On the joint distribution of an infinite-buffer discrete-time batch-size-dependent service queue with single and multiple vacations

Nilanjan Nandy and Sourav Pradhan

Department of Mathematics, Visvesvaraya National Institute of Technology, Nagpur, India

ABSTRACT

Due to the widespread applicability of discrete-time queues in wireless networks or telecommunication systems, this paper analyzes an infinite-buffer batch-service queue with single and multiple vacations where customers/messages arrive according to the Bernoulli process and service time varies with the batch-size. The foremost focal point of this analysis is to get the complete joint distribution of queue length and server content at the service completion epoch, for which first the bivariate probability generating function has been derived. We also acquire the joint distribution at an arbitrary slot. We also provide several marginal distributions and performance measures for the utilization of the vendor. During the transmission of data through a particular communication channel, high transmission error may take place due to several factors. For this reason, one may skip the transmission through that particular channel. However, the discrete phase-type distribution plays a noteworthy role to control this transmission error which eventually motivates us to include a numerical example where service time distribution follows discrete phase-type distribution. A comparison between batch-size-dependent and independent services has been drawn through the graphical representation of some performance measures and total system cost.

1. Introduction

In recent years, Broadband Integrated Services Digital Network (B-ISDN) based on Asynchronous Transfer Mode (ATM) technology, IEEE802.11 n WLANs, circuit-switched time-division multiple access (TDMA) systems, etc. have received abound attention of the researchers mainly due to its compatibility of providing a common interface for multimedia services. The underlying mechanism of those systems is completely based on the packet switching principle in which the transmission of packets is allowed only at regularly spaced points in time generally called as a slot. Consequently, those systems can be adequately modeled and analyzed using discrete-time queues which have a notable diverse spectrum of applications in modern telecommunication/wireless systems, see, e.g. Bruneel and Kim (1993), Alfa (2010), Samanta et al. (2007b), Gupta et al. (2014), (2007), and Claeyts et al. (2010a) and references therein.

Batch-service queues are ubiquitous in several practical situations such as automatic manufacturing technology (very large-scale integrated (VLSI) circuits), blood pooling, ovens in a manufacturing system, mobile crowdsourcing app for smart cities, recreational devices in amusement park, etc. A tremendous research is rapidly growing on batch-service queues for the betterment of human civilization, see, e.g. Chaudhry and Gupta (2003), Ho Chang et al.
(2004), Goswami et al. (2006), Goswami and Vijaya Laxmi (2011), Germs and van Foreest (2013), Barbhuiya and Gupta (2019), and Bank and Samanta (2020) and references therein. In recent times, a few researchers have focused on batch-service queues with batch-size-dependent service due to their usefulness in production and transportation, package delivery, group testing of blood/urine samples, etc. For more detail see Pradhan and Gupta (2020); Pradhan et al. (2016), Claey’s et al. (2010b, 2011).

For the discrete-time batch-service

\[ K(r)(x) = S_r^\tau(\lambda + \lambda x) \]

and D-BMAP + \[ \sum_{n=b+1}^{\infty} p_n^+ K_n^{(b)}(x) x^{n-b} y^b + \sum_{n=b+1}^{\infty} q_n^+ K_n^{(b)}(x) x^{n-b} y^b \]

Claey’s et al. (2013a, 2011) derived joint probability/vector generating function (pgf/vgf) of queue and server content at arbitrary slot. Claey’s et al. (2013b) also focused on the analysis of tail probabilities for the customer delay. They also extended their investigation on the influence of the correlation of the arrival process on the behavior of the system. However, it should be noted here that the complete extraction procedure of the joint distribution was lacking. Banerjee et al. (2014) provided the complete joint distribution of queue and server content for a discrete-time finite-buffer

\[ P^+(x, y) = (1 - \delta) \sum_{n=0}^{a-1} q_n^+ K_n^{(a)}(x) y^n + \sum_{n=a}^{b} p_n^+ K_n^{(b)}(x) x^n + \sum_{n=a}^{b} q_n^+ K_n^{(b)}(x) y^n \]

queue in which they reported the queue length and server content distribution together using both the embedded Markov chain technique (EMCT) and the quasi birth and death (QBD) process. It is worthwhile to mention here that these above analysis does not focus on any type of vacation.

In several real-life circumstances, it is observed that after the service completion of a batch, the server may be unavailable for a random period of time if the server finds less number of customers than the minimum threshold. The server may utilize this time in order to carry out some additional work. These types of queues are known as vacation queues which have potential applications in polling protocols frequently used in high-speed telecommunication networks, designing of local area networks, processor schedules in computer and switching systems, shared resources maintenance, manufacturing system with server breakdown, etc. The literature clearly exhibits the extensive studies on vacation queues, for example, see Doshi (1986), Lee et al. (1994), Gupta and Sikdar (2006), Banik et al. (2006), Tadj et al. (2006), Ke (2007), Tadj and Ke (2008), Vijayashree and Anjuka (2018), and Ke et al. (2019) and the references therein. Ke et al. (2010) provided a brief and concise survey on the recent developments of vacation queues with different vacation rules such as single/multiple vacations, Bernoulli vacation, multiple adaptive vacations, working vacation, etc. They have also provided some literatures on multi-server vacation models. Based on the supplementary variable technique (SVT) and the difference equation method Barbhuiya and Gupta (2020) analyzed an infinite buffer batch arrival GI^X/M/1 queue with N threshold policy in which as soon as the system becomes empty, the server enters into an idle phase. The service will be resumed when the number of customers in the queue reaches N or more. They have obtained the probability distribution of the system-size and waiting time in the steady-state.

Moreover several studies include the different types of vacation policies in batch-service queueing models for continuous- and discrete-time set-up. Lee et al. (1992) obtained pgf of the queue length distribution for a batch service queue with a single vacation policy. Using difference operator Choi and Han (1994) derived the queue length distribution for a queueing system with general arrival and multiple vacations. A vacation queueing model with a fixed batch-size rule had been discussed by Lee et al. (1996) using SVT. In a series of papers, Reddy and Anitha (1998); Reddy and Anitha et al. (1999, 1998) derived pgfs of queue length distribution for batch-service queues with different vacation policies. The analysis of queueing models with vacation and close-down time has been carried out by Arumuganathan and Jeyakumar (2004, 2005). The finite- and infinite-buffer bulk queues with single/multiple vacation rules have been discussed by Gupta and Sikdar (2004, 2005); Sikdar and Gupta (2005a, b), where they employed SVT to compute the queue length.
distributions at different epochs. Jeyakumar and Arumuganathan (2011); Jeyakumar and Senthilnathan (2012) discussed batch-service queues with vacation and control policy on request for re-service, and close-down time with server breakdown, respectively. System cost analysis with Markovian arrival process, \((a, b)\) rule, multiple vacations, and close-down times have been carried out by Vadivu and Arumuganathan (2015). On the other hand, Chang and Choi (2005) analyzed a discrete-time finite-buffer Geo\(^X\)/\(G\)/1 vacation queue with the random serving capacity rule, where they focused on the steady-state queue length distribution. Employing both EMCT and SVT, Samanta et al. (2007a) discussed a discrete-time group-arrival queue with single and multiple vacations where service is provided according to the general bulk service \((a, b)\) rule. It should be noted here that none of these analyses provides the server content distribution which is an essential tool to increase the serving capability of the server. Moreover, no one focuses on batch-size-dependent service policy. In recent times, G. K. Gupta et al. (2019) have focused on the joint distribution for a finite-buffer batch-size-dependent service queue with single and multiple vacations where vacation time depends on queue length in continuous-time set-up.

However, to the best of the authors’ knowledge, the joint queue and server content distribution and performance measures for an infinite-buffer discrete-time batch-size-dependent batch-service queue with single and multiple vacations are not available so far in the literature. From the designing and applicability perspective, one of the major concerns is to achieve the most cost-effective combination of performance or reliability of the system output so that the server utilization and processing rate can be met together. The server content distribution plays a noteworthy role to calculate the average number of customers/packets with the server by which the server’s capability can be maximized. With this information, one can focus on the optimal control of the concerned queue at the pre-implementation stage so that total system cost can be reduced. Moreover, a batch-size-dependent service mechanism is a powerful tool to reduce the congestion arising in digital communication systems and to avoid the excessive delays of the transmission of packets.

Getting motivated from the above perspective, in this paper, we analyze an infinite-buffer discrete-time queue with the Bernoulli arrival process, general bulk service \((a, b)\) rule, single and multiple vacations, and batch-size-dependent service policy where our center of focus is on joint queue and server content distribution. The practical application of this system and detailed model description are depicted in the subsequent sections. It is worthwhile to mention here that the inclusion of single and multiple vacations together along with batch-size-dependent service makes the mathematical analysis much more complex. Firstly, the steady-state governing equations of the concerned model have been developed. Secondly, we move toward our foremost focal point for the derivation of the bivariate pgf of queue and server content distribution together at the service completion epoch. Employing the partial fraction technique, the joint distribution has been successfully extracted and presented in a quite simple and elegant form. The probabilities at the arbitrary slot are acquired using the service completion/post transmission epoch probabilities. In order to bring out the essence of this system to the system designer/vendor, several marginal distributions along with pivotal performance measures are also provided. At the endmost, some assorted numerical examples and graphical observations are included to show the usefulness of the analytic procedure which can be fruitful to the readers. In one example, the service time follows the discrete phase-type distribution as it significantly monitors the transmission error occurring in the telecommunication system. We sketch a comparison between batch-size dependent and independent models through some graphical observations.

As a counterpart of discrete-time, in continuous-time set-up, Banerjee and Gupta (2012), Banerjee et al. (2015), (2013), Pradhan (2020b), and Pradhan and Gupta (2019) have analyzed some finite- and infinite-buffer batch-size-dependent batch-service queues wherein they provided complete joint distribution of queue and server content at different epochs employing EMCT as well as supplementary variable technique. However, in their analysis, no vacation policy has been incorporated.
The rest portion of this paper is organized as follows: the next section provides a practical application of the concerned model while Section 3 describes the model in detail. In Section 4, system equations are derived in the steady-state and Section 5 provides the complete procedure for getting joint distribution of queue and server content at service completion epoch. A relationship between the probabilities at service completion and arbitrary epoch is established in Section 6. Some special cases are provided in Section 7. Marginal distributions along with performance measures are sketched in the subsequent section. Several numerical results are appended in Section 9 followed by the final conclusion.

2. Practical application of the suggested queueing system

The above queueing system can be suitably used to model several practical scenarios. One such example is sketched here. The messages, data, images, videos, signals are first broken into manageable information packets. In ATM multiplexing and switching technology, IEEE802.11 n WLANs, internet protocol or ethernet, etc., those packets are transmitted during a fixed slot as a single entity through a common interface with a minimum threshold and maximum limit which may be thought of as general bulk service \((a, b)\) rule. The main advantage is the efficiency and improvement of the quality of service as the construction of only one header per aggregated batch instead of one header per single information unit reduces the time as well as the cost. Whenever the minimum number of packets are not available for the transmission, the multiplexer enters into the random vacation period with well pre-defined rule and that time may be used to execute some subsidiary works. In order to prevent the congestion in overload control telecommunication and to bypass the transmission errors, excessive delays of the time required to transmit the packets, the processing/transmission rate dynamically varies with the size of the packets which is exactly the batch-size-dependent service time policy.

3. Model description

- *Discrete-time set-up*: Let the time axis be slotted in intervals of equal length where the length of each slot is unity. More specifically, let us assume that the time axis be marked by \(0, 1, 2, \ldots, k, \ldots\). Here we discuss the model for late arrival delayed access system (LAS-DA) and therefore a potential arrival takes place in \((k−, k)\) and a potential batch departure occurs in \((k, k + )\). It may be noted here that under this discrete-time set-up, an arrival and a departure/transmission may take place simultaneously at a slot boundary. Different epochs of the LAS-DA system (which were discussed by Hunter (1983), Gravey et al. (1990)) are presented pictorially in Figure 1.

- *Arrival process*: We assume that the customers/messages in terms of packets arrive according to the Bernoulli process with parameter \(\lambda\) as it is analytically tractable and does not possess any fractal features like self-similarity. Hence, the inter-arrival times of the customers/messages are independent and geometrically distributed with probability mass function (pmf) \(\psi_n = \lambda^{n-1} \lambda, 0 < \lambda < 1, n \geq 1\) and \(\lambda = 1 - \lambda\).

- *Batch-service rule*: The packets are transmitted in group/batches according to the general bulk service \((a, b)\) rule which was introduced by Neuts (1967). The server only starts the service if the queue contains at least as many customers as the service threshold \(‘a’\). For the queue size \(r\) \((a \leq r \leq b)\), the entire group of customers is taken for the service. When the queue size exceeds \(‘b’\), then the server can transmit maximum \(‘b’\) packets and others remain in the queue for the next round of service. It is worthwhile to mention here that a newly arriving customer/message cannot join the ongoing service even if there is a free capacity.

- *Service process*: In most of the practical scenarios, it is observed that the transmission/processing times do not follow any specific well-known probability distribution. Keeping this in mind, here it is assumed that the service times are generally distributed so that a large class of probability...
distribution can be covered. We also assume that the service time depends on the batch-size under service as this service mechanism plays a pivotal role to reduce congestions and to improve the productivity of the system. Let us define the random variable \( T_i, a \leq i \leq b \), as the service time of a batch of ‘i’ customers with pmf \( s_i(n) = Pr(T_i = n), n = 1, 2, 3, \ldots \) with corresponding pgf \( S^*_i(z) = \sum_{n=1}^{\infty} s_i(n)z^n \), the mean service time \( s_i = \frac{1}{\mu_i} = S^*_i(1), a \leq i \leq b \), where \( S^*_i(1) \) is the first-order derivative evaluated at \( z = 1 \).

- **Vacation process**: The concerned queueing model is analyzed with two types of vacation rules viz. single vacation and multiple vacations using an indicator variable \( \delta_p \) as follows:

\[
\delta_p = \begin{cases} 
0, & \text{for single vacation} \\
1, & \text{for multiple vacation.}
\end{cases}
\]

It has to be kept in mind that the server must decide the pre-defined vacation rule before the service initiation. The results for the corresponding queue with a single vacation can be obtained by substituting \( \delta_p = 0 \) and that with multiple vacations by substituting \( \delta_p = 1 \).

- **Single vacation rule**: After the completion of a batch-service, if the server finds at least ‘a’ customers waiting, it continues the service process; otherwise, it leaves for a vacation of random length with pmf \( v_n, n \geq 1 \), pgf \( V^*(z) = \sum_{n=1}^{\infty} v_n z^n \), and mean \( E(V) = \bar{v} = V^*(1) \). At the vacation termination epoch, if the queue length is still less than the threshold value ‘a’, then the server remains dormant till the minimum threshold has been accumulated.

- **Multiple vacation rule**: After the completion of a batch-service, if there are at least ‘a’ customers waiting in the queue, it continues the service. If the queue length is less than ‘a’, then the server leaves for a vacation of random time. After returning from the vacation, if still ‘a’ customers have not been accumulated then the server leaves for another vacation and so on until at least ‘a’ customers waiting in the queue. The length of random vacation time is generally distributed with pmf \( v_n, n \geq 1 \), pgf \( V^*(z) = \sum_{n=1}^{\infty} v_n z^n \), and mean \( E(V) = \bar{v} = V^*(1) \).

- **Traffic intensity**: The traffic intensity of the system is \( \rho = \frac{1}{b_{pk}} < 1 \), which ensures the stability of the system.

### 4. Steady-state governing system equations

In this section, our main goal is to develop the governing equations of the concerned model in the steady-state. Let us first define the states of the system at time \( t \) as:

- \( N_q(k-) \) = Number of packets/customers in the queue waiting to be transmitted,
• $N_s(k-) = \text{Number of packets/customers with the server}$,
• $U(k-) = \text{Remaining service time of a batch in service (if any)}$,
• $V(k-) = \text{Remaining vacation time of the server (if any)}$,
• $\zeta(k-) = \text{State of the server defined as:}$

$$\zeta(k-) = \begin{cases} 2 & \text{when the server is busy}, \\ 1 & \text{when the server is on vacation}, \\ 0 & \text{when the server is in the dormant state}. \end{cases}$$

Further, let us define the joint probabilities as:

$$p_{n,0}(k-) = \Pr \{N_q(k-) = n, N_s(k-) = 0, \zeta(k-) = 0\},$$

$$p_{n,r}(u,k-) = \Pr \{N_q(k-) = n, N_s(k-) = r, U(k-) = u, \zeta(k-) = 2\},$$

$$u \geq 1, \ n \geq 0, \ a \leq r \leq b.$$  

$$q_{n}(u,k-) = \Pr \{N_q(k-) = n, V(k-) = u, \zeta(k-) = 1\}, \quad u \geq 1, \ n \geq 0.$$  

As the model is analyzed in the steady state, let us define the limiting probabilities as:

$$p_{n,0} = \lim_{k-\to\infty} p_{n,0}(k-),$$

$$p_{n,r}(u) = \lim_{k-\to\infty} p_{n,r}(u,k-), \ u \geq 1, \ n \geq 0, \ a \leq r \leq b.$$  

$$q_{n}(u) = \lim_{k-\to\infty} q_{n}(u,k-), \ u \geq 1, \ n \geq 0.$$  

Observing the states of the system at the epochs $k-$ and $(k+1)-$, and using the supplementary variable technique (SVT), we obtain the following equations in the steady state:

$$p_{0,0} = (1 - \delta_p)(\lambda_0,0 + (1 - \delta_p)\lambda q_0(1))$$

$$p_{n,0} = (1 - \delta_p)(\lambda p_{n,0} + \lambda p_{n-1,0} + \lambda q_n(1) + \lambda q_{n-1}(1)),$$  

$$\quad 1 \leq n \leq a - 1$$

$$p_{0,a}(u) = \lambda_0 p_{0,a}(u + 1) + \lambda \sum_{m=a}^{b} p_{a,m}(1)s_a(u) + \lambda \sum_{m=a}^{b} p_{a,m}(1)s_a(u) + \lambda q_{a-1}(1)s_a(u) + \lambda(1 - \delta_p)p_{a-1,0}s_a(u)$$

$$\quad + \lambda q_{a-1}(1)s_a(u) + \lambda(1 - \delta_p)p_{a-1,0}s_a(u)$$

$$p_{0,r}(u) = \lambda_0 p_{0,r}(u + 1) + \lambda \sum_{m=a}^{b} p_{r,m}(1)s_r(u) + \lambda \sum_{m=a}^{b} p_{r,m}(1)s_r(u)$$

$$\quad + \lambda q_{r-1}(1)s_r(u) + \lambda q_{r-1}(1)s_r(u), \quad a + 1 \leq r \leq b$$

$$p_{n,r}(u) = \lambda p_{n,r}(u + 1) + \lambda p_{n-1,r}(u + 1), \quad n \geq 1, \quad a \leq r \leq b - 1$$
\[ p_{n,b}(u) = \lambda p_{n,b}(u + 1) + \lambda p_{n-1,b}(u + 1) + \lambda \sum_{m=a}^{b} p_{n+b,m}(1)s_b(u) \]

\[ + \lambda \sum_{m=a}^{b} p_{n+b-1,m}(1)s_b(u) + \lambda q_{n+b}(1)s_b(u) + \lambda q_{n+b-1}(1)s_b(u), \quad n \geq 1 \]  \hspace{1cm} (6)

\[ q_0(u) = \lambda q_0(u + 1) + \lambda \sum_{m=a}^{b} p_{0,m}(1)v(u) + \delta_p \lambda q_0(1)v(u) \]  \hspace{1cm} (7)

\[ q_n(u) = \lambda q_n(u + 1) + \lambda q_{n-1}(u + 1) + \lambda \sum_{m=a}^{b} p_{n,m}(1)v(u) + \lambda \sum_{m=a}^{b} p_{n-1,m}(1)v(u) \]

\[ + \delta_p [\lambda q_n(1) + \lambda q_{n-1}(1)]v(u), \quad 1 \leq n \leq a - 1 \]  \hspace{1cm} (8)

\[ q_n(u) = \lambda q_n(u + 1) + \lambda q_{n-1}(u + 1), \quad n \geq a. \]  \hspace{1cm} (9)

Further, let us define

\[ p^*_n(z) = \sum_{u=1}^{\infty} p_{n,r}(u)z^u, \quad n \geq 0, \quad a \leq r \leq b \]

\[ q^*_n(z) = \sum_{u=1}^{\infty} q_n(u)z^u, \quad n \geq 0. \]

Consequently, the two results provided below immediately follow which will be used in the analysis.

\[ p_{n,r} = p^*_n(1) = \sum_{u=1}^{\infty} p_{n,r}(u), \quad a \leq r \leq b, \quad n \geq 0 \]

\[ q_n = q^*_n(1) = \sum_{u=1}^{\infty} q_n(u), \quad n \geq 0. \]

Our foremost focal point is to get the joint distributions from (1) – (9) for which first we take the \( z \)-transform of those equations. Multiplying (3) – (9) by \( z^u \) and adding over \( u \) from 1 to \( \infty \), we obtain

\[ \left( \frac{z - \lambda}{z} \right) p_{0,a}^*(z) = \lambda \sum_{m=a}^{b} p_{a,m}(1)S^*_a(z) + \lambda \sum_{m=a}^{b} p_{a-1,m}(1)S^*_a(z) + \lambda q_a(1)S^*_a(z) \]

\[ + \lambda q_{a-1}(1)S^*_a(z) + (1 - \delta_p)\lambda p_{a-1,0}S^*_a(z) - \lambda p_{0,a}(1) \]  \hspace{1cm} (10)

\[ \left( \frac{z - \lambda}{z} \right) p_{0,r}^*(z) = \lambda \sum_{m=a}^{b} p_{r,m}(1)S^*_r(z) + \lambda \sum_{m=a}^{b} p_{r-1,m}(1)S^*_r(z) + \lambda q_r(1)S^*_r(z) \]

\[ + \lambda q_{r-1}(1)S^*_r(z) - \lambda p_{0,r}(1), \quad a + 1 \leq r \leq b \]  \hspace{1cm} (11)
\[
\left( \frac{z - \bar{\lambda}}{z} \right) \mathcal{P}_{n,r}^+(z) = \frac{\lambda}{z} \mathcal{P}_{n-1,r}^+(z) - \bar{\lambda} \mathcal{P}_{n,r}(1) - \lambda \mathcal{P}_{n-1,r}(1), \quad n \geq 1, \ a \leq r \leq b - 1
\] (12)

\[
\left( \frac{z - \bar{\lambda}}{z} \right) \mathcal{P}_{n,b}^+(z) = \frac{\lambda}{z} \mathcal{P}_{n-1,b}^+(z) + \bar{\lambda} \sum_{m=a}^{b} \mathcal{P}_{n+b,m}(1) \mathcal{S}_{b}^+(z) + \lambda \sum_{m=a}^{b} \mathcal{P}_{n+b-1,m}(1) \mathcal{S}_{b}^+(z)
+ \bar{\lambda} \mathcal{Q}_{n+b}(1) \mathcal{S}_{b}^+(z) + \lambda \mathcal{Q}_{n+b-1}(1) \mathcal{S}_{b}^+(z) - \bar{\lambda} \mathcal{P}_{n,b}(1) - \lambda \mathcal{P}_{n-1,b}(1), \quad n \geq 1
\] (13)

\[
\left( \frac{z - \bar{\lambda}}{z} \right) \mathcal{Q}_{n}^+(z) = \frac{\lambda}{z} \mathcal{Q}_{n-1}^+(z) + \bar{\lambda} \lambda \mathcal{P}_{n}^+(1) + \lambda \mathcal{P}_{n-1}^+(1) \right) \mathcal{V}(z)
\]

\[
\left( \frac{z - \bar{\lambda}}{z} \right) \mathcal{Q}_{n}^+(z) = \frac{\lambda}{z} \mathcal{Q}_{n-1}^+(z) + \bar{\lambda} \mathcal{Q}_{n}(1) \mathcal{V}(z) - \bar{\lambda} \mathcal{Q}_{n}(1) - \lambda \mathcal{Q}_{n-1}(1), \quad 1 \leq n \leq a - 1
\] (15)

\[
\left( \frac{z - \bar{\lambda}}{z} \right) \mathcal{Q}_{n}^+(z) = \frac{\lambda}{z} \mathcal{Q}_{n-1}^+(z) - \bar{\lambda} \mathcal{Q}_{n}(1) - \lambda \mathcal{Q}_{n-1}(1), \quad n \geq a.
\] (16)

As our center of attention is on achieving the joint distribution of queue content and number in a served batch at departure epoch, we define the corresponding steady-state probabilities as follows:

- \( \mathcal{P}_{n,r}^+ \) be the probability that \( n \) customers in the queue at departure epoch of a batch and \( r \) customers with the departing batch,
- \( \mathcal{P}_{n}^+ \left( = \sum_{r=a}^{b} \mathcal{P}_{n,r}^+ \right) \) be the probability that \( n \) customers in the queue at departure epoch, \( n \geq 0 \),
- \( \mathcal{Q}_{n}^+ \) be the probability that \( n \) customers are present in the queue at the vacation termination epoch of the server, \( n \geq 0 \).

Now using equations (1), (2) and (10) – (16) we derive two results (presented below in the form of lemmas) which will be used in the sequel.

**Lemma 4.1.** The probabilities \( \left( \mathcal{P}_{n,r}^+, \mathcal{P}_{n,r}(1) \right) \) and \( \left( \mathcal{Q}_{n}^+, \mathcal{Q}_{n}(1) \right) \) are connected by the relation

\[
\mathcal{P}_{0,r}^+ = \tau^{-1} \{ \bar{\lambda} \mathcal{P}_{0,r}(1) \}, \quad a \leq r \leq b
\] (17)

\[
\mathcal{P}_{n,r}^+ = \tau^{-1} \{ \bar{\lambda} \mathcal{P}_{n,r}(1) + \lambda \mathcal{P}_{n-1,r}(1) \}, \quad n \geq 1, \quad a \leq r \leq b
\] (18)

\[
\mathcal{Q}_{0}^+ = \tau^{-1} \{ \bar{\lambda} \mathcal{Q}_{0}(1) \}
\] (19)

\[
\mathcal{Q}_{n}^+ = \tau^{-1} \{ \bar{\lambda} \mathcal{Q}_{n}(1) + \lambda \mathcal{Q}_{n-1}(1) \}, \quad n \geq 1
\] (20)

where \( \tau = \sum_{m=0}^{\infty} \sum_{r=a}^{b} \mathcal{P}_{m,r}(1) + \sum_{m=0}^{\infty} \mathcal{Q}_{m}(1) \).

Equations (17) and (18) provide the relation between the joint queue and server content distribution at service completion epoch with the joint distribution of queue and server content when the service is about to complete. Similarly equations (19) and (20) can be interpreted. Here \( \tau^{-1} \) gives us the
mean of service completion or vacation termination per unit time, i.e., mean departure rate from the busy state or vacation state.

**Proof:** Now we establish the relationship between \( p_{n,r}^+ \) and \( p_{n,r}(1) \), where \( p_{n,r}(1) \) denotes the joint probability that there are \( n \) customers in the queue and \( r \) with the server and remaining service time is just one slot.

Applying conditional probability argument, for \( n \geq 0, a \leq r \leq b \), we have

\[
p_{n,r}^+ = \lim_{k \to -\infty} \Pr \{ N_q(k-) = n, N_s(k-) = r, \zeta(k-) = 2 | U(k-) = 1 \}
\]

\[
= \lim_{k \to -\infty} \frac{\Pr \{ N_q(k-) = n, N_s(k-) = r, \zeta(k-) = 2, U(k-) = 1 \}}{\Pr \{ U(k-) = 1 \}}
\]

\[
= \left\{ \begin{array}{ll}
\frac{1}{\tau} \Pr \{ N_q(1) = n \}, & n = 0 \\
\frac{1}{\tau} \left[ \lambda p_{n,r}(1) + \lambda p_{n-1,r}(1) \right], & n \geq 1
\end{array} \right.
\]

where \( \tau = \sum_{m=0}^{\infty} \sum_{r=a}^{b} p_{m,r}(1) + \sum_{m=0}^{\infty} q_m(1) \).

In the similar fashion equations (19) and (20) can be established.

**Lemma 4.2.** The value of \( \tau \) is given by

\[
\tau = \sum_{m=0}^{\infty} \sum_{r=a}^{b} p_{m,r}(1) + \sum_{m=0}^{\infty} q_m(1) = \frac{1 - (1 - \delta_p) \sum_{n=0}^{a-1} p_{n,0}}{\omega}
\]

where

\[
\omega = \sum_{n=0}^{a-1} \left\{ p_{n}^+ E(V) + (1 - \delta_p) q_n^+ s_a + \delta_p q_n^+ E(V) \right\} + \sum_{n=a}^{b} \left( p_{n}^+ + q_n^+ \right) s_n + \sum_{n=b+1}^{\infty} \left( p_{n}^+ + q_n^+ \right) s_b
\]

**Proof.** For single vacation substituting \( \delta_p = 0 \) in (1) and (2), we obtain

\[
\lambda p_{n,0} = \lambda q_0(1) + \sum_{i=1}^{n} \{ \lambda q_i(1) + \lambda q_{i-1}(1) \}, \quad 1 \leq n \leq a - 1
\]

Using (23) in (10) and then summing over \( r \) from \( a \) to \( b \) and \( n \) from 0 to \( \infty \) in the equations (10) – (16), after some simplification, we get

\[
\left( \frac{z - 1}{z} \right) \left\{ \sum_{n=0}^{\infty} \sum_{r=a}^{b} p_{n,r}^+(z) + \sum_{n=0}^{\infty} q_n(z) \right\}
\]

\[
= \left[ \lambda q_0(1) + \sum_{n=1}^{a-1} \{ \lambda q_n(1) + \lambda q_{n-1}(1) \} \right] \{ (1 - \delta_p) S_a^*(z) + \delta_p V^*(z) \}
\]

\[
+ \sum_{r=a}^{b} \left[ \lambda p_{0,r} + \sum_{n=1}^{a-1} \{ \lambda p_{r,n}(1) + \lambda p_{n-1,r}(1) \} \right] V^*(z)
\]

\[
+ \sum_{n=a}^{b} \left[ \sum_{r=a}^{b} \{ \lambda p_{n,r}(1) + \lambda p_{n-1,r}(1) \} + \lambda q_n(1) + \lambda q_{n-1}(1) \right] S_n^*(z)
\]
5. Joint queue length and server content distribution at service completion epoch of a batch

The primary intention of this section is to provide the complete queue and server content distribution together. On account of this, we adopt the probability generating function approach with double variables and this methodology is well used by Pradhan (2020a); Pradhan and Gupta (2020). Firstly, our center of focus is to derive the bivariate pgf of the joint distribution of queue and server content at the service completion epoch of a batch. Secondly, we turn our attention to extract those joint distributions in a simple and elegant way.

In view of this, we first define the following pgfs as:

\[
P(z, x, y) = \sum_{n=0}^{\infty} \sum_{r=a}^{b} \left( P_{n,r}(z) x^n y^r \right), \quad |x| \leq 1, \quad |y| \leq 1
\]

\[
P^+(x, y) = \sum_{n=0}^{\infty} \sum_{r=a}^{b} \left( P_{n,r}^+(z) x^n y^r \right), \quad |x| \leq 1, \quad |y| \leq 1
\]

\[
P^+(x, 1) = \sum_{n=0}^{\infty} \sum_{r=a}^{b} \left( P_{n,r}^+(z) x^n \right) = \sum_{n=0}^{\infty} P_{n}^+(z) = P^+(x), \quad |x| \leq 1
\]

\[
Q^+(x) = \sum_{n=0}^{\infty} q^+_n x^n, \quad |x| \leq 1.
\]

Since our major concern is to find the bivariate pgf, multiplying (10)–(13) by appropriate powers of \( x \) and \( y \), summing over \( n \) from 0 to \( \infty \) and \( r \) from \( a \) to \( b \), and using (23), we get

\[
\left\{ \frac{z - (\lambda + \lambda x)}{z} \right\} P(z, x, y)
\]

\[
= (1 - \delta_p) \left\{ \lambda q_0(1) + \sum_{n=1}^{d-1} (\lambda q_n(1) + \lambda q_{n-1}(1)) \right\} S^*_d(z) y^d
\]

\[
+ \sum_{n=a}^{b} \left\{ \lambda q_n(1) + \lambda q_{n-1}(1) \right\} \left[ \sum_{r=a}^{b} \left\{ \lambda p_{n,r}(1) + \lambda p_{n-1,r}(1) \right\} \right] S^*_n(z) y^n
\]
Putting Eq. (27), we have
\[ S_n(z) = \sum_{n=0}^{\infty} S_n(x) z^n = \sum_{n=0}^{\infty} \left\{ \tilde{\lambda} q_n(1) + \lambda q_{n-1}(1) \right\} x^n y^n \]

Now substituting \( z = \tilde{\lambda} + \lambda x \) in (28) and using (17)–(20) and (25), we obtain
\[
P^+(x, y) = (1 - \delta_p) \sum_{n=0}^{a-1} q_n^+ K^{(a)}(x) y^n + \sum_{n=a}^{b} \lambda p_n + \sum_{n=a}^{b} q_n^+ K^{(n)}(x) y^n
\]
\[
+ \sum_{n=b+1}^{\infty} p_n^+ K^{(b)}(x) x^n y^n + \sum_{n=b+1}^{\infty} q_n^+ K^{(b)}(x) x^n y^n
\]
where \( K^{(r)}(x) = S_n(\tilde{\lambda} + \lambda x) \) is the pgf of \( k^{(r)}_j \), \((a \leq r \leq b, j \geq 0)\) and

\[
k^{(r)}_j = \text{Pr} \{ j \text{ arrivals during the service time of a batch of size } r \}.
\]

Now multiplying (14)–(16) with appropriate powers of \( x \) and summing over \( n \) from 0 to \( \infty \), we get
\[
\left\{ \frac{z - (\tilde{\lambda} + \lambda x)}{z} \right\} \sum_{n=0}^{\infty} q_n^+(z) x^n = \sum_{m=0}^{a-1} \left\{ p_{0,m}(1) + \sum_{n=1}^{a-1} (\lambda p_{n,m}(1) + \lambda p_{n-1,m}(1)) x^n \right\} V^+(z)
\]
\[
+ \delta_p \sum_{n=1}^{a-1} (\tilde{\lambda} q_n(1) + \lambda q_{n-1}(1)) x^n V^+(z)
\]
\[
- \tilde{\lambda} q_0(1) - \sum_{n=1}^{a-1} (\tilde{\lambda} q_n(1) + \lambda q_{n-1}(1)) x^n
\]

Putting \( z = \tilde{\lambda} + \lambda x \) in (30) and using (17)–(20) and (27), we get
\[
Q^+(x) = \sum_{n=0}^{a-1} \left( p_n^+ + \delta_p q_n^+ \right) x^n H(x)
\]
where \( H(x) = \sum_{j=0}^{\infty} h_j x^j = V^+(\tilde{\lambda} + \lambda x) \) is the pgf of \( h_j \) and

\[
h_j = \text{Pr} \{ j \text{ arrivals during the vacation time of the server} \}.
\]

Now substituting \( y = 1 \) in (29) and using (26) and (31) and after some algebraic manipulation, we obtain
\[
P^+(x) = \frac{\sum_{n=0}^{a-1} \left[ p_n^+ x^n \{ H(x) - 1 \} K^{(b)}(x) + q_n^+ (1 - \delta_p) x^n K^{(a)}(x) \right] + \sum_{n=1}^{b-1} \left[ (p_n^+ + q_n^+) x^n K^{(n)}(x) - x^n K^{(b)}(x) \right] \} \}
\]
\[
\frac{x^b - K^{(b)}(x)}{x^b - K^{(b)}(x)}
\]

The above expression represents the pgf of only queue length distribution at service completion epoch.

Now looking back into (29), using (31) and (32), after some algebraic simplification, we get
\[
\sum_{n=0}^{a-1} \left[ p_n^+ x^n \{ H(x) - 1 \} y^n K^{(b)}(x) \right. \\
\left. + q_n^+ \left\{ (1 - \delta_p) \left( (x^b - K^{(b)}(x)) y^n + y^n K^{(a)}(x) \right) x^n y^n K^{(b)}(x) \right\} \\
+ \sum_{n=a}^{b-1} \left( p_n^+ + q_n^+ \right) \left\{ (x^b - K^{(b)}(x)) (y^n K^{(n)}(x) - x^n y^n K^{(b)}(x)) \right\} \\
\left. + (K^{(a)}(x) - x^{n-b} K^{(b)}(x)) y^n K^{(b)}(x) \right\} \\
\right]
\]

\[
P^+(x, y) = \frac{\sum_{n=0}^{a-1} \left[ p_n^+ x^n \{ H(x) - 1 \} y^n K^{(b)}(x) \right. \\
\left. + q_n^+ \left\{ (1 - \delta_p) \left( (x^b - K^{(b)}(x)) y^n + y^n K^{(a)}(x) \right) x^n y^n K^{(b)}(x) \right\} \\
+ \sum_{n=a}^{b-1} \left( p_n^+ + q_n^+ \right) \left\{ (x^b - K^{(b)}(x)) (y^n K^{(n)}(x) - x^n y^n K^{(b)}(x)) \right\} \\
\left. + (K^{(a)}(x) - x^{n-b} K^{(b)}(x)) y^n K^{(b)}(x) \right\} \\
\right]}{x^b - K^{(b)}(x)}
\]

(33)

The above expression designates the bivariate pgf of queue and server content distribution at service completion epoch, which is the key ingredient of our analysis. No such result is available in the literature so far to the best of the authors’ knowledge. One can grab the joint distribution by inverting this bivariate pgf which is discussed in the subsequent analysis. It may be remarked here that direct use of transition probability matrix would lead the same expression. However, we feel that the use of the supplementary variable technique reduces the complexity involved in the derivation for the same.

**Remark 1.** The expression presented in (32) is the pgf of only queue length distribution at service completion epoch, which plays a noteworthy role to calculate the tail distribution instead of using the bivariate pgf. In order to compute the cell loss ratio in ATM switching network, the tail distribution is a powerful tool. To the best of authors’ knowledge, this result is also not available so far in the literature. As a consequence, we hope that this also leads to a new contribution in the queueing literature.

### 5.1. Steady-state conditions and determination of unknowns in the numerator of (33)

Although we have assumed the stability of the concerned queue in the model description, it can be proved using the stability condition given in Abolnikov and Dukhovny (1991). We conclude that the corresponding Markov chain is ergodic if and only if \( \frac{d}{dx} K^{(b)}(x) \big|_{x=1} < b \). Consequently, the steady-state distribution exists if \( \frac{1}{\mu_b} < b \), i.e. \( \frac{1}{\mu_b} = \rho < 1 \).

Before extracting the joint probabilities from the bivariate pgf presented in (33), we need to find the unknown quantities \( p_n^+ \) (0 \( \leq n \leq b - 1 \)) and \( q_n^+ \) (0 \( \leq n \leq a - 1 \)), appearing in the numerator. As both the expressions presented in (33) and (32) contains the same unknown quantities, we use (32) instead of (33) to find those unknowns, without loss of any generality. Although the procedure of determination of these unknowns is standard in the literature, for the sake of completeness we describe very briefly. Here we need to determine total 2\((a + b)\) unknowns. Using the result given below in the Lemma form, we express \( q_n^+ \) in terms of \( p_n^+ \) which eventually leads to total \( b \) unknowns. Employing Rouche’s theorem, it is known that, \( x^b - K^{(b)}(x) = D(x) \), (say) has exactly \( b \) zeroes in the unit disk \(|x| \leq 1\). Using these zeroes and normalizing condition, we obtain total \( b \) linear simultaneous equations and hence solving them, we get those unknowns.

**Lemma 5.1.** The relation between \( p_n^+ \) and \( q_n^+ \) are given by:

\[
q_n^+ = \sum_{i=0}^{n} \xi_i p_{n-i}^+, \quad n = 0, 1, 2, \ldots, a - 1, \text{ where}
\]

\[
\xi_0 = \frac{h_0}{1 - \delta_p h_0}
\]
\[ \xi_n = \frac{1}{1 - \delta \rho \theta_0} \left[ h_n + \delta \sum_{i=1}^{n} h_i \xi_{n-i} \right], \quad n = 1, 2, \ldots, a - 1. \]

### 5.2. Extraction of probabilities from the known bivariate pgf

The completely known bivariate pgf, the key ingredient of our analysis, is in our grip. In this section, we now proceed to extract the joint probabilities using the partial fraction technique. First, we accumulate the coefficients of \( y^j, a \leq j \leq b \) from the bivariate pgf (33) and are given below.

Coefficient of \( y^a \):

\[
\sum_{n=0}^{\infty} p_{n,a}^+ x^n = (1 - \delta \rho) K^{(a)}(x) + \sum_{n=0}^{a-1} q_n^+ \left( p_a^+ + q_a^+ \right) K^{(a)}(x) \quad (34)
\]

Coefficient of \( y^j, a + 1 \leq j \leq b - 1 \):

\[
\sum_{n=0}^{\infty} p_{n,j}^+ x^n = \left( p_j^+ + q_j^+ \right) K^{(j)}(x) \quad (35)
\]

Coefficient of \( y^b \):

\[
\sum_{n=0}^{\infty} p_{n,b}^+ x^n = \frac{1}{x^b - K^{(b)}(x)} \left[ \sum_{n=0}^{a-1} p_n^+ x^n \left( H(x) - 1 \right) K^{(b)}(x) 

+ (1 - \delta \rho) \sum_{n=0}^{a-1} q_n^+ K^{(a)}(x) K^{(b)}(x) 

+ (\delta \rho H(x) - 1) \sum_{n=0}^{a-1} q_n^+ x^n K^{(b)}(x) 

+ \sum_{n=a}^{b-1} \left( p_n^+ + q_n^+ \right) \left( K^{(n)}(x) - x^{n-b} K^{(b)}(x) \right) K^{(b)}(x) \right] 

- \sum_{n=a}^{b-1} \left( p_n^+ + q_n^+ \right) x^{n-b} K^{(b)}(x) \quad (36)
\]

Collecting the coefficients of \( x^n \) from both the side of (34) and (35), we obtain the joint distribution as:

\[
p_{n,a}^+ = \left( 1 - \delta \rho \right) \sum_{n=0}^{a-1} q_n^+ \left( p_a^+ + q_a^+ \right) k_n^{(a)} \quad (37)
\]

\[
p_{n,j}^+ = \left( p_j^+ + q_j^+ \right) k_n^{(j)}, \quad a + 1 \leq j \leq b - 1. \quad (38)
\]

Now we are left with only \( p_{n,b}^+ \) for which we need to invert the right-hand side of (36). For the known service and vacation time distributions, the right-hand side of (36) is completely known functions of \( x \). The corresponding inversion process is discussed in the subsequent analysis.
After substituting \( K^{(r)}(x) = S^r(\tilde{\Lambda} + \lambda x) \), \( a \leq r \leq b \), and \( H(x) = V^r(\tilde{\Lambda} + \lambda x) \) in the expression (36), let us denote numerator of (36) as \( \Lambda(x) \) and denominator as \( D(x) \). The degrees of \( \Lambda(x) \) and \( D(x) \) depend on the distributions of service/vacation time and service threshold value ‘\( a \)’ and maximum capacity ‘\( b \)’. Let the degree of numerator be \( L_1 \) and that of denominator be \( M_1 \), respectively. Now, we obtain \( p_{n,b}^+ \) in terms of roots of \( D(x) = 0 \) by discussing the following cases:

- **When all the zeroes of \( D(x) \) in \( |x| > 1 \) are distinct**

  Let us assume \( \alpha_1, \alpha_2, \ldots, \alpha_{M_1} \) to be the roots of \( D(x) = 0 \), out of which \( M_1 - b \) are the distinct roots in \( |x| > 1 \), as the degree of \( D(x) \) is \( M_1 \) and \( b \) insides roots are canceled out with the roots of the numerator. It is also assumed that \( D(x) = 0 \) has \( b \) simple roots inside the unit circle.

  **Case 1: \( L_1 \geq M_1 \)**

  Applying the partial-fraction expansion, the rational function \( \sum_{n=0}^{\infty} p_{n,b}^+ x^n \) can be uniquely written as

  \[
  \sum_{n=0}^{\infty} p_{n,b}^+ x^n = \sum_{i=0}^{L_1-M_1} \tau_i x^i + \sum_{k=1}^{M_1-b} \frac{c_k}{\alpha_k - x},
  \]

  for some constants \( \tau_i \) and \( c_k \)'s. The first sum is the result of the division of the polynomial \( \Lambda(x) \) by \( D(x) \) and the constants \( \tau_i \) are the coefficients of the resulting quotient. Using the residue theorem, we have \( c_k = -\frac{\Lambda(\alpha_k)}{D'(\alpha_k)} \), \( k = 1, 2, \ldots, M_1 - b \).

  Now, collecting the coefficient of \( x^n \) from both the sides of (39), we have

  \[
  p_{n,b}^+ = \begin{cases} 
  \tau_n + \sum_{k=1}^{M_1-b} \frac{c_k}{\alpha_k - x}, & 0 \leq n \leq L_1 - M_1, \\
  \sum_{k=1}^{M_1-b} \frac{c_k}{\alpha_k - x}, & n > L_1 - M_1.
  \end{cases}
  \]

  **Case 2: \( L_1 < M_1 \)**

  Using partial-fraction technique on \( \sum_{n=0}^{\infty} p_{n,b}^+ x^n \), we have

  \[
  \sum_{n=0}^{\infty} p_{n,b}^+ x^n = \sum_{k=1}^{M_1-b} \frac{c_k}{\alpha_k - x},
  \]

  where \( c_k = -\frac{\Lambda(\alpha_k)}{D'(\alpha_k)} \), \( k = 1, 2, \ldots, M_1 - b \).

  Now, collecting the coefficient of \( x^n \) from both the sides of (41), we obtain

  \[
  p_{n,b}^+ = \sum_{k=1}^{M_1-b} \frac{c_k}{\alpha_k^{n+1}}, \quad n \geq 0.
  \]

- **When some zeroes of \( D(x) \) in \( |x| > 1 \) are repeated**

  The denominator \( D(x) \) may have some multiple zeroes with an absolute value greater than one. Let \( D(x) \) has total \( m \) multiple zeroes, say \( \beta_1, \beta_2, \ldots, \beta_m \) with multiplicity \( \pi_1, \pi_2, \ldots, \pi_m \), respectively. Further, it is clear that \( D(x) \) has total \( (M_1 - b - \chi) \) distinct zeroes, where \( \chi = \sum_{i=1}^{m} \pi_i \), say, \( \alpha_1, \alpha_2, \ldots, \alpha_{M_1-b-\chi} \). It is also assumed that \( D(x) \) has \( b \) simple zeroes inside the unit circle.

  **Case 1: \( L_1 \geq M_1 \)**

  Applying the partial-fraction method, we can uniquely write \( \sum_{n=0}^{\infty} p_{n,b}^+ x^n \), as

  \[
  \sum_{n=0}^{\infty} p_{n,b}^+ x^n = \sum_{i=0}^{L_1-M_1} \tau_i x^i + \sum_{k=1}^{M_1-b-\chi} \frac{c_k}{\alpha_k - x} + \sum_{y=1}^{m} \frac{\xi_{\nu,i}}{\beta_\nu - x^{\nu-i+1}}.
  \]

  Using residue theorem, we obtain
\[ c_k = \frac{\Lambda(\alpha_k)}{D(\alpha_k)}, \quad k = 1, 2, \ldots, M_1 - b - m, \]

\[ \zeta_{v,i} = \frac{1}{(\pi_v - i)!} \lim_{x \to \beta} \frac{d^{(\pi_v - i)}}{dx^{(\pi_v - i)}} \left[ \frac{(\beta_v - x)^{\pi_v} \Lambda(x)}{D(x)} \right], \quad v = 1, 2, \ldots, m, i = 1, 2, \ldots, \pi_v. \]

Now collecting the coefficient of \( x^n \) from both the sides of (43), we have the distribution \( p^n_{n,b}, n \geq 0 \) as:

\[ p^n_{n,b} = \begin{cases} 
\tau_n + \sum_{k=1}^{M_1 - b - \chi} \sum_{v=0}^{\pi_v} \sum_{i=1}^{\pi_v} \left( \frac{\pi_v + n - i}{\pi_v - i} \right) \frac{\zeta_{\pi_v,i}}{p^n_{\pi_v,n+i}}, & 0 \leq n \leq L_1 - M_1, \\
\sum_{k=1}^{M_1 - b - \chi} \sum_{v=0}^{\pi_v} \sum_{i=1}^{\pi_v} \left( \frac{\pi_v + n - i}{\pi_v - i} \right) \frac{\zeta_{\pi_v,i}}{p^n_{\pi_v,n+i}}, & n > L_1 - M_1.
\end{cases} \tag{44} \]

**Case 2: \( L_1 < M_1 \)**

Here, in partial-fraction, we omit only the first summation term of the right-hand side of (43). Then collecting the coefficients of \( x^n \), we obtain \( p^n_{n,b} \), which are given by

\[ p^n_{n,b} = \sum_{k=1}^{M_1 - b - \chi} \frac{c_k}{\alpha_k^n} + \sum_{v=0}^{\pi_v} \sum_{i=1}^{\pi_v} \left( \frac{\pi_v + n - i}{\pi_v - i} \right) \frac{\zeta_{\pi_v,i}}{p^n_{\pi_v,n+i}}, \quad n \geq 0. \tag{45} \]

This completes the extraction of the joint probabilities \( p^n_{n,b} \).

**Remark 2.** One may feel the necessity of having the probabilities at vacation termination epoch i.e., \( q^n_{n}, n \geq 0 \), which can be easily extracted from \( Q^+(x) \) and are given below.

\[ q^n_{n} = \begin{cases} 
\sum_{i=0}^{n} \left( p^n_i + \delta p^n_i \right) h_{n-i}, & 0 \leq n \leq a - 1 \\
\sum_{i=0}^{a-1} \left( p^n_i + \delta p^n_i \right) h_{n-i}, & n \geq a.
\end{cases} \tag{46} \]

**Remark 3.** It should also be noted here that the queue length distribution at service completion epoch i.e., \( p^n_{n} \) can be achieved by summing over \( r \) from \( a \) to \( b \) the joint probabilities \( p^n_{n,r} \), without the requirement of further separate extraction from \( P^+(x) \).

This completes the determination of the joint distribution of queue length and server content at service completion epoch as well as the only queue length distribution at vacation termination epoch. Now in the next section, the probability distribution at arbitrary slot has been acquired.

**6. Joint distribution of queue and server content at arbitrary slot**

The steady-state queue and server content distribution at arbitrary slot plays an important role in computing several notable key performance measures. A relationship between the probabilities at service completion/post transmission and arbitrary epochs has been generated and presented below.

**Theorem 6.1.** The probabilities at service completion and arbitrary epochs i.e. \( (p_{n,r}, p^n_{n,r}, p^n_{n}) \) and \( (q_n, q^n_n) \) are related by

\[ p_{n,0} = \frac{1}{E^*} \sum_{m=0}^{n} q^+_m, \quad 0 \leq n \leq a - 1 \tag{47} \]
\[ p_{0,a} = \frac{1}{E^*}(p_a^+ + q_a^+ - p_{0,a}^+ + (1 - \delta_p)\sum_{m=0}^{a-1} q_m^+) \quad (48) \]

\[ p_{0,r} = \frac{1}{E^*}(p_r^+ + q_r^+ - p_{0,r}^+), \quad a + 1 \leq r \leq b \quad (49) \]

\[ p_{n,r} = p_{n-1,r} - \frac{p_{n,r}}{E^*}, \quad n \geq 1, \quad a \leq r \leq b - 1 \quad (50) \]

\[ p_{n,b} = p_{n-1,b} + \frac{1}{E^*}(p_{n,b}^+ + q_{n+b}^+ - p_{n,b}^+), \quad n \geq 1 \quad (51) \]

\[ q_0 = \frac{1}{E^*}(p_0^+ - (1 - \delta_p)q_0^+) \quad (52) \]

\[ q_n = q_{n-1} + \frac{1}{E^*}(p_n^+ - (1 - \delta_p)q_n^+), \quad 1 \leq n \leq a - 1 \quad (53) \]

\[ q_n = q_{n-1} - \frac{q_n^+}{E^*}, \quad n \geq a \quad (54) \]

\[
E^* = \lambda \omega + (1 - \delta_p)\sum_{i=0}^{a-1} \sum_{m=0}^{i} q_m^+,
\]

\[
\omega = \sum_{n=0}^{a-1} \{p_n^+ E(V) + (1 - \delta_p)q_n^+ s_a + \delta_p q_n^+ E(V)\} + \sum_{n=a}^{b} (p_n^+ + q_n^+)s_n
\]

\[ + \sum_{n=b+1}^{\infty} (p_n^+ + q_n^+)s_b \]

**Proof:** Use of (23) in (1)–(2), and then division by \( \tau \) and use of (17)–(20) leads

\[ p_{0,0} = \frac{1 - (1 - \delta_p)\sum_{i=0}^{a-1} p_{i,0}}{\lambda \omega} q_0^+ \quad (55) \]

\[ p_{n,0} = \frac{1 - (1 - \delta_p)\sum_{i=0}^{a-1} p_{i,0}}{\lambda \omega} \sum_{m=0}^{n} q_m^+, \quad 1 \leq n \leq a - 1. \quad (56) \]

Now dividing (56) by (55), we get

\[ \frac{p_{n,0}}{q_0^+} = \frac{p_{0,0}}{\sum_{m=0}^{n} q_m^+}, \quad 0 \leq n \leq a - 1. \quad (57) \]

Use of (57) in (55) and (56) yields

\[ p_{n,0} = \frac{\sum_{m=0}^{n} q_m^+}{\lambda \omega + (1 - \delta_p)\sum_{i=0}^{a-1} \sum_{m=0}^{i} q_m^+}, \quad 0 \leq n \leq a - 1. \quad (58) \]
which is the desired result of (47).

Putting \(z = 1\) in (10) and (11), use of (23), and division by \(\tau\) gives

\[
p_{0,a} = \frac{1 - (1 - \delta_p)}{\lambda \omega} \sum_{i=0}^{a-1} p_{i,0} \left( p_a^+ + q_a^+ + (1 - \delta_p) \sum_{m=0}^{a-1} q_m^+ - p_{0,a}^+ \right)
\]

(59)

\[
p_{0,r} = \frac{1 - (1 - \delta_p)}{\lambda \omega} \sum_{i=0}^{a-1} p_{i,0} \left( p_r^+ + q_r^+ - p_{0,r}^+ \right), \quad a + 1 \leq r \leq b
\]

(60)

As

\[
1 - (1 - \delta_p) \sum_{i=0}^{a-1} p_{i,0} = \frac{\lambda \omega}{\lambda \omega + (1 - \delta_p) \sum_{i=0}^{a-1} q_m^+},
\]

(61)

Using (61) in (59) and (60), we get the required results of (48) and (49).

Substituting \(z = 1\) in (12)–(16) and then dividing by \(\tau\), using the similar approach, we get our desired results of (50)–(54).

7. Special cases

From the concerned Geo/G\(_n^{(a,b)}/1\) queue with single and multiple vacations, one can deduce a large class of well-known queueing models as special cases which are listed below.

(i) Geo/G\(_n^{(a,b)}/1\) queue without a vacation.

If we assume that the random vacation time is zero, i.e. \(H(x) = V^*(\bar{\lambda} + \lambda x) = 1\) then the queue and server content distribution, the only queue length distribution for Geo/G\(_n^{(a,b)}/1\) queue without vacation can be obtained as special cases from the model discussed in this paper. This batch-service queue without any vacation policy has been analyzed by Claeyts et al. (2010b). However, they did not provide the complete extraction procedure of the joint probabilities. From our analysis, one can easily extract the joint probabilities from the bivariate pgf using the partial fraction method.

(ii) Geo/G\(_n^{(a,b)}/1\) queue with vacation.

The customers are served according to \((a, b)\) rule where the service times for all the batches are the same. Assuming \(K^{(i)}(x) = K(x), a \leq i \leq b\), we can achieve this queue and corresponding queue and server content distribution together can be obtained.

(iii) Geo/G\(_n^{(a,b)}/1\) queue without a vacation.

In this case, the service times for all the batches are the same and the random vacation time is zero. This queue can be achieved by assuming \(K^{(i)}(x) = K(x), a \leq i \leq b\) and \(h_0 = 1, h_1 = 0, j \geq 1, i.e., H(x) = V^*(\bar{\lambda} + \lambda x) = 1\).

(iv) Geo/G\(_n^b/1\).

If we assume \(a = b\) it reduces to fixed batch-size rule and eventually, the service times for all the batches are the same, i.e. the substitution of \(a = b\) and \(K^{(i)}(x) = K(x), a \leq i \leq b\) in our concerned model leads to Geo/G\(_n^b/1\) queue with single and multiple vacation policies. Moreover, the assumption of \(H(x) = V^*(\bar{\lambda} + \lambda x) = 1\) (the vacation time is zero) leads to Geo/G\(_n^b/1\) queueing system without a vacation.

(v) Geo/G/1.

Here the customers arrive singly and are also served individually. This can be achieved by assuming \(a = b = 1, K^{(i)}(x) = K(x), a \leq i \leq b\). Moreover, if we assume vacation time is zero then it reduces to the most basic discrete queueing model Geo/G/1 without any type of vacation.
Table 1. PH₀ representation for service time of batch size r.

| r  | βᵣ | Tᵣ | E(Tᵣ) = sᵣ |
|----|-----|-----|-------------|
| 5  | (0.4, 0.2, 0.4) | (0.5, 0.3, 0.1) | 6.988764 |
| 6  | (0.25, 0.5, 0.5) | (0.2, 0.5, 0.1) | 7.131147 |
| 7  | (0.5, 0.3, 0.2) | (0.1, 0.8, 0.1) | 8.000000 |
| 8  | (0.3, 0.3, 0.4) | (0.6, 0.2, 0.1) | 13.863636 |
| 9  | (0.3, 0.4, 0.3) | (0.2, 0.1, 0.2) | 15.750000 |
| 10 | (0.6, 0.2, 0.2) | (0.4, 0.4, 0.1) | 18.000000 |

8. Marginal distributions and performance measures

From the application perspective of the concerned model, the marginal distributions along with the performance measures are the key ingredients for the system designers/vendors. We first present the following utmost marginal distributions in terms of known distributions:

- the distribution of the number of customers in the system (which includes total number of customers in the queue and with the server), \( p_{n}^{\text{sys}}, n \geq 0, \)

\[
p_{n}^{\text{sys}} = \begin{cases} (1 - \delta_p) p_{n,0} + q_n, & 0 \leq n \leq a - 1 \\ \sum_{r=a}^{\min(b,n)} p_{n-r,r} + q_n, & a \leq n \leq b, \\ \sum_{r=a}^{b} p_{n-r,r} & n \geq b + 1. \end{cases}
\]

- the distribution of the number of customers in the queue \( p_{n}^{\text{queue}}, n \geq 0, \)

\[
p_{n}^{\text{queue}} = \begin{cases} (1 - \delta_p) p_{n,0} + \sum_{r=a}^{b} p_{n,r} + q_n, & 0 \leq n \leq a - 1 \\ \sum_{r=a}^{b} p_{n,r} + q_n & n \geq a. \end{cases}
\]

- the probability that the server is in a busy state \( (P_{\text{busy}}), \) in the vacation state \( (Q_{\text{vac}}), \) and in the dormant state \( (P_{\text{dor}}) \) are given by \( P_{\text{busy}} = \sum_{n=0}^{\infty} \sum_{r=a}^{b} p_{n,r}, \ Q_{\text{vac}} = \sum_{n=0}^{\infty} q_n, \ P_{\text{dor}} = \sum_{n=0}^{a-1} p_{n,0}. \)

- the conditional distribution of the number of customers undergoing service with the server given that the server is busy, \( p_{r}^{\text{ser}}, (a \leq r \leq b) \)

\[
p_{r}^{\text{ser}} = \sum_{n=0}^{\infty} p_{n,r}/P_{\text{busy}}(a \leq r \leq b).
\]

Performance measures are very crucial from the application point of view as it improves the efficiency of the system. Although some performance measures can be derived from the pgf, here we obtain several pivotal performance measures using completely known state probabilities and marginal distributions in a very simplified manner and are given by:

- average number of customers waiting in the queue at any arbitrary time \( (L_q) = \sum_{n=0}^{\infty} np_{n}^{\text{queue}}, \)

- average number in the system \( (L) = \sum_{n=0}^{\infty} np_{n}^{\text{sys}}, \)

- average number of customers with the server \( (L_s) = \sum_{r=a}^{b} r p_{r}^{\text{ser}}, \)

- average waiting time of a customer in the queue \( (W_q) = \frac{L_q}{\lambda} \) as well as in the system \( (W) = \frac{L}{\lambda}. \)
Table 2. Probability distribution at service completion epoch for a single vacation.

| \(n\) | \(p_{15}^{+}\) | \(p_{15}^{-}\) | \(p_{12}^{+}\) | \(p_{12}^{-}\) | \(p_{8}^{+}\) | \(p_{8}^{-}\) | \(p_{10}^{+}\) | \(p_{n}^{+}\) | \(q_{n}^{+}\) |
|------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| 0    | 0.087474 | 0.007327 | 0.004266 | 0.001853 | 0.001040 | 0.000633 | 0.102593 | 0.036708 |
| 1    | 0.112271 | 0.009846 | 0.005402 | 0.002351 | 0.001390 | 0.001320 | 0.132580 | 0.083705 |
| 2    | 0.079254 | 0.007201 | 0.003892 | 0.001824 | 0.001136 | 0.001689 | 0.094996 | 0.097162 |
| 3    | 0.053329 | 0.004864 | 0.002771 | 0.001487 | 0.000948 | 0.001870 | 0.065269 | 0.085335 |
| 4    | 0.035646 | 0.003254 | 0.001964 | 0.001219 | 0.000795 | 0.001935 | 0.044813 | 0.066972 |
| 5    | 0.023802 | 0.002171 | 0.001390 | 0.000667 | 0.001927 | 0.003892 | 0.030959 | 0.038741 |
| 10   | 0.003157 | 0.000284 | 0.000245 | 0.000377 | 0.000278 | 0.001935 | 0.005828 | 0.007111 |
| 25   | 0.000007 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 50   | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 75   | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| >= 90| 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |

9. Numerical illustration

The principal objective of this section is to manifest the feasibility and applicability of the proposed method and results by the inclusion of numerical examples. Although we have generated extensive computations works, only a few of them are appended here due to lack of space. All the calculations were performed using Maple 15 on PC having configuration Intel (R) Core (TM) i5-3470 CPU Processor @ 3.20 GHz with 4.00 GB of RAM, and the results are presented in the tabular form in which some relevant performance measures are also included.

Example 1: Transmission errors are the inescapable part of the communication channel as it may occur due to electrical faults, bad weather, fading channel etc. In order to monitor this error, discrete phase type distribution (PHD) significantly fits as a powerful tool. It covers a wide range of almost all relevant discrete distributions such as geometric distribution, negative binomial distribution etc. On account of this, in this example we consider service time to follow PHD distribution with the representation \((\beta_i, T_i)\) for \(a \leq i \leq b\), where \(\beta_i\) is a row vector of order \(v\) and \(T_i\) is a square matrix of order \(v\). The associated pgf is \(z^\beta_i(I - zT_i)^{-1}\eta_i\) with \(\eta_i + T_i e = e\), where \(I\) is the identity matrix of appropriate order and \(e\) is the column vector. Consequently, \(K^{(i)}(x) = (1 - \lambda + \lambda x)\beta_i(I - (1 - \lambda + \lambda x)T_i)^{-1}\eta_i\). Here we consider \(a = 5, b = 10, \lambda = 0.5, \rho = 0.693181\), and PHD distribution for different batch-size is provided in the Table 1. The vacation time follows geometric distribution with pmf \((v_j) = p(1-p)^{j-1}, j \geq 1\) with \(p = 0.3\), pgf \(z^p = \frac{pz}{1 - (1 - p)z}\), mean \((\overline{v}) = 3.333333\) and hence \(H(x) = \frac{0.15 + 0.15x}{0.65 + 0.35x}\). The joint distribution at service completion and arbitrary epochs along with some performance measures are displayed in [Tables 2–5].

Example 2: In this example, the service time has been considered as negative binomial (NB) distribution with pmf

\[
s_r(n) = \binom{n - 1}{r - 1} \mu_i^{r - 1} \mu_i, \quad n = r, r + 1, r + 2, \ldots \text{ which consequently leads}
\]

\[
K^{(i)}(x) = \binom{\mu_i(1 - \lambda + \lambda x)}{1 - (1 - \mu_i)(1 - \lambda + \lambda x)}^r.
\]
Table 3. Probability distribution at arbitrary slot for a single vacation.

| n  | \(p_{n,0}\)    | \(p_{n,5}\)    | \(p_{n,6}\)    | \(p_{n,7}\)    | \(p_{n,8}\)    | \(p_{n,9}\)    | \(p_{n,10}\)   | \(q_n\)  | \(p_{n,queue}\) |
|----|----------------|----------------|----------------|----------------|----------------|----------------|--------------|--------|-----------------|
| 0  | 0.011705       | 0.112279       | 0.01098        | 0.005986       | 0.003996       | 0.002687       | 0.001883     | 0.021009 | 0.169643        |
| 1  | 0.038396       | 0.076479       | 0.006958       | 0.004263       | 0.003246       | 0.002244       | 0.002966     | 0.036595 | 0.171147        |
| 2  | 0.069379       | 0.051207       | 0.004662       | 0.003022       | 0.002664       | 0.001881       | 0.003553     | 0.035904 | 0.172272        |
| 3  | 0.096590       | 0.034202       | 0.003111       | 0.002138       | 0.002190       | 0.001579       | 0.003827     | 0.029507 | 0.173144        |
| 4  | 0.117946       | 0.022835       | 0.002073       | 0.001512       | 0.001801       | 0.001325       | 0.003903     | 0.022442 | 0.173837        |
| 5  | 0.015245       | 0.001380       | 0.001069       | 0.001481       | 0.001112       | 0.003852       | 0.010088     | 0.034227 | 0.006519        |
| 10 | 0.002021       | 0.000180       | 0.000188       | 0.000558       | 0.000463       | 0.002924       | 0.00185      | 0.006519 | 0.000868        |
| 25 | 0.000070       | 0.000000       | 0.000000       | 0.000000       | 0.000000       | 0.000000       | 0.000000     | 0.000000 | 0.000000        |
| 50 | 0.000000       | 0.000000       | 0.000000       | 0.000000       | 0.000000       | 0.000000       | 0.000000     | 0.000000 | 0.000000        |
| 75 | 0.000000       | 0.000000       | 0.000000       | 0.000000       | 0.000000       | 0.000000       | 0.000000     | 0.000000 | 0.000000        |
| ≥ 90 | 0.000000   | 0.000000       | 0.000000       | 0.000000       | 0.000000       | 0.000000       | 0.000000     | 0.000000 | 0.000000        |

\(L=6.262998, L_q=3.202430, L_t=6.094380,\)
\(P_{busy}=0.502195, W=17.894281, W_q=9.149802\)
Table 4. Probability distribution at service completion epoch for multiple vacations.

| n  | \(p_{n,5}^\ast\) | \(p_{n,6}^\ast\) | \(p_{n,7}^\ast\) | \(p_{n,8}^\ast\) | \(p_{n,9}^\ast\) | \(p_{n,10}^\ast\) | \(q_{n}^\ast\) | \(p_{n}^\text{queue}\) |
|----|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|
| 0  | 0.025228         | 0.011491         | 0.005669         | 0.002080         | 0.001003         | 0.000541         | 0.046014         | 0.025636         |
| 1  | 0.032380         | 0.015443         | 0.007179         | 0.002640         | 0.001340         | 0.001097         | 0.060079         | 0.072914         |
| 2  | 0.022857         | 0.011294         | 0.005172         | 0.002048         | 0.001095         | 0.001367         | 0.043833         | 0.115362         |
| 3  | 0.015380         | 0.007629         | 0.003683         | 0.001670         | 0.000914         | 0.001487         | 0.030763         | 0.145653         |
| 4  | 0.010280         | 0.005104         | 0.002610         | 0.001369         | 0.000766         | 0.001522         | 0.021651         | 0.166949         |
| 5  | 0.006864         | 0.003405         | 0.001847         | 0.001125         | 0.000643         | 0.001507         | 0.015391         | 0.111387         |
| 10 | 0.000910         | 0.000446         | 0.000326         | 0.000423         | 0.000268         | 0.001167         | 0.003540         | 0.002044         |
| 25 | 0.000002         | 0.000000         | 0.000000         | 0.000000         | 0.000000         | 0.000000         | 0.000000         | 0.000000         |
| 50 | 0.000000         | 0.000000         | 0.000000         | 0.000000         | 0.000000         | 0.000031         | 0.000000         | 0.000000         |
| 75 | 0.000000         | 0.000000         | 0.000000         | 0.000000         | 0.000000         | 0.000000         | 0.000000         | 0.000000         |
| \(\geq 90\) | 0.000000         | 0.000000         | 0.000000         | 0.000000         | 0.000000         | 0.000000         | 0.000000         | 0.000000         |

In particular, here \(r = 3, \lambda = 0.15, a = 12, b = 20\) and service rates are taken as follows: \(\mu_{12} = 0.346153, \mu_{13} = 0.321428, \mu_{14} = 0.300000, \mu_{15} = 0.281250, \mu_{16} = 0.264705, \mu_{17} = 0.250000, \mu_{18} = 0.236842, \mu_{19} = 0.225000, \mu_{20} = 0.214285\). On the other hand, the vacation time follows PHD distribution with \(\beta = (0.2, 0.2, 0.6)\), \(T = \begin{pmatrix} 0.5 & 0.3 & 0.2 \\ 0.4 & 0.3 & 0.1 \end{pmatrix}\), mean = 17.736842 so that the traffic intensity of the system \(\rho = 0.315000\). The joint distribution at service completion and arbitrary epochs along with some performance measures are displayed in [Tables 6–9].

**Example 3:** In the modern wireless telecommunication networks, we often deal with the transmission of voice, video, data and various other types of informations (usually refer as messages which are broken into packets). These packets arrive according to the Bernoulli process with parameter \(\lambda = 0.7\), (say). It is observed that an individual node operates/transmits the packets during a fixed slot as a single entity through a common interface. A point-to multi point communication is used most frequently in wireless Internet and IP telephony via gigahertz radio frequencies. The key ingredients of 802.16 systems are a Base Station (BS) and a Subscriber Station (SS). According to the bandwidth demand BS allocates variable number of physical slots to each SS. The application must initiate and establish a connection between a BS and SS before the initiation of the data transmission. In such systems, the data transfer takes place in uplink (SS to BS) and downlink (BS to SS) directions with the help of Time Division Multiple Access (TDMA). The time is divided into the same amount of slots which is the principal mechanism behind a discrete-time queueing model. The BS can dynamically allocate different time slots for downlink and uplink. The BS transmits packets in batches with a minimum threshold \(a = 8\) and maximum limit \(b = 15\). The transmission/
Table 6. Probability distribution at service completion epoch for a single vacation.

| $n$ | $p^+_{n,12}$ | $p^+_{n,13}$ | $p^+_{n,14}$ | $p^+_{n,15}$ | $p^+_{n,16}$ | $p^+_{n,17}$ | $p^+_{n,18}$ | $p^+_{n,19}$ | $p^+_{n,20}$ | $q^+_n$ |
|-----|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|----------|
| 0   | 0.142104     | 0.000858     | 0.000564     | 0.000372     | 0.000245     | 0.000162     | 0.000107     | 0.000071     | 0.000047     | 0.144530  |
| 1   | 0.169353     | 0.001073     | 0.000737     | 0.000506     | 0.000346     | 0.000237     | 0.000162     | 0.000110     | 0.000109     | 0.172633  |
| 2   | 0.104665     | 0.000706     | 0.000513     | 0.000369     | 0.000265     | 0.000189     | 0.000134     | 0.000094     | 0.000144     | 0.107079  |
| 3   | 0.046869     | 0.000340     | 0.000263     | 0.000200     | 0.000151     | 0.000113     | 0.000083     | 0.000061     | 0.000147     | 0.048227  |
| 4   | 0.017560     | 0.000137     | 0.000114     | 0.000092     | 0.000074     | 0.000057     | 0.000044     | 0.000034     | 0.000130     | 0.018242  |
| 5   | 0.005904     | 0.000050     | 0.000044     | 0.000038     | 0.000032     | 0.000026     | 0.000021     | 0.000017     | 0.000106     | 0.06238   |
| 8   | 0.000157     | 0.000002     | 0.000002     | 0.000002     | 0.000002     | 0.000002     | 0.000002     | 0.000002     | 0.000045     | 0.000216  |
| 12  | 0.000000     | 0.000000     | 0.000000     | 0.000000     | 0.000000     | 0.000000     | 0.000000     | 0.000000     | 0.000011     | 0.000111  |
| 16  | 0.000000     | 0.000000     | 0.000000     | 0.000000     | 0.000000     | 0.000000     | 0.000000     | 0.000000     | 0.000003     | 0.00003   |
| 30  | 0.000000     | 0.000000     | 0.000000     | 0.000000     | 0.000000     | 0.000000     | 0.000000     | 0.000000     | 0.000000     | 0.000009  |
| $\geq 40$ | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
Table 7. Probability distribution at arbitrary slot for a single vacation.

| n   | \( p_{n,0} \) | \( p_{n,12} \) | \( p_{n,13} \) | \( p_{n,14} \) | \( p_{n,15} \) | \( p_{n,16} \) | \( p_{n,17} \) | \( p_{n,18} \) | \( p_{n,19} \) | \( p_{n,20} \) | \( q_n \) | \( p_n^{\text{queue}} \) |
|-----|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|-------------|----------------|
| 0   | 0.005511      | 0.057487      | 0.000386      | 0.000281      | 0.000203      | 0.000147      | 0.000106      | 0.000076      | 0.000055      | 0.000039      | 0.018435    | 0.082726      |
| 1   | 0.017355      | 0.029427      | 0.000208      | 0.000158      | 0.000120      | 0.000090      | 0.000067      | 0.000049      | 0.000036      | 0.000055      | 0.035194    | 0.082759      |
| 2   | 0.031550      | 0.012086      | 0.000091      | 0.000073      | 0.000058      | 0.000046      | 0.000035      | 0.000027      | 0.000021      | 0.000055      | 0.038741    | 0.082783      |
| 3   | 0.044580      | 0.004320      | 0.000035      | 0.000030      | 0.000025      | 0.000021      | 0.000017      | 0.000013      | 0.000011      | 0.000048      | 0.033702    | 0.082802      |
| 4   | 0.055047      | 0.001410      | 0.000012      | 0.000011      | 0.000010      | 0.000009      | 0.000008      | 0.000006      | 0.000005      | 0.000038      | 0.026258    | 0.082814      |
| 5   | 0.062914      | 0.000432      | 0.000004      | 0.000004      | 0.000003      | 0.000003      | 0.000003      | 0.000002      | 0.000002      | 0.000029      | 0.019425    | 0.082826      |
| 6   | 0.068641      | 0.00126       | 0.000001      | 0.000001      | 0.000001      | 0.000001      | 0.000001      | 0.000001      | 0.000001      | 0.000001      | 0.014302    | 0.082807      |
| 7   | 0.072750      | 0.000035      | 0.000000      | 0.000000      | 0.000000      | 0.000000      | 0.000000      | 0.000000      | 0.000000      | 0.000000      | 0.010030    | 0.082830      |
| 8   | 0.075679      | 0.000009      | 0.000000      | 0.000000      | 0.000000      | 0.000000      | 0.000000      | 0.000000      | 0.000000      | 0.000011      | 0.007138    | 0.082837      |
| 9   | 0.077760      | 0.000003      | 0.000000      | 0.000000      | 0.000000      | 0.000000      | 0.000000      | 0.000000      | 0.000000      | 0.000000      | 0.005070    | 0.082841      |
| 10  | 0.079238      | 0.000000      | 0.000000      | 0.000000      | 0.000000      | 0.000000      | 0.000000      | 0.000000      | 0.000000      | 0.000000      | 0.003598    | 0.082841      |
| 11  | 0.080286      | 0.000000      | 0.000000      | 0.000000      | 0.000000      | 0.000000      | 0.000000      | 0.000000      | 0.000000      | 0.000000      | 0.002553    | 0.082843      |
| 12  | 0.080286      | 0.000000      | 0.000000      | 0.000000      | 0.000000      | 0.000000      | 0.000000      | 0.000000      | 0.000000      | 0.000000      | 0.001809    | 0.082812      |
| 20  | 0.080286      | 0.000000      | 0.000000      | 0.000000      | 0.000000      | 0.000000      | 0.000000      | 0.000000      | 0.000000      | 0.000000      | 0.001158    | 0.082815      |
| 30  | 0.080286      | 0.000000      | 0.000000      | 0.000000      | 0.000000      | 0.000000      | 0.000000      | 0.000000      | 0.000000      | 0.000000      | 0.000004    | 0.000000      |
| \( \geq 40 \) | 0.080286 | 0.000000      | 0.000000      | 0.000000      | 0.000000      | 0.000000      | 0.000000      | 0.000000      | 0.000000      | 0.000000      | 0.000000    | 0.000000      |

\( L=6.866692, L_q=5.556901, L_r=12.095813, \)

\( P_{\text{busy}}=0.108284, W=45.777950, W_q=37.046006 \)
Table 8. Probability distribution at service completion epoch for multiple vacations.

| n  | $p_{n,12}$ | $p_{n,13}$ | $p_{n,14}$ | $p_{n,15}$ | $p_{n,16}$ | $p_{n,17}$ | $p_{n,18}$ | $p_{n,19}$ | $p_{n,20}$ | $p_{n}$ | $q_{n}$ |
|----|------------|------------|------------|------------|------------|------------|------------|------------|------------|--------|-------|
| 0  | 0.014508   | 0.009600   | 0.006318   | 0.004163   | 0.002747   | 0.001816   | 0.001202   | 0.000797   | 0.000529   | 0.041680 | 0.012462 |
| 1  | 0.017291   | 0.012009   | 0.008260   | 0.005665   | 0.003878   | 0.002652   | 0.001812   | 0.001237   | 0.001219   | 0.054023 | 0.031604 |
| 2  | 0.010686   | 0.007896   | 0.005740   | 0.004139   | 0.002966   | 0.002114   | 0.001500   | 0.001061   | 0.001061   | 0.037714 | 0.046955 |
| 3  | 0.004785   | 0.003799   | 0.002944   | 0.002247   | 0.001695   | 0.001266   | 0.000938   | 0.000689   | 0.000689   | 0.016464 | 0.055918 |
| 4  | 0.001792   | 0.001538   | 0.001276   | 0.001035   | 0.000825   | 0.000647   | 0.000501   | 0.000384   | 0.000384   | 0.001458 | 0.009456 |
| 5  | 0.000602   | 0.000560   | 0.000499   | 0.000431   | 0.000363   | 0.000300   | 0.000243   | 0.000194   | 0.000194   | 0.001187 | 0.004379 |
| 10 | 0.000001   | 0.000002   | 0.000003   | 0.000003   | 0.000003   | 0.000003   | 0.000003   | 0.000003   | 0.000003   | 0.000260 | 0.002810 |
| 15 | 0.000000   | 0.000000   | 0.000000   | 0.000000   | 0.000000   | 0.000000   | 0.000000   | 0.000000   | 0.000000   | 0.000047 | 0.017898 |
| 20 | 0.000000   | 0.000000   | 0.000000   | 0.000000   | 0.000000   | 0.000000   | 0.000000   | 0.000000   | 0.000000   | 0.000008 | 0.003202 |
| 30 | 0.000000   | 0.000000   | 0.000000   | 0.000000   | 0.000000   | 0.000000   | 0.000000   | 0.000000   | 0.000000   | 0.000000 | 0.000102 |
| ≥45| 0.000000   | 0.000000   | 0.000000   | 0.000000   | 0.000000   | 0.000000   | 0.000000   | 0.000000   | 0.000000   | 0.000000 | 0.000000 |
Table 9. Probability distribution at arbitrary slot for multiple vacations.

| n  | $p_{n.12}$ | $p_{n.13}$ | $p_{n.14}$ | $p_{n.15}$ | $p_{n.16}$ | $p_{n.17}$ | $p_{n.18}$ | $p_{n.19}$ | $p_{n.20}$ | $q_n$ | $p_n^{\text{queue}}$ |
|-----|------------|------------|------------|------------|------------|------------|------------|------------|------------|-----|------------------|
| 0   | 0.014370   | 0.010580   | 0.007706   | 0.005590   | 0.004045   | 0.002920   | 0.002104   | 0.001514   | 0.001087   | 0.016909 | 0.066825        |
| 1   | 0.007356   | 0.005708   | 0.004355   | 0.003292   | 0.002471   | 0.001844   | 0.001369   | 0.001012   | 0.001516   | 0.038825 | 0.067748        |
| 2   | 0.003021   | 0.002505   | 0.002026   | 0.001613   | 0.001268   | 0.000986   | 0.000760   | 0.000581   | 0.001516   | 0.054125 | 0.068401        |
| 3   | 0.001080   | 0.000964   | 0.000832   | 0.000701   | 0.000580   | 0.000473   | 0.000380   | 0.000302   | 0.001312   | 0.062244 | 0.068868        |
| 4   | 0.000352   | 0.000340   | 0.000314   | 0.000281   | 0.000246   | 0.000210   | 0.000176   | 0.000146   | 0.001049   | 0.066081 | 0.069195        |
| 5   | 0.000108   | 0.000113   | 0.000112   | 0.000106   | 0.000098   | 0.000088   | 0.000078   | 0.000067   | 0.000800   | 0.067859 | 0.069429        |
| 10  | 0.000000   | 0.000000   | 0.000000   | 0.000000   | 0.000000   | 0.000000   | 0.000000   | 0.000000   | 0.000000   | 0.000159 | 0.069735        |
| 15  | 0.000000   | 0.000000   | 0.000000   | 0.000000   | 0.000000   | 0.000000   | 0.000000   | 0.000000   | 0.000000   | 0.000028 | 0.017674        |
| 20  | 0.000000   | 0.000000   | 0.000000   | 0.000000   | 0.000000   | 0.000000   | 0.000000   | 0.000000   | 0.000000   | 0.000005 | 0.003162        |
| 30  | 0.000000   | 0.000000   | 0.000000   | 0.000000   | 0.000000   | 0.000000   | 0.000000   | 0.000000   | 0.000000   | 0.000000 | 0.000101        |
| $\geq 45$ | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |

$L=8.628551, L_q=7.060092, L_s=14.658480,$

$P_{\text{busy}}=0.107000, W=57.523677, W_q=47.067284$
Table 10. Probability distribution at service completion epoch for a single vacation.

| n  | $p_{n,8}$ | $p_{n,9}$ | $p_{n,10}$ | $p_{n,11}$ | $p_{n,12}$ | $p_{n,13}$ | $p_{n,14}$ | $p_{n,15}$ | $p_n$ | $q_n^*$ |
|----|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-------|-------|
| 0  | 0.030363  | 0.001452  | 0.000861  | 0.000576  | 0.000411  | 0.000303  | 0.000228  | 0.000173  | 0.034367 | 0.007030 |
| 1  | 0.094465  | 0.004549  | 0.002712  | 0.001823  | 0.001307  | 0.000967  | 0.000729  | 0.000696  | 0.107248 | 0.040578 |
| 2  | 0.073472  | 0.003633  | 0.002212  | 0.001514  | 0.001101  | 0.000826  | 0.000629  | 0.001050  | 0.084437 | 0.081371 |
| 3  | 0.057145  | 0.002902  | 0.001805  | 0.001257  | 0.000928  | 0.000705  | 0.000543  | 0.001278  | 0.066563 | 0.079808 |
| 4  | 0.044446  | 0.002317  | 0.001473  | 0.001044  | 0.000782  | 0.000602  | 0.000468  | 0.001141  | 0.052543 | 0.067912 |
| 5  | 0.034569  | 0.001851  | 0.001202  | 0.000867  | 0.000659  | 0.000513  | 0.000404  | 0.0011475 | 0.041540 | 0.055185 |
| 10 | 0.009839  | 0.000601  | 0.000434  | 0.000342  | 0.000280  | 0.000232  | 0.000193  | 0.001279  | 0.013200 | 0.002391 |
| 25 | 0.000226  | 0.000020  | 0.000020  | 0.000020  | 0.000021  | 0.000021  | 0.000021  | 0.000332  | 0.000681 | 0.000000 |
| 50 | 0.000000  | 0.000000  | 0.000000  | 0.000000  | 0.000000  | 0.000000  | 0.000000  | 0.000000  | 0.000000 | 0.000000 |
| 75 | 0.000000  | 0.000000  | 0.000000  | 0.000000  | 0.000000  | 0.000000  | 0.000000  | 0.000000  | 0.000001 | 0.000000 |
| $\geq$ 80 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
Table 11. Probability distribution at arbitrary slot for a single vacation.

| n  | \( p_{n,0} \) | \( p_{n,8} \) | \( p_{n,9} \) | \( p_{n,10} \) | \( p_{n,11} \) | \( p_{n,12} \) | \( p_{n,13} \) | \( p_{n,14} \) | \( p_{n,15} \) | \( q_n \) | \( p_n^{queue} \) |
|-----|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| 0   | 0.001460       | 0.083827       | 0.004695       | 0.003061       | 0.002235       | 0.001729       | 0.001374       | 0.001106       | 0.000896       | 0.005680       | 0.110563       |
| 1   | 0.009892       | 0.068699       | 0.003749       | 0.002498       | 0.001856       | 0.001457       | 0.001173       | 0.000954       | 0.001510       | 0.019534       | 0.111322       |
| 2   | 0.026800       | 0.053432       | 0.002994       | 0.002038       | 0.001541       | 0.001228       | 0.001001       | 0.000823       | 0.001911       | 0.020172       | 0.111940       |
| 3   | 0.043383       | 0.041558       | 0.001291       | 0.001663       | 0.001280       | 0.001035       | 0.000854       | 0.000710       | 0.002153       | 0.017420       | 0.112447       |
| 4   | 0.057494       | 0.032323       | 0.001910       | 0.001357       | 0.001063       | 0.000873       | 0.000729       | 0.000613       | 0.002277       | 0.014228       | 0.112867       |
| 5   | 0.068961       | 0.025140       | 0.001525       | 0.001107       | 0.000883       | 0.000735       | 0.000622       | 0.000529       | 0.002316       | 0.011393       | 0.113211       |
| 10  | 0.007155       | 0.000495       | 0.000400       | 0.000349       | 0.000313       | 0.000282       | 0.000253       | 0.000289       | 0.001899       | 0.002316       | 0.113777       |
| 25  | 0.000164       | 0.000017       | 0.000018       | 0.000021       | 0.000024       | 0.000026       | 0.000027       | 0.000047       | 0.000478       | 0.000000       | 0.000775       |
| 50  | 0.000000       | 0.000000       | 0.000000       | 0.000000       | 0.000000       | 0.000000       | 0.000000       | 0.000000       | 0.000000       | 0.000322       | 0.000000       |
| 75  | 0.000000       | 0.000000       | 0.000000       | 0.000000       | 0.000000       | 0.000000       | 0.000000       | 0.000000       | 0.000000       | 0.000000       | 0.000000       |
| 80  | 0.000000       | 0.000000       | 0.000000       | 0.000000       | 0.000000       | 0.000000       | 0.000000       | 0.000000       | 0.000000       | 0.000000       | 0.000000       |

\( L=9.184420, L_q=4.496307, L_s=9.008487, \)
\( P_{busy}=0.5204122, W=13.120601, W_q=6.423296 \)
Table 12. Probability distribution at service completion epoch for multiple vacations.

| $n$ | $p_{n,8}$ | $p_{n,9}$ | $p_{n,10}$ | $p_{n,11}$ | $p_{n,12}$ | $p_{n,13}$ | $p_{n,14}$ | $p_{n,15}$ | $p_n$ | $q_n$ |
|-----|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-------|-------|
| 0   | 0.008061  | 0.002563  | 0.000922  | 0.000402  | 0.000140  | 0.000099  | 0.000074  | 0.012479  | 0.003209|
| 1   | 0.025081  | 0.008028  | 0.002904  | 0.001273  | 0.000447  | 0.000319  | 0.000299  | 0.039044  | 0.020740|
| 2   | 0.019507  | 0.006411  | 0.002370  | 0.001057  | 0.000382  | 0.000275  | 0.000452  | 0.031038  | 0.052151|
| 3   | 0.015172  | 0.005120  | 0.001934  | 0.000878  | 0.000492  | 0.000237  | 0.000552  | 0.024711  | 0.077131|
| 4   | 0.011800  | 0.004089  | 0.001578  | 0.000729  | 0.000415  | 0.000205  | 0.000319  | 0.019705  | 0.097027|
| 5   | 0.009178  | 0.003266  | 0.001287  | 0.000605  | 0.000350  | 0.000177  | 0.000641  | 0.015741  | 0.112901|
| 10  | 0.002612  | 0.001061  | 0.000465  | 0.000239  | 0.000148  | 0.000084  | 0.000565  | 0.005281  | 0.011420|
| 25  | 0.000069  | 0.000036  | 0.000022  | 0.000014  | 0.000011  | 0.000010  | 0.000009  | 0.000151  | 0.000313  | 0.000000|
| 50  | 0.000000  | 0.000000  | 0.000000  | 0.000000  | 0.000000  | 0.000000  | 0.000000  | 0.000010  | 0.000010  | 0.000000|
| 75  | 0.000000  | 0.000000  | 0.000000  | 0.000000  | 0.000000  | 0.000000  | 0.000000  | 0.000000  | 0.000000  | 0.000000|
| $\geq 80$ | 0.000000  | 0.000000  | 0.000000  | 0.000000  | 0.000000  | 0.000000  | 0.000000  | 0.000000  | 0.000000  | 0.000000|
Table 13. Probability distribution at arbitrary slot for multiple vacations.

| n   | \(p_{n,8}\) | \(p_{n,9}\) | \(p_{n,10}\) | \(p_{n,11}\) | \(p_{n,12}\) | \(p_{n,13}\) | \(p_{n,14}\) | \(p_{n,15}\) | \(q_n\) | \(p_{n,\text{queue}}\) |
|-----|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|--------|-----------------|
| 0   | 0.059036    | 0.020856    | 0.008255    | 0.003929    | 0.002308    | 0.001599    | 0.001218    | 0.000972    | 0.006529 | 0.104702       |
| 1   | 0.045916    | 0.016657    | 0.006736    | 0.003263    | 0.001946    | 0.001365    | 0.001051    | 0.001637    | 0.026954 | 0.105525       |
| 2   | 0.035713    | 0.013303    | 0.005496    | 0.002710    | 0.001640    | 0.001165    | 0.000907    | 0.002075    | 0.043190 | 0.106199       |
| 3   | 0.027776    | 0.010624    | 0.004484    | 0.002250    | 0.001382    | 0.000995    | 0.000783    | 0.002344    | 0.056118 | 0.106756       |
| 4   | 0.021604    | 0.008485    | 0.003659    | 0.001869    | 0.001165    | 0.000849    | 0.000676    | 0.002488    | 0.066426 | 0.107221       |
| 5   | 0.016803    | 0.006777    | 0.002985    | 0.001552    | 0.000982    | 0.000725    | 0.000583    | 0.002539    | 0.074662 | 0.107608       |
| 10  | 0.004782    | 0.002202    | 0.001079    | 0.000613    | 0.000418    | 0.000328    | 0.000279    | 0.002116    | 0.002787 | 0.014604       |
| 25  | 0.000110    | 0.000075    | 0.000051    | 0.000037    | 0.000032    | 0.000030    | 0.000030    | 0.000054    | 0.000000 | 0.000914       |
| 50  | 0.000000    | 0.000000    | 0.000000    | 0.000000    | 0.000000    | 0.000000    | 0.000000    | 0.000037    | 0.000000 | 0.000037       |
| 75  | 0.000000    | 0.000000    | 0.000000    | 0.000000    | 0.000000    | 0.000000    | 0.000000    | 0.000002    | 0.000000 | 0.000002       |
| ≥80 | 0.000000    | 0.000000    | 0.000000    | 0.000000    | 0.000000    | 0.000000    | 0.000000    | 0.000000    | 0.000000 | 0.000000       |

\(L=9.687667\), \(L_q=4.791601\), \(L_s=9.452979\),  
\(p_{\text{busy}}=0.517940\), \(W=13.839524\), \(W_q=6.845144\)
Table 14. Service rates for Case 1 and Case 2.

| r  | $\mu$ (Case 1) | $\mu$ (Case 2) |
|----|----------------|----------------|
| 8  | 0.277778       | 0.199378       |
| 9  | 0.250000       | 0.199378       |
| 10 | 0.227273       | 0.199378       |
| 11 | 0.208333       | 0.199378       |
| 12 | 0.192307       | 0.199378       |
| 13 | 0.178571       | 0.199378       |
| 14 | 0.166666       | 0.199378       |
| 15 | 0.156250       | 0.199378       |

Figure 2. Number of customers in the queue versus traffic intensity.

Figure 3. Busy probability of the server versus traffic intensity.
processing/service time follows geometric distribution with pmf \( \mu_i(1 - \mu_i)^{j-1}, j \geq 1, 8 \leq i \leq 15 \) with \( \mu_i = \frac{1.5}{i+1} \), pgf \( \frac{\mu_x^j}{1-(1-\mu_x)^z} \), so that \( K^{(i)}(x) = \frac{\mu_i(1-\lambda+\lambda x)}{1-(1-\mu_i)(1-\lambda+\lambda x)} \). The excessive delays of the transmission time is bypassed by varying the transmission time with the size of the batches i.e., \( s_8 = 6.000000 \), \( s_9 = 6.666667 \), \( s_{10} = 7.333333 \), \( s_{11} = 8.000000 \), \( s_{12} = 8.666667 \), \( s_{13} = 9.333333 \), \( s_{14} = 10.000000 \), \( s_{15} = 10.666667 \) which shows that the required transmission times increase with the batch size. If the minimum 8 packets are not available for the onward transmission, the server enters into the vacation period which has geometric distribution with pmf \( (\nu_j) = p(1-p)^{j-1}, j \geq 1 \) with \( p = 0.6 \), pgf \( \frac{\nu_x^j}{1-(1-p)^z} \), mean \( \hat{(\nu)} = 1.666667 \) and hence \( H(x) = \frac{0.18+0.42x}{0.88-0.28x} \). This vacation time period has been used to resolve some technical discrepancy of the system. The main advantage of this scheme of transmission of packets (e.g., from BS to SS) in batches is the efficiency and improvement of the quality of service as the construction of only one header per aggregated batch instead of one header per single information unit reduce the time as well as the cost.

The joint distribution at service completion and arbitrary epochs along with some performance measures are displayed in [Tables 10–13]. Moreover, from [Tables 11–13], it is clearly visible that the average waiting time of a packet in the queue/system \( (W_q=6.423296, W=13.120601) \) is less for the single vacation policy as compared to multiple vacation policy \( (W_q=6.845144, W=13.839524) \). The same conclusion can be drawn for the average number of packets waiting in the queue/system. The probability that the network system is busy in the case of a single vacation \( (P_{busy}=0.5204122) \) is higher as compared to multiple vacations \( (P_{busy}=0.517940) \). These observations turn out to be significant are very useful for the system designer.

After the tabular representation of the numerical results, we now turn our focus to study the behavior of the system through some graphical representation. In order to show the significance of our proposed model, we figure out a comparison between batch-size dependent and independent service policies by considering two cases, viz.

**Case I:** The service times of the batches vary with the size of the batch undergoing service.

![Figure 4. Waiting time of a customer in the system versus arrival rate.](image-url)
Case 2: The mean service times of the batches are the same irrespective of the size of the batches. The same service rates for all the batches are the weighted average of the service rates of Case 1. The weighted average is calculated by the formula $\sum_{r=0}^{b} rP_{r}$.

For Figures 2–4, the input parameters are taken as: service time follows NB distribution, vacation time follows geometric distribution, $a = 8, b = 15$, and service rates for two cases are precisely given in Table 14.

In Figures 2 and 3, the average number of customers in the queue at arbitrary slot ($L_{q}$) and the probability that the server is busy ($P_{busy}$) are depicted versus the traffic load $\rho$, respectively. From the figures 2 and 3, it is easily visible that $L_{q}$ and $P_{busy}$ are higher for the Case 2 as compared to Case 1 which suggests that batch-size-dependent service time is more significant rather than batch-size-independent service policy. Also it may be noted here that $L_{q}$ is slightly greater in case of multiple vacation which is in the expected line as the availability of the server in case of single vacation is more compared to multiple vacation. Similar type of conclusion has been drawn in G. K. Gupta et al. (2019).

From the Figure 3, one may observe that $P_{busy}$ is higher for single vacation than the corresponding multiple vacation for Case 2 and both of them become equal and stable as $\rho$ increases. This behavior of the system is also in the expected line and G. K. Gupta et al. (2019) and Samanta et al. (2007a) pointed out the similar type of observation.

In Figure 4, we sketch the average waiting time of a customer in the system for different values of $\lambda$. One can notice that waiting time is less in Case 1 compared to Case 2, which eventually justifies the potentiality of batch-size-dependent service policy from the application perspective. Moreover, it may be pointed out that a customer has to wait more time in the system in case of multiple vacation which is quite obvious behavior of the system.

### 9.1. System cost optimization

It is frequently observed that the system designers or vendors want to minimize the total system cost at the pre-implantation stage. Keeping this in mind, in this section, our foremost concern is to demonstrate a comparison between the batch-size-dependent (the present model, (Case 1)) and
batch-size-independent model (Case 2) through the cost of the corresponding models. We first associate some cost to the system characteristics viz.,

- $C_w$ be the waiting/holding cost per customer per unit time in the queue.
- $C_{idle}$ be the idleness cost of the server per unit time.
- $C_{operating}$ be the operating (serving/transmission) cost per unit time.

Hence in the long run the total system cost ($TSC$) is presented by:

$$TSC = C_wL_q + C_{idle}((1 - \delta_p)P_{dor} + Q_{vac}) + C_{operating}L_s$$  \hspace{1cm} (62)

Determination of the optimal value of the cost function through some conventional optimization techniques is very difficult as the analytic expressions of $L_q$, $P_{dor}$, $Q_{vac}$, $L_s$ are not in our grip. On account of this, from the expression presented in (62), the associated cost of the system is calculated numerically. We consider the cost parameter as: $C_w = 5$ unit, $C_{idle} = 2$ unit, $C_{operating} = 1$ unit, and other parameters are the same as taken for Figure 2–4.

The Figure 5 exhibits that the total system cost is always less in Case 1 compared to Case 2. This outcome suggests that vendor/system designer should adopt the batch-size-dependent service policy as it produces less cost rather than batch-size-independent service policy. It may also be remarked here that total system cost is higher for the multiple vacation compared to single vacation. This behavior is also relevant in the sense that the more availability of the server in case of single vacation produces less system cost.

10. Conclusion

In this paper, we have analyzed a batch service queue with infinite waiting space, Bernoulli arrival process, batch-size-dependent service policy, single and multiple vacation. The generating function approach with double variables is adopted to derive the bivariate probability generating function of queue length and server content distribution at service completion epoch. We have also presented the complete extraction of the probabilities in a simple and elegant way. The joint distribution at arbitrary slot is also acquired using which several noteworthy marginal distribution and performance measures are reported. The analytic procedure and results are illustrated numerically and graphically through the inclusion of significant distributions which can cope with real-life circumstances. Towards the end it is hoped that the results will be fruitful to the vendors/system designers. The inclusion of the batch Bernoulli arrival process or discrete Markovian arrival process will make the investigation very interesting, and in future, we will keep track for the analysis of those models with single/multiple vacation.

Acknowledgments

The authors would like to thank the anonymous referees for their valuable comments and suggestions which led to the presentation and improvements of the paper.

Disclosure statement

No potential conflict of interest was reported by the authors.

Notes on contributors

Nilanjan Nandy is currently a research scholar in the Department of Mathematics at Visvesvaraya National Institute of Technology, Nagpur, India. Nandy completed his Master degree in Mathematics from University of Calcutta. His research interests are directed towards the analysis of discrete and continuous time queueing models from analytical and computational perspective.
Sourav Pradhan is currently a faculty of the Department of Mathematics at Visvesvaraya National Institute of Technology, Nagpur, India. Pradhan did his Master degree in Mathematics from the Indian Institute of Technology, Madras, Chennai, India, in 2011, and Ph.D. from Department of Mathematics, Indian Institute of Technology Kharagpur, India in 2016. His main research interest is in analytic and computational aspects of Queueing Models. He has published research articles in various reputed journals such as Journal of the Korean Statistical Society, OPSEARCH, Performance Evaluation, Annals of Operation Research, Communication in Statistics-Theory & Methods, etc.

ORCID

Sourav Pradhan http://orcid.org/0000-0002-0479-3496

References

Abolnikov, L., & Dukhovny, A. (1991). Markov chains with transition delta-matrix: Ergodicity conditions, invariant probability measures and applications. International Journal of Stochastic Analysis, 4(4), 333–355. https://doi.org/10.1155/S104895339100254

Alfa, A. S. (2010). Queueing theory for telecommunications: Discrete time modelling of a single node system. Springer Science & Business Media.

Arumuganathan, R., & Jeyakumar, S. (2004). Analysis of a bulk queue with multiple vacations and closedown times. International Journal of Information and Management Sciences, 15(1), 45–60.

Arumuganathan, R., & Jeyakumar, S. (2005). Steady state analysis of a bulk queue with multiple vacations, setup times with N-policy and closedown times. Applied Mathematical Modelling, 29(10), 972–986. https://doi.org/10.1016/j.apm.2005.02.013

Banerjee, A., & Gupta, U. C. (2012). Reducing congestion in bulk-service finite-buffer queueing system using batch-size-dependent service. Performance Evaluation, 69(1), 53–70. https://doi.org/10.1016/j.peva.2011.09.002

Banerjee, A., Gupta, U. C., & Chakravarthy, S. R. (2015). Analysis of a finite-buffer bulk-service queue under Markovian arrival process with batch-size-dependent service. Computers & Operations Research, 60., 138–149. https://doi.org/10.1016/j.cor.2015.02.012

Banerjee, A., Gupta, U. C., & Goswami, V. (2014). Analysis of finite-buffer discrete-time batch-service queue with batch-size-dependent service. Computers & Industrial Engineering, 75., 121–128. https://doi.org/10.1016/j.cie.2014.06.009

Banerjee, A., Gupta, U. C., & Sikdar, K. (2013). Analysis of finite-buffer bulk-arrival bulk-service queue with variable service capacity and batch-size-dependent service. International Journal of Mathematics in Operational Research, 5 (3), 358–386. https://doi.org/10.1504/IJMOR.2013.053629

Banik, A. D., Gupta, U. C., & Pathak, S. S. (2006). Finite buffer vacation models under e-limited with limit variation service and markovian arrival process. Operations Research Letters, 34(5), 539–547. https://doi.org/10.1016/j.orl.2005.08.006

Bank, B., & Samanta, S. (2020). Analytical and computational studies of the BMAP/G(α,Y)/1 queue. Communications in Statistics-Theory and Methods, pages 1–29. https://doi.org/10.1080/03610926.2019.1708941

Barbhuiya, F. P., & Gupta, U. (2020). Analytical and computational aspects of the infinite buffer single server N policy queue with batch renewal input. Computers & Operations Research, 118, 104916. https://doi.org/10.1016/j.cor.2020.104916

Barbhuiya, F. P., & Gupta, U. C. (2019). Discrete-time queue with batch renewal input and random serving capacity rule: GI^V/Geo^V/1. Queueing Systems, 91(3-4), 347–365. https://doi.org/10.1007/s11134-019-09600-7

Brunel, H., & Kim, B. G. (1993). Discrete-time models for communication systems including ATM. Kluwer Academic Publishers.

Chang, S. H., & Choi, D. W. (2005). Performance analysis of a finite-buffer discrete-time queue with bulk arrival, bulk service and vacations. Computers & Operations Research, 32(9), 2213–2234. https://doi.org/10.1016/j.cor.2004.01.004

Chaudhry, M. L., & Gupta, U. C. (2003). Analysis of a finite-buffer bulk-service queue with discrete-markovian arrival process: D-MAP/G(α,b)/1/N. Naval Research Logistics (NRL), 50(4), 345–363. https://doi.org/10.1002/nav.10047

Choi, B. D., & Han, D. H. (1994). GI/M(α,b)/1 queues with server vacations. Journal of the Operations Research Society of Japan, 37(3), 171–181. https://doi.org/10.15807/jorsj.37.171

Claeys, D., Laevens, K., Walraevens, J., & Bruneel, H. (2010a). Complete characterisation of the customer delay in a queuing system with batch arrivals and batch service. Mathematical Methods of Operations Research, 72(1), 1–23. https://doi.org/10.1007/s00186-009-0297-2
Claeys, D., Steyaert, B., Walraevens, J., Laevens, K., & Bruneel, H. (2013a). Analysis of a versatile batch-service queueing model with correlated arrivals in the arrival process. *Performance Evaluation*, 70(4), 300–316. https://doi.org/10.1016/j.peva.2012.12.004

Claeys, D., Steyaert, B., Walraevens, J., Laevens, K., & Bruneel, H. (2013b). Tail probabilities of the delay in a batch-service queueing model with batch-size dependent service times and a timer mechanism. *Computers & Operations Research*, 40(5), 1497–1505. https://doi.org/10.1016/j.cor.2012.10.009

Claeys, D., Walraevens, J., Laevens, K., & Bruneel, H. (2010b). A queueing model for general group screening policies and dynamic item arrivals. *European Journal of Operational Research*, 207(2), 827–835. https://doi.org/10.1016/j.ejor.2010.05.042

Claeys, D., Walraevens, J., Laevens, K., & Bruneel, H. (2011). Analysis of threshold-based batch-service queueing systems with batch arrivals and general service times. *Performance Evaluation*, 68(6), 528–549. https://doi.org/10.1016/j.peva.2011.03.007

Doshi, B. T. (1986). Queueing systems with vacations - a survey. *Queueing Systems*, 1(1), 29–66. https://doi.org/10.1007/BF01149327

Germ, R., & van Foreest, N. (2013). Analysis of finite-buffer state-dependent bulk queues. *OR Spectrum*, 35(3), 563–583. https://doi.org/10.1007/s00291-012-0282-7

Goswami, V., Gupta, U. C., & Samanta, S. K. (2006). Analyzing discrete-time bulk-service Geo/Geo^d/m queue. *RAIRO-Operations Research*, 40(3), 267–284. https://doi.org/10.1051/ro:2006021

Goswami, V., & Vijaya Laxmi, P. (2011). Performance analysis of a renewal input bulk service queue with accessible and non-accessible batches. *Quality Technology & Quantitative Management*, 8(2), 87–100. https://doi.org/10.1080/16843703.2011.11673249

Gravey, A., Louvion, J.-R., & Buyer, P. (1990). On the Geo/D/1 and Geo/D/1/n queues. *Performance Evaluation*, 11(2), 117–125. https://doi.org/10.1016/0166-5316(90)90018-E

Gupta, G. K., Banerjee, A., & Gupta, U. C. (2019). On finite-buffer batch-size-dependent bulk service queue with queue-length dependent vacation. *Quality Technology & Quantitative Management*, pages 1–27. https://doi.org/10.1080/16843703.2019.1675568

Gupta, U. C., Samanta, S. K., & Goswami, V. (2014). Analysis of a discrete-time queue with load dependent service under discrete-time markovian arrival process. *Journal of the Korean Statistical Society*, 43(4), 545–557. https://doi.org/10.1016/j.jkss.2014.03.003

Gupta, U. C., Samanta, S. K., Sharma, R. K., & Chaudhry, M. L. (2007). Discrete-time single-server finite-buffer queues under discrete Markovian arrival process with vacations. *Performance Evaluation*, 64(1), 1–19. https://doi.org/10.1016/j.peva.2006.01.001

Gupta, U. C., & Sikdar, K. (2004). The finite-buffer M/G/1 queue with general bulk-service rule and single vacation. *Performance Evaluation*, 57(2), 199–219. https://doi.org/10.1016/j.peva.2003.10.004

Gupta, U. C., & Sikdar, K. (2005). On the finite buffer bulk service M/G/1 queue with multiple vacations. *Journal of Probability and Statistical Science*, 3(2), 175–189.

Gupta, U. C., & Sikdar, K. (2006). Computing queue length distributions in MAP/G/1/N queue under single and multiple vacation. *Applied Mathematics and Computation*, 174(2), 1498–1525. https://doi.org/10.1016/j.amc.2005.07.001

Ho Chang, S., Won Choi, D., & Kim, T.-S. (2004). Performance analysis of a finite-bulk buffer arrival and bulk-service queue with variable server capacity. *Stochastic Analysis and Applications*, 22(5), 1151–1173. https://doi.org/10.1081/SAP-200026427

Hunter, J. J. (1983). *Mathematical techniques of applied probability: Discrete time models: Basic theory*. Vol II.

Jeyakumar, S., & Arumuganathan, R. (2011). A non-Markovian bulk queue with multiple vacations and control policy on request for re-service. *Quality Technology & Quantitative Management*, 8(3), 253–269. https://doi.org/10.1080/16843703.2011.11673258

Jeyakumar, S., & Senthilnathan, B. (2012). A study on the behaviour of the server breakdown without interruption in a M/G\((a,b)\)/1 queueing system with multiple vacations and closedown time. *Applied Mathematics and Computation*, 219(5), 2618–2633. https://doi.org/10.1016/j.amc.2012.08.096

Ke, J. C. (2007). Batch arrival queues under vacation policies with server breakdowns and startup/downs time. *Applied Mathematical Modelling*, 31(7), 1282–1292. https://doi.org/10.1016/j.apm.2006.02.010

Ke, J.-C., Chang, F.-M., & Liu, T.-H. (2019). M/M/c balkin retrial queue with vacation. *Quality Technology & Quantitative Management*, 16(1), 54–66. https://doi.org/10.1080/16843703.2017.1365280

Ke, J.-C., Wu, C.-H., & Zhang, Z. G. (2010). Recent developments in vacation queueing models: A short survey. *International Journal of Operations Research*, 7(4), 3–8.

Lee, H. W., Lee, S. S., Chae, K. C., & Nadarajan, R. (1992). On a batch service queue with single vacation. *Applied Mathematical Modelling*, 16(1), 36–42. https://doi.org/10.1016/0307-904X(92)90113-H

Lee, H. W., Lee, S. S., & Chae, K.-C. (1996). A fixed-size batch service queue with vacations. *International Journal of Stochastic Analysis*, 9(2), 205–219. https://doi.org/10.1155/S104895339600196

Lee, H. W., Lee, S. S., Park, J. O., & Chae, K.-C. (1994). Analysis of the M/G/1 queue by N-policy and multiple vacations. *Journal of Applied Probability*, 31(2), 476–496. https://doi.org/10.2307/3215040
Neuts, M. F. (1967). A general class of bulk queues with poisson input. *The Annals of Mathematical Statistics*, 38(3), 759–770. https://doi.org/10.1214/aoms/1177698869

Pradhan, S. (2020a). A discrete-time batch transmission channel with random serving capacity under batch-size-dependent service. *International Journal of Computer Mathematics: Computer Systems Theory*, 5(3), 1–23. https://doi.org/10.1080/23799927.2020.1792998

Pradhan, S. (2020b). On the distribution of an infinite-buffer queueing system with versatile bulk-service rule under batch-size-dependent service policy: $M/G_a^{(a,b)}/1$. *International Journal of Mathematics in Operational Research*, 16(3), 407–434. https://doi.org/10.1504/IJMOR.2020.106908

Pradhan, S., & Gupta, U. (2020). Stationary queue and server content distribution of a batch-size-dependent service queue with batch Markovian arrival process: $BMAP/G_a^{(a,b)}/1$. *Communications in Statistics: Theory and Methods*, 1–28. pages ... https://doi.org/10.1080/03610926.2020.1813304

Pradhan, S., & Gupta, U. C. (2019). Arrival of an infinite-buffer batch-size-dependent service queue with Markovian arrival process. *Annals of Operations Research*, 277(2), 161–196. https://doi.org/10.1007/s10479-017-2476-5

Pradhan, S., & Gupta, U. C. (2020). Stationary distribution of an infinite-buffer batch-arrival and batch-service queue with random serving capacity and batch-size-dependent service. *International Journal of Operational Research*, 40(1): 1–31. https://doi.org/10.1504/IJOR.2020.10025128

Pradhan, S., Gupta, U. C., & Samanta, S. K. (2016). Analyzing an infinite buffer batch arrival and batch service queue under batch-size-dependent service policy. *Journal of the Korean Statistical Society*, 45(1), 137–148. https://doi.org/10.1016/j.jkss.2015.08.004

Reddy, G. K., & Anitha, R. (1998). Markovian bulk service queue with delayed vacations. *Computers Operations Research*, 25(12), 1159–1166. https://doi.org/10.1016/S0305-0548(98)00003-3

Reddy, G. K., & Anitha, R. (1999). Non-Markovian bulk service queue with different vacation policies. *Information and Management Sciences*, 10(4), 1–17.

Reddy, G. K., Nadarajan, R., & Arumuganathan, R. (1998). Analysis of a bulk queue with N-policy multiple vacations and setup times. *Computers & Operations Research*, 25(11), 957–967. https://doi.org/10.1016/S0305-0548(97)00098-1

Samanta, S. K., Chaudhry, M. L., & Gupta, U. C. (2007a). Discrete-time Geo$^X/G^{(a,b)}/1/N$ queues with single and multiple vacations. *Mathematical and Computer Modelling*, 45(1–2), 93–108. https://doi.org/10.1016/j.mcm.2006.04.008

Samanta, S. K., Gupta, U. C., & Sharma, R. K. (2007b). Analyzing discrete-time D- $BMAP/G/1/N$ queue with single and multiple vacations. *European Journal of Operational Research*, 182(1), 321–339. https://doi.org/10.1016/j.ejor.2006.09.031

Sikdar, K., & Gupta, U. C. (2005a). Analytic and numerical aspects of batch service queues with single vacation. *Computers & Operations Research*, 32(4), 943–966. https://doi.org/10.1016/j.cor.2003.09.007

Sikdar, K., & Gupta, U. C. (2005b). The queue length distributions in the finite buffer bulk-service $MAP/G/1$ queue with multiple vacations. *Top*, 13(1), 75. https://doi.org/10.1007/BF02578989

Tadj, L., Choudhury, G., & Tadj, C. (2006). A quorum queueing system with a random setup time under n-policy and with bernoulli vacation schedule. *Quality Technology & Quantitative Management*, 3(2), 145–160. https://doi.org/10.1080/16843703.2006.11673106

Tadj, L., & Ke, J. (2008). A hysteretic bulk quorum queue with a choice of service and optional re-service. *Quality Technology & Quantitative Management*, 5(2), 161–178. https://doi.org/10.1080/16843703.2008.11673394

Vadivu, A. S., & Arumuganathan, R. (2015). Cost analysis of $MAP/G^{(a,b)}/1/N$ queue with multiple vacations and closedown times. *Quality Technology Quantitative Management*, 12(4), 605–626. https://doi.org/10.1080/16843703.2015.11673438

Vijayashree, K., & Anjuka, A. (2018). Stationary analysis of a fluid queue driven by an $M/M/1$ queue with working vacation. *Quality Technology Quantitative Management*, 15(2), 187–208. https://doi.org/10.1080/16843703.2016.1208935

Yu, M., & Alfa, A. S. (2015). Algorithm for computing the queue length distribution at various time epochs in D-$MAP/G^{(a,b)}/1/N$ queue with batch-size-dependent service time. *European Journal of Operational Research*, 244(1), 227–239. https://doi.org/10.1016/j.ejor.2015.01.056