SILESIAN UNIVERSITY OF TECHNOLOGY PUBLISHING HOUSE

SCIENTIFIC PAPERS OF SILESIAN UNIVERSITY OF TECHNOLOGY 2024
ORGANIZATION AND MANAGEMENT SERIES NO. 213

ON TIME TO BUFFER OVERFLOW IN A PROCESSING MODEL WITH PERIODIC SUSPENSION OR SERVICE SLOWDOWN

Wojciech M. KEMPA[1*], Martyna KOBIELNIK[2], Piotr PIKIEWICZ[3], Arkadiusz BANASIK[4]

[1] Silesian University of Technology, Faculty of Applied Mathematics, Department of Mathematical Methods in Technics and Informatics; wojciech.kempa@polsl.pl, ORCID: 0000-0001-9476-2070
[2] Silesian University of Technology, Faculty of Applied Mathematics Department of Mathematical Methods in Technics and Informatics; martyna.kobielnik@polsl.pl, ORCID: 0000-0001-5589-4249
[3] Silesian University of Technology, Faculty of Applied Mathematics, Department of Mathematical Methods in Technics and Informatics; piotr.pikiewicz@polsl.pl, ORCID: 0000-0002-1348-9381
[4] Silesian University of Technology, Faculty of Applied Mathematics Department of Mathematical Methods in Technics and Informatics; arkadiusz.banasik@polsl.pl, ORCID: 0000-0002-4267-2783
* Correspondence author

Purpose: Contemporary challenges related to the need for energy savings and optimization determine the design of service systems (computer and telecommunications networks, production systems, logistic centers, transport solutions, etc.) equipped with appropriate mechanisms supporting the reduction of their operating costs and optimizing the use of available sources and machines. On the other hand, it is essential to maintain the highest quality of service (QoS) for the developing system as much as possible simultaneously.
Designing/methodology/approach: The study employs a queueing-theory-based model to analyze a FIFO-type finite-buffer service system with an embedded optimization mechanism. The primary goal is to balance energy savings and service quality (QoS) in computer networks, telecommunications, production systems, and logistics.
Findings: An explicit Laplace transform representation is derived for the time to first buffer overflow, based on the initial system state. The methodology allows calculating the mean time to buffer overflow for any starting condition.
Originality/value: In the paper, we propose a queueing-theory-based model of the FIFO-type finite-buffer service system in which an optimization algorithm is implemented. Namely, every time the system becomes empty whenever the server chooses either to go on a vacation, during which the processing is blocked completely or to reduce the service intensity temporarily. For a model with a Poisson-type input stream and generally distributed service time and vacation duration, we obtain the explicit formula for the Laplace transform of the time to the first and next buffer overflows conditioned by the accumulating buffer state at the beginning of an appropriate busy period. Hence the mean duration of such times can be found as well.
Keywords: Working vacation, stochastic processes, mathematical modeling.
Category of the paper: Research paper.

http://dx.doi.org/10.29119/1641-3466.2024.213.16

1. Introduction and Motivation

Currently, many papers are available regarding queueing models and their applications to real-life problems. The literature still grows fast, and more complex models are analyzed in different settings to solve problems that arise due to the fast development of technology. From the perspective of computer networks, telecommunication, and production systems, today, one of the most critical problems is reducing energy consumption to keep operating costs low.

One way to keep power consumption low is to stop processing the arriving jobs temporarily. This method has been studied for years and is still one of the main energy-consumption reduction approaches. To prevent the system from switching the state often, which may lead to higher energy consumption when extra energy is needed to turn the server on, or to a higher maintenance cost due to the more excellent wear of the server components, the so-called N-policy can be applied, see, e.g. (Kempa et al., 2010), where the queue length for batch arrival model with N-policy and setup times were considered. The energy-saving capability of the sleep mode was analyzed, e.g., in (Chunxia, Shunfu, 2018; Yin et al., 2020; Yin et al., 2022), where the queueing model was used to model the virtual machine sleep schedule; in (Kempa, 2019), where the vacation queue with N-policy was used to model the Wireless Sensor Network node, in (Harini, Indhira, 2024), where the queueing model with vacation was used to analyze a 5G base station, or in (Jin et al., 2020), where vacation queue with N-policy was considered to save energy in cloud data center.

Unfortunately, applying the sleep mode can lead to a decrease in the quality of service. When the jobs are not processed, they accumulate in the buffer, and due to the finite capacity of the real-life systems, they can be lost when the buffer is saturated. To help reduce the negative impact of completely turning the server off, the working vacation mode can be introduced instead of the vacation mode. In working vacation (or so-called semi-sleep, semi-vacation) mode, the server processes the jobs at reduced speed instead of closing entirely for the arriving customers. In Zhu et al. (2004), Bostoen et al. (2013), one can find the justification for using the working vacation mode as an energy-saving mechanism in the context of cloud data centers and real-time embedded systems. The queue with working vacation policy was considered, for example, in Qin et al. (2019), where this policy was applied to conserve energy on the cloud platform, or in Gong et al. (2020), where the working vacation model with N-policy was used to improve the cost-performance ratio in cyber-physical systems.

Recently, many researchers have considered models with a mixed vacation and working vacation policy. In Jin et al. (2019), a model with two service speeds and vacations was used to model a cloud data center. Based on the state, virtual machines are put into sleep mode or slow down the service. In Ait Braham et al. (2023), two types of vacations are implemented. When the system becomes empty, it takes a type I vacation at random times, and if at the end of the type I vacation, no jobs are waiting, it takes a type II vacation. A similar solution was described in Mohammed Shapique et al. (2024), where the energy-saving capability of this approach is analyzed in the context of WiMAX and tethered high-altitude platform systems. In Divya, Indhira (2024), the cost analysis of the hybrid vacation policy is performed. As one can note, vacation and working vacation are usually used to model servers that can enter more than one successive vacation period. In the following periods, the type of vacation can change. Sometimes, a different approach would be of great value to better reflect the behavior of the modeled system.

This paper considers the finite buffer queueing model with Poisson arrivals and general service times. When the system empties, it changes its operating mode to vacation mode, when no jobs are processed, or to working vacation mode, when jobs are processed but at lower speed. The operating mode is chosen randomly, which may reflect, for example, a system where some background tasks need to be done. The server waits until there are no main tasks to do and then moves some resources to work on secondary tasks, which results in a slower speed of processing main tasks if they appear during this period. When there are no side jobs to be done, the server goes to sleep mode.

The paper is organized as follows. In Section 2, a detailed description of the model is given. In Section 3, the system of integral equations for the transient time to the first buffer overflow is stated and solved in terms of the Laplace transform in Section 4. The Theorem summarizing the result is given with an additional result for the mean time to the first buffer overflow. In Section 5, the instructions to obtain the distribution to k-th buffer overflow are given for $k = 2,3, \ldots$. The mean time to the k-th buffer overflow is also presented.

2. Model description

In this section, we give a detailed description of the service model considered and introduce the necessary notation. Let us consider a single-server queueing model in which messages arrive according to a simple Poisson process with constant rate $a > 0$ and are processed individually, according to a FIFO service rule. A processing time of an individual message is randomly distributed with a cumulative distribution function (CDF) $F(\cdot)$. A message departs the system immediately after its processing is completed. An arriving message that finds the server busy with processing joins the queue and waits for service. An accumulation buffer (waiting room) has a predefined capacity. Hence, the maximum system size equals B, namely $B - 1$ places are available in the buffer, and one spot is reserved for the message being processed. Consequently, if the entering message finds the system saturated (the server is busy with processing and all places in the buffer are occupied) it is lost due to buffer overflow.

Every time when the system empties (at the service completion epoch of the message that leaves the system empty), the optimization mode is being started. Namely, either the service station becomes unavailable for a certain period of time of random length (server vacation), or when a message enters the empty system, its service is started, but its duration has a CDF $F^{\star}(\cdot)$ and with probability one lasts at least as long as in the case of other customers (service slowdown), so we have:

$$F^{\star}(t) \leq F(t) \tag{1}$$

for each $t > 0$.

The choice of one of the two above options is determined by the value of the parameter $\gamma \in [0,1]$. More precisely speaking, with probability γ the server goes to the vacation which is randomly distributed with a CDF $V(\cdot)$. During the vacation, the processing of messages is completely blocked. Otherwise, however, the server with probability $1 - \gamma$chooses the option of the service slowdown. In this case, the server is always ready to process, but the first message incoming to the empty system is served "longer" than other messages.

The above-described mechanism of temporary suspension or slowing down of service allows for practical modeling of a real system in which the optimization mechanism has been implemented. The server handles secondary tasks when there are no messages in the system. If these tasks are large, the server chooses to suspend the handling of primary functions for some time entirely. When the number of secondary tasks is small, the server does not suspend the handling of primary tasks. Still, while the first one is being handled, it simultaneously ultimately finalizes handling secondary tasks. Of course, the period of suspension or slowing down of service can also be used to perform a periodic server inspection or update the appropriate IT system.

Let us denote by $N(t)$ the number of messages (jobs, customers, packets, etc.) present in the system at time $t \geq 0$, including the one being processed at this time, if any.

Besides, let δ_k, $k = 1,2,\ldots$, stands for the time to the kth buffer overflow, so the length of the time period between the completion epoch of the $(k-1)$th busy period of the system and the first moment after this time at which the system becomes saturated (the number of messages present equals B). Obviously, δ_1 represents the time from the opening of the system at $t = 0$ to the first overflow occurrence, so we denote:

$$\delta_1 \overset{def}{=} \min\{t > 0 \colon N(t) = B\}. \tag{2}$$

3. Integral Equations for the Time to the First Buffer Overflow

In this section, we derive the system of integral equations for the tail probability distribution of the time δ_1 to the first buffer overflow, conditioned by the state of the accumulation buffer at the starting epoch $t = 0$.

Introduce the following notation:

$$D_n(t) \overset{def}{=} \mathbf{P}\{\delta_1 > t \mid N(0) = n\}, \tag{3}$$

where: $n \in \{0, \dots, B-1\}$ and $t > 0$.

Assume firstly that the buffer is empty at the opening of the system ($t = 0$). Evidently, in this case the server can choose either going for a vacation or processing with a slower speed. Observe that in such a case, to have a non-zero probability that $\delta_1 > t$, the following mutually excluding random events can occur for fixed $t > 0$:

- $\mathbb{A}_1(t)$: the server goes for a vacation that finishes before time t and the first message arrives after the vacation completion but still before t;
- $\mathbb{A}_2(t)$: the server goes for a vacation that finishes before time t with k messages present, where $k \in \{1, \dots, B-1\}$;
- $\mathbb{A}_3(t)$: the server goes for a vacation that finishes after time t and the number of messages accumulated in the buffer at time t equals k, where $k \in \{1, \dots, B-1\}$;
- $\mathbb{A}_4(t)$: the server chooses slowing down the service and the first message enters before t;
- $\mathbb{A}_5(t)$: the first message occurs after time t.

The formula of total probability gives:

$$D_0(t) = \sum_{i=1}^{5} D_0^{(i)}(t), \tag{4}$$

where:

$$D_0^{(i)}(t) \overset{def}{=} \mathbf{P}\{\delta_1 > t, \mathbb{A}_i(t) \mid N(0) = 0\}. \tag{5}$$

It is easy to check that the following representation is true, considering the random event $\mathbb{A}_1(t)$:

$$D_0^{(1)}(t) = \gamma \int_0^t a e^{-ax} V(x) D_1(t-x) dx. \tag{6}$$

Similarly, for $\mathbb{A}_2(t)$ we get:

$$D_0^{(2)}(t) = \gamma \int_0^t \sum_{k=1}^{B-1} \frac{(ay)^k}{k!} e^{-ay} D_k(t-y) dV(y). \tag{7}$$

For the random event $\mathbb{A}_3(t)$ we obtain the following expression:

$$D_0^{(3)}(t) = \gamma\overline{V}(t)\sum_{k=1}^{B-1}\frac{(at)^k}{k!}e^{-at}. \tag{8}$$

where: $\overline{V}(t) \overset{def}{=} 1 - V(t)$.

In the case of the random event $D_0^{(4)}(t)$ two separate sub-cases should be considered: the first one relates to the situation in that the first (slower) service completes before t, while the second one describes the opposite case. So, we have:

$$\begin{aligned} D_0^{(4)}(t) = (1-\gamma)\int_{x=0}^{t} ae^{-ax}dxD_0^{(3)}(t) = \gamma\overline{V}(t)\sum_{k=1}^{B-1}\frac{(at)^k}{k!}e^{-at}. \\ \times [\int_{y=0}^{t-x}\sum_{k=0}^{B-2}\frac{(ay)^k}{k!}e^{-ay}D_{k+1}(t-x-y)dF^{\star}(y) \\ +F^{\star}(t-x)e^{-a(t-x)}\sum_{k=0}^{B-2}\frac{[a(t-x)]^k}{k!}], \end{aligned} \tag{9}$$

where two summands on the right side of (9) correspond to the first and second sub-cases, respectively.

Finally, obviously, we have:

$$D_0^{(5)}(t) = e^{-at}. \tag{10}$$

Now let us consider the system that is non-empty at the opening. Conditioning by the first departure moment after the starting of the system (this moment is a renewal moment in the evolution of the considered queueing system due to memoryless property of exponential distribution of interarrival times), we obtain:

$$\begin{aligned} D_n(t) = \sum_{k=0}^{B-n-1}\int_0^t\frac{(ay)^k}{k!}e^{-ay}D_{n+k-1}(t-y)dF(y) \\ +\overline{F}(t)e^{-at}\sum_{k=0}^{B-n-1}\frac{(at)^k}{k!}, \end{aligned} \tag{11}$$

where: $k \in \{1,\ldots,B-1\}$ and $\overline{F}(t) \overset{def}{=} 1 - F(t)$.

Indeed, the first summand on the right side of (11) relates to the case in which the first message leaves the system after completing its service before time t, while the second one to the opposite case.

Introduce now the Laplace transform (LT) of $D_n(t)$ as follows:

$$\widehat{D}_n(s) \overset{def}{=} \int_0^{\infty} e^{-st}D_n(t)dt, \tag{12}$$

where: $s > 0$.

We are interested in writing representations obtained for $D_0(t),\ldots,D_{B-1}(t)$ in terms of their LTs.

Let us note that (compare the right side of (6))

$$\begin{aligned}&\gamma \int_{t=0}^{\infty} e^{-st} dt \int_{x=0}^{t} a e^{-ax} V(x) D_1(t-x) dx \\ &= \gamma a \int_{x=0}^{\infty} e^{-(a+s)x} V(x) dx \int_{t=x}^{\infty} e^{-s(t-x)} D_1(t-x) dt = A(s)\widehat{D}_1(s),\end{aligned} \tag{13}$$

where:

$$A(s) \overset{def}{=} \gamma a \int_{0}^{\infty} e^{-(a+s)x} V(x) dx. \tag{14}$$

Next, we have (see the right side of (7)):

$$\begin{aligned}&\gamma \int_{t=0}^{\infty} e^{-st} dt \int_{y=0}^{t} \sum_{k=1}^{B-1} \frac{(ay)^k}{k!} e^{-ay} D_k(t-y) dV(y) \\ &= \gamma \sum_{k=1}^{B-1} \int_{y=0}^{\infty} \frac{(ay)^k}{k!} e^{-(a+s)y} dV(y) \times \int_{t=y}^{\infty} e^{-s(t-y)} D_k(t-y) dt \\ &= \sum_{k=1}^{B-1} B_k(s)\widehat{D}_k(s),\end{aligned} \tag{15}$$

where:

$$\widehat{B}_k(s) \overset{def}{=} \int_{0}^{\infty} e^{-(a+s)y} \frac{(ay)^k}{k!} dV(y). \tag{16}$$

According to (8), let us define:

$$C(s) \overset{def}{=} \gamma \int_{0}^{\infty} e^{-(a+s)t} \overline{V}(t) \sum_{k=1}^{B-1} \frac{(at)^k}{k!} dt. \tag{17}$$

Changing the order of integration according to the following scheme (compare the right side of (9)):

$$\int_{t=0}^{\infty} \int_{x=0}^{t} \int_{y=0}^{t-x} \rightarrow \int_{x=0}^{\infty} \int_{t=x}^{\infty} \int_{y=0}^{t-x} \rightarrow \int_{x=0}^{\infty} \int_{y=0}^{\infty} \int_{t=x+y}^{\infty}, \tag{18}$$

we obtain:

$$\begin{aligned}&(1-\gamma) \int_{x=0}^{\infty} a e^{-(a+s)x} dx \int_{y=0}^{\infty} \sum_{k=1}^{B-2} \frac{(ay)^k}{k!} e^{-(a+s)y} dF^{\star}(y) \\ &\times \int_{t=x+y}^{\infty} e^{-s(t-x-y)} D_{k+1}(t-x-y) dt = \sum_{k=0}^{B-2} E_k(s)\widehat{D}_{k+1}(s),\end{aligned} \tag{19}$$

where:

$$E_k(s) \overset{def}{=} \frac{(1-\gamma)a}{a+s} \int_{0}^{\infty} \frac{(ay)^k}{k!} e^{-(a+s)y} dF^{\star}(y). \tag{20}$$

Similarly (see (9)):

$$(1-\gamma)\sum_{k=0}^{B-2}\frac{a^{k+1}}{k!}\int_{x=0}^{\infty}e^{-(a+s)x}dx\times\int_{t=x}^{\infty}e^{-(a+s)(t-x)}(t-x)^{k}F^{\star}(t-x)dt$$
$$=\sum_{k=0}^{B-2}G_k(s), \tag{21}$$

where:

$$G_k(s)\overset{def}{=}\frac{(1-\gamma)a}{a+s}\int_0^{\infty}\frac{(at)^k}{k!}e^{-(a+s)t}F^{\star}(t)dt. \tag{22}$$

Obviously, the LT of the right side of (10) gives:

$$\frac{1}{a+s}. \tag{23}$$

Referring now to (12)-(23) we can write:

$$\widehat{D}_0(s)=\sum_{k=1}^{B-1}\left(\delta_{k,1}A(s)+B_k(s)+E_{k-1}(s)\right)\widehat{D}_k(s)+C(s)+\sum_{k=0}^{B-2}G_k(s)+\frac{1}{a+s}. \tag{24}$$

So, defining:

$$\Theta_k(s)\overset{def}{=}\delta_{k,1}A(s)+B_k(s)+E_{k-1}(s), \tag{25}$$

where: $k\in\{1,\dots,B-1\}$, and:

$$\Phi(s)\overset{def}{=}C(s)+\sum_{k=0}^{B-2}G_k(s)+\frac{1}{a+s}, \tag{26}$$

we obtain:

$$\widehat{D}_0(s)=\sum_{k=1}^{B-1}\Theta_k(s)\widehat{D}_k(s)+\Phi(s). \tag{27}$$

Similarly, taking LTs of both sides of (11) we get:

$$\widehat{D}_n(s)=\sum_{k=0}^{B-n-1}\alpha_k(s)\widehat{D}_{n+k-1}(s)+\beta_n(s), \tag{28}$$

where: $n\in\{1,\dots,B-1\}$ and:

$$\alpha_k(s)\overset{def}{=}\int_0^{\infty}e^{-(a+s)y}\frac{(ay)^k}{k!}dF(y) \tag{29}$$

and:

$$\beta_n(s)\overset{def}{=}\sum_{k=0}^{B-n-1}\int_0^{\infty}e^{-(a+s)t}\frac{(at)^k}{k!}\overline{F}(t)dt. \tag{30}$$

4. Explicit Solution in Terms of Laplace Transforms

In this section, we give an explicit solution of the linear system of equations (27)-(28), which is written using a certain auxiliary functional sequence. Firstly, we should reformulate (27)-(28).

Let us apply the following substitution:

$$\widehat{H}_n(s) \overset{def}{=} \widehat{D}_{B-n}(s) \tag{31}$$

Equations (27)-(28) has now the following forms:

$$\widehat{H}_B(s) = \sum_{k=1}^{B-1} \Theta_{B-k}(s)\widehat{H}_k(s) + \Phi(s) \tag{32}$$

and:

$$\sum_{k=-1}^{n-1} \alpha_{k+1}(s)\widehat{H}_{n-k}(s) - \widehat{H}_n(s) = \varphi_n(s), \tag{33}$$

where: $n \in \{1, \dots, B-1\}$ and:

$$\varphi_n(s) \overset{def}{=} \widehat{H}_1(s)\alpha_n(s) - \beta_{B-n}(s). \tag{34}$$

To obtain the solution of the system (32)–(33) in a compact form, we use an auxiliary algebraic result. The following lemma can be found in (Korolyuk, 1974; see also Kempa, 2016).

Lemma 1.

Assume that $(u_n(s))$ and $(v_n(s))$ are two given functional sequences, where additionally $u_0(s) \neq 0$. Each solution of the system of infinite number equations of the form:

$$\sum_{k=-1}^{n-1} u_{k+1}(s)x_{n-k}(s) - x_n(s) = v_n(s), \tag{35}$$

where: $n \geq 1$, can be expressed as follows:

$$x_n(s) = M(s)R_n(s) + \sum_{k=1}^{n} R_{n-k}(s)v_k(s), \tag{36}$$

where: $n \geq 1$, $M(s)$ is certain function (independent on n) and the functional sequence $(R_k(s))$ is defined as follows:

$$R_0(s) = 0, R_1(s) = u_0^{-1}(s),$$
$$R_{k+1}(s) = R_1(s)\left[R_k(s) - \sum_{i=0}^{k} u_{i+1}(s)R_{k-i}(s)\right] \tag{37}$$

for $k \geq 1$.

Let us note that in (32)-(33) the role of $u_k(s)$ and $v_k(s)$ play $\alpha_k(s)$ and $\varphi_k(s)$, respectively, and the unknown functional sequence is now $(\widehat{H}_n(s))$. Moreover, because the number of equations in (32)-(33) is finite, one can use the equation (32) as a kind of a boundary condition that will be helpful to express $M(s)$ explicitly.

In consequence, we have:

$$\widehat{H}_n(s) = M(s)R_n(s) + \sum_{k=1}^{n} R_{n-k}(s)\varphi_k(s), \tag{38}$$

where: $n \geq 1$ and:

$$R_0(s) = 0, R_1(s) = \alpha_0^{-1}(s),$$
$$R_{k+1}(s) = R_1(s)\left[R_k(s) - \sum_{i=0}^{k} \alpha_{i+1}(s)R_{k-i}(s)\right]. \tag{39}$$

Obviously, it is necessary to find the representation for $\widehat{H}_1(s)$ and $M(s)$ occurring in (34) and (38), respectively.

Substituting $n = 1$ into (38) we obtain:

$$\widehat{H}_1(s) = M(s)R_1(s). \tag{40}$$

Next, let us note that, taking $n = B$ in (38) and applying (34) and (40), we get:

$$\widehat{H}_B(s) = M(s)R_B(s) + \sum_{k=1}^{B} [M(s)R_1(s)\alpha_k(s) - \beta_{B-k}(s)]R_{B-k}(s). \tag{41}$$

Simultaneously, from the other side we have from (32), referring to (34) and (38):

$$\widehat{H}_B(s) = \sum_{k=1}^{B-1} \Theta_{B-k}(s)[M(s)R_k(s)$$
$$+\sum_{i=1}^{k} (M(s)R_1(s)\alpha_i(s) - \beta_{B-i}(s))R_{k-i}(s)] + \Phi(s). \tag{42}$$

Introduce now the following auxiliary notations:

$$\Gamma_k(s) \overset{def}{=} R_k(s) + R_1(s)\sum_{i=1}^{k} \alpha_i(s)R_{k-i}(s)M(s) = FAC1^{-1}(s)FAC2(s), \tag{43}$$

and:

$$\Delta_k(s) \overset{def}{=} \sum_{i=1}^{k} \beta_{B-i}(s)R_{k-i}(s). \tag{44}$$

Comparing the right sides of representations (41) and (42) we eliminate $M(s)$ in the following form:

$$M(s) = FAC1^{-1}(s)FAC2(s), \tag{45}$$

where:

$$FAC1(s) \overset{def}{=} \Gamma_B(s) - \sum_{k=1}^{B-1} \Theta_{B-k}(s)\Gamma_k(s) \tag{46}$$

and:

$$FSC2(s) \overset{def}{=} \Delta_B(s) - \sum_{k=1}^{B-1} \Theta_{B-k}(s)\Delta_k(s) + \Phi(s). \tag{47}$$

Now we have (see (38)):

$$\begin{gathered}\widehat{H}_n(s) = FAC1^{-1}(s)FAC2(s)R_n(s) \\ +\sum_{k=1}^{n} [FAC1^{-1}(s)FAC2(s)R_1(s)\alpha_k(s) - \beta_{B-k}(s)]R_{n-k}(s) \\ = FAC1^{-1}(s)FAC2(s)[R_n(s) + R_1(s)\sum_{k=1}^{n} \alpha_k(s)R_{n-k}(s)] - \sum_{k=1}^{n} \beta_{B-k}(s)R_{n-k}(s).\end{gathered} \tag{48}$$

Returning to functional sequence $\widehat{D}_n(s)$ (see (32)), we can formulate the following theorem.

Theorem 1.

In the queueing model, the representation for the LT of the tail CDF of the time to the first buffer overflow is given by the following formula:

$$\begin{gathered}\widehat{D}_n(s)(t) = FAC1^{-1}(s)FAC2(s)[R_{B-n}(s) \\ +R_1(s)\sum_{k=1}^{B-n} \alpha_k(s)R_{B-n-k}(s)] - \sum_{k=1}^{B-n} \beta_{B-k}(s)R_{B-n-k}(s),\end{gathered} \tag{49}$$

where: $n \in \{0,\ldots,B-1\}$ the and the representations for $FAC1(s)$, $FAC2(s)$, $R_k(s)$, $\alpha_k(s)$ and $\beta_k(s)$ are given in (46), (47), (39), (29) and (30), respectively.

Just from the definition of $\widehat{D}_n(t)$ we get, as a corollary from Theorem 1, the following representation for the mean value of the time to the first buffer overflow conditioned by the initial buffer state n.

Corollary 1.

The mean time $\boldsymbol{E}_n(\delta_1)$ to the first buffer overflow under condition that the accumulation buffer contains exactly n messages initially is given by:

$$\mathbf{E}_n(\delta_1) = \int_0^\infty \mathbf{P}\{\delta_1 > t \mid N(0) = n\}dt = \widehat{D}_n(0). \tag{50}$$

5. Next Buffer Overflow Periods

Defining the next times to buffer overflow by δ_k, where $k = 2,3,\ldots$ (we assume here that appropriate time is measured beginning with the completion epoch of the previous buffer overflow), let us note that after finishing each buffer overflow period the number of messages present in the system equals $B-1$ due to the individual service process organization.

Hence, we have the following corollary.

Corollary 2.

The LT of the probability that the time δ_k to the kth buffer overflow (counting from the completion epoch of the $(k-1)$th such period) exceeds t, where $k = 2,3,\dots$, is given by:

$$\int_0^{\infty} e^{-st}\mathbf{P}\{\delta_1 > t \mid N(0) = B-1\}dt = \widehat{D}_{B-1}(s), \tag{51}$$

So is the same as the analogous transform given for the time to the first buffer overflow δ_1 on condition $N(0) = B-1$.

Therefore, we obtain

Corollary 3.

The mean value of the time δ_k to the kth buffer overflow period, where $k = 2,3,\dots$, is given by:

$$\mathbf{E}(\delta_k) = \widehat{D}_{B-1}(0). \tag{52}$$

6. Conclusions and future work

The paper analyzed the theoretical model of a service unit with periodic suspension or service slowdown. The finite buffer queueing model was proposed with two types of vacations to obtain a compromise between energy savings ability and maintaining the highest possible quality of service. The system of integral equations was solved using an algebraic approach in terms of Laplace transforms. The Laplace transform of the time to the first buffer overflow is obtained, and the mean time to the first overflow is given. The main result is then followed by the time to the k-th buffer overflow ($k = 2,3,\dots$) distribution and the respective mean time to the k-th buffer overflow. With the explicit solution, a numerical study can be conducted after the inversion of the Laplace transform. It can be done using one of the methods that can be found in the literature (see, for example Abate et al., 2000).

References

1. Abate, J., Choudhury, G.L., Whitt, W. (1999). An Introduction to Numerical Transform Inversion and Its Application to Probability Models. In: W. Grassman (Ed.), *Computational Probability* (pp. 257-323). Boston, MA: Springer.
2. Ait Braham, K., Taleb, S., Aissani, A. (2023). Performance Analysis of a Non-markovian Queue with Differentiated Vacations. *SN Computer Science, 4(5)*, p. 512. Retrieved from: https://link.springer.com/10.1007/s42979-023-02057-9.

3. Bostoen, T., Mullender, S., Berbers, Y. (2013). Power-reduction techniques for data-center storage systems. *ACM Computing Surveys, 45(3)*. Retrieved from: https://doi.org/10.1145/2501654.
4. Chunxia, Y., Shunfu, J. (2018). An energy-saving strategy based on multi-server vacation queuing theory in cloud data center. *The Journal of Supercomputing*, *74(12), 6766-6784*. Retrieved from: http://link.springer.com/10.1007/s11227-018-2513-4.
5. Divya, K., Indhira, K. (2024). Performance analysis and ANFIS computing of an unreliable Markovian feedback queueing model under a hybrid vacation policy. *Mathematics and Computers in Simulation, 218*, pp. 403-419. Retrieved from: https://linkinghub.elsevier .com/ retrieve/ pii/S0378475423005104.
6. Gong, H., Li, R., An, J., Bai, Y., Li, K. (2020). Quantitative Modeling and Analytical Calculation of Anelasticity for a Cyber-Physical System. *IEEE Transactions on Systems, Man, and Cybernetics: Systems, 50(11)*, 4746-4761. Retrieved from: https://ieeexplore.ieee.org/ document/8454896/.
7. Harini, R., Indhira, K. (2024). Dynamical modelling and cost optimization of a 5G base station for energy conservation using feedback retrial queue with sleeping strategy. *Telecommunication Systems, 86(4), 661-690*. Retrieved from: https://link.springer.com/10.1007/s11235-024-01155-0.
8. Jin, S., Hao, S., Qie, X., Yue, W. (2019). A virtual machine scheduling strategy with a speed switch and a multi-sleep mode in cloud data centers. *Journal of Systems Science and Systems Engineering, 28(4)*, pp. 567-589.
9. Jin, S., Qie, X., Zhao, W., Yue, W., Takahashi, Y. (2020). A clustered virtual machine allocation strategy based on a sleep-mode with wake-up threshold in a cloud environment. *Annals of Operations Research, Vol. 293(1)*, pp. 193-212.
10. Kempa, W.M., Venkov, G., Pasheva, V., Kovacheva, R. (2010). *The transient analysis of the queue-length distribution in the batch arrival system with N-policy, multiple vacations and setup times.* AIP Conference Proceedings, Sozopol (Bulgaria): AIP Publishing, pp. 235-242. Retrieved from https://pubs.aip.org/aip /acp/article/1293/1/235-242/815464.
11. Kempa, W.M. (2016). Transient workload distribution in the M/G/1 finite-buffer queue with single and multiple vacations. *Annals of Operations Research, 239(2),* pp. 381-400. Retrieved from http://link.springer.com/10.1007/s10479-015-1804-x.
12. Kempa, W.M. (2019). Analytical Model of a Wireless Sensor Network (WSN) Node Operation with a Modified Threshold-Type Energy Saving Mechanism. *Sensors, 19(14), 3114*. Retrieved from: https://www.mdpi.com/1424-8220/19/14/3114.
13. Korolyuk, V.S. (1974). Boundary Problems for a Compound Poisson Process. *Theory of Probability & Its Applications, 19(1)*, pp. 1-13. Retrieved from: http://epubs.siam.org/doi/10.1137/1119001.
14. Mohammad Shapique, A., Sudhesh, R., Dharmaraja, S. (2024). Transient Analysis of a Modified Differentiated Vacation Queueing System for Energy-Saving in WiMAX.

Methodology and Computing in Applied Probability, 26(3), p. 23. Retrieved from: https://link.springer.com/10.1007/s11009-024-10094-x.

15. Qin, B., Jin, S., Zhao, D. (2019). Energy-Efficient Virtual Machine Scheduling Strategy with Semi-Sleep Mode on the Cloud Platform. *International Journal of Innovative Computing, Information and Control, 15(1), 337-350.* Retrieved from: https://doi.org/10.24507/ijicic.15.01.337.
16. Yin, C., Liu, J., Jin, S. (2020). An Energy-Efficient Task Scheduling Mechanism with Switching On/Sleep Mode of Servers in Virtualized Cloud Data Centers. *Mathematical Problems in Engineering.* Retrieved from: https://www.hindawi.com/journals/mpe/2020/4176308/.
17. Yin, C., Liu, J., Jin, S. (2022). A virtualized data center energy-saving mechanism based on switching operating mode of physical servers and reserving virtual machines. Concurrency and Computation: *Practice and Experience, 34(9), e5785*. Retrieved from: https://onlinelibrary.wiley.com/doi/10.1002/cpe.5785.
18. Zhu, D., Melhem, R., Mosse, D. (2004). The effects of energy management on reliability in real-time embedded systems. *Proceedings of IEEE/ACM International Conference on Computer-Aided Design (ICCAD-2004)*, pp. 35-40.