a) shows the case of a high event rate with $\alpha = 0.60$ and $\beta = 0.30$, where the AVG approximation is typically accurate. This is because the random environment events arrive too frequently with respect to the system representative time scales, hence the network is weakly sensitive to the stage changes.

Figure 2(c) shows the opposite case with rare arrivals of events with $\alpha = 0.006$ and $\beta = 0.003$. Note that the time scale represented on the x-axis is much longer than that in the other plots since stage transitions happen infrequently. Upon arrival of an event, the network enters a long transient period where jobs migrate from the queue into the delay station during the normal state 1; instead, upon a breakdown the jobs quickly accumulate in the queue. DEC typically performs accurately on such models because the length of the transient period remains quite short compared to the overall sojourn time in a stage.

Figure 2(b) considers $\alpha = 0.12$ and $\beta = 0.06$, i.e., a typical condition of comparable time scales between environmental events and service rates. Similarly to the decomposable model in Figure 2(c), the network cyclically enters transient periods where jobs migrate from one station into the other. However, the length of such transients is often comparable to the sojourn time of the random environment in each stage. Noticeably, the queue-length value observed at the instant of occurrence of a stage change is a random variable that is tightly coupled with the specific duration of each stage sojourn time.

In conclusion to this section, the analysis of a sample MMPP environment suggests that one source of inaccuracy in approximate analysis may come from the fact that both AVG and DEC fail to take into account the transient behaviour of the network between stage transitions, whose effects are more pronounced when the environment's time scales are comparable to the service rates. This motivates our interest for the application of transient analysis methods to the problem under study. In order to develop a computationally competitive analytical method, we study the transient behaviour using fluid approximation, as discussed next.

[1] Indeed, π_0 and π_1 are component of the solution for the Markov process Q. Thus, (1)-(2) allow only to draw considerations on the whether such components can also be seen as solutions for a queueing network model *not* in a random environment.

III. The Differential Model

We here focus on the transient analysis of ordinary closed queueing networks with exponential servers, leaving it to Section IV the extension to random environments. Throughout the paper, we consider a closed queueing network with M stations having exponential service rate μ_i. Let s_i be the number of servers at station i, we consider only delay stations ($s_i = +\infty$) and first-come first-serve queues (finite $s_i \geq 1$). The number of circulating jobs is N and the topology is specified by the routing probabilities p_{ij}, $1 \leq i, j, \leq M$, which give the probability that upon departure from station i a job joins station j. The CTMC underlying the queueing network has state vector $n = (n_1, n_2, \ldots, n_M)$, where n_i gives the number of jobs at station i. Let δ_i be a vector of zeros with a 1 in the ith component. The CTMC transitions are

$$n \xrightarrow{p_{ij}\mu_i \min(n_i, s_i)} n - \delta_i + \delta_j, \quad 1 \leq i \leq M. \tag{3}$$

Note that the state space size grows as $O(N^M)$, thus it is prohibitive to evaluate transient performance metrics using Kolmogorov forward equations directly.

A. Deterministic Approximation

We now overview a fluid limit approximation methodology proposed by Kurtz [13] which enables the scalable evaluation of transient and equilibrium behavior of a queueing network. The methodology allows one to obtain a deterministic limit of the CTMC as the network size goes to infinity. The approach considers general population models and holds under mild assumptions. As such, it has found a wealth of applications in a very broad range of disciplines, including chemistry [14], ecology [19], randomized algorithms [17], and communication networks [5].

A *fluid* limit is obtained by taking family of CTMCs $\{X_V(t), V \in \mathbb{N}\}$ indexed by the parameter V that has the interpretation of the system's size. Let $s \equiv (s_1, \ldots, s_M)$ denote an initial vector of server multiplicities and let $\hat{x}$ be the initial state of the CTMC. The CTMC $X_1(t)$ ($V = 1$) is the one with state space reachable from $\hat{x}$ with server multiplicities s. The CTMC $X_2(t)$ ($V = 2$) has initial state $2\hat{x}$ and server vector $2s$, and so on, i.e., $X_V(t)$ has the initial state $V\hat{x}_1$ and server vector Vs, for all V. The fluid approximation of interest here consists in studying the limit behaviour of the *normalized stochastic process* $X_V(t)/V$ for V large. This process is still a CTMC, with rates of order $O(V)$. For a queueing network the transitions of the normalised process become

$$n/V \xrightarrow{Vp_{ij}\mu_i \min(n_i/V, s_i/V)} n/V - \delta_i/V + \delta_j/V,\ 1 \leq i \leq M.$$

Intuitively, for sufficiently large V a sample path of the normalised process takes increasingly small jumps at increasingly fast rates. The transition rates are in the so-called *density-dependent* form $Vf(n/V)$, where n/V is interpreted as a density. This is a continuous approximation of the sample path and is obtained by considering the vector field

$$F(x) = \sum_{n/V \xrightarrow{Vf(n/V)} (n-\delta_i+\delta_j)/V} (\delta_j - \delta_i) f(x)$$
$$= \left\{ \sum_{j=1}^{M} (\delta_j - \delta_i) p_{ij} \mu_i \min(x_i, s_i) \,\middle|\, 1 \leq i \leq M \right\}$$

where the first summation is across all the transitions that define the normalized CTMC. The vector field is set to define the system of ordinary differential equations (ODEs)

$$\frac{dx(t)}{dt} = F(x(t)),$$

where $x(t) = (x_1(t), x_2(t), \ldots, x_M(t))$, where $x_i(t)$ is intended to be the continuous approximation of the sample path of the normalized queue-length process at station i. These equations, together with the initial condition $x(0) = \hat{x}$, give an initial value problem whose solution is the fluid limit of the CTMC in the sense of [13]. Informally it states that a sample path of $X_V(t)/V$, for V large, becomes indistinguishable from $x(t)$ *at all time points* in which the differential equation is defined. In particular, in our model the fluid limit can be shown to imply *convergence in mean* [2]:

$$\lim_{V \to \infty} \mathbb{E}\left\{ |X_V(t)/V - x(t)| \right\} = 0, \quad \text{for any } t,$$

which constitutes the motivation for us to consider the approximation $\mathbb{E}\{X_V(t)\} \approx V x(t)$ that relates the expected queue length to the rescaled differential trajectory. This is the interpretation herein used because the ODE solution will be compared against the mean performance indices of the queueing network.

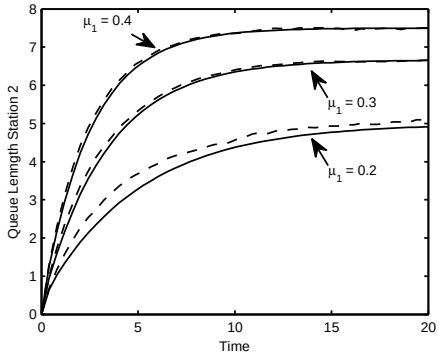
In practice, for a given network under study with initial population vector $\hat{n}$, $\sum_{j=1}^{M} \hat{n}_j = N$, one would solve the ODE system with initial condition $\hat{x} = \hat{n}$, i.e., the first CTMC of the family for the fluid limit is exactly the one under study. Therefore, since we consider $V = 1$ the solution $x(t)$ will need no rescaling.

B. Transient Analysis of Queueing Networks

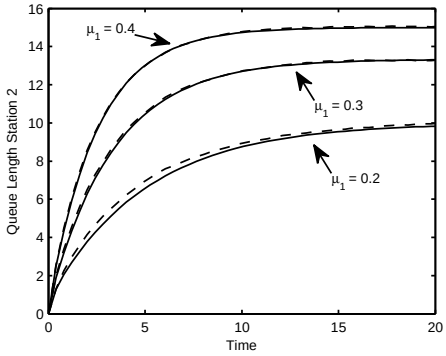
To illustrate the accuracy of the fluid limit technique for transient analysis of queueing networks, we consider again a simple tandem network composed by a delay server with rate μ_1 and a first-come first-served queue with rate μ_2. The ODE system for this running example is

$$\frac{dx_1(t)}{dt} = -\mu_1 x_1(t) + \mu_2 \min(x_2(t), s_2),$$
$$\frac{dx_2(t)}{dt} = +\mu_1 x_1(t) - \mu_2 \min(x_2(t), s_2),$$

Figure 3 shows some representative plots of the CTMC expected values and their fluid approximations during the transient regimes of the running example. Figure 3(a) plots



(a) Plot of $X_1(t)$.



(b) Plot of $X_2(t)$.

Figure 3: Representative comparisons between the expected values and the fluid approximation of the queue length at station 2 in the running example, with $\mu_2 = 1.0$. (a) $\hat{x} = (10, 0), s = (\infty, 1)$; (b) $\hat{x} = (20, 0), s = (\infty, 2)$. Solid lines: ODE solutions; dashed lines: CTMC expected values.

the queue-length processes at station 2 for the first CTMC $\{X_1(t)\}$ of the family induced by the initial conditions $s = (+\infty, 1)$ and $\hat{x} = (10, 0)$. Three instances are considered by varying the delay at station 1, while μ_2 was kept fixed at 1.0. The ODE solutions (solid lines) are solved by standard numerical integration whereas the CTMCs were simulated with tight convergence criteria to obtain smooth trajectories for the expectations (dashed lines). The figure shows a generally acceptable quality of the approximation which here tends to increase with μ_1. The case $\mu_1 = 0.2$ has a maximum error of about 10% (at $t \approx 3.2$) although the trend is still captured well. It is interesting to note that these conditions are quite far away from the many jobs/many servers situation considered by the limit theorem, nevertheless the fluid approximation is still usable.

This is further backed up by Figure 3(b) which considers the CTMC $\{X_2(t)\}$, i.e., a total population of 20 jobs and 2 servers in station 2. Here, the maximum error is reduced to about 6% (at $t \approx 4.0$, for $\mu_1 = 0.2$), and generally behaves very well in all the cases considered. Summarizing, the examples illustrate that fluid limit approximations provide an inexpensive way of evaluating the transient of a queueing network. This feature is at the basis of the blending approximation we introduce in the next section for evaluating queueing in random environments.

IV. Blending

We focus on two-stage environments with generally distributed stage durations. For ease of exposition, we describe the algorithm for the *state-independent Markov* environment with two stages (0 and 1) and infinitesimal generator E which was discussed in Section II. The state-dependent case is presented as an extension in Section IV-C. The generalization concerning environments with more than two stages is discussed in Section VII.

We assume that in environment stage 0, the queueing network has routing probabilities p^0_{ij}, number of servers s^0_i, and service rates μ^0_i, $1 \le i, j \le M$; in stage 1 the network has instead parameterization p^1_{ij}, s^1_i, and μ^1_i; the number of jobs N is always constant.

The influence of the Markov environment is represented at the CTMC level by extending the state vector to (n, e) where the random variable $e \in \{0, 1\}$ describes the active stage. Transitions $(n, e) \to (n', e)$ are defined as in Section III, whereas the stage transitions in the random environment are

$$(n, 0) \xrightarrow{\alpha} (n, 1), \quad (n, 1) \xrightarrow{\beta} (n, 0). \tag{4}$$

As observed in Section II, the activation of stage transitions requires the network to undergo a transient period where jobs migrate to compensate the changes in the system parameters. For instance, under a Markov-modulated service, the stage changes make a server cyclically alternate between fast and slow rates, thus creating job migrations which potentially lead to bottleneck-switch effects, as discussed in [4]. This complicates the analysis of the model by techniques concerned with equilibrium probability distributions. Complications also arise when determining the fluid limit in the sense defined in Section III, since the stage transition rates (4) are not in the density-dependent form $Vf(n/V)$.

To resolve these issues, blending performs a separate fluid analysis for each of the two stages. The inter-dependence between the two stages, i.e., the state of the stations the instant in which the random environment changes, is encoded in the initial conditions provided to each fluid analysis. Hence, the denomination *blending* hints at the fact that we attempt to combine the transient analyses so as to capture the coupling of the two stages.

A. Stage Analysis

The transient analysis of a single stage follows by a direct application of the technique discussed in Section III. That is, for stage $e \in \{0, 1\}$ we consider a system of ODEs

$$\frac{dx^e(t)}{dt} = F^e(x^e(t)),$$

where the superscript e highlights that the queue-lengths and the vector field depend on the network parameters for stage e, i.e., p_{ij}^e, s_i^e, and μ_i^e.

The main challenge of this formulation, addressed in Section IV-B, is to determine an appropriate initial condition $x^e(0) = \hat{x}^e$ such that the equilibrium solution of the global model may be approximated as a function of time-averaged values of the vectors $x^0(t)$ and $x^1(t)$. That is, let L_i^e denote the time-averaged queue length of station i at the instant when the sojourn in stage e elapses, and let L_i be the average queue-length at equilibrium for the global queueing network model. For a Markov environment this may be approximated using the fluid limit technique as

$$L_i = \sum_e w_e L_i^e, \quad L_i^e = \int_0^{+\infty} x_i^e(t) p_e(t) dt, \qquad (5)$$

for $1 \leq i \leq M$, where w_e is the probability that the random environment is in stage e at equilibrium and $p_e(t)$ is the probability density function for the sojourn time in stage e. For instance, in the example two-state Markov environment we have $p_0(t) = \alpha e^{-\alpha t}$ and $p_1(t) = \beta e^{-\beta t}$. From (5), the average queue length at equilibrium is then

$$L_i = \left(\frac{\beta}{\alpha+\beta}\right) L_i^0 + \left(\frac{\alpha}{\alpha+\beta}\right) L_i^1$$

The system throughput X_j, measured at the output of an arbitrary reference station j, is similarly approximated by

$$X_j = \sum_e w_e X_j^e, \quad X_j^e = \int_0^{+\infty} \mu_j \min(x_j^e(t), s_j) p_e(t) dt.$$

Other performance measures such as utilizations and waiting times are approximated[2] from mean queue-lengths and throughputs using Little's law and basic operational analysis formulas [15, Chap. 3]. Finally, we remark that for semi-Markov random environments, where sojourn times in a stage might not be exponentially-distributed, our technique generalizes easily by replacing $p_e(t)$ with the sojourn time distribution for stage e.

B. Blending Algorithm

The choice of suitable initial conditions $\hat{x}^e$ for the ODE model requires a technique to determine the number of jobs seen in the queues at the instant of a random environment transition into stage e. In general, this is a random variable with a distribution dependent on the sojourn time in the previous stage, as shown in Section II.B. The blending approximation simplifies this representation by using the estimated expectation of such a variable as the initial condition $\hat{x}^e$. In other words, $\hat{x}_i^0$ would be set to L_i^1, and, conversely we would let $\hat{x}_i^1 = L_i^0$. An iterative scheme, initialized with the initial condition of the overall network under study, breaks this circular dependence.

[2]In a strict sense, transient metrics the usual Little's law and operational analysis formulas do not apply. However, these may be used as reasonable first approximations.

1. $k = 0$;
2. $L^0(k) = L^1(k) = \hat{x}$, $\sum_i \hat{x}_i = N$;
3. **repeat:**
   - 3a. $k = k + 1$;
   - 3b. *Stage 0*
     Solve the initial value problem:
     $$dx^0(t)/dt = F^0(x^0(t)), \quad \hat{x}^0 = L^1(k-1)$$
   - 3c. Compute, for all $1 \leq i \leq M$:
     $$L_i^0(k) = \int_0^{+\infty} x_i^0(t) p_0(t) dt$$
   - 3d. Compute system throughput at reference station j:
     $$X_j^0(k) = \int_0^{+\infty} \mu_j \min(x_j^0(t), s_j) p_0(t) dt$$
   - 3e. *Stage 1*
     Solve the initial value problem:
     $$dx^1(t)/dt = F^1(x^1(t)), \quad x^1(0) = L^0(k-1)$$
   - 3f. Compute, for all $1 \leq i \leq M$:
     $$L_i^1(k) = \int_0^{+\infty} x_i^1(t) p_1(t) dt$$
   - 3g. Compute system throughput at reference station j:
     $$X_j^1(k) = \int_0^{+\infty} \mu_j \min(x_j^1(t), s_j) p_1(t) dt$$
4. **until** $\max_{1 \leq i \leq M} |L_i^0(k) - L_i^0(k-1) + L_i^1(k) - L_i^1(k-1)| < \varepsilon$;
5. Compute mean queue lengths and throughputs:
   $$L_i = w_0 L_i^0(k) + w_1 L_i^1(k), \text{ for all } 1 \leq i \leq M$$
   $$X_j = w_0 X_j^0(k) + w_1 X_j^1(k)$$
6. **return** L_i, X_j, $1 \leq i \leq M$, $1 \leq j \leq M$.

Figure 4: Algorithmic description of *blending* for generally distributed environmental transitions between two stages. The variable k ranges over the iterations and $A(k)$ means the value of the estimate A computed at the k-th iteration.

The algorithm, shown in Figure 4, runs until the difference between the estimates of two successive iterations is less than a threshold ε, set to 10^{-6} in our experiments. We have not observed cases where this technique does not converge, apart for rare cases where very small oscillations ($< 1\%$) around a fixed-point are presumably due to the numerical tolerances of the ODE integrators. Such cases can be easily handled by averaging the oscillating values.

We also remark that blending implicitly carries the fol-

lowing assumption on the class of models that can be evaluated by the approach. Blending implicitly assumes that upon arrival of an event from the environment *all* jobs are affected by the stage change. This is true for queues with first-come first-server scheduling, but it is not necessarily representative of other disciplines such as processor sharing. For instance, assume that a queue has a Markovian arrival process such that requests alternate *deterministically* (i.e., in a cyclic manner) between a fast rate μ_1 and a slow rate μ_0. With our present definition, upon arrival of a stage change while the current stage is 1, all jobs in the queues would switch into rate μ_0, which introduces an approximation error with the Markov model where only half of them could be served with a given rate. We have observed this error to be significant on some models.

C. State-Dependent Environments

The generalization of the blending algorithm in Figure 4 is simple and involves a different definition of the sojourn time densities $p_0(t)$ and $p_1(t)$. The main issue is that, if the environment events are enabled only while the network is in a subset of states $\mathcal{S}$, then the rate of stage change in the environment should be scaled by the probability $\Pr(\mathcal{S})$ of being in such a subset. In blending, we simply assume that rates are directly multiplied by probabilities as in steady-state; this is an approximation because event rates are used to weight transient analysis results, not equilibrium values. That is, events become slower proportionally to the probability $\Pr(\mathcal{S})$. In the fluid limit approximation, restrictions arise on the ability of computing individual probability terms, while the computation of mean values (e.g., utilization and queue-lengths) is simple and follows the formulas discussed in Section IV-A. In particular, for Markov-modulated service processes in station j one would set during the i-th iteration of blending:

$$p_0(t) = \alpha^{(k)} e^{-\alpha^{(k)} t}, \quad \alpha^{(k)} = X_j^0(k-1)\alpha/\mu_j^0 \qquad (6)$$

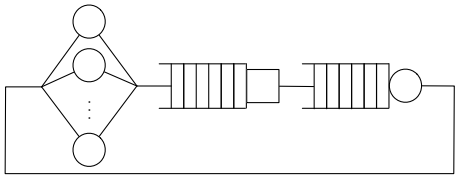
$$p_1(t) = \beta^{(k)} e^{-\beta^{(k)} t}, \quad \beta^{(k)} = X_j^1(k-1)\beta/\mu_j^1 \qquad (7)$$

where $X_j^0(k-1)$ and $X_j^1(k-1)$ are the throughputs computed at iteration $k-1$.

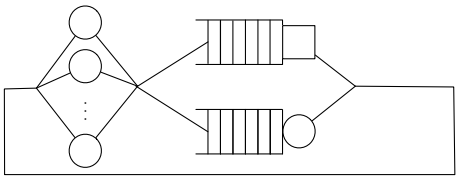
V. Numerical Results

This section assesses the quality of the blending approximation on state-independent and state-dependent Markov environments. The aim is to perform a comparative evaluation of our methodology against the average-environment and decomposition methods.

The validation considers networks with $M = 3$ stations and N jobs, where N is varied such that the CTMCs are kept sufficiently small to allow a direct numerical solution. Blending scales well to larger networks since the system of ODEs grows in cardinality efficiently as $O(M)$. However, for validating purposes, M was not changed because even tiny networks have been shown to give very useful insights into the error behavior of analytical approximations (cf. [7] for a notable example).



(a) Cyclic Network



(b) Parallel Network

Figure 5: Case Studies Topologies. The switching queue is represented with a boxed server.

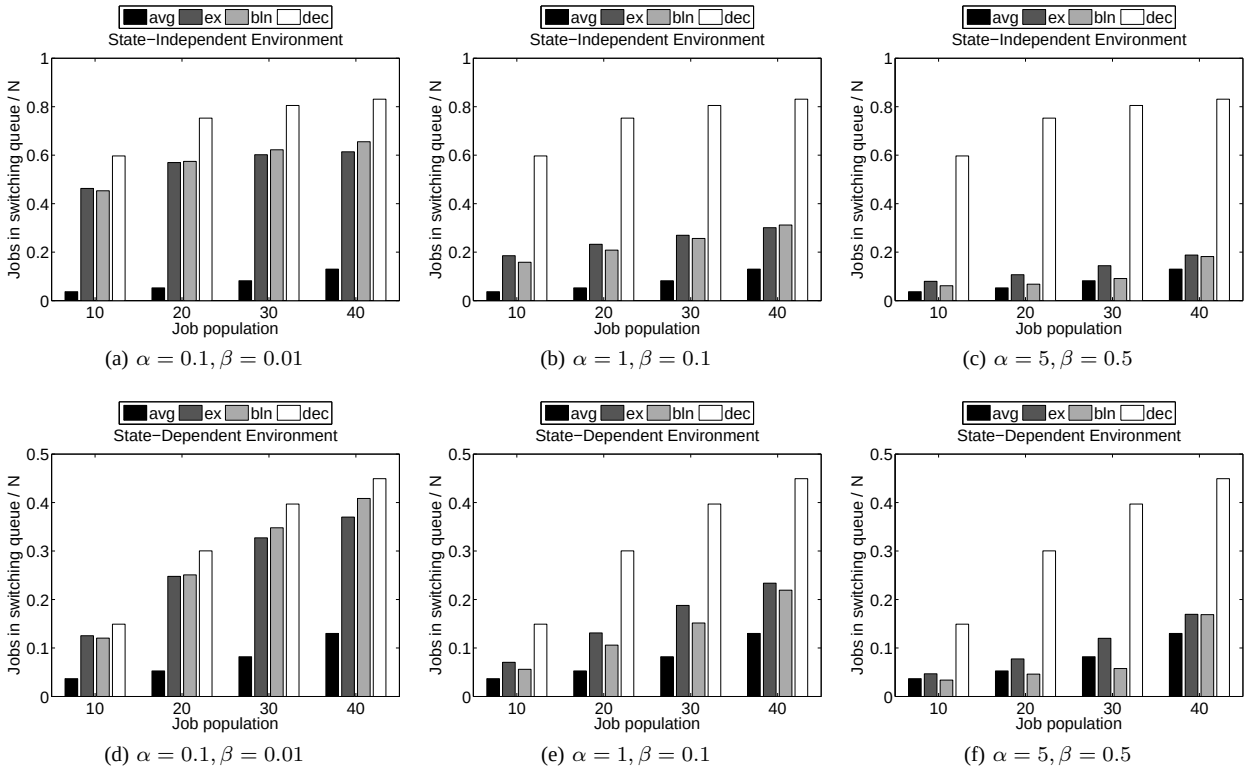
Due to limited space, we focus on random environments that change the service rate of a single queue, referred to as the *switching queue*. The metric of interest is the queue-length value for such a queue. For ease of interpretation, we consider a set of representative case studies which show the behavior of the methods under varying network topologies and different values for service times and job populations. We believe this to be more representative than random experiments which often tend to skew the results towards models with a dominating bottleneck queue.

We consider both the state-independent and the state-dependent cases. For the latter, we assume that the environment stage can change only while the switching queue is busy, as introduced in Section II. Thus, such an environment defines service times at the switching queue that are governed by a Markov-Modulated Poisson Process (MMPP) with infinitesimal generator E and rate matrix $\Lambda = \text{diag}(\mu_2^0, \mu_2^1)$.

Another choice in our parametrization is to consider models with a limited number of servers and jobs, even though this is sufficient to cover utilization values for the switching queue in the range 0.30–0.90. It is worth noticing that these may be regarded as very unfavourable worst-case conditions for the fluid approximation, which is instead expected to yield more accurate estimates for larger population sizes.

The following two case studies are investigated, with topologies shown in Figure 5 where the switching queue is represented by a boxed server.

Cyclic Network. This is a case of a tandem network with $M = 3$ stations having $s_1 = +\infty$, $s_2 = 1$, and $s_3 = 1$ servers, see Figure 5(a). This topology simplifies the redistribution of jobs across the two queues since jobs find no obstacles in migrating to another queue. The service



(a) $\alpha = 0.1, \beta = 0.01$ (b) $\alpha = 1, \beta = 0.1$ (c) $\alpha = 5, \beta = 0.5$

(d) $\alpha = 0.1, \beta = 0.01$ (e) $\alpha = 1, \beta = 0.1$ (f) $\alpha = 5, \beta = 0.5$

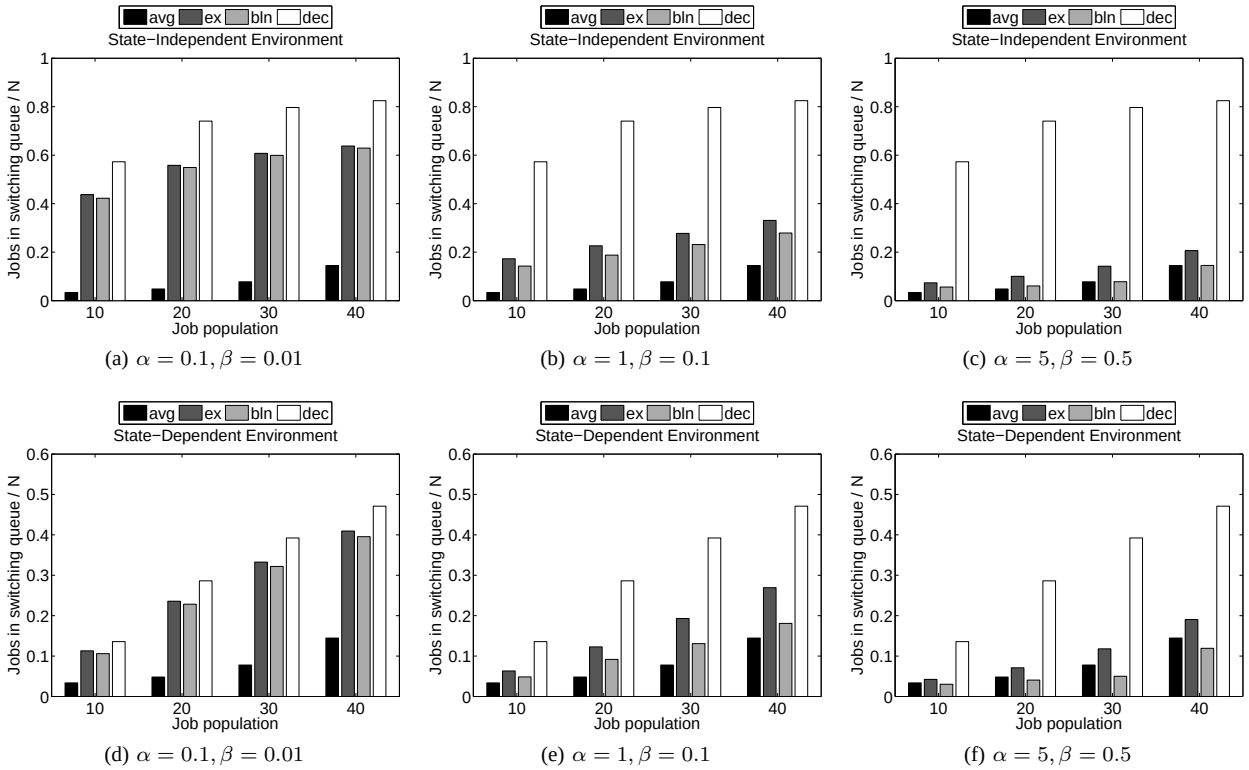
Figure 6: Case Study: Cyclic Network with $\mu_1 = 0.03, \mu_2^0 = 10, \mu_2^1 = 0.1, \mu_3 = 1$. The topmost figures are for a state-independent Markov environment; the bottomost figures for a state-dependent one (MMPP service).

rates are $\mu_1 = 0.03, \mu_2^0 = 10, \mu_2^1 = 0.1, \mu_3 = 1$. Results are shown in Figure 6.

Parallel Network. A parallel topology, shown in Figure 5(b), is considered using the same server rates and multiplicities as in the cyclic network. The branch towards the switching queue has routing probability $p_{12} = 0.9$, making it hard for the jobs to migrate from the switching queue into the other exponential queue due to the high probability of loops. Results are shown in Figure 7.

Discussion. By performing an experimental campaign on several model parameterizations, not limited to the ones shown in the figures, we have found the following qualitative considerations arising from Figure 6 and Figure 7 to be representative of the performance of blending with respect to the AVG and DEC approximations:

- As already noted in Section II, AVG and DEC are *insensitive* to the time scales of the random environment. This is evident by comparing plots on the same rows of Figure 6 and Figure 7, which show that the queue lengths predicted by AVG and DEC are always the same.
- Conversely, blending is highly sensitive to the parametrization of the random environment and yields excellent accuracy in general. As expected, the case of a parallel network appears to be more difficult than the cyclic network, but the overall accuracy is affected in a non-negligible way in a single case, the state-dependent model for $\alpha = 1$ and $\beta = 0.1$.
- As expected, state-dependent environments are harder to approximate than state-independent ones—again, this is visible in particular in the intermediate case with $\alpha = 1$ and $\beta = 0.1$.
- Relatively to AVG and DEC, blending outperforms DEC in all the cases considered here. It improves significantly over AVG except for the case $\alpha = 5$ and $\beta = 0.5$. A reason for this is that in this case the environment evolves very quickly with respect to the service rates. This complicates the estimation of the queue-lengths L_i^0 and L_i^1 observed at occurrence of stage transitions. In fact, we observe that for such models the number of required iterations for blending is in the range 30–200, slightly higher than in the state-independent case. Conversely, for $\alpha = 1$ and $\beta = 0.1$ the range is 9–40 iterations, while for $\alpha = 0.1$ and $\beta = 0.01$ it is just 5–11 iterations. Despite a slightly better accuracy of AVG for $\alpha = 5$ and $\beta = 0.5$, we stress that blending turns out to be much more robust than both AVG and DEC across all the situations considered in this validation study. Indeeed, AVG and DEC both grossly fail to capture the network's



(a) $\alpha = 0.1, \beta = 0.01$ (b) $\alpha = 1, \beta = 0.1$ (c) $\alpha = 5, \beta = 0.5$

(d) $\alpha = 0.1, \beta = 0.01$ (e) $\alpha = 1, \beta = 0.1$ (f) $\alpha = 5, \beta = 0.5$

Figure 7: Case Study: Parallel Network with $\mu_1 = 0.03, \mu_2^0 = 10, \mu_2^1 = 0.1, \mu_3 = 1$. The topmost figures are for a state-independent Markov environment; the bottommost figures for a state-dependent one (MMPP service).

behavior in a large number of cases; particularly, AVG in decomposable networks and DEC in models with high event rate.

- In all experiments the algorithm converges without numerical difficulties. A prototype MATLAB implementation has runtimes per iteration in the range 100–200 ms on an ordinary personal computer, with the majority of cases being around 130 ms. Indeed, we expect an optimized implementation in a compiled programming language to be at least one order of magnitude faster. The memory occupation was negligible in all experiments.

Summarizing, the above results indicate that blending is able to capture the qualitative trend of a queueing network in a random environment. The main result is that the method is *sensitive* to the representative time scale of the random environment events, whereas AVG and DEC are *insensitive* as they depend only on the ratio between α and β, but not on their absolute values. Overall, this makes blending a valuable new approach for accurate and scalable analysis of queueing models in random environments.

VI. Related Work

Queueing models in random environments have been investigated for several decades as a tool for modelling problems in reliability and performance evaluation [18], [16], [1]. Economou derives in [8] necessary and sufficient conditions for a queueing network model in a random environment to enjoy a product-form solution. The RCAT methodology can be used to determine similar characterizations in state-dependent environments [11], [10]. Approximations are instead the main analysis tool for models without product-form solution, popular examples are the average-environment approach [1], which reduces to product-form analysis for queueing networks where service is exponential in each environment state, and decomposition methods [6], [21].

Recent work in approximation techniques has mainly investigated queueing networks with arrival and service processes of Markov-modulated type [12], [3]. In [4] it is shown that ingoring the random environment behavior at departure instants of jobs can lead to severe underestimation of performance indices. Our approach differs from the above methods in focusing on the evaluation of the system state observed at arrival instants of environment events (i.e., the instant of phase change in a Markov-modulated service or arrival process). This allows to apply transient analysis techniques that are not readily applicable to the standard model description.

To the best of our knowledge, the only method that proposes the application of transient analysis methods for

Markov processes to describe a system in a random environment has been recently proposed by Romano *et al.* [20]. Based on analytical expression for the $M/M/1$ queue transient state, accurate equilibrium approximations for a $MMPP/M/1$ queue are derived by numerical methods. Our work differs from this technique in several ways. First, we use fluid limits for scalable transient analysis in models that suffer the steady-state explosion problem. Furthermore, blending tackles the difficult issue of determining the equilibrium of the system at the arrival instant of environment events.

VII. Conclusion

Blending is a new technique for the analysis of queueing models in random environments. The approach defines a system of ordinary differential equations (ODEs) to estimate the mean queue-lengths at the instants of occurrence of random environment events. These values are used to initialize coupled ODE systems by an iterative approach that returns an approximate solution for the model. Comparison with average-environment and decomposition approximations confirms that blending captures trends in performance indexes that established techniques do not capture.

A natural extension of this work would be to consider environments with more than two states, where the fluid limits would hold with minor changes. Instead, it is unclear if the time-averaged queue-lengths L_j^e would be sufficient for characterizing the transients in such models since these may show correlations with the last stage (or sequence of stages) visited before entering in stage e. Another generalization of interest would involve the definition of blending for multi-class models. This is complicated by the problem of representing first-come first-served scheduling in a multi-class setting within the fluid approach.

Acknowledgement

The work of Giuliano Casale was supported by the Imperial College Junior Research Fellowship. Mirco Tribastone is supported by the EU project ASCENS.

References

[1] V. V. Anisimov. *Switching Processes in Queueing Models*. Wiley, 2008.

[2] P. Billingsley. *Probability and Measure*. Wiley, 3rd edition, 1995.

[3] G. Casale, N. Mi, and E. Smirni. Bound analysis of closed queueing networks with workload burstiness. In *Proc. of ACM SIGMETRICS 2008*, pages 13–24. ACM Press, 2008.

[4] G. Casale and E. Smirni. MAP-AMVA: Approximate mean value analysis of bursty systems. In *DSN*, pages 409–418. IEEE, 2009.

[5] A. Chaintreau, J.-Y. L. Boudec, and N. Ristanovic. The age of gossip: spatial mean field regime. In *SIGMETRICS/Performance*, pages 109–120, 2009.

[6] P. Courtois. Decomposability, instabilities, and saturation in multiprogramming systems. *Comm. of the ACM*, 18(7):371–377, 1975.

[7] D. L. Eager and J. N. Lipscomb. The AMVA priority approximation. *Perf. Eval.*, (8):173–193, 1988.

[8] A. Economou. Generalized product-form stationary distributions for markov chains in random environments with queueing applications. *Advances in Applied Probability*, 37(1):pp. 185–211, 2005.

[9] A. Gandhi, V. Gupta, M. Harchol-Balter, and M. A. Kozuch. Optimality analysis of energy-performance trade-off for server farm management. *Perform. Eval*, 67(11):1155–1171, 2010.

[10] G.Casale, P.G.Harrison, and M.G.Vigliotti. Product-form approximation of tandem queues via matrix-geometric methods. In *Proc. of NSMC 2010 Workshop*, Sept. 2010.

[11] P. G. Harrison. Turning back time in markovian process algebra. *Theor. Comput. Sci*, 290(3):1947–1986, 2003.

[12] A. Horváth, G. Horváth, and M. Telek. A joint moments based analysis of networks of MAP/MAP/1 queues. *Perform. Eval*, 67(9):759–778, 2010.

[13] T. G. Kurtz. Solutions of ordinary differential equations as limits of pure Markov processes. *J. Appl. Prob.*, 7(1):49–58, April 1970.

[14] T. G. Kurtz. The relationship between stochastic and deterministic models for chemical reactions. *The Journal of Chemical Physics*, 57(7):2976–2978, 1972.

[15] E. D. Lazowska, J. Zahorjan, G. S. Graham, and K. C. Sevcik. *Quantitative System Performance*. Prentice-Hall, 1984.

[16] I. Mitrani. *Probabilistic Modelling*. Cambridge University Press, 1998.

[17] M. Mitzenmacher. *The power of two choices in randomized load balancing*. PhD thesis, University of California at Berkeley, Department of Computer Science, Berkeley, CA, 1996.

[18] M. F. Neuts. *Matrix-Geometric Solutions in Stochastic Models, an Algorithmic Approach*. The Johns Hopkins University Press, Baltimore, MD, 1981.

[19] P. Pollet. On a model for interference between searching insect parasites. *J. Austral. Math. Soc. Ser. B*, 32(2):133–150, 1990.

[20] P. Romano, B. Ciciani, A. Santoro, and F. Quaglia. Fast computation of hyper-exponential approximations of the response time distribution of MMPP/M/1 queues. In *Annual Simulation Symposium*, pages 113–120. IEEE Computer Society, 2008.

[21] J. Zahorjan, E. D. Lazowska, and R. L. Garner. A decomposition approach to modelling high service time variability. *Performance Evaluation*, 3:35–54, 1983.

APPENDIX

In order to prove (1)-(2), let us first study the evolution of the Markov process Q embedded at arrival instants of repair events. Define $\eta_0^{(i)}$ to be the exit probability distribution vector from Q_0 after the ith repair event. Then for the Markovian random environment under study

$$\eta_0^{(i)} = \eta_0^{(i-1)} \int_0^{+\infty} \int_0^{+\infty} \alpha e^{(Q_1 - \beta I)b_i} \beta e^{(Q_0 - \alpha I)r_i} db_i dr_i$$

for $i \geq 1$, where b_i (resp. r_i) denote the inter-arrival time of the jth breakdown (resp. repair) in the random environment. However, since for a subgenerator A it is $\int_0^{\infty} e^{At} dt = (-A)^{-1}$, we write

$$\eta_0^{(i+1)} = \alpha\beta\eta_0^{(i)}(Q_1 - \beta I)^{-1}(Q_0 - \alpha I)^{-1}$$

Observe now that a limiting vector $\eta_0 = \lim_{i \to +\infty} \eta_0^{(i)}$ surely exists if Q admits an equilibrium distribution π, since $\eta_0 = \pi_0\alpha/\alpha_{tot}$, being α_{tot} the aggregate arrival rate of repair events. (Note that for state-independent random environments $\alpha_{tot} = \alpha$, while in general it may depend on the probability mass of states where random environments events are enabled.) Thus we have

$$\eta_0 = \alpha\beta\eta_0(Q_1 - \beta I)^{-1}(Q_0 - \alpha I)^{-1}$$

or equivalently since $\eta_0 \propto \pi_0$

$$(\alpha\beta I - (Q_0 - \alpha I)(Q_1 - \beta I))\,\pi_0 = 0$$

Using a similar argument for the exit distribution η_1 after a breakdown we write

$$(\alpha\beta I - (Q_1 - \beta I)(Q_0 - \alpha I))\,\pi_1 = 0$$

The characterization (1)-(2) is then obtained with simple passages after expanding the products and dividing both sides by $\alpha + \beta$.