APPLICATIONES MATHEMATICAE
Online First version

Katsiaryna Kosarava

ACCURACY ASSESSMENT OF ANALYTICAL APPROXIMATIONS IN RETRIAL QUEUEING NETWORKS WITH IMPATIENT CUSTOMERS

Abstract. The study examines the accuracy of analytical results for a call center model represented as a retrial queueing network with impatient customers. A discrete-event simulation algorithm was developed that explicitly incorporates the structural specifics of the model, including a loss-type distributor, retrial orbit, and finite-capacity queues. Through simulation experiments, we identified the optimal number of model replications that provide sufficiently accurate approximations of the mean values of the network performance characteristics while maintaining an acceptable computational time. To enable an adequate comparison between the analytical and simulation results, the times required for both models to reach the stationary regime were analyzed. Subsequently, the accuracy of the analytical results was evaluated for the selected performance characteristics of the call center model over the stationary interval of the network. Comparative analysis revealed close agreement between simulation and analytical results in steady state, particularly under heavy load conditions. These results confirm that the asymptotic approach, based on the diffusion approximation and Fokker–Planck–Kolmogorov equation, provides accurate predictions under high load.

1. Introduction. The rapid advancement of scientific research and the digital economy has triggered exponential growth in the volume of data being collected, processed, and transmitted. Modern integrated systems – encompassing cloud computing, artificial intelligence, the Internet of Things (IoT),

2020 *Mathematics Subject Classification*: Primary 60K25; Secondary 68M20.
Key words and phrases: queueing theory, retrial system, simulation modeling.
Received 22 September 2025; revised 2 December 2025.
Published online 3 August 2026.

DOI: 10.4064/am2593-2-2026

and next-generation 5G/6G networks – have created a new paradigm of distributed computing and intelligent information infrastructures. To enhance the efficiency and reliability of such complex systems, mathematical modeling remains a cornerstone during both the design and analytical phases.

In this context, queueing network (QN) theory has been the subject of sustained scholarly interest due to its broad applicability across diverse domains, particularly in networking and wireless communications, where effective congestion management and resource allocation mechanisms are essential. Numerous studies have employed queuing theory to analyze network performance, enhance quality of service, allocate resources in accordance with fluctuating demands, identify performance bottlenecks, and improve overall system efficiency [PD19]. Within the context of wireless communications, queuing models serve a critical role in evaluating medium access control protocols, managing retransmissions, controlling congestion, and optimizing resource utilization.

Among the various queuing models, queueing networks with retrials (QN with retrials) represent an important extension of classical systems. In such models, the concept of orbiting customers is introduced, whereby requests encountering a busy server are not discarded but instead enter an orbit and attempt service again after a random delay. This framework provides a more realistic representation of systems in which blocked requests do not immediately exit the system but generate future service attempts.

Retrial queuing models have proven particularly relevant for modern communication systems, including cellular networks [MSB22, TG25], local and optical area networks, and large-scale infrastructures such as call centers and data centers. Such networks often operate under high-load conditions with fluctuating demand and constrained service capacity. Retrial queueing models accurately capture this behavior by allowing blocked or delayed calls to reattempt connection after a random time interval. This approach enables a more realistic representation of user behavior and system dynamics compared to classical queueing models. In these contexts, retrial mechanisms provide analytical tools for addressing unique challenges of resource allocation, ultra-high frequency and directional communications, as well as the dynamics of repeated user requests [PD19].

The present paper addresses the problem of evaluating the accuracy of analytical results for a call center model represented as a retrial QN [RP24]. To this end, a discrete-event simulation algorithm was designed that explicitly incorporates the structural characteristics of the model. In particular, the distributor is treated as the central routing node of the network, which operates under a loss system discipline – that is, it maintains no waiting buffer – and directs arrivals from the external Poisson input flow either to the appropriate service stations, to the orbit (retrial buffer), or to queues with limited waiting times. On the basis of extensive simulation experiments, the accuracy of the analytical performance characteristics was assessed, and the dynamic behavior of the network under varying load conditions was analyzed.

2. Methods and models

2.1. Queueing systems with retrials. Consider a multi-channel RQ subject to a homogeneous Poisson arrival stream with rate λ. The system comprises c statistically identical service devices, *servers*. The service time in the system devices is distributed according to some law with a set of parameters ν. If an incoming request finds a free service device, it immediately enters it and leaves the system after servicing. If an incoming request finds that all service devices are busy, it is directed to a special waiting area, called the *orbit*, and repeats the attempt after some random time distributed by some law with a set of parameters μ. We will assume that the intervals between request arrivals, the service time, and the time of repeated request calls from the orbit are mutually independent.

The stochastic state of the system at time t can be represented by the two-dimensional process $X = \{(C(t), N(t)) : t \geq 0\}$, where $C(t)$ denotes the number of busy servers and $N(t)$ denotes the number of requests in the orbit. If both the service times and retrial times are exponentially distributed, the process X constitutes a continuous-time Markov chain with state space $S = \{0, 1, \dots, c\} \times \mathbb{Z}_+$. Thus, in analyzing RQ, it is necessary to monitor not only the occupancy of the servers but also the orbit size. Since retrial attempts occur independently, the flow of repeated arrivals renders the underlying stochastic process non-uniform, thereby significantly increasing the analytical complexity relative to conventional queueing models with infinite waiting buffers [TF97].

A retrial QN generalizes this idea to a network setting with multiple nodes (systems), where requests may move between nodes, join orbits at different nodes if blocked, and retry after random times. In contrast to classical Markovian QN, retrial QN generally lack product-form stationary distributions [AE05]. Exact analytical solutions are available only in some restricted cases [PD12, ML01]. For semi-open queueing networks with state-dependent Markovian input processes, retrials, and impatient requests, explicit formulas for selected performance measures have been obtained [CDDS19]. When retrial times follow non-exponential distributions, exact analysis becomes intractable, and research is usually confined to approximate methods or simulation-based studies [AY10, FN17, SM11, TBS17]. Consequently, simulation of the behavior of RQ and retrial QN remains an essential tool for performance evaluation.

2.2. Call center stochastic model. Let's consider the stochastic model of a large inbound call center as a closed queueing network with a fixed number of customers K described in [RP24]. It consists of $n+1$ service nodes S_i, $i = \overline{1, n+1}$, and a hypothetical node S_0 representing the external environment. In the network model of the call center, requests correspond to calls from customers. Node S_{n+1} (the so-called distributor, which determines the appropriate node for further servicing of the customer calls) is a RQ without a waiting buffer, equipped with an orbit-system O_{n+2}, that acts as a virtual waiting room. Customers transition from S_0 to S_{n+1} with probability $p_{0,n+1} = 1$, initiating a call to the center. The arrival process from S_0 to S_{n+1} follows a non-stationary Poisson process with rate $\lambda(t)k_0(t)$, proportional to the number of customers in S_0.

If distributor S_{n+1} has free servers, the call is serviced immediately; otherwise, it is sent to the orbit O_{n+2}. Customers in the orbit retry after an exponentially distributed time $\tau_{n+2} \sim \mathrm{Exp}(\gamma)$, and may leave the orbit after time $\tau^o_{n+2} \sim \mathrm{Exp}(\eta_{n+2})$, transferring to S_0 with probability $q_{n+2,0} = 1$. After service in S_{n+1}, customers calls are routed to service nodes S_i, $i = \overline{1, n}$, with probability $p_{n+1,i}$, or to S_0 with probability $p_{n+1,0}$ if misrouted. Each node S_i has m_i identical servers and unlimited queue capacity, with service times exponentially distributed with rate μ_i. Customers are served FIFO, and if their waiting time ν_i^{-1} expires, they are considered impatient and leave to S_0 with probability $q_{i0} = 1$.

After service in S_i, a customer either exits to S_0 with probability p_{i0}, or is redirected to another node S_j, $j \neq i$, with probability p_{ij}, $i, j = \overline{1, n}$. All routing probabilities form a stochastic matrix $P = (p_{ij})$, and impatient transitions form a matrix $Q = (q_{ij})$, $i, j = \overline{0, n+2}$. The matrix Q has the non-zero entries $q_{i0} = 1$, $i = \overline{1, n}$, and $q_{n+2,0} = 1$, where q_{ij} is the probability that a customer's waiting time in system S_i expires, in which case the customer is considered impatient and moves to system S_0. The system state at time t is fully described by the vector $k(t) = (k_1(t), \ldots, k_{n+1}(t), k_{n+2}(t))$, where $k_i(t)$ is the number of customers in node S_i, and $k_{n+2}(t)$ in the orbit. The number of customers in the external environment is $k_0(t) = K - \sum_{i=1}^{n+2} k_i(t)$.

The asymptotic case of the functioning of the closed exponential QN with retrial and impatient customers, subject to the critical assumption of a large number of circulating customers and based on the theory of diffusion approximation of Markov processes, was studied in [RP24]. The authors considered the passage to the limit from a Markov chain $k(t)$ to a continuous-state Markov process $\xi(t)$ and derived a system (2.1) of ordinary differential equations using the drift coefficients of the Fokker–Planck–Kolmogorov equation.

THEOREM 2.1 ([RP24, Theorem 1]). *In the asymptotic case of a large number of customers K the probability density function $p(x, t)$ of the random* *process*

$$\xi(t) = \frac{k(t)}{K} = \left(\frac{k_1(t)}{K}, \dots, \frac{k_{n+2}(t)}{K}\right),$$

provided that it is differentiable with respect to t *and twice continuously differentiable with respect to* x_i, $i = \overline{1, n+2}$, *satisfies up to* $O(\varepsilon^2)$, *where* $\varepsilon = 1/K$, *the multidimensional Fokker–Planck–Kolmogorov equation*

$$\frac{\partial p(x,t)}{\partial t} = -\sum_{i=1}^{n+2} \frac{\partial}{\partial x_i}(A_i(x,t)p(x,t)) + \frac{\varepsilon}{2}\sum_{i,j=1}^{n+2}(B_{ij}(x,t)p(x,t)), \tag{2.1}$$

with drifts

$$\begin{aligned} A_i(x,t) &= \sum_{j=1}^{n+2} \mu_j(p_{ji} - \delta_{ji}) \min(l_j, x_j) \\ &\quad - \eta_i(x_i - l_i)\theta(x_i - l_i), \qquad i = \overline{1, n}, \\ A_{n+1}(x,t) &= -\mu_{n+1} \min(l_{n+1}, x_{n+1}) \\ &\quad + \lambda(t)\Big(1 - \sum_{i=1}^{n+2} x_i\Big)\theta(l_{n+1} - x_{n+1}) \\ &\quad + \gamma x_{n+2}\theta(l_{n+1} - x_{n+1}), \\ A_{n+2}(x,t) &= \lambda(t)\Big(1 - \sum_{i=1}^{n+2} x_i\Big)(1 - \theta(l_{n+1} - x_{n+1})) \\ &\quad - \gamma x_{n+2}\theta(l_{n+1} - x_{n+1}) - \eta_{n+2}x_{n+2}. \end{aligned} \tag{2.2}$$

Here, δ_{ij} *is the Kronecker delta,* $l_i = m_i/K$, $i, j = \overline{1, n+2}$, *and*

$$\theta(x) = \begin{cases} 1, & x > 0, \\ 0, & x \le 0 \end{cases}$$

is the Heaviside function.

Using (2.1) and (2.2), and taking into account that $e_{k_i}(t) = \mathbb{E}(k_i(t)) = \mathbb{E}(K\xi_i(t))$, the system

$$\frac{de_{k_i}(t)}{dt} = KA_i\left(\frac{1}{K}e_{k_i}(t)\right), \qquad i = \overline{1, n+2}, \tag{2.3}$$

where $A_i(\cdot)$ are drifts given by formula (2.2), of homogeneous differential equations can be derived for the expected number of requests $e_{k_i}(t)$ in the network systems at time t. Since the drift coefficients (2.2) involve the minimum function and the Heaviside function, this system can only be solved numerically. Clearly, such a solution is approximate and requires further investigation of its accuracy as well as the conditions of its applicability, for example, the values of K.

2.3. Simulation of the network with distributor, orbit and multichannel systems with timeout. Simulation modeling is a method of conducting computational experiments on computers with mathematical models that imitate the behavior of real-world objects, processes, and systems over a specified period. Simulation modeling is able to forecast and plan the behavior of systems in the future and can be used to verify the accuracy of theoretical results obtained when studying analytical system models. Ready-made libraries for QN simulation are convenient and effective for modeling classical systems. However, they face limitations when applied to networks with non-standard distributions, complex dependencies, or unique routing behaviors. As a result, modeling advanced features, such as retrials or specialized queue types, often requires the development of custom simulation algorithms.

To reproduce the process of the QS functioning, it is necessary to describe the vector $k(t)$ of the system states at the discrete times $\tau_i \geq 0$, $i = 0, 1, 2, \ldots$. The initial state of the system at time τ_0 is set, and the next states $\tau_1, \tau_2, \ldots$ are generated according to predefined model. As a result, a vector trajectory of the system states in the specified intervals is obtained. The most adequate way to obtain τ_i is the "special events" method or 0-moments. Each event has a corresponding value τ_i, a random moment of time at which the state of the system changes, and only one component of this vector changes, and in the interval (τ_{i-1}, τ_i) no changes occur. Therefore, in this method we need to monitor only system's "special events", not fixed moments of time. Then, the 0-moments are calculated and processed. In order to obtain average values for QS characteristics, such as the number of busy service devices, the number of requests in queue or the number of requests in orbit, etc., the simulation is "run" several times (the more "runs", the more accurate the result) and for each moment in time, the average value of the desired characteristics is found for all "runs".

To generalize the simulation algorithm for QN, it is sufficient to expand the state vector $k(t)$ to include multiple systems and incorporate the probability matrix $P = (p_{ij})_{(n+1)\times(n+1)}$ of the request transitions between network systems. Here, p_{ij} represents the probability of a request transitioning to the jth queuing system after being served in the ith system; p_{0i} and p_{i0} indicate the probability of requests entering the ith system from the external environment and the probability of the request leaving the network after being served in the ith system, respectively, $i, j = \overline{1, n}$. To determine which queuing system the request enters from the external environment, a random variable η uniformly distributed over the interval $[0, 1]$ is used. Based on the first row of the transition matrix P, the interval into which η falls is determined. If η is in $(p_{j-1}, p_j]$, the request enters the jth system from the external environment. Similarly, the target system to which the request will be transferred after servicing in the ith system is determined.

Let us describe the simulation algorithm for the call center stochastic model described in Section 2.2. The following classes of "special events" may occur during the functioning of the network under study: (1) a request arrives to the network from the external environment S_0; (2) a request has finished servicing in the distributor S_{n+1}; (3) a request has finished servicing in the standard system $S_i, i = \overline{1,n}$; (4) a request makes a repeated call from orbit O_{n+2}; (5) a request leaves the system queue without being serviced; (6) a request leaves the orbit.

The simulation algorithm based on the processing of the described "special" events is presented below.

ALGORITHM.

1. Initialize `distributor`, `orbit`, and list of `standard` systems.
2. Set `events` $\leftarrow \emptyset$, `time` $\leftarrow 0$.
3. Generate first external arrival t_{arr}, put event `('arrival',` t_{arr}`)` into `events` list.
4. **While** `events` not empty **and** `time` $\leq T$:
   - Extract earliest `event` from `events`, update `time`, determine `event.kind`.
   - **If** `event.kind == 'arrival'`:
     - Compute total number of requests in the network.
     - Schedule next arrival t_{arr} based on current load.
     - **If** `distributor` is available:
       Send request to service, schedule `'dist_finish'` event.
     - **Else**:
       Place request into `orbit`, schedule `'orbit_retry'` or `'orbit_leave'` event.
   - **Else if** `event.kind == 'dist_finish'`:
     - Free server in `distributor`, route request to target `standard` system.
     - **If** `standard` has free server:
       Start service, schedule `'service_finish'` event.
     - **Else**:
       Place request into `standard.queue`, schedule `'timeout'` event.
   - **Else if** `event.kind == 'orbit_retry'`:
     - **If** `distributor` is available:
       Move request from `orbit` to service.
     - **Else**:
       Reschedule `'orbit_retry'` or `'orbit_leave'` event.
   - **Else if** `event.kind == 'orbit_leave'`:
     - Remove request from `orbit`.

- **Else if** `event.kind == 'timeout'`:
  - Remove request from `standard.queue`.
- **Else if** `event.kind == 'service_finish'`:
  - Free server, possibly route to another `standard` system.
  - **If** `queue not empty`:
    Start service for next request.

3. Computational experiments

3.1. Choosing the number of simulation runs. In stochastic simulation modeling, the occurrence times of "special events" are generated based on random variables following specified probability distributions. Consequently, each independent run of the simulation produces different event sequences and, therefore, different statistical outcomes. To obtain reliable results, the number of simulation runs must be sufficiently large so that the averaged statistics across runs converge to the true expected values of the system under study. This convergence is guaranteed by the law of large numbers, but the rate of convergence depends on the variability of the modeled system and the type of performance indicators being measured.

At the same time, simulation experiments are computationally expensive. This issue becomes especially significant when analyzing large-scale networks with many interconnected systems or under high input load conditions, where the number of events grows rapidly. In such cases, excessive runs may lead to prohibitive computation times without substantially improving the accuracy of the results.

Therefore, the central task is to determine an optimal balance: the number of simulation runs should be high enough to provide statistical accuracy of the estimated indicators, yet small enough to keep computational costs reasonable. In present study we run preliminary pilot experiments, where a small number of runs is performed first to estimate the variance of key indicators. Based on this variance and confidence intervals we determine how many runs are required to reach the desired precision.

To determine the optimal number of simulation runs, we analyzed convergence plots of the sample mean for the number of busy servers in the systems, queue lengths, and the number of requests in orbit. In parallel, we examined the corresponding confidence intervals. This analysis allowed us to select a number of simulation runs such that the mean values stabilize, with changes not exceeding the predefined tolerance $\delta = |M_{I_l} - M_{I_{l-1}}| < 0.05$, where M_{I_l} is the mean value for I_l runs.

Let us consider a network with the following parameters: the number of standard systems is 2; the number of servers in the systems is $m_1 = m_2 = 4$, the number of servers in distributor is $m_3 = 6$; the service rate in the systems and distributor are $\mu_1 = \mu_2 = 15$, $\mu_3 = 20$. The waiting rates for the standard systems are $\eta_1 = \eta_2 = 5$ and $\eta_4 = 10$ for the orbit. The retrial rate is $\gamma = 50$. The arrival rate is $\lambda = 0.01$ and the initial number of customers in the network is $K = 15000$.

The structure of the network is set by the following non-zero elements of the transition matrix: $p_{03} = 1$, $p_{10} = p_{02} = 0.8$, $p_{12} = p_{21} = 0.2$, $p_{30} = 0.1$, $p_{31} = 0.4$, $p_{32} = 0.5$.

Figures 1 and 2 present the mean number of busy servers $n_i(N_{\text{iter}})$, $i = \overline{1,3}$, the queue length $q_i(N_{\text{iter}})$, $i = 1, 2$, and the orbit size $n_4(N_{\text{iter}})$ as functions of the number N_{iter} of simulation replications. The shaded region denotes the 95% confidence interval within which the corresponding sample means are expected to lie.

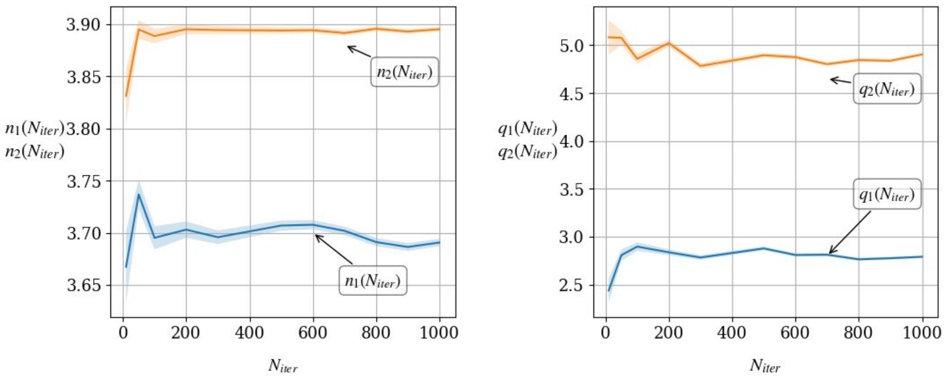

Fig. 1. Average number of busy servers and queue length

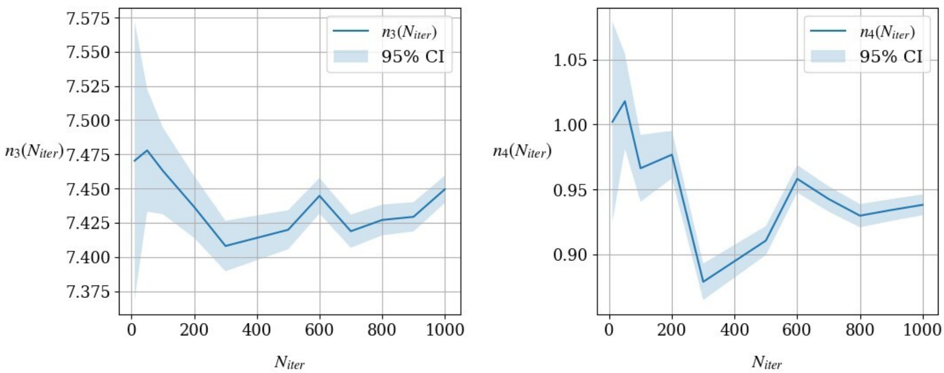

Fig. 2. Average number of busy servers in distributor and orbit size

For the example under consideration, the maximum value of δ over 800 replications of the simulation model is 0.0483 for q_2, while for the other averaged network characteristics δ ranges from 0.004 to 0.0427. When the number of replications exceeds 800, the δ does not exceed 0.0078, for all averaged characteristics. Therefore, in subsequent experiments, we adopt 800 replications of the simulation model as the optimal choice.

3.2. Analysis of the network's stationary regime time. After selecting the optimal number of model runs, an analysis was conducted to determine the time required for the studied network to reach the stationary regime. For a proper comparison between the analytical and simulation models, it is essential to analyze network characteristics only after the system has reached the stationary regime. In the stationary state, the network characteristics (such as the mean number of requests in the systems) are independent of the initial conditions and reflect the stable behavior of the system, enabling accurate comparison with analytical calculations.

A network is considered stationary if all its nodes (queueing systems, distributor, orbit) reach a state in which the observed statistics do not change significantly and fluctuate only within stationary limits. To determine the time for the network to reach the stationary regime, a statistical stabilization criterion was applied to the mean and variance of the number of requests at each network node.

The algorithm for evaluating the time to reach the stationary regime consists of the following steps:

- For each network node, divide the time series of observations into windows of fixed length W (e.g., $W = 5$).
- Compute the mean and variance of the total number of requests $r_i(t)$, $i = \overline{1,4}$, for each window (note that $r_3(t) = n_3(t)$ and $r_4(t) = n_4(t)$).
- Consider the system to have reached the stationary regime if, within the window interval, the mean and variance values do not change significantly, remaining stable within a specified threshold: $|\mu_j - \mu_{j-1}|/\mu_j < \varepsilon$, $|\sigma_j^2 - \sigma_{j-1}^2|/\sigma_j^2 < \varepsilon$, where μ_j and σ_j^2 are the mean value and variance in the jth window.

Experimental results indicate that an increase in window width W leads to a bigger value returned by the proposed algorithm. This effect arises from the fact that, for the given configuration, the network stabilizes relatively quickly, already at $j = 2$. However, the proposed algorithm defines the point of transition to steady-state as $T_{st} = j \cdot W$. Consequently, as W increases, the computed value of T_{st} also increases, even though the actual network behavior reaches stability much earlier.

With a window width $W = 5$ and $\epsilon = 0.05$, the results presented in Figures 3, 4 were obtained. The analysis reveals that the analytical model reaches a steady-state regime slightly faster than the simulation-based model. This discrepancy can be attributed to the variability observed in each indi-

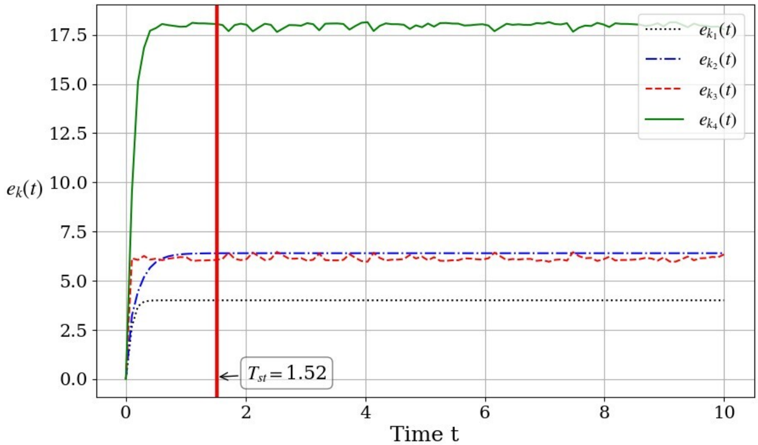

Fig. 3. Time T_{st} for the analytical model to reach steady state, $K = 30000$

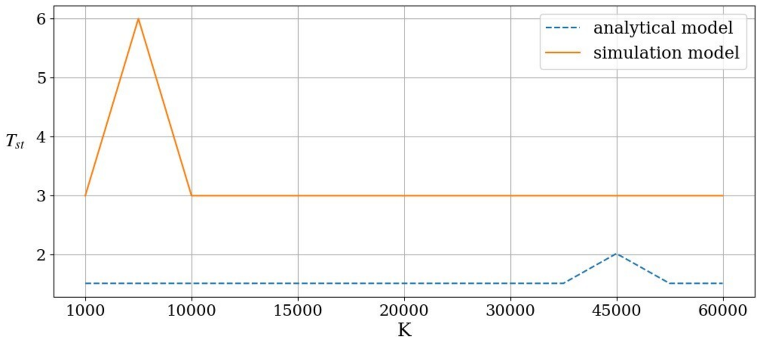

Fig. 4. Time T_{st} to reach steady state depending on network load

vidual run of the simulation model. To mitigate these fluctuations, it is necessary to increase the number of simulation runs, which consequently leads to higher computational costs. As demonstrated earlier, beyond a certain threshold, increasing the number of runs does not result in a substantial improvement in model accuracy, despite the considerable rise in computation time. Based on the presented graphs, it is reasonable to conclude that further investigation into the accuracy of the analytical model should be conducted within the interval of [3; 10], where both models operate in a steady-state regime.

3.3. Comparative analysis of simulation results and analytical estimates considering network load. This section presents a comparative evaluation of the steady-state characteristics of the network described earlier, based on both analytical and simulation models. The analysis focuses on the average number of busy servers in the systems and the distributor, as well as queue lengths and the number of requests in the orbit (orbit size), depending on the intensity of the incoming request flow.

Since analytical formulas (2.2) provide only the total number of requests in standard systems (with the distributor lacking a queue and the orbit defined solely by its size), approximate expressions are used for the analytical model:

- $n_i^a(K) = \min(e_{k_i}(K), m_i)$, the average number of busy servers in the node i for initial request number K;
- $q_i^a(K) = \max(0, e_{k_i}(K) - m_i)$, the average number of queue length in the node i for initial request number K.

Figures 5 and 6 illustrate the steady-state characteristics of the network nodes as a function of the load coefficient K, enabling a qualitative comparison of model behavior under varying load conditions. Figure 7 illustrates the value of the absolute difference between the average values $\Delta_i(K)$ of the characteristics of network nodes depending on the load K. We have $\Delta_i(K) = |r_i(K) - e_{k_i}(K)|$, $i = \overline{1,4}$, where $r_i(K)$ denotes the simulated sample mean of the total number of requests in system i, and $e_{k_i}(K)$ is the corresponding analytical expectation for system i, both evaluated for steady-state under initial network load K.

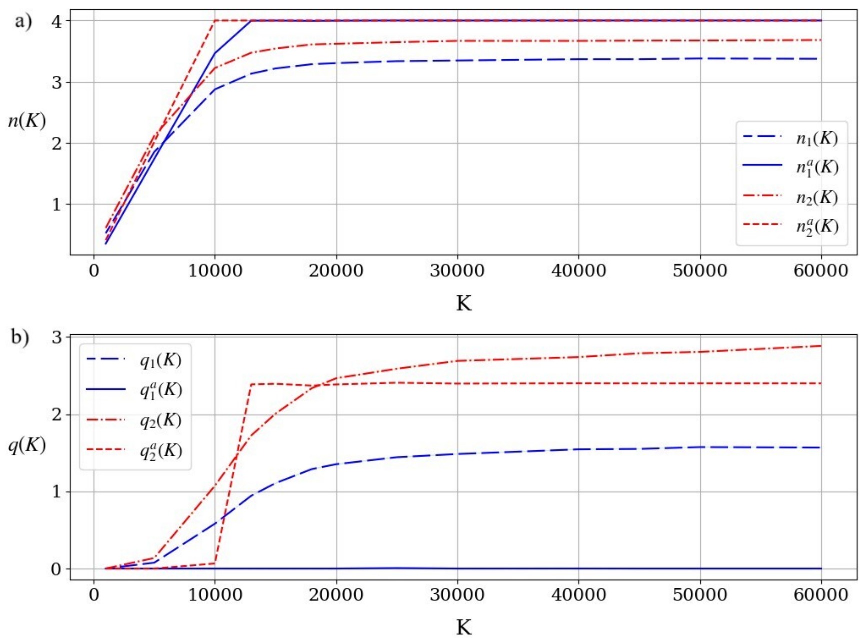

Fig. 5. Average values of (a) the number $n(K)$ of busy servers and (b) the queue length $q(K)$ in standard systems

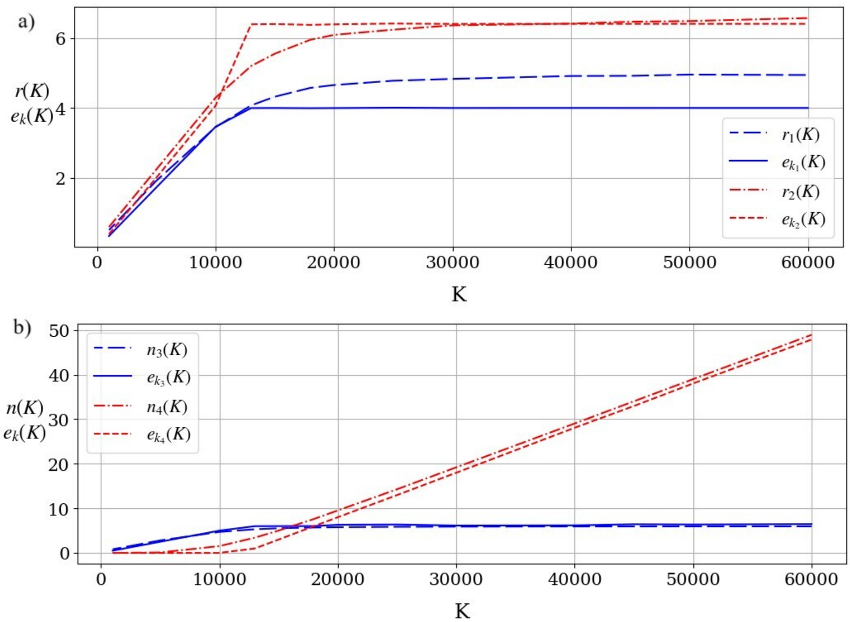

Fig. 6. Average values of (a) the request number $n_3(K)$ in distributor and (b) the orbit size $n_4(K)$

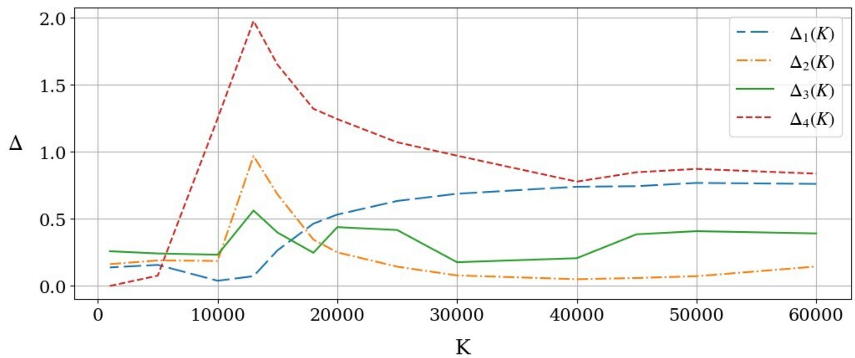

Fig. 7. Absolute difference $\varDelta$ between analytical and simulation values

The simulation results align well with the analytical predictions of the model. This is especially evident at large values of K. For instance, at $K = 60000$, the absolute deviations are $\varDelta_1(K) = 0.7615$, $\varDelta_2(K) = 0.1448$, $\varDelta_3(K) = 0.392$ and $\varDelta_4(K) = 0.838$, which correspond to the values of the relative deviation $\frac{\varDelta_i(K)}{r_i(K)}(\%)$ being 15.41%, 2.21%, 6.58% and 1.71%. These results indicate that the system's behavior matches theoretical expectations. Therefore, the analytical expressions for the expected number of requests in the studied network, obtained through asymptotic analysis based on diffusion approximation and the Fokker–Planck–Kolmogorov equation [TF97], have demonstrated a sufficiently high level of accuracy under heavy load conditions.

4. Conclusions. The study focused on assessing the analytical results for a call center model represented within the framework of a retrial QN with impatient customers. A discrete-event simulation algorithm was developed, incorporating the structural specifics of the distributor as a loss node, the retrial orbit, and finite-capacity queues. Simulation experiments demonstrated that 800 replications are sufficient for obtaining statistically reliable estimates, with the maximum observed deviation δ equal to 0.0483. For a larger number of replications, deviations did not surpass 0.0078 for any performance measure, confirming that 800 iterations constitute an optimal balance between accuracy and computational cost.

Analysis of the transient phase revealed that the choice of window width W in the steady-state detection procedure substantially impacts the estimated stabilization time T_{st}. For instance, with $W = 5$ and $\epsilon = 0.05$, the analytical model reached stationarity considerably earlier than the simulation model, where variability between individual replications delayed convergence. Nevertheless, once stationarity was achieved, both models showed close agreement. At $K = 60000$, the absolute deviations between simulation and analytical results were 0.7615 for system 1, 0.1448 for system 2, 0.392 for the distributor, and 0.838 for the orbit, and deviation decreases as the load increases, indicating improved accuracy under heavier load.

The results confirm that the asymptotic analytical approach, based on diffusion approximation and the Fokker–Planck–Kolmogorov equation, yields highly accurate predictions of steady-state performance measures in large-scale networks. However, despite the high precision of analytical methods under asymptotic assumptions, simulation modeling remains indispensable. This is due to its ability to account for finite-size effects, capture transient dynamics, and analyze configurations that deviate from classical exponential assumptions. Consequently, simulation not only validates analytical approximations but also extends the scope of applicability to more general and practically relevant queueing systems.

References

[AE05] J. R. Artalejo and E. Antonis, *On the non-existence of product-form solutions for queueing networks with retrials*, Electronic Modeling 27 (2005), 13–19.

[AY10] K. Avrachenkov and U. Yechiali, *On tandem blocking queues with a common retrial queue*, Computers Operations Res. 37 (2010), 1174–1180.

[FN17] I. Florea and C. Nănău, *A simulation algorithm for a single server retrial queuing system with batch arrivals*, An. Ştiinţ. Univ. "Ovidius" Constanţa Ser. Mat. 23 (2015), 83–98.

[CDDS19] C. Kim, S. Dudin, A. Dudin and K. Samouylov, *Analysis of a semi-open queuing network with a state dependent marked Markovian arrival process, customers retrials and impatience*, Mathematics 7 (2019), art. 715, 19 pp.

[MSB22] D. Moltchanov, E. Sopin, V. Begishev, A. Samuylov, Y. Koucheryavy and K. Samouylov, *A tutorial on mathematical modeling of 5G/6G millimeter wave and terahertz cellular systems*, IEEE Comm. Surveys Tutorials 24 (2022), 1072–1116.

[ML01] E. Moutzoukis and C. Langaris, *Two queues in tandem with retrial customers*, Probab. Engrg. Informational Sci. 15 (2001), 311–325.

[PD12] T. Phung-Duc, *An explicit solution for a tandem queue with retrials and losses*, Operations Res. Int. J. 12 (2012), 189–207.

[PD19] T. Phung-Duc, *Retrial queueing models: A survey on theory and applications*, arXiv:1906.09560v1 (2019).

[RP24] T. V. Rusilko and A. V. Pankov, *Queueing network model of a call center with customer retrials and impatient customers*, Izv. Saratov Univ. Math. Mech. Inform. 24 (2024), 287–297.

[SM11] Y. Shin and D. Moon, *Approximation of M/M/c retrial queue with PH-retrial times*, Eur. J. Operational Res. 213 (2011), 205–209.

[TG25] D. Talbi and Z. Gal, *Integrating reinforcement learning into M/M/1/K retry queueing models for 6G applications*, Sensors 25 (2025), art. 3621.

[TF97] J. G. C. Templeton and G. I. Falin, *Retrial Queues*, 1st ed., Routledge, 1997.

[TBS17] Á. Tóth, T. Bérczes, J. Sztrik and A. Kvach, *Simulation of finite-source retrial queueing systems with collisions and non-reliable server*, in: Distributed Computer and Communication Networks, DCCN 2017, Lecture Notes in Computer Sci. 700, Springer, Cham, 2017, 146–158.

Katsiaryna Kosarava
Institute of Computer Siences
Cardinal Stefan Wyszyński University in Warsaw
01-938 Warszawa, Poland
E-mail: k.kosarava@uksw.edu.pl