

RESEARCH ARTICLE

Queues with Dropping Functions and General Arrival Processes

Andrzej Chydzinski*, Pawel Mrozowski

Institute of Informatics, Silesian University of Technology, Gliwice, Poland

* andrzej.chydzinski@polsl.pl

Abstract

In a queueing system with the dropping function the arriving customer can be denied service (dropped) with the probability that is a function of the queue length at the time of arrival of this customer. The potential applicability of such mechanism is very wide due to the fact that by choosing the shape of this function one can easily manipulate several performance characteristics of the queueing system. In this paper we carry out analysis of the queueing system with the dropping function and a very general model of arrival process—the model which includes batch arrivals and the interarrival time autocorrelation, and allows for fitting the actual shape of the interarrival time distribution and its moments. For such a system we obtain formulas for the distribution of the queue length and the overall customer loss ratio. The analytical results are accompanied with numerical examples computed for several dropping functions.



OPEN ACCESS

Citation: Chydzinski A, Mrozowski P (2016) Queues with Dropping Functions and General Arrival Processes. PLoS ONE 11(3): e0150702. doi:10.1371/journal.pone.0150702

Editor: Quan Zou, Tianjin University of Science and Technology, CHINA

Received: November 2, 2015

Accepted: February 17, 2016

Published: March 4, 2016

Copyright: © 2016 Chydzinski, Mrozowski. This is an open access article distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Data Availability Statement: All relevant data are within the paper.

Funding: The authors have no support or funding to report.

Competing Interests: The authors have declared that no competing interests exist.

1 Introduction

Consider a simple queueing system with a stream of arriving customers, the queue of customers waiting for service and one service station (server) performing the service that takes random time. In order to control the performance of such system (e.g. to keep the mean queue length below 10) we have three options:

- (a) manipulate dynamically the service rate;
- (b) manipulate dynamically the customer arrival rate;
- (c) deny the service to some customers and not allow them to the queue.

All the listed approaches are equally good for control purposes in the sense that we can achieve the same results using each of them. The difference between them is in the practical applicability, which depends on the character and purpose of the queueing system of interest.

In many everyday-life queues, approach (a) can be used. At a bank for instance, additional cashiers can be assigned for serving an exceptionally long queue, thus multiplying the service rate. For serving exceptional passenger traffic, a bus (train, ferry) of larger capacity can be used occasionally, etc.

Usually, the application of (a) is connected with some additional costs (salary, equipment, energy consumption etc.). Of course, it can be applied only from time to time, using some sophisticated policy based on current queue size or predictions on its future size.

In some cases however, it is not possible to use (a). In networking for instance, when a queue of packets is being transmitted at a router's output interface, it is not possible to enlarge the throughput of the physical output link on demand.

Approach (b) cannot be used in most everyday-life applications of queueing systems. This is due to the fact, that the operator of the queue has usually no means to reduce quickly the customer arrival rate (think of a bank, for instance). On the other hand, method (b) works very well in networking. In particular, the Internet hosts, which use the TCP protocol, are forced to reduce their packet sending rates immediately, when the network congestion is expected. This proved to be an efficient way for preventing congestion collapses of the Internet, which were observed in its early years of operation.

As for (c), it is certainly the simplest method. Rejecting a customer is not technically difficult and does not require any spare resources, as method (a). Its disadvantage is that a fraction of customers leaves the system unserved. Therefore, it can be used in these applications only, in which such losses are tolerated. This is the case of the Internet, where packet losses are unavoidable anyway due to buffer overflows and traffic burstiness, so they may be as well caused rather in a planned, controlled manner. There are also many everyday-life examples, in which approach (c) can be used. For instance, when a waiting line at a call center is long, it might be better to reject a customer at once, instead of allowing him to the queue and keeping online for a long time before serving. Of course, there are also every-day life examples in which customers cannot be rejected—in those cases we are practically limited to method (a).

The queueing models of type (a) and their solutions can be found, for instance, in [1–4], while the models of type (b), in [5–7].

In this paper we deal with type (c) of controlling the performance of a queueing system. The literature on this method will be discussed later.

In order to decide in method (c), whether a customer has to be accepted to the queue or rejected, a large number of different disciplines can be proposed. One of the most natural ones is that each customer can be rejected randomly, with the probability that depends on the length of the queue upon the arrival of this customer (see Fig 1). In this paper we deal with such discipline. The function mapping the queue lengths into probabilities is called the dropping function and will be denoted by $d(n)$, where n is the queue length. In most practical situations the dropping function would be non-decreasing, i.e. the longer the queue, the more likely the customer is rejected. (However, this assumption is not necessary in our analysis).

The idea of a queueing system with the dropping function was for the first time proposed in networking, in the RED algorithm [8], which uses linear dropping function to drop packets incoming to an Internet router. Following the first, linear dropping function, some other types of functions were studied: doubly linear [9], exponential [10] and polynomial [11]. In addition, a large number of studies were carried out on systems, in which the packet dropping probability is not a simple function of the queue length, but depends on several other factors, often in a very complex way (see e.g. [12–20]). In networking, this is called the active queue management (AQM).

However, besides networking, the dropping functions have great potential for applications in many other systems involving queueing of customers (or jobs, tasks etc.). This is connected with their powerful control capabilities. Namely, by shaping the dropping function according to our needs, we can keep the mean queue length, its variance, the loss ratio and several other performance characteristics at some required levels.

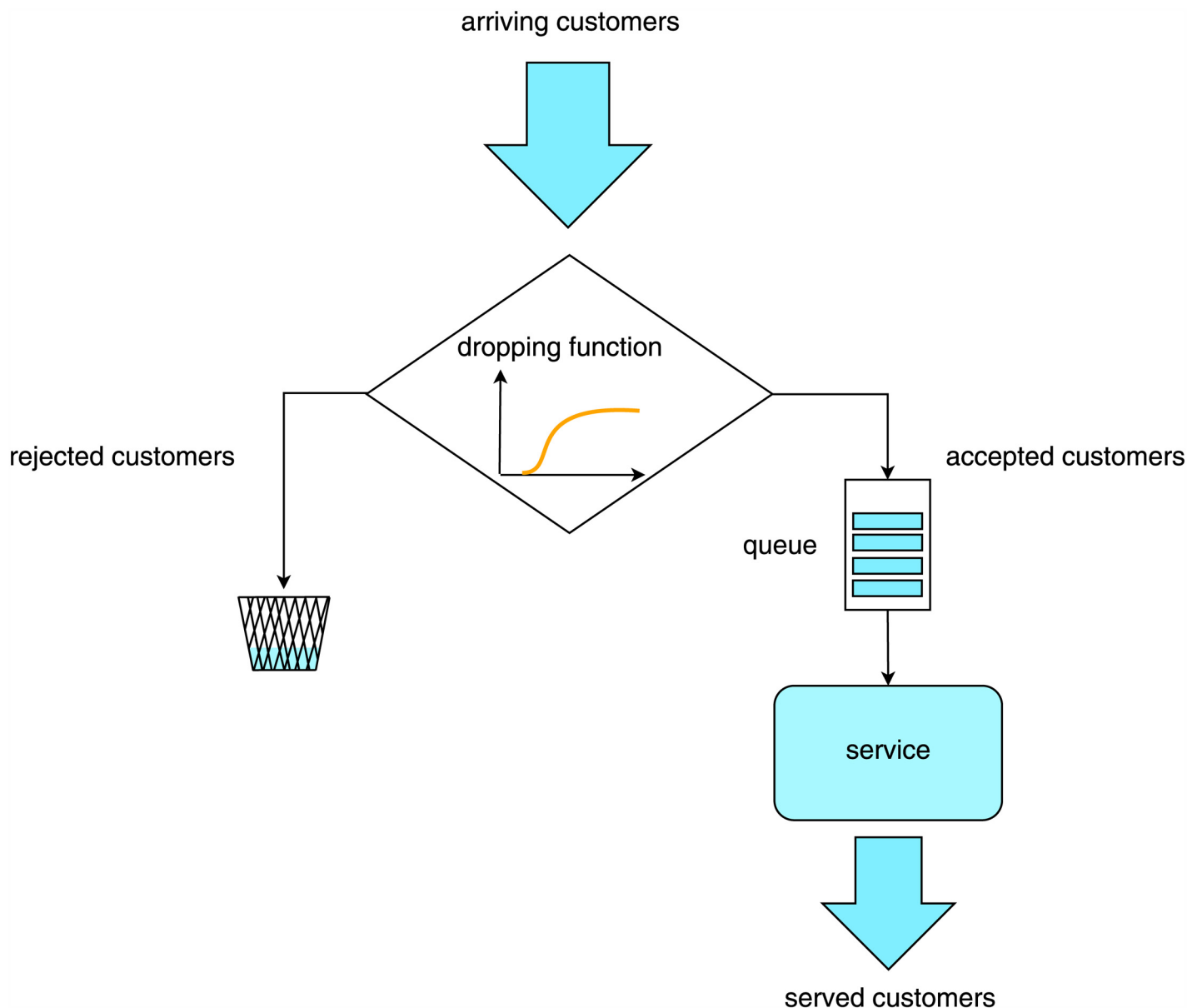


Fig 1. The idea of the queueing system with the dropping function.

doi:10.1371/journal.pone.0150702.g001

The literature on analytical solutions of queueing systems with dropping functions is not extensive. What is important, all the previous works were devoted to systems with very simple arrival processes, namely Poisson or renewal processes without interarrival times correlation. In particular, in [21], an approximate analysis of the queue with batch Poisson arrivals, linear dropping function and exponential service times was carried out. In [22], an approximate analysis of the system with batch renewal arrivals and exponential service times was performed. In [23] an exact analysis of the queue with Poisson arrivals, arbitrary dropping function and arbitrary service times was presented for systems in the equilibrium. In [24], an exact analysis of the queue with renewal arrivals, arbitrary dropping function and exponential service times was shown. Then, in [25], an analysis of the system with Poisson arrivals and arbitrary dropping

function was carried out in the transient case. Finally, in [26] a solution of the system with Poisson arrivals and general distribution of the job size was shown.

Due to the simplicity of the arrival processes, all the mentioned models are not adequate for modeling of many real systems. For instance, it is well known that streams of packets in the Internet have strongly autocorrelated structure (see e.g. [27–29])—not only are the interarrival times correlated, but also the packet sizes may be correlated with the local intensity of the arrival stream. On the other hand, it is well known that not taking these properties into account may lead to an optimistic underestimation of the queueing characteristics by several orders of magnitude (see e.g. [30]). For these reasons, the previous analytical results on queues with dropping functions are of little use in networking and in other applications with sophisticated, autocorrelated traffic of customers.

The novelty of this paper is that we carry out analysis of the system with:

- a very general model of the arrival process, taking into account the correlation of the interarrival times, correlation between local rate and job (customer) size, batch arrivals, the actual shape of the interarrival time distribution and its moments;
- arbitrary distribution of the customer service time;
- arbitrary dropping function.

To the best of the authors' knowledge, there are no results of this generality in the literature.

Given the presented requirements, a rather obvious choice of the model of the arrival process is the batch Markovian arrival process (BMAP, see [31]). It not only fulfills all the presented requirements, but also is analytically tractable and several efficient procedures for fitting precisely the BMAP parameters to the real, observed arrival processes have been proposed, [32–35].

As for the queueing characteristics of interest, we present analytical formulas for the two most important ones: the queue length distribution and the loss ratio (the fraction of rejected customers). The results on the queue length are presented in the form which allows obtaining both transient and stationary distributions.

In addition to analytical solutions, we present numerical examples for several dropping functions.

The remaining part of the paper is structured as follows. In Section 2, a formal description of the queueing system is presented, including the definition and basic characteristics of the batch Markovian arrival process. Then, in Section 3, the main results on the distribution of the queue length and loss ratio are proven. In Section 4, calculations of two auxiliary functions, H and q , needed to obtain numerical values of queue lengths and loss ratios, are presented. In Section 5, numerical examples are shown. Finally, remarks concluding the paper are gathered in Section 6.

2 Model of the queue

We deal with a queueing system with single service station, whose arrival process is the batch Markovian arrival process (defined below). The customers are served in the arrival order; those who cannot be served immediately form a queue. The service time is random and can have an arbitrary distribution, with distribution function denoted by $F(t)$. It is assumed, that the capacity of the waiting room (buffer) is limited and equal to b customers. This means that the number of customers present in the system, in the queue and the service position, must not exceed b . A customer who arrives when the waiting room is full is rejected and never returns.

Moreover, a customer who arrives when the waiting room is not full can be rejected with probability $d(n)$, where n is the queue length at the time of arrival of this customer (including the service position, if occupied). Function $d(n)$ will be called the dropping function. The dropping function assumes values in $[0, 1]$ for $n = 0, \dots, b-1$. As the waiting room is finite, the dropping function must fulfill the condition: $d(n) = 1$ for $n \geq b$. The queue length (including the service position, if occupied) at time t will be denoted by $X(t)$. If $X(0) > 0$, then it is assumed that the time origin corresponds to a service completion. The load offered to the queueing system is defined in a natural way as:

$$\rho = \Lambda \int_0^{\infty} x dF(x),$$

where Λ is the average rate of the arrival process, given by Eq (5).

The batch Markovian arrival process was proposed (using different parameterization) in [36] and initially called N-process. An easier-to-use parameterization of it was introduced in [31]—since then the acronym BMAP has been used. A rich description of the BMAP process, its characteristics and the bibliography can be found in [37].

Let I be the unit matrix of size $m \times m$, $\mathbf{0}$ be the square matrix of zeroes, while $\mathbf{1}$ be the column vector of size m with all entries equal to 1.

BMAP is defined as the two-dimensional Markov chain $(N(t), J(t))$ on state space $\{(i, j): i \geq 0, 1 \leq j \leq m\}$, with the intensity matrix Q in the following form:

$$Q = \begin{bmatrix} D_0 & D_1 & D_2 & D_3 & \cdots \\ & D_0 & D_1 & D_2 & \cdots \\ & & D_0 & D_1 & \cdots \\ & & & \ddots & \ddots \end{bmatrix},$$

where $D_k, k \geq 0$, are $m \times m$ matrices. Moreover, D_k for $k \geq 1$ are non-negative, D_0 has negative entries on its diagonal and non-negative elsewhere, matrix D defined as

$$D = \sum_{k=0}^{\infty} D_k$$

is an irreducible intensity matrix and $D \neq D_0$.

The components of the Markov chain $(N(t), J(t))$ are to be interpreted as follows: $N(t)$ is the total number of customers arriving in interval $(0, t]$, while $J(t)$ is the state at time t of the one-dimensional Markov chain modulating the customer arrival process, whose intensity matrix is D . The stationary distribution of $J(t)$ will be denoted by π , and, as always, we have:

$$\pi D = [0, \dots, 0], \quad \pi \mathbf{1} = 1.$$

The evolution of the BMAP process can be presented in the following manner. Say, at $t = 0$ the modulating chain J is in some state i . The modulating chain remains in this state for some random time, which is exponentially distributed with mean $1/\mu_i$ where

$$\mu_i = -(D_0)_{ii}. \quad (1)$$

After this random time, the state of the modulating chain changes into k , possibly with an

arrival of a batch of j customers. Precisely, this happens with probability $p_i(j, k)$, where:

$$p_i(0, i) = 0, \quad 1 \leq i \leq m, \quad (2)$$

$$p_i(0, k) = \frac{1}{\mu_i} (D_0)_{ik}, \quad 1 \leq i, k \leq m, \quad k \neq i, \quad (3)$$

$$p_i(j, k) = \frac{1}{\mu_i} (D_j)_{ik}, \quad 1 \leq i, k \leq m, \quad j \geq 1. \quad (4)$$

(As it is possible that $p_i(0, k) > 0$, a change of the modulating state without arrival of any customers can happen as well). After the change, the modulating chain remains in state k for some random time, which is exponentially distributed with mean $1/\mu_k$, and so on.

The most important characteristics of the BMAP process can be computed as follows. The average rate of the process, Λ , is

$$\Lambda = \pi \sum_{k=1}^{\infty} k D_k \mathbf{1}, \quad (5)$$

while the average rate of arrivals of batches is

$$\Lambda_g = \pi(-D_0)\mathbf{1}.$$

Therefore, the average size of a batch is equal to

$$\eta = \frac{\Lambda}{\Lambda_g}.$$

The variance of the time between arrivals of consecutive batches equals

$$\text{Var} = -\frac{2}{\Lambda_g} \pi D_0^{-1} \mathbf{1} - \frac{1}{\Lambda_g^2}.$$

The k -lag autocorrelation of batch interarrival times is

$$\text{Corr}(k) = p D_0^{-1} C (C^{k-1} - \mathbf{1} p) D_0^{-1} C \mathbf{1} / \text{Var},$$

with $C = -D_0^{-1}(D - D_0)$, where p is the stationary vector for intensity matrix $C-I$, i.e. it fulfills

$$p(C - I) = [0, \dots, 0], \quad p \mathbf{1} = 1.$$

The counting function of the BMAP process is defined as

$$P_{ij}(n, t) = \mathbb{P}(N(t) = n, J(t) = j | N(0) = 0, J(0) = i),$$

where \mathbb{P} denotes probability. Using the matrix notation, we can denote the counting functions as

$$P(n, t) = [P_{ij}(n, t)]_{ij}, \quad 1 \leq i, j \leq m.$$

The generating function of the counting function is then

$$P^*(z, t) = \sum_{n=0}^{\infty} P(n, t) z^n = e^{D(z)t},$$

where

$$D(z) = \sum_{k=0}^{\infty} z^k D_k, \quad |z| \leq 1.$$

BMAP processes have been successfully used in the modeling of traffic in telecommunication networks, e.g. [38–40], as well as of vehicular traffic, [41].

As regards previous research on queueing models with BMAP arrivals, but without the dropping function, in [31, 42] the steady-state analysis of models with infinite buffers was carried out. In [43–45], the transient states of queues with infinite buffers were studied. The systems with the finite waiting room were studied in steady state in [46–48], while in transient case in [49, 50], using the potential method (see also [51]).

3 Queue length distribution and losses

We will denote by $\Phi_{n,i}(t, l)$ the probability that the queue length at time t is l , under assumption that at the beginning ($t = 0$) the queue length was n , and the state of the modulating chain was i :

$$\Phi_{n,i}(t, l) = \mathbb{P}(X(t) = l | X(0) = n, J(0) = i),$$

where $0 \leq n \leq b$, $1 \leq i \leq m$, $t > 0$, $0 \leq l \leq b$.

A very important role in the computation of $\Phi_{n,i}(t, l)$ will be played by two functions: $H_{n,k,i,j}(u)$ and $q_n(v, k)$.

Firstly, $H_{n,k,i,j}(u)$ is the counting function of the arrival process filtered by the dropping mechanism. It is defined as the probability that in a system with suspended service, exactly k customers would be accepted into the waiting room in time interval $(0, u]$ and there would be $J(u) = j$, assuming that initially there was $X(0) = n$ and $J(0) = i$. In this definition, the suspended service means that the number of customers present in the queue follows from the BMAP arrivals and the dropping mechanism only; the service is turned off and no served customers leave the system in interval $(0, u]$.

Secondly, $q_n(v, k)$ denotes the probability, that when a batch of v customers arrives to the system in which n customers are already present, exactly k customers from this batch are accepted. The fact that sometimes only a part of the batch is accepted is, of course, a consequence of the assumed dropping policy and finite waiting room.

Both $H_{n,k,i,j}(u)$ and $q_n(v, k)$ will be computed in Section 4.

Having defined $H_{n,k,i,j}(u)$ and $q_n(v, k)$, we can build a system of integral equations for function $\Phi_{n,i}(t, l)$. If at $t = 0$ there is at least one customer in the system, the total probability for formula used with respect to the first arrival time, u , allows us to write:

$$\Phi_{n,i}(t, l) = \sum_{j=1}^m \sum_{k=0}^{b-n} \int_0^t H_{n,k,i,j}(u) \Phi_{n+k-1,j}(t-u, l) dF(u) + \rho_{n,i}(t, l), \quad (6)$$

where $1 \leq n \leq b$, $1 \leq i \leq m$ and

$$\rho_{n,i}(t, l) = \begin{cases} 0, & \text{if } l < n, \\ (1 - F(t)) \sum_{j=1}^m H_{n,l-n,i,j}(t), & \text{if } n \leq l \leq b. \end{cases}$$

In the first sum on the right-hand side of Eq (6), all the events in which the first service completion time, u , happens before t , are taken into account. On the other hand, in function $\rho_{n,i}(t, l)$ all the events in which there is no service completion by the time t are taken into account.

If at $t = 0$ there are no customers in the system, the total probability formula used with respect to the first event in the arrival process, gives:

$$\Phi_{0,i}(t, l) = \sum_{j=1}^m \sum_{k=0}^b \sum_{v=0}^{\infty} \int_0^t \Phi_{k,j}(t-u, l) q_0(v, k) p_i(v, j) \mu_i e^{-\mu_i u} du + \delta_{0i} e^{-\mu_i t}, \quad (7)$$

where $1 \leq i \leq m$ and δ_{ij} equals 1 if $i = j$ and 0 otherwise. As presented in the definition of BMAP, the first event in BMAP happens after an exponentially distributed time with density $\mu_i e^{-\mu_i u}$. The first summand of Eq (7) corresponds to the case $u < t$. The first event in BMAP can be an arrival of a batch of size v or a change of the modulating state into j , or both. If a batch of v customers arrives indeed, k of these customers are accepted with probability $q_0(v, k)$. Therefore, at time u we have k customers in the system and the state of the modulating chain is j . The second summand of Eq (7) corresponds to the case, where the first event in BMAP is after t . In this case the queue length at time t is equal to 0.

Applying the following notation

$$a_{n,k,i,j}(s) = \int_0^{\infty} e^{-st} H_{n,k,i,j}(t) dF(t),$$

$$d_{n,k,i,j}(s) = \int_0^{\infty} e^{-st} H_{n,k,i,j}(t) (1 - F(t)) dt,$$

and using the Laplace transform to Eqs (6) and (7) yields:

$$\phi_{n,i}(s, l) = \sum_{j=1}^m \sum_{k=0}^{b-n} a_{n,k,i,j}(s) \phi_{n+k-1,j}(s, l) + \int_0^{\infty} e^{-st} \rho_{n,i}(t, l) dt, \quad 1 \leq n \leq b, 1 \leq i \leq m, \quad (8)$$

$$\phi_{0,i}(s, l) = \sum_{j=1}^m \sum_{k=0}^b \sum_{v=0}^{\infty} \phi_{k,j}(s, l) q_0(v, k) p_i(v, j) \frac{\mu_i}{s + \mu_i} + \delta_{0i} \frac{1}{s + \mu_i}, \quad 1 \leq i \leq m, \quad (9)$$

respectively. Now we will rewrite Eqs (8) and (9) in the matrix form. The following square matrices will be used for this purpose:

$$A_{n,k}(s) = \left[a_{n,k,i,j}(s) \right]_{i,j}, \quad 1 \leq i, j \leq m, \quad (10)$$

$$\overline{D}_{n,k}(s) = \left[d_{n,k,i,j}(s) \right]_{i,j}, \quad 1 \leq i, j \leq m, \quad (11)$$

$$K_{n,k}(s) = \sum_{v=0}^{\infty} \left[\frac{q_n(v, k) p_i(v, j) \mu_i}{s + \mu_i} \right]_{i,j}, \quad 1 \leq i, j \leq m, \quad (12)$$

$$\mathbf{0} = [0]_{i,j}, \quad 1 \leq i, j \leq m, \quad (13)$$

as well as column vectors:

$$\phi_n(s, l) = [\phi_{n,1}(s, l), \dots, \phi_{n,m}(s, l)]^T, \quad \phi_{n,i}(s, l) = \int_0^\infty e^{-st} \Phi_{n,i}(t, l) dt, \quad (14)$$

$$z(s) = \left[\frac{1}{s + \mu_1}, \dots, \frac{1}{s + \mu_m} \right]^T, \quad (15)$$

$$r_n(s, l) = \begin{cases} \mathbf{0} \cdot \mathbf{1}, & \text{if } l < n, \\ \bar{D}_{n,l-n}(s) \cdot \mathbf{1}, & \text{if } n \leq l \leq b. \end{cases} \quad (16)$$

Using the introduced notation we can rewrite Eqs (8) and (9) as

$$\phi_n(s, l) = \sum_{k=0}^{b-n} A_{n,k}(s) \phi_{n+k-1}(s, l) + r_n(s, l), \quad 1 \leq n \leq b, \quad (17)$$

$$\phi_0(s, l) = \sum_{k=0}^b K_{0,k}(s) \phi_k(s, l) + \delta_{0l} z(s), \quad (18)$$

respectively.

Introducing the following $(b+1)m \times (b+1)m$ matrix:

$$M(s) = [M_{ij}(s)]_{i=0 \dots b, j=0 \dots b},$$

where

$$M_{ij}(s) = \begin{cases} K_{0,0}(s) - I, & \text{if } i = 0, j = 0, \\ K_{0,j}(s), & \text{if } i = 0, 1 \leq j \leq b, \\ A_{i,j-i+1}(s), & \text{if } 1 \leq i \leq b-2, i < j \leq b-1, \\ A_{i,1}(s) - I, & \text{if } i = j, 1 \leq i \leq b-1, \\ A_{i,0}(s), & \text{if } i = j+1, \\ -I, & \text{if } i = b, j = b, \\ \mathbf{0}, & \text{otherwise,} \end{cases} \quad (19)$$

system Eqs (17) and (18) becomes equivalent to

$$M(s) \phi(s, l) = R(s, l), \quad (20)$$

where $R(s, l)$, $\phi(s, l)$ are column vectors of size $(b+1)m$ defined as follows:

$$\phi(s, l) = [\phi_0(s, l), \dots, \phi_b(s, l)]^T, \quad (21)$$

$$R(s, l) = [R_0(s, l), \dots, R_b(s, l)]^T, \quad R_i(s, l) = \begin{cases} -\delta_{0l} z(s), & \text{if } i = 0, \\ -r_i(s, l), & \text{if } 1 \leq i \leq b. \end{cases} \quad (22)$$

Finally, rewriting Eq (20), the following theorem has been proven.

Theorem 1. *The Laplace transform of the queue length distribution at time t of a finite-buffer queue with BMAP arrivals and the dropping function equals:*

$$\phi(s, l) = M^{-1}(s)R(s, l), \quad 0 \leq l \leq b, \quad (23)$$

where $M(s)$ and $R(s, l)$ are given in formulas (19) and (22), respectively.

Using Theorem 1 we can obtain both the transient and stationary distribution of the queue length. For calculations of the transient distribution, we have to exploit one of the several available methods for numerical inversion of the Laplace transform (see e.g. [52–56]). The stationary distribution does not need that. It can be obtained directly from Eq (23) by using the basic properties of the Laplace transform. Namely, we have

$$P_l = \lim_{t \rightarrow \infty} \mathbb{P}(X(t) = l) = \lim_{s \rightarrow 0+} s[M^{-1}(s)R(s, l)]_1, \quad (24)$$

where $[\cdot]_1$ is the first entry of a vector (but it can be any other entry as well—they are all equal in the stationary case).

Having computed the stationary distribution of the queue length, we can compute the loss ratio of the system. The loss ratio is the fraction of customers rejected upon arrival, measured in a long time interval. It will be denoted by L . Its relation with the empty system probability is not difficult to derive. Namely, in a long time interval of length T , the service station is busy for approximately $(1 - P_0)T$ time. As $\int_0^\infty x dF(x)$ is the average service time, the number of customers served in a long interval of length T is approximately $(1 - P_0)T / \int_0^\infty x dF(x)$. On the other hand, there are approximately ΛT new customers arriving to the system in a long interval of length T . Therefore, the loss ratio must be:

$$L = 1 - \frac{(1 - P_0)T}{\int_0^\infty x dF(x)} / (\Lambda T) = 1 - \frac{1 - P_0}{\rho}. \quad (25)$$

4 Auxiliary results

In this section, functions $H_{n,k,i,j}(t)$ and $q_n(v, k)$ for the BMAP process and the dropping mechanism will be computed. This is the last step necessary to use Theorem 1 in numerical calculations.

Denoting

$$h_{n,k,i,j}(s) = \int_0^\infty e^{-st} H_{n,k,i,j}(t) dt,$$

$$h_{n,k}(s) = [h_{n,k,i,j}(s)]_{i,j}, \quad 1 \leq i, j \leq m,$$

we will prove first the following theorem.

Theorem 2. *It holds*

$$h_{n,0}(s) = Y_n(s)Z(s), \quad 0 \leq n \leq b, \quad (26)$$

$$h_{n,k}(s) = Y_n(s) \sum_{w=1}^k K_{n,w}(s) h_{n+w,k-w}(s), \quad k \geq 1, \quad 0 \leq n \leq b, \quad (27)$$

where $h_{n,k}(s)$, $Y_n(s)$ and $Z(s)$ are the following $m \times m$ matrices:

$$Z(s) = \text{diag}(z(s)),$$

$$Y_n(s) = (I - K_{n,0}(s))^{-1}.$$

proof. Consider first, that in time interval $(0, u]$ one or more customers have been accepted to the waiting room. In this case, using the total probability formula with respect to the first event in BMAP, we can write for any $k \geq 1$, $0 \leq n \leq b$, $1 \leq i \leq m$, $1 \leq j \leq m$ the following equation:

$$H_{n,k,i,j}(t) = \sum_{a=1}^m \sum_{v=0}^{\infty} \sum_{w=0}^k \int_0^t q_n(v, w) p_i(v, a) \mu_i e^{-\mu_i u} H_{n+w, k-w, a, j}(t-u) du. \quad (28)$$

The reasoning is somewhat similar to the one presented for Eq (7). The first event in BMAP happened after an exponentially distributed time with density $\mu_i e^{-\mu_i u}$. The first event could have been an arrival of a batch of size v , or a change of the modulating state into a , or both. What is important, the first event must have happened before t . If a batch of v customers arrived, w of these customers were accepted with probability $q_n(w, k)$. Therefore, at time u we had $n+w$ customers in the system and the state of the modulating chain was a .

Now, consider that no customers have been accepted in $(0, u]$. In this case, the first event in $(0, u]$ could have been either a change of the modulating state (without arrival), or an arrival of a batch of customers, of which all customers were rejected. Moreover, it could have happened that the first event was after t . Therefore, we have:

$$H_{n,0,i,j}(t) = \sum_{a=1}^m \sum_{v=0}^{\infty} \int_0^t q_n(v, 0) p_i(v, a) \mu_i e^{-\mu_i u} H_{n,0,a,j}(t-u) du + \delta_{ij} e^{-\mu_i t}, \quad (29)$$

for $0 \leq n \leq b$, $1 \leq i \leq m$, $1 \leq j \leq m$.

Application of the Laplace transform to Eqs (28) and (29) yields:

$$h_{n,k,i,j}(s) = \sum_{a=1}^m \sum_{v=0}^{\infty} \sum_{w=0}^k \frac{q_n(v, w) p_i(v, a) \mu_i}{s + \mu_i} h_{n+w, k-w, a, j}(s), \quad k \geq 1, \quad (30)$$

and

$$h_{n,0,i,j}(s) = \sum_{a=1}^m \sum_{v=0}^{\infty} \frac{q_n(v, 0) p_i(v, a) \mu_i}{s + \mu_i} h_{n,0,a,j}(s) + \frac{\delta_{ij}}{s + \mu_i}, \quad (31)$$

respectively. Applying the matrix notation to Eq (30) we get:

$$h_{n,k}(s) = \sum_{w=0}^k K_{n,w}(s) h_{n+w, k-w}(s), \quad k \geq 1, \quad (32)$$

while Eq (31) yields:

$$h_{n,0}(s) = K_{n,0}(s) h_{n,0}(s) + Z(s). \quad (33)$$

This finishes the proof, as Eq (32) is equivalent to Eq (27), while Eq (33) is equivalent to Eq (26).

As for the numerical calculations of $H_{n,k,i,j}(u)$ values from $h_{n,k,i,j}(s)$, one can use the aforementioned methods of transform inversion [52–56]. We use the Spinelli method of [52].

Calculating $q_n(v, k)$ is straightforward. We obviously have:

$$q_n(v, k) = 0, \quad k > \min \{b - n, v\}, \quad (34)$$

and

$$q_n(0, k) = 0, \quad k > 0, \quad (35)$$

and

$$q_n(v, 0) = (d(n))^v, \quad v > 0, \quad (36)$$

and

$$q_n(0, 0) = 1. \quad (37)$$

Finally, we obtain:

$$q_n(v, k) = d(n)q_n(v - 1, k) + (1 - d(n))q_{n+1}(v - 1, k - 1), \quad v > 0, k > 0. \quad (38)$$

5 Examples

For numerical purposes, we use the following BMAP parameterization, [50]:

$$D_0 = \begin{bmatrix} -0.0499514 & 0.00399715 & 0.00128940 \\ 0.00528656 & -0.0774334 & 0.00528656 \\ 0.00141834 & 0.00141834 & -0.274511 \end{bmatrix},$$

$$D_1 = \begin{bmatrix} 0.0181806 & 0.00141834 & 0.00270775 \\ 0.00141834 & 0.00399715 & 0.00270775 \\ 0.00270775 & 0.00399715 & 0.00657596 \end{bmatrix},$$

$$D_4 = \begin{bmatrix} 0.00141834 & 0.00270775 & 0.00141834 \\ 0.00141834 & 0.0413899 & 0.00141834 \\ 0.00528656 & 0.00141834 & 0.00270775 \end{bmatrix},$$

$$D_{10} = \begin{bmatrix} 0.00483682 & 0.00714007 & 0.00483682 \\ 0.00253357 & 0.00944332 & 0.00253357 \\ 0.00714007 & 0 & 0.241841 \end{bmatrix}.$$

As can be easily computed, the average rate of this BMAP is 1, while the average size of an arriving batch is 8. Moreover, the process is strongly autocorrelated (see [50] for the graph of its autocorrelation function). It is assumed that the capacity of the waiting room is 200, i.e. $b = 200$. If not stated otherwise, the service time is 1.1, which gives the load of the system equal to 110%—the system is overloaded.

To demonstrate capabilities of dropping functions, we will present several functions providing arbitrary values of the average queue length, as well as arbitrary values of the loss ratio. Then a function that keeps two different values of the average queue length, depending on the load of the system, will be presented.

Firstly, how can we obtain an arbitrary value of the average queue length? To show this, let us start with the class of simple linear dropping functions in the form:

$$d_a(n) = \begin{cases} 0, & \text{if } n < 0, \\ a n, & \text{if } 0 \leq n < 200 \text{ and } a n < 1, \\ 1, & \text{if } n \geq 200 \text{ or } a n \geq 1, \end{cases}$$

where a is a parameter. Let us assume also, that the required average queue length is x . If $\mathbb{E}_{d_a}(\Phi)$ denotes the average queue length for dropping function d_a (which can be computed using Theorem 1, then we have to solve numerically the following equation:

$$\mathbb{E}_{d_a}(\Phi) = x,$$

with respect to a . This equation can be solved, for instance, using the bisection method.

In Table 1, five computed values of a are presented. They were obtained assuming that the required average queue length is $x = 10.0, 20.0, 30.0, 40.0$ and 50.0 customers, respectively. In Fig 2, the five linear dropping functions with the values of a from Table 1 are depicted. Finally, in Fig 3 the distributions of the queue length for the first and the last dropping function from Table 1 are depicted. The spikes, which can be observed in low queue ranges, are typical when

Table 1. Performance characteristics of systems with dropping function d_a for five values of a .

a	average queue length	std. dev. queue length	loss ratio
0.0219900	10.0	11.4	45.0%
0.0105600	20.0	21.0	37.3%
0.0067440	30.0	29.6	32.6%
0.0048400	40.0	37.9	29.3%
0.0037315	50.0	45.6	26.9%

doi:10.1371/journal.pone.0150702.t001

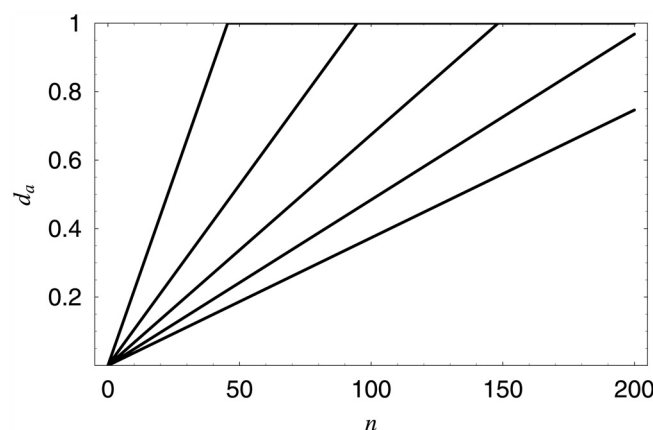


Fig 2. Five linear dropping functions providing the average queue length of 50, 40, 30, 20 and 10 customers, respectively (counting from the bottom).

doi:10.1371/journal.pone.0150702.g002

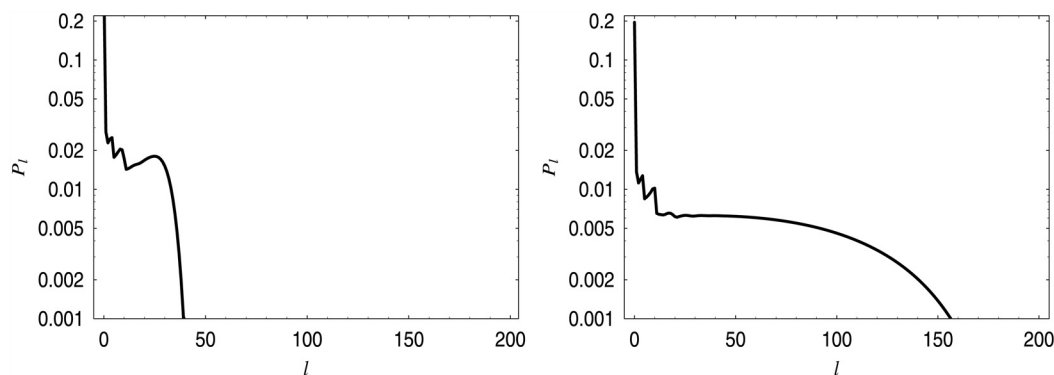


Fig 3. Queue length distributions for dropping function d_a . On the left, for $a = 0.02199$, on the right, for $a = 0.0037315$.

doi:10.1371/journal.pone.0150702.g003

a batch arrivals are involved. They are connected with possible batch sizes—herein 1, 4, and 10—and their combinations.

Arbitrary values of the loss ratio can be achieved in the same way as the queue lengths. To show this, let us consider the following class of quadratic dropping functions:

$$d_b(n) = \begin{cases} 0, & \text{if } n < 0, \\ bn^2, & \text{if } 0 \leq n < 200 \text{ and } bn^2 < 1, \\ 1, & \text{if } n \geq 200 \text{ or } bn^2 \geq 1, \end{cases}$$

where b is a parameter.

Let L_{d_b} denote the loss ratio for dropping function d_b (which can be computed using Eq (25) and Theorem 1) and γ be the required loss ratio. Thus we have to solve numerically the following equation:

$$L_{d_b} = \gamma,$$

with respect to b . This again can be done using bisections.

For instance, the obtained five values of b which provide the loss ratio of $\gamma = 20.0\%$, 25.0% , 30.0% , 35.0% and 40.0% , are presented in in Table 2, respectively. They are accompanied with basic performance characteristics of systems with dropping functions d_b .

In addition, in Fig 4 all quadratic functions with the values of b from Table 2 are depicted, while in Fig 5 the distributions of the queue length for the first and the third dropping function from Table 2 are depicted.

Two questions arise after the first two examples.

Firstly, what are the minimum and maximum values of the average queue length and the loss ratio that can be obtained using the dropping function? To answer this, we have to compute the performance characteristics of the system without the dropping function, or, in other words, with the trivial dropping function in the form:

$$d(n) = \begin{cases} 0, & \text{for } n < 200, \\ 1, & \text{for } n \geq 200. \end{cases}$$

The resulting average queue length obtained from Theorem 1 is 103.1, while the loss ratio is 18.1%. Therefore, using dropping functions we can obtain any average queue length in interval $[0, 103.1]$ and any loss ratio in interval $[18.1\%, 100\%]$.

Table 2. Performance characteristics of systems with dropping function d_b for five values of b .

b	average queue length	std. dev. queue length	loss ratio
0.00000935	88.3	64.8	20.0%
0.00003180	54.4	45.2	25.0%
0.00007900	34.6	31.2	30.0%
0.00018170	22.6	22.0	35.0%
0.00042000	14.6	15.1	40.0%

doi:10.1371/journal.pone.0150702.t002

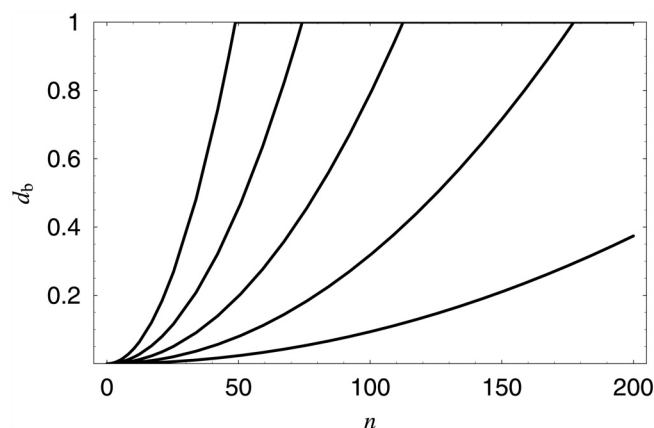


Fig 4. Five dropping functions providing the loss ratio of 20%, 25%, 30%, 35% and 40%, respectively (counting from the bottom).

doi:10.1371/journal.pone.0150702.g004

It is a simple matter to check, that in the second example we could have also used linear dropping functions to obtain loss ratios of 20%, 25%, 30%, etc. (This can be verified using bisections for functions d_a).

This observation leads to the second question. If the simple, linear dropping functions seem to have full control capabilities regarding the average queue length and the loss ratio, then what is the use of more complicated shapes of dropping functions?

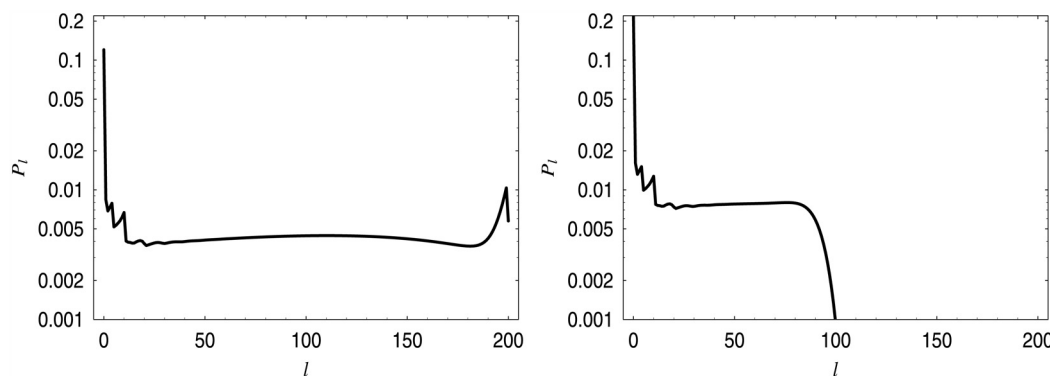


Fig 5. Queue length distributions for dropping function d_b . On the left, for $b = 0.00000935$, on the right, for $b = 0.00007900$.

doi:10.1371/journal.pone.0150702.g005

To answer that, we will show first that there are many different dropping functions providing the same average queue length, but different other performance characteristics (e.g. the loss ratio).

For instance, the following three dropping functions were found, each of them providing the average queue length of 75:

$$d_1(n) = \begin{cases} 0, & \text{for } n < 0, \\ -0.00001(n - 100)^2 + 0.20597, & \text{for } 0 \leq n < 200, \\ 1, & \text{for } n \geq 200, \end{cases}$$

$$d_2(n) = \begin{cases} 0, & \text{for } n < 0, \\ 0.02021018\sqrt{n}, & \text{for } 0 \leq n < 200, \\ 1, & \text{for } n \geq 200, \end{cases}$$

$$d_3(n) = \begin{cases} 0, & \text{for } n \leq 50, \\ 0.00004035(n - 50)^2, & \text{for } 50 < n < 200, \\ 1, & \text{for } n \geq 200. \end{cases}$$

(In the case of dropping functions with two parameters we can set one parameter manually and find the other using bisections). The shapes of these dropping functions are presented in Fig 6. As we can see, they are quite different—one function is non-monotonic, two functions are monotonic, one function is convex, two are concave.

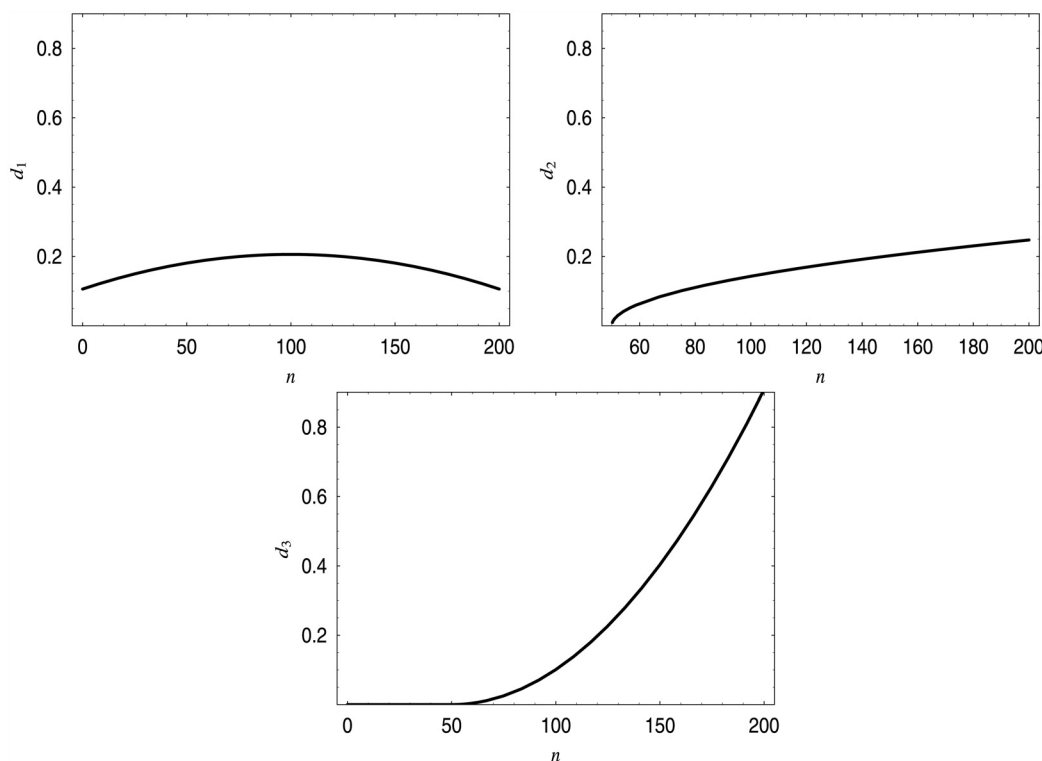


Fig 6. Three different dropping functions providing the average queue length of 75 customers.

doi:10.1371/journal.pone.0150702.g006

Table 3. Performance characteristics of systems with dropping functions d_1 – d_3 .

dropping function	average queue length	std. dev. queue length	loss ratio
d_1	75.0	65.6	23.9%
d_2	75.0	63.5	22.8%
d_3	75.0	56.5	21.2%

doi:10.1371/journal.pone.0150702.t003

In [Table 3](#), the performance characteristics of the systems with dropping functions d_1 – d_3 are presented. As we can see, the average queue length is common, but other characteristics are different.

In [Fig 7](#), distributions of the queue length in systems with dropping functions d_1 – d_3 are presented.

Similarly, there are many different dropping functions providing the same value of the loss ratio, but different other performance characteristics (e.g. the queue length). For instance, in addition to the first dropping function in [Table 2](#), the following three different dropping functions provide the customer loss ratio of 20%:

$$d_4(n) = \begin{cases} 0, & \text{for } n < 0, \\ 0.22606729e^{-(n-120)^2/1800}, & \text{for } 0 \leq n < 200, \\ 1, & \text{for } n \geq 200, \end{cases}$$

$$d_5(n) = \begin{cases} 0, & \text{for } n < 0, \\ (n/12000)(n/12500)|\sin(n)| + 0.063 & \text{for } 0 \leq n < 200, \\ 1, & \text{for } n \geq 200, \end{cases}$$

$$d_6(n) = \begin{cases} 0, & \text{for } n < 0, \\ 0.0005n & \text{for } 0 \leq n < 120, \\ 0.0063n - 0.696 & \text{for } 120 \leq n < 200, \\ 1, & \text{for } n \geq 200. \end{cases}$$

The shapes of these functions are depicted in [Fig 8](#), while in [Table 4](#) the performance characteristics of the systems exploiting them are shown. In [Fig 9](#), distributions of the queue length in systems with dropping functions d_4 – d_6 are depicted.

As we can see, the queue length and its deviation can vary, even if the loss ratio is common in all three systems.

We can conclude that using different shapes of dropping function opens the possibility to control more than one performance characteristics at the same time. (However, every next characteristic in a smaller interval).

In the final example, the dropping function was designed in such a way, that it provides the average queue length of 30 customers when the load of the system is 90% (underloaded system), and 60 customers—when the load is 110% (overloaded system).

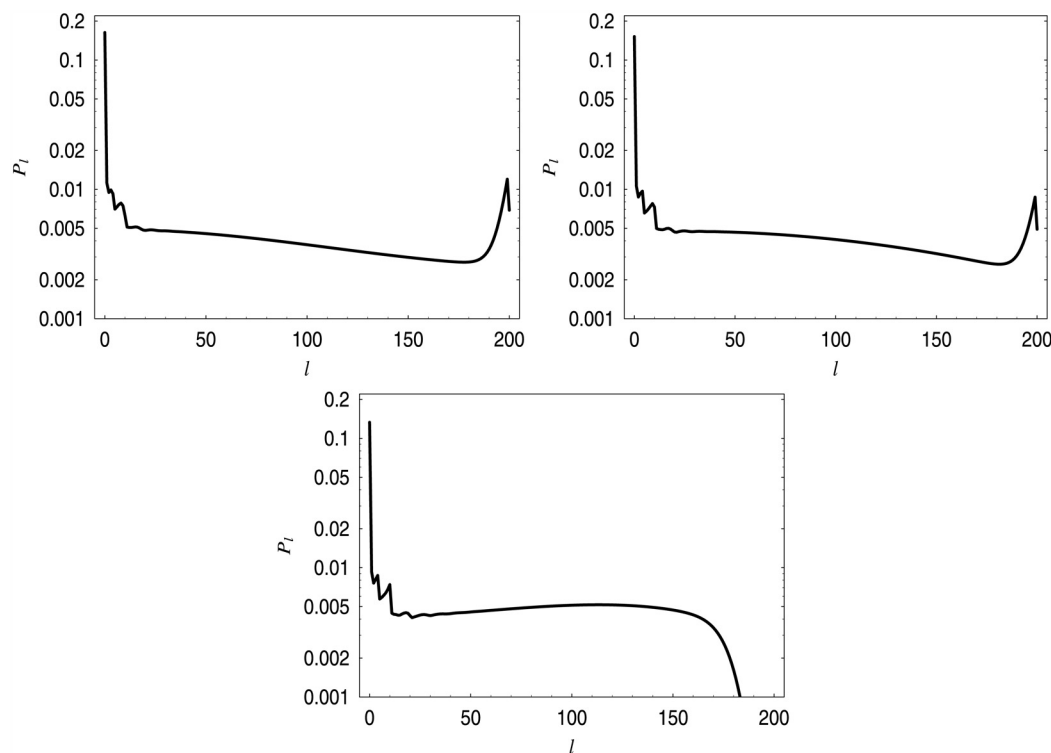


Fig 7. Queue length distributions for dropping functions d_1 – d_3 . Distribution for d_1 in the upper-left corner, for d_2 in the upper-right corner, for d_3 in the bottom.

doi:10.1371/journal.pone.0150702.g007

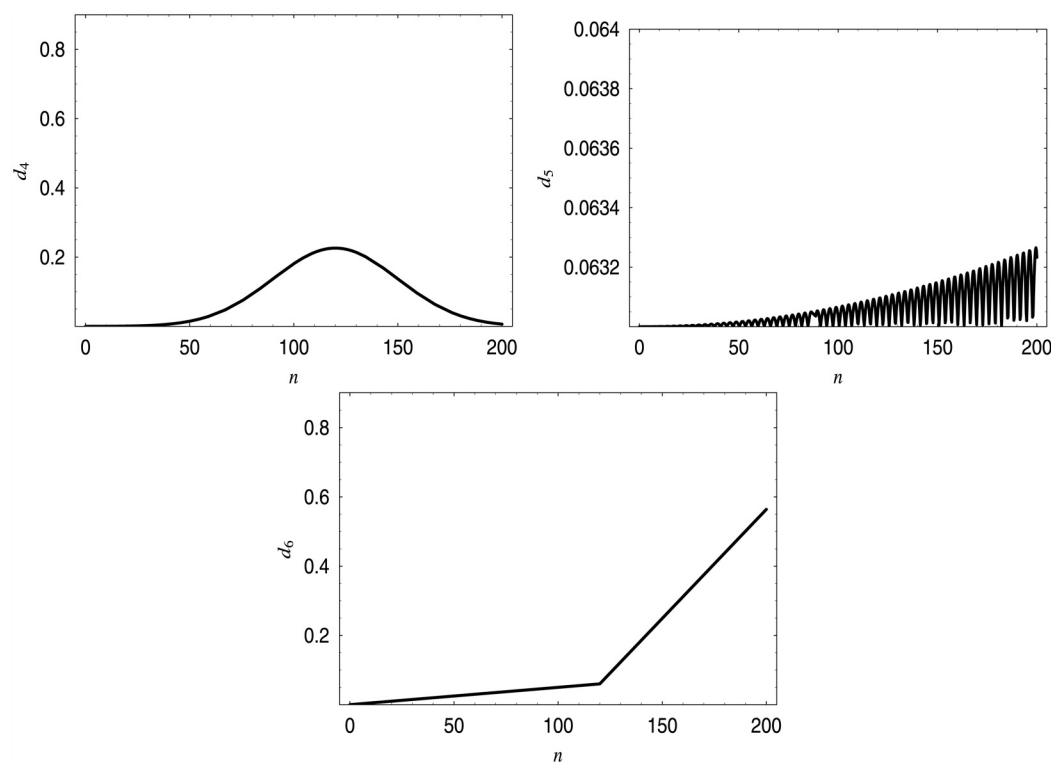


Fig 8. Three different dropping functions providing the loss ratio of 20% of customers.

doi:10.1371/journal.pone.0150702.g008

Table 4. Performance characteristics of systems with dropping functions d_4 – d_6 .

dropping function	average queue length	std. dev. queue length	loss ratio
d_4	88.1	66.6	20.0%
d_5	94.2	68.7	20.0%
d_6	86.8	63.3	20.0%

doi:10.1371/journal.pone.0150702.t004

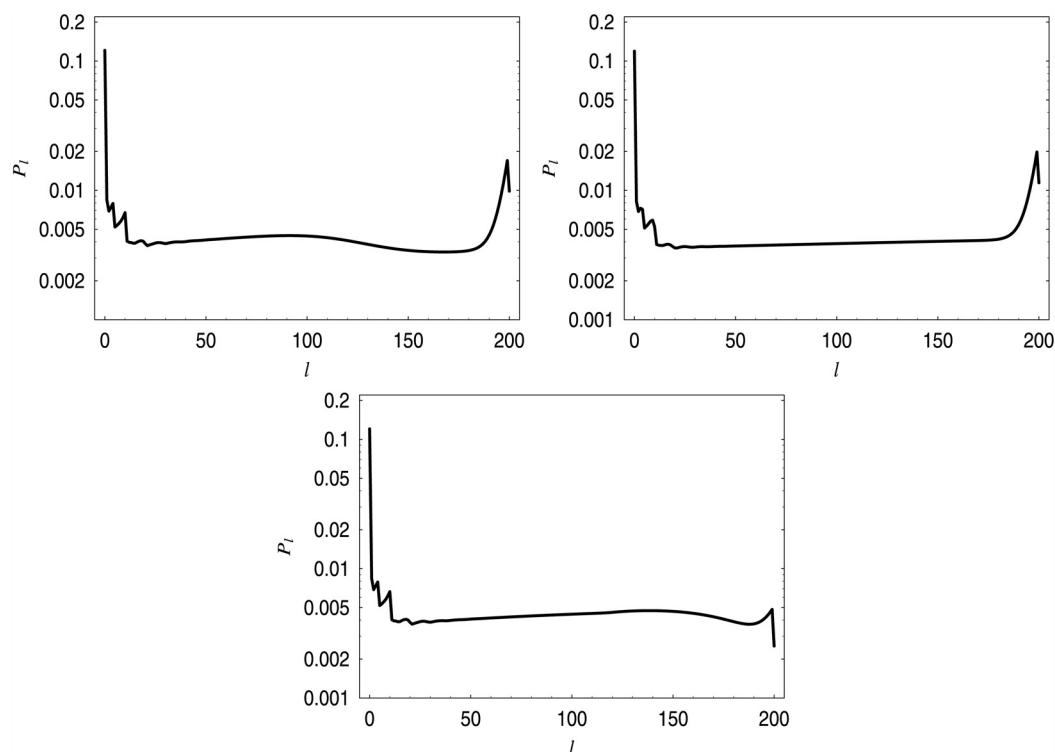


Fig 9. Queue length distributions for dropping functions d_4 – d_6 . Distribution for d_4 in the upper-left corner, for d_5 in the upper-right corner, for d_6 in the bottom.

doi:10.1371/journal.pone.0150702.g009

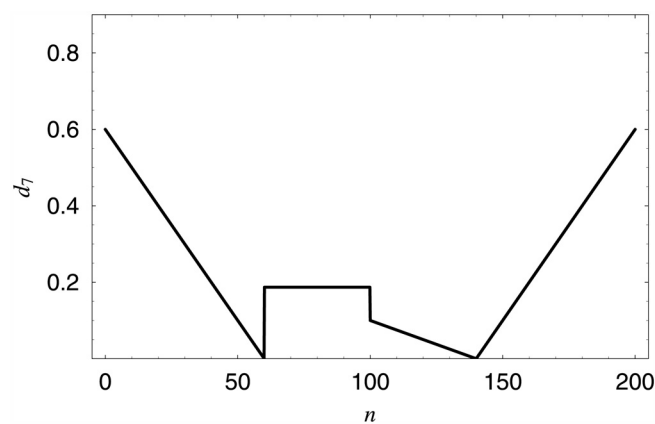


Fig 10. Dropping function providing the average queue length of 30 customers for $\rho = 90\%$, and 60 customers for $\rho = 110\%$.

doi:10.1371/journal.pone.0150702.g010

Table 5. Performance characteristics of systems with dropping function d_7 and two different loads.

dropping function	average queue length	std. dev. queue length	loss ratio
$d_7, \rho = 90\%$	30.0	50.1	39.8%
$d_7, \rho = 110\%$	60.0	65.9	35.3%

doi:10.1371/journal.pone.0150702.t005

To find the parameters of the dropping function in this case, the system of two equations had to be solved numerically. The obtained function has the following form:

$$d_7(n) = \begin{cases} 0, & \text{for } n < 0, \\ -0.01 + 0.6n & \text{for } 0 \leq n < 60, \\ 0.187 & \text{for } 60 \leq n < 100, \\ -0.0025(n - 140) & \text{for } 100 \leq n < 140, \\ 0.01(n - 140) & \text{for } 140 \leq n < 200, \\ 1, & \text{for } n \geq 200, \end{cases}$$

and is depicted in Fig 10.

The performance characteristics for d_7 and different loads are given in Table 5, while the distributions of the queue length are shown in Fig 11.

This example demonstrates the possibility of designing the dropping function in such a way, that for different loads it keeps performance characteristics at some arbitrary levels.

6 Conclusions

We presented an analysis of the queueing system with the dropping function of arbitrary type, arbitrary distribution of the service time, and a very general customer arrival process, which allows for modeling of autocorrelation, batch arrivals and arbitrary shape of the interarrival time distribution. We obtained the distribution of the queue length both in transient and stationary regime, computed the loss ratio and presented several numerical examples.

As for the future work and open questions, probably the most interesting is the question on the stability condition for the systems with BMAP arrivals and infinite waiting room. Naturally, if the dropping function is applied in a system with infinite waiting room, the stability

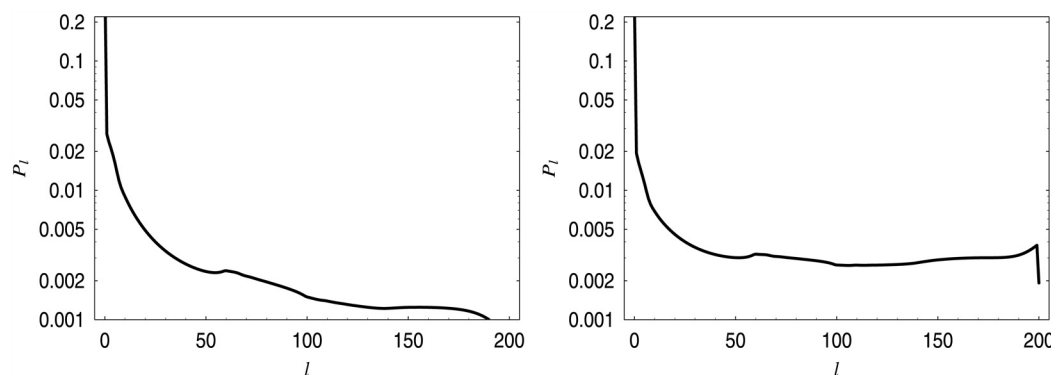


Fig 11. Queue length distributions for dropping function d_7 . On the left, for $\rho = 90\%$, on the right, for $\rho = 110\%$.

doi:10.1371/journal.pone.0150702.g011

condition is not $\rho < 1$ any more. For instance, if $\rho = 2$ and the dropping function is $d(n) = 0.51$ for every n , then the system is obviously stable. But would it be stable for a more sophisticated dropping function, say $d(n) = 0.51 - \frac{1}{n}$, $n \geq 2$? Does the stability depend only on ρ and the dropping function or on the BMAP structure as well?

Author Contributions

Conceived and designed the experiments: AC PM. Performed the experiments: AC PM. Analyzed the data: AC PM. Contributed reagents/materials/analysis tools: AC PM. Wrote the paper: AC PM.

References

1. Abolnikov L, Dshalalow J, Dukhovny A. On stochastic processes in a multilevel control bulk queueing system. *Stochastic Analysis and Applications*. 1992; 10(2):155–179. doi: [10.1080/07362999208809261](https://doi.org/10.1080/07362999208809261)
2. Fakinos D. The G/G/1 (LCFS/P) queue with service depending on queue size. *European Journal of Operational Research*, 192; 59(2):303–307. doi: [10.1016/0377-2217\(92\)90145-Y](https://doi.org/10.1016/0377-2217(92)90145-Y)
3. Chydzinski A. The oscillating queue with finite buffer. *Performance Evaluation*. 2004; 57(3):341–355. doi: [10.1016/j.peva.2004.02.001](https://doi.org/10.1016/j.peva.2004.02.001)
4. Pacheco A, Ribeiro H. Consecutive customer losses in oscillating $GI^X/M/1$ systems with state dependent services rates. *Annals of Operations Research*. 2008; 162(1):143–158. doi: [10.1007/s10479-008-0313-6](https://doi.org/10.1007/s10479-008-0313-6)
5. Dshalalow J. Queueing systems with state-dependent parameters. In: *Frontiers in Queueing: Models and Applications in Science and Engineering*, ed. Dshalalow J, CRC Press, 1997.
6. Chydzinski A. The M–M/G/1–type oscillating systems. *Cybernetics and Systems Analysis*. 2003; 39(2):316–324. doi: [10.1023/A:1024755710076](https://doi.org/10.1023/A:1024755710076)
7. Bekker R. Queues with Levy input and hysteretic control. *Queueing Systems*. 2009; 63(1):281–299. doi: [10.1007/s11134-009-9123-z](https://doi.org/10.1007/s11134-009-9123-z)
8. Floyd S, Jacobson V. Random early detection gateways for congestion avoidance. *IEEE/ACM Transactions on Networking*. 1993; 1:397–413. doi: [10.1109/90.251892](https://doi.org/10.1109/90.251892)
9. Rosolen V, Bonaventure O, Leduc G. A RED discard strategy for ATM networks and its performance evaluation with TCP/IP traffic. *ACM SIGCOMM Computer Communication Review*. 1999; 29(3):23–43. doi: [10.1145/505724.505728](https://doi.org/10.1145/505724.505728)
10. Athuraliya S, Low SH, Li VH, Yin Q. REM: active queue management. *IEEE Network*. 2001; 15(3):48–53. doi: [10.1109/65.923940](https://doi.org/10.1109/65.923940)
11. Zhou K, Yeung KL, Li VOK. Nonlinear RED: A simple yet efficient active queue management scheme. *Computer Networks*. 2006; 50(18):3784–3794. doi: [10.1016/j.comnet.2006.04.007](https://doi.org/10.1016/j.comnet.2006.04.007)
12. Hollot CV, Misra V, Towsley D, Gong W. Analysis and design of controllers for AQM routers supporting TCP flows. *IEEE Transactions on Automatic Control*. 2002; 47(6):945–959. doi: [10.1109/TAC.2002.1008360](https://doi.org/10.1109/TAC.2002.1008360)
13. Wu W, Ren Y, Shan X. A self-configuring PI controller for active queue management. In: *Proc. Asia-Pacific Conference on Communications*. 2001.
14. Feng W, Shin KG, Kandlur DD, Saha D. The BLUE active queue management algorithms. *IEEE/ACM Transactions on Networking*. 2002; 10(4):513–528. doi: [10.1109/TNET.2002.801399](https://doi.org/10.1109/TNET.2002.801399)
15. Wyrowski B, Zukerman M. GREEN: an active queue management algorithm for a self managed Internet. In: *Proceeding of IEEE International Conference on Communications*. 2002; 4:2368–2372.
16. Kunniyur SS, Srikant R. An adaptive virtual queue (AVQ) algorithm for active queue management. *IEEE/ACM Transactions on Networking*. 2004; 12(2):286–299. doi: [10.1109/TNET.2004.826291](https://doi.org/10.1109/TNET.2004.826291)
17. Na Z, Guo Q, Gao Z, Zhen J, Wang C. A novel adaptive traffic prediction AQM algorithm *Journal. Telecommunication Systems*. 2012; 49(1):149–160. doi: [10.1007/s11235-010-9359-2](https://doi.org/10.1007/s11235-010-9359-2)
18. Suzer MH, Kang KD, Basaran C. Active queue management via event-driven feedback control. *Computer Communications*. 2012; 35(4):517–529. doi: [10.1016/j.comcom.2011.12.001](https://doi.org/10.1016/j.comcom.2011.12.001)
19. Farzaneh N, Reza Monsefi R, Yaghmaee MH, Mohajerzadeh AH. A novel congestion control protocol with AQM support for IP-based networks. *Telecommunication Systems*. 2013; 52(1):229–244. doi: [10.1007/s11235-011-9559-4](https://doi.org/10.1007/s11235-011-9559-4)

20. Chrost L, Chydzinski A. On the deterministic approach to active queue management. *Telecommunication Systems*. 2015;. doi: [10.1007/s11235-015-9969-9](https://doi.org/10.1007/s11235-015-9969-9)
21. Bonald T, May M, Bolot JC. Analytic evaluation of RED performance. In: *Proceedings of INFOCOM*. 2000, 1415–1424.
22. Hao W, Wei Y. An Extended $G^X/M/1/N$ Queueing Model for Evaluating the Performance of AQM Algorithms with Aggregate Traffic. In: *Proceedings of ICCNMC, Lecture Notes in Computer Science*. 2005; 3619:395–414. doi: [10.1007/11534310_43](https://doi.org/10.1007/11534310_43)
23. Chydzinski A, Chrost L. Analysis of AQM queues with queue-size based packet dropping. *International Journal of Applied Mathematics and Computer Science*. 2011; 21(3):567–577. doi: [10.2478/v10006-011-0045-7](https://doi.org/10.2478/v10006-011-0045-7)
24. Kempa WM. On non-stationary queue-size distribution in a finite-buffer queue controlled by a dropping function. In: *Proc. International Conference on Informatics, Spisska Nova Ves, Nov. 2013*; 67–72.
25. Kempa WM. A direct approach to transient queue-size distribution in a finite-buffer queue with AQM. *Applied Mathematics & Information Sciences*. 2013; 7(3):909–915. doi: [10.12785/amis/070308](https://doi.org/10.12785/amis/070308)
26. Tikhonenko O, Kempa WM. Queue-size distribution in M/G/1-type system with bounded capacity and packet dropping. *Communications in Computer and Information Science*. 2013; 356:177–186. doi: [10.1007/978-3-642-35980-4_20](https://doi.org/10.1007/978-3-642-35980-4_20)
27. Leland W, Taqqu M, Willinger W, Wilson D. On the self-similar nature of ethernet traffic (extended version). *IEEE/ACM Transactions on Networking*. 1994; 2(1):1–15. doi: [10.1109/90.282603](https://doi.org/10.1109/90.282603)
28. Paxson V, Floyd S. Wide-area traffic: the failure of Poisson modeling. *IEEE/ACM Trans. Netw.* 1995; 3(3):226–244. doi: [10.1109/90.392383](https://doi.org/10.1109/90.392383)
29. Crovella M, Bestavros A. Self-similarity in World Wide Web traffic: Evidence and possible causes. *IEEE/ACM Transactions on Networking*. 1997; 5(6):835–846. doi: [10.1109/90.650143](https://doi.org/10.1109/90.650143)
30. Chydzinski A. Transient Analysis of the MMPP/G/1/K Queue. *Telecommunication Systems*. 2006; 32(4):247–262. doi: [10.1007/s11235-006-9001-5](https://doi.org/10.1007/s11235-006-9001-5)
31. Lucantoni DM. New results on the single server queue with a batch Markovian arrival process. *Commun. Stat., Stochastic Models*. 1991; 7(1):1–46. doi: [10.1080/15326349108807174](https://doi.org/10.1080/15326349108807174)
32. Salvador P, Pacheco A, Valadas R. Modeling IP traffic: joint characterization of packet arrivals and packet sizes using BMAPs. *Computer Networks*. 2004; 44:335–352. doi: [10.1016/j.comnet.2003.10.004](https://doi.org/10.1016/j.comnet.2003.10.004)
33. Klemm A, Lindemann C, Lohmann M. Modeling IP traffic using the batch Markovian arrival process. *Performance Evaluation*. 2003; 54(2). doi: [10.1016/S0166-5316\(03\)00067-1](https://doi.org/10.1016/S0166-5316(03)00067-1)
34. Breuer L. An EM algorithm for batch Markovian arrival processes and its comparison to a simpler estimation procedure. *Annals of Operations Research*. 2002; 112:123–138. doi: [10.1023/A:1020981005544](https://doi.org/10.1023/A:1020981005544)
35. Buchholz P. An EM-Algorithm for MAP Fitting from Real Traffic Data. In: Kemper P., Sanders W. H. (eds.), *Computer Performance Evaluation Modelling Techniques and Tools, Lecture Notes in Computer Science*. 2003; 2794:218–236.
36. Neuts MF. A Versatile Markovian Point Process. *Journal of Applied Probability*. 1997; 14:764–779.
37. Chakravarthy S. The Batch Markovian Arrival Process: A Review and Future Work. *Advances in Probability and Stochastic Processes*. In: Eds. Krishnamoorthy A, Raju N, Ramaswami V. 2001, 21–49.
38. Klemm A, Lindemann C, Lohmann M. Traffic Modeling and Characterization for UMTS Networks. In: *Proc. GLOBECOM*. 2001; 1741–1746.
39. Kang SH, Kim YH, Sung DK, Choi BD. An application of Markovian arrival process (MAP) to modeling superposed ATM cell streams. *IEEE Transactions on Communications*. 2002; 50(4):633–642. doi: [10.1109/26.996078](https://doi.org/10.1109/26.996078)
40. Fiems D, Steyaert B, Bruneel H. Performance of a Partially Shared Buffer with Correlated Arrivals. *Lecture Notes in Computer Science*. 2007; 4479:1196–1200. doi: [10.1007/978-3-540-72606-7_112](https://doi.org/10.1007/978-3-540-72606-7_112)
41. Alfa AS, Neuts MF. Modelling vehicular traffic using the discrete time Markovian arrival process. *Transportation Science*. 1995; 29:109–117. doi: [10.1287/trsc.29.2.109](https://doi.org/10.1287/trsc.29.2.109)
42. Lee HW, Park NI, Jeon J. A new approach to the queue length and waiting time of BMAP/G/1 queues. *Computers & Operations Research*. 2003; 30:2021–2045. doi: [10.1016/S0305-0548\(02\)00122-3](https://doi.org/10.1016/S0305-0548(02)00122-3)
43. Lucantoni DM, Choudhury GL, Whitt W. The transient BMAP/G/1 queue. *Commun. Stat., Stochastic Models*. 1994; 10(1):145–182. doi: [10.1080/15326349408807291](https://doi.org/10.1080/15326349408807291)
44. Lucantoni D. Further transient analysis of the BMAP/G/1 queue. *Commun. Stat., Stochastic Models*. 1998; 14(1–2):461–478. doi: [10.1080/15326349808807482](https://doi.org/10.1080/15326349808807482)

45. Le Ny LM, Sericola B. Transient Analysis of the BMAP/PH/1 Queue. *International Journal of Simulation: Systems, Science & Technology. Special Issue on Analytical & Stochastic Modeling Techniques*. 2002; 3(3–4).
46. Blondia C. The finite Capacity N/G/1 Queue. *Communications in Statistics: Stochastic Models*. 1989; 5(2):273–294.
47. Gouweleeuw FN. Calculating the loss probability in a BMAP/G/1/N+1 queue. *Commun. Stat., Stochastic Models*. 1996; 12(3):473–492. doi: [10.1080/15326349608807395](https://doi.org/10.1080/15326349608807395)
48. Dudin AN, Shaban AA, Klimenok VI. Analysis of a queue in the BMAP/G/1/N system. *International Journal of Simulation: Systems, Science and Technology*. 2005; 6(1–2):13–23.
49. Chydzinski A. Queue Size in a BMAP Queue with Finite Buffer. *Lecture Notes in Computer Science*. 2006; 4003:200–210. doi: [10.1007/11759355_20](https://doi.org/10.1007/11759355_20)
50. Chydzinski A, Adamczyk B. Transient and stationary losses in a finite-buffer queue with batch arrivals. *Mathematical Problems in Engineering*. 2012; 326830:1–17. doi: [10.1155/2012/326830](https://doi.org/10.1155/2012/326830)
51. Chydzinski A. Duration of the buffer overflow period in a batch arrival queue. *Performance Evaluation*. 2006; 63(4–5):493–508. doi: [10.1016/j.peva.2005.06.005](https://doi.org/10.1016/j.peva.2005.06.005)
52. Spinelli RA. Numerical Inversion of Laplace Transforms, *SIAM J. Num. Anal.* 1996; 3:636–649. doi: [10.1137/0703056](https://doi.org/10.1137/0703056)
53. Lyness JN, Giunta G. A modification of the Weeks method for numerical inversion of the Laplace transform. *Math. Comput.* 1986; 47(175):313–322.
54. Abate J, Choudhury GL, Whitt W. An introduction to numerical transform inversion and its application to probability models. *Computational Probability*. 2000, 257–323. doi: [10.1007/978-1-4757-4828-4_8](https://doi.org/10.1007/978-1-4757-4828-4_8)
55. Abate J, Valko PP. Multi-precision Laplace transform inversion, *Int. J. Numer. Meth. Engng.* 2004; 60:979–993.
56. Cohen AM. *Numerical Methods for Laplace Transform Inversion*, Springer, New York, 2007.