Hindawi
Mathematical Problems in Engineering
Volume 2019, Article ID 3464389, 11 pages
https://doi.org/10.1155/2019/3464389

Research Article

A Fast Simulation Model Based on Lindley's Recursion for the G/G/1/K Queue

J. L. Vazquez-Avila,[1] **R. Sandoval-Arechiga**,[2] **Agustin Perez-Ramirez**,[1] **R. Sanchez-Lara**,[1] **Homero Toral-Cruz**,[3] **and Y. El Hamzaoui**[1]

[1]*Facultad de Ingeniería, Universidad Autónoma del Carmen, Carmen, CAM 24180, Mexico*
[2]*Centro de Investigación, Innovación y Desarrollo en Telecomunicaciones, Universidad Autónoma de Zacatecas, Zacatecas, ZAC, Mexico*
[3]*Department of Sciences and Engineering, University of Quintana Roo, Chetumal, QROO 77019, Mexico*

Correspondence should be addressed to J. L. Vazquez-Avila; jvazquez@pampano.unacar.mx

Received 28 March 2019; Revised 31 May 2019; Accepted 2 June 2019; Published 31 July 2019

Academic Editor: Jean Jacques Loiseau

There are many applications where it is necessary to model queuing systems that involve finite queue size. Most of the models consider traffic with Poisson arrivals and exponentially distributed service times. Unfortunately, when the traffic behavior does not consider Poisson arrivals and exponentially distributed service times, closed-form solutions are not always available or have high mathematical complexity. Based on Lindley's recursion, this paper presents a fast simulation model for an accurate estimation of the performance metrics of G/G/1/K queues. One of the main characteristics of this approach is the support for long-range dependence traffic models. The model can be used to model queuing systems in the same way that a discrete event simulator would do it. This model has a speedup of at least two orders of magnitude concerning implementations in conventional discrete event simulators.

1. Introduction

Traditionally, queuing theory applications are limited to systems with assumptions that derive in closed-form expressions. For example, most of them restrict the tractability of solutions or the tools available to achieve numerical results. Some features that impact tractability are queue size, arrival process, service time distribution, and queue service's policy. In terms of queue size, the queuing systems solutions divided into infinite and finite queue size.

For practical reasons, real-world applications in manufacturing, transportation, communication, networking, and computation systems have finite queues [1, 2]. Moreover, classical works have demonstrated that the queue size has a significant impact on the performance metrics of queuing systems [1, 2].

In the same way, arrival processes can be loosely classified by the decay of its autocorrelation function in ones with Short Range Dependence (SRD), i.e., fast decay, and ones with Long-Range Dependence (LRD) or slow decay [3, 4]. Many applications consider Poisson arrivals, which are SRD and produce solutions with closed-form expressions [1]. However, real systems have an arrival process different from that of Poisson. For example, some applications exhibit Long-Range Dependence (LRD) traffic [3–6]. Therefore, it is desirable to have models for general queuing systems, such as the $G/G/1/K$ queue, which is generally difficult to analyze; e.g., the $G/M/1/K$ queue was studied through approximations that result in high mathematical complexity in [7, 8], while, in [9], asymptotic representations were used for the metrics of the system.

Service time distribution broadly classifies queuing systems in exponentially distributed (Markovian property, i.e., memoryless, M), heavy-tail distributions, among others. In the literature, there are exact solutions and approximations for the $M/G/1/K$ queue [2, 10, 11]. For example, the Poisson Pareto Burst Process (PPBP) was studied by their applications in real broadband traffic modeling [5, 6, 12]. In PPBP burst

arrivals occur as Poisson processes with Pareto distributed length, and each burst is divided into small pieces that will represent the work in the system (packets, entities, users, etc.). Some interesting, theoretical, and simulation results for the PPBP model are presented in [5], where short and long bursts are divided in order to study the impact of each subprocess.

On the other hand, simulations can solve queuing systems with no closed expressions but, unfortunately, have high computation times in many cases. A fast discrete event simulation model for a priority round robin multiplexer based on Lindley-type recursions is presented in [13]. In [14], Lindley-type recursive representations for multiserver tandem queues are presented. The models presented both in [13, 14] have run times faster than those implemented in conventional simulations. This paper, based on Lindley's recursion [15], proposes a fast discrete event simulation (FDES) model for the study of the $G/G/1/K$ queue. The model can accurately estimate the metrics of interest and can be applied in optimization problems [11, 16] even when the traffic model has LRD features. Additionally, the proposed model can easily be used to the performance analysis of queuing systems [11, 16, 17].

The rest of the paper is organized as follows. Section 3 introduces the queueing system model. First, the well known single server model with infinite queue size is presented. Later, based on the single server model for infinite queues, an algorithm for the single server model with finite queue size is presented. The performance analysis results for the $G/G/1/K$ queue with different traffic behavior are presented in Section 4. Finally, Section 5 gives some conclusions.

2. Mathematical Preliminary

This section introduces a brief description of some analytical results for finite queuing systems.

2.1. The M/M/1/K Queue. As mentioned in the previous section, there are few results for the $G/G/1/K$ queueing system. This subsection presents some average results for this kind of systems. Perhaps, the most studied queueing model is $M/M/1/K$. For example, in [10, 17–19], there are results for the $M/M/1/K$ queueing systems. These results are the blocking probability, the expected number of entities in the system, and the average response time. Consider that the customers or entities arrive at the system with an average arrival rate λ and are served with an average service rate μ. The traffic intensity is defined as $\rho = \lambda/\mu$. From [19] we have the following:

The blocking probability

$$p_b = \rho^K \frac{1-\rho}{1-\rho^{K+1}} \tag{1}$$

The effective arrival rate or throughput from the input side is given by

$$Th = (1-p_b)\lambda = \frac{1-\rho^K}{1-\rho^{K+1}}\lambda \tag{2}$$

and the expected number of entities in the system

$$L = \frac{\rho\left[1-(K+1)\rho^K + K\rho^{K+1}\right]}{(1-\rho)(1-\rho^{K+1})} \tag{3}$$

while the expected number of entities in the queue

$$L_q = L - (1-p_0) \tag{4}$$

The average response time that is the expected sojourn time of an entity (in other words, the expected waiting time of an entity given that the entity must wait in the queue) can be derived from Little's theorem and using equations (2) and (3) [19]:

$$W = \frac{\rho^{K+1}(K\rho - K - 1) + \rho}{\lambda(1-\rho)(1-\rho^K)} \tag{5}$$

2.2. The Imbedded-Markov-Chain Method for M/G/1/K Queueing Systems. The method is based on the consideration that the M/G/1/K queueing system can be seen as an imbedded-Markov-chain observing the number of entities left behind upon the departure of an entity [2, 10, 18, 19]. Then we define the single-step transition matrix truncated to $K-1$, as

$$P = \begin{pmatrix} a_0 & a_1 & a_2 & \cdots & 1-\sum_{k=0}^{K-2} a_k \\ a_0 & a_1 & a_2 & \cdots & 1-\sum_{k=0}^{K-2} a_k \\ 0 & a_0 & a_1 & \cdots & 1-\sum_{k=0}^{K-3} a_k \\ 0 & 0 & a_0 & \cdots & 1-\sum_{k=0}^{K-4} a_k \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1-a_0 \end{pmatrix} \tag{6}$$

which implies that the stationary equation is

$$\pi_i = \begin{cases} \pi_0 a_i + \sum_{j=1}^{i+1} \pi_j a_{i-j+1} & (i = 0, 1, 2, \ldots, K-2) \\ 1 - \sum_{j=0}^{K-2} a_j & (i = K-1) \end{cases} \tag{7}$$

where $\{\pi_i\}$ is the set of stationary probabilities at points of departures. The probability of k arrivals during a service time, a_k, is given by

Table 1: Distribution of the number of arrivals during a service time $S = x$.

a_k		
Exponential	Erlang-2	Deterministic
$\frac{\lambda^k \mu}{(\lambda+\mu)^{k+1}}$	$\frac{\lambda^k \mu^2 (k+1)}{(\lambda+\mu)^{k+2}}$	$\frac{\rho^k e^{-\rho}}{k!}$

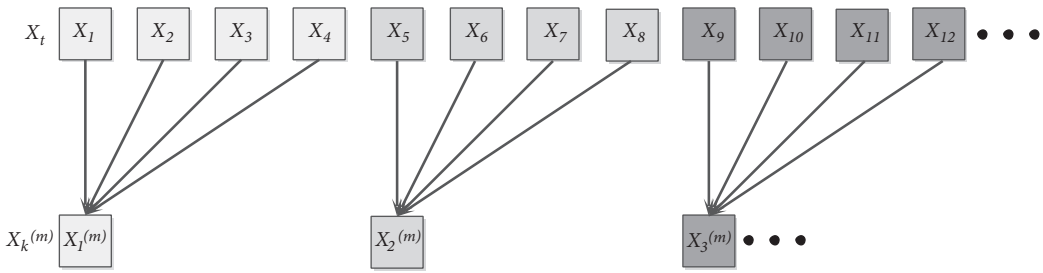

Figure 1: Aggregated processes, $m = 4$.

$$a_k = P\{k \text{ arrivals during a service time } S = x\} = \int_0^\infty \frac{e^{-\lambda x}(\lambda x)^k}{k!} dF_B(x) \quad (8)$$

where $F_B(x)$ represents the distribution of the service time. Table 1 shows results for a_k for distributions of the service time exponential, Erlang-2 and deterministic. The blocking probability depends on the probability that an arrival finds the system full, and in general, the probability distribution of the system size encountered by an arrival (here $\{p_i\}$) will be different from $\{\pi_i\}$ as

$$p_i = \frac{\pi_i}{\pi_0 + \rho} \quad (9)$$

and the blocking probability is given by

$$P_b = p_K = 1 - \frac{1}{\pi_0 + \rho} \quad (10)$$

2.3. Long-Range Dependence and Self-Similarity Basis. Modern investigations of traffic measurements suggest that self-similar processes and Long-Range dependence (LRD) can be applied to the study and accurate modeling of network traffic [20, 21]. A desirable feature of modeling network traffic through self-similar processes is that the correlation structure is expressed in terms of a single parameter called the Hurst index or Hurst parameter. The self-similarity is defined through continuous and discrete stochastic processes [22]. In this work, the discrete self-similarity is used.

Let $X = X_t;\ t \in N$ be a discrete stochastic process or discrete time series with mean μ, variance σ^2, and autocorrelation function $r(k)$, $k \geq 0$. Assume $r(k)$ to be of the form $r(k) \sim k^{2H-2}$ as $k \longrightarrow \infty$, where H is the Hurst index [23]. It is known that stochastic processes with $H < 0.5$ present short range dependence (SRD), while stochastic processes with $H > 0.5$ present LRD. If $H = 0.5$, neither SRD nor LRD are present; i.e., not presenting any dependency. This is the well-known property of white Gaussian noise [22]. When considering discrete time series, the definition of self-similarity is given in terms of the aggregated processes, as is shown in (11) [24]:

$$X_k^m = \left(X_k^m;\ k \in \mathbb{N}\right) \quad (11)$$

Figure 1 illustrates an aggregated process, where m represents the aggregation level and $X_k^{(m)}$ is obtained by averaging the original series X_t over nonoverlapping blocks of size m, and each term $X_k^{(m)}$ is given by

$$X_k^{(m)} = \frac{1}{m} \sum_{t=(k-1)m+1}^{km} X_t; \quad k \in \mathbb{N} \quad (12)$$

Then it is said that X_t is self-similar with Hurst parameter $(0 < H < 1)$ if [25]

$$X_k^{(m)} \stackrel{d}{=} m^{H-1} X_t \quad (13)$$

where $\stackrel{d}{=}$ denotes equality in distribution.

The variance of the aggregated time series is defined by equation (14) as follows [24]:

$$\text{var}\left(X_k^{(m)}\right) = m^{2H-2} \text{var}\left(X_t\right) \quad (14)$$

The plot $\log[\text{var}(X_k^{(m)})]$ versus $\log(m)$ is known as Variance Plot. It is a straight line of slope $2H - 2$ for self-similar processes. This plot is the basis of the variance based estimator of the Hurst parameter [22]. Several methods have been developed, then in order to estimate the Hurst parameter [22, 26]; in this work, the periodogram method is used [26].

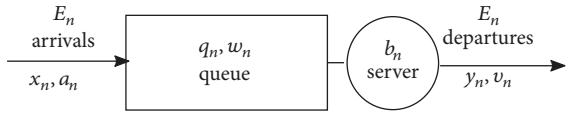

Figure 2: Model of a queueing system.

3. The Fast Discrete Event Simulation Model

This approach is based on difference equations based on Lindley's recursion, in order to model the waiting and service times of a queuing system. Performance metrics such as mean service time and mean queue size can be calculated numerically from the data processed by the difference equations. This process is the same as any simulation software executes. However, this approach takes off all the unnecessary details needed for a full-scale simulation. In the Fast Discrete Event Simulation (FDES) approach we call every customer an entity because it can be a client, packet, call, etc. depending on the system modeled. However, FDES only takes into consideration the arrival and service times for every entity that comes into the system. Then FDES takes these two quantities and processes them into the recursive equations to obtain the waiting and departure times for every entity.

3.1. The Single Server Model: Infinite Queue. The model consists of a $G/G/1$ single queue server with an infinite queue and operates according to the First-In-First-Out (FIFO) queuing discipline, as shown in Figure 2. Once an entity arrives into the system and the server is free, the entity is attended. If the entity finds the server busy, it is placed into the queue and waits to be served. In this model, the events of the system are represented by the arrivals and departures of the entities.

Let $\{a_n\}$, $\{b_n\}$, and $\{v_n\}$ denote the sequences that represent random variables of the arrival instant to the system, service time, and departure instant, respectively, of the nth entity, E_n. Here the entity may be interpreted as a packet or a customer. Furthermore, let $\{x_n\}$ ($\{y_n\}$) denote the sequence that represents the interarrival (interdeparture) time between the arrivals (departures) of E_{n-1}th and E_n. Entities arrive to the system with an average arrival rate λ and are served with an average service rate μ. The traffic intensity is defined as $\rho = \lambda/\mu$.

Figure 3 shows a timing diagram for an infinite queuing system. Observing the figure, the arrival of the nth entity occurs at the a_n time epoch, and this entity might have to wait for a random time w_n before being served for a random service time b_n. When the nth entity is served, it departs from the system at the v_n time epoch. Initially, the system is considered to be empty. The random sequences that describe the system-time behavior of the nth entity can be obtained.

We define the random variable u_{n-1} as

$$u_{n-1} = b_{n-1} - x_n \tag{15}$$

where u_{n-1} represents a stability variable since for a stable system the expectation of u_{n-1} must be negative. This is $E[u_{n-1}] < 0$. From Figure 3, we can see that the waiting time can be expressed in terms of the stability variable and the waiting time of the previous entity, as is shown in equation (16):

$$w_n = w_{n-1} + b_{n-1} - x_n \tag{16}$$

A general representation for the waiting time that considers those entities that do not wait in the queue because the server is available results in Lindley's recursion [1, 15]; then

$$w_n = \begin{cases} w_{n-1} + u_{n-1} & \text{if } w_{n-1} + u_{n-1} > 0 \\ 0 & \text{if } w_{n-1} + u_{n-1} \leq 0 \end{cases} \tag{17}$$

Equation (17) can also be expressed as follows:

$$w_n = \max\left[0, w_{n-1} + u_{n-1}\right] \tag{18}$$

The departure instants from the system for the nth entity can be expressed as

$$v_n = a_n + w_n + b_n \tag{19}$$

and the interdeparture time for the nth entity is given by

$$y_n = v_n - v_{n-1} \tag{20}$$

From the bottom of Figure 3, the number of entities found in the system immediately prior to the arrival of E_n (or the number of entities in the queue upon the arrival of E_n), denoted by q_n, can be determined as

$$q_n = q_{n-1} - \varphi_n + 1 \tag{21}$$

where φ_n is the number of entities served between the arrivals of E_{n-1} and E_n. In order to determine q_n, we need first to find φ_n. We propose Algorithm 1 in order to find the number of entities served between two consecutive arrivals.

The description of Algorithm 1 is as follows: First, the inputs of the algorithm are defined. *LastEntity* is the index of the last entity served until the arrival of entity E_{n-1}. Initially, it is considered that $LastEntity = 0$. The algorithm also requires the nth arrival epoch a_n and the departure epochs previous to v_n. In line 1, the **for** loop sequentially searches the entities that have departed between the arrivals of E_{n-1} and E_n. In line 2, a comparison between a_n and v_{n-j} is necessary in order to know which of the entities have departed before the arrival of entity E_n. If a departing entity is found, the number of entities between E_{n-1} and E_n, φ_n can be calculated (line 3), and the auxiliary variable is updated (line 4). Then, the loop is broken and the algorithm finishes.

3.2. The Single Server Model: Finite Queue. In the previous section, we presented a $G/G/1$ system model for a single server with an infinite queue. Based on this model, we propose a $G/G/1/K$ model. In this queuing system, there is

Input: The last entity attended until the arrival of entity E_{n-1} $LastEntity$, at the begin $LastEntity = 0$; the n-th arrival epoch a_n; the departures epochs previous to v_n.
Output: Number of entities served between the arrival of E_{n-1} and E_n, φ_n.
1: **for** $j = 1$ **to** $n - LastEntity - 1$ **do**
2: **if** $v_{n-j} \leq a_n$ **then**
3: $\varphi_n = n - LastEntity - j$
4: $LastEntity = LastEntity + \varphi_n$
5: brake the for loop.
6: **end if**
7: **end for**

Algorithm 1: Number of entities served between the arrivals of E_{n-1} and E_n, φ_n.

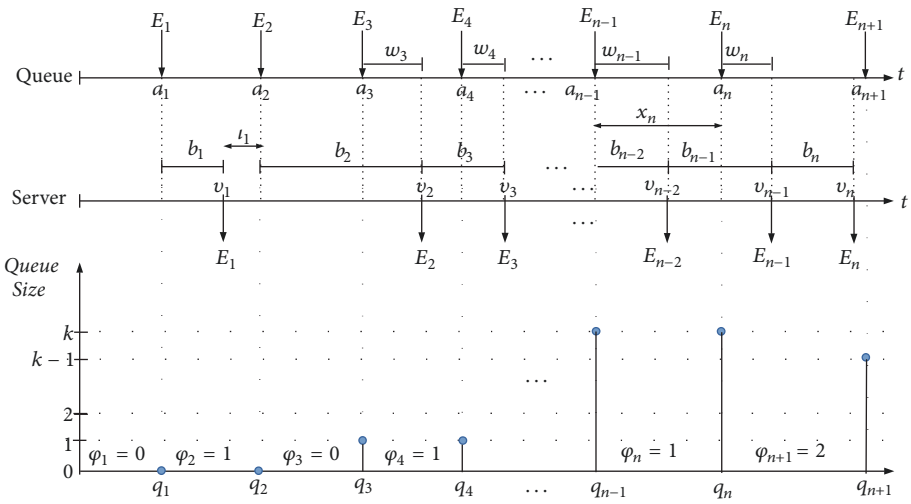

Figure 3: Time diagram for the queueing system and the queue size.

a finite amount of waiting room K, which includes the room for the server (according to Kendall's notation).

The model consists of a $G/G/1/K$ single queue server with a finite queue and operates according to the First-In-First-Out (FIFO) queuing discipline, as shown in Figure 2. The entities (E_n) arrive to the system with an arbitrary distribution. Once an entity arrives to the system, one of the following cases occurs: (1) if the server is empty, an entity will be immediately served for a time that depends on an arbitrary distribution; (2) if the server is busy and there is at least one place in the queue the entity will wait in the queue until all the entities that arrived previously have been served; (3) if the queue is full, the entity will be blocked and cleared from the system.

In order to estimate the performance metrics of the $G/G/1/K$ queuing system, we propose Algorithm 2. In Algorithm 2, we first define the input and output variables. N_E is the number of entities to consider in the model. Initially, the system is empty. Line 1 shows the initialization of the variables. In line 2, a loop **for** is utilized to evaluate the different variables of interest for each entity under consideration. In line 3, the indexes for the entities are updated so that the expression obtained for the $G/G/1$ queuing system in the previous section can be utilized without considering the blocked entities. When an entity arrives into the system, the number of entities in the queue q_i is calculated (line 4). Notice that q_i in (21) represents the number of entities found in the system immediately prior to the arrival of E_i, so the distribution of the system size can be determined. Then, we ask for the availability of the queue (line 5); if there is at least one place in the queue, then the entity is accepted to be served and the waiting time in the queue and the departure epoch of the entity is calculated (lines 6 and 7). Index j is updated only for nonblocked entities (line 8); if the queue is full upon the arrival of entity i, then the entity is blocked and a counter to determine the number of blocked entities is increased (line 10).

The blocking probability P_K, that is, the probability that an arriving entity finds the queue full, can be estimated through the distribution of q_n or simply by the relation of $BlockedEntity$ and N_E. The average waiting time can be estimated as follows:

$$\overline{w}_{N_E} = \frac{1}{N_E} \sum_{i=1}^{N_E} w_i \tag{22}$$

Require: $\{a_i\}$, $\{b_i\}$, K, the number of entities to simulate N_E.
Ensure: $\{w_i\}$, $\{v_i\}$, $\{q_i\}$, $BlockedEntity$.
1: **INITIALIZE**: $w_1 = 0$, $v_1 = a_1 + b_1$, $q_1 = 0$, $j = 2$, the number of blocked entities $BlockedEntity = 0$.
2: **for** each entity $i \in [2 : N_E]$ **do**
3: Set $a_j = a_i$, $b_j = b_i$.
4: Use Algorithm 1 and (21) to find q_i.
5: **if** $q_i \leq K$ **then**
6: Use (18) and (19) to find w_j and v_j, respectively.
7: $j = j + 1$.
8: **else**
9: $BlockedEntity = BlockedEntity + 1$.
10: **end if**
11: **end for**
12: $P_K = \dfrac{BlockedEntity}{N_E}$

Algorithm 2: The $G/G/1/K$ queuing system.

4. Results

This section presents numerical results for the performance analysis of the $G/G/1/K$ queue. Additionally, the FDES runtime is analyzed and an estimation of the time complexity is presented.

4.1. $G/G/1/K$ Performance Analysis Results. Exact (or theoretical) results are obtained as follows (unless otherwise stated): the $M/G/1/K$ queue is solved through the method of embedded Markov chains; the $D/M/1/K$ (D refers to deterministic) queue estimation was generated according to the numerical approximation in the computer program McQueue [27]; the $E_2/M/1/K$ was generated from the numerical approximation presented in [7]. The average service rate is fixed to 1 second. For the simulation and the FDES model, independent and identically distributed (i.i.d.) Gamma random variables were used to model nonexponentials, such as the 2-stage Erlang, labeled as E_2. In the representation of $G/G/1/K$-x, x = {t, s, f} refers to theoretical (t), simulation (s), and FDES model (f). The simulations were implemented in SimEvents of Mathworks Matlab and in OMNET++. Both simulations and the FDES model were fed with the same traces. An Intel core i7 2600 @ 3.4 GHz computer with 8 GB of RAM running on a 64-bit Windows 10 was used to obtain model outcomes.

Table 2 shows results for the average number in the system (L), the average number in the queue (L_q), the probability that there is no customer in the system (P_0), the blocking probability (P_K), and the runtime. It can be observed that the c FDES model accurately estimates the metrics with respect to the theoretical results and the results of the simulation and the FDES model are exactly the same since the models are fed with the same traces. However, the FDES model is two orders of magnitude faster than the simulation implemented in SimEvents.

Likewise, Table 3 shows results for the $G/G/1/K$ queue considering traffic with different burstiness. Markov Modulated Poisson Processes (MMPP) as arrival processes were considered in order to simulate LRD traffic. Exponential service times were considered and the average service time is fixed to 1 second. As the table shows, both simulation and FDES metrics agree. Although Markov Modulated Poisson Processes (MMPP) cannot exhibit LDR features, in this paper they are used as an emulation of LDR such as in [19, 28]. The purpose of the aforementioned is to specify different sources of traffic. The Hurst parameter (H) was estimated using the periodogram method [29, 30]. Markov Modulated Poisson Processes with two states were considered. The state transition rates r_1 and r_2, for states 1 and 2, were 0.001 and 0.01 transitions/seconds, respectively. The arrival rate in state 1 is $\lambda_1 = \lambda B$, and the arrival rate in state 2 is $\lambda_2 = ((r_1 + r_2)\lambda - B\lambda r_1)/r_2$, where B is the burstiness. Notice that $B = 1$ corresponds to a simple Poisson process (nonmodulated).

Figures 4 and 5 show the average waiting time in the queue versus the traffic intensity and the blocking probability for diverse values of the capacity K, respectively. Different queuing systems and 30 realizations of the fast simulation are considered. Results show that the FDES model perfectly matches the theoretical outcomes; this is an expected result since the FDES model is a fast simulation based on Lindley's formula.

Table 4 shows results for the FDES and simulation considering Pareto Modulated Poisson Processes (PMPP), labeled as $PMPP/M/1/K$, and pure Pareto arrivals processes, labeled as $P/M/1/K$, both of them considering exponential service times. For $1 < \alpha < 2$ the mean of the Pareto distribution is $E\{X\} = \delta\alpha/(\alpha - 1)$ and the variance is infinite. In the Pareto distribution, α denotes the thickness of the tail of the distribution and δ the minimum value of the random variable X [5, 12]. On-Off Pareto Modulate Poisson Processes can be used to model LDR traffic [31]. In the same way, pure Pareto arrivals exhibit self-similar properties. The Hurst parameter can be calculated as $H = (3-\alpha)/2$, where α denotes the thickness of the tail of the Pareto distribution [31]. The state transition rates r_{on} and r_{off}, for states *on* and *off*, were 0.001 and 0.01 transitions/seconds, respectively. Besides, we consider that the thickness parameter is $\alpha_{on} = \alpha_{off}$. As can

Table 2: Average results for G/G/1/K queueing systems, $\rho = 0.6$, $K = 4$, and $N_E = 10^5$.

Results (Average)	L	L_q	P_0	P_K	Runtime (sec.)
$M/E_2/1/K$-t	1.0246	0.4468	0.4223	0.0371	2.4538e-04
$M/E_2/1/K$-s	1.0263	0.4473	0.4245	0.0363	9.4462
$M/E_2/1/K$-f	1.0263	0.4473	0.4245	0.0363	0.0193
$D/M/1/K$-t	0.8544	0.2574	0.6844	0.0051	2.3045e-04
$D/M/1/K$-s	0.8547	0.2585	0.6802	0.0050	9.3824
$D/M/1/K$-f	0.8547	0.2585	0.6802	0.0050	0.0190

t: theoretical; s: simulation; f: FDES.

Table 3: Average results for G/G/1/K queueing systems, $\rho = 0.6$, $K = 4$, and $N_E = 10^5$.

Results (Average)	B	H	L	$\overline{w}_{N_E}$	P_0	P_K
$M/M/1/K$-t	1	0.5	1.0784	0.9044	0.4337	0.0562
$M/M/1/K$-s	1	0.5146	1.0813	0.9091	0.4306	0.0571
$M/M/1/K$-f	1	0.5146	1.0813	0.9091	0.4306	0.0571
$MMPP/M/1/K$-s	1	0.5109	1.0806	0.9106	0.4311	0.0577
$MMPP/M/1/K$-f	1	0.5109	1.0806	0.9106	0.4311	0.0577
$MMPP/M/1/K$-s	5	0.6259	3.4982	2.5002	0.0117	0.6629
$MMPP/M/1/K$-f	5	0.6259	3.4982	2.5002	0.0117	0.6629
$MMPP/M/1/K$-s	10	0.7618	3.7941	2.7763	0.0018	0.8321
$MMPP/M/1/K$-f	10	0.7618	3.7941	2.7763	0.0018	0.8321

t: theoretical; s: simulation; f: FDES.

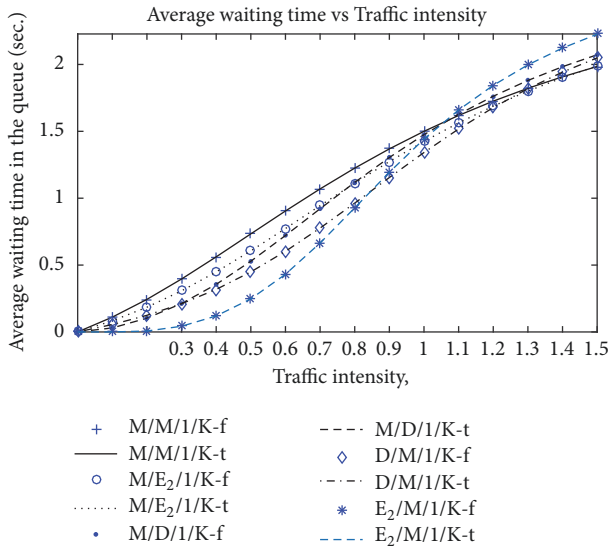

Figure 4: Average waiting time in the queue for the $G/G/1/K$ queue.

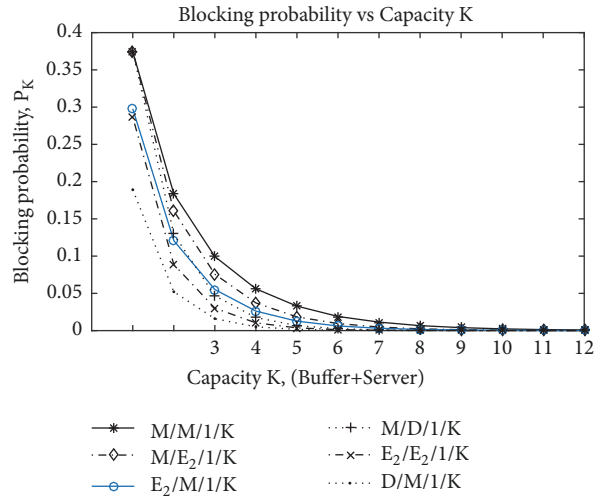

Figure 5: Blocking probability for different values of K.

be observed, from Table 4, the FDES model and simulation agree perfectly. Furthermore, for high Hurst parameter, the worst performance is shown, but the pure Pareto traffic model presents a worse performance than the PMPP.

Figures 6 and 7 show the performance for the pure Pareto arrivals queue, $P/M/1/K$, considering the FDES model. In both figures, the well-known $M/M/1/K$ model is used as a reference for comparison purposes, and, theoretically, the exponential arrival processes have a Hurst parameter of 0.5. Figure 6 shows the blocking probability versus the buffer capacity for different values of the Hurst parameter. As the capacity is increased, the blocking probability is reduced, which is expected since the buffer size increases. Besides, notice that for a higher Hurst parameter we obtain a higher blocking probability. This last result is also expected since the Hurst parameter represents a measure of the self-similarity and the Pareto distribution is heave tailed with self-similarity behavior. On the other hand, Figure 7 shows the average waiting time in the queue versus the buffer capacity. Notice that, as the capacity increases, the average waiting time in

Table 4: Average results for G/G/1/K queueing systems, $\rho = 0.6$, $K = 4$, and $N_E = 10^5$.

Results (Average)	α	H	L	$\overline{w}_{N_E}$	P_0	P_K
$PMPP/M/1/K$-s	1.2	0.9	1.0719	0.8962	0.4340	0.0554
$PMPP/M/1/K$-f	1.2	0.9	1.0719	0.8962	0.4340	0.0554
$PMPP/M/1/K$-s	1.8	0.6	1.0698	0.8884	0.4386	0.0542
$PMPP/M/1/K$-f	1.8	0.6	1.0698	0.8884	0.4386	0.0542
$P/M/1/K$-s	1.2	0.9	1.1875	1.6893	0.1525	0.2646
$P/M/1/K$-f	1.2	0.9	1.1875	1.6893	0.1525	0.2646
$P/M/1/K$-s	1.8	0.6	1.0631	0.8385	0.4715	0.0363
$P/M/1/K$-f	1.8	0.6	1.0631	0.8385	0.4715	0.0363

s: simulation; f: FDES.

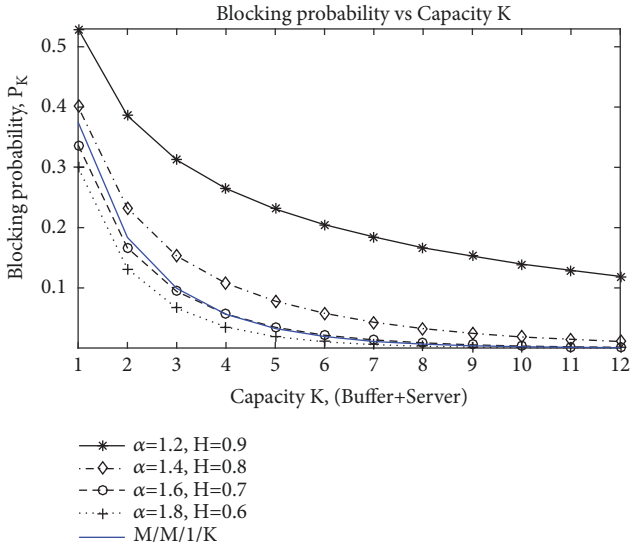

Figure 6: Blocking probability for different values of K for the $P/M/1/K$ queue model.

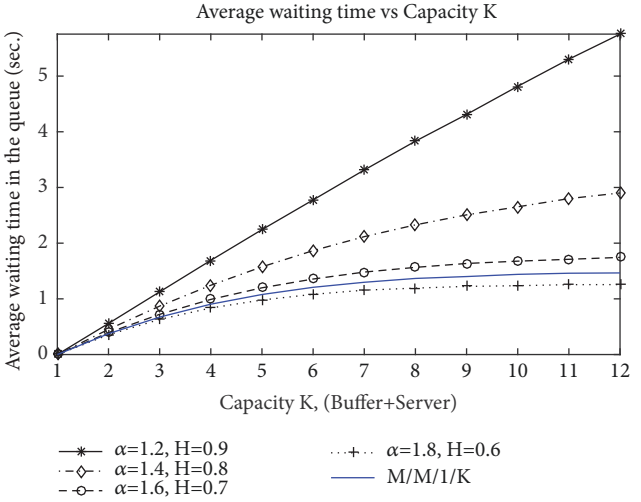

Figure 7: Average waiting time in the queue for different values of K for the $P/M/1/K$ queue model.

the queue increases too. Again, the worst case for the waiting time occurs for a high Hurst parameter. Both in Figures 6 and 7 the $M/M/1/K$ keeps below the curves that have values of the Hurst parameter greater than 0.5, while those curves that are near to that value of H have a performance similar to the $M/M/1/K$ model. This kind of performance analysis can be accomplished through the utilization of the FDES model.

Other interesting results can be observed when analyzing a type of process similar to the PPBP model. Consider that, in PPBP, burst arrivals occur as Poisson processes with Pareto distributed length. In PPBP a burst is divided into small pieces at a rate r. These kinds of systems are used to model real traffic where the buffer size is fixed and the service time is constant [5, 6]. The methodology presented in this work does not represent an exact PPBP model since Algorithms 1 and 2 are not designed to handle the partition of bursts into small entities. However, the FDES queue model could study the performance analysis of the $M/P/1/K$ model. In the $M/P/1/K$ model burst arrivals occur as Poisson processes, the service times are Pareto distributed random variables, there is a single server, and the queue size is K. The $M/P/1/K$ model and the PPBP model are similar in the sense that the burst duration (for PPBP) and the service time (for $M/P/1/K$) are Pareto distributed and both of them represent the amount of work that the server most processes.

Figures 8 and 9 show the performance for the $M/P/1/K$ queue considering the FDES model. As before, there are some results for different values of the Hurst parameter. Results for the Pareto and the truncated Pareto distribution (labeled as Tr) are presented. The truncated Pareto distribution is considered to limit the maximum duration of the service time. For the case of a PPBP model, the truncated Pareto distribution limits the maximum bust duration. In the case of truncated Pareto service times, a limit of the service time of 1000 was considered. Additionally, a mean service time of 1 second and a traffic intensity (ρ) of 0.6 were considered. Figure 8 shows the blocking probability versus the buffer size for different values of the Hurst parameter. As the buffer size is increased the blocking probability is decreased. When the Hurst parameter is increased the blocking probability is increased too. Long service time can occur due to the infinite variance of Pareto distribution, which contributes to system instability. Indeed, as H increases the performance is

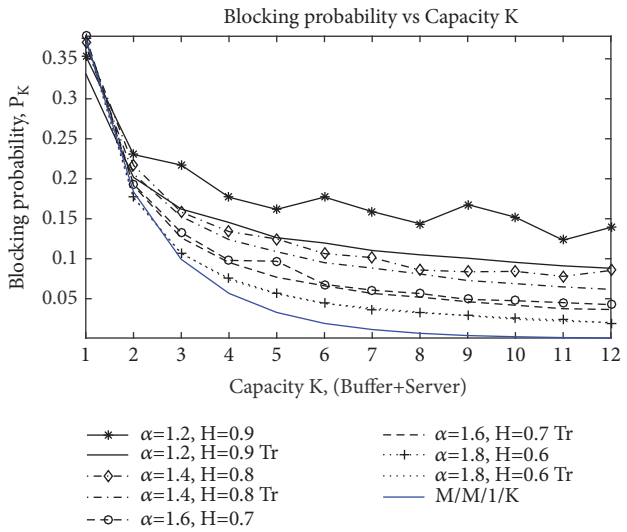

Figure 8: Blocking probability for different values of K for the $M/P/1/K$ queue model.

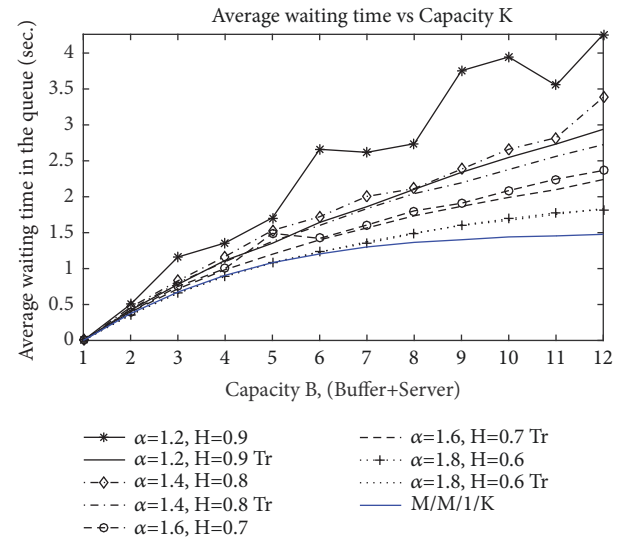

Figure 9: Average waiting time in the queue for different values of K for the $M/P/1/K$ queue model.

worst and, for each realization of the simulation, there will be different results. On the other hand, when a truncated Pareto distribution is considered, the system tends to stability, which is expected since the system does not consider long service times [5]. A similar result is presented in Figure 9, where the waiting time in the queue versus the buffer size is analyzed. Notice that for high values of H the waiting time increases and the system tends to the instability. However, when the service time is limited, the system tends to stability. Of course, it is necessary to determine the limit value, such as what was done in [5], in order to ensure the stability of the system.

4.2. Runtime. Finally, the average runtime for the different models was estimated and was shown in Figure 10. For these results, the following assumptions were considered: the traffic intensity is fixed to 0.6; the number of simulated entities was $N_E = \{10^1, 10^2, 10^3, 10^4, 10^5, 10^6, 10^7\}$; and the considered models are the FDES model, the model implemented in OMNET++, and the model implemented in SimEvents. Although the theoretical model implemented with imbedded Markov chains does not depend on the number of simulated entities, it was also considered for the runtime results. The runtime was averaged over 30 realizations. Figure 10 shows that the runtime for the FDES model is at least one order of magnitude faster than the model implemented in OMNET++ and at least two orders of magnitude faster than the model implemented in SimEvents.

On the other hand, Figure 11 shows results for the runtime as a function of the number of simulated entities and the buffer capacity K. Again, the traffic intensity is fixed to 0.6 and the number of simulated entities was $N_E = \{10^1, 10^2, 10^3, 10^4, 10^5, 10^6, 10^7, 10^8\}$. The runtime for different queue models, using the FDES model, was analyzed: the $M/P/1/K$, $M/M/1/K$, $M/P/1/Kn$, and $M/M/1/Kn$. The label n in Figure 11 corresponds to simulations that do not consider the runtime of the generated traffic sequences. From Figure 11, it can be observed that buffer size does not have an impact over the runtime, at least for moderated values of the number of simulated entities. Nevertheless, the runtime is considerably affected by the number of simulated entities, as can be observed in Figure 11. Furthermore, the runtime of the FDES is not considerably affected by the implemented model, as can be shown in Figure 11. Data from Figure 11 has been processed utilizing genetic programming to best fit the curve [32]. The worst cases represented by $M/P/1/K$ and $M/M/1/Kn$ have been considered. The resulting fitting model for the runtime is approximated as a polynomial expression given as $1.57\text{e}{-7}N_E + (5.68\text{e}{-16} - 5.24\text{e}{-16}/K)N_E^2$ and $(1.43\text{e}-7 - 4.77\text{e}-8/K)N_E + 2.7\text{e}-16N_E^2$, for $M/P/1/K$ and $M/M/1/Kn$, respectively. The R-squared goodness of the polynomial fit was 0.9999 in both cases, which indicates a good fit to the curves. The dominant term in both expressions is N_E^2, but it is attenuated by a coefficient of the order of 10^{-16}. Therefore, considering that the maximum value of N_E in the simulation was 10^8, the time complexity could be approximated to N_E. Moreover, Algorithm 1 has a computational complexity of $O(K)$, whereas the computational complexity for Algorithm 2 is $O(KN_E)$ for $K \geq 1$. When $K << N_E$ the computational complexity is approximately $O(N_E)$, which corresponds to the aforementioned time complexity. Therefore, since the computational complexity is $O(N_E)$, the FDES model is a good option to study the performance analysis of $G/G/1/K$ queuing systems.

5. Conclusions

Based on Lindley's recursion, a fast simulation model for the performance analysis of the $G/G/1/K$ queue was presented. Different traffic sources can be considered including LRD traffic. The model can accurately estimate the main metrics of the queuing system quickly compared with the models implemented in OMNET++ and SimEvents of Mathworks.

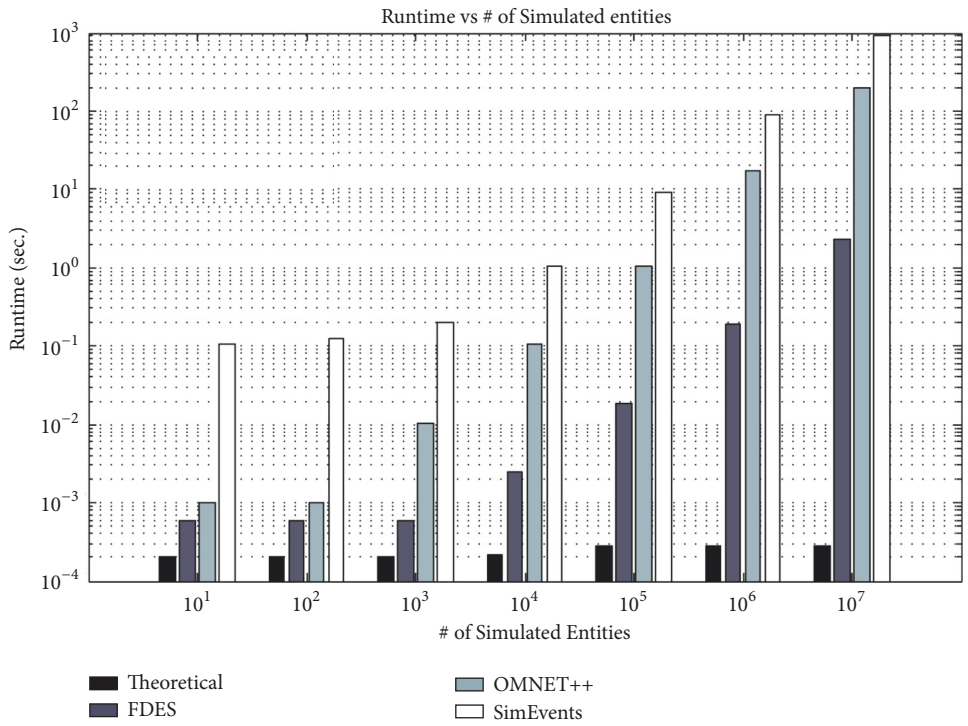

Figure 10: Runtime comparison for different implementations of the $G/G/1/K$ queue.

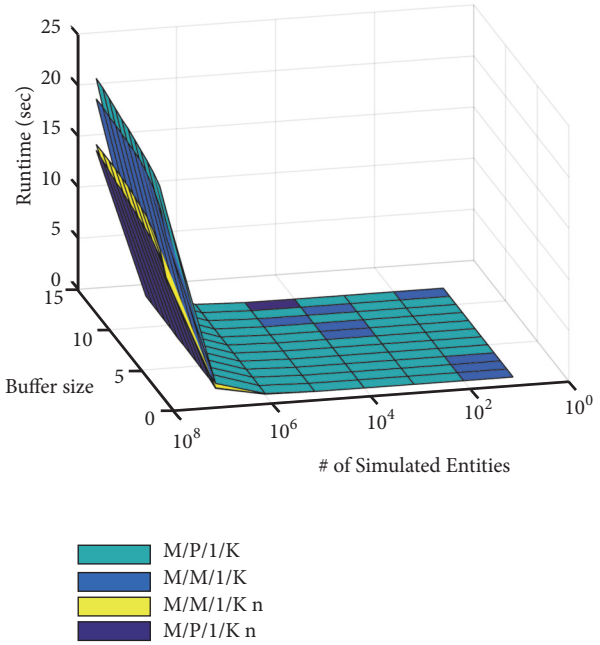

Figure 11: Runtime comparison for different values of K and N_E considering the FDES model.

The model has a speedup of at least two orders of magnitude with respect to implementations in the aforementioned discrete-event simulators. Our model can be exploited in the performance analysis of $G/G/1/K$ queuing systems embedded in optimization problems.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] L. Kleinrock, *Theory, Volume 1, Queueing Systems*, Wiley-Interscience, 1975.

[2] D. Gross, *Fundamentals of Queueing Theory*, John Wiley & Sons, 2008.

[3] M. Yu and M. Zhou, "A model reduction method for traffic described by MMPP with unknown rate limit," *IEEE Communications Letters*, vol. 10, no. 4, pp. 302–304, 2006.

[4] P. Bogdan and R. Marculescu, "Non-stationary traffic analysis and its implications on multicore platform design," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 30, no. 4, pp. 508–519, 2011.

[5] R. G. Addie, T. D. Neame, and M. Zukerman, "Performance evaluation of a queue fed by a Poisson Pareto burst process," *Computer Networks*, vol. 40, no. 3, pp. 377–397, 2002.

[6] C. Xing, R. G. Addie, Y. Peng et al., "Resource provisioning for a multi-layered network," *IEEE Access*, vol. 7, pp. 226–245, 2019.

[7] A. Brandwajn and T. Begin, "A recurrent solution of ph/m/c/n-like and ph/m/c-like queues," *Journal of Applied Probability*, vol. 49, no. 1, pp. 84–99, 2012.

[8] K.-H. Wang, C.-C. Kuo, and W. Pearn, "A recursive method for the f-policy g/m/1/k queueing system with an exponential

startup time," *Applied Mathematical Modelling*, vol. 32, no. 6, pp. 958–970, 2008.

[9] A. Baiocchi, "Asymptotic behaviour of the loss probability of the *M/G/1/K* and *G/M/1/K* queues," *Queueing Systems*, vol. 10, no. 3, pp. 235–247, 1992.

[10] J. M. Smith, "Properties and performance modelling of finite buffer m/g/1/k networks," *Computers & Operations Research*, vol. 38, no. 4, pp. 740–754, 2011.

[11] L. Tang, H. Sheng Xi, J. Zhu, and B. Qun Yin, "Modeling and optimization of m/g/1-type queueing networks: An efficient sensitivity analysis approach," *Mathematical Problems in Engineering*, vol. 2010, Article ID 130319, 20 pages, 2010.

[12] R. G. Addie, M. Zukerman, and T. D. Neame, "Broadband traffic modeling: Simple solutions to hard problems," *IEEE Communications Magazine*, vol. 36, no. 8, pp. 88–95, 1998.

[13] J. Vazquez-Avila, R. Sandoval-Arechiga, and R. Parra-Michel, "A fast discrete event simulation model for queueing network systems," in *Proceedings of the 8th International Conference on Simulation Tools and Techniques*, Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, pp. 302–304, Athens, Greece, August 2015.

[14] W. Kin and V. Chan, "Generalized lindley-type recursive representations for multiserver tandem queues with blocking," *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, vol. 20, no. 4, p. 21, 2010.

[15] D. V. Lindley, "The theory of queues with a single server," *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 48, no. 2, pp. 277–289, 1952.

[16] Z. Liu, W. Deng, and G. Chen, "Analysis of the optimal resource allocation for a tandem queueing system," in *Mathematical Problems in Engineering*, vol. 2017, p. 10, 2017.

[17] F. R. B. Cruz, M. A. C. Almeida, M. F. S. V. D. Angelo, and T. van Woensel, "Traffic intensity estimation in finite markovian queueing systems," in *Mathematical Problems in Engineering*, vol. 2018, p. 15, 2018.

[18] J. MacGregor Smith, "Optimal design and performance modelling of *M/G/1/K* queueing systems," *Mathematical and Computer Modelling*, vol. 39, no. 9-10, pp. 1049–1081, 2004.

[19] G. Dattatreya, *Performance Analysis of Queuing and Computer Networks*, Crc Press, 2008.

[20] H. Toral-Cruz, A.-S. K. Pathan, and J. C. R. Pacheco, *Accurate Modeling of VoIPTraffic in Modern Communication*, Institution of Engineering and Technology, 2019, https://digital-library.theiet.org/content/books/10.1049/pbpc018e_ch7.

[21] H. Toral-Cruz, A. K. Pathan, and J. C. R. Pacheco, "Accurate modeling of voip traffic qos parameters in current and future networks with multifractal and markov models," *Mathematical and Computer Modelling*, vol. 57, no. 11-12, pp. 2832–2845, 2013.

[22] L. E. Vargas, D. T. Roman, and H. T. Cruz, "A study of wavelet analysis and data extraction from second-order self-similar time series," in *Mathematical Problems in Engineering*, vol. 2013, p. 14, 2013.

[23] J. Gao, Y. Cao, W. W. Tung, and J. Hu, *Multiscale Analysis of Complex Time Series: Integration of Chaos and Random Fractal Theory, and beyond*, John Wiley & Sons, 2007.

[24] B. Tsybakov and N. D. Georganas, "Self-similar processes in communications networks," *IEEE Transactions on Information Theory*, vol. 44, no. 5, pp. 1713–1725, 1998.

[25] I. W. C. Lee and A. O. Fapojuwo, "Stochastic processes for computer network traffic modeling," *Computer Communications*, vol. 29, no. 1, pp. 1–23, 2005.

[26] J. Ramírez-Pacheco, D. Torres-Román, H. Toral-Cruz, and L. E. Vargas, "High-performance tool for the test of long-memory and self-similarity," *Simulation Technologies in Networking and Communications: Selecting the Best Tool for the Test*, pp. 93–114, 2014.

[27] M. Mandjes and H. Tijms, *Mcqueue*, Department of Econometrics and Operations Research, VrijeUniversity, Amsterdam, The Netherlands, 2005.

[28] L. Muscariello, M. Meillia, M. Meo, M. Marsan, and R. Cigno, "An MMPP-based hierarchical model of Internet traffic," in *Proceedings of the 2004 IEEE International Conference on Communications*, vol. 4, pp. 2143–2147, Paris, France, June 2004.

[29] Q. Yu, Y. Mao, T. Wang, and F. Wu, "Hurst parameter estimation and characteristics analysis of aggregate wireless lan traffic," in *Proceedings of the International Conference on Communications, Circuits and Systems*, vol. 1, pp. 339–345, 2005.

[30] O. Rose, *Estimation of The Hurst Parameter of Long-Range Dependent Time Series*, University of Wurzburg, Institute of Computer Science Research Report Series, February 1996.

[31] T. Le-Ngoc and S. N. Subramanian, "Pareto-modulated Poisson process (PMPP) model for long-range dependent traffic," *Computer Communications*, vol. 23, no. 2, pp. 123–132, 2000.

[32] M. Schmidt and H. Lipson, "Distilling free-form natural laws from experimental data," *Science*, vol. 324, no. 5923, pp. 81–85, 2009.