Centrum voor Wiskunde en Informatica

REPORT*RAPPORT*

PNA

Probability, Networks and Algorithms

Probability, Networks and Algorithms

Importance sampling in rate-sharing networks

P.M.D. Lieshout, M.R.H. Mandjes

REPORT PNA-R0709 OCTOBER 2007

Centrum voor Wiskunde en Informatica (CWI) is the national research institute for Mathematics and Computer Science. It is sponsored by the Netherlands Organisation for Scientific Research (NWO).
CWI is a founding member of ERCIM, the European Research Consortium for Informatics and Mathematics.

CWI's research has a theme-oriented structure and is grouped into four clusters. Listed below are the names of the clusters and in parentheses their acronyms.

Probability, Networks and Algorithms (PNA)

Software Engineering (SEN)

Modelling, Analysis and Simulation (MAS)

Information Systems (INS)

P.O. Box 94079, 1090 GB Amsterdam (NL)
Kruislaan 413, 1098 SJ Amsterdam (NL)
Telephone +31 20 592 9333
Telefax +31 20 592 4199

ISSN 1386-3711

Importance sampling in rate-sharing networks

ABSTRACT
We consider a network supporting elastic traffic, where the service capacity is shared among the various classes according to an alpha-fair sharing policy. Assuming Poisson arrivals and exponentially distributed service requirements for each class, the dynamics of the user population may be described by a Markov process. We focus on the probability that, given the network is in some state n_0 at time 0, the network is in some set of states A (not containing n_0) at time T. In particular, we assume that the underlying event is rare, i.e., the probability of interest is small. As in general no explicit expressions are known for this probability, an attractive approach may be to resort to Monte-Carlo (MC) simulation. However, due to the rarity of the event under consideration, MC simulation is infeasible. A natural approach to speed up the simulation is to use Importance Sampling (IS). We present an IS algorithm to accelerate the simulation that is based on large deviations results. With extensive simulation experiments we assess the performance of the algorithm; under rather general conditions a considerable speed-up is achieved.

2000 Mathematics Subject Classification: 60K25
Keywords and Phrases: Alpha-fair sharing; Importance sampling; Rare events
Note: This research has been funded by the Dutch Bsik/BRICKS (Basic Research in Informatics for Creating the Knowledge Society) project.

Importance Sampling in Rate-Sharing Networks

P. Lieshout[1] and M. Mandjes[1,2,3]

[1]CWI
P.O. Box 94079, 1090 GB Amsterdam, the Netherlands

[2]Kortweg-de Vries Institute
University of Amsterdam
Plantage Muidergracht 24, 1018 TV Amsterdam, the Netherlands

[3]EURANDOM
P.O. Box 513, 5600 MB Eindhoven, the Netherlands

Email: lieshout@cwi.nl and mmandjes@science.uva.nl

5th October 2007

Abstract

We consider a network supporting elastic traffic, where the service capacity is shared among the various classes according to an alpha-fair sharing policy. Assuming Poisson arrivals and exponentially distributed service requirements for each class, the dynamics of the user population may be described by a Markov process. We focus on the probability that, given the network is in some state n_0 at time 0, the network is in some set of states A (not containing n_0) at time T. In particular, we assume that the underlying event is rare, i.e., the probability of interest is small. As in general no explicit expressions are known for this probability, an attractive approach may be to resort to Monte-Carlo (MC) simulation. However, due to the rarity of the event under consideration, MC simulation is infeasible. A natural approach to speed up the simulation is to use Importance Sampling (IS). We present an IS algorithm to accelerate the simulation that is based on large deviations results. With extensive simulation experiments we assess the performance of the algorithm; under rather general conditions a considerable speed-up is achieved.

Categories and Subject Descriptors

C.4 [**Performance of systems**]: Design Studies; D.4.8 [**Performance**]: Queueing Theory

General Terms

Design, Performance

Keywords

Alpha-fair sharing, Importance sampling, Rare events

1 Introduction

Over the past several years the Processor-Sharing (PS) discipline has been widely used for evaluating the flow-level performance of elastic data transfers competing for bandwidth on a single bottleneck link. In a multi-link setting, bandwidth-sharing networks as considered by e.g. Massoulié & Roberts [13] provide a natural extension for modeling the dynamic interaction among competing elastic flows.

It is well-known that the queue length distribution in a single-server PS system with Poisson arrivals has a simple geometric distribution that only depends on the service requirement distribution through its mean. The distribution of the number of active users in bandwidth-sharing networks with several nodes has, in contrast, remained generally intractable, even for exponentially distributed service requirements. Bonald & Massoulié [5] established the crucial result that the wide family of *alpha-fair bandwidth-sharing* policies as introduced by Mo & Walrand [14] achieve stability under the simple condition that no individual link is overloaded. The family of alpha-fair policies covers several common notions of fairness as special cases, such as max-min fairness ($\alpha \to \infty$), Proportional Fairness ($\alpha \to 1$), and maximum throughput ($\alpha \downarrow 0$). In [15] it has also been shown that the case $\alpha = 2$, with additionally class weights set inversely proportional to the respective round trip times, provides a reasonable modeling abstraction for the bandwidth sharing realized by TCP (Transmission Control Protocol) in the Internet.

In this paper we consider a network operating under a alpha-fair sharing policy. Since the service rate allocated to a flow is restricted in practice, we impose class-dependent access-link rate limitations, similar as in [2]. Assuming Poisson arrivals and exponentially distributed service requirements for each class, the dynamics of the user population may be described by a Markov process.

An essential requirement of modern bandwidth-sharing networks is their capability of providing a variety of Quality-of-Service (QoS) guarantees, where QoS is usually expressed in term of constraints on a set of performance measures, such as mean transfer delays, but also the probability that there are many flows (per class) active in the network. Typically, such a probability is required to be below some small threshold, as this can prevent flows from experiencing large delays. Motivated by this, we analyze in this paper the probability that, given that the network is in some specific state n_0 at time 0, the network is in some set of states A after some predefined time T. In particular, we assume that the underlying event is rare, i.e., this probability is small. As in general no explicit expressions are known for the probability of interest, an attractive approach may be to resort to *Monte-Carlo* (MC) *simulation*. In general, one may say that the number of runs needed to obtain an estimate with predefined accuracy and confidence, is inversely proportional to the probability to be estimated [10], implying that MC simulation is infeasible due to the rarity of the event under consideration. A natural method to accelerate the simulation is to use *Importance Sampling* (IS). The idea underlying IS is to simulate the system with a new set of input probability distributions, i.e., new interarrival and service time distributions, such that the rare event becomes more likely, and then to correct the simulation output with appropriate likelihood ratios, in order to obtain an unbiased estimate.

To obtain promising new input distributions we first identify the *most probable path* (MPP) for the event to occur. Informally speaking, given that this rare event occurs, with overwhelming probability it will happen by a path close to this MPP. For the M/M/1-PS queue the MPP is

already known [16], whereas this is not the case for a general alpha-fair sharing network. We develop an approach for detecting the MPP, that exploits the *large deviations* results of [16]. The underlying idea is that locally the flow-level dynamics of a particular class in the network can be approximated as a M/M/1-PS queue. It is noted that, in contrast to the M/M/1-PS queue where the most likely path has a linear shape, in case of a general alpha-fair sharing network the MPP has a non-linear shape. The path is then subsequently translated into new input distributions, that are such that the event under consideration occurs by realizations close to this MPP.

Extensive numerical experiments indicate that the above approach is quite effective: we are able to estimate probabilities up to 10^{-13} quickly. It is emphasized that we do not prove that our IS technique is *asymptotically optimal* or *asymptotically efficient* [6]. The numerical experiments, however, suggest that the IS scheme is close is to being asymptotically optimal.

The remainder of the paper is organized as follows. In Section 2 we first provide a detailed model description, discuss the use of IS, and present a key large deviations theorem. Section 3 deals with the M/M/1-PS queue, which is in fact a special case of our network. In Section 4 we derive (that is, approximate, but the approximation can be made arbitrarily close) the MPP for a rare event to occur in a general alpha-fair sharing network, by exploiting the results of the M/M/1-PS queue. Section 5 shows how one can translate this MPP into new input distributions that can be incorporated in an IS algorithm. The pseudo-code of the IS algorithm is presented in an Appendix. Section 6 examines the performance of the IS algorithm for two special networks, and shows that the IS scheme performs well. Finally, Section 7 concludes with some final observations.

2 Preliminaries

In this section we first describe our queueing model. Next we discuss IS, a simulation technique designed for estimating rare event probabilities. Finally, we briefly discuss some large deviations results, which are needed in the analysis.

2.1 Queueing model

We consider a network consisting of L nodes, where node j has capacity c_j, $j = 1, \ldots, L$. There are I classes of users in the network, where each class corresponds to a specific route in the network. We assume that class-i users arrive according to a Poisson process of rate λ_i, and have i.i.d. exponentially distributed service requirements with mean μ_i^{-1}, $i = 1, \ldots, I$. The arrival processes and service distributions are all assumed to be independent. The traffic load of class i is then $\rho_i := \lambda_i/\mu_i$, $i = 1, \ldots, I$. If a user requires service at multiple nodes, then we assume that it is served at all nodes simultaneously. Let $S(j)$ denote the set of classes that require service at node j, $j = 1, \ldots, L$. Finally, let $N(t) = (N_1(t), \ldots, N_I(t)) \in \mathbb{N}_0^I$ be a vector denoting the state of the network at time $t \geq 0$, with $N_i(t)$ representing the number of class-i users at time $t \geq 0$.

The network operates under a so-called alpha-fair sharing policy, as introduced in [14]. When the network is in state $n = (n_1, \ldots, n_I) \neq 0$, the service rate x_i^* allocated to each of the class-i

users is obtained by solving the following optimization problem:

$$\begin{array}{rc} \max & \sum_{i=1}^{I} U_i(x_i) \\ \text{subject to} & \sum_{i \in S(j)} n_i x_i \leq c_j, \quad j = 1, \ldots, L \\ \text{over} & x_i \geq 0, \quad i = 1, \ldots, I, \end{array}$$

where the utility function $U_i(x_i)$ is defined by

$$U_i(x_i) := \begin{cases} \kappa_i n_i \frac{x_i^{1-\alpha}}{1-\alpha} & \text{if} \quad \alpha \in (0, \infty) \backslash \{1\}; \\ \kappa_i n_i \log x_i & \text{if} \quad \alpha = 1. \end{cases}$$

The κ_is are non-negative weights, and $\alpha \in (0, \infty)$ can be interpreted as a fairness coefficient. The cases $\alpha \to 0$, $\alpha \to 1$ and $\alpha \to \infty$ correspond to allocations which achieve maximum throughput, proportional fairness, and max-min fairness, respectively. If $\alpha = 2$ and κ_i is the reciprocal of the square of the round-trip time on path i, then alpha-fair sharing approximates the allocation that is achieved by the congestion avoidance algorithm of TCP.

Let $s_i(n) := n_i x_i^*$ denote the total service rate allocated to class i. Since the rate allocated to single flows is often restricted in practice, we assume that the effective total rate allocated to class-i users is [2]

$$d_i(n) := \min \{s_i(n), n_i r_i\},$$

where r_i can be thought of as the access-link rate limitation for a class-i flow, $i = 1, \ldots, I$.

Then it can be proven that $N(t)$ is a Markov process with state space $\mathbb{N}_0^I$, equipped with transition rates:

$$q(n, n + e_i) = \lambda_i; \quad q(n, n - e_i) = \nu(n), \quad i = 1, \ldots, I,$$

where $\nu_i(n) := \mu_i d_i(n)$. Given that $r_i \geq c_i$, $i = 1, \ldots, I$, i.e., given that there are *no* access-link rate limitations, Bonald and Massoulié [5] showed the plausible result that $N(t)$ is an ergodic Markov process if

$$\sum_{i \in S(j)} \rho_i < c_j, \quad j = 1, \ldots, L. \tag{1}$$

Since the down rates of our system differ only for a finite number of states from those in a similar system without rate limitations, it follows from Proposition 1 in [11] that $N(t)$ is ergodic for all values of $r_i > 0$, $i = 1, \ldots, I$, given that (1) holds. We emphasize that in general no explicit expressions are known for the steady-state distribution of $N(t)$.

In this paper our goal is to estimate

$$P := \mathbb{P}(N(T) \in A | N(0) = n_0),$$

i.e., the probability that, given that network is in state n_0 at time 0, the state of the network at time $T > 0$ is contained in set A. For example, here n_0 might be a state where the network operates around most of the time, and A might be an 'overflow set':

$$\left\{ (x_1, \ldots, x_I) \geq 0 | \sum_{i=1}^{I} x_i > b \right\},$$

where $b \geq 0$ is a scalar.

2.2 Importance Sampling

As in general no analytical expression for P is known, a natural approach to obtain an estimate of P is to perform simulation experiments. Let $\Omega = \{f_i, i = 1, 2, \dots\}$ be the set of all paths f in the evolution of the system, given that the system is in state n_0 at time $t = 0$, i.e., $f(0) = n_0$. Let 1_x be an indicator of the event x, and $p(f)$ the probability density function of the sample path f. Then we obtain that

$$P = \int_\Omega 1_{f(T)\in A}\, p(f)\mathrm{d}f = \mathbb{E}_p\left(1_{f(T)\in A}\right), \tag{2}$$

where the subscript p indicates sampling from the density p. An unbiased estimate of (2) can be obtained by performing MC simulation, i.e., we run R independent simulations, with the system starting in state n_0, and we determine

$$P_{\mathrm{MC}} := \frac{1}{R}\sum_{i=1}^{R} 1_{f_i(T)\in A},$$

where f_i is the path obtained in the i-th run. In case n_0 and A are such that $f(T) \in A$ occurs relatively often, then we can accurately estimate P in a relatively small amount of time by P_{MC}. The number of runs needed to obtain an estimate with predefined accuracy and confidence, is in general inversely proportional to the probability to be estimated, see e.g. [10].

If n_0 and A are such that $f(T) \in A$ is a rare event, then the above properties entail that we need a large number of simulations to provide an accurate statistical estimate of P. In this case the simulation can be accelerated by using IS. The idea underlying IS is to simulate the system with a new set of input probability distributions, such that the rare event becomes more likely. To this end, let us consider a new probability measure p'. Then, (2) is equivalent to

$$\begin{aligned} P &= \int_\Omega 1_{f(T)\in A}\frac{p(f)}{p'(f)}p'(f)\mathrm{d}f \\ &= \int_\Omega 1_{f(T)\in A}L(f)p'(f)\mathrm{d}f \\ &= \mathbb{E}_{p'}\left(1_{f(T)\in A}L(f)\right), \end{aligned} \tag{3}$$

where $L(f) := p(f)/p'(f)$ is called the *likelihood ratio*. Note that (3) is valid for any density $p'(\cdot)$, given that $p'(f) > 0$ for all f that are such that $f(T) \in A$. Hence, an unbiased IS estimator is given by

$$P_{\mathrm{IS}} := \frac{1}{R}\sum_{i=1}^{R} 1_{f_i(T)\in A}L(f_i),$$

where f_i is now simulated under the measure p', with $f_i(0) = n_0$, $i = 1, \dots, R$.

Clearly, the simulation can be accelerated considerably if p' is properly chosen, in the sense that the number of runs needed to obtain an accurate statistical estimate of P with IS, is less compared to the number of runs needed with MC simulation. Hence, IS can be seen as a variance-reduction technique. We note, however, that not every choice of p' will reduce the variance. In fact, if p' is badly chosen, then this may increase the variance, or even make it infinite.

In this paper we assume that n_0 and A are such that $f(T) \in A$ is a rare event. As mentioned above, in this case MC simulation is unattractive, and one may resort to IS to obtain an estimate of P. We derive an IS scheme that considerably speeds up the simulation. This scheme is based on sample-path large deviations results, see e.g. Shwartz and Weiss [16].

2.3 Large deviations

In this subsection we present large deviations results of Shwartz and Weiss [16], which will be needed in the next sections.

Let $X(t)$ be a Markovian jump process with state space $\mathbb{R}^d$, equipped with transition rates:

$$q(x, x + v_i) = \psi_i(x),$$

where v_i is a vector in $\mathbb{R}^d$ and $\psi_i(x)$ is the rate of the jump in that direction when the state is x, $i = 1, \ldots, k$. Also, let $\overline{X}^n(t) := X(nt)/n$, $t \geq 0$, $n \geq 1$, be the fluid scaled process, which is obtained by making the jumps smaller, but faster. Define the 'local' rate function

$$\ell(x, y) := \sup_{\theta} \left(\langle \theta, y \rangle - \sum_{i=1}^{k} \psi_i(x) \left(e^{\langle \theta, v_i \rangle} - 1 \right) \right),$$

where x, y and θ are in $\mathbb{R}^d$, and $\langle \cdot, \cdot \rangle$ denotes the usual inproduct: $\langle a, b \rangle := \sum_{i=1}^{d} a_i b_i$. Finally, define the rate function

$$I_T(f) := \begin{cases} \int_0^T \ell(f(s), f'(s)) \mathrm{d}s & \text{if } f \text{ is absolutely continuous;} \\ \infty & \text{otherwise,} \end{cases}$$

where f is in $\mathbb{R}^d$. The following sample-path large deviations principle (LDP) now holds (see Theorem 5.1 in [16]).

Theorem 2.1 *For any well-defined x_0 and set F,*

$$-\lim_{n \to \infty} \frac{1}{n} \log \mathbb{P}\left(\overline{X}^n(\cdot) \in F | \overline{X}^n(0) = x_0 \right) = \inf_{f \in F,\ f(0) = x_0} I_T(f).$$

Remark: Intentionally, Theorem 2.1 has been formulated slightly imprecise. In fact, the LDP consists of an upper and lower bound, which apply to closed and open sets, respectively, see Theorem 5.1 in [16]. However, for the purpose of the paper, it is sufficient to state the theorem as above. For more details we refer to Chapter 5 of [16].

Let us write $g(x) \sim h(x)$ when $g(x)/h(x) \to 1$ if $x \to \infty$. Then it follows from the above that

$$\mathbb{P}\left(\overline{X}^n(\cdot) \in F | \overline{X}^n(0) = x_0 \right) \approx g(n, F, x_0) e^{-n I_T(f^*)}, \quad n \to \infty,$$

where f^* is the optimizing path in Theorem 2.1, and $g(n, F, x_0)$ is a subexponential function, i.e.,

$$\lim_{n \to \infty} \frac{\log g(n, F, x_0)}{n} = 0.$$

From the above it follows that Theorem 2.1 only gives us the *logarithmic asymptotics*. Therefore, in general Theorem 2.1 does not provide us with any information on the function $g(n, F, x_0)$, which implies that we can only use it to obtain a rough estimate of $\mathbb{P}\left(\overline{X}^n(\cdot) \in F | \overline{X}^n(0) = x_0\right)$.

In the next section we apply Theorem 2.1 to the so-called free M/M/1-PS process.

3 Free M/M/1-PS process

We first assume that $X(t)$ corresponds to the free M/M/1-PS process, i.e., the M/M/1-PS queue that is not reflected at 0, meaning that the state space of $X(t)$ is $\mathbb{Z}$ (whereas the state space of a M/M/1-PS queue is $\mathbb{N}_0$). We note that the flow-level dynamics of the M/M/1-PS queue coincide with those of the M/M/1-FIFO queue, which also follows from the well-known property that both have the same steady-state queue length distribution. This implies that the results derived in this section in fact also hold for the free M/M/1-FIFO process.

In this section we treat the free M/M/1-PS process, because this plays a key role in the analysis of a general alpha-fair sharing network, as we will see in Section 4. This may sound surprising, as the down rates corresponding to free M/M/1-PS process are constant, whereas the down rates corresponding to a general alpha-sharing network are variable. The idea underlying this analysis is that we can locally approximate the flow-level dynamics of a particular class in a general alpha-fair sharing network by a free M/M/1-PS process with class-specific arrival and service rate, which will be exploited in the next sections to obtain an estimate of P.

Since $X(t)$ corresponds to the free M/M/1-PS process, we have that $X(t) = X_{\text{up}}(t) - X_{\text{down}}(t)$, where $X_{\text{up}}(t)$ is a Poisson process of rate λ and $X_{\text{down}}(t)$ is an independent Poisson process of rate μ. Assume that $\lambda < \mu$, such that $X(t)$ has a negative drift. The transition structure of $X(t)$ is then, in the terminology of Section 2.3,

$$\begin{aligned} v_1 &= +1; \quad \psi_1(x) = \lambda; \\ v_2 &= -1; \quad \psi_2(x) = \mu, \end{aligned}$$

with $x \in \mathbb{Z}$. Then,

$$\ell(x, y) = \ell(y) = \sup_\theta \left\{ \theta y - \lambda \left(e^\theta - 1 \right) - \mu \left(e^{-\theta} - 1 \right) \right\},$$

i.e., the local rate function is independent of the current state x. Straightforward calculus shows that the optimizer satisfies

$$e^{\theta^*} = \frac{y + \sqrt{y^2 + 4\lambda\mu}}{2\mu},$$

which yields

$$\begin{aligned} \ell(y) &= y \log \left(\frac{y + \sqrt{y^2 + 4\lambda\mu}}{2\lambda} \right) + \lambda + \mu - \sqrt{y^2 + 4\lambda\mu} \\ &=: \ell(y|\lambda, \mu). \end{aligned}$$

We now focus on the overflow probability

$$\mathbb{P}(\overline{X}^n(T) > k | \overline{X}^n(0) = k_0),$$

with $k > k_0$. Using Theorem 2.1, we have that

$$\mathbb{P}(\overline{X}^n(T) > k | \overline{X}^n(0) = k_0) \approx e^{-nI^*},$$

where

$$I^* := \inf_{f \in G,\ f(0)=k_0} I_T(f), \text{ with } G := \{f : f(T) > k\}.$$

In Lemma 5.16 of [16] it is shown that the MPP, i.e., the path f^* in set G that minimizes I_T, is a straight line from k_0 to k in the interval $[0, T]$, with cost

$$\begin{aligned} I^* &= I_T(f^*) = T \times \ell\left(\frac{k-k_0}{T}\middle|\lambda,\mu\right) \\ &= T\left(\frac{k-k_0}{T}\log\left(\frac{k-k_0}{2T\lambda} + \frac{1}{2\lambda}\sqrt{\frac{(k-k_0)^2}{T^2} + 4\lambda\mu}\right)\right. \\ &\quad \left. +\lambda + \mu - \sqrt{\frac{(k-k_0)^2}{T^2} + 4\lambda\mu}\right) \\ &=: \mathbb{C}(k-k_0, T|\lambda,\mu). \end{aligned} \tag{4}$$

We now show that (4) can also be obtained in another way. First recall that the cost of a Poisson process of rate λ behaving like a Poisson process of rate λ^* is, during one unit of time,

$$\tilde{I}(\lambda^*|\lambda) := \lambda^* \log\left(\frac{\lambda^*}{\lambda}\right) + \lambda - \lambda^*,$$

see p. 20 of [16]. Note that $\tilde{I}(\lambda^*|\lambda)$ is the Legendre transform of the logarithmic Moment Generating Function (MGF) of a random variable that has a Poisson distribution with mean λ. Clearly, $\tilde{I}(\mu^*|\mu)$ follows in the same way. Observe that indeed $\tilde{I}(p^*|p) = 0$, $p = \lambda, \mu$, as required. In order to have $X(T) > k$, given that $X(0) = k_0$, we should have that X_{up} (X_{down}) behaves as a different Poisson process of rate λ^* (μ^*), where $(\lambda^* - \mu^*)T > k - k_0$. We thus get the minimization problem:

$$T \min_{\lambda^*,\mu^*} \left\{\tilde{I}(\lambda^*|\lambda) + \tilde{I}(\mu^*|\mu)\right\},$$

over all λ^*, μ^* such that $(\lambda^* - \mu^*)T > k - k_0$. Straightforward calculations yield that the optimizers are

$$\begin{aligned} \lambda^* &= \frac{k-k_0}{2T} + \frac{1}{2}\sqrt{\frac{(k-k_0)^2}{T^2} + 4\lambda\mu}; \\ \mu^* &= -\frac{k-k_0}{2T} + \frac{1}{2}\sqrt{\frac{(k-k_0)^2}{T^2} + 4\lambda\mu}, \end{aligned} \tag{5}$$

and the corresponding objective function value indeed equals (4).

4 Most probable path

In the previous section we obtained an approximation for the overflow probability in the M/M/1-PS queue (where we assumed that there was no reflection at 0). In this section we use the same ideas to derive an approximation for P in a general alpha-fair sharing network.

We first consider the cost $\mathbb{K}(f,T)$ of a path f, with $f(0)=n_0$, in the interval $[0,T]$. We find that

$$\mathbb{K}(f,T)=\sum_{i=1}^{I}\int_0^T \ell\left(f_i'(t)|\lambda_i,\nu_i(f(t))\right)\mathrm{d}t.$$

From the logarithmic asymptotics stated in Theorem 2.1 it then follows that the following approximation applies:

$$\begin{aligned} P &= \mathbb{P}(N(T)\in A|N(0)=n_0) \\ &\approx \exp\left(-\inf_{f:f(T)\in A,f(0)=n_0}\mathbb{K}(f,T)\right). \end{aligned} \tag{6}$$

Let f^* denote the path that minimizes the cost, i.e., the MPP. Since the down rates in our model are state-dependent, in contrast to what is the case for the free M/M/1-PS process, the MPP in general has a non-linear shape. In fact, in general no closed-form expression is available for the path that minimizes $\mathbb{K}(f,T)$. Equation (6) suggests that we should try to find an accurate approximation of f^* to obtain an estimate of P, which is done below.

Divide T into n (which is typically a large number) subintervals of length $\Delta_n := T/n$. Consider the contribution to a path of the k-th subinterval, i.e., the interval $[k\Delta_n,(k+1)\Delta_n)$, for $k=0,\ldots,n-1$, and assume that the down rates are $\nu_i(f(k\triangle_n))$, $i=1,\ldots,I$, in this subinterval. Then the cost of this time interval, related to class i are given by

$$\mathbb{C}(f_i((k+1)\Delta_n)-f_i(k\Delta_n),\Delta_n|\lambda_i,\nu_i(f(k\Delta_n)).$$

Hence, we find that the total cost $\mathbb{K}_n(f,T)$ are

$$\sum_{i=1}^{I}\sum_{k=0}^{n-1}\mathbb{C}(f_i((k+1)\Delta_n)-f_i(k\Delta_n),\Delta_n|\lambda_i,\nu_i(f(k\Delta_n))).$$

Note that the higher the value of n, the more accurate the approximation will be, i.e.,

$$\lim_{n\to\infty}\mathbb{K}_n(f,T)=\mathbb{K}(f,T).$$

Using the above, we can approximate $\mathbb{K}(f,T)$, for given $n\in\mathbb{N}$, by $\mathbb{K}_n(f,T)$. Also, the path that minimizes $\mathbb{K}_n(f,T)$ can be regarded as an approximation of f^*. In order to obtain this approximating path, optimization should be performed over all $f_i(j\Delta_n)$, $i=1,\ldots,I$, $j=0,\ldots,n$, i.e., $(n+1)I$ entries, given that $f(0)=n_0$ and $f(n\Delta_n)=f(T)\in A$.

Approximation (6) turns out not to be very accurate in general. Clearly, this is no surprise, as in Section 2.3 we already argued that Theorem 2.1 just gives us the logarithmic asymptotics, and that we therefore have only a rough estimate of P.

5 New input distributions

In the previous section we derived an approximation for P that required the calculation of an optimizing path. This path can be regarded as an approximation for the most likely way for the event to happen. That is, given that the event occurs, with overwhelming probability $N(T) \in A$ is reached by a path close to this optimizing path. In this section we show how we can exploit the results of Section 4 to develop methodology to obtain an accurate estimate of P.

Assume that we have (an accurate approximation of) the MPP

$$f^* := \arg \inf_{f: f(T) \in A, f(0)=n_0} \mathbb{K}(f, T),$$

as discussed in the previous section. Suggested by (5), the following change-of-measure at time t corresponds to f^*:

$$\lambda_i^*(t) := \frac{1}{2}(f_i^*)'(t) + \frac{1}{2}\sqrt{((f_i^*)'(t))^2 + 4\lambda_i\nu_i(f^*(t))};$$

$$\nu_i^*(t) := -\frac{1}{2}(f_i^*)'(t) + \frac{1}{2}\sqrt{((f_i^*)'(t))^2 + 4\lambda_i\nu_i(f^*(t))},$$

$i = 1, \ldots, I$. When, at time $t \geq 0$, the process is simulated with arrival rates $\lambda^*(t)$ and departure rates $\nu^*(t)$, given that the process starts at n_0 at $t = 0$, it is not hard to see that the i-th coordinate of the expected position of the process at time t is

$$\begin{aligned} n_{0,i} + \int_0^t \lambda_i^*(s)\mathrm{d}s - \int_0^t \nu_i^*(s)\mathrm{d}s &= f_i^*(0) + \int_0^t (f_i^*)'(s)\mathrm{d}s \\ &= f_i^*(t), \end{aligned}$$

$i = 1, \ldots, I$, i.e., the process has the 'correct' expected position, under this change-of-measure.

In an Appendix we present an IS scheme that can be used to obtain an estimate of P. The basic idea underlying this scheme is to simulate the model with rates $\lambda_i^*(t)$ and $\nu_i^*(t)$, $i = 1, \ldots, I$. Typically, we only know these rates at $n+1$ time points, as in general the MPP is not explicitly known, but it is approximated, see Section 4. However, if one assumes the rates to be constant between two consecutive time points, i.e., in a subinterval, then each class essentially behaves as a free M/M/1-PS process with class-specific arrival and service rate in this subinterval, which is easy to simulate. For more details we refer to the Appendix.

In the next section we show that, compared to MC simulation, this scheme can considerably speed up the simulation, given that the underlying event is rare. That is, the number of runs that are needed to get some fixed level of confidence with the IS scheme, is substantially less than the number of runs needed with MC simulation.

6 Simulation results

In this section the performance of the IS algorithm is examined in case of a single-node network (shared by multiple traffic classes) and a linear network, respectively. These are the two simplest networks, and therefore of particular interest to gain insight. We have performed extensive simulation experiments for each of these two networks, and the results are presented below. We mention that, besides the results reported in this section, we have considered many other examples, in which usually a substantial speed-up is achieved

6.1 Single-node network

We first consider a single-node network with capacity c, where capacity is shared between I classes. In order to obtain the alpha-fair allocation we have to solve the following optimization problem for state $n \neq 0$:

$$\begin{array}{rl} \max & \sum_{i=1}^{I} U_i(x_i) \\ \text{subject to} & \sum_{i=1}^{I} n_i x_i \leq c \\ \text{over} & x_i \geq 0, \quad i = 1, \ldots, I, \end{array}$$

where $U_i(x_i)$ is defined as before. It is a straightforward exercise to show that the optimizers are such that

$$s_i(n) = n_i x_i^* = \frac{\kappa_i^{1/\alpha} n_i c}{\sum_{j=1}^{I} \kappa_j^{1/\alpha} n_j}, \quad i = 1, \ldots, I. \tag{7}$$

From (7) it follows that alpha-fair sharing in a single-node network corresponds to sharing in a discriminatory-processor-sharing fashion, with relative weights $\kappa_i^{1/\alpha}$, $i = 1, \ldots, I$, see [8]. We find [2] that

$$d_i(n) = \min \left\{ \frac{\kappa_i^{1/\alpha} n_i c}{\sum_{j=1}^{I} \kappa_j^{1/\alpha} n_j}, n_i r_i \right\}, \quad i = 1, \ldots, I.$$

The steady-state distribution of $N(t)$ is only known in case $\kappa_i = \kappa$ and $r_i \geq c$, $i = 1, \ldots, I$, i.e., if all weights are equal and if the rate limitation for each flow is at least as large as the capacity of the node, so that there is essentially no rate limitation.

Then this model is equivalent to a processor-sharing model, and it is straightforward to derive that the steady-state distribution of $N(t)$ is [3]

$$\pi(n) = \frac{c - \sum_{i=1}^{I} \rho_i}{c} \frac{(n_1 + \cdots + n_I)!}{n_1! \ldots n_I!} \prod_{i=1}^{I} \left(\frac{\rho_i}{c} \right)^{n_i}, \quad n \in \mathbb{N}_0^I,$$

given that the stability condition $\sum_{i=1}^{I} \rho_i < c$ holds.

The first part of the IS algorithm consists of finding a MPP. We have performed numerical experiments to gain insight on the typical shape of such a minimizing path. We consider the setting with $I = 2$, $\lambda_1 = 0.75$, $\lambda_2 = 1.5$, $\mu_1 = 2$, $\mu_2 = 4$, $\kappa_1^{1/\alpha} = 1/3$, $\kappa_2^{1/\alpha} = 2/3$, $r_1 = 0.9$, $r_2 = 0.8$, and $c = 1$, and we let T, n_0 and set A vary. The results are depicted in Figure 1, which are obtained by using an optimization procedure in Mathematica 5.2. We solved the problem for $n = 2^p$, $p = 1, \ldots, 5$, and we used the minimizing path found for $n = 2^{q-1}$ as starting path in the optimization procedure for $n = 2^q$, $q = 2, \ldots, 5$ (for $n = 2$ we do not have a nice starting path). Hence, the depicted paths are associated with $n = 2^5 = 32$. We note that the above approach is much faster than solving the optimizing problem directly for $n = 32$ (without an appropriate starting path). We observed that the optimizing problem can be solved in a relatively small amount of time if $n \leq 32$. For higher values of n the obtained path is almost similar to the one obtained for $n = 32$, but the computation requires more time. In the first, second and third column of Figure 1 we depict $(f_1(i\Delta_{32}), i\Delta_{32})$, $(f_2(i\Delta_{32}), i\Delta_{32})$ and

T	a	$n_{0,1}$	$n_{0,2}$	$P_{\rm IS}$	$\#_{\rm IS}$	$\tau_{\rm IS}$	$P_{\rm MC}$	$\#_{\rm MC}$	$\tau_{\rm MC}$
1	6	1	4	$6.1 \cdot 10^{-4}$	1559	12.4	$6.2 \cdot 10^{-4}$	622085	341.1
1	6	1	0	$3.0 \cdot 10^{-4}$	3312	16.4	$2.7 \cdot 10^{-4}$	1468866	523.4
3	6	1	4	$1.5 \cdot 10^{-2}$	2441	17.2	$1.4 \cdot 10^{-2}$	27518	35.9
3	6	1	0	$5.1 \cdot 10^{-3}$	22745	91.4	$4.9 \cdot 10^{-3}$	78851	77.7
6	6	1	4	$4.0 \cdot 10^{-2}$	12821	80.4	$3.7 \cdot 10^{-2}$	9397	23.4
6	6	1	0	$2.0 \cdot 10^{-2}$	74237	390.1	$1.8 \cdot 10^{-2}$	22397	44.9

Table 1: Simulation results for structure (i): comparison with MC simulation (times in seconds).

$(f_1(i\Delta_{32}), f_2(i\Delta_{32}))$, $i = 0, \ldots, 32$, respectively. We note that we have, besides the ones depicted in Figure 1, considered many other scenarios. Also in these cases, the minimizing paths do not seem to be linear.

Although the shapes of the MPPs corresponding to scenarios (a)-(c) are not always trivial, the shape of the path corresponding to scenario (d) perhaps requires some more explanation. In particular, the shape of the path corresponding to class 1 is surprising in this scenario: it first slightly decreases, and then it starts to increase. A possible explanation for this phenomenon may be the following. In [2] it was shown that there exists a unique point $n^* = (n_1^*, n_2^*)$ such that $\lambda_i = d_i(n^*)$, $i = 1, 2$. This is the equilibrium point of the so-called fluid limit: the system operates (most likely) most of the time around this point. The fluid limit is obtained by both speeding up the arrivals and service speed by a fixed factor, and then letting this factor go to infinity. It can be shown that the resulting normalized Markov process converges to a deterministic limit. From Proposition 2.1 in [2] it follows that $n_1^* = 0.5625$ and $n_2^* = 0.46875$ in scenarios (a)-(d). Recalling that the path starts in $n_0 = (3, 0)$ in scenario (d), we see that the MPP initially evolves in the direction of the fluid limit, but then changes its direction to make sure that $f_2(T) > 6$. It remains, however, hard to fully explain the shapes of the MPPs in general. One can expect that the MPP from any n_0 to any set A is more or less linear if T is relatively small. In contrast, if T is relatively large, then one can expect that the MPP first drifts to n^*, and then changes its direction towards set A, see e.g. [12].

To quantify the performance of the proposed IS scheme we take the same parameter values as above, where we let T, n_0 and set A vary. We consider three structures for A: (i) $\{f | f_1(T) > a\}$, (ii) $\{f | f_2(T) > a\}$ and (iii) $\{f | f_1(T) + f_2(T) > a\}$, with $a > 0$. The results are presented in Tables 1-4. These results (and also the ones in the next subsection) are obtained with Mathematica 5.2 and are tested on a personal computer with an AMD Athlon 64 3500+ processor (2.2 GHz). In the tables $\#_{\rm IS}$ ($\#_{\rm MC}$) denotes the number of runs needed with IS (MC) simulation to obtain a confidence of 95% and a relative efficiency (i.e., the ratio of the confidence interval half-length to the estimated value) of 10%, and $\tau_{\rm IS}$ ($\tau_{\rm MC}$) denotes the time needed with IS (MC simulation). Note that $\tau_{\rm IS}$ consists of two parts: (a) finding the optimal path and (b) performing the simulation with the new input distributions.

Table 1 compares IS with MC simulation. The MC estimator is obtained by simulating independent runs of the original model (starting in n_0) until time T, and subsequently determining the fraction of the runs that are such that $f(T) \in A$. The table shows that for a relatively large value of P (larger than 0.01), MC simulation yields an accurate estimate much faster than the IS scheme does. In contrast, for a relatively small value of P (smaller than 0.01), IS signifi-

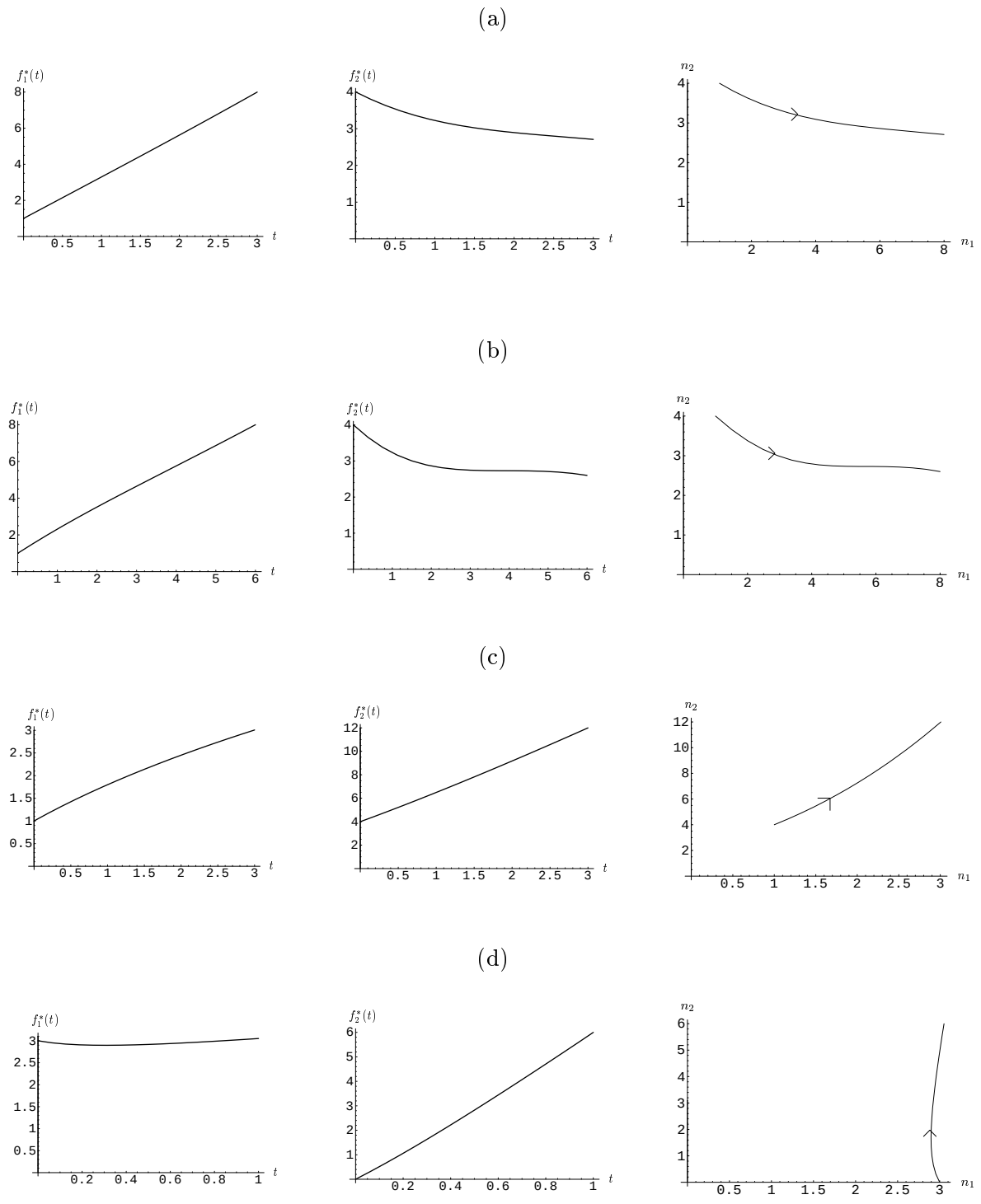

Figure 1: Single-node network: The minimizing paths of $\mathbb{K}_{32}(f, T)$ in four scenarios. Scenario (a): $T = 3$, $A = \{f | f_1(n) > 8\}$ and $n_0 = (1, 4)$. Scenario (b): $T = 6$, $A = \{f | f_1(T) > 8\}$ and $n_0 = (1, 4)$. Scenario (c): $T = 3$, $A = \{f | f_2(T) > 12\}$ and $n_0 = (1, 4)$. Scenario (d): $T = 1$, $A = \{f | f_2(T) > 6\}$ and $n_0 = (3, 0)$. The left panel shows $f_1^*(\cdot)$ as function of time t. The middle panel shows $f_2^*(\cdot)$ as function of time t. The right panel shows the parametric plot of $(f_1^*(\cdot), f_2^*(\cdot))$. Since we only know both $f_1^*(\cdot)$ and $f_2^*(\cdot)$ at $n + 1 = 33$ time points, we linearly interpolate between consecutive points.

T	a	$n_{0,1}$	$n_{0,2}$	P_{IS}	$\#_{\text{IS}}$	τ_{IS}
1	10	1	0	$3.3 \cdot 10^{-8}$	3509	18.9
1	14	1	0	$5.1 \cdot 10^{-13}$	4744	24.4
2	10	1	0	$2.9 \cdot 10^{-6}$	6762	33.9
2	14	1	0	$7.3 \cdot 10^{-10}$	8123	40.7
3	10	1	0	$2.6 \cdot 10^{-5}$	12439	59.2
3	14	1	0	$2.5 \cdot 10^{-8}$	12526	63.7
1	10	1	4	$4.2 \cdot 10^{-8}$	2861	17.5
1	14	1	4	$7.8 \cdot 10^{-13}$	3349	20.6
2	10	1	4	$6.0 \cdot 10^{-6}$	2680	19.5
2	14	1	4	$1.2 \cdot 10^{-9}$	4686	28.5
3	10	1	4	$6.2 \cdot 10^{-5}$	4173	25.8
3	14	1	4	$5.6 \cdot 10^{-8}$	5847	37.4

Table 2: Simulation results for structure (i): rare events (times in seconds).

T	a	$n_{0,1}$	$n_{0,2}$	P_{IS}	$\#_{\text{IS}}$	τ_{IS}
1	10	1	0	$5.4 \cdot 10^{-7}$	2702	15.3
1	14	1	0	$1.0 \cdot 10^{-10}$	3624	21.5
2	10	1	0	$3.1 \cdot 10^{-5}$	8958	40.4
2	14	1	0	$8.4 \cdot 10^{-8}$	4756	28.6
3	10	1	0	$1.6 \cdot 10^{-4}$	10423	51.0
3	14	1	0	$1.7 \cdot 10^{-6}$	18663	94.6
1	16	1	4	$5.5 \cdot 10^{-9}$	2776	16.2
1	20	1	4	$7.2 \cdot 10^{-13}$	3580	21.1
2	16	1	4	$1.1 \cdot 10^{-6}$	2613	17.6
2	20	1	4	$1.7 \cdot 10^{-9}$	3792	24.4
3	16	1	4	$8.0 \cdot 10^{-6}$	3517	24.6
3	20	1	4	$6.2 \cdot 10^{-8}$	4152	25.9

Table 3: Simulation results for structure (ii): rare events (times in seconds).

cantly outperforms MC simulation. Clearly, this is no surprise: the IS scheme presented in the Appendix is based on large deviations results, and therefore one expects this scheme to perform well in case the underlying event is rare, i.e., if P is relatively small.

Tables 2, 3 and 4 show the performance of our scheme in case of rare events. As mentioned in Section 2.2, in this case MC simulation is infeasible. Therefore, we have decided not to compare the performance of the IS scheme with that of the MC simulation. These tables show that our scheme works remarkably well for rare events: we are able to estimate probabilities up to 10^{-13} in a fast way.

The results also show that the performance of the IS scheme decreases as T increases (for fixed other model parameters), i.e., more runs are needed to achieve the required efficiency. This can be explained as follows. As T increases and n (the number of subintervals) remains constant, the approximation of the minimizing path becomes less accurate, and therefore the performance of the IS algorithm is also negatively affected.

T	a	$n_{0,1}$	$n_{0,2}$	P_{IS}	$\#_{\text{IS}}$	τ_{IS}
1	20	1	0	$2.3 \cdot 10^{-13}$	4396	67.6
2	25	1	0	$5.1 \cdot 10^{-13}$	6605	80.8
3	30	1	0	$1.2 \cdot 10^{-13}$	12017	107.1
1	20	1	4	$9.6 \cdot 10^{-10}$	3281	52.1
2	25	1	4	$3.5 \cdot 10^{-10}$	5156	60.4
3	30	1	4	$4.0 \cdot 10^{-11}$	8483	86.2

Table 4: Simulation results for structure (*iii*): rare events (times in seconds).

We also empirically observed that, for fixed arbitrarily chosen n_0 and T,

$$\lim_{k\to\infty} \frac{\log \mathbb{E}_{p'}\left(1_{f(T)\in A\cdot k} L^2(f)\right)}{\log \mathbb{E}_{p'}\left(1_{f(T)\in A\cdot k} L(f)\right)}, \tag{8}$$

is close to (but smaller than) 2, where p' is the IS-distribution and $A \cdot k := \{n : n/k \in A\}$. It is noted that one can estimate both denominator and numerator in (8) by using the simulation output. The above suggests that our IS scheme is nearly asymptotically optimal [6], which we can, however, not formally prove.

6.2 Linear network

We next consider a linear network that consists of L nodes, where node i has capacity c_i. There are $I = L + 1$ classes of users: each class corresponds to a specific route in the network. Class-i users require service at node i only, $i = 1, \ldots, L$, whereas class-$(L + 1)$ users require service at all L nodes simultaneously. In order to obtain the alpha-fair allocation we have to solve the following optimization problem for state $n \neq 0$:

$$\begin{aligned} \max \quad & \textstyle\sum_{i=1}^{I} U_i(x_i) \\ \text{subject to} \quad & n_i x_i + n_{L+1} x_{L+1} \leq c_i \\ \text{over} \quad & x_i \geq 0, \quad i = 1, \ldots, L+1. \end{aligned}$$

Only in case $c_i = c$, $i = 1, \ldots, L$, i.e., if all nodes have the same capacity, there exist explicit expressions for the optimizing x_i^*s. In that case the optimizers are such that

$$s_{L+1}(n) = n_{L+1} x_{L+1}^* = \frac{(\kappa_{L+1} n_{L+1}^\alpha)^{1/\alpha}}{(\kappa_{L+1} n_{L+1}^\alpha)^{1/\alpha} + (\sum_{j=1}^{L} \kappa_j n_j^\alpha)^{1/\alpha}};$$

$$s_i(n) = n_i x_i^* = (1 - s_{L+1}(n)) 1_{n_i > 0}, \quad i = 1, \ldots, L.$$

Therefore, we find that $d_{L+1}(n)$ equals

$$\min\left\{ \frac{(\kappa_{L+1} n_{L+1}^\alpha)^{1/\alpha}}{(\kappa_{L+1} n_{L+1}^\alpha)^{1/\alpha} + (\sum_{j=1}^{L} \kappa_j n_j^\alpha)^{1/\alpha}}, n_{L+1} r_{L+1} \right\},$$

and $d_i(n)$, for $i = 1, \ldots, L$, equals

$$\min\left\{ \left(\frac{(\sum_{j=1}^{L} \kappa_j n_j^\alpha)^{1/\alpha}}{(\kappa_{L+1} n_{L+1}^\alpha)^{1/\alpha} + (\sum_{j=1}^{L} \kappa_j n_j^\alpha)^{1/\alpha}} \right) 1_{n_i > 0}, n_i r_i \right\}.$$

T	a_1	a_2	P_{IS}	$\#_{\text{IS}}$	τ_{IS}	P_{MC}	$\#_{\text{MC}}$	τ_{MC}
1	6	6	$6.1 \cdot 10^{-2}$	2308	126.9	$6.5 \cdot 10^{-2}$	5527	8.8
1	8	8	$7.8 \cdot 10^{-3}$	2102	140.3	$7.7 \cdot 10^{-3}$	49912	78.5
2	8	8	$4.6 \cdot 10^{-2}$	5045	145.2	$4.6 \cdot 10^{-2}$	7908	23.4
2	10	10	$8.5 \cdot 10^{-3}$	4573	172.5	$8.9 \cdot 10^{-3}$	40859	121.3
3	10	10	$2.8 \cdot 10^{-2}$	9722	274.2	$3.0 \cdot 10^{-2}$	12010	52.6
3	12	12	$6.7 \cdot 10^{-3}$	19131	306.3	$6.8 \cdot 10^{-3}$	53159	232.3

Table 5: Simulation results for the linear network: comparison with MC simulation (times in seconds).

T	a_1	a_2	P_{IS}	$\#_{\text{IS}}$	τ_{IS}
1	15	15	$1.6 \cdot 10^{-7}$	6481	191.3
1	20	20	$7.0 \cdot 10^{-12}$	10782	199.7
2	20	20	$3.7 \cdot 10^{-8}$	10994	175.1
2	25	25	$1.1 \cdot 10^{-11}$	19255	272.7
3	25	25	$3.0 \cdot 10^{-9}$	19326	312.0
3	30	30	$2.0 \cdot 10^{-12}$	48310	631.5

Table 6: Simulation results for the linear network: rare events (times in seconds).

The steady-state distribution of $N(t)$ is only tractable if $\alpha = 1$, $\kappa_i = \kappa$, $c_j = c$, and $r_i \geq c$, $i = 1, \ldots, L+1$, $j = 1, \ldots, L$. Under the stability condition $\max_{1 \leq i \leq L} \rho_i + \rho_{L+1} < c$, the steady-state distribution of $N(t)$ is such that $\pi(n)$ equals [13]

$$\frac{\prod_{i=1}^{L}(c - \rho_i - \rho_{L+1})}{c(c - \rho_{L+1})^{L-1}} \frac{(n_1 + \cdots + n_{L+1})!}{(n_1 + \cdots + n_L)! n_{L+1}!} \prod_{i=1}^{L+1} \left(\frac{\rho_i}{c}\right)^{n_i},$$

$n \in \mathbb{N}_0^{L+1}$.

We test the performance of our IS scheme in case $L = 2$, $\lambda_1 = 1$, $\lambda_2 = 1.75$, $\lambda_3 = 2$, $\mu_1 = 2$, $\mu_2 = 4$, $\mu_3 = 5$, $r_1 = 0.6$, $r_2 = 0.3$, $r_3 = 0.8$, $g_1 = 2$, $g_2 = 1$, $g_3 = 0.5$, $\alpha = 1$, and starting state $(1, 2, 1)$. Furthermore, we assume $c_i = c = 1$, $i = 1, 2$, such that we have a closed-form expression for $d_i(n)$, $i = 1, 2, 3$, and we let T and A vary. We assume that A has structure $\{f | f_1(T) + f_3(T) > a_1, f_2(T) + f_3(T) > a_2\}$, with $a_1, a_2 > 0$. The results are given in Tables 5-6.

The results again show that the rare event probabilities can be estimated rather efficiently. Compared to the single-node network, it now takes much more time to find the MPP (which in general has a non-linear shape), as one needs to optimize over more entries.

7 Conclusion

We analyzed a network where classes share capacity according to an alpha-fair sharing policy. We focused on the probability P that, given that the network is in some state n_0 at time 0, the network is in some set of states A at time T, i.e., $P = \mathbb{P}(N(T) \in A | N(0) = n_0)$. In particular, we assumed that n_0 and A are such that P is small, i.e., the underlying event is rare. As

no analytical expressions are known for P, we devised an IS scheme for quick and accurate estimation.

In this paper we studied the transient behavior, but a topic for further research is the derivation of an approximation of $\pi(A)$, where $\pi(\cdot)$ denotes the steady-state distribution of $N(t)$. Using regenerative arguments, one can obtain $\pi(A)$ by dividing the expected time that the process spends in set A during a cycle from n_0 to n_0, by the associated expected cycle time, see e.g. Corollary 1.4 in [1]. One may use specific measures to estimate both numerator and denominator, so-called *measure specific dynamic IS*, see e.g. [9]. Dynamic refers to the fact that per run the IS is turned on until the event of interest occurs and turned off thereafter.

Acknowledgments

This research has been funded by the Dutch BSIK/BRICKS (Basic Research in Informatics for Creating the Knowledge Society) project.

References

[1] S. Asmussen. *Applied probability and queues.* Springer-Verlag, New York, USA, 2003.

[2] U. Ayesta and M. Mandjes. Bandwidth-sharing networks under a diffusion scaling. *Accepted for publication in Annals of Operations Research*, 2007.

[3] F. Baskett, K.M. Chandy, R.R. Muntz, and F. Palacios-Gomez. Open, closed and mixed networks of queues with different classes of customers. *Journal of the ACM*, 22:248–260, 1975.

[4] J. Blanchet, P.W. Glynn, and J.C. Liu. State-dependent importance sampling and large deviations. In: *Proceedings of ValueTools 2006*, Pisa, Italy, 2006.

[5] T. Bonald and L. Massoulié. Impact of fairness on Internet performance. In: *Proceedings of ACM SIGMETRICS 2001*, pages 82–91, Boston MA, USA, 2001.

[6] J. Bucklew. *Large deviation techniques in decision, simulation and estimation.* Wiley, New York, USA, 1990.

[7] P. Dupuis and H. Wang. Dynamic importance sampling for uniformly recurrent Markov chains. *Annals of Applied Probability*, 15:1–38, 2005.

[8] G. Fayolle, I. Mitrani, and R. Iasnogorodski. Sharing a processor among many classes. *Journal of the ACM*, 27:519–532, 1980.

[9] A. Goyal, P. Shahabuddin, P. Heidelberger, V. Nicola, and P.W. Glynn. A unified framework for simulating Markovian models of highly dependable systems. *IEEE Transactions on Computers*, 41:36–51, 1992.

[10] P. Heidelberger. Fast simulation of rare events in queueing and reliability models. *ACM Transactions on Modeling and Computer Simulation*, 5:43–85, 1995.

[11] L. Leskelä. Stabilization of an overloaded queueing network using measurement-based admission control. *Journal of Applied Probability*, 43:231–244, 2006.

[12] M. Mandjes. Rare event analysis of the state frequencies of a large number of Markov chains. *Stochastic Models*, 15:577–592, 1999.

[13] L. Massoulié and J. Roberts. Bandwidth sharing and admission control for elastic traffic. *Telecommunication Systems*, 15:185–201, 2000.

[14] J. Mo and J. Walrand. Fair end-to-end window-based congestion control. *IEEE/ACM Transactions on Networking*, 8:556–567, 2000.

[15] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose. Modeling TCP Reno performance: A simple model and its empirical validation. *IEEE/ACM Transactions on Networking*, 8:133–145, 2000.

[16] A. Shwartz and A. Weiss. *Large deviations for performance analysis*. Chapman & Hall, Londen, UK, 1995.

Appendix

Below we present the pseudocode of an IS scheme that can be used to estimate rare event probabilities.

IS Algorithm

Compute (or approximate) the minimizing path f^*.
Divide T into n subintervals of length $\Delta_n := T/n$.
FOR $j = 1$ TO R
  $\tilde{N}_i(0) \leftarrow n_{0,i}$, $i = 1, \ldots, I$.
  Set the likelihood ratio equal to 1: $L_j \leftarrow 1$.
  FOR $k = 1$ TO n
    $\tilde{N}_i(k\Delta_n) \leftarrow \tilde{N}_i((k-1)\Delta_n)$, $i = 1, \ldots, I$.
    Simulate Arrivals of type i as Poisson process of rate $\lambda_i^*(k\Delta_n)$.
    Simulate Departures of type i as Poisson process of rate $\nu_i^*(k\Delta_n)$.
    Thus K events are generated, with inter-event times $t_1, \ldots, t_K$.
    FOR $\ell = 1$ TO K
      IF Event(ℓ) = Arrival of type i
      THEN
        Update likelihood:
          $L_j \leftarrow L_j \times \exp((\lambda_i^*(k\Delta_n) - \lambda_i)t_\ell) \times (\lambda_i/\lambda_i^*(k\Delta_n))$.
        $\tilde{N}_i(k\Delta_n) \leftarrow \tilde{N}_i(k\Delta_n) + 1$.
      IF Event(ℓ) = Departure of type i AND $\tilde{N}_i(k\Delta_n) > 0$
      THEN
        Update likelihood:
          $L_j \leftarrow L_j \times \exp((\nu_i^*(k\Delta_n) - \nu_i(\tilde{N}(k\Delta_n)))t_\ell) \times (\nu_i(\tilde{N}(k\Delta_n))/\nu_i^*(k\Delta_n))$.
        $\tilde{N}_i(k\Delta_n) \leftarrow \tilde{N}_i(k\Delta_n) - 1$.
      IF Event(ℓ) = Departure of type i AND $\tilde{N}_i(k\Delta_n) = 0$
      THEN

```
            Set the likelihood ratio equal to 0:  L_j ← 0.
            Abort current simulation run and proceed with the next run.
        END
        Set t_K equal to 0 when K = 0.
        FOR i = 1 TO I
           Update likelihood:
              L_j ← L_j × exp((λ*_i(kΔ_n) − λ_i)(Δ_n − t_K)) × exp((ν*_i(kΔ_n) − ν_i(Ñ(kΔ_n)))(Δ_n − t_K))
        END
     END
     Put I_j ← 1 if N(nΔ_n) ∈ A, and 0 else.
  END
  Estimator P_IS ← R^{-1} · Σ_{j=1}^{R} L_j I_j.
```

Justification of the IS algorithm: We simulate the process $\tilde{N}(t) = (\tilde{N}_1(t), \ldots, \tilde{N}_I(t))$ during a time period of T units, given that $\tilde{N}(0) = n_0$, where

$$\tilde{N}_i(t) := \tilde{N}_{i,\mathrm{up}}(t) - \tilde{N}_{i,\mathrm{down}}(t), \quad i = 1, \ldots, I,$$

with $\tilde{N}_{i,\mathrm{up}}(t)$ being a Poisson process of rate $\lambda_i^*(k\Delta_n)$ and $\tilde{N}_{i,\mathrm{down}}(t)$ being a Poisson process of rate $\nu_i^*(k\Delta_n)$ if $t \in [(k-1)\Delta_n, k\Delta_n)$, $k = 1, \ldots, n$. Clearly, this corresponds to the process described in Section 2.1, but with different input distributions and with a different state space, as the state space of $\tilde{N}(t)$ is $\mathbb{Z}^I$, whereas that of $N(t)$ is $\mathbb{N}_0^I \subset \mathbb{Z}^I$. Since $\tilde{N}(t)$ can take any vector in $\mathbb{Z}^I$ (and thus in $\mathbb{N}_0^I$) with positive probability, it follows from Section 2.2 that we can obtain an unbiased IS estimator of P by simulating $\tilde{N}(t)$ and by keeping track of the likelihood ratio in each run.

We use that the interarrival times are exponentially distributed with mean $1/\lambda_i^*(k\Delta_n)$ $(1/\lambda_i)$ under the new (old) measure if $t \in [(k-1)\Delta_n, k\Delta_n)$. Also, we exploit that the service requirements are exponentially distributed with mean $1/\nu_i^*(k\Delta_n)$ $(1/\nu_i(\tilde{N}(t))$, with $\tilde{N}_i(t) > 0)$ under the new (old) measure, if $t \in [(k-1)\Delta_n, k\Delta_n)$. Clearly, if $\tilde{N}_i(t) = 0$ and a departure of class i occurs, then we reach a state that is infeasible in our model (that is, under the original probability measure), so that we set L equal to zero when this occurs. Since the likelihood ratio will stay zero once it has reached zero, one can abort the current simulation run. By simulating R independent runs, adding all the likelihood ratios at time $n\Delta_n = T$ of the runs that are such that $\tilde{N}(T) \in A$, and dividing this sum by R, we obtain an unbiased estimator of P.

Remark: The obvious advantage of the above algorithm is that the change-of-measure has to be computed just once, and can be applied in all runs. The drawback is that there is no control *within* the run: if the process happens to deviate from the minimizing path, it is not directed back towards this path. These considerations may lead to the following approach. Denote by $f^*(\cdot|n_0, A, T)$ the minimizing path corresponding to the probability P. Define

$$g(s) := f^*(s|\tilde{N}(t), A, T - t),$$

i.e., suppose that we find ourselves in state $\tilde{N}(t)$ at time t, and we wish to reach set A at time T, then $g(s)$ defines the most likely position at time $s + t$. Note that this implies that $g(0) = f^*(t)$. This gives rise to use the rates

$$\tilde{\lambda}_i(t) := \frac{1}{2} g_i'(t) + \frac{1}{2}\sqrt{(g_i'(t))^2 + 4\lambda_i \nu_i(\tilde{N}(t))};$$

$$\tilde{\nu}_i(t) := -\frac{1}{2}g_i'(t) + \frac{1}{2}\sqrt{(g_i'(t))^2 + 4\lambda_i\nu_i(\tilde{N}(t))},$$

$i = 1, \ldots, I$. It can be checked that also for these rates the expected position at time t is $f^*(t)$, but the difference with the first algorithm is that the process evolution is better controlled, cf. [4, 7]. In practice the interval $[0, T]$ is again split into n subintervals, and the rates $\tilde{\lambda}_i(k\Delta_n)$ and $\tilde{\nu}_i(k\Delta_n)$ are used in the k-th interval. Unfortunately, this approach is very time-consuming, as it requires the calculation of a minimizing path in each of the n subintervals.