Study on Transient Queue-Size Distribution in the Finite-Buffer Model with Batch Arrivals and Multiple Vacation Policy

Wojciech M. Kempa and Rafał Marjasz

Abstract: The transient behavior of the finite-buffer queueing model with batch arrivals and generally distributed repeated vacations is analyzed. Such a system has potential applications in modeling the functioning of production systems, computer and telecommunication networks with energy saving mechanism based on cyclic monitoring the queue state (Internet of Things, wireless sensors networks, etc.). Identifying renewal moments in the evolution of the system and applying continuous total probability law, a system of Volterra-type integral equations for the time-dependent queue-size distribution, conditioned by the initial buffer state, is derived. A compact-form solution for the corresponding system written for Laplace transforms is obtained using an algebraic approach based on Korolyuk’s potential method. An illustrative numerical example presenting the impact of the service rate, arrival rate, initial buffer state and single vacation duration on the queue-size distribution is attached as well.

Keywords: finite buffer; multiple vacation policy; potential method; queue size; transient state

1. Introduction

Mathematical models of queueing systems with limited capacity of buffers accumulating customers waiting for service are widely used in practice. Among many different areas of their application, it is worth mentioning issues related to modeling the functioning of production systems, problems arising in transport issues or logistics. However, particular applications of systems with finite waiting rooms have been found in modeling the functioning of telecommunication and computer network nodes. Indeed, phenomena, such as the merging of many independent streams of incoming packets (packet queuing), packet processing (service) but also possible overflows of accumulating buffers and packet losses because of too much disproportion between the intensity of the input traffic and the speed of processing, occur there. Dynamically developing the branch of wireless communication (Wi-Fi, LTE, sensor networks, etc.) forces the search for solutions that, in addition to modeling the packet processing itself, ensure the implementation of tasks resulting from the need for energy saving. In the field of modeling the energy saving mode in wireless network nodes, queueing systems with various types of mechanisms of the periodic suspension of customer processing are used. Among many solutions, it is worth mentioning the $N$-policy threshold-type discipline, originally proposed in [1], in which the processing restarts after an idle period simultaneously with the $N$th arrival, and the $T$-policy, introduced in [2], in which the server is being activated exactly $T$ time units after the last busy period. Moreover, in a fundamental paper [3], single and multiple vacation policies were proposed in which a single or repeated independent vacations are taken until at least one customer present in the buffer is detected. The results obtained in [1,3] for the $M/G/1$-type queue were next generalized in [4,5] for the batch Poisson arrival stream and for the $M/M/c$ queue, respectively. An overview of results obtained by using the
supplementary variable technique for non-Markovian queues with different-type server vacations can be found in [6]. In [7], the strategic joining behavior of jobs in a Markovian queueing system with Bernoulli vacation is investigated.

In the article, we consider the queueing model with multiple vacation periods launched after the completion of each busy period of the system. Queueing systems of this type can be used in the practical modeling of the functioning of, for example, computer and telecommunication network nodes (in particular wireless, e.g., based on Wi-Fi or LTE standards), in which an energy saving mechanism is implemented based on the cyclical checking the state of queue of packets waiting for processing. A multiple vacation period may also be used to perform some other operations for the service station (such as e.g., maintenance, software update, etc.). As one can observe, the literature on this type of model is extensive and constantly growing. However, as it seems, the results relate almost exclusively to the stationary state of the system. It is rare to encounter new analytical results for the transient state (at the fixed time moment $t_i$). In [8], a batch-arrival model with Poisson input flow, in which the processing consists of two phases, was studied. Autocorrelated input streams with models governed by the multiple vacation policy were considered in [9,10], where setup and close-down times are added into the control mechanism, and the MAP and BMAP arrival processes were assumed, respectively. In [11], a model with dependent arrival and service processes was analyzed. A BMAP-type arrival stream in a queueing system with multiple vacations and setup times was considered in [12]. An interesting modification of the classical multiple vacation scheme can be found in [13], where the model with a fixed maximal number of vacations was investigated. A queue with a control mechanism based on Active Queue Management (AQM) was considered in [14] via diffusion approximation. In [15], an infinite-buffer system of the $GI/M/1$ type with multiple exponentially distributed vacations was considered in the stationary state. The results obtained there were generalized for the case of multiple working vacations in [16], where the server repeatedly starts the operation at a slower speed instead of suspending it completely. Discrete-time queueing models with a multiple vacations policy were analyzed, for example, in [17,18]. A finite-buffer model of the $M/G/1/N$-type with vacations and exhaustive service was investigated in [19]. The problem of a stochastic decomposition structure of the busy period in a finite-buffer discrete-time vacation queue was discussed in [20]. One can find a comprehensive review of many important analytical results for different vacation queueing models, for example, in [21–23]. Equilibrium strategies in models with working vacations are studied in [24,25]. In [26], a Markovian queueing model with server vacations caused by breakdowns was investigated in a stationary regime. Various steady-state results for queueing models with an energy saving based on multiple vacation mechanism with additional, more complex control policies can be found, for example, in [27–32]. In fact, queueing models with a multiple vacation policy have many applications in different real-life areas, for example, in the case of customer service in a bank or at the post office, individual checkout windows are periodically closed in the event of a lack of customers: then clerks deal with other tasks (e.g., organizing documentation, preparing reports, etc.), periodically monitoring the state of the queue of customers (see e.g., [33]). An example of applying the concept of a queueing system with multiple vacations in the financial sector can be found, for example [34]. The optimal pricing server strategies under the ex-postpayment and the ex-antepayment schemes are analyzed there.

The transient analysis of the $M/M/1$-type system with multiple exponential vacations and additional threshold-type $N$-policy can be found in [35]. In [36], a compact-form solution for the queue-size distribution in the $M/M/1$-type queue with a multiple vacation policy and at most $J$ single vacations are obtained. An infinite-capacity queueing system of the $M/G/1$ type with batch arrivals was investigated in [37,38] in the non-stationary state. Analytical results for departure process and queue-size distribution were obtained there, for the case of multiple and single vacation policies, respectively.

To the best of the authors’ knowledge, in the literature there are no analytical results in the transient state for queueing models with a multiple vacation policy, in which processing
time and single vacation duration are generally distributed. In this article, we study the time-dependent queue-size distribution in the $M^X/G/1/N$-type queueing model with a multiple vacation policy and an exhaustive service. Applying a theoretical approach based on identifying Markovian moments in the evolution of the system, we construct a Volterra-type system of integral equations for the transient queue-size distribution conditioned by the buffer filling level at starting time $t = 0$. For the corresponding system written for Laplace transforms (LTs for short) we use the idea of Korolyuk’s potential (see [39]) to obtain the solution in the compact form. So, the main contribution of the paper is the closed-form representation for the LT of the conditional queue-size distribution in the non-stationary state of the system. The formula is written by means of a certain functional sequence (in [39] called a potential) with terms, which can be obtained recursively or with the help of power expansion. Numerical examples illustrate analytical results, namely the impact of the vacation duration, arrival intensity, service speed, buffer size and the initial system state on the behavior of the queue-size distribution.

The remaining part of the article is organized as follows. In Section 2, we provide a detailed description of the considered queueing model and state the necessary nomenclature. In Section 3, we obtain the system of integral equations for the conditional queue-size distribution in the transient state. In Section 4, we sketch the idea of Korolyuk’s potential approach and use it to obtain the compact-form solution of the corresponding system written for LTs. Section 5 is devoted to numerical computations illustrating theoretical results. To visualize the appropriate probabilities we use one of the algorithms of numerical LT inversion. Moreover, comparing numerical results to the simulation is conducted. Finally, Section 6 contains a summary and short conclusions.

2. Mathematical Description of the Model

We deal with the $M^X/G/1/N$-type queueing model in which groups (batches) of packets arrive according to a compound Poisson process with intensity $\lambda$. Group sizes are generally-distributed random variables and $p_k$ stands for the probability that the arriving group consists of $k$ packets exactly, where $\sum_{k=1}^{\infty} p_k = 1$. The processing is organized according to the natural FIFO discipline and the service time of an individual packet is a random variable with a general-type cumulative distribution function (CDF for short) $F(\cdot)$. The system capacity equals $N$, that is, there is an accumulating buffer with $N - 1$ places for waiting packets and one place reserved for the packet being processed.

The considered queueing system is governed by the multiple vacation policy. Namely, every time the server becomes idle (at the completion epoch of each busy period) the server takes a vacation of a random duration with a general CDF $G(\cdot)$. After the vacation, the buffer state is examined. If there are no packets waiting for service, the next independent vacation is started and so on. The last vacation is the one at the completion epoch of which at least one packet waiting for processing is detected. Successive server vacations started as long as the buffer is empty are called a multiple vacation period (MVP for short). So, the operation of the system can be observed in successive busy periods, during which the processing of packets is uninterrupted, followed by MVPs during which the processing is suspended.

We end this section by introducing necessary notations. We denote by $f(\cdot)$ and $g(\cdot)$ Laplace–Stieltjes transforms (LST for short) of CDFs $F(\cdot)$ and $G(\cdot)$, respectively. Similarly, tails of CDFs $\overline{F}(\cdot)$ and $\overline{G}(\cdot)$ are denoted by $\overline{F}(\cdot)$ and $\overline{G}(\cdot)$, respectively. We define the $i$-fold Stieltjes transform of $G(\cdot)$ with itself as follows:

$$G^{0s}(t) = 1, \quad G^{is}(t) = \int_0^t G^{(i-1)s}(t - y) dG(y),$$

where $t > 0$ and $i \geq 1$.

Similarly, $p_k^{is}$ stands for the $j$th term of the $i$-fold convolution of the sequence $(p_k)$ with itself, namely:
\[ p_j^{0+} = I\{ j = 0 \}, \quad p_j^{i+} = \sum_{k=1}^{i} p_k^{(i-1)+} p_{j-k}, \]  

(2)

where \( i \geq 1 \), and \( I\{ \cdot \} \) is the indicator defined as:

\[ I\{ \mathcal{E} \} = \begin{cases} 1, & \text{if the event } \mathcal{E} \text{ occurs,} \\ 0, & \text{otherwise.} \end{cases} \]

(3)

Finally, let us denote by \( X(t) \) the number of packets present in the system at time \( t \) (including the one being processed at this time, if any).

3. Equations for Conditional Queue-Size Distribution

In this section, identifying Markovian moments in the evolution of the system, we build a system of integral equations for the transient queue-size distribution conditioned by the buffer state at the starting time \( t = 0 \).

We introduce the following notation:

\[ P_n(t, m) \overset{\text{def}}{=} P\{ X(t) = m \mid X(0) = n \}, \]

(4)

where \( t > 0 \) and \( 0 \leq m, n \leq N \).

Let us consider firstly the case of the system being empty at the starting epoch \( t = 0 \). In such a case we assume that an MVP begins at this moment, so we identify the starting moment of the busy period. Note that the continuous total probability law leads to the following integral equation (having in mind that \( \lambda > 0 \)):

\[
P_0(t, m) = \sum_{i=0}^{m} \int_{v=0}^{m} \int_{y=0}^{m} \lambda e^{-\lambda y} dy \int_{v=y}^{m} \left[ \sum_{k=1}^{N-1} p_k \sum_{j=0}^{N-k-1} \sum_{r=0}^{i} p_j^{r+} \frac{[\lambda(u+v+y)]^r}{r!} e^{-\lambda(u+v+y)} P_k+j(t-u-v,m) \right] \lambda e^{-\lambda y} \int_{v=y}^{m} G(v) \left( t-u-v \right) dv \int_{y=0}^{m} \lambda e^{-\lambda y} \tau(t-u) dy 
\]

(5)

where \( I\{ 1 \leq m \leq N - 1 \} \) is the indicator function. The first summand relates to the case in which there is at least one place free in the accumulating buffer at time \( u + v \) (this moment also coincides with the starting moment of the busy period). So, if \( k < N - 1 \) packets enter at time \( y \), then next, at most \( N - k - 1 \) packets arrive during a time period of length \( u + v - y \). The second summand describes the case of the buffer being saturated at the starting time of the busy period. The second square brackets on the right side of (5) represent the situation in which the MVP ends after time \( t \), however, at least one packet arrives before \( t \). If \( m \leq N - 1 \) then the number of arrivals up to the moment \( t \) equals \( m \) exactly (the first summand). In the case \( m = N \), the number of arrivals must be greater than or equal to \( N \) (the second summand).
Finally, the expression \( I\{m = 0\} e^{-\lambda t} \) corresponds to the case of no arrivals before time \( t \). In this case, \( X(t) = m \) if and only if \( m = 0 \).

Consider now the case where the buffer contains at least one packet at the starting moment \( t = 0 \). As it is well known, \( (X(t_1 +)) \), where \( t_1 < t_2 < \ldots \), are successive service completion (departure) epochs, is an embedded Markov chain (EMC) in the \( M^X/G/1 \)-type queue. The continuous version of the total probability law applied with respect to the first departure epoch after the starting moment \( t = 0 \) leads to the following system of integral equations:

\[
P_n(t, m) = \int_0^t \left[ \sum_{k=0}^{N-n-1} \sum_{r=0}^{k} p_k^r \frac{e^{-\lambda y}}{r!} e^{-\lambda t} p_{n+k-1}(t - y, m) \right. \]

\[
+ \left. P_{N-1}(t - y, m) \sum_{k=N-n}^{\infty} \sum_{r=0}^{k} p_k^r \frac{e^{-\lambda y}}{r!} e^{-\lambda y} \right] dF(y) \]

\[
+ \left( I\{n \leq m \leq N - 1\} \sum_{k=n-m}^{m-n} \frac{(\lambda t)^r}{r!} + I\{m = N\} \sum_{k=n-N}^{\infty} \sum_{r=0}^{k} p_k^r \frac{e^{-\lambda t}}{r!} \right) e^{-\lambda t} F(t),
\]

where \( \lambda > 0 \) stands for the group arrival intensity and \( n \in \{1, \ldots, N\} \).

Indeed, the integral on the right side of (6) relates to the situation in which the first packet leaves the system at time \( y < t \): the first summand in brackets corresponds to the case where there is at least one packet before that moment in the accumulating buffer; in the second summand, the buffer is saturated just before the first departure, so after the first service completion epoch the state of the system equals \( N - 1 \). The other two components after the integral on the right side of (6) represent the situation in which the first service ends after time \( t \).

Let us introduce the LT of \( P_n(t, m) \) as:

\[
\hat{P}_n(s, m) \overset{\text{df}}{=} \int_0^\infty e^{-st} P_n(t, m) dt, \quad \text{Re}(s) > 0,
\]

and define for \( s > 0 \) the following functional sequences:

\[
\alpha_k(s) \overset{\text{df}}{=} \int_0^{\infty} e^{-(s+\lambda)y} \sum_{r=0}^{k} p_k^r \frac{(\lambda y)^r}{r!} dF(y),
\]

\[
\beta_k(s) \overset{\text{df}}{=} \frac{1}{1 - G(s + \lambda)} \int_0^{\infty} e^{-(s+\lambda)y} \sum_{r=0}^{k} \frac{p_k^r (\lambda y)^{r+1}}{(r+1)!} dG(y),
\]

\[
\gamma(s) \overset{\text{df}}{=} \frac{G(s) - G(s + \lambda)}{1 - G(s + \lambda)} \sum_{k=N}^{\infty} \frac{p_k}{\lambda k}
\]

\[
\delta(s, m) \overset{\text{df}}{=} \frac{1}{1 - G(s + \lambda)} \left[ I\{1 \leq m \leq N - 1\} \sum_{k=1}^{m} p_k H^{(G)}(m - k, s) \right.
\]

\[
+ I\{m = N\} \left( \sum_{k=N}^{\infty} p_k H^{(G)}(s) + \sum_{k=1}^{N-1} p_k \sum_{j=N-k}^{\infty} H^{(G)}(j, s) \right)
\]

and

\[
\epsilon_k(s, m) \overset{\text{df}}{=} I\{n \leq m \leq N - 1\} H^{(F)}(m - k, s) + I\{m = N\} \sum_{k=N-k}^{\infty} H^{(F)}(k, s),
\]

where
where \( n \) will be relevant in the context of the method that we will use to obtain the solution. Similarly, Equation (16) can be rewritten as:

\[
H^{(G)}(k, s) \overset{\text{def}}{=} \int_{0}^{\infty} e^{-(s+\lambda)t} \sum_{r=0}^{k} \frac{p_{r}^{*} (\lambda t)^{r+1}}{(r+1)!} \mathcal{T}(t) dt,
\]

(13)

\[
H^{(F)}(k, s) \overset{\text{def}}{=} \int_{0}^{\infty} e^{-(s+\lambda)t} \sum_{r=0}^{k} \frac{p_{r}^{*} (\lambda t)^{r}}{r!} \mathcal{F}(t) dt,
\]

(14)

\[
H^{(G)}(s) \overset{\text{def}}{=} \int_{0}^{\infty} e^{-st} - e^{-(s+\lambda)t} \mathcal{T}(t) dt.
\]

(15)

Note that the system (5)–(6) leads now to the following corresponding one written for LTs:

\[
\hat{P}_{0}(s, m) = \sum_{k=1}^{N-1} p_{k} \left[ \sum_{j=0}^{N-k-1} \beta_{j}(s) \hat{P}_{k+j}(s, m) + \hat{P}_{N}(s, m) \sum_{j=N-k}^{\infty} \beta_{j}(s) \right]
\]

\[
+ \hat{P}_{N}(s, m) \gamma(s) + \delta(s, m) + \frac{I\{m = 0\}}{s + \lambda}
\]

(16)

and

\[
\hat{P}_{n}(s, m) = \sum_{k=0}^{N-n-1} a_{k}(s) \hat{P}_{n+k-1}(s, m) + \hat{P}_{N-1}(s, m) \sum_{k=N-n}^{\infty} a_{k}(s) + e_{n}(s, m),
\]

(17)

where \( n \in \{1, \ldots, N\} \).

We will now rewrite the last system of Equations (16) and (17) in a specific form that will be relevant in the context of the method that we will use to obtain the solution.

By introducing the following notation:

\[
\hat{Q}_{n}(s, m) \overset{\text{def}}{=} \hat{P}_{N-n}(s, m), \quad n \in \{0, \ldots, N\},
\]

(18)

Equation (17) will take the form:

\[
\sum_{k=-1}^{n} a_{k+1}(s) \hat{Q}_{n-k}(s, m) - \hat{Q}_{n}(s, m) = \psi_{n}(s, m),
\]

(19)

where \( n \in \{0, \ldots, N-1\} \) and

\[
\psi_{n}(s, m) \overset{\text{def}}{=} a_{n+1}(s) \hat{Q}_{0}(s, m) - \hat{Q}_{1}(s, m) \sum_{k=n+1}^{\infty} a_{k}(s) - e_{N-n}(s, m).
\]

(20)

Similarly, Equation (16) can be rewritten as:

\[
\hat{Q}_{N}(s, m) = \sum_{k=1}^{N-1} p_{k} \left[ \sum_{j=0}^{N-k-1} \beta_{j}(s) \hat{Q}_{N-k-j}(s, m) + \hat{Q}_{0}(s, m) \sum_{j=N-k}^{\infty} \beta_{j}(s) \right]
\]

\[
+ \hat{Q}_{0}(s, m) \gamma(s) + \delta(s, m) + \frac{I\{m = 0\}}{s + \lambda}.
\]

(21)

4. Main Analytical Result

In this section, we use Korolyuk’s potential approach to obtain the solution of the system (19) and (21) in a compact form. In [39], the following linear system with an infinite number of equations is considered:

\[
\sum_{k=-1}^{n} a_{k+1} q_{n-k} - q_{n} = \psi_{n}, \quad n \geq 0,
\]

(22)
where \((a_n)\) and \((\psi_n)\) are two known number sequences, and \((q_n)\) is the sequence of unknowns. Such a system has infinitely many solutions; however, as was proved in [39], each solution of (22) can be written as follows:

\[
q_n = LR_n + \sum_{k=1}^{n} R_{n-k} \psi_k, \quad n \geq 1, \tag{23}
\]

where \(L \in \mathbb{R}\) and, in general, it can be different for different solutions, and the sequence \((R_k)\), called a potential in [39], is defined by the sequence of coefficients \((a_n)\) in two equivalent ways. Firstly, its successive terms can be obtained recursively, one by one, in the following way:

\[
R_0 = 0, \quad R_1 = \frac{1}{a_0}, \quad R_k = R_1 \left( R_{k-1} - \sum_{j=0}^{k-1} a_{j+1} R_{k-1-j} \right), \tag{24}
\]

where \(k \geq 2\). Alternatively, successive terms of the sequence \((R_k)\) can be obtained by means of power expansion. Indeed, defining:

\[
\Omega(z) \overset{\text{def}}{=} \sum_{k=1}^{\infty} a_k z^k, \quad |z| < 1, \tag{25}
\]

one can easily show that

\[
\sum_{k=0}^{\infty} R_k z^k = \frac{z}{\Omega(z) - z}. \tag{26}
\]

In consequence, expanding the right side of (26) in the Maclaurin series and next comparing coefficients at successive powers of \(z\) on the left and right sides of (26), we obtain:

\[
R_k = \frac{1}{k!} \frac{d^k}{dz^k} \left( \frac{z}{\Omega(z) - z} \right) \bigg|_{z=0}. \tag{27}
\]

Note that the systems of Equations (19) and (22) have the same form. However, there are two main differences. Firstly, functions in the system (22) depend on some arguments: we have \(a_k = a_k(s), q_k = q_k(s, m)\) and \(\psi_n = \psi_n(s, m)\). Secondly, the number of equations in (19) compared to (22) is finite. To obtain the representation for the solution of (19), we can use the result (23) assuming now that \(L = L(s, m)\) and \(R_k = R_k(s)\). As it turns out, Equation (21) written for \(n = N\) will, however, serve as the boundary condition allowing for the explicit determination of \(L(s, m)\).

We prove the following main theorem:

**Theorem 1.** In the \(\mathcal{M}^X/G/1/N\)-type queueing model with multiple vacation policy the representation for the Laplace transform \(\hat{P}_n(s, m)\) of the conditional transient queue-size distribution is the following:

\[
\hat{P}_n(s, m) = \chi_{N-n}(s, m) + \Lambda_{N-n}(s) \left( \sum_{k=1}^{N-1} p_k \sum_{j=1}^{N-k} \beta_{N-k-j}(s) \chi_{j}(s, m) \right.
\]

\[
+ \delta(s, m) \left( 1 - \frac{1}{s + \lambda} \right) - \chi_N(s, m) \right)
\]

\[
\times \left( \Lambda_N(s) - \gamma(s) - \sum_{k=1}^{N-1} p_k \left( \sum_{j=1}^{N-k} \beta_{N-k-j}(s) \Lambda_j(s) + \sum_{j=N-k}^{\infty} \beta_j(s) \right) \right) \left( \chi_{N}(s, m) \right)^{-1}, \tag{28}
\]
where
\[ \Lambda_n(s) = \alpha_0(s) R_{n+1}(s) + \sum_{k=0}^{n} R_{n-k}(s) \left( \alpha_{k+1}(s) - \frac{1}{f(s)} \sum_{i=k+1}^{\infty} \alpha_i(s) \right) \] (29)
and
\[ \chi_n(s, m) = \sum_{k=0}^{n} R_{n-k}(s) \left( \frac{e_N(s, m)}{f(s)} \sum_{i=k+1}^{\infty} \alpha_i(s) - \epsilon_{N-k}(s, m) \right), \] (30)
the formulae for \( \alpha_k(s) \), \( \beta_k(s) \), \( \gamma(s) \), \( \delta(s, m) \), \( \epsilon_k(s, m) \) are given in (8), (9), (10), (11) and (12), respectively, and the sequence \( \{ R_k(s) \} \) is defined recursively as follows (compare (24)):
\[ R_0(s) = 0, \quad R_1(s) = \frac{1}{\alpha_0(s)}, \quad R_k(s) = R_1(s) \left( R_{k-1}(s) - \sum_{j=0}^{k-1} \alpha_{j+1}(s) R_{k-1-j}(s) \right), \] (31)
where \( k \geq 1 \).

**Proof.** Obviously, since (19) has the same form as (22), then each solution of the system (19), (21) can be written in the following way (compare (23)):
\[ \tilde{Q}_n(s, m) = L(s, m) R_{n+1}(s) + \sum_{k=0}^{n} R_{n-k}(s) \psi_k(s, m), \] (32)
where \( n \in \{0, \ldots, N\} \) and the sequence \( \{ R_k(s) \} \) is defined in (31).

To obtain the explicit representation for \( Q_n(s, m) \) we need the formula for \( L(s, m) \). Taking \( n = 0 \) in (32), we have:
\[ L(s, m) = \alpha_0(s) \tilde{Q}_0(s, m). \] (33)
Similarly, substituting \( n = 0 \) into (19), we get:
\[ \alpha_0(s) \tilde{Q}_1(s, m) + \alpha_1(s) Q_0(s, m) - \tilde{Q}_0(s, m) = \psi_0(s, m), \] (34)
where (see (20))
\[ \psi_0(s, m) = \alpha_1(s) \tilde{Q}_0(s, m) - \tilde{Q}_1(s, m) \sum_{k=1}^{\infty} \alpha_k(s) - \epsilon_N(s, m). \]
Due to the fact that \( \sum_{k=0}^{\infty} \alpha_k(s) = f(s) \) we have from (34)
\[ \tilde{Q}_1(s, m) = \frac{\tilde{Q}_0(s, m) - \epsilon_N(s, m)}{f(s)}. \] (35)

Using now (33) and (35) in (32), we obtain
\[ \tilde{Q}_n(s, m) = \alpha_0(s) R_{n+1}(s) \tilde{Q}_0(s, m) + \sum_{k=0}^{n} R_{n-k}(s) \left[ \alpha_{k+1}(s) \tilde{Q}_0(s, m) - \tilde{Q}_1(s, m) \sum_{i=k+1}^{\infty} \alpha_i(s) - \epsilon_{N-k}(s, m) \right] \]
\[ = \Lambda_n(s) \tilde{Q}_0(s, m) + \chi_n(s, m), \] (36)
where functions \( \Lambda_n(s) \) and \( \chi_n(s, m) \) are defined in (29) and (30), respectively.

Now let us use Equation (21) as a specific boundary condition in order to find the formulae for \( \tilde{Q}_0(s, m) \). Applying (36) in (21) gives:
\[
\hat{Q}_0(s, m) = \left( \sum_{k=1}^{N-1} p_k \sum_{j=1}^{N-k} \beta_{N-k-j}(s) \chi_j(s, m) + \delta(s, m) + \frac{I\{m = 0\}}{s + \lambda} - \chi_N(s, m) \right) \\
\cdot \left( \Lambda_N(s) - \gamma(s) - \sum_{k=1}^{N-1} p_k \sum_{j=1}^{N-k} \beta_{N-k-j}(s) \Lambda_j(s) + \sum_{j=N-k}^{\infty} \beta_j(s) \right)^{-1}.
\] (37)

Now, referring to (18) and (36), completes the proof. \[\square\]

5. Numerical Study

In this section, we investigate numerically the impact of the main “input” system parameters (like arrival intensity, service rate or mean vacation duration) on the queue-size distribution. To obtain the transient queue-size distribution from the formula (28) for particular system parameters we use the Gaver–Stehfest algorithm of the numerical Laplace transform inversion described in detail in [40,41] (see also [42,43]). In computations, we deal with a model described as follows:

- Poisson arrivals of packets of sizes 100 B;
- CDF of the processing time is a mixture of two exponential distributions (second-order hyper-exponential distribution) and is defined as:
  \[
  F(t) = \alpha \cdot (1 - e^{-\mu_1 t}) + (1 - \alpha) \cdot (1 - e^{-\mu_2 t}), \quad t > 0,
  \]
  where \(\alpha, \mu_1\) and \(\mu_2\) are given;
- CDF of a single server vacation has 2-Erlang distribution with fixed parameter \(\xi\), namely:
  \[
  G(t) = 1 - e^{-\xi t} (1 + \xi t), \quad t > 0;
  \]
- buffer size equals \(N = 10\) packets (so 1 kB).

A hyper-exponential probability distribution of single packet’s processing time can be used in modeling the situation in which a single packet (customer, task, message, etc.) can receive two types of service: normal (typical) and special, for which the average duration times differ significantly. The value of parameter \(\alpha\) indicates the fraction of incoming packets, which receive typical service that takes on average \(\mu_1^{-1}\) time units. The other packets are processed according to a special mode.

Since during the server idle mode (multiple vacation period) the service station can perform other tasks (such as e.g., background tasks of the CPU processor), the 2-Erlang distribution can be used to approximate a single vacation period duration: we can assume that it consists of two consecutive subperiods of random length. During the first of them, the task is prepared to run in the background, and its direct execution takes place in the second period.

The result obtained in Theorem 1 is of a general nature (i.e., CDFs \(F(\cdot), G(\cdot)\) and the sequence \((p_k)\) are general). However, in numerical computations we use the fact that the right side of (28) is a rational function of variable \(s\). If \(\hat{P}_n(s, m)\) is not a rational function of \(s\) we can use, for example, the classical technique of Padé approximation of a given Laplace transform by an appropriate rational function (see, e.g., [44]).

5.1. Impact of Service Rate

Let us investigate firstly the impact of the service rate. Consider three scenarios in which the parameters of the service time CDF are \(\alpha = 0.4\) and

\[
(\mu_1, \mu_2) \in \{ (960, 720); (780, 520); (320, 720) \}.
\]
Hence, $\mu$ is equal to 800, 600 and 480 packets/s that correspond to processing rates 640, 480 and 384 kb/s, respectively. Keeping constant the arrival intensity $\lambda = 600$ packets/s, we have the traffic load of the system $\varrho = 0.75, 1$ and 1.25, respectively. Assume that the distribution of the arriving group size is given by the sequence $p_k = \{0.5, 0.5, 0, \ldots\}$ so the mean group size equals 1.5. Moreover, let $\xi = 1000$ so the mean single vacation duration equals 0.002 s. In Figure 1, transient conditional distributions $P\{X(t) = 1 \mid X(0) = 0\}$ are visualized.

![Figure 1](image)

**Figure 1.** Impact of service rate on probability $P\{X(t) = 1 \mid X(0) = 0\}$.

Observe that the probability $P\{X(t) = 1 \mid X(0) = 0\}$ is the highest just after the opening of the system. This is due to the fact that, after starting, the server immediately goes into the vacation mode due to the lack of packets to handle.

5.2. Impact of Arrival Rate

Analyze now the impact of the arrival rate $\lambda$ on the transient queue-size distribution. Consider three scenarios in which $\lambda = 450, 600$ and 750 packets/s, which corresponds to data transfer rates of 360, 480 and 600 kb/s. The parameters of the CDF of the processing time are $\alpha = 0.6, \mu_1 = 680$ and $\mu_2 = 510$, which gives the mean service rate $\mu = 600$. The values of the traffic load for successive scenarios are 0.75, 1 and 1.25, respectively. Moreover, as previously, $\xi = 1000$ so the average single vacation duration equals 0.002 s. Taking $(p_k) = \{0.5, 0.25, 0.25, 0, \ldots\}$, in Figure 2 probabilities $P\{X(t) = 1 \mid X(0) = 0\}$ are presented.

As time passes, the probability decreases until the system reaches equilibrium. The nature of the decreasing probability depends on the arrival intensity: the higher the $\lambda$, the faster the system reaches equilibrium.
5.3. Impact of Initial Buffer State

Investigate now the impact of the initial buffer state \( n \) on the transient probability \( P\{X(t) = 1 \mid X(0) = n\} \) for \( n = 0, 5 \) and 10. Assume \( \lambda = 450 \) packets/s, which corresponds to the arrival rate 600 kb/s. Moreover, take the service rate \( \mu = 600 \) packets/s so the offered traffic load \( \rho \) equals 0.75. Taking the same parameter \( \xi = 1000 \) of the single vacation duration CDF and the sequence \( (p_k) = \{0.5, 0.25, 0.25, 0, \ldots\} \) describing sizes of arriving groups, the transient behavior of the considered probability is visualized in Figure 3.

As can be seen, for small values of time \( t \), the influence of the initial buffer state is clearly visible and the differences between corresponding probabilities are relatively large.
5.4. Impact of Single Vacation Duration

Finally, let us check the response of the transient queue-size distribution to changes in the average length of a single vacation. Consider three different scenarios in which the arrival intensity \( \lambda \) takes, as previously, values 450, 600 and 750 packets/s and the service rate \( \mu = 600 \) packets/s. In consequence, \( \varrho = 0.75, 1 \) and 1.25, respectively. In each scenario we analyze the behavior of the transient probability \( P\{X(t) = 1 \mid X(0) = 0\} \) for three different values of parameter \( \xi = \lambda v \) of the single vacation duration CDF, namely 500, 1000 and 1500 (corresponding to mean single vacation durations 0.004, 0.002 and 0.0013 s). The size of an arriving group of packets is distributed according to sequence \( (p_k) = \{0.5, 0.25, 0.25, 0, \ldots\} \) so the mean group size equals 1.75. Results of the experiment are presented in Figures 4–6.

**Figure 4.** Impact of single vacation duration on probability \( P\{X(t) = 1 \mid X(0) = 0\} \) for \( \lambda = 450 \) packets/s.

**Figure 5.** Impact of single vacation duration on probability \( P\{X(t) = 1 \mid X(0) = 0\} \) for \( \lambda = 600 \) packets/s.
Let us comment briefly on the final results. As one can observe, the longer the mean single vacation duration (the smaller the value of parameter $\xi = \lambda \nu$), the smaller the probability $P\{X(t) = 1 \mid X(0) = 0\}$. Indeed, longer vacations are conducive to the accumulation of more packets in the buffer before the start of the processing. This phenomenon is particularly visible at the lowest traffic load value $\varrho = 0.75$.

5.5. Numerical Computations vs. Simulation Results

In this subsection, we compare numerical results with the simulation model created in Objective Modular Network Testbed in C++ (OMNeT++) [45]. We modify the standard FIFO model imbedded in OMNeT++ discrete event simulator in a way to transmit and queue batches of packets according to the following scenarios:

- The transient probability $P\{X(t) = 1 \mid X(0) = 0\}$ is calculated for $\lambda = \mu = 600$ packets/s (resulting in $\varrho = 1$). We take $\xi = 1000$ as a parameter of the single vacation duration CDF (corresponding to mean single vacation duration 0.002 s) and the sequence $(p_k) = \{0.5, 0.25, 0.25, 0, \ldots\}$ describing sizes of arriving groups (so the mean group size equals 1.75);
- As in the previous scenario except for the sequence $(p_k) = \{1, 0, 0, \ldots\}$ (single arrivals).

The simulation model is extended by a time step of $t_c = 0.0005$ s, describing the time moments $k \cdot t_c = t \in [0, 0.06]$ for $k = 0, 1, 2, \ldots, 60$, at which the number of packets present in the system is measured. The simulation run is divided into 10,000 individual trials. Each attempt is a separate time period $t \in [0, 0.06]$ in which the random processing is simulated in accordance with the multiple vacation policy assumed in our queueing model. A unique seed is randomly selected for each sample and used to initialize the pseudorandom number generator representing random arrival and service as well as the lengths of individual vacation periods.

Figures 7 and 8 confirm the compliance of the numerical calculations with the simulation results. The maximum difference in the probability values between the numerical calculations and the simulation results measured at times $k \cdot t_c$ equals $0.00599106$. 

![Figure 6. Impact of single vacation duration on probability $P\{X(t) = 1 \mid X(0) = 0\}$ for $\lambda = 750$ packets/s.](image-url)
Figure 7. Comparison of the $P\{Y(t) = 1 \mid Y(0) = 0\}$ probability distribution for the compound Poisson process, obtained: (1) by numerical calculations using the formula (28)—black line on the graph; (2) as a statistical result of a 10,000th random sample—the red squares in the diagram.

Figure 8. Comparison of the $P\{Y(t) = 1 \mid Y(0) = 0\}$ probability distribution for the simple Poisson process, obtained: (1) by numerical calculations using the formula (28)—black line on the graph; (2) as a statistical result of a 10,000th random sample—the red squares in the diagram.

6. Conclusions

In this article, transient queue-size distribution in the batch-arrival system with multiple vacation policy is studied. For the purposes of the analysis, the approach based on the idea of the embedded Markov chain, continuous formula of total probability, integral equations and linear algebra is proposed. As the main result, the closed-form formula for the Laplace transform of the transient queue-size distribution conditioned by the initial
buffer state is obtained. In the numerical study, the impact of the service rate, arrival intensity, initial buffer state and mean single vacation duration on the studied characteristic is investigated for examples of the considered queueing model. The considered queueing model has many potential applications in modeling the operation of production systems, computer and telecommunication networks with an energy saving mechanism based on cyclic monitoring the state of the queue of packets (messages, tasks, etc.) waiting for service. During the multiple vacation period, the service station may perform other types of tasks, such as, for example, the time the CPU is running in the background, installing software updates or maintaining the server. In applications in modeling the operation of an automated production line, during downtime, a single machine (service station) can be used in secondary production; during this time, the machine can also be calibrated or the settings can be changed. Due to the fact that, in some practical applications, interarrival times are not exponentially distributed, it is planned in future to extend the analysis for the case of a non-Poisson arrival stream (e.g., hyper-exponential or a general renewal one).

Author Contributions: Conceptualization, W.M.K. and R.M.; methodology, W.M.K.; software, R.M.; validation, W.M.K. and R.M.; formal analysis, W.M.K. and R.M.; investigation, W.M.K. and R.M.; resources, W.M.K. and R.M.; data curation, R.M.; writing—original draft preparation, W.M.K. and R.M.; writing—review and editing, W.M.K. and R.M.; visualization, R.M.; supervision, W.M.K.; project administration, W.M.K.; funding acquisition, W.M.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Silesian University of Technology.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are available upon reasonable request. Additional data (beyond those included in the main text are available from the corresponding author upon request.

Conflicts of Interest: The authors declared that there was no conflict of interest.

References
1. Yadin, M.; Naor, P. Queueing system with a removable service station. Oper. Res. Quart. 1963, 14, 393–405. [CrossRef]
2. Heyman, D.P. The T-policy for the M/G/1 queue. Manag. Sci. 1977, 23, 775–778. [CrossRef]
3. Levy, Y.; Yechiali, U. Utilization of idle time in an M/G/1 queueing system. Manag. Sci. 1975, 22, 202–211. [CrossRef]
4. Lee, H.W.; Lee, S.S.; Chae, K.C. Operating Characteristics of M^X/G/1 queue with N-policy. Queuing Syst. 1994, 15, 387–399. [CrossRef]
5. Levy, Y.; Yechiali, U. An M/M/c queue with servers vacations. INFOR 1976, 14, 163. [CrossRef]
6. Jain, M.; Kaur, S.; Singh, P. Supplementary variable technique (SVT) for non-Markovian single server queue with service interruption (QSI). Oper. Res. 2019, 21, 2203–2246. [CrossRef]
7. Liu, J.; Wang, J. Strategic joining rules in a single server Markovian queue with Bernoulli vacation. Oper. Res. 2017, 17, 413–434. [CrossRef]
8. Choudhury, G.; Deka, M. A batch arrival unreliable server delaying repair queue with two phases of service and Bernoulli vacation under multiple vacation policy. Qual. Technol. Quant. M. 2018, 15, 157–186. [CrossRef]
9. Niu, Z.; Takahashi, Y. A finite-capacity queue with exhaustive vacation/close-down/setup times and Markovian arrival processes. Queuing Syst. 1999, 31, 1–23. [CrossRef]
10. Niu, Z.; Shu, T.; Takahashi, Y. A vacation queue with setup and close-down times and batch Markovian arrival processes. Perform. Eval. 2003, 54, 225–248. [CrossRef]
11. Horvath, G.; Saffer, Z.; Telek, M. Queue length analysis of a Markov-modulated vacation queue with dependent arrival and service processes and exhaustive service policy. J. Ind. Manag. Optim. 2017, 13, 1365–1381. [CrossRef]
12. Peng, Y.; Yang, X.-Q. On a BMAP/G/1 G-queue with setup times and multiple vacations. Acta Math. Appl. Sin. Engl. Ser. 2011, 27, 625–638. [CrossRef]
13. Ke, J.-C.; Huang, H.-I.; Chu, Y.-K. Batch arrival queue with N-policy and at most J vacations. Appl. Math. Model. 2010, 34, 451–466. [CrossRef]
14. Marek, D.; Doamański, A.; Doamańska, J.; Szygula, J.; Czachórski, T.; Klamka, J. Diffusion model of a non-integer order PI gamma controller with TCP/UDP streams. Entropy 2021, 23, 619. [CrossRef]
15. Baba, Y. Analysis of a G1/M/1 queue with multiple working vacations. Oper. Res. Lett. 2005, 33, 201–209. [CrossRef]
16. Chae, K.C.; Kim, S.J. Busy period analysis for the $G_1/M/1$ queue with exponential vacations. *Oper. Res. Lett.* 2007, 35, 114–118. [CrossRef]

17. Goswami, V.; Mund, G.B. Analysis of discrete-time queues with batch renewal input and multiple vacations. *J. Syst. Sci. Complex.* 2012, 25, 486–503. [CrossRef]

18. Kim, S.J.; Kim, N.K.; Park, H.-M. On the discrete-time Geo$(X)/G/1$ queues under $N$-policy with single and multiple vacations. *J. Appl. Math.* 2013, 2013, 587163. [CrossRef]

19. Takagi, H. $M/G/1/N$ queues with server vacations and exhaustive service. *Oper. Res.* 1994, 42, 926–939. [CrossRef]

20. Yu, M.; Alfa, A.S. Simple method to obtain the stochastic decomposition structure of the busy period in Geo/Geo/1/N vacation queue. *4OR* 2015, 13, 361–380. [CrossRef]

21. Doshi, B.T. Queueing systems with vacations-a survey. *Queueing Syst.* 1986, 1, 29–66. [CrossRef]

22. Takagi, H. *Queueing Analysis: A Foundation of Performance Evaluation, Vacation and Priority Systems, Part I, Volume I*; North-Holland: Amsterdam, The Netherlands, 1991.

23. Tian, N.; Zhang, Z.G. *Vaccation Queueing Models: Theory and Applications*; Springer: Berlin/Heidelberg, Germany, 2006.

24. Wang, S.; Xu, X. Equilibrium strategies of the fluid queue with working vacation. *Oper. Res.* 2019, 21, 1211–1228. [CrossRef]

25. Xu, B.; Xu, X. Equilibrium strategic behavior of customers in the $M/M/1$ queue with partial failures and repairs. *Oper. Res.* 2018, 18, 273–292. [CrossRef]

26. Bounkhel, M.; Tadj, L.; Hedjar, R. Steady-state analysis of a flexible Markovian queue with server breakdowns. *Entropy* 2019, 21, 259. [CrossRef]

27. Ma, Z.; Li, C.; Wang, P. Analysis of $G$-queue with pseudo-fault and multiple Working vacations. *J. Syst. Sci. Complex.* 2020, 33, 1144–1162. [CrossRef]

28. Nandy, N.; Pradhan, S. Stationary joint distribution of a discrete-time group-arrival and batch-size-dependent service queue with single and multiple vacation. *Commun. Stat. Theory* 2021. [CrossRef]

29. Sun, W.; Li, S.; Cheng-Guo, E. Equilibrium and optimal balking strategies of customers in Markovian queues with multiple vacations and N-policy. *Appl. Math. Model.* 2016, 40, 284–301. [CrossRef]

30. Yang, B.; Hou, Z.; Wu, J. Analysis of the equilibrium strategies in the $Geo/Geo/1$ queue with multiple working vacations. *Qual. Technol. Quant. M.* 2018, 15, 663–685. [CrossRef]

31. Yang, D.-Y.; Chen, Y.-H.; Wu, C.-H. Modelling and optimisation of a two-server queue with multiple vacations and working breakdowns. *Int. J. Prod. Res.* 2020, 58, 3036–3048. [CrossRef]

32. Wang, Z.; Liu, L.; Shao, Y.; Zhao, Y.Q. Joining strategies under two kinds of games for a multiple vacations retrial queue with N-policy and breakdowns. *AIMS Math.* 2021, 6, 9075–9099. [CrossRef]

33. Kim, B.; Kim, J.; Bueker, O. Non-preemptive priority $M/M/m$ queue with servers’ vacations. *Comput. Ind. Eng.* 2021, 160, 107390. [CrossRef]

34. Tian, R.; Skouri, K. Social optimization and pricing strategies in unobservable queues with delayed multiple vacations. *Math. Probl. Eng.* 2019, 2019, 4684957. [CrossRef]

35. Vijayashree, K.V.; Janani, B. Transient analysis of an $M/M/1$ queue with multiple exponential vacation and N-policy. *Pak. J. Stat. Oper. Res.* 2015, 11, 587–600. [CrossRef]

36. Vadvivakarasi, M.; Kalidass, K. Discussion on the transient behaviour of single server Markovian multiple variant vacation queues. *Oper. Res. Dec.* 2021, 31, 123–146.

37. Kempa, W.M. Analysis of departure process in batch arrival queue with multiple vacations and exhaustive service. *Commun. Stat. Theory* 2011, 40, 2856–2865. [CrossRef]

38. Kempa, W.M. On transient queue-size distribution in the batch-arrivals system with a single vacation policy. *Kybernetika* 2014, 50, 126–141. [CrossRef]

39. Korolyuk, V.S. Boundary-value problems for compound Poisson processes. *Theor. Prob. Appl.* 1974, 19, 1–13. [CrossRef]

40. Abate, J.; Choudhury, G.L.; Whitt, W. An introduction to numerical transform inversion and its application to probability models. In *Computational Probability*; Grassmann, W.K., Ed.; Springer: Boston, FL, USA, 2000; pp. 257–323.

41. Abate, J.; Whitt, W. A unified framework for numerically inverting Laplace transforms. *INFORMS J. Comput.* 2006, 18, 408–421. [CrossRef]

42. Gaver, D.P. Observing stochastic processes, and approximate transform inversion. *Oper. Res.* 1966, 13, 444–459. [CrossRef]

43. Stehfest, H. Algorithm 368: Numerical inversion of Laplace transforms. *Commun. ACM* 1970, 13, 47–49. [CrossRef]

44. Singh, G.; Gupta, U.C.; Chaudhry, M.L. Analysis of queuing-time distributions for MAP/D$_N$/1 queue. *Int. J. Comp. Math.* 2014, 94, 1911–1930. [CrossRef]

45. OMNeT++ Discrete Event Simulator. Available online: [https://omnetpp.org](https://omnetpp.org) (accessed on 12 October 2021).