TAIL QUANTILE ESTIMATION FOR NON-PREEMPTIVE PRIORITY QUEUES

Jin Guang

Shenzhen Research Institute of Big Data
The Chinese University of Hong Kong, Shenzhen
No.2001 Longxiang Street
Shenzhen, Guangdong 518172, P. R. CHINA

Guiyu Hong
Xinyun Chen

School of Data Science
The Chinese University of Hong Kong, Shenzhen
No.2001 Longxiang Street
Shenzhen, Guangdong 518172, P. R. CHINA

Xi Peng
Li Chen
Bo Bai
Gong Zhang

Theory Lab, Central Research Institute, 2012 Labs
Huawei Technologies, Co., Ltd.
No.2 Science Park West Avenue
Shatin, N.T., Hong Kong, P. R. CHINA

ABSTRACT

Motivated by applications in computing and telecommunication systems, we investigate the problem of estimating p-quantile of steady-state sojourn times in a single-server multi-class queueing system with non-preemptive priorities for p close to 1. The main challenge in this problem lies in efficient sampling from the tail event. To address this issue, we develop a regenerative simulation algorithm with importance sampling. In addition, we establish a central limit theorem for the estimator to construct the confidence interval. Numerical experiments show that our algorithm outperforms benchmark simulation methods. Our result contributes to the literature on rare event simulation for queueing systems.

1 INTRODUCTION

In computing and telecommunication industries, service-level agreement (SLA) usually takes the form of a guarantee on the tail probability of response times,

$$P(\text{response time} > \gamma) < 1 - p.$$

For example, $p = 0.99$ and $\gamma = 400$ms (Harchol-Balter 2021). In practice, the tail probability $1 - p$ is usually fixed, and the service provider needs a good estimator of p-quantile of response time before drafting an SLA contract. A computing or telecommunication system usually involves multiple priority classes and protocols, for which an analytic solution of the response time distribution is unavailable. In this paper, we investigate simulation techniques to estimate the p-quantile of steady-state response time for a given $p \approx 1$.

In particular, we consider a single node with priority queueing (PQ) scheduling as in Huawei CloudEngine 12800 and 12800E (Huawei 2021). Mathematically, it can be modeled as a non-preemptive single server with multiple priority classes. Suppose F is the distribution function of steady-state response time. Then

the *p*-quantile of F, $0 < p < 1$, is defined to be $Q(p) = \inf\{x : F(x) \geq p\}$. We develop a simulation algorithm that efficiently estimates the quantile $Q(p)$ for each priority class. To estimate the steady-state distribution and facilitate output analysis, we apply the regenerative simulation framework (Asmussen and Glynn 2007). A good estimator for the quantile $Q(p)$ essentially relies on sufficient samples from the rare event {response time $\approx Q(p)$} for $p \approx 1$. For this purpose, we apply importance sampling (IS) method that is widely used in rare event simulation and apply the cross-entropy (CE) method (De Boer et al. 2004) to optimize the IS distribution. Based on the regenerative cycles generated under the importance distribution, we design a new quantile estimator and establish the corresponding central limit theorem. As a consequence, we can construct confidence intervals for the quantile using batching methods. The numerical results show that our algorithm outperforms benchmark algorithms. Finally, we apply our algorithm to SLA estimation for a node with 8 priority classes.

Our work is related to the literature on rare event simulation of queueing systems, see Blanchet and Mandjes (2009) and the references therein. Most existing works are on single-class systems, including G/G/1 queue (Blanchet et al. 2007), fluid network (Chang et al. 1994) and Jackson network (Dupuis and Wang 2009). There are a few exceptions on processing sharing system (Mandjes and Zwart 2006) and priority queues (Setayeshgar 2012). In those works, the IS distributions are typically derived via large deviation principle (LDP), see for example Parekh and Walrand (1989) or Dupuis et al. (2007). Among these works, the setting in Setayeshgar (2012) is probably the closest to ours. In Setayeshgar (2012), the author studies IS for a preemptive server with 2 priority classes based on LDP analysis. To the best of our knowledge, however, there is no existing result on non-preemptive priority queues. This is probably because both LDP and differential game approaches rely on sophisticated analysis on the system dynamics and hence are hard to be applied to systems with multiple customer classes and general service protocols. In this light, we apply the CE method to optimize the IS distribution numerically. In our numerical experiments, we show that preemptive and non-preemptive queues behave differently conditional on the rare event. As a consequence, our algorithm outperforms the IS algorithm proposed in Setayeshgar (2012), which was designed for preemptive systems. We also note that most existing works in the area of rare event simulation have focused on the distribution of total workload in the system, while our algorithm works for the sojourn time of individual customers.

Organization of the rest of this paper: The queueing model and the simulation goal are explained in Section 2. In Section 3, we first explain the key components in algorithm design and then present the complete algorithm. In Section 4, we establish CLT for our quantile estimator and construct the confidence intervals. Numerical results are reported in Section 5.

2 PROBLEM SETTING

We consider a single server with jobs of multiple priority classes. Let K be the number of priority classes. We index the classes by $k = 1, 2, ..., K$, with class 1 having the highest priority and class K the lowest priority. For $k = 1, 2, ..., K$, jobs of class k arrive according to a Poisson process with the rate λ_k and have i.i.d. exponential service times with mean $1/\mu_k$. Following the setting in real systems, we assume that the server is **non-preemptive**, i.e., low priority jobs in service are not interrupted by high priority jobs. The arrival processes and service times are assumed to be mutually independent.

Our goal is to estimate a commonly used performance metric in SLA (Harchol-Balter 2021) for the above non-preemptive system via simulation. In detail, for a given $p \in (0,1)$ and each class k, let $Q_k(p)$ be the *p*-tile of total sojourn time, also known as response time, $R_{k,\infty}$ of class-*k* jobs in the **steady state**, i.e.,

$$Q_k(p) = \inf\{\gamma : P(R_{k,\infty} \leq \gamma) \geq p\} = \inf\{\gamma : P(R_{k,\infty} > \gamma) < 1 - p\}.$$

For a given $p \in (0,1)$, we need to estimate $Q_k(p)$ for all $k = 1, 2, ..., K$. In the setting of SLA, p is typically close to 1, say $p = 0.999$ or 0.99999. In other words, the problem is essentially to estimate the quantile corresponding to an extreme event in the steady-state distribution of a queueing system.

3 ALGORITHM DESIGN

For a given priority class k, Let $F(\gamma) = P(R_{k,\infty} \leq \gamma)$ be the CDF of steady-state response time. Intuitively, we can obtain a good estimation of $Q(p)$ if we have a good estimation of $F(\gamma)$ for γ in a neighborhood around $Q(p)$. Our algorithm design is based on this idea and contains three key components. First, to deal with the steady-state distribution, we apply the regenerative simulation technique (Section 3.1). As $\{R_\infty > \gamma\}$ are rare event for γ close to $Q(p)$, we use IS to improve simulation efficiency (Section 3.2). The IS distribution is optimized by cross-entropy method (Section 3.3). Finally, we present our estimator for $Q(p)$ in Section 3.4 along with the complete algorithm.

3.1 Regenerative Simulation

The dynamics of the queueing model described in Section 2 can be viewed as a regenerative process. In particular, the system regenerates whenever a job finds the server idle upon departure. For a regenerative cycle, denote by α the cycle length in unit of time and α_k the total number of jobs of class k served in this cycle. For $n = 1, 2, ..., \alpha_k$, let $R_{k,n}$ be the response time of the n-th job of class k. Then by renewal reward theorem (Crane and Lemoine 1977), we have, for any priority class k and $\gamma > 0$

$$P(R_{k,\infty} > \gamma) = \frac{\mathbb{E}\left[\sum_{n=1}^{\alpha_k} 1_{\{R_{k,n} > \gamma\}}\right]}{\mathbb{E}[\alpha_k]}.$$

The enumerator $\mathbb{E}\left[\sum_{n=1}^{\alpha_k} 1_{\{R_{k,n} > \gamma\}}\right]$ involves a rare event when γ is large and hence will be estimated by IS method which we explain in the Section 3.2. But to estimate the denominator $\mathbb{E}[\alpha_k]$ does not necessarily requires importance sampling.

3.2 Importance Sampling for Tail Probability Estimation

Intuitively, in order to get good estimators for $Q_k(p)$, we need to choose the importance distribution such that the events $\{R_{k,n} > \gamma\}$ happens frequently for $\gamma \approx Q(p)$. Suppose $Q(p) < \gamma_{\max}$ and the constant $\gamma_{\max}$ is known. Then, we shall design the importance distribution such that event $\{R_{k,n} > \gamma_{\max}\}$ happens frequently and hence so are $\{R_{k,n} > \gamma\}$ for $\gamma \approx Q_k(p)$.

The queueing system is driven by the jobs' inter-arrival times and services times. Denote by $f_l^A(\cdot)$ and $f_l^S(\cdot)$ be the density of inter-arrival times and service times of class-l jobs, for $l = 1, 2, ..., K$. Our design of IS simulation follows the so-called switching change of measure (Rubinstein and Kroese 2004). In particular, starting from time 0, we generate the inter-arrival times and service times according to some IS density $\tilde{f}_l^A$ and $\tilde{f}_l^S$, $l = 1, 2, ..., K$ and simulate the system dynamics accordingly. The choice of $\tilde{f}_l^A$ and $\tilde{f}_l^S$ will be discussed in Section 3.3. So for, let's take $\tilde{f}_l^A$ and $\tilde{f}_l^S$ as given. In switching change of measure, the simulator will switch back to the original distribution f_l^A and f_l^S once the event of interest $\{R_{k,n} > \gamma_{\max}\}$ is observed. In principle, the event $\{R_{k,n} > \gamma_{\max}\}$ is observed only after job n departs. To further reduce the variance caused by the random likelihood ratio, we shall define a more sophisticated switching rule as follows.

Suppose for jobs of classes $l = 1, 2, ..., k$, their service times are realized upon arrival. At the arrival time of job n in class k, define $R'_{k,n}$ as the sum of service times of all jobs of class $1 \leq l \leq k$ waiting in queues and the remaining service time of the job in service. Note that $R'_{k,n}$ is a lower bound of the actual response time $R_{k,n}$ and is known immediately upon arrival. Define τ_k as the arrival time of the first class-k job such that $R'_{k,n} > \gamma_{\max}$.

$$\tau_k = \inf\left\{\sum_{i=1}^{n} A_{k,i} : R'_{k,n} > \gamma_{\max}\right\}.$$

By definition, τ_k is a stopping time with respect to filtration $\{\mathscr{F}_t : t \geq 0\}$, where $\mathscr{F}_t$ includes events corresponding to all arrivals before time t, service times of jobs in classes $l \leq k$ that have arrived by time t, and service times of jobs in classes $l > k$ that have entered service by time t. In our algorithm, the simulator will switch back to the original distribution after $\tau_k \wedge \alpha$ and terminate at α when a job finds the server idle upon departure.

Now we explain how to compute the likelihood ratio function for a sample path of the regenerative cycle generated by our IS algorithm. For each class $l = 1, 2, ..., K$, denote by $A_{l,n}$ the epoch between the $(n-1)$-th and n-th arrivals of class l, and denote by $S_{l,n}$ the service time of n-th arrival of class l. Let $G_l^A(\cdot)$ be the tail probability of inter-arrival times in class l. For any time $t \in [0, \alpha]$, denote by $N_l^A(t)$ and $N_l^S(t)$ the number of job arrivals and jobs that have entered service of class l by time t, respectively. Then, the likelihood ratio upon time t can be computed as

$$L(t) = \frac{\prod_{l=1}^{K} \left(\prod_{n=1}^{N_l^A(t \wedge \tau_k)} f_l^A(A_{l,n}) \cdot G_l^A(H_l^A) \right) \cdot \prod_{l=1}^{k} \prod_{n=1}^{N_l^A(t \wedge \tau_k)} f_l^S(S_{l,n}) \cdot \prod_{l=k+1}^{K} \prod_{n=1}^{N_l^S(t \wedge \tau_k)} f_l^S(S_{l,n})}{\prod_{l=1}^{K} \left(\prod_{n=1}^{N_l^A(t \wedge \tau_k)} \tilde{f}_l^A(A_{l,n}) \cdot \tilde{G}_l^A(H_l^A) \right) \cdot \prod_{l=1}^{k} \prod_{n=1}^{N_l^A(t \wedge \tau_k)} \tilde{f}_l^S(S_{l,n}) \cdot \prod_{l=k+1}^{K} \prod_{n=1}^{N_l^S(t \wedge \tau_k)} \tilde{f}_l^S(S_{l,n})}, \quad (1)$$

where H_l^A is the age since last arrival of class l at time $t \wedge \tau_k$. Note that the above formula of likelihood still holds if t is replaced with a stopping time with respect to the filtration $\{\mathscr{F}_t\}$, see Chapter 5.1c of Asmussen and Glynn (2007).

Let $E_{k,n}$ be the enter-service time of the n-th job in class k. Then, $R_{k,n} \in \mathscr{F}_{E_{k,n}}$. As a consequence, for any $\gamma > 0$, we have

$$\mathbb{E}\left[\sum_{n=1}^{\alpha_k} 1_{\{R_{k,n} > \gamma\}} \right] = \tilde{\mathbb{E}}\left[\sum_{n=1}^{\alpha_k} 1_{\{R_{k,n} > \gamma\}} L(E_{k,n}) \right],$$

where $L(\cdot)$ is the likelihood function defined in (1).

Suppose we have independently generated m_1 regenerative cycles via IS simulation and m_2 cycles via naive simulation. We use superscript i to denote the i-th cycle and "~" to denote samples generated by IS simulation. Our IS estimator for tail probability of the steady-state response time is given by

$$\hat{P}_{m_1, m_2}(R_{k,\infty} > \gamma) = \frac{m_1^{-1} \sum_{i=1}^{m_1} \sum_{n=1}^{\tilde{\alpha}_k^i} L(\tilde{E}_{k,n}^i) 1_{\{\tilde{R}_{k,n}^i > \gamma\}}}{m_2^{-1} \sum_{i=1}^{m_2} \alpha_k^i}. \quad (2)$$

3.3 Cross-Entropy Method for Searching Importance Measure

Theoretically, the optimal IS distribution (Asmussen and Glynn 2007) for the numerator $\mathbb{E}\left[\sum_{n=1}^{\alpha_k} 1_{\{R_{k,n} > \gamma\}} \right]$ is

$$\frac{\sum_{n=1}^{\alpha_k} 1_{\{R_{k,n} > \gamma\}} \cdot h(\alpha)}{\mathbb{E}\left[\sum_{n=1}^{\alpha_k} 1_{\{R_{k,n} > \gamma\}} \right]},$$

where $h(\alpha)$ is the density of a regenerative cycle under the original distribution. Note that the optimal IS distribution is not practical as it involves the unknown constant $\mathbb{E}\left[\sum_{n=1}^{\alpha_k} 1_{\{R_{k,n} > \gamma\}} \right]$. Still, if we can choose a distribution close to the optimal IS distribution, it is likely that using it as IS distribution could efficiently reduce the variance. Based on this idea, cross-entropy method (Rubinstein and Kroese 2004) uses KL divergence to measure the distance between two distributions and chooses IS distribution by solving the following optimization problem:

$$\max_{\boldsymbol{f}'} \mathbb{E}\left[\sum_{n=1}^{\alpha_k} 1_{\{R_{k,n} > \gamma\}} \cdot \log L'(\alpha) \right], \quad (3)$$

where $\boldsymbol{f}' = (f'^A_1, \ldots, f'^A_K, f'^S_1, \ldots, f'^S_K)$ are the densities of the inter-arrival and service times under the IS distributions, and $L'(t)$ is the likelihood ratio with respect to the IS distribution $\boldsymbol{f}'$ as defined in (1).

The objective function (3) can be estimated by simulation data, and we can use the IS distribution $\tilde{f}$ for variance reduction. Suppose we use N simulated regenerative cycles to estimate the objective function, (3) is approximated by

$$\max_{\boldsymbol{f}'} \frac{1}{N} \sum_{i=1}^{N} \left(\sum_{n=1}^{\tilde{\alpha}^i_k} 1_{\{\tilde{R}^i_{k,n} > \gamma\}} L(\tilde{E}^i_{k,n}) \cdot \log L'(\tilde{\alpha}^i) \right). \tag{4}$$

For computational efficiency, N should be much smaller than the number of cycles m_1 of (2). In our setting, we set $\boldsymbol{f}'$ as exponential distributions with rates $(\lambda'_1, \ldots, \lambda'_K, \mu'_1, \ldots, \mu'_K)$, which belong to the same parametric family as the original distributions. Then, problem (4) has a closed-form solution:

$$\lambda'_l = \frac{\sum_{i=1}^{M} \left(\sum_{n=1}^{\tilde{\alpha}^i_k} 1_{\{\tilde{R}^i_{k,n} > \gamma\}} L(\tilde{E}^i_{k,n}) \cdot \tilde{N}^{A,i}_l(\tilde{\tau}^i_k \wedge \tilde{\alpha}^i) \right)}{\sum_{i=1}^{M} \left(\sum_{n=1}^{\tilde{\alpha}^i_k} 1_{\{\tilde{R}^i_{k,n} > \gamma\}} L(\tilde{E}^i_{k,n}) \right) \left(\sum_{n=1}^{\tilde{N}^{A,i}_l(\tilde{\tau}^i_k \wedge \tilde{\alpha}^i)} \tilde{A}^i_{l,n} + \tilde{H}^{A,i}_l \right)}, \quad l = 1, \ldots, K \tag{5}$$

and

$$\mu'_l = \begin{cases} \dfrac{\sum_{i=1}^{M} \left(\sum_{n=1}^{\tilde{\alpha}^i_k} 1_{\{\tilde{R}^i_{k,n} > \gamma\}} L(\tilde{E}^i_{k,n}) \right) \tilde{N}^{A,i}_l(\tilde{\tau}^i_k \wedge \tilde{\alpha}^i)}{\sum_{i=1}^{M} \left(\sum_{n=1}^{\tilde{\alpha}^i_k} 1_{\{\tilde{R}^i_{k,n} > \gamma\}} L(\tilde{E}^i_{k,n}) \right) \left(\sum_{n=1}^{\tilde{N}^{A,i}_l(\tilde{\tau}^i_k \wedge \tilde{\alpha}^i)} \tilde{S}^i_{l,n} \right)}, & l = 1, \ldots, k, \\[4mm] \dfrac{\sum_{i=1}^{M} \left(\sum_{n=1}^{\tilde{\alpha}^i_k} 1_{\{\tilde{R}^i_{k,n} > \gamma\}} L(\tilde{E}^i_{k,n}) \right) \tilde{N}^{S,i}_l(\tilde{\tau}^i_k \wedge \tilde{\alpha}^i)}{\sum_{i=1}^{M} \left(\sum_{n=1}^{\tilde{\alpha}^i_k} 1_{\{\tilde{R}^i_{k,n} > \gamma\}} L(\tilde{E}^i_{k,n}) \right) \left(\sum_{n=1}^{\tilde{N}^{S,i}_l(\tilde{\tau}^i_k \wedge \tilde{\alpha}^i)} \tilde{S}^i_{l,n} \right)}, & l = k+1, \ldots, K. \end{cases} \tag{6}$$

However, the cross-entropy method is not suitable for the rare event simulation directly, as the objective (4) is 0 with a high probability. To address the issue, we use the adaptive cross-entropy method to select good IS distributions by adopting a two-stage procedure where both the level γ and the IS distributions are iteratively updated (Rubinstein and Kroese 2004). Specifically, each iteration consists of two steps. In the first step, the algorithm samples N regenerative cycles from the IS distribution with rates $\lambda_{l,t-1}, \mu_{l,t-1}$ for $l = 1, \ldots, K$, and updates the level γ_t as the $1 - \rho$ sample quantile of the maximum response times within each cycle. In the second step, the algorithm solves (4) with $\gamma = \gamma_t$ and updates the IS rates $\lambda_{l,t} = \lambda'_l$, $\mu_{l,t} = \mu'_l$ for $l = 1, \ldots, K$. The iteration terminates when γ_t exceeds the target level γ. More details are given in the Algorithm 1.

3.4 From Tail Probability to SLA

Finally, we explain how to estimate $Q_k(p)$ using the simulated regenerative cycles under IS distribution. Let $G(\gamma) = P(R_{k,\infty} > \gamma)$ be the true tail probability of the response time. By definition,

$$Q_k(p) = \inf\{\gamma : P(R_{k,\infty} \leq \gamma) \geq p\} = \inf\{\gamma : G(\gamma) < 1 - p\}.$$

According to (2), for any $\gamma \in [0, \gamma_{\max}]$, we can estimate $\hat{G}(\gamma) = \hat{P}_{m_1,m_2}(R_{k,\infty} > \gamma)$. Then, a natural estimator for $Q_k(p)$ is

$$\hat{Q}^k_{m_1,m_2}(p) = \inf\left\{ \gamma : \hat{P}_{m_1,m_2}(R_{k,\infty} > \gamma) < 1 - p \right\}. \tag{7}$$

Equation (7) can be evaluated in the following way. Suppose the m_1 regenerative cycles generated by IS contain in total $\tilde{\beta}_{m_1}$ jobs of type k. We index their response time and likelihood ratio by $\tilde{R}_{k,n}$ and $L_{k,n}$,

denoting the response time of class-k job n and the corresponding likelihood ratio, respectively. Sort $\tilde{R}_{k,n}$ in ascending order, thereby forming the ordered samples $(\tilde{R}_{k,(1)},\ldots,\tilde{R}_{k,(\tilde{\beta}_{m_1})})$. Then,

$$\hat{Q}^k_{m_1,m_2}(p) = \tilde{R}_{k,(n_p)}, \text{ where } n_p = \min\left\{ n : \sum_{i=n}^{\tilde{\beta}_{m_1}} L_{k,(i)} \leq \frac{(1-p)m_1}{m_2} \sum_{i=1}^{m_2} \alpha_k^i \right\}. \tag{8}$$

Now we are ready to present the complete SLA simulation algorithm.

Algorithm 1: SLA Simulation Algorithm

Input: target class k, the number of cycles m_1 and m_2, and $\gamma_{\max} > Q_k(p)$.
1. Denominator Estimation. Generate m_2 cycles by direct Monte Carlo. Obtain α_k^i for $i = 1,\ldots,m_1$.
2. Call CE algorithm for Searching IS measure. Input parameter $N \geq 1$ and $\rho \in (0,1)$.
Initialize $\lambda_{l,0} = \lambda_l, \mu_{l,0} = \mu_l$ for $l = 1,2,\ldots,K$.
for $t = 1,\ldots$ **do**
> **Adaptive update of γ_t.** Generate N cycles under the distribution with rates $\lambda_{l,t-1}, \mu_{l,t-1}$, and compute the sample $(1-\rho)$-quantile γ_t, according to $\left\{ \max_{n=1,\ldots,\alpha_k^i} R_{k,n} : l = 1,\ldots,N \right\}$.
> Update $\gamma_t = \min\{\gamma_t, \gamma_{\max}\}$.
> **Adaptive update of IS distributions.** Plugin the data from the N cycles into (5) and (6) to obtain $\lambda_{l,t}$ and $\mu_{l,t}$.
> **if** *The adaptive level γ_t has reached $\gamma_{\max}$, i.e. $\gamma_t = \gamma_{\max}$* **then**
> > Set the IS measure with rates $\tilde{\lambda}_l = \lambda_{l,t}, \tilde{\mu}_l = \mu_{l,t}$ for $l = 1,\ldots,K$, and break the loop.

3. IS for Tail Probability Estimation. Sample m_1 cycles under the switching change of measure with $\tilde{f}_l^A \sim \exp(\tilde{\lambda}_l)$ and $\tilde{f}_l^S \sim \exp(\tilde{\mu}_l)$. For all $\tilde{\beta}_{m_1}$ jobs of class k, record their response time $\tilde{R}_{k,n}$ and likelihood ratio $L_{k,n}$ computed according to (1).
4. Quantile Estimation. Compute $\hat{Q}^k_{m_1,m_2}(p)$ according to (8).
Output: $\hat{Q}^k_{m_1,m_2}(p)$.

Remark 1 The assumption on the availability of $\gamma_{\max}$ can actually be relaxed. Given the CLT result in Section 4, we can heuristically test whether γ_t obtained in each iteration in CE is an upper bound for $Q_k(p)$. Then, we terminate the CE iteration when the test result is yes for the first time and use γ_t as $\gamma_{\max}$ in the following IS step. In the numerical experiments, we do not assume any knowledge on $\gamma_{\max}$ and apply this heuristic method.

4 CONFIDENCE INTERVAL

Our next step is to construct the confidence interval for the estimator $\hat{Q}^k_{m_1,m_2}(p)$. To do this, we first need to establish the central limit theorem. We consider the case where $m_1 = m_2 = m$. For any given class k and SLA level p, write $\hat{Q}_m(p) = \hat{Q}^k_{m,m}(p)$. For $\gamma > 0$ and $i = 1,2,\ldots,m$, define

$$\tilde{Y}_{k,i}(\gamma) = \sum_{n=1}^{\tilde{\alpha}_k^i} L_{k,n} 1_{\{\tilde{R}_{k,n} > \gamma\}}, \text{ and } Z_{k,i}(\gamma) = \tilde{Y}_{k,i} - P(R_{k,\infty} > \gamma)\alpha_k^i.$$

Denote $\sigma^2(\gamma) = \text{var}(Z_{k,i}(\gamma))$. Following the approach in Iglehart (1976), we establish the following CLT result for our SLA estimator $\hat{Q}_m^p$.

Theorem 1 Suppose $m_1 = m_2 = m$. For any k and p, denoted $\hat{Q}_m(p) = \hat{Q}_{m,m}^k(p)$. Let F be the CDF of $R_{k,\infty}$. Suppose for all γ in some neighborhood of $Q(p)$, $F''(\gamma)$ exists, $|F''(\gamma)| \leq M < \infty$ and $E[|Z_{k,1}(\gamma)|^{2+\varepsilon}] \leq M < \infty$ for some constants $\varepsilon, M > 0$ independent of γ. Then, as $m \to \infty$,

$$\frac{\sqrt{m}(\hat{Q}_m(p) - Q(p))}{\sigma(Q(p))/(E[\alpha_k]F'(Q(p)))} \Rightarrow N(0,1).$$

Proof. Define $\hat{F}_m(\cdot) = 1 - \hat{P}_{m,m}(R_{k,\infty} > \cdot)$. Then, $\hat{F}_m(\gamma)$ is right-continuous and non-decreasing for all $\gamma > 0$. Therefore,

$$P\left(\hat{Q}_m(p) \leq \gamma\right) = P\left(p \leq \hat{F}_m(\gamma)\right).$$

Let

$$a_m(\gamma) = Q(p) + \frac{\gamma\sigma(Q(p))}{\sqrt{m}E[\alpha_k]F'(Q(p))}.$$

Then,

$$
\begin{aligned}
A_m(\gamma) &\triangleq P\left(\frac{\sqrt{m}(\hat{Q}_m(p) - Q(p))}{\sigma(Q(p))/(E[\alpha_k]F'(Q(p)))} \leq \gamma\right) \\
&= P\left(\hat{Q}_m(p) \leq a_m(\gamma)\right) = P\left(p \leq \hat{F}_m(a_m(\gamma))\right) = P\left(\hat{P}_m(R_{k,\infty} > a_m(\gamma)) \leq 1 - p\right) \\
&= P\left(\sum_{i=1}^{m} Z_{k,i}(a_m(\gamma)) \leq \sum_{i=1}^{m} \alpha_{k,i} \cdot (F(a_m(\gamma)) - p)\right).
\end{aligned}
$$

Following the regularity conditions on $F(\cdot)$, we can expand $F(\cdot)$ around $Q(p)$ as

$$F(a_m(\gamma)) = p + \frac{\gamma\sigma(Q(p))}{\sqrt{m}E[\alpha_k]} + O\left(\frac{1}{m}\right).$$

Then,

$$A_m(\gamma) = P\left(\frac{1}{\sqrt{m}\sigma(Q(p))}\sum_{i=1}^{m} Z_{k,i}(a_m(\gamma)) \leq \frac{\gamma}{mE[\alpha_k]}\left(\sum_{i=1}^{m}\alpha_{k,i}\right)\left(1 + O\left(\frac{1}{m}\right)\right)\right)$$

By strong law of large number, $\left(\sum_{i=1}^{m}\alpha_{k,i}\right)/(mE[\alpha_k]) \to 1$ almost surely as $m \to \infty$. Therefore,

$$\frac{\gamma}{mE[\alpha_k]}\left(\sum_{i=1}^{m}\alpha_{k,i}\right)\left(1 + O\left(\frac{1}{m}\right)\right) \to \gamma,$$

almost surely as $m \to \infty$. On the other hand, as $E[|Z_{k,1}(\gamma)|^{2+\varepsilon}] \leq M < \infty$ for all γ in the neighborhood of $Q(p)$, the Liapunov condition holds and therefore by Corollary 9.8.1 of Resnick (2019) , as $m \to \infty$,

$$\frac{1}{\sqrt{m}\sigma(a_m(\gamma))}\sum_{i=1}^{m} Z_{k,i}(a_m(\gamma)) \Rightarrow N(0,1).$$

Finally, by the continuity of $\sigma(\cdot)$ around $Q(p)$ and the continuous mapping theorem, we have

$$\frac{1}{\sqrt{m}\sigma(Q(p))}\sum_{i=1}^{m} Z_{k,i}(a_m(\gamma)) \Rightarrow N(0,1).$$

Therefore, we can conclude

$$A_m(\gamma) \to \Phi(\gamma), \quad \text{as } m \to \infty.$$

where Φ is the standard normal distribution function. $\qquad\square$

The CLT result indicates that $\hat{Q}_m(p)$ is a consistent estimator for $Q(p)$. In addition, given the CLT result, we can use batching method (Hsieh and Glynn 2002) to construct confidence intervals. Suppose $m = cr$ so that we divide m cycles into r batches of c cycles each. Let $\hat{Q}_{c,i}(p)$ denote the sample quantile computed from (8) using data in the i-th batch. Then, the variance is estimated by

$$\hat{\sigma}_m^2 = \frac{1}{r-1} \sum_{i=1}^{r} \left(\hat{Q}_{c,i}(p) - \hat{Q}_m(p) \right)^2,$$

and the corresponding 95% confidence interval is

$$CI_m = \left[\hat{Q}_m(p) - \frac{1.96\hat{\sigma}_m}{\sqrt{r}}, \hat{Q}_m(p) + \frac{1.96\hat{\sigma}_m}{\sqrt{r}} \right].$$

5 NUMERICAL RESULTS

In the numerical experiments, we first illustrate the difference in the choice of IS distribution for preemptive and non-preemptive systems with an example of 2 priority queues. Then, we implement Algorithm 1 using 2 priority queues, and compare its performance with two benchmark simulation algorithms. Finally, we apply our algorithm to a system with 8 priority queues as used in Huawei CloudEngine 12800 and 12800E to estimate the SLA performance metrics.

5.1 Non-preemptive versus Preemptive

In this part, we illustrate the difference between non-preemptive and preemptive systems using a simple example of 2 priority queues. The system parameter $(\lambda_1, \lambda_2, \mu_1, \mu_2) = (0.2, 0.4, 2, 1)$, i.e. high priority jobs arrive less frequent and are served faster, which is consistent to our observation in real computing system as described in Section 5.3. In Setayeshgar (2012), the author derives an asymptotic optimal IS distribution via LDP analysis for a preemptive system, and we refer to this distribution as LDP in the rest of the section. In Table 1 below, we compare the conditional distribution of the system under LDP, and the IS distribution learned by CE procedure as in Algorithm 1 (referred as CE), and the original distribution, given that a class-1 job experiences a long delay ($> \hat{Q}_1(0.999) \approx 6.913$).

Table 1: Conditional probabilities of LDP and CE importance distribution compared to that of the original distribution estimated by 1000,000 regenerative cycles. A cycle is said effective if in this cycle there exists at least one class-1 job experiencing long delay.

Method	# effective cycle	Conditional probability	
		after a class-1 job	after a class-2 job
Naive	514	0.128	0.872
LDP	546652	0.454	0.544
CE	548942	0.107	0.892

In Table 1, we report the estimated conditional probabilities that a long-delayed job is blocked by a class-1 and that by a class-2 job right before it. According to the results, we can see that the conditional probabilities estimated by data simulated by LDP are quite different from those by the original distribution, while those by CE are close to those by the original distribution. The comparison results explain why IS results derived from preemptive systems can not be directly applied to non-preemptive systems.

5.2 Performance Test

We implement Algorithm 1 with a system with 2 priority queues for performance test. In particular, we compare Algorithm 1 with two benchmarks:

- Naive: regenerative simulation using original distribution.

- LDP: regenerative simulation using IS distribution derived for preemptive system via LDP analysis as given in Setayeshgar (2012).

We consider a 2 priority queue with parameter $\mu_1 = \mu_2$ so that the true value of $Q_k(p)$ can be computed via Laplace transformation (Kella and Yechiali 1985). Given the true value of $Q_k(p)$, we use mean square error (MSE) and CI coverage rate to measure the performance of a simulation algorithm. For the purpose of efficiency comparison, we fixed the number of regenerative cycles in each simulation round and estimated MSE and CI coverage rate by 100 rounds of simulation. For Algorithm 1, we set $N = 10,000$ and $\rho = 10\%$ for the CE step. The simulation results for 0.999-quantile of the steady-state response times are reported in Table 2. The results show that our algorithm is more efficient than the two benchmarks as it obtains a significantly smaller MSE and a higher CI coverage rate, given the same number of regenerative cycles.

Compared to Naive and LDP, Algorithm 1 has an extra numerical routine to optimize importance distribution by CE, which will introduce extra computation cost in addition to simulating the regenerative cycles. Although we did not have a theoretic guarantee on the convergence speed of CE, in the numerical experiments with 2 priority-queues, we find CE obtains good IS parameters in just a few iterations and thus does not add much extra computation time. For example, the IS parameter used by Algorithm 1 as reported in Table 2 is obtained by CE in 8 iterations that take 5.94 seconds per iteration.

Table 2: Results for 0.999-quantile estimation of steady-state response time in a 2 priority queue with $(\lambda_1, \lambda_2, \mu_1, \mu_2) = (0.1, 0.2, 1, 1)$. Sample mean, MSE and CI coverage rate are estimated based on 100 rounds of simulation.

class	Method	#cycle	$\tilde{\lambda}_1$	$\tilde{\lambda}_2$	$\tilde{\mu}_1$	$\tilde{\mu}_2$	$Q(p)$	sample mean	MSE	coverage rate of CI
high	Naive	10,000	-	-	-	-	8.524	8.598	0.535	65%
	Naive	100,000	-	-	-	-	8.524	8.477	0.046	65%
	LDP	10,000	0.333	0.500	0.300	0.300	8.524	8.451	0.0394	63%
	LDP	100,000	0.333	0.500	0.300	0.300	8.524	8.500	0.0170	31%
	Alg 1	10,000	0.330	0.224	0.234	0.238	8.524	8.527	0.0035	91%
	Alg 1	100,000	0.330	0.224	0.234	0.238	8.524	8.522	0.0005	86%
low	Naive	10,000	-	-	-	-	11.541	11.618	1.560	61%
	Naive	100,000	-	-	-	-	11.541	11.524	0.127	54%
	LDP	10,000	0.333	0.500	0.300	0.300	11.541	11.429	0.305	35%
	LDP	100,000	0.333	0.500	0.300	0.300	11.541	11.488	0.0414	35%
	Alg 1	10,000	0.219	0.435	0.341	0.339	11.541	11.496	0.0227	78%
	Alg 1	100,000	0.219	0.435	0.341	0.339	11.541	11.542	0.0004	71%

We also test the simulation algorithms for estimating more extreme quantiles and the results are reported in Table 3. We find the difference in the performance measures between Algorithm 1 and the benchmarks becomes more significant as p gets closer to 1.

5.3 Application to SLA Estimation

We consider a priority queueing model used in Huawei CloudEngine as illustrated by Figure 1.

There are eight queues with indexes ranging from 0 to 7 on each interface in the outbound direction of a switch. The priorities of queues 7 to 0 are in the descending order defined by the Internet Engineering Task Force (IETF), i.e., class selector 7 (CS7), CS6, expedited forwarding (EF), assured forwarding 4 (AF4), AF3, AF2, AF1, and best effort (BE), respectively. Current Internet practice suggests CS7 and CS6 for the network control and signaling, EF for telephony, AF4 to AF1 for multimedia streaming, and BE for flows without bandwidth assurance such as email and telnet services. According to the flow characteristics, the model parameters are set as in Table 4. We then apply Algorithm 1 with number of regenerative cycles

Table 3: Results for quantile estimation of steady-state response time in a 2 priority queue with $(\lambda_1, \lambda_2, \mu_1, \mu_2) = (0.1, 0.2, 1, 1)$ with $p_1 = 1 - 10^{-5}$ and $p_2 = 1 - 10^{-10}$. Sample mean, MSE and CI coverage rate are estimated based on 100 rounds of simulation with 100,000 regenerative cycles in each round.

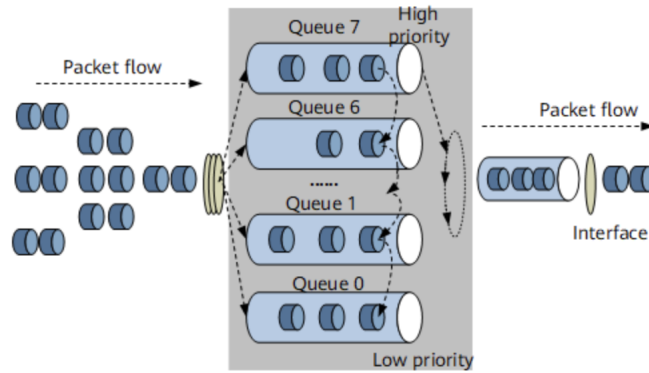| class	Method	$Q(p_1)$	sample mean	MSE	coverage rate of CI	$Q(p_2)$	sample mean	MSE	coverage rate of CI
high	Naive	13.809	13.177	1.8733	8%	26.753	-	-	-
	LDP	13.809	13.778	0.0153	65%	26.753	26.591	0.1331	47%
	Alg 1	13.809	13.804	0.0008	82%	26.753	26.739	0.0035	71%
low	Naive	20.637	20.251	5.5800	26%	44.211	-	-	-
	LDP	20.637	20.411	0.4468	26%	44.211	42.430	6.0641	4%
	Alg 1	20.637	20.617	0.0166	54%	44.211	44.208	0.0270	72%

Figure 1: Illustration of a network device with Priority Queueing scheduling.

$m_1 = m_2 = 100,000$, $N = 10,000$ and $\rho = 0.1$ for the CE step. The simulation results and running times are reported in Table 5 and Table 6.

Table 4: Model parameters of 8 priority queues.

Queue index	7	6	5	4	3	2	1	0
Service class	CS7	CS6	EF	AF4	AF3	AF2	AF1	BE
mean packet size (byte)	100	100	200	1400	1400	1400	1400	1400
mean arrival rate (Mbps)	10	10	10	400	200	100	100	450

Our goal is to estimate $Q_k(p)$ for $k = 0, 1, ..., 7$ and $p = 0.999$ and 0.99999. To control the estimation accuracy, we use a pre-simulation to estimate the CLT variance via batching and then choose the number of regenerative cycles m so that the relative error (defined as $\hat{\sigma}_m/\hat{Q}_m(p)$) is less than 0.1%. In this case, as μ_k are unequal, the true value of $Q_k(p)$ is not available. So to verify the simulation estimation $\hat{Q}_k(p)$, we run independent IS simulation using 100,000 regenerative cycles to estimate $P(R_{k,\infty} > \hat{Q}_k(p))$. If our estimation is accurate, the estimated probability should be close to p. The simulation is implemented in Python and runs on a 16-cores server with Intel Cascade Lake 3.0GHz CPU. The simulation results and computation times (including pre-simulation and all steps in Algorithm 1) are reported in Table 5 and Table 6 below. It is notable that when the tail probability $1 - p$ shrinks by 100 times, the total computation time of our method only doubles from 12327s to 26875s to achieve the same relative precision level.

Table 5: Simulation results and running time for 0.999-quantile estimation of steady-state response time of 8 priority classes. The unit of response time is microsecond (μs).

class	sample quantile	CI	#cycle	probability	CI	running time (s)
7	22.855	(22.811, 22.900)	2836384	0.001028	(0.000946, 0.001111)	1409
6	22.880	(22.835, 22.925)	3074112	0.001020	(0.000949, 0.001091)	1738
5	23.260	(23.215, 23.305)	3060256	0.001041	(0.000923, 0.001159)	1774
4	33.970	(33.906, 34.034)	817056	0.001003	(0.000962, 0.001044)	470
3	45.013	(44.926, 45.099)	1987216	0.000962	(0.000911, 0.001014)	1101
2	53.259	(53.155, 53.364)	3847392	0.001059	(0.000983, 0.001135)	2434
1	59.139	(59.028, 59.251)	4140432	0.001018	(0.000943, 0.001093)	2592
0	74.354	(74.208, 74.500)	1604864	0.001006	(0.000964, 0.001048)	809

Table 6: Simulation results and running time for 0.99999-quantile estimation of steady-state response time of 8 priority classes. The unit of response time is microsecond (μs).

class	sample quantile	CI	#cycle	probability	CI	running time (s)
7	39.818	(39.748, 39.888)	4155440	0.0000093	(8.554e-06, 1.012e-05)	3321
6	39.664	(39.587, 39.742)	3614400	0.0000101	(9.194e-06, 1.107e-05)	2801
5	40.480	(40.400, 40.559)	4278560	0.0000091	(8.142e-06, 1.001e-05)	3054
4	54.354	(54.247, 54.460)	268624	0.0000110	(9.499e-06, 1.247e-05)	268
3	81.637	(81.478, 81.797)	1950080	0.0000096	(8.794e-06, 1.036e-05)	1717
2	100.922	(100.725, 101.120)	5303456	0.0000089	(7.536e-06, 1.023e-05)	7127
1	113.862	(113.639, 114.085)	6569632	0.0000099	(7.976e-06, 1.191e-05)	7758
0	139.734	(139.460, 140.008	1217664	0.0000106	(9.981e-06, 1.130e-05)	828

6 CONCLUSION

In this article, we proposed a new simulation algorithm to estimate tail quantiles of the steady-state sojourn time in non-preemptive priority queues. Our algorithm is designed based on regenerative simulation, and importance sampling is used to improve efficiency. The importance distribution is optimized by the cross-entropy method. The confidence interval is also constructed. The numerical experiments show that our algorithm obtains significant improvement compared to the benchmarks. Future directions of interest include extending the current framework to queues with other service protocols such as WFQ and DRR, commonly used in computing and telecommunication systems.

REFERENCES

Asmussen, S., and P. W. Glynn. 2007. *Stochastic Simulation: Algorithms and Analysis*, Volume 57. New York: Springer.

Blanchet, J., P. Glynn, and J. C. Liu. 2007. "Fluid heuristics, Lyapunov Bounds and Efficient Importance Sampling for a Heavy-tailed G/G/1 Queue". *Queueing Systems* 57:99–113.

Blanchet, J. H., and M. Mandjes. 2009. "Rare Event Simulation for Queues". In *Rare Event Simulation using Monte Carlo Methods*, edited by B. T. Gerardo Rubino, Chapter V, 87–124. West Sussex: John Wiley & Sons.

Chang, C.-S., P. Heidelberger, S. Juneja, and P. Shahabuddin. 1994. "Effective Bandwidth and Fast Simulation of ATM Intree Networks". *Performance Evaluation* 20(1-3):45–65.

Crane, M. A., and A. J. Lemoine. 1977. *An Introduction to the Regenerative Method for Simulation Analysis*. Berlin: Springer.

De Boer, P.-T., D. P. Kroese, and R. Y. Rubinstein. 2004. "A Fast Cross-Entropy Method for Estimating Buffer Overflows in Queueing Networks". *Management Science* 50(7):883–895.

Dupuis, P., A. D. Sezer, and H. Wang. 2007. "Dynamic Importance Sampling for Queueing Networks". *The Annals of Applied Probability* 17(4):1306–1346.

Dupuis, P., and H. Wang. 2009. "Importance Sampling for Jackson Networks". *Queueing Systems* 62(1):113–157.

Harchol-Balter, M. 2021. "Open Problems in Queueing Theory Inspired by Datacenter Computing". *Queueing Systems* 97(1):3–37.

Hsieh, M.-h., and P. W. Glynn. 2002. "Confidence Regions for Stochastic Approximation Algorithms". In *Proceedings of the 2002 Winter Simulation Conference*, edited by J. L. Snowdon, J. M. Charnes, E. Yücesan, and C.-H. Chen, 370–376. San Diego, CA, USA: Institute of Electrical and Electronics Engineers, Inc.

Huawei 2021. "CloudEngine 12800 and 12800E V200R019C10 Configuration Guide - QoS". https://support.huawei.com/enterprise/en/doc/EDOC1100138142/3ff19e15/congestion-managementl. accessed 29th April 2022.

Iglehart, D. L. 1976. "Simulating Stable Stochastic Systems, VI: Quantile Estimation". *Journal of the ACM (JACM)* 23(2):347–360.

Kella, O., and U. Yechiali. 1985. "Waiting Times in the Non-Preemptive Priority M/M/c Queue". *Stochastic Models* 1(2):257–262.

Mandjes, M., and B. Zwart. 2006. "Large Deviations of Sojourn Times in Processor Sharing Queues". *Queueing Systems* 52(4):237–250.

Parekh, S., and J. Walrand. 1989. "A Quick Simulation Method for Excessive Backlogs in Networks of Queues". *IEEE Transactions on Automatic Control* 34(1):54–66.

Resnick, S. 2019. *A Probability Path*. New York: Springer.

Rubinstein, R. Y., and D. P. Kroese. 2004. *The Cross-Entropy Method: A Unified Approach to Combinatorial Optimization, Monte-Carlo Simulation, and Machine Learning*, Volume 133. New York: Springer.

Setayeshgar, L. 2012. *Large Deviations for a Feed-forward Network & Importance Sampling for a Single Server Priority Queue*. Ph.D. thesis, Division of Applied Mathematics, Brown University, Providence, Rhode Island. https://repository.library.brown.edu/studio/item/bdr:297528/, accessed 29th September 2022.

AUTHOR BIOGRAPHIES

JIN GUANG is a Ph.D. student in the joint training program by Shenzhen Research Institute of Big Data and The Chinese University of Hong Kong, Shenzhen. He earned his bachelor degree in the School of Electronic Science and Engineering at Nanjing University. His research interests include stochastic simulation and queueing theory. His email address is jinguang@link.cuhk.edu.cn.

GUIYU HONG is a Ph.D. student in the School of Data Science at The Chinese University of Hong Kong, Shenzhen. He earned his bachelor degree in the School of Statistics at Renmin University of China. His research interests include stochastic simulation design and online learning methods for queueing systems. His email address is guiyuhong@link.cuhk.edu.cn.

XINYUN CHEN is an Assistant Professor in the School of Data Science at The Chinese University of Hong Kong, Shenzhen. She earned her Ph.D. in Operations Research at Columbia University. Her research interests include applied probability, stochastic simulation, queueing theory and reinforcement learning. Her email address is chenxinyun@cuhk.edu.cn and her website is https://myweb.cuhk.edu.cn/chenxinyun.

XI PENG is a researcher at Theory Lab, Central Research Institute, 2012 Labs, Huawei Technologies Co., Ltd., Hong Kong. She received the Ph.D. degree in Electronic and Computer Engineering from the Hong Kong University of Science and Technology (HKUST) in 2017. Her research interests include system performance evaluation, network simulation, and resource allocation. Her email address is pancy.pengxi@huawei.com.

LI CHEN is a researcher at Theory Lab, Central Research Institute, 2012 Labs, Huawei Technologies Co., Ltd., Hong Kong. He graduated from HKUST with Ph.D. in Computer Science and Engineering in 2018. His work on parallel computing, network transport design, and novel networking architectures has been published in renowned conferences and journals, including SIGCOMM and NSDI. He can be reached at chen.li7@huawei.com.

BO BAI is an Information Theory Scientist and the Director of Theory Lab, Central Research Institute, 2012 Labs, Huawei Technologies Co., Ltd., Hong Kong. His research interests include classic & post-Shannon information theory, B5G/6G mobile communications, and graph informatics. He is an IEEE Senior Member, and has authored more than 130 papers in major IEEE and ACM journals and conferences, 2 book chapters, and 1 textbook. His email address is baibo8@huawei.com.

GONG ZHANG is a Principal researcher at Theory Lab, Central Research Institute, 2012 Labs, Huawei Technologies Co., Ltd., Hong Kong. His research interests include networking architectures and large-scale distributed systems, and communication systems. He had contributed more than 90 patents globally. His email address is nicholas.zhang@huawei.com.